WORKSHOP ON DISTANCE GEOMETRY AND APPLICATIONS 2013

Edited by

Alessandro Andrioni
University of Campinas, Campinas, Brazil

Rosiane de Freitas
IComp, Federal University of Amazonas, Manaus, Brazil

Carlile Lavor
University of Campinas, Campinas, Brazil

Leo Liberti
LIX, École Polytechnique, Palaiseau, France
IBM "T. J. Watson" Research Center, Yorktown Heights, USA

Nelson Maculan
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

Antonio Mucherino
IRISA, University of Rennes I, Rennes, France

Preface

Welcome to the workshop on Distance Geometry and Applications (DGA13)! This is, to the best of our knowledge, the first workshop wholly dedicated to Distance Geometry (DG).

DG sets the concept of distance at the basis of Euclidean geometry. The fundamental problem of DG is an inverse problem, i.e., finding a set of points in Euclidean space, such that a given subset of their pairwise distances are equal to some given values. Besides the beauty of the mathematical theory associated to DG, the interest in this research topic is explained by the richness and variety of its applications. To cite the main ones: structural biology, mobile sensor networks, statics, analysis of data, robotics, clock synchronization, astronomy and music.

Some time ago, we noticed that the academic community working on DG is fragmented. It seems that the primary interest is in the applications, rather than the theory and methods that stand behind it. Researchers focusing on molecular structures publish regularly in bioinformatics and global optimization journals; those focusing on sensor networks often publish in network-related as well as on SIAM journals; those working on structural rigidity mostly publish on graph theory and combinatorics journals. Other communities (for example in robotics or data analysis) target yet other journals. All of us send papers to a very diverse variety of conferences: discrete mathematics, computer science, network technology, robotics, statistics and more. Although these boundaries are far from strict, the most visible effect of this fragmentation is the different formalizations of very similar ideas across the application fields. Although it is certainly very positive to have such a diverse and seemingly all-encompassing application range at our disposal, we feel we can all profit from referring to a somewhat better defined "DG community".

This workshop is part of a set of actions some of us are carrying out as an effort towards shaping the DG community: an edited book and several surveys were recently published (one will appear in SIAM Review). We hope this is just the beginning, and shall work towards making DGA2013 the first of a long sequence. A special issue of Discrete Applied Mathematics (DAM) will be dedicated to this workshop. All participants are invited to submit full papers.

We wish to thank the invited speakers, the scientific and local organizing committee members, the referees of the contributed papers, as well as our funding sponsors: CNPq, CAPES, FAPESP, FAPEAM, EMC2, MCM, iNdT, SECTI, Ecole Polytechnique (France).

Alessandro Andrioni (Campinas, Brazil)
Rosiane de Freitas (Manaus, Brazil)
Carlile Lavor (Campinas, Brazil)
Leo Liberti (Yorktown Heights, USA)
Nelson Maculan (Rio de Janeiro, Brazil)
Antonio Mucherino (Rennes, France)

Scientific Committee

Alberto Krone-Martins	Universidade de Lisboa, Portugal
Antonio Mucherino	Université de Rennes 1, France
Bruce Donald	Duke University, USA
Celina Herrera de Figueiredo	Universidade Federal do Rio de Janeiro, Brazil
Daniel Aloise	Universidade Federal do Rio Grande do Norte, Brazil
Deok-Soo Kim	Hanyang University, South Korea
Di Wu	Western Kentucky University, USA
Dieter Rautenbach	Universität Ulm, Germany
Fabio Schoen	Universitá di Firenze, Italy
Guilherme Fonseca	Universidade Federal do Estado do Rio de Janeiro, Brazil
Guilherme Liberali	Erasmus University, The Netherlands
Hans Colonius	Universität Oldenburg, Germany
Jayme Szwarcfiter	Universidade Federal do Rio de Janeiro, Brazil
Jose Mario Martinez	Universidade Estadual de Campinas, Brazil
Julius Zilinskas	Vilnius University, Lithuania
Kelson Mota	Universidade Federal do Amazonas, Brazil
Kim-Chuan Toh	National University of Singapore, Singapore
Leo Liberti	École Polytechnique, France; and IBM TJ Watson Research Center, USA
Lu Yang	East China Normal University, China
Manfred Sippl	University of Salzburg, Austria
Marcelo Firer	Universidade Estadual de Campinas, Brazil
Michael Nilges	Institut Pasteur, France
Michel Petitjean	Université Paris 7, France
Mitre Costa Dourado	Universidade Federal do Rio de Janeiro, Brazil
Monique Laurent	CWI and Tilburg University, The Nertherlands
Nair Abreu	Universidade Federal do Rio de Janeiro, Brazil
Ramachrisna Teixeira	Universidade de São Paulo, Brazil
Raphael Machado	Instituto Nacional de Metrologia, Qualidade e Tecnologia, Brazil
Rosiane de Freitas	Universidade Federal do Amazonas, Brazil
Rumen Andonov	Université de Rennes 1, France
Sueli Costa	Universidade Estadual de Campinas, Brazil
Tibérius Bonates	Universidade Federal do Semi-Árido, Brazil

Sponsors

Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq	Brazil
Comissão de Aperfeiçoamento de Pessoal de Nível Superior - CAPES	Brazil
École Polytechnique	France
Fundação de Amparo à Pesquisa do Estado do Amazonas - FAPEAM	Brazil
Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP	Brazil

Contents

Invited Speakers

Extended Abstracts

INVITED SPEAKERS

DGA 2013, pp. 3 – 3.

Discrete conformational states and the energy landscape of proteins: demand for computational methods for structure calculation of excited states

Fábio Almeida[1]

[1]*Federal University of Rio de Janeiro, Brazil*

Abstract Proteins are dynamic entities that move in a hierarchy of timescales that goes from picoseconds to seconds. The energy landscape of a protein defines the thermally accessible conformational states. The energy of each state defines the relative population and the energy of the transition-state defines protein dynamics. Motions that occur in microseconds to seconds are a result of energy barriers that are bigger than thermal energy. They are known as conformational exchange and define biologically relevant processes that are frequently involved in binding and allostery. In this talk we will show the importance of computational methods to calculate the structure of discrete excited states and to evaluate the energy landscape of proteins. We will show how the mapping of regions in conformational exchange leaded to the discovery of membrane binding sites in plant defensins. Defensins share the same fold, but display significant difference in dynamics. Structure of excited states reveals the reason of success of Cys-knot folding of defensins. We will also show how water-permeable excited states contribute to proton transfer and catalysis of thioredoxins.

DGA 2013, pp. 5 – 5.

An Alternative Approach to Distance Geometry Using L^∞ Distances

Gordon Crippen[1]

[1] *University of Michigan, USA*

Abstract A standard task in distance geometry is to calculate one or more sets of Cartesian coordinates for a set of points that satisfy given geometric constraints, such as bounds on some of the L^2 distances. Using instead L^∞ distances is attractive because distance constraints can be expressed as simple linear bounds on coordinates. Likewise, a given matrix of L^∞ distances can be rather directly converted to coordinates for the points. It can happen that multiple sets of coordinates correspond precisely to the same matrix of L^∞ distances, but the L^2 distances vary only modestly. Practical examples are given of calculating protein conformations from the sorts of distance constraints that one can obtain from nuclear magnetic resonance experiments.

DGA 2013, pp. 7 – 7.

Distances and Geometry

Michel-Marie Deza[1]

[1] *École Normale Supérieure, France*

Abstract It is a tutorial-like survey, focused on definitions, of main distances used in Geometry. The Contents is: 1-) Application example: distances in Data Clustering, 2-) Birdview on metric spaces (Metric repairs, Generalizations of metric spaces, Metric transforms, Dimension, radii and other numeric invariants of metric spaces, Relevant notions: special subsets, mappings, completeness, Main classes of metric spaces), 3-) Example: distance geometry and similar graph problems, 4-) Metric/Geodesic Geometry: curves, convexity etc., and 5-) Other geometric distances (Projective and A ne Geometry, Distances on surfaces and knots, Distances on convex bodies and cones).

DGA 2013, pp. 9 – 9.

The self-calibrating solutions of all-sky space astrometry

Floor van Leeuwen[1]

[1] *University of Cambridge, England*

Abstract In space astrometry we determine positions of stars on the sky as a function of time, to derive their distances, distribution and motions in space. This is done by measuring at very high accuracy large angular distances between stars on the sky over a period of several years. One such experiment is finished (the Hipparcos satellite mission), and one is to be launched later this year (the Gaia satellite mission). Although this is not directly an application of distance geometry, the solution mechanisms that transfer the 13 million one-dimensional measurements of large arcs on the sky, collected over a three-year period by the Hipparcos satellite, to a final catalogue of positional information for 118000 (moving) stars, is based on similar processes and faces similar problems. I will present a brief background of space astrometry, the way it is done, and its possibilities and limitations. Then I will show the basic measurements and their characteristic features, and how one gets from these measurements to a full-sky catalogue of positional information. In particular the measurement of the stellar parallax and the overall importance in astrophysics of distance measurements will be described. Finally, some statistical properties of the catalogue are shown for a case where the calibration of the instrumental effects has not been completely successful.

DGA 2013, pp. 11 – 11.

Distance Geometry: the past and the present

Leo Liberti[1,2]

[1] *École Polytechnique, France*

[2] *IBM TJ Watson Research Center, USA*

Abstract We present an overview of the themes and trends in Distance Geometry (DG) from its birth to current research. Although DG appeared formally in the 1930s, some applications delve their roots in more ancient times. Famous mathematicians (such as Godel) worked in DG. Nowadays, DG is being developed by researchers in the following application fields: proteomics, wireless networks, statics, robotics and statistics. Techniques for solving DG problems include local and global optimization, semi-definite programming, differential equations, polynomial rings, combinatorial analysis, group theory, oriented matroids and others.

DGA 2013, pp. 13 – 13.

The protein structures as constrained geometric objects

Thérèse Malliavin[1]

[1]*Institut Pasteur, France*

Abstract Proteins are polypeptides of amino-acids involved in most of the biological processes. In the last 50 years, the study of their structures at the molecular level revolutionized the vision of biology. The three-dimensional structures of the proteins are geometric objects defined by the relative positions of the protein atoms. The determination of these objects attract much interest as it is closely related to the identification of their biological function. These objects can be determined from inter-atomic distances measured by Nuclear Magnetic Resonance (NMR), and the lack of precision of the measure produces variability in the protein structure. But the variability of the protein structure does not only come from measurement imprecision, but is also due to protein conformational equilibrium, which plays a major role into biological processes. Due to this intrinsic variability, the protein structure is calculated by repeating the same optimization procedure with changing the initialization seed. The algorithm for this iterative procedure stops when the repeated protein structures are sufficiently superimposed to each other. The choice of the required level of superimposition from a Bayesian analysis of the structure determination problem permits to obtain a least-biased geometry in agreement with the best measure fit. As the protein structures are 3D Euclidean geometric objects, the inter-atomic distances are linked by triangle inequalities. In that way, the distances can be hierarchized through the estimation of their redundancy. I shall show that this redundancy can be related to experimental observations on the energetic bases of protein stability, and to protein dynamics and function.

DGA 2013, pp. 15 – 15.

Discretization Orders for Distance Geometry

Antonio Mucherino[1]

[1]*Université de Rennes 1, France*

Abstract The discretization of Distance Geometry Problems (DGPs) allows to reduce their search domains to trees which are binary when all distances are exact. DGPs can be therefore seen as combinatorial optimization problems, which we solve by employing an ad-hoc Branch & Prune (BP) algorithm, that is potentially able to enumerate the entire solution set. Essential for the discretization are some assumptions to be verified by DGP instances (we say that such instances belong to the DMDGP class). When DGPs related to molecules are considered, the order given to the atoms of the molecule plays an important role, because the discretizability of the instance is strongly related to this order. In this talk, I will discuss on different approaches to this ordering problem, which becomes a fundamental pre-processing step for applying BP. The case in which all distances are exact, as well as the more realistic one in which there are imprecise distances, will be discussed in details.

DGA 2013, pp. 17 – 17.

Distance-based formulations for the position analysis of kinematic chains

Nicolas Rojas[1]

[1] *SUTD-MIT International Design Center, Singapore*

Abstract This talk addresses the problem of finding all possible assembly modes that a multi-loop linkage can adopt. This problem arises when solving, for instance, the inverse kinematics of serial robots or the forward kinematics of parallel robots. The first step to solve it consists in deriving a set of closure conditions, that is, a set of equations that are satisfied if, and only if, the linkage is correctly assembled. Most of the current techniques use as closure conditions a set of independent loop equations. The use of independent loop equations has seldom been questioned despite the resulting system of equations becomes quite involved even for simple linkages. In this talk, it will be shown how Distance Geometry is of great help to get simpler sets of closure conditions. The developed technique will be exemplified using different Baranov trusses, Assur kinematic chains, and pin-jointed Grübler kinematic chains. As by-product of this technique, an efficient procedure for tracing coupler curves of pin-jointed linkages will be also presented.

DGA 2013, pp. 19 – 19.

Topological Dimensionality Reduction

Vin de Silva[1]

[1]*Pomona College, USA*

Abstract High-dimensional data sets often carry meaningful low-dimensional structures. There are different ways of extracting such structural information. The classic (circa 2000, with some anticipation in the 1990s) strategy of nonlinear dimensionality reduction (NLDR) involves exploiting geometric structure (geodesics, local linear geometry, harmonic forms etc) to find a small set of useful real-valued coordinates. The classic (circa 2000, with some anticipation in the 1990s) strategy of persistent topology calculates robust topological invariants based on a parametrized modification of homology theory. In this talk, I will describe a marriage between these two strategies, and show how persistent cohomology can be used to find circle-valued coordinate functions. I will go on to describe some applications to dynamical systems. This is joint work with Dmitry Morozov, Primoz Skraba, and Mikael Vejdemo-Johansson.

DGA 2013, pp. 21 – 21.

Localization by Global Registration

Amit Singer[1]

[1]*Princeton University, USA*

Abstract The distance geometry problem consists of estimating the locations of points from noisy measurements of a subset of their pair-wise distances. The problem has received a great deal of attention in recent years, due to its importance in applications such as wireless sensor networks and structural biology. This talk will focus on recent divide-and-conquer approaches that solve the problem in two steps: In the first step, the points are partitioned into smaller subsets and each subset is localized separately into a local map, whereas in the second step a global map is obtained by stitching together all the local maps. Results of numerical simulations demonstrate the advantages of this approach in terms of accuracy and running time.

DGA 2013, pp. 23 – 23.

Distance Geometry Optimization and Applications

Zhijun Wu[1]

[1]*Iowa State University, USA*

Abstract A distance geometry problem is to find the coordinates for a set of points in a given metric space given the distances for the pairs of points. The distances can be dense (given for all pairs of points) or sparse (given only for a subset of all pairs of points). They can be provided with exact values or with small errors. They may also be given with a set of ranges (lower and upper bounds). In any case, the points need to be determined to satisfy all the given distance constraints. The distance geometry problem has many important applications such as protein structure determination in biology, sensor network localization in communication, and multidimensional scaling in statistical classification. The problem can be formulated as a nonlinear system of equations or a nonlinear least-squares problem, but it is computationally intractable in general. On the other hand, in practice, many problem instances have tens of thousands of points, and an efficient and optimal solution to the problem is required. In this talk, I will give a brief review on the formulation of the distance geometry problem and its solution methods. I will then present a so-called geometric buildup method and show how it can be applied to solve a distance geometry problem efficiently and deal with various types of distance data, dense or sparse, exact or inexact, effectively. I will also show how the method can be applied to a set of distance bounds and obtain an ensemble of solutions to the problem. Some computational results on protein structure determination and sensor network localization will be demonstrated.

DGA 2013, pp. 25 – 25.

Multicoloring of 3D hexagonal graphs

Janez Žerovnik[1]

[1]*Fakulteta za strojništvo Ljubljana, Slovenia*

Abstract A fundamental problem that appeared in the design of cellular networks is to assign sets of channels to transmitters in order to avoid unacceptable interferences. In the 2D case, good approximation algorithms exist that use the coordinates of the nodes that run in linear time (and even constant time in parallel mode). Some results for the 3D have been recently obtained, again the coordinates are assumed to be known. Because of the importance of this information it is interesting to ask how difficiult it is, knowing the distances to the neighbors, to find an embedding of the graph that would allow assigning at least approximate coordinates. This may provide efficient methods for assigning channels to ad-hoc sensor networks.

EXTENDED ABSTRACTS

DGA 2013, pp. 29 – 32.

Counting the number of solutions of the Discretizable Molecular Distance Geometry Problem *

Germano Abud[1,2] and Jorge Alencar[1]

[1]*Universidade Estadual de Campinas, IMECC- Unicamp, Campinas, São Paulo, Brazil.* jorge.fa.lima@gmail.com

[2]*Universidade Federal de Uberlândia, FAMAT-UFU, Uberlândia, Minas Gerais, Brazil.* germano@famat.ufu.br

Abstract The Discretizable Molecular Distance Geometry Problem (DMDGP) is a subset of the Molecular Distance Geometry Problem, where the solution space has a finite number of solutions. We propose a way to count this value, based on the symmetric properties of the DMDGP.

Keywords: Branch-and-Prune, molecular distance geometry problem, number of solutions

1. Introduction

The Molecular Distance Geometry Problem (MDGP) arises in nuclear magnetic resonance (NMR) spectroscopy analysis, which provides a set of inter-atomic distances d_{ij} for certain pairs of atoms (i, j) of a given protein [3]. The question is how to use this set of distances in order to calculate the positions $x_1, \ldots, x_n \in \mathbb{R}^3$ of the atoms forming the molecule [11].

A simple undirected graph $G = (V, E, d)$ can be associated to the problem, where V represents the set of atoms, E models the set of atom pairs for which a Euclidean distance is available, and the function $d : E \to \mathbb{R}^+$ assigns distance values to each pair in E. The MDGP can then be formally defined as the following: given a weighted simple undirected graph $G = (V, E, d)$, is there a function $x : V \to \mathbb{R}^3$ such that

$$||x_i - x_j|| = d_{ij} \quad \forall (i, j) \in E? \tag{1}$$

Many algorithms have been proposed for the solution of the MDGP, and most of them are based on a search in a continuous space [15].

Exploring some rigidity properties of the graph G, the search space can be discretized where a subset of MDGP instances is defined as the Discretizable MDGP (DMDGP) [14]. The main idea behind the discretization is that the intersection of three spheres in the three-dimensional space consists of at most two points under the hypothesis in which their centers are not aligned. The definition of an ordering on the atoms of the protein satisfying the conditions that distances to at least three immediate predecessors are known and suggests a recursive search on a binary tree containing the potential coordinates for the atoms of the molecule [5]. The binary tree of possible solutions is explored starting from its top, where the first three atoms are positioned

*Thanks to CAPES for financial support

and by placing one vertex per time. At each step, two possible positions for the current vertex v are computed, and two new branches are added to the tree. As soon as a position is found to be infeasible, the corresponding branch is pruned and the search is backtracked. This strategy defines an efficient algorithm called Branch and Prune (BP) [5].

We propose a way to count the number of solutions of the DMDGP, based on its symmetric properties established in [8].

2. The Euclidean Distance Matrix Completion Problem

Functions (or realizations) $x : V \to \mathbb{R}^3$ satisfying (1) are called valid realizations. Once a valid realization is found, distances between all pairs of vertices can be determined, which extends $d : E \to \mathbb{R}^+$ to a function $d' : V \times V \to \mathbb{R}^+$, where the values of the function d' can be arranged into a square *Euclidean distance matrix* on the set $D = \{x_v : v \in V\} \subset \mathbb{R}^3$. The pair (D, d') is known as a *distance space* [1].

In the *Euclidean Distance Matrix Completion Problem* (EDMCP) [9], the input is a partial square symmetric matrix M and the output is a pair (M', k), where M' is a symmetric completion of M and $k \in \mathbb{N}$ such that: (a) M' is a Euclidean distance matrix in $\mathbb{R}^k$ and (b) k is minimum as possible. We consider a variant of the EDMCP, called EDMCP$_k$, where $k = 3$ is actually given as part of the input and the output certificate for YES instances only consists of the completion matrix M' of the partial matrix M as a Euclidean distance matrix (M' is called a valid completion) [7].

There is a strong relationship between the MDGP and the EDMCP$_3$: each MDGP instance G can be transformed in linear time to an EDMCP$_3$ instance (and vice versa [11]) by just considering the weighted adjacency matrix of G where vertex pairs $\{u, v\} \notin E$ correspond to entries missing from the matrix related to the EDMCP$_3$ instance.

3. Counting the number of solutions of the DMDGP

As remarked in [10], the completion in $\mathbb{R}^3$ of a partial distance matrix with the structure

$$\begin{bmatrix} 0 & d_{12} & d_{13} & d_{14} & ? \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & 0 & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & 0 & d_{45} \\ ? & d_{52} & d_{53} & d_{54} & 0 \end{bmatrix}$$

can be carried out in constant time by solving a quadratic system in the unknown d_{15}, represented as a question mark in the matrix above derived from setting the Cayley-Menger determinant [1] of the related distance space to zero.

The matrix above is an EDMCP$_3$ instance related to some DMDGP instance. In fact, for any DMDGP instance, we have an EDMCP$_3$ instance given by a matrix M such that (at least) the elements (M_{ij}) satisfying $|i - j| \leq 3$ are known [14].

We need now some results related to the symmetric properties of the DMDGP [8] (for a given DMDGP instance $G = (V, E)$ with $|V| = n$, let the distances d_{ij} of the associated EDMCP$_3$ instance given according to the ordering on V that guarantees that all d_{ij} satisfying $|i - j| \leq 3$ are known and consider that x_1, x_2, x_3, x_4 are fixed):

Theorem 1. *Given an EDMCP$_3$ instance of order n, related to some DMDGP instance, the results below hold with probability 1 [8].*

1. *If the distance $d_{1,n}$ is known, there is just one solution to the given EDMCP$_3$ instance.*

2. *If all the distances $d_{i,i+4}$, $i = 1, \dots n-4$, are known, there is also just one solution to the given EDMCP$_3$ instance.*

3. *There are just 2 possible (distinct) values for the unknown distances $d_{i,i+4}$, $i = 1, \dots n-4$, related to the EDMCP$_3$ instance.*

In order to illustrate how to count the number of solutions of the DMDGP, consider the following example of the EDMCP$_3$ associated to some DMDGP instance (by the symmetry, we only consider d_{ij} such that $i \leq j$, for $i, j = 1, \cdots, n$):

$$\begin{bmatrix}
0 & d_{12} & d_{13} & d_{14} & ? & d_{16} & ? & ? & ? & ? & ? & ? \\
 & 0 & d_{23} & d_{24} & d_{25} & ? & ? & ? & ? & ? & ? & ? \\
 & & 0 & d_{34} & d_{35} & d_{36} & ? & ? & ? & ? & ? & ? \\
 & & & 0 & d_{45} & d_{46} & d_{47} & ? & ? & d_{4,10} & ? & ? \\
 & & & & 0 & d_{56} & d_{57} & d_{58} & ? & ? & ? & ? \\
 & & & & & 0 & d_{67} & d_{68} & d_{69} & ? & ? & ? \\
 & & & & & & 0 & d_{78} & d_{79} & d_{7,10} & ? & ? \\
 & & & & & & & 0 & d_{89} & d_{8,10} & d_{8,11} & ? \\
 & & & & & & & & 0 & d_{9,10} & d_{9,11} & d_{9,12} \\
 & & & & & & & & & 0 & d_{10,11} & d_{10,12} \\
 & & & & & & & & & & 0 & d_{11,12} \\
 & & & & & & & & & & & 0
\end{bmatrix}.$$

Define the k-diagonal as the subdiagonal of a simmetric matrix A of order n, whose elements (A_{ij}) satisfy $|j - i| = k$, $k = 0, \dots, n-1$.

Since the distance d_{16} is known, there is just one possible value for the distances d_{15} and d_{26} (by Result 1, considering $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$). Also, since the distance $d_{4,10}$ is known, there is just one possible value for the distances d_{48}, d_{49},d_{59},$d_{5,10}$ and $d_{6,10}$ (by Result 1, considering $V = \{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$). In order to complete the 4-diagonal, the only missing distances are d_{37}, $d_{7,11}$, and $d_{8,12}$. So, by Results 2 and 3, there are 2^3 possible solutions to this EDMCP$_3$ instance.

Based on these ideas, it is possible to define an efficient algorithm to count the number of solutions of a given EDMCP$_3$ instance related to some DMDGP instance. From the example above, we can also notice that if we know, in fact, any k-diagonal of the matrix related to the EDMCP$_3$ instance, for $k = 4, \dots, n-1$, there is also just one solution to the EDMCP$_3$ instance.

Now given a DMDGP instance, if we know the number of solutions to the related EDMCP$_3$ then we also known the number of solutions (realizations) to the DMDGP instance. In fact, each solution of the given EDMCP$_3$ is asociated to two realizations (solutions) of the related DMDGP, up to rotations and translations.

In [7], it is proposed a coordinate-free BP, called the dual BP, that takes decisions about distance values on missing edges rather than on realizations of vertices in $\mathbb{R}^3$. The original algorithm (the primal BP) decides on points $x_v \in \mathbb{R}^3$ to assign to the next vertex v, whereas the dual BP decides on distances δ to assign to the next missing distance incident to v and to a predecessor of v. In addition to the formalization of the results of this work, we are studying the possibilities to define a primal-dual BP algorithm in order to get a more efficient method to solve DMDGP instances.

Acknowledgments

The authors would like to thank the Brazilian research agency CAPES for their financial support.

References

[1] L. Blumenthal, Theory and Applications of Distance Geometry, Oxford University Press, Oxford, 1953.

[2] G. Crippen and T. Havel, Distance Geometry and Molecular Conformation, Wiley, New York, 1988.

[3] B. Donald, Algorithms in Structural Molecular Biology, MIT Press, Boston, 2011.

[4] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, The discretizable molecular distance geometry problem, Computational Optimization and Applications, 52 (2012), 115–146.

[5] C. Lavor, L. Liberti, and A. Mucherino, The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances, Journal of Global Optimization, (DOI:10.1007/s10898-011-9799-6).

[6] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, Molecular distance geometry methods: from continuous to discrete, International Transactions in Operational Research, 18 (2010), 33–51.

[7] L. Liberti and C. Lavor, On a relationship between graph realizability and distance matrix completion, in Optimization theory, decision making, and operational research applications, A. Migdalas, ed., Proceedings in Mathematics, pp. 2-9, Springer, Berlin, 2012.

[8] L. Liberti, B. Masson, J. Lee, C. Lavor, and A. Mucherino, On the number of realizations of certain Henneberg graphs arising in protein conformation, Discrete Applied Mathematics, (accepted).

[9] M. Laurent, Cuts, matrix completions and graph rigidity, Mathematical Programming, 79 (1997), pp. 255–283.

[10] J. Porta, L. Ros, and F. Thomas, Inverse kinematics by distance matrix completion, in Proceedings of the 12th International Workshop on Computational Kinematics, 2005, pp. 1–9.

[11] Q. Dong and Z. Wu, A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances, Journal of Global Optimization, 22:365?375, 2002.

DGA 2013, pp. 33 – 34.

Combinatorial generalizations of Jung's theorem*

Arseniy Akopyan

[1]*Institute for Information Transmission Problems, Russian Academy of Sciences and P. G Demidov Yaroslavl State University, Russia.* akopjan@gmail.com

Abstract We consider combinatorial generalizations of Jung's theorem on covering the set with unit diameter by a ball. We prove "fractional" and "colorful" versions of the theorem.

Keywords: Jung's theorem, Helly's theorem

The famous theorem of Jung states that any set with diameter 1 in $\mathbb{R}^d$ can be covered by a ball of radius $R_d = \sqrt{\frac{d}{2(d+1)}}$ (see [1]).

The proof of this Theorem is based on Helly's theorem:

Theorem 1 (Helly's theorem)**.** *Let $\mathscr{P}$ be a family of convex compact sets in $\mathbb{R}^d$ such that a intersection of any $d+1$ of them is not empty, than the intersection of all of the sets from $\mathscr{P}$ is not empty.*

Helly's theorem has many generalizations. M. Katchalski and A. Liu in 1979 [3] proved "fractional" version of Helly's theorem and G. Kalai in 1984 [2] gave a strongest version of it. L. Lovász in 1979 suggested a "colorful" version of Helly's theorem. We give analogues generalizations of Jung's theorem.

Theorem 2 (The fraction version of Jung's theorem)**.** *For every $d \geq 1$ and every $\alpha \in (0,1]$ there exists a $\beta = \beta(d,\alpha) > 0$ with the following property. Let $\mathcal{V}$ be a n-point set in $\mathbb{R}^d$ such that for at least αC_n^2 of pairs $\{x,y\}$ $(x,y \in \mathcal{V})$ distance between x and y less than 1. Then there exists a ball with radius R_d, which covers βn points of $\mathcal{V}$. And $\beta \to 1$ as $\alpha \to 1$.*

We will use the following definition,

Definition 3. *We call two nonempty sets $\mathcal{V}_1$ and $\mathcal{V}_2$ close, if for any points $x \in \mathcal{V}_1$ and $y \in \mathcal{V}_2$, the distance between x and y is not greater than 1.*

It is easy to see that if two close sets $\mathcal{V}_1$ and $\mathcal{V}_2$ are given, diameter of each of them is not greater than 2. Moreover, the following theorem holds.

Theorem 4. *Union of several pairwise close sets in $\mathbb{R}^d$ can be covered by a ball of radius 1.*

It is clear that the diameter of the cover ball in this theorem could not be decreased. The following two question have sense.

Suppose a family of pairwise close sets $\mathcal{V}_1$, $\mathcal{V}_2$, ..., $\mathcal{V}_n$ in $\mathbb{R}^d$ is given.

1. What is the minimal R, so that at least one of the sets $\mathcal{V}_i$ can be covered by a ball of radius R.

*This research is supported by the Dynasty Foundation, Russian Foundation for Basic Research grants 12-01-31281 and 11-01-00735, and the Russian government project 11.G34.31.0053.

2. What is the minimal D, so that at least one of the sets $\mathcal{V}_i$ has diameter no greater than D.

Theorem 5. *Let $\mathcal{V}_1$, $\mathcal{V}_2$, ..., $\mathcal{V}_n$ be pairwise close sets in $\mathbb{R}^d$. Then one of the set $\mathcal{V}_i$ can be covered by a ball with radius R.*

$$R = \tfrac{1}{\sqrt{2}} \text{ if } n \le d;$$
$$R = R_d = \sqrt{\tfrac{d}{2(d+1)}} \text{ if } n > d.$$

Through $D_d(n)$ we denote the minimal diameter of optimal spherical antipodal code of cardinality $2n$ on the unit sphere $\mathbb{S}^{d-1}$.

Theorem 6. *Let $\mathcal{V}_1$, $\mathcal{V}_2$, ..., $\mathcal{V}_n$ be pairwise close sets in $\mathbb{R}^d$. Then one of the set $\mathcal{V}_i$ has diameter not greater than*

$$D = \frac{2}{\sqrt{4 - D_d(n)^2}}.$$

References

[1] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Proc. Sympos. Pure Math., Vol. VII*, pages 101–180. Amer. Math. Soc., Providence, R. I., 1963.

[2] G. Kalai. Intersection patterns of convex sets. *Israel Journal of Mathematics*, 48(2):161–174, 1984.

[3] M. Katchalski and A. Liu. A problem of geometry in $\mathbb{R}^n$. *Proceedings of the American Mathematical Society*, 75(2):284–288, 1979.

DGA 2013, pp. 35 – 39.

Clustering of Fuzzy Data via Spectral Methods

Jorge Alencar,[1] Estevão Esmi[1] and Laécio C. Barros,[1]

[1] *University of Campinas, Campinas, Brazil,* {jorge.fa.lima,eelaureano}@gmail.com, laeciocb@ime.unicamp.br

Abstract Clustering are widely found in various applications on pattern recognition area as a tool for data analysing. The vague and uncertain nature of data from many practical problems suggests the need to develop clustering algorithms able to deal with such kind of datasets. Since the fuzzy set theory provide a mathematical basis to handle uncertain concepts and informations, we introduce a clustering method for datasets whose elements are represented by fuzzy sets. Our approach corresponds to a modified version of a clustering algorithm of the literature for partitioning of the sets of graphs that is based on spectral theory.

Keywords: Clustering, Fuzzy Sets, Spectral Methods, Graph, Distance Matrix.

1. Introduction

Clustering algorithms aim to divide the dataset in groups or clusters according to some rule, so that, in the end, elements of a same cluster are similar while elements of disjoint clusters are dissimilar in a certain sense [11]. Thus, clustering tasks depend on the choice of a certain (dis)similarity measure for evaluating the (dis)similarity between elements of the considered dataset. Clustering plays a important rule for data analysing and its application can founded in a variety of areas, such as pattern recognition, image segmentation, genetics, and etc. [3, 5].

In this work, we introduce a new clustering algorithm based on spectral theory for dealing with uncertain data represented by the class of fuzzy sets. In the following, we will recall some basic concepts of fuzzy set theory.

A fuzzy subset A of non-empty universe U is represented by a function $\varphi_A : U \longrightarrow [0,1]$, called membership function of A, where the value $\varphi_A(u)$ denotes the degree of membership of $u \in U$ in the fuzzy subset A. In particular, a classic (crisp) subset A of U is a fuzzy subset such that its membership function is its the characteristic function $\chi_A : U \longrightarrow \{0,1\}$. For all $\alpha \in (0,1]$, we define the α-cut of a fuzzy subset A of U, denoted by $[A]^\alpha$, by means of set $\{u \in U | \varphi_A(u) \geq \alpha\} \subseteq U$. By definition, we set $[A]^0$ as the closure of the set $supp(A) = \{u \in U | \varphi_A(u) > 0\}$. Every fuzzy subset A of U is uniquely identified by its family of α-cuts $(\{[A]^\alpha\}_{\alpha\in[0,1]})$ [8]. From now on, for simplicity, we assume that $U = \mathbb{R}$.

Let $\mathcal{F}(\mathbb{R})$ be the symbol that denotes the class of fuzzy set such that their α-cut are compact subset of $\mathbb{R}$ for all $\alpha \in [0,1]$. The proposed clustering algorithm aims to partition $\mathcal{F}(\mathbb{R})$ from a given finite subset of $\mathcal{F}(\mathbb{R})$. To this end, we consider as dissimilarity measure the metric $\mathcal{D}$ on $\mathcal{F}(\mathbb{R})$ defined as

$$\mathcal{D}(A,B) = \sup_{0\leq\alpha\leq 1} d_H([A]^\alpha, [B]^\alpha), \forall A, B \in \mathcal{F}(\mathbb{R}), \tag{1}$$

where d_H denotes Hausdorff's metric for compact subset of $\mathbb{R}$, i.e. for compact subsets I, J of $\mathbb{R}$ we have

$$d_H(I,J) = \max\left\{\sup_{x\in I}\left(\inf_{j\in J}|x-j|\right), \sup_{y\in J}\left(\inf_{i\in I}|y-i|\right)\right\}. \tag{2}$$

By definition, the metric $\mathcal{D}$ extends d_H, that is, if $A, B \in \mathcal{F}(\mathbb{R})$ represent compact crisp sets then $\mathcal{D}(A,B) = d_H(A,B)$. In particular, we have $\mathcal{D}(\{a\},\{b\}) = |a-b|$ if $a, b \in \mathbb{R}$.

2. Methodology

Using the metric $\mathcal{D}$ on $\mathcal{F}(\mathbb{R})$ given in Equation (1), we propose a spectral-based clustering algorithm for classes of fuzzy sets based on the algorithms named *unnormalized spectral clustering* [6], *normalized spectral clustering of Shi and Malik* [9], and *normalized spectral clustering of Ng et al. [7].* In contrast with these algorithms, our approach takes account of a different set of eigenvectors as well as includes a pre-processing in the input adjacency matrix of graph and automatically adjusts the number of clusters. Let us point out such changes into following two synthetic examples.

Let $R_1 = \{p_i\}_{i=1}^{15}$ be a subset of $\mathbb{R}$ given as

$$R_1 = \left\{\begin{array}{lll} p_1 = 8.147E-001, & p_2 = 9.058E-001, & p_3 = 1.270E-001, \\ p_4 = 9.134E-001, & p_5 = 6.324E-001, & p_6 = 5.098E+000, \\ p_7 = 5.278E+000, & p_8 = 5.547E+000, & p_9 = 5.958E+000, \\ p_{10} = 5.965E+000, & p_{11} = 1.016E+001, & p_{12} = 1.097E+001, \\ p_{13} = 1.096E+001, & p_{14} = 1.049E+001, & p_{15} = 1.080E+001 \end{array}\right\}.$$

The set R_1 comprises five elements of three disjoint clusters C_1, C_2, and C_3. More specifically, we have

$$\begin{aligned} C_1 &= \{p_1, p_2, p_3, p_4, p_5\}, \\ C_2 &= \{p_6, p_7, p_8, p_9, p_{10}\}, \\ C_3 &= \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}. \end{aligned}$$

We can interpret R_1 as a set of fuzzy sets $\{\hat{p}_i\}_{i=1}^{15}$ whose respective membership functions $\varphi_i : \mathbb{R} \longrightarrow [0,1]$ are given by

$$\varphi_i(x) = \left\{\begin{array}{ll} 1, & \text{if } x = p_i \\ 0, & \text{if } x \neq p_i \end{array}\right.$$

for $i = 1, \ldots, 15$. Using these fuzzy sets, we can yield a distance matrix $D = (d_{ij})$, where d_{ij} is the distance with respect to the metric $\mathcal{D}$ between the fuzzy sets $\hat{p}_i$ e $\hat{p}_j$ for $i, j = 1, \ldots, 15$. We associate the matrix D to a weighted complete simple graph G such that its adjacency matrix, denoted by $A(G)$, is the matrix D, i.e. $A(G) = D$. Moreover, we can produce a minimum spanning tree T from the graph G.

The number of produced clusters is an user-defined parameter of algorithms described in [6]. Our approach adjusts automatically a suitable number of cluster based on edge's weight of the tree T. Let p be the number of edges in T that their weights are greater than the sum of the mean value and standard deviation of all weight values of edges in T. For $j = 2, \ldots, p+1$, we apply the spectral algorithms in order to determine j clusters. Thus, in the end, we have p families of clusters for each spectral algorithm.

We chose the family of cluster among the p families that one with greatest value of Dunn's index [2, 10] by means of the following equation

$$\min_{1\leq i_1<i_2\leq j}\left\{\frac{d_C(i_1,i_2)}{\max_{1\leq i_3\leq j} d'(i_3)}\right\}$$

where j denotes the number of clusters, $d_C(i_1, i_2)$ denotes the distance between the clusters i_1 and i_2, and $d'(i_3)$ denotes the greatest distance among the elements of cluster i_3.

Given the maximum number of clusters, we can produce a subgraph H in G such that $H = \cup_{i=1}^{p+1} T_i$, where $T_1 = T$ and T_i, for $i = 2, \ldots, p+1$, denotes, respectively, the minimum spanning tree of $G[E(G) \setminus \cup_{j=1}^{i-1} E(T_j)]$, i.e., the minimum spanning tree of the subgraph which was obtained by cutting the edges in $\cup_{j=1}^{i-1} E(T_j)$ from the original graph G.

Let $A(H) = (a_{ij})$ be the adjacency matrix of H, we can obtain the graph H' such that the coefficients of the adjacency $A(H') = (a'_{ij})$ are given by

$$a'_{ij} = 1 - \frac{1}{2 \cdot \|A(H)\|_{\max}} a_{ij}, \quad \text{if} \quad i \sim j \quad \text{in} \quad H,$$

where $\|A(H)\|_{\max} = \max_{i,j} |a_{ij}|$. In the resulting graph H', we apply the three aforementioned spectral methods [6]. Each one uses, respectively, the eigenvectors of the following Laplacian matrices from the matrix $A' = A(H')$:

$$L = E - A' \tag{3}$$

$$L_{rw} = E^{-1} L \tag{4}$$

$$L_{sym} = E^{-\frac{1}{2}} L E^{-\frac{1}{2}} \tag{5}$$

where $E = (e_{ij})$ is a $n \times n$ diagonal matrix such that $e_{ii} = \Sigma_{j=1}^{n} a'_{ij}$ for $i = 1, \ldots, n$. Equation (3) is said to be unnormalized, while the Equations (5) and (4) are said to be normalized.

Each algorithm presented in [6] uses a subset of the normalized eigenvectors of one of above matrices which are obtained from a diagonalization method. However, the proposed method considers a subset of eigenvectors such that their magnitudes are equal to the root square of the corresponding eigenvalue. Since that the graph H is connected, the second smallest eigenvalues of three matrices above are strictly positive [6], avoiding pathologies which involve the zero vector.

Note that, for the set R_1 we have the $p = 2$. Thus, we applied each spectral clustering algorithm twice, searching from 2 to 3 clusters on R_1. In all cases, the family with 3 clusters reached the greatest Dunn's index. Moreover, as expected, the families of clusters produced by the algorithms were identical since the corresponding graph H' is very regular and at most of its vertices have approximately the same degree.

The next example illustrates a generalization of the above idea to deal with fuzzy sets. To this end, we consider the following family of fuzzy triangular numbers $R_2 = \{t_i\}_{i=1}^{15} \subset \mathcal{F}(\mathbb{R})$:

$$R_2 = \left\{ \begin{array}{lll} t_1 = (1;2;8), & t_2 = (3;9;10), & t_3 = (1;5;10), \\ t_4 = (5;9;10), & t_5 = (6;8;10), & t_6 = (51;57;58), \\ t_7 = (50;54;57), & t_8 = (54;58;59), & t_9 = (57;58;59), \\ t_{10} = (52;57;60), & t_{11} = (104;107;108), & t_{12} = (100;104;107), \\ t_{13} = (103;103;108), & t_{14} = (100;108;110), & t_{15} = (100;101;102) \end{array} \right\}$$

Recall that the membership function of a fuzzy triangular number $t = (a; b; c)$ is given by

$$\varphi_t(x) = \left\{ \begin{array}{l} 0, \quad \text{if} \quad x \leq a \\ \frac{x-a}{b-a}, \quad \text{if} \quad a < x \leq b \\ \frac{x-c}{b-c}, \quad \text{if} \quad b < x \leq c \\ 0, \quad \text{if} \quad x > c \end{array} \right. .$$

Figure 1 shows the membership functions of fuzzy sets in R_2 which are clearly separated into three clusters or groups.

Following the same approach of the last example, we obtained the same 3 clusters on R_2 for each one of three the spectral clustering under consideration. Figure 1 reveals that the algorithms identified perfectly the all three clusters as desired.

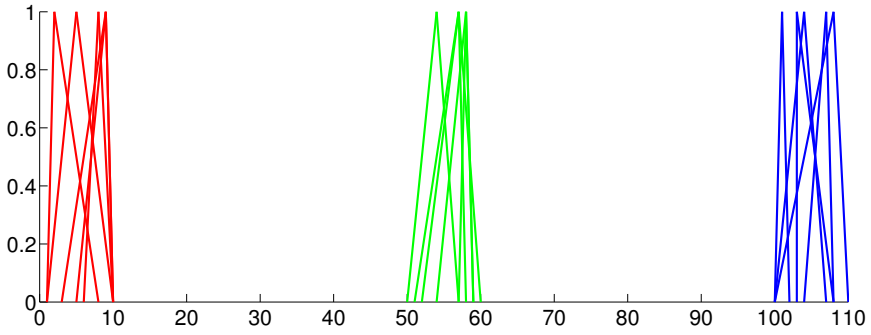


Figure 1: Fuzzy sets in R_2 for each one of three cluster.

3. Simulations on Fisher's Dataset

In this section we test the proposed clustering method in the well-known Fisher's dataset [4]. This dataset is composed of 150 samples of Iris flower that are equally divided into 3 species (setosa, virginica, and versicolor).

In order to use our method, we have to extend the metric $\mathcal{D}$ to deal with n-tuples of fuzzy sets. Let $F_1, F_2 \in \mathcal{F}(\mathbb{R})^n$, where the symbol $\mathcal{F}(\mathbb{R})^n$ denotes the set of n-tuples of fuzzy sets in $\mathcal{F}(\mathbb{R})$, and let $\mathcal{D}$ be the metric given in Equation (1). We define the metric $\mathcal{D}_n : \mathcal{F}(\mathbb{R})^n \times \mathcal{F}(\mathbb{R})^n \to \mathbb{R}$ as

$$\mathcal{D}_n(F_1, F_2) = \sqrt{\sum_{i=1}^{n} \mathcal{D}(F_1(i), F_2(i))^2}.$$

Note that, if $F_1, F_2 \in \mathbb{R}^n$, i.e, if F_1 and F_2 are n-tuples of real numbers, then we have that $\mathcal{D}_n(F_1, F_2)$ coincides to the usual Euclidean distance between the n-tuples F_1 and F_2.

Let $\mathbb{F} = \{F_i : i = 1, \ldots, 150\}$, where $F_i \in \mathbb{R}^n$ coresponds to the ith sample of Fisher's dataset. We obtain 3 clusters by applying the proposed clustering method to the distance matrix $D = (d_{ij})$, where $d_{ij} = \mathcal{D}_{150}(F_i, F_j)$ for $i, j = 1 \ldots, 150$.

In general, clustering algorithms yield an indexes vector $\mathbf{p}$ such that the ith element of dataset is associated to the cluster $p_i \in \mathbb{N}_k = \{1, \ldots, k\}$, where k denotes the number of resulting clusters. Thus, in order to compare different clustering approaches, we use the *cluster misclassification function* $d_M : \mathbb{N}_k^n \times \mathbb{N}_k^n \to \mathbb{Z}_+$ defined in [1].

We compare our methodology to the well-known k-means algorithm in the Fisher's dataset, with $k = 3$, by means of the metric d_M. Let $\mathbf{v}$ be the desired indexes vector (which corresponds to the three groups, i.e. clusters, of species of Iris flowers) and let $\mathbf{p}, \mathbf{q}$ be the resulting indexes vectors from the proposed algorithm and k-means algorithm, respectively.

In conclusion, the values $d_M(\mathbf{p}, \mathbf{v}) = 10$ and $d_M(\mathbf{v}, \mathbf{q}) = 16$ indicate that our method produced clusters that are more similar to the original groups than the ones produced using the k-means algorithm.

4. Conclusion and Future Works

On the one hand, the results obtained on two examples above indicate that the proposed approach works well on data with high value of Dunn's indices. In order to verify this hypothesis, we intend to apply our method in other sets of general fuzzy sets, not only those with triangular

membership function, with more number of clusters. On the other hand, the preliminary result on the Fisher's dataset suggests the potential of our method in real clustering tasks.

Since our method yields group of fuzzy sets, it can apply for analysing and reduction of fuzzy rule-based systems. The idea is to find conflicting or redundant rules by means of clustering of both antecedents and consequences fuzzy sets. We will compare the performances of fuzzy rule-based systems obtained before and after applying of reduction via our method.

Acknowledgments

This work was partially support by CAPES, FAPESP under grant no. 2009/16284-2, and CNPq under grant no. 306872/2009-9.

References

[1] J. Alencar, C. Lavor, T. Bonates, G. Liberali, and D. Aloise. Multidimensional scaling of clustered data. In *Proceedings of the Workshop on Distance Geometry and Applications*, 2013.

[2] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[3] Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Wiley, 4th edition, January 2009.

[4] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[5] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[6] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[7] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[8] C. V. Nogoita and D. A. Ralescu. *Applications of fuzzy sets to systems analysis*. John Wiley & Sons, Inc., New York, NY, USA, 1975.

[9] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[10] Rui Xu and Don Wunsch. *Clustering (IEEE Press Series on Computational Intelligence)*. Wiley-IEEE Press, October 2008.

[11] Rui Xu and II Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645 –678, may 2005.

DGA 2013, pp. 41 – 45.

Branch-and-prune algorithm for multidimensional scaling preserving cluster partition *

Jorge Alencar[1], Tibérius Bonates[2], Guilherme Liberali[3] and Daniel Aloise[4]

[1]*Universidade Estadual de Campinas, IMECC-Unicamp, Campinas, São Paulo, Brazil.* jorge.fa.lima@gmail.com

[2]*Universidade Federal do Semiárido, UFERSA, Mossoró, Rio Grande do Norte, Brazil.* tbonates@ufersa.edu.br

[3]*Erasmus University Rotterdam, EUR, Rotterdam, Netherlands.* liberali@ese.eur.nl

[4]*Universidade Federal do Rio Grande do Norte, UFRN, Natal, Rio Grande do Norte, Brazil.* aloise@dca.ufrn.br

Abstract In standard Multidimensional Scaling (MDS) one is concerned with finding a low-dimensional representation of a set of n objects, so that pairwise dissimilarities among the original objects are represented as distances in the embedded space with minimum error. We propose an MDS algorithm that simultaneously optimizes the distance error and the cluster membership discrepancy between a given cluster structure in the original data and the resulting cluster structure in the low-dimensional representation. We report on preliminary computational experience, which shows that the algorithm is able to find MDS representations that preserve the original cluster structure while incurring a relatively small increase in the distance error, as compared to standard MDS.

Keywords: Branch-and-Prune, Distance Geometry, Multidimensional Scaling

1. Introduction

Multidimensional scaling (MDS) is a set of techniques concerned with variants of the following problem: given the information on pairwise dissimilarities between elements of a set of n objects, find a low-dimensional representation of the given objects, while minimizing a loss function that measures the error between the original dissimilarities and the distances resulting from the low-dimensional embedding [3]. This low-dimensional embedding of the given objects is usually referred to as an *MDS representation*.

Let us consider a set P of points in $\mathbb{R}^N$ to which a clustering procedure (*e.g.*, k-means) has been applied. The application of a standard MDS procedure to P provides no guarantee that, if the clustering procedure were also applied to the MDS representation, a cluster structure similar to the one obtained for the original data would result.

Despite this fact, attempts at integrating MDS and clustering into a single technique are not entirely absent from the MDS literature. *Cluster Differences Scaling* (CDS) is one such technique [5]. Given pairwise distances between a set of objects, CDS assigns objects to clusters and creates a low-dimensional representation for each cluster. Therefore, the resulting

*Thanks to CAPES and CNPq for financial support

representation includes as many points as the number of clusters. The distance error is measured over the cluster representations for pairs of points that are assigned to different clusters. Another line of work relating clustering and MDS is the one described in [7]. There, an MDS representation is determined with the property that a k-means partition of the embedded data is identical to the optimal partition in the original space given by a so-called pairwise clustering cost function. One of the advantages of such an approach is that, instead of carrying out an expensive pairwise clustering cost procedure on the original data, one can apply a standard k-means algorithm to the embedded data and recover precisely the same information.

Unlike these approaches, in which clusters are determined as part of the process, our approach requires a cluster partition obtained *a priori*. More specifically, we assume that, in addition to the pairwise dissimilarity information, cluster membership data is given as part of the input, specifying to which cluster each point is assigned. The current availability of highly specialized optimization algorithms for clustering (see, *e.g.*, [2]) allows for instances to be solved with good accuracy, even when the data involves a large number of entities and/or complex data types. Thus, it is justified to argue for an MDS algorithm that preserves cluster partition but does not enforce the use of a specific clustering method, unlike [5, 7]. By considering the cluster partition structure as part of the input, the approach pursued in this paper can be applied in conjunction with virtually any clustering algorithm, including ones that are not based exclusively on dissimilarities. Given an appropriate cluster partition for the original data, the question is whether or not there is a low-dimensional representation of the data, which preserves the dissimilarities to an extent that makes it still possible to recover the original cluster partition structure.

This presentation is organized as follows. In Section 2 we describe an existing combinatorial algorithm for MDS and how it can be modified in order to take into account the preservation of cluster membership in the resulting MDS representation. In Section 3 we discuss the results of computational experiments carried out on a classic clustering dataset.

2. A Cluster-Partition Preserving MDS Algorithm

Let us consider a set $V \subset \mathbb{R}^N$ of n points, for which pairwise Euclidean distances (to which we shall refer as *dissimilarities*) δ_{ij} are known. In [1] a Branch-and-Prune (BP) algorithm was proposed for finding an MDS representation in $\mathbb{R}^3$ while minimizing a Stress function given by

$$S(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(d(x_i, x_j) - \delta_{ij}\right)^2, \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the resulting MDS representation and $d(x_i, x_j)$ stands for the Euclidean distance between points x_i and x_j.

Given a total order on the original points, the BP assigns standard positions for the first 3 points in such a way as to exactly match the dissimilarities among them. From the 4-th point and on, the algorithm determines the possible coordinates of each point x_i by exactly matching distances and dissimilarities of x_i with respect to the previous 3 points in the order. It is possible to show that, with probability 1, there are two possible positions for each such point [6].

This fact naturally leads to a combinatorial procedure, which is the basis of the tree-search BP algorithm. Since the algorithm determines the placement of points in a sequential manner, we shall say that a point has been *mapped* if its coordinates have already been determined. Thus, MDS representations are available at the $(n-2)$-th level of the search tree, once all points have been mapped. Moreover, since the algorithm does not enforce that *all* distances match the corresponding dissimilarities, different MDS representations might have different values of

the Stress function. An implicit enumeration scheme can then be applied based on the value of the Stress function, with tree nodes that correspond to Stress values higher than that of the best known MDS representation being removed from further investigation.

We next show how to extend this algorithm to incorporate cluster membership information, assumming that a clustering procedure was applied to the original data and that such information is available. First, we include among the input points a *reference point* for each cluster. This reference point can be, for instance, a cluster centroid, or simply an original point belonging to the cluster and preferably occupying a somewhat "central" position with respect to other points in the cluster. The only requirement on the choice of a reference point y is that the dissimilarities between y and all other points (including other reference ones) are known.

Thus, based on a total order on this augmented set of input points, we can apply the BP algorithm with the caveat that nodes corresponding to MDS representations having a high number of *cluster-partition discrepancies* (with respect to the original partition) are pruned. A cluster-partition discrepancy can be detected in a node of the search tree whenever a point that has already been mapped is closer (in the embedded space) to the mapped reference point of a different cluster than to the mapped reference point of its own cluster. Note that, for this kind of pruning to take place, it is necessary to have some reference points already mapped. We propose to order the input points in such a way that points belonging to the same cluster are grouped together, with the reference point of each cluster preceding the remaining points of its cluster.

Algorithm 1 summarizes the procedure. In line 10 of Alg. 1, we refer to the property of a node being *prunable*. A node s is said to be prunable if it has a larger Stress value (or cluster-partition discrepancy) than that of the best known MDS representation.

Algorithm 1 Pseudocode of cluster-partition preserving BP algorithm.

Require: Pairwise dissimilarities δ_{ij} between n points $(i, j = 1, \ldots, n)$.
Ensure: An MDS representation.

```
1:  Establish total order on points, reference ones included;
2:  T ← {r}, where r is the initial node, with positions for the first 3 points;
3:  while (T ≠ ∅) do
4:      Select a node t ∈ T, T ← T \ {t};
5:      for each (possible position of the first not yet mapped point in t) do
6:          Create new node s, updated with newly placed point;
7:          if (s is an MDS representation) then
8:              Consider updating best known MDS representation;
9:          else
10:             if (s is not prunable) then
11:                 T ← T ∪ {s};
12:             end if
13:         end if
14:     end for
15: end while
```

Among all solutions produced during the search, the algorithm will report, as the best solution found, one with the smallest value of *cluster misclassification*, a concept that we introduce in what follows. Let $p, q \in \mathbb{N}_k^n$, with $\mathbb{N}_k = \{1, \ldots, k\}$, be cluster index vectors, each of which assigns a cluster index i $(1 \leq i \leq k)$ to each point in V. In order to compare two such point-cluster assignments, we must account for a possible permutation of cluster labels. Thus, we define *cluster misclassification* as the function $d_M : \mathbb{N}_k^n \times \mathbb{N}_k^n \to \mathbb{Z}_+$, such that $d_M(p, q) = \min_{\sigma \in P_k} d_H(\sigma(p), q)$, where P_k is the set of permutations of $\mathbb{N}_k$, d_H is the Hamming

	Standard BP [1]			Partition-Preserving BP		
k	Stress	Misclass.	Discr.	Stress	Misclass.	Discr.
3	9.8625e+002	2	2	1.9366e+003	0	0
5	8.3719e+002	2	8	6.9674e+003	0	0
8	1.0173e+003	12	5	3.0981e+004	0	2

Table 1: Comparison between the standard BP algorithm and the proposed cluster-partition preserving BP algorithm.

distance, and $\sigma(p)$ is an index vector obtained from p via the application of $\sigma \in P_k$ (with $\sigma(p)_i = \sigma(p_i)$, for $i = 1, \ldots, n$). Function d_M is a metric that allows us to assess how dissimilar the index vector p produced by a clustering procedure applied to the embedded data is with respect to the original index vector q, obtained by clustering the original data.

3. Computational Experiments

In order to validate the proposed MDS algorithm we conducted a series of computational experiments using the classical Fisher data set [4]. Prior to the application of the MDS algorithm, duplicate points were removed and the data was clustered using a standard k-means procedure, with the number k of clusters equal to 3, 5 and 8. We used as the reference point of each cluster its centroid, defined as the average of the points belonging to the cluster.

To allow for pruning to take place since early levels of the search tree – and still focus on producing MDS representations with small deviations from the given dissimilarities δ_{ij} – we attempted to minimize the Stress function given by (1), while using the following function as pruning criterion:

$$\sigma(\mathbf{x}) = \max_{i,j=1\ldots,n} |d(x_i, x_j) - \delta_{ij}| \,. \tag{2}$$

The first column of Table 1 displays the number of clusters used for clustering the original data. The next three columns refer to: (i) the value of the Stress function corresponding to the best MDS representation found by applying the original BP algorithm of [1], (ii) the value of the cluster misclassification metric and (iii) the corresponding number of cluster-partition discrepancies. The following three columns provide similar information concerning our cluster-partition preserving BP algorithm. In both cases, the BP search was limited to a maximum of $5 \cdot 10^6$ nodes.

The results show that our algorithm was able to construct MDS representations with low (in fact, zero) misclassification counts and low cluster-partition discrepancies, while incurring a relatively small increase in the value of the Stress function.

It is important to remark that Table 1 shows a simultaneous decrease in misclassification and discrepancy for the Partition-Preserving BP, for all values of k. On the other hand, the Stress value for the Partition-Preserving BP is greater than that for the standard BP, for all values of k. Since the search tree is pruned with respect to discrepancy, this scenario is to be expected: discarding certain solutions that were taken into consideration by the Standard BP search might lead to an increase in Stress. However, since both BP searches were limited to exploring 5 million nodes, it is conceivable that the search carried out by the Partition-Preserving BP could lead to a solution with better Stress value than that of the best solution found by the Standard BP search.

As far as running time is concerned, the Partition-Preserving BP search has practically the same performance as that of the Standard BP search, since we introduce a negligible amount of extra computation in each node of the tree due to the discrepancy calculation. The computation

of the cluster misclassification metric is currently carried out as a post-processing phase, applied only to a set of elite solutions generated during the search.

While different orders of the points – as well as different reference points – may be used, our preliminary experiments showed that the order suggested here provides a good compromise between quality of the MDS representation and running time.

References

[1] Alonso, A., Carvalho, S., Lavor, C., Oliveira, A. (2012). "Escalonamento Multidimensional: uma Abordagem Discreta", Proceedings of the Congreso Latino-Iberoamericano de Investigación Operativa, Rio de Janeiro, Brazil.

[2] Aloise, D., Hansen, P., Liberti, L. (2012). "An improved column generation algorithm for minimum sum-of-squares clustering", Mathematical Programming, v. 131, p. 195-220.

[3] Borg, I., Groenen, P. (2005). Modern Multidimensional Scaling: Theory and Applications, Springer.

[4] Fisher, R. (1936). "The use of multiple measurements in taxonomic problems", Annals of Eugenics, v. 7, p. 179-188.

[5] Heiser, W., Groenen, P. (1997). "Cluster differences scaling with a within-clusters loss component and a fuzzy successive approximation strategy to avoid local minima", Psychometrika, v. 62, p. 63-83.

[6] Lavor, C., Liberti, L., Maculan, N., Mucherino, A. (2012). "The discretizable molecular distance geometry problem", Computational Optimization and Applications, v. 52, p. 115-146.

[7] Roth, V., Laub, J., Kawanabe, M., Buhmann, J. (2003). "Optimal cluster preserving embedding of nonmetric proximity data". IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 25, p. 1540-1551.

DGA 2013, pp. 47 – 50.

The Kissing Number Problem from a Distance Geometry Viewpoint *

Jorge Alencar[1], Cristiano Torezzan[1], Sueli I. R. Costa[1], Alessandro Andrioni[1]

[1] *University of Campinas, SP, Brazil.*
jorge.fa.lima@gmail.com, cristiano.torezzan@fca.unicamp.br, sueli@ime.unicamp.br, andrioni@member.ams.org

Abstract In this paper we present a formulation for the generalized kissing number problem from a distance geometry point of view. The formulation allows for to construct lower bounds for the maximal number of non overlapping spheres of radius r that can touch a unit sphere the 3 dimensional space. The solution is obtained by finding iteratively the intersection between 3 spheres and then searching for the clique number of an attached representation graph. Besides the main idea, a pseudo-code algorithm is included and charts of an example are presented. An extension of what is presented here might be extended to approach the problem in higher dimensions.

Keywords: Kissing Number, Discretizable Distance Geometry, Graphs, Spherical Codes

1. Introduction

The *kissing number problem* is a classical geometric problem in which the goal is to find the largest number $\mathrm{KN}(n)$ of equal nonoverlapping spheres in $\mathbb{R}^n$ that can touch another sphere of the same radius. If we arrange coins in a table, it easy to see (and also to prove) that the answer in $\mathbb{R}^2$ is exactly six, i.e. $\mathrm{KN}(2) = 6$.
In three dimensions the kissing number problem is also called the *thirteen spheres problem* due to a famous discussion between Isaac Newton and David Gregory in 1694. Newton believed that $\mathrm{KN}(3) = 12$ while Gregory thought that 13 might be possible

The most symmetrical configuration of 12 balls around another is achieved when the balls are placed at positions corresponding to the vertices of a regular icosahedron concentric with the central ball. However, these 12 outer balls do not kiss each other and may all be moved freely. So perhaps a 13th ball would possibly fit in. If we look at the correspond packing of non overlapping caps on the surface of the central sphere and divide the area of the central sphere by the area of one spherical cap of angular radius $\alpha = \frac{\pi}{6}$, we may get an upper bound for the kissing number in $\mathbb{R}^3$. In this case, $\mathrm{KN}(3) \leqslant 14.99282$ which somehow argue in favour of Gregory. After some preliminary works (see [7] and references therein), the problem was formally solved only in 1953 by Shütte van der Waerden [2] in behalf of Newton, $\mathrm{KN}(3) = 12$.

For dimensions greater than 3 optimal solutions are known only in three cases: $\mathrm{KN}(4) = 24$ [4], $\mathrm{KN}(8) = 240$ [5], $\mathrm{KN}(24) = 196.560$ [6]. In each of them the center of the spheres coincide with the shortest vectors of high symmetry lattices, namely, D_4, E_8 and Leech lattices, respectively. For all other dimensions there are only upper and lower bounds on $\mathrm{KN}(n)$. Good references on this subject can be found in [7].

*The authors would like to thank the Brazilian research agencies FAPESP, CAPES and CNPq for their financial support.

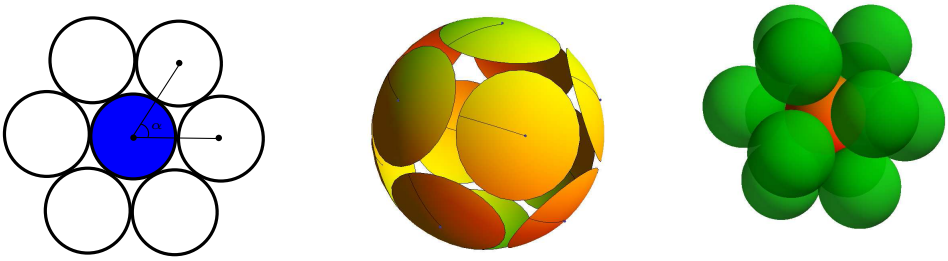


Figure 1: On the left, the perfect kissing number arrangements for $n = 2$. At the center, 12 spherical caps of angular radius $\frac{\pi}{6}$. On the right, 12 equal balls placed on the vertices of a icosahedron concentric with the central ball

Besides the geometric aspects, the battle for new records of KN(n) is also an interesting problem in mathematical programming and several formulations have been proposed (see, for instance [8]).

1.1 The Generalized Kissing Number Problem - GKNP

We can generalize the KNP by considering a different radius for the surrounding spheres. In this case we are interested in the largest number GKN(n, r) of $n-$dimensional spheres of radius r that can be placed around a central unit sphere in $\mathbb{R}^n$, so that each of the surrounding spheres touches the central one without overlapping.

This problem is equivalent to the problem of maximize the number of spherical caps packed on the surface of a unit sphere, which is related to the design of spherical codes for signal transmissions over a Gaussian Channel [9, 10].

In this paper we look to this problem from a discrete distance geometry point of view and present a constructive method to obtain lower bounds on GKN$(3, r)$ by finding the clique number[1] of an attached representation graph. The ideas introduced here can be directly applied to the GKNP(2,r), as a particular case, and might be extended to approach the GKNP in $\mathbb{R}^n$. In the next section we summarize the Discretizable Distance Geometry Problem and in Section 3 we present our approach. The ideas introduced here might be extended to approach the GKNP in $\mathbb{R}^n$.

2. The Discretizable Distance Geometry Problem

The Discretizable Distance Geometry Problem (DDGP) is a subclass of the Distance Geometry Problem (DGP), where the solution space can be discretized [14]. The interest of the DGP resides in its possible applications (molecular conformation, wireless sensor networks, statics, data visualization and robotics among others), as well as in the mathematical theory behind the results [14].

The DGP can then be formally defined as the following question: given a weighted simple undirected graph $G = (V, E, d)$, is there a function $x : V \to \mathbb{R}^K$ such that $||x_i - x_j|| = d_{ij} \quad \forall (i, j) \in E$?

When G is a complete graph (all the distances are given), a unique three-dimensional structure can be determined by a linear time algorithm [12]. Otherwise, DGP is strongly **NP**-complete when $K = 1$ and strongly **NP**-hard for general $K > 1$ [16].

[1] We remark that the clique number of a graph is the number of vertices of a maximal clique with largest number of vertices

The DGP can be naturally formulated as a nonlinear global minimization problem, where the objective function can be written as $f(x_1, \ldots, x_n) = \sum_{(i,j)\in E}(||x_i - x_j||^2 - d_{ij}^2)^2$. Assuming that all the distances are correctly given, a set $\{x_1, \ldots, x_n\} \subset \mathbb{R}^K$ is a solution if and only if $f(x_1, \ldots, x_n) = 0$. Many algorithms have been proposed for the solution of the DGP, and most of them are based on a search in a continuous space [15].

By exploring some rigidity properties of the graph G, the search space can be discretized and a the DDGP problem come into the place. For this case and when the given distances are precise, the algorithm Branch-and-Prune (BP) can be used to solve the DDGP [14].

The main idea behind of the discretization, and behind of the algorithm BP, is that the intersection among K spheres in $\mathbb{R}^K$ can produce at most two points under the hypothesis of their centers are in a hyperplane but not in a $(K-2)$-dimensional affine subspace. Consider $(K+1)$ points $\{u_i\}_{i=1}^K$ and v. If the coordinates for $\{u_i\}_{i=1}^K$ are known, as well as the distances $\{d(u_i, v)\}_{i=1}^K$ then K spheres can be defined and their intersection provides the two possible positions for the point v.

The definition of an ordering on a set of vertices satisfying such conditions suggests a recursive search on a binary tree containing the potential coordinates for the vertices [14]. The binary tree of possible solutions is explored starting from its top, where the first K points are positioned, and by placing one vertex per time. At each step, two possible positions for the current vertex v are computed, and two new branches are added to the tree. As a consequence, the size of the binary tree can increase quite quickly, but the presence of additional distances (not employed in the construction of the tree) can help in verifying the feasibility of the computed positions. As soon as a position is found to be infeasible, the corresponding branch can be pruned and the search can be backtracked.

3. The Kissing Number as a Distance Geometry Problem

Let $c_0 = (0, 0, 0)$ be the center of the central unit sphere s_0. We wish to place a collection $\mathcal{S}$ of 3-dimensional spheres $\mathcal{S} = \{s_1, s_2, \cdots, s_M\}$ of radius r, centered at the points $(c_1, c_2, \cdots, c_M)$ respectively, such that $||c_i|| = (1 + r)$ and $||c_i - c_j|| \geq 2r$ for all $i \neq j$, $i, j = 1, 2, \cdots, M$. The goal in GKNP is to increase M is as much as possible.

Our approach starts by setting $c_1 = (0, 0, 1 + r)$ and then adding the spheres $\{s_2, \cdots, s_6\}$ tangent to s_0 and s_1 (Figure 2). Then, new spheres will be included by solving the problem of intersection among 2 existent spheres and s_0. In each step, the method will design a kind of "belt" around s_0, going from up to down, as illustrated in Figure 2, where s_0 is represented in orange.

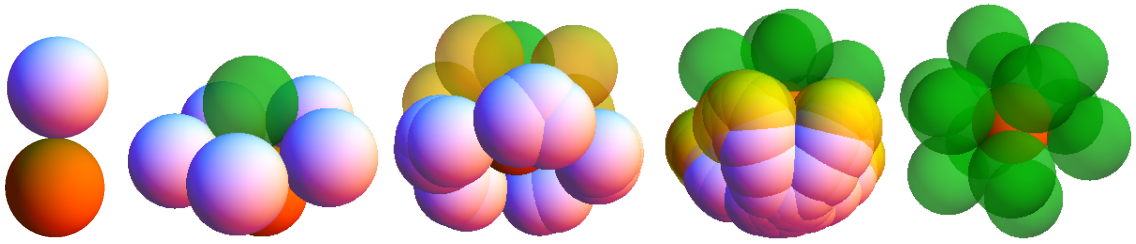


Figure 2: Belts designed using Algorithm 1 for the GKN$(3, 1)$.

After designing all possible "belts" there will be many overlapping spheres which must be eliminated in order to get the final solution. This elimination process will be done by searching

for maximal cliques of a representation graph $\mathbb{G}_M$ associated to the matrix M where:

$$m_{ij} = \begin{cases} 0 \text{ if } (i = j \text{ or } ||c_i - c_j|| < 2r) \\ 1 \text{ if } ||c_i - c_j|| \geq 2r \end{cases}$$

A lower bound for $\mathrm{GKN}(3, r)$ will be the clique number $\omega(\mathbb{G})$ of $\mathbb{G}_M$. In Algorithm 1 we present a pseudo code for the algorithm which places the spheres around s_0 and, in Figure 2, we show the steps for a lower bound of the classical $\mathrm{GKN}(3, 1) = \mathrm{KN}(3) = 12$, which is, in this case, equals to the exact solution.

References

[1] G.G. Szpiro, Newton and the kissing problem, http://plus.maths.org/issue23/features/kissing/.

[2] Schutte, K. and van der Waerden, B. Das problem der drizehn kugeln. Math. Ann. 125 (1953), 325-334.

[3] J. Leech, The problem of the thirteen spheres, Math. Gazette 41 (1956), 22-23.

[4] O. R. Musin, The kissing number in four dimensions, Ann. of Math., 168 (2008), 1-32.

[5] V.qual I. Levenshtein, On bounds for packing in n-dimensional Euclidean space. Sov. Math. Dokl. 20(2), 1979, 417-421.

[6] A.M. Odlyzko and N.J.A. Sloane, New bounds on the number of unit spheres that can touch a unit sphere in n dimensions, J. of Combinatorial Theory A, 26 (1979), 210-214.

[7] Pfender, F., Ziegler, G. M., Unter, G., and Ziegler, M. Kissing numbers, sphere packings, and some unexpected proofs. Notices Amer. Math. Soc 51 (2004), 873-883.

[8] Sergei Kucherenko and Pietro Belotti and Leo Liberti and Nelson Maculan. New formulations for the kissing number problem. Discrete Applied Mathematics. Vol. 155, 2007.

[9] T Ericson and V Zinoviev. Codes on Euclidean Spheres. North-Holland Mathematical Library, 2001.

[10] Torezzan, C.; Costa, S.I.R.; Vaishampayan, V.A.; , "Spherical codes on torus layers," Information Theory, 2009. ISIT 2009. IEEE International Symposium on , vol., no., pp.2033-2037, June 28 2009-July 3 2009.

[11] G. Crippen and T. Havel, Distance Geometry and Molecular Conformation, Wiley, New York, 1988.

[12] Q. Dong and Z. Wu, A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances, Journal of Global Optimization 22 (2002), 365-375.

[13] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, Euclidean Distance Geometry and Applications, Tech. Rep. 1205.0349v1 [q-bio.QM], arXiv, 2012.

[14] A. Mucherino, C. Lavor, L. Liberti, The Discretizable Distance Geometry Problem, Optimization Letters 6(8), 1671-1686, 2012.

[15] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, Molecular distance geometry methods:from continuous to discrete, International Transactions in Operational Research, 18 (2010), 33–51.

[16] J. Saxe, Embeddability of weighted graphs in k-space is strongly NP-hard, in Proc. of 17th Allerton Conference in Communications, Control, and Computing (1979), 480-489.

Algorithm 1 Algorithm to place spheres around c_0. In this code $f(x, y, z)$ represents a procedure to find the two solutions for the problem of intersection 3 tangent spheres in $\mathbb{R}^{\not\vdash}$.

Require: $Old = \{c_1\}$ and $Rec = \{c_2, c_3, c_4, c_5, c_6\}$.
Ensure: A set of centers in belts around central sphere.

$r^* \longleftarrow \frac{4r}{1+r}\sqrt{1+2r}$;
$test \longleftarrow 0$;
while $(test = 0)$ **do**
  $New \longleftarrow \emptyset$;
  for all $(c_i, c_j \in Rec$ such that $2r \leq \|c_i - c_j\| \leq r^*)$ **do**
    Calculate $S = \{n_1, n_2\} \longleftarrow f(c, c_i, c_j)$;
    for $(k = 1, 2)$ **do**
      if $(\exists s \in Old$ such that $\|s - n_k\| < 2r)$ OR $(\exists s \in Rec \cup New$ such that $\|s - n_k\| = 0)$ **then**
        $S \longleftarrow S \setminus \{n_k\}$;
      end if
    end for
    $New \longleftarrow New \cup S$;
  end for
  $Aux \longleftarrow \emptyset$;
  while $(New \neq \emptyset)$ **do**
    $J \longleftarrow \emptyset$;
    for all $(c_i \in Rec \wedge c_j \in New$ such that $\|c_i - c_j\| = 2r)$ **do**
      Calculate $S = \{n_1, n_2\} \longleftarrow f(c, c_i, c_j)$;
      for $(k = 1, 2)$ **do**
        if $(\exists s \in Old$ such that $\|s - n_k\| < 2r)$ OR $(\exists s \in Rec \cup New \cup Aux$ such that $\|s - n_k\| = 0)$ **then**
          $S \longleftarrow S \setminus \{n_k\}$;
        end if
      end for
      $J \longleftarrow J \cup S$;
    end for
    $Aux \longleftarrow Aux \cup New$;
    $New \longleftarrow J$;
  end while
  $Old \longleftarrow Old \cup Rec$;
  if $(Aux = \emptyset)$ **then**
    $test \longleftarrow 1$;
  else
    $Rec \longleftarrow Aux$;
  end if
end while

DGA 2013, pp. 53 – 57.

Comparison of branch-and-prune algorithm for metric multidimensional scaling with principal coordinates analysis

Ana Camila Rodrigues Alonso *,[1] Aurelio R. L. Oliveira †[2]

[1]*Departamento de Matemática Aplicada - IMECC-UNICAMP* acamila@ime.unicamp.br

[2]*Departamento de Matemática Aplicada - IMECC-UNICAMP,* aurelio@ime.unicamp.br

Abstract The metric multidimensional scaling (MDS) originates from a set of techniques for analyzing proximity of data, which is obtained through the judgment of participants who concomitantly compare several stimuli in various dimensions. In this work, we propose an approach to the problem of multidimensional scaling using a Branch-and-Prune algorithm. Moreover, we will compare it with the Principal Coordinates Analysis technique which is a classical approach for data compression (or dimensionality reduction).

Keywords: Principal Coordinates Analysis, Branch-and-Prune Algorithm, Metric Multidimensional Scaling.

1. Introduction

Multidimensional scaling (MDS) is a method that represents measurements of similarity (or dissimilarity) among pairs of objects through distances between points of a low-dimensional multidimensional space[2]. Multidimensional scaling is most often used to visualize data when only their distances or dissimilarities are available. However, when the original data are available, multidimensional scaling can also be used as a dimension reduction method, by reduction the data to a distance matrix, creating a new configuration of points [3].

The graphical display of the correlations provided by MDS enables the data analyst to literally look at the data and visually exploit their structure. This often shows regularities that remain hidden when studying arrays of numbers [2].

Pairwise Euclidean distances among n objects are given by the matrix (δ_{ij}), $i, j = 1, \ldots, n$. A set of points in an embedding metric space is considered as an image of the objects set. Usually, an m-dimensional vector space is used, and $x_i \in \mathbb{R}^m$, $i = 1, ..., n$, should be found whose inter-point distances fit the given Euclidean distances. Images of the considered objects can be found minimizing a fit criterion, e.g. the most frequently used least squares stress function [6]:

$$S(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{ij} (d(x_i, x_j) - \delta_{ij})^2, \tag{1}$$

*bolsista de Doutorado CNPq - Processo 140239/2009-0

†Bolsista de Produtividade CNPq - Processo 309561/2009-4, Projeto Fapesp: 2010/06822-4

$$d(x_i, x_j) = \left(\sum_{k=1}^{3} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}, \tag{2}$$

where $x = (x_1, \ldots, x_n), x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$; $d(x_i, x_j)$ denotes the distance between the points x_i and x_j; it is assumed that the weights are positive: $w_{ij} > 0, i, j = 1, \ldots, n$.

One way of obtaining a representation in $\mathbb{R}^3$ for this data is to determine $x_i \in \mathbb{R}^3$, $i = 1, \ldots, n$, using the classical multidimensional scaling. Classical multidimensional scaling, also known as principal coordinates analysis (PCoA), takes a matrix of interpoint distances, and creates a configuration of points. Ideally, those points can be constructed in two or three dimensions, and the Euclidean distances between them approximately reproduce the original distance matrix. Thus, scatter plot of the those points provides a visual representation of the original distances [3].

In this study, we investigate an alternative approach for obtaining the points $x_i \in \mathbb{R}^3$. This approach consists of an *Branch-and-Prune* type algorithm [4], allowing greater accuracy in comparison with the technique of principal coordinates analysis.

2. Mathematical Formulation

Consider a sequence of n points with Cartesian coordinates given by $x_1, \ldots, x_n \in \mathbb{R}^3$. The Euclidean distance between points $i-1$ and i is denoted by r_i for all $i = 2, \ldots, n$, the angle $\theta_i \in [0, \pi]$ is formed by the segments joining points $i-2$, $i-1$ and i, for all $i = 3, \ldots, n$, and the torsion angle $\omega_i \in [0, 2\pi]$ is formed by the normals through the planes defined by the points $i-3, i-2, i-1$ and $i-2, i-1, i$, for all $i = 4, \ldots, n$.

Once r_i, θ_i and ω_i are known, it is possible to fix the first three points according to determined sequence. The fourth point is determined by the torsion angle ω_4, r_2, r_3 and θ_3, the fifth point, in turn, is determined by torsion angles ω_4 and ω_5, and so on. The Cartesian coordinates $x_i = (x_{i1}, x_{i2}, x_{i3})$, for each point i, can be obtained using the following relations [5]:

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix} = B_1 B_2 \ldots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \forall i = 1, \ldots, n, \tag{3}$$

where B_1 is the identity matrix of dimension 4,

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -r_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

$$B_3 = \begin{bmatrix} -\cos\theta_3 & -\sin\theta_3 & 0 & -r_3\cos\theta_3 \\ \sin\theta_3 & -\cos\theta_3 & 0 & r_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and

$$B_i = \begin{bmatrix} -\cos\theta_i & -\sin\theta_i & 0 & -r_i\cos\theta_i \\ \sin\theta_i\cos\omega_i & -\cos\theta_i\cos\omega_i & -\sin\omega_i & r_i\sin\theta_i\cos\omega_i \\ \sin\theta_i\sin\omega_i & -\cos\theta_i\sin\omega_i & \cos\omega_i & r_i\sin\theta_i\sin\omega_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

for $i = 4, \ldots, n$.

Given the distances r_2, r_3 and the angle θ_3, it is possible to compute the torsion matrices B_2 and B_3 to determine the first three points:

$$x_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$x_2 = \begin{pmatrix} -r_2 \\ 0 \\ 0 \end{pmatrix} \text{ and}$$

$$x_3 = \begin{pmatrix} r_3\cos\theta_3 - r_2 \\ r_3\sin\theta_3 \\ 0 \end{pmatrix}.$$

The *sin* of the torsion angle ω_4 can have only two possible values: $\sin\omega_4 = \pm\sqrt{1-(\cos\omega_4)^2}$ [4]. Consequently, we obtain only two possible positions x_4 and $x^{'}{}_4$ for the fourth point:

$$x_4 = \begin{bmatrix} -r_2 + r_3\cos\theta_3 - r_4\cos\theta_3\cos\theta_4 + r_4\sin\theta_3\sin\theta_4\cos\omega_4 \\ r_3\sin\theta_3 - r_4\sin\theta_3\cos\theta_4 - r_4\cos\theta_3\sin\theta_4\cos\omega_4 \\ -r_4\sin\theta_4\sqrt{1-(\cos\omega_4)^2} \end{bmatrix},$$

$$x_4^{'} = \begin{bmatrix} -r_2 + r_3\cos\theta_3 - r_4\cos\theta_3\cos\theta_4 + r_4\sin\theta_3\sin\theta_4\cos\omega_4 \\ r_3\sin\theta_3 - r_4\sin\theta_3\cos\theta_4 - r_4\cos\theta_3\sin\theta_4\cos\omega_4 \\ r_4\sin\theta_4\sqrt{1-(\cos\omega_4)^2} \end{bmatrix}.$$

In the metric multidimensional scaling problem, instead of $\mathbb{R}^3$ points, the n points are in $\mathbb{R}^m$. It is possible do define any order among these points such that the triangle inequality is strictly satisfied: $\forall i \in 2, \ldots, n-1,\ \delta_{i-1,i+1} < \delta_{i-1,i} + \delta_{i,i+1}$. However, in $\mathbb{R}^3$, we obtain a representation of n points which maintains the same distances given in $\mathbb{R}^m$ between points $i-1$ and i, for $i = 2, \ldots, n$, and also between points $i-2$ and i, for $i = 3, \ldots, n$ [1]. With this $\mathbb{R}^3$ representation, and using the known distances among points on $\mathbb{R}^m$, we apply the *Branch-and-Prune* algorithm for the Cartesian coordinates to project the points on $\mathbb{R}^3$.

3. The Branch-and-Prune Algorithm

In this section, we shall present a Branch-and-Prune algorithm designed for solving the considered problem. The approach is very simple and mimics the structure of the problem closely: at each step, we can place the ith point in two possible positions x_i and $x_i^{'}$ [4]. We, then, branch the search and prune away the infeasible branches. More precisely, each of these possible positions must satisfy, for all pairs of preceding distances d_{ij}, $|\|x_i - x_j\| - d_{ij}| \leq \epsilon$, where $\epsilon > 0$ is a given tolerance. There are four possible outcomes:

1. x_i and $x_i^{'}$ are feasible: in this case we store both positions and exploit both branches in a depth-first fashion;

2. only x_i is feasible: we only store the feasible position x_i and prune the infeasible branch $x_i^{'}$;

3. only $x_i^{'}$ is feasible: we only store the feasible position $x_i^{'}$ and prune the infeasible branch x_i;

4. neither position is feasible: we prune both branches and backtrack the search.

In the original approach of the Branch-and-Prune algorithm pruning is performed based upon computational errors, since the data are from the same space in which the structure must be built. The pruning strategy for MDS is based on the Dijkstra's algorithm. In this case, only the best results obtained thus far are used, ignoring, but not discarding, the remaining ones, since, in another further step, they can become the best results obtained that far. The pruning strategy was modified because we can work with data that are in different spaces in which the structure must be built.

4. Computational Results

In this section, we compare the Branch-and-Prune algorithm for MDS as the principal coordinates analysis technique. The first method minimizes stress function (1) and second one minimizes the function strain, both generate approximate solutions. The errors

$$\text{Max Norm} = \max_{1\leq i\leq n} \sum_{j=1}^{n} \|\delta_{ij} - d_{ij}\|$$

and

$$\text{2-Norm} = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (\delta_{ij} - d_{ij})^2 \right)^2$$

are used in the presentation of the results. The algorithm was implemented in *Matlab2010*, in a processor Intel Core 2 duo 2.66GHz and operating system MAC OSX.

Table 1: Comparison between the two approaches. 10 points in $\mathbb{R}^4$.

Norms	*Algorithm*	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*
Max Norm	PCoA	2.4447e+001	2.4549e+001	1.2968e+001	2.4845e+001	1.8924e+001
	BP	6.3943e+001	6.3591e+001	4.8128e+001	6.8481e+001	3.3440e+001
2-Norm	PCoA	5.6808e+001	5.5648e+001	3.3469e+001	4.9149e+001	3.9135e+001
	BP	1.0944e+002	1.1.95e+002	8.5909e+001	1.2072e+002	4.8618e+001

Table 1 presents an 10 points instance in $\mathbb{R}^4$. As we can see, the clear advantage that the principal coordinates analysis algorithm has over the Branch-and-Prune algorithm in the two tested norms.

In Tables 2 and 3, the tests are performed for 50 points in Euclidean space with dimensions 30 and 50, respectively. It can be obsereved that the branch-and-prune approcah performs better as the number of points increases.

Table 2: Comparison between the two approaches. 50 points in $\mathbb{R}^{30}$.

Norms	*Algorithm*	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*
Max Norm	PCoA	7.3033e+002	7.1910e+002	6.7215e+002	8.3959e+002	6.9022e+002
	BP	7.4140e+002	7.6529e+002	7.3575e+002	7.1736e+002	7.5809e+002
2-Norm	PCoA	1.9101e+004	1.8804e+004	1.8232e+004	1.9057e+004	1.8871e+004
	BP	6.7788e+003	7.4840e+003	6.2412e+003	6.4460e+003	6.7704e+003

Table 3: Comparison between the two approaches. 50 points in $\mathbb{R}^{50}$.

Norms	*Algorithm*	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*
Max Norm	PCoA	1.1228e+003	1.1568e+003	1.1757e+003	1.1305e+003	1.0819e+003
	BP	2.3067e+001	1.1817e+003	1.1257e+003	1.2390e+003	1.1842e+003
2-Norm	PCoA	3.3441e+004	3.4091e+004	3.3082e+004	3.3131e+004	3.2803e+004
	BP	1.1755e+004	1.1717e+004	1.0973e+004	1.1312e+004	1.0866e+004

5. Conclusion

The proposed algorithm has been tested with data generated randomly. It is well known that the principal coordinates analysis algorithm accumulates errors as the source dimension. In comparison with the principal coordinates analysis algorithm, the branch-and-prune algorithm for metric multidimensional scaling accumulates fewer errors increase as the amount of data and size data source. Which confirmed that the efficiency of branch-and-prune algorithm for metric multidimensional scaling is better than the efficiency of principal coordinates analysis.

References

[1] Alencar, J., Alonso, A., Carvalho, S., Lavor, C., Oliveira, A. (2012), *Different orders for discretization of multidimensional scaling problems*, Em Preparação.

[2] Borg, I. and Groenen, P. (2010), *Modern Multidimensional Scaling: Theory and Applications*, (Springer, Berlin).

[3] Classical Multidimensional Scaling, disponível em: <http://www.mathworks.com/products/statistics/examples.html> Acesso em: 25 de março de 2013. Às 10:00hs.

[4] Liberti, L., Lavor, C., Maculan, N. (2008), *A Branch-and-Prune algorithm for the Molecular Distance Geometry Problem*, International Transactions in Operational Research, 15:1-17.

[5] Phillips, A.T., Rosen, J.B., and Walke, V.H. (1996), *Molecular structure determination by convex underestimation of local energy minima*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 23:181-198.

[6] Ziilinskas, A., Ziilinskas, J. (2009), Branch and Bound algorithm for multidimensional scaling with city-block metric, Journal of Global Optimization, 43:357-372.

DGA 2013, pp. 59 – 64.

A distance based sensor location algorithm

Júlio C. Alves[1,3], Ricardo M. A. Silva[2], Geraldo R. Mateus[3], and Mauricio G.C. Resende[4]

[1] *Department of Computer Science, Federal University of Lavras, Lavras, MG 37200-000, Brazil,* julio.caburu@gmail.com

[2] *Center of Informatics, Federal University of Pernambuco, Recife, PE 50740-560, Brazil,* rmas@cin.ufpe.br

[3] *Department of Computer Science, Federal University of Minas Gerais, BH, MG 31270-010, Brazil,* mateus@dcc.ufmg.br

[4] *Algorithms and Optimization Research Department, AT&T Labs Research, 180 Park Avenue, Room C241, FP, NJ 07932, USA,* mgcr@research.att.com

Abstract The sensor location problem (SLP) in a wireless sensor network consists in estimating the position or sensors geographic coordinates from (1) a subset of all pair-wise distances between sensors (often affected by noise) and (2) the positions previously known of some of them. In this paper, we propose a heuristic for the SLP. Experimental results illustrate the effectiveness of the algorithm on four instances of Niewiadomska-Szynkiewicz and Marks (2009) [6].

Keywords: Wireless sensor network, sensor location, optimization heuristic

1. Introduction

Let $X = \{1, 2, \ldots, n\}$ be a set of sensors, $A = \{1, 2, \ldots, m\}$ be a set of anchors with known locations $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\} \ : \ \mathbf{a}_i \in \mathbf{R}^k$, $d_{ij} : i, j \in X, i \neq j$ and $e_{ik} : i \in X, k \in A$ be distances between two sensors and between sensors and anchors, respectively. The sensor location problem consists in finding location for the sensors in X, say, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \ : \ \mathbf{x}_i \in \mathbf{R}^k$, such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = d_{ij}^2 \ \forall \ i, j \in X, i \neq j, \tag{1}$$

$$\|\mathbf{x}_i - \mathbf{a}_k\|_2^2 = e_{ik}^2 \ \forall \ i \in X, k \in A. \tag{2}$$

Since this polynomial system may be inconsistent if the distances d_{ij} or e_{ik} have errors or noise, the sensor location problem can be formulated as a global optimization problem of finding the minimizer $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of

$$\sum_{i,j \in X, i \neq j} |\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 - d_{ij}^2| + \sum_{i \in X, k \in A} |\|\mathbf{x}_i - \mathbf{a}_k\|_2^2 - e_{ik}^2|. \tag{3}$$

If the minimum of objective function (3) is zero for $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, then the constraints in Equations (1) and (2) are also satisfied, and thus the formulations are equivalent. On the other hand, second formulation is relaxed in the sense that it allows approximate solutions while quantifying the approximation error.

The most frequently used approach to solve the sensor location problem is based on Semidefinite Programming (SDP) or Second-Order Conic Programming (SOCP) relaxations. In [1, 4], the authors show that if the optimal solution is unique and noise is additive and multiplicative, then the SDP is able to find the solution. In [7], further SDP relaxations have been proposed

to reduce the dimension of the SDP relaxations, both in variables and constraints, while in [2] the author proposes an heuristic multistage approach to solve the problem formulated as a box-constrained optimization problem.

The remainder of this paper is organized as follows. In Sections 2 and 3 the algorithm is described and related computational results are discussed, respectively. Finally, concluding remarks are made in Section 4.

2. Heuristic for the sensor location problem

At a glance, the proposed heuristic is an iterative procedure that determines the location of one sensor at a time. We process the sensors with unknown location in decreasing order of neighbors with known position, be them anchors or sensors whose location were determined in the previous iterations. Having selected a sensor $i \in X$, $\mathbf{x}_i$ is determined via trilateration first, if possible, falling back to circles intersection if two reference neighbors are available, and then to a neighbor's radius. Trilateration determines the position of a given sensor based on distances to three reference nodes whose locations are known. However, at least two problems may arise when relying on trilateration alone: (1) we cannot ignore the possible existence of noise in the distance labels; and (2) there might exist sensors for which no three reference nodes exist at all. Under such circumstances, we may adopt a tolerance parameter to accept the resulting location. For those sensors with only two neighbors with known position, we employ circles intersection. Knowing that it results in two distinct positions, an extra step is required to decide which one should be accepted as the true location of the sensor. Finally, for sensors with just one reference neighbor, we must rely on the circumference around such a neighbor, i.e., the neighbor's radius alone. To improve the resulting locations even further, the heuristic applies the path-relinking intensification strategy ([3]). The basic element of path-relinking is the construction of an *elite set* of top and diverse solutions (with respect to Equation 3).

Before we detail the algorithm described in Figure 1, let us first introduce its input parameters: X: set of sensors; A: set of anchors; inf: network's lower bound; sup: network's upper bound; d: distances between sensors; e: distances between sensors and anchors; f: function that returns the average error in node distances; ar: radio reach; nf: noise factor; ne: number of solutions in the elite-set; t: error lower bound, i.e., errors less than or equal to t are considered zero; ft: tolerance with respect to noise; $ftmin$: tolerance with respect to the network area; $maxI$: maximum tries with the same error tolerance.

The heuristic is called multi-start, that is, it repeats the instructions in lines 1–30 until a stopping condition is not reached, e.g., a given number of iterations, running time, solution quality, etc. For each start, in line 2 a tolerance of any node's error ϵ is defined as the maximum between (1) the network's coverage divided by $ftmin$ and (2) the noise divided by ft. We let T be the set of nodes whose position is already determined and S be the elite-set, both initialized with the empty set in line 3. For convenience, in line 2 we also set s as the input data to the problem, and then build a list P of nodes with unknown location — those that are to be located —, sorted in decreasing order of neighbors with known location.

In the inner loop spanning lines 4–20, we try to determine the location of any sensor in P by trilateration (line 5), if unsuccessful then by circles intersection (line 6), and finally by neighbor's radius (line 7) if still unsuccessful. If, on the one hand, a node's position, say $\mathbf{x}$pn, is successfully determined, we update the error associated with each neighbor of pn (line 17), update T by adding sensor pn to the set, and update P (line 18). On the other hand, if there is no node whose location can be deducted (line 8), we increase the number i of iterations without success (line 9) and repeat the inner loop, since our implementation of trilateration, circles intersection, and neighbor radius are randomized and can yield different outcomes in each run. However, if no solution can be found after $i = maxI$ iterations (line 10), we increase

```
Algorithm LocRSSF (X, A, low, up, d, e, f, r, nf, ec, t, nc, ft, ftmin, maxI)
1    while (stopping condition not reached) do
2       s := {X, A, d, e}; P :=BuildPending(s); ε := max(((up − low)/ftmin), (nf/ft));
3       i := 0; T := ∅; S := ∅; nrr := false;
4       while (|P| > 0) do
5          pn :=TryTrilateration(s, P, t, ε, nc)
6          if (pn = null) then pn := Try2CircleInt(s, P, rr, nf, t, ε, nc);
7          if (pn = null) then pn := TryNeighborRadius(s, P, t, ε, nc, nrr);
8          if (pn = null) then
9             i := i + 1;
10            if (i ≥ maxI) then
11               if (nf > 0 and ε < nf) then
                       { ε := ε + nf/ft; i := 0; }
12               else { nrr := true; ε := ε ∗ 10; }
13            endif
14         else
15            i := 0;
16            if (nrr) then { nrr := false; ε := ε/10; }
17            for each neighbor n already positioned of pn do
                    n.e := ComputeNodeError(n, s, n.x, n.y, nf);
18            T := T ∪ {pn}; P := BuildPending(s);
19         endif
20      endwhile
21      s.e := f(s);
22      if (|S| < ec and NewSolution(s, S)) then S.InsertInOrder(s);
23      else
24         if (s.e < S[1].e) then { S.InsertAtHead(s); S := S \ {MostSimilar(s)}; }
25         else
26            i := Random(|S|); s′ := PathRelinking(.., s, S[i], t);
27            if (s′.e > S[|S|].e and SolutionIsDifferent(s′, S)) then
                    { S.InsertInOrder(s′); S := S \ {MostSimilar(s′)}; }
28            endif
29      endif
30   endwhile;
31   return (S[1]);
end LocRSSF
```

Figure 1: WSNL algorithm.

the error tolerance in lines 11–12: for instances with noise, we increment ϵ with nf/ft (line 11), whereas for instances without noise we simply increase it ten times (line 12), also relaxing the subsequent calls to neighbor's radius back.

When a solution is built, or equivalently, when the set of sensors whose position is unknown is empty, we consider adding it to the elite-set S (lines 22–29) that will feed the path-relinking heuristic (line 26). In this process, if the elite-set is full but the current solution is attractive (i.e., different from those already in S), it then enters S replacing a solution in S worse than the current one and most similar to the current one (line 27). Otherwise, if the elite-set is not full, the current solution is inserted, in increasing order of error, into S in line 22. Moreover, if the solution is not the best among those already in S, the path-relinking procedure is executed.

Otherwise, it is inserted at head of S, replacing a solution most similar to it (line 24). Given two solutions K1 and K2 such that the error of of K1 is smaller than the error of K2, path-relinking iteratively transforms K1 into K2 as follows: for each iteration, an "unmarked" sensor of K2 is randomly selected, "marked", and its position is copied into K1. At the end of each iteration, we evaluate the error in the updated K1, saving the best solution among all iterations. At the end of the procedure, we save this best solution into S. When finished, the algorithm returns the best solution in the elite-set as the ideal solution to the problem.

3. Experimental results

In this section, we report the experimental results comparing our heuristic with the four algorithms introduced by Niewiadomska-Szynkiewicz and Marks in 2009 [6] for the WSNL problem: Semidefinite programming (SDP), Simulated Annealing (SA), Trilateration with Simulated Annealing (TSA), and Trilateration with Genetic Algorithm (TGA). Among the five test instances used by Niewiadomska-Szynkiewicz and Marks, we selected four: evenly, unevenlyA, unevenlyB, and unevenlyC. The instances have 200 sensors, 20 anchors, a radio range (rr) equals to 0.18 and a noise factor equals to 0.1. While the instance evenly has sensors uniformly distributed, the instances unevenlyA, B and C have sensors concentraded in some place, with unevenlyC having anchors also concentrated.

All experiments were run on a 2 GHz Core 2 Duo CPU with 2 GBytes memory, running under Linux. The algorithm was implemented in C++ and compiled with GCC version 4.3.3. For each test instance, we made 100 independent runs of the algorithm, using as random-number generator an implementation of the Mersenne Twister algorithm performed by [5]. After a parameter tuning phase, we set the input parameters as follows: stopping criteria = 20 iterations, ne = 10, t = 0.001, ft = 10, ftmin = 10, and maxI = 10. Besides this, we adopted as quality measure of the solutions the error metric ϵ used by Niewiadomska-Szynkiewicz and Marks in [6] and its corresponding standard deviation (s.d.), as well as the average running times of the algorithms in seconds as performance measure. Although the average running times coming from our heuristic are considerably higher than all of other algorithm's, once that it generates multiple greedy randomized solutions while also executing path-relinking in between; Table 1 shows that, except for the instance evenly, the quality of our results was always better than those presented in [6]. For example, while the best error found by Niewiadomska-Szynkiewicz and Marks's algorithms (in this case, the TGA method) was 133.78% on instance unevenlyC, our heuristic achieved 3.74%. Therefore, contradicting the authors' statement described in [6]: "As a final result, we can say that it is not suggested to apply distance-based location methods to networks with unevenly distributed non-anchor and anchor nodes". Representing through a line segment the error between the real and estimated position of the sensors given by the algorithms, Figures 2 and 3 illustrate in more details the differences among positions calculated by our heuristic on instance unevenlyC, and those generated by the algorithms of Niewiadomska-Szynkiewicz and Marks (2009). While the first one determines with high precision the positions of sensors, the others do not.

Table 1: Summary of results for the four algs. of Niewiadomska-Szynkiewicz and Marks (2009) and our heuristic (Alg) on four instances: evenly, unevenlyA, B, and C [6]. Times are given in seconds and errors in percentage.

Instance	ϵ **SDP**	t **SDP**	ϵ **SA**	t **SA**	ϵ **TSA**	t **TSA**	ϵ **TGA**	t **TGA**	ϵ **Alg**	s.d.(ϵ **Alg**)	t **Alg**
evenly	0.18	6.95	2.76	3.04	**0.13**	0.46	3.80	2.85	0.27	0.060	13.13
unevenlyA	174.91	5.51	233.89	2.85	1.78	0.44	20.61	2.34	**1.21**	0.548	24.41
unevenlyB	330.56	6.25	293.01	3.06	1.81	0.47	56.06	2.90	**0.65**	0.209	18.47
unevenlyC	434.83	8.95	446.13	3.84	433.09	0.61	133.78	3.46	**3.74**	2.934	27.97

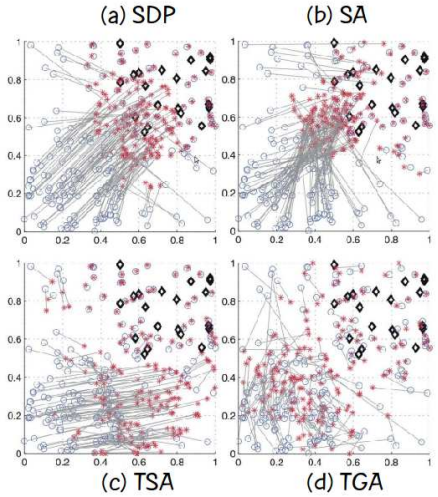


Figure 2: Positions (star notation) of sensors determined by the SDP, SA, TSA and TGA algorithms on unevenlyC instance

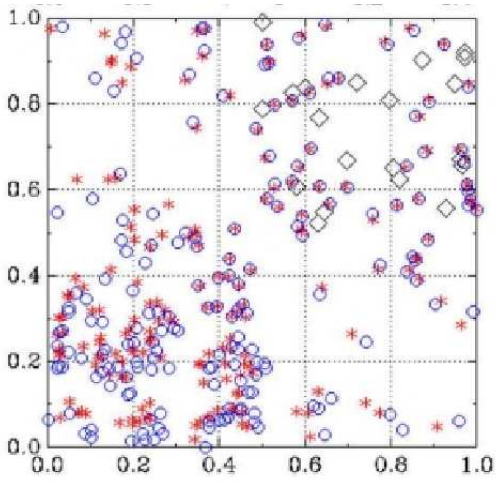


Figure 3: Positions (star notation) of sensors determined by the heuristic on unevenlyC instance.

4. Concluding remarks

In this paper, we propose an algorithm for sensor location problem. We have shown the results of applying this heuristic to four instances introduced by Niewiadomska-Szynkiewicz and Marks in 2009 [6]. The promising results shown here, indicate that it is appropriate for solving sensor location problem.

Acknowledgment

The research was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq), the Foundation for Support of Research of the State of Minas Gerais, Brazil (FAPEMIG), Coordination for the Improvement of Higher Education Personnel, Brazil (CAPES), AT&T Labs Research in Florham Park, NJ, USA, and Foundation for the Support of Development of the Federal University of Pernambuco, Brazil (FADE).

References

[1] Pratik Biswas, Kim-Chuan Toh, and Yinyu Ye. A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM J. Sci. Comput.*, 30(3):1251–1277, 2008.

[2] A. Cassioli. Solving the sensor network localization problem using an heuristic multistage approach. *Optimization Online*, 2009.

[3] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39:653–684, 2000.

[4] T.C. Liang, T.C. Wang, and Y. Ye. A gradient search method to round the semidenite programming relaxation solution for ad hoc wireless sensor network localization. Technical Report SOL 2004-2, Dep. of Management Science and Engineering, Stanford University, California, USA, 2004.

[5] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8:3–30, 1998.

[6] Ewa Niewiadomska-Szynkiewicz and Michał Marks. Optimization schemes for wireless sensor network localization. *International Journal of Applied Mathematics and Computer Science*, 19(2):291–302, 2009.

[7] Z. Wang, S. Zheng, Y. Ye, and S. Boyd. Further Relaxations of the Semidefinite Programming Approach to Sensor Network Localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008.

DGA 2013, pp. 65 – 69.

Adaptive Branching in *i*BP with Clifford Algebra

Rafael Alves,[1] Andrea Cassioli,[2] Antonio Mucherino,[3] Carlile Lavor,[1] and Leo Liberti [2]

[1] *IMECC-UNICAMP, Campinas-SP, Brazil.* rafaelsoalves@uol.com.br,clavor@ime.unicamp.br

[2] *LIX, École Polytechnique, Palaiseau, France.* cassioli@lix.polytechnique.fr,liberti@lix.polytechnique.fr

[3] *IRISA, University of Rennes 1, Rennes, France.* antonio.mucherino@irisa.fr

Abstract We consider the *interval* Discretizable Molecular Distance Geometry Problem (*i*DMDGP). This is a subclass of instances of the Distance Geometry that can be discretized; they are related to biological molecules and can contain imprecise measurements of the available distances. The *interval* branch-and-prune (*i*BP) is an algorithm for the *i*DMDGP. In this short paper, we integrate *i*BP with Clifford algebra, with the aim of improving the branching phase of the algorithm, by making it adaptive.

Keywords: molecular conformations, distance geometry, branch-and-prune, Clifford algebra

1. Introduction

Experiments of Nuclear Magnetic Resonance (NMR) are able to identify a subset of distances between pairs of atoms of a given protein. This information, together with some additional information on the chemical structure of the protein, can be exploited for finding the possible three-dimensional conformations for the molecule. This problem is known as the *interval* Molecular Distance Geometry Problem (*i*MDGP) [3].

In this context, we are working on a subclass of *i*MDGPs that can be discretized. In other words, we consider all instances of this problem for which the search domain can be reduced to a tree, whose nodes at level j represent all possible Cartesian coordinates for the j^{th} atom of the molecule. We say that such instances belong to the class of the *interval* Discretizable Molecular Distance Geometry Problem (*i*DMDGP) [1].

The discretization allows for employing an *interval* Branch & Prune (*i*BP) algorithm for the solution of *i*DMDGPs [1]. The idea is to explore the search tree recursively and to verify, as soon as they are generated, the feasibility of the computed atomic positions. Infeasible positions are immediately pruned, so that the search can be focused on the feasible parts of the tree. On each layer of the tree, a finite number of possible Cartesian coordinates for the current atom are computed by intersecting three Euclidean objects in the three-dimensional space. Two of such objects always consist of spheres, whereas the third one may be either a sphere or a spherical shell, depending on the fact the available distance is precise or represented by an interval, respectively.

When the three Euclidean objects are three spheres, their intersection, in our assumptions, always gives two disjoint points in $\mathbb{R}^3$, with probability one (the set of DMDGP instances for which this fails has zero Lebesgue measure in the set of all possible DMDGP instances; many of our statements hold with probability one). However, when one of the three objects is a spherical shell, this intersection gives two curves in the three-dimensional space. A curve is

a continuous object, and therefore, in order to discretize it, it was proposed in [1] to take a certain predefined number D of points from such curves, and to include a new branch in the tree for each new generated point.

Computational experiments, reported in some previous publications [4, 4–1], showed that the solutions to iDMDGPs can be strongly influenced by the choice of D. If D is too small, only infeasible branches may be generated, so that the whole tree is pruned and no solutions can be found. On the other hand, if D is too large, the consequent combinatorial explosion might make the experiments too computationally expensive. Finding a trade-off D value is not an easy task in general.

This work presents a strategy for an adaptive branching during the execution of the iBP algorithm, which is based on the so-called Clifford algebra [3, 7]. The main idea is to generate branches that comply with the pruning distances, i.e. a minimal number of branches are kept in the tree, while the tree width can be controlled.

In Section 2, we shortly describe how to extend the iBP algorithm by implementing a strategy based on Clifford algebra. Section 5 shows some preliminary computational experiments.

2. Extending iBP with Clifford algebra

The Clifford algebra Cl_3 over the real numbers is a 8-dimensional space with basis elements $\{1, e_1, e_2, e_3, e_{12}, e_{13}, e_{23}, e_{123}\}$ representing scalars, vectors, bivectors and trivectors. The bivectors and trivectors are obtained by the *geometric product*, the main product in a Clifford algebra. This product is represented by the juxtaposition of the elements, and its rules for basis vectors are shown in Equations (1) and (2). Other two products can be derived from the geometric product: the outer product "$\wedge$" and the contractions, left "$\rfloor$" and right "$\lfloor$". For vectors, the contractions are equivalent to the scalar product in $\mathbb{R}^3$ "$\cdot$". Some relations among these products are shown in Equations (3) and (4) below. The geometric product between two vectors is the sum of the scalar and the exterior products between them. In general, the geometric product between a vector and an arbitrary element of Cl_3, called a multivector, is the sum of the left contraction and the exterior product.

$$e_i^2 = 1, \tag{1}$$
$$e_i e_j = -e_j e_i, \tag{2}$$
$$uv = u \cdot v + u \wedge v, \tag{3}$$
$$uB = u \rfloor B + u \wedge B. \tag{4}$$

In Equations (3) and (4), u and v are vectors and B is a multivector of Cl_3.

With the addition of two vectors to the $\mathbb{R}^3$ basis (e_∞ and e_0), it is possible to construct a model of geometry that allows us to handle with several basic geometric entities in a simple way. We refer to the Clifford algebra associated to this model as the Conformal Geometric Algebra (CGA) [3]. The basis elements for the conformal space are $\{e_1, e_2, e_3, e_\infty, e_0\}$, where e_∞ represents a point at infinity and e_0 represents the origin of $\mathbb{R}^3$ in the conformal space.

In the CGA, basic geometric entities such as points, spheres, circles, lines and planes are represented in a simple way, and their intersections are performed intuitively. The outer product is used to compute intersections or to construct objects from points. The contractions can also be used for intersections, and are often used to compute orthogonal projections, distances and angles. Our approach is based on the geometric interpretation of the iDMDGP. A sphere is represented by Eq. (5), while the circle can be defined simply as a two sphere intersection, Eq. (6). Another important element is called a *Point Pair*, which is the result of a three sphere

intersection, Eq. (7). These elements are the most important ones in our approach:

$$S = X - \frac{1}{2}r^2 e_\infty, \tag{5}$$
$$C = S_1 \wedge S_2, \tag{6}$$
$$Pp = S_1 \wedge S_2 \wedge S_3. \tag{7}$$

In Eq. (5), r is the sphere radius and $X = x + \frac{1}{2}x^2 e_\infty + e_0$ is the projection of a point $x \in \mathbb{R}^3$ in the conformal space.

The basic idea behind the proposed strategy is the following. Every time the current atom has one reference distance that is represented by an interval, two spheres S_{i-1} and S_{i-2}, related respectively to the atomic positions x_{i-1} and x_{i-2}, are intersected with the spherical shell S'_{i-3}, related to the atomic position x_{i-3} such that the distance $d_{i-3,i}$ is an interval. Due to the discretization assumptions, this intersection produces up to two disjoint curves c'_{i-3} and c''_{i-3} in the three-dimensional space. These two curves belong to the circle C obtained by intersecting the two spheres S_{i-1} and S_{i-2} (Figure 1). In symbols, c'_{i-3}, $c''_{i-3} \in C$.

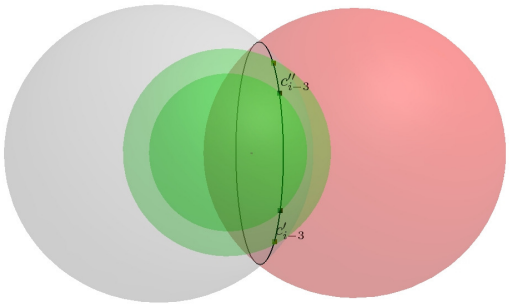


Figure 1: Intesection between two spheres and a spherical shell.

Let us suppose that there is an atom k that is already positioned, for which $d_{k,i}$ is a known distance. This distance generates another spherical shell S'_k, which can be intersected with the circle C, by producing other two curves: c'_k, $c''_k \in C$ (Figure 3). The same intersection can be computed for any other atom j for which $d_{j,i}$ is known. After all the intersections have been computed, the remaining curves contain only feasible points. These curves can be described by the rotation of its endpoints (Figures 2 and 3). In other words, if E_p is the set of all known distances, and $E_p(i)$ is the subset of E_p containing only the distances related to the atom i, the feasible points on the circle C can be computed by the following formula:

$$F(i) = \bigcap_{j \in E_p(i) \cup \{i-3\}} \left(c'_j \cup c''_j \right).$$

The set $F(i)$ is a curve segment in Euclidean space. It only contains feasible points, i.e. points that satisfy all the available distances for the current atom. As a consequence, for any relatively small value for D, there is no risk of pruning all the feasible points in the pruning phase. In some sense, D can be considered as the precision of the conformations that are going to be included in iBP solution set.

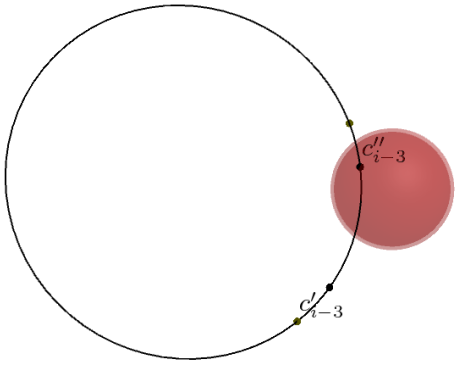


Figure 2: Intersection with a pruning distance $d(i,k)$.

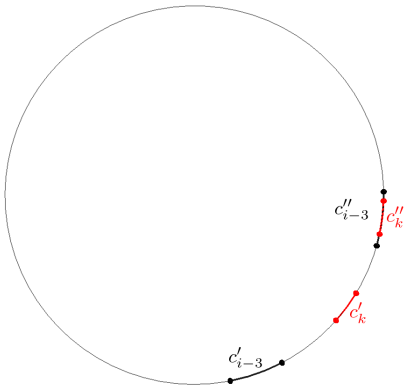


Figure 3: The curve c''_k is the only feasible region.

name	n_{aa}	n_a	l	C-iBP	iBP
hm30	4	18	28	4/13803	6/929
2jmy-s	5	26	42	3/5220	4/3902
2jmy-m	10	51	90	3/11172	6/140270
2kxa	23	117	206	3/1275	8/7942
2ppz	36	170	323	4/15618	6/43073
2jmy	15	77	134	3/19063	12/47681

Table 1: Computational results: for each instance we report D/#nodes.

3. Preliminary computational experiments

In this section, we summarize our preliminary experiments on the iBP algorithm with Clifford algebra. We show that the number of discretization points that are necessary for finding at least one feasible solution decreases when the new strategy based on Clifford algebra is employed.

Both algorithms are coded in C++ and share most of the code, which has been compiled with the `g++` compiler version 4.7 with optimization flags `-O3 -DNDEBUG`. The tests have been performed on a laptop having an i3 Intel processor and running Linux.

Our instances contain real data from NMR experiments, that can be downloaded from the Protein Data Bank [4, 8]. They have different sizes, and they are related to protein backbones only [1]. For each instance, we report the number of aminoacids n_{aa}, the number of atoms n_a, the order length l, i.e. the tree depth, the minimum number of discretization points to find at least one solution D, and number of generated nodes #nodes. We refer to iBP with Clifford algebra as C-iBP.

The results in Table 1 clearly show that the use of the Clifford algebra allows for a great reduction of the branching factor to find feasible solutions.

References

[1] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliand, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, *The protein data bank*, Nucleic Acids Research, 28, 235–242, 2000.

[2] V. Costa, A. Mucherino, C. Lavor, L.M. Carvalho, N. Maculan, *On Suitable Orders for Discretizing Molecular Distance Geometry Problems related to Protein Side Chains*, IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS12), Workshop on Computational Optimization (WCO12), Wroclaw, Poland, 397–402, 2012.

[3] L. Dorst, D. Fontijne, S. Mann, *Geometric Algebra for Computer Science: An object-oriented approach for geometry*, Morgan Kaufmann Publishers Inc., 2007.

[4] C. Lavor, L. Liberti, A. Mucherino, *On the Solution of Molecular Distance Geometry Problems with Interval Data*, IEEE Conference Proceedings, International Workshop on Computational Proteomics (IWCP10), International Conference on Bioinformatics & Biomedicine (BIBM10), Hong Kong, 77–82, 2010.

[5] C. Lavor, L. Liberti, A. Mucherino, *The interval Branch-and-Prune Algorithm for the Discretizable Molecular Distance Geometry Problem with Inexact Distances*, to appear in Journal of Global Optimization, 2013.

[6] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, Tech. Rep. 1205.0349v1 [q-bio.QM], arXiv, 2012.

[7] P. Lounesto, *Clifford Algebras and Spinnors*, Cambridge University Press, 2001.

[8] A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, N. Maculan, *Influence of Pruning Devices on the Solution of Molecular Distance Geometry Problems*, Lecture Notes in Computer Science **6630**, P.M. Pardalos and S. Rebennack (Eds.), Proceedings of the 10^{th} International Symposium on Experimental Algorithms (SEA11), Crete, Greece, 206–217, 2011.

DGA 2013, pp. 71 – 75.

A Clifford Algebra approach to the Discretizable Molecular Distance Geometry Problem

Alessandro Andrioni[1]

[1]*IMECC, University of Campinas, Campinas, Brazil,* andrioni@member.ams.org

Abstract The Discretizable Molecular Distance Geometry Problem (DMDGP) consists in a subclass of the Molecular Distance Geometry Problem for which an embedding in $\mathbb{R}^3$ can be found using a Branch & Prune (BP) algorithm in a discrete search space. We propose a Clifford Algebra model of the DMDGP with an accompanying version of the BP algorithm.

Keywords: Distance geometry, Clifford algebra, Branch and prune

1. The discretizable molecular distance geometry problem

The molecular distance geometry problem (MDGP) consists in finding coordinates in a three-dimensional space of a set of points $\{x_1, x_2, \ldots, x_n\}$ for which some of the Euclidean distances between them are known [4]. Let $G = (V, E, d)$ be a simple weighted undirected graph where each vertex in V corresponds to a point in $\mathbb{R}^3$, and the weight of an edge corresponds to the distance d between the respective points. Formally, the MDGP can be defined as follows [12]:

Definition 1. *(MDGP). Let $G = (V, E, d)$ be a simple weighted undirected graph. The MDGP is the problem of finding a function*

$$x : V \to \mathbb{R}^3$$

such that

$$\forall (u, v) \in E,\ ||x_u - x_v|| = d_{uv},$$

where $x_u = x(u)$ and $x_v = x(v)$.

The discretizable molecular distance geometry problem (DMDGP) is a subset of the MDGP, but with two extra assumptions [12]:

Definition 2. *(DMDGP). Let $G = (V, E, d)$ be a simple weighted undirected graph associated to an instance of the MDGP. Let us suppose that there is a total order relation on the vertices of V. The DMDGP consists in all the instances of the MDGP satisfying the following two assumptions:*

1. *E contains all cliques on quadruplets of consecutive vertices;*
2. *the following strict triangular inequality must hold:*

$$\forall v \in \{1, \ldots, n-2\},\ d_{v,v+2} < d_{v,v+1} + d_{v+1,v+2},$$

where n is the number of vertices in V.

The reasons why these assumptions are useful and realistic is outside of the scope of this work, and is extensively addressed in [12] and [14], but they allow us to discretize the problem in the following manner: suppose we have three points in $\mathbb{R}^3$, and the distances to a fourth point. We can then construct three spheres, each one centered in one of the points and with a radius of its distance to the fourth one. These three spheres have an intersection characterized by two points (with probability 1, thanks to the assumptions above [14]): the two possibilities for the fourth point. However, if we have additional information, we can decide whether one of them is invalid or not. Repeating this process, we have at most 2^{n-3} ways to position n points (up to rotation and translation).

The Branch & Prune (BP) is an algorithm proposed in [13] to solve the DMDGP by exploiting this discretization. The BP was further developed [15] to use another characteristic of the discretization, namely that when constructing a new point, the two possibilities are symmetric in regard to the plane determined by the three previous points. This means we can construct other realizations from an initial one by knowing just the different branches taken and then applying reflections through the right planes.

2. Clifford algebra

Clifford algebras are a refinement of both the Hamilton quaternions and the extensive algebra of Hermann Grassmann [3] condensed in one structure. Clifford himself called his work *geometric algebra* [2], but the term most commonly used now is Clifford algebra. His work had a geometric flavor, and was heavily explored by both mathematicians and physicists, including Élie Cartan and Paul Dirac, usually in the context of differential geometry and quantum mechanics [5].

A revival of the use of real Clifford algebras for geometric purposes was spearheaded by David Hestenes [10], and culminated in the modern geometric algebra and its operational models of Euclidean geometry, including conformal geometric algebra (CGA) [9]. CGA allows a rich representation of Euclidean motions in a coordinate-free manner, and the link between distance geometry and conformal geometric algebra was already studied by Dress and Havel [6].

We try to follow the notation and the formulation introduced in [5], and recommend it as a good introduction to the subject, but we give a summary of central Clifford algebra ideas used in this work.

There are two main products which are used: the geometric (or Clifford) product, and the outer (or wedge) product[1]. Both of them algebraically encode the idea of working with *oriented subspaces* of a vector space, allowing a *"multivector"* representation of points, lines, planes and hyperplanes. An interesting fact is that the geometric product allows the *inverse* of some multivectors to be defined, and thus it permits the representation of orthogonal or even conformal (in CGA) transformations using an object called versor.

A versor is the result of a multiplication of vectors using the geometric product, it is always invertible and it is applied to another multivector by "sandwiching", that is, if V is a versor and A a multivector, we can apply the versor by calculating VAV^{-1}. Versors also correctly preserve its underlying geometric structure without need for adaptations, being then a suitable basis for the representation of geometric computations.

Definition 3. *(Conformal geometric algebra of the three-dimensional Euclidean space). The conformal geometric algebra (CGA) of the three-dimensional Euclidean space is an extension of* $\mathbb{R}^3$ *by means of two extra vectors* e *and* $\bar{e}$*, which square respectively to* 1 *and* -1*. However, it is more convenient to use* $\infty = \bar{e} - e$ *and* $o = \frac{1}{2}(\bar{e} + e)$*, both of which square to* 0*, and represent*

[1]The geometric product of a and b is denoted by ab, and their outer product by $a \wedge b$.

a point at infinity and a point at the origin, respectively. This permits a fully coordinate-free representation of Euclidean geometry, a fact which will be exploited in our algorithm.

In CGA, versors encode all conformal transformations, including isometries and homotheties. In fact, it is the smallest known model of Euclidean geometry which allows the full representation of Euclidean transformations as versors. For convenience, we introduce special names to two kinds of versors: those which represent rotations (rotors), and those which represent translations (translators).

Composition of rotors and translators is more efficient than that of rotation matrices in up to 10 dimensions and uses less storage in up to 6 dimensions, a good evidence of the appropriateness of using CGA for geometric computing. It is also simple to convert versors to matrices, if the need arises [5].

3. BP with Clifford algebra

We assume that our instance is a molecule of n atoms for which we denote the *bond lengths* $d_{i-1,i}$ for $i = 2, \ldots, n$, the *bond angles* $\theta_{i-2,i}$ for $i = 3, \ldots, n$ and the *dihedral angles* $\omega_{i-3,i}$ for $i = 4, \ldots, n$.

The main idea of the BP with Clifford Algebra is to use *rotors* and *translators* to represent the calculation of a point from its predecessors, instead of transformation matrices based on homogeneous coordinates [12]. Since a combination of rotor and translator needs at most 8 coordinates to be represented, this represents a memory gain against the traditional 4×4 matrices.

Another advantage of using CGA to work on the DMDGP is to exploit the inherent symmetries in the problem, since reflection through a plane is a simple operation in CGA represented by a reflection versor, cheapening considerably the cost of calculating alternative conformations.

An algorithm to calculate the two possible points from its predecessors is presented here as Algorithm 2. It creates two rotors: one for the bond angle and one for the dihedral angle, and apply them to a translator to generate F, an Euclidean transformation which takes x_{i-1} to x_i. Notice in the algorithm the particular way the rotors are constructed, which is reminiscent of Euler's formula and the polar forms of complex numbers or quaternions. The pruning phase is implemented as in the original BP [14].

A computer implementation of this new version of the BP already exists for the GAViewer software [5], but as the software was not made with efficiency needs in mind, it is only useful as a visualization tool. A "production-ready" implementation using the software Gaigen [7] capable of using data extracted from the Protein Data Bank is currently being written, and should be complete in time for the DGA2013, along with a performance analysis and a comparison with the existing implementations of the original BP [13].

Algorithm 1 Initial potential solution

```
procedure INITIAL POTENTIAL SOLUTION(n, d, θ, ω)
    solution ← {the three initial points}
    i ← 4
    while i ≤ n do
5:      Compute i-th point and add to solution
    end while
    return solution
end procedure
```

Algorithm 2 Compute i-th point

procedure ALGORITHM I: COMPUTE i-TH POINT($x_{i-3}, x_{i-2}, x_{i-1}, \theta, \omega, d$)
$\Pi \leftarrow x_{i-3} \wedge x_{i-2} \wedge x_{i-1} \wedge \infty$;
$R_1 \leftarrow e^{\frac{\theta}{2}((\Pi x_{i-2} \wedge x_{i-1}) \wedge \infty)^*}$; ▷ Create the bond angle rotor
$v \leftarrow x_{i-1} - x_{i-2}$;
$\omega' \leftarrow \omega - \frac{\pi}{2}$;
$R_2 \leftarrow e^{\frac{\omega'}{2}(x_{i-2} \wedge x_{i-1} \wedge \infty)^*}$; ▷ Create the dihedral angle rotor
$T \leftarrow 1 - \frac{d}{2}\frac{v}{||v||}\infty$; ▷ Create the translator
$F \leftarrow R_2 R_1 T R_1^{-1} R_2^{-1}$; ▷ Combine two rotors and a translator in one versor
$x_i \leftarrow F x_{i-1} F^{-1}$ ▷ Apply it x_{i-1}
$x_i' \leftarrow \Pi x_i \Pi^{-1}$ ▷ Reflect x_i
return x_i, x_i'
end procedure

4. Conclusions and future work

This is one of the first practical applications of conformal geometric algebra in distance geometry and it shows its excellence in representing complex geometric operations in a simple manner. The subsumption of both quaternions and homogeneous coordinates by CGA allows it to clarify the notation and to better expound inherent geometric properties in problems.

We expect to see more developments in that regard in the future, as more accessible, efficient and high-level implementations of geometric algebra appear, as it is suited for scientific computing and is already being used in the fields of robotics, computer graphics, computer vision and artificial neural networks [1] [5] [16].

An extension of this work to handle generalizations of the DMDGP is expected, as conformal geometric algebra has a great richness in ways of creating and manipulating its objects. Another possible line of work would be to try to apply CGA techniques to other problems involving molecular symmetries and rotations, such as [8] and [11].

Acknowledgments

The authors would like to thank the Brazilian research agencies FAPESP, CNPq and CAPES for the financial support.

References

[1] E. Bayro-Corrochano. *Geometric Computing - for Wavelet Transforms, Robot Vision, Learning, Control and Action.* Springer, 2010.

[2] W. Clifford. Applications of Grassmann's extensive algebra. *American Journal of Mathematics*, 1:350–358, 1878.

[3] J. Collins. An elementary exposition of Grassmann's "Ausdehnungslehre," or theory of extension. *The American Mathematical Monthly*, 6:193–198, 1899.

[4] G. Crippen and T. Havel. *Distance Geometry and Molecular Conformation.* Wiley, New York, 1988.

[5] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry.* Morgan Kaufmann Publishers Inc., San Francisco, 2007.

[6] A. Dress and T. Havel. Distance geometry and geometric algebra. *Foundations of Physics*, 23:1357–1374, 1993.

[7] D. Fontijne. Gaigen 2: a geometric algebra implementation generator. In *Proceedings of the 5th international conference on Generative programming and component engineering*, GPCE '06, New York, 2006. ACM.

[8] H. Fritzer. Molecular symmetry with quaternions. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 57:1919–1930, 2001.

[9] D. Hestenes. Old wine in new bottles: A new algebraic framework for computational geometry. In *Advances in Geometric Algebra with Applications in Science and Engineering*, pages 1–14, 2001.

[10] D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Fundamental Theories of Physics. Springer, 1987.

[11] C. Karney. Quaternions in molecular modeling. *Journal of Molecular Graphics and Modelling*, 25:595–604, 2007.

[12] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[13] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.

[14] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. Tech. Report q-bio.qm/1205.0349, arXiv, 2012.

[15] A. Mucherino, C. Lavor, and L. Liberti. Exploiting symmetry properties of the discretizable molecular distance geometry problem. *Journal of Bioinformatics and Computational Biology*, 10, 2012.

[16] C. Perwass. *Geometric Algebra with Applications in Engineering*. Springer Berlin Heidelberg, 2009.

DGA 2013, pp. 77 – 82.

Performance Comparison of Overdetermined Multilateration Algorithms for Estimating Aircraft Position

Anderson Avila,[2] Fabiano Prado,[1] Guiou Kobayashi[2] and Eduardo Rocha[2]

[1]*Universidade Federal de Uberlândia (UFU), Minas Gerais, Brazil* fprado.ufu@gmail.com

[2]*Universidade Federal do ABC (UFABC), São Paulo, Brazil* anderson.avila@ufabc.edu.br, guiou.kobayashi@ufabc.edu.br, eduardo.elias.tsi@gmail.com

Abstract The problem of obtaining information on position location can be solved by using *time of arrival* and *time difference of arrival* based techniques. Throughout this paper, we compare both iterative and non-iterative (closed-form) methods for surveillance purpose in airports. As iterative solutions, we present four methods: Quasi-Newton, Taylor and two Genetic Algorithms. As non-iterative solutions, four algorithms are considered. Three of them had to be adapted in order to obtain an overdetermined system. The results show how the number of nodes influences accuracy, failure rate and processing time.

Keywords: Multilateration, TOA, TDOA, Hyperbolic Positioning

1. Introduction

The goal of locating the coordinates of an aircraft can be achieved by measuring the range between a point of interest and reference nodes[1]. Some of the techniques include calculating angle of arrival (AOA), received signal strength intensity (RSSI), time of arrival (TOA) and time difference of arrival (TDOA)[1]. The last two approaches will be addressed in this paper and are considered the promising solution to the next generation surveillance (NextGen)[2][4].

The problem of estimating an object location in the airport area has received considerable attention due to the importance of securing the constantly increasing global air traffic. In fact, MLAT[2] systems can be considered as the transition from Secondary Surveillance Radar (SSR) systems to Automatic Dependent Surveillance - Broadcast (ADS-B) systems[2][3][5].

An overdetermined system are assumed during the simulations (i.e, the number of linear equations are greater than the number of unknowns). In another words, it consists of more than three reference nodes for TOA and TDOA based algorithms.

The performance of several MLAT algorithms are analysed in this paper under different numbers of receivers. The goal was to observe how the process of increasing the number of reference nodes affects the average error, failure rate and the processing time.

This paper is organized as follows. Section I discusses the geometry of a MLAT system. Section II describes the methodology used in order to conduct this research. In Section III, the results are presented and in Section IV, the conclusion are discussed.

[1]Reference nodes and receivers will be used interchangeably throughout the paper.
[2]Abbreviation of Multilateration: systems based on time difference of arrival (TDOA).

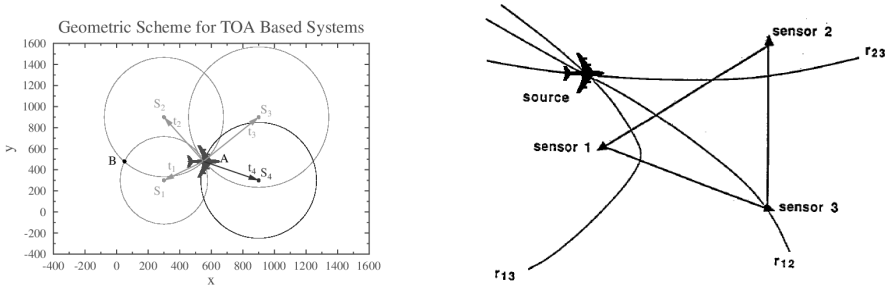
2. Multilateration System and its Geometric Interpretation

In a Multilateration system, the signal transmitted by a source (e.g, a transponder) and received by three or more ground based receiver sites is denominated time of arrival (TOA). Considering a 2-dimensional Euclidean space, let $S_i = (x_i, y_i)$ be the reference node locations and t_i the respective arrival time, where $i = 1, ..., N$. The unknowns aircraft position and time of emission (TOE) are given by $S_a = (x_a, y_a)$ and t_e, respectively.

Assuming a Line of Sight scenario (LOS), the distance between S_i and S_a can be represented by (1), being c the speed of light.

$$c\,(t_i - t_e) = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2} \tag{1}$$

If we consider that the emission occurs at time 0, the TOA based techniques will require at least three equations (i.e, three reference nodes) for two unknowns (x_a, y_a)[11]. Figure 1-a depicts an overdetermined system, where $N > 3$.



(a) Overdetermined system (TOA based) (b) Hyperbolas intersection (TDOA based)

Figure 1: Multilateration schemes

Notice that using only two receivers, e.g, S_1 and S_2, gives us two solutions, A and B, depicting an ambiguity. Thus, it is necessary the use of at least another reference node, S_3, in order to determining A as the correct solution. The fourth node characterizes an overdetermined system.

Another approach for estimating an aircraft position is to calculate the difference in time of arrival (TOA) between pairs of reference nodes. Hyperbolic positioning, or time difference of arrival (TDOA), is a technique that consider the intersection of hyperbolas as the mobile position[7][8]. The distance-difference to a source can be represented by (2).

$$c\,(t_i - t_j) = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2} - \sqrt{(x_j - x_a)^2 + (y_j - y_a)^2} \tag{2}$$

The emission time is common for all the receivers, thus the equation above eliminates the unknown t_e which is in accordance with our goal, since we are only interested in measuring the arrival time range. The system is depicted in Figure 1-b[3].

[3]Font: A Passive Localization Algorithm and Its Accuracy Analysis.[9]

3. Performance Assessment and Experiments

3.1 Algorithms

Eight location estimation algorithms form the object of interest of this research. Four of them are non-iterative methods[4]: Bancroft [11], Bucher[7], Bakhoum[10], Least Squares[12]. Among these algorithms, only Bancroft[11] is based on TOA. The Least Squares method can optionally be implemented as TOA or TDOA based[12]. All closed-form algorithms, except for the Least Squares, required an adaptation to ensure an overdetermined approach.

As iterative methods, four solutions were considered. Two genetic algorithms based on heuristic search were implemented, besides Taylor and Quasi-Newton methods [13][12][14][15]. All of these solutions are based on TDOA, except for one of the genetic algorithms.

3.2 Surveillance Area

Although, all the algorithms were implemented using the C++ programming language, to build a simulation environment, we used Matlab. We were able to link our C++ code with Matlab through an interface named MEX[5].

The airport surface was assumed to be $1Km^2$ of surveillance area discretized in m^2. The reference nodes were placed arbitrarily in 16 fixed sites as shown in Figure 2. For each algorithm,

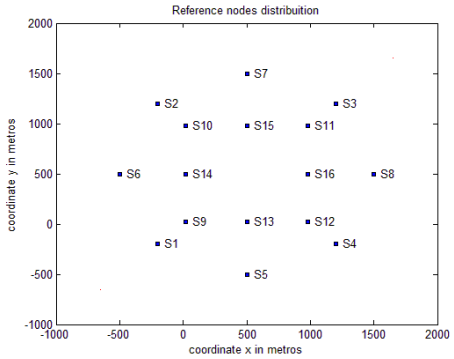


Figure 2: Position of the reference nodes

the simulation started with the number of 4 receivers (S_1, S_2, S_3, S_4), geometrically forming a square. The other nodes were added up sequentially, according to its number. Sensors 1 to 8 were located outside of the surveillance area and Sensors 9 to 16 were located inside that area. Since the errors associated to the hardware are unknown, we hypothetically assumed a Gaussian error distribution associated to the TOA measurements, where $\mu = 0$ and $\sigma = 1$,2,3.

3.3 Simulation

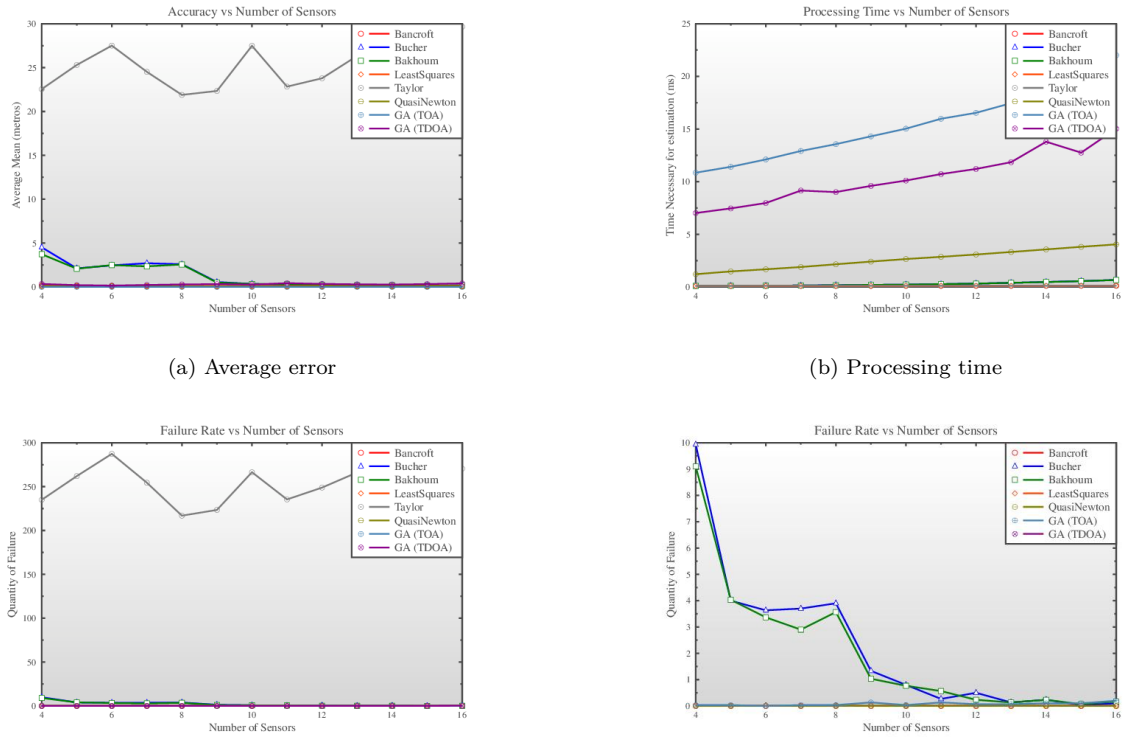
To obtain the three parameters analysed in this paper, all algorithms were submitted to estimate 500 random points in the surveillance area. This process were repeated 31 times[6] for each algorithm and every time a reference node were added up to the system. Therefore, for a specific number of receivers, 15.500 points were estimated by each algorithm. The average error were obtained by summing up the error and dividing it by the number of processed

[4]In this paper, the non-iterative methods will be referred as Bancroft, Bucher and Bakhoum
[5]`http://www.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html`
[6]31 showed to be a suited number to reach a reliable result.

points. The failure rate were characterized by the number of times that the algorithm could not define the coordinates (i.e, NaN[7] or Inf[8] were returned) or the coordinates represented an error above 20 meters. Finally, the processing time were determined by summing up the time required for estimation and dividing it by the number of processed points.

4. Results

The goal of the simulations were to measure the performance of each algorithm as the number of reference nodes were increased. It is clear that the Taylor method has the worst precision as we can see on Figure 3-a and 3-c and the performance remains the same as the number of receivers grows. The same behaviour is not experienced by the other algorithms. Both average error and failure rate improves as the number of reference nodes are increased. This effect is even more evident on Bancroft's and Bucher's algorithms. As expected, the most computationally demanding algorithms are those base on iterative methods, especially the ones that use evolutionary approach, as we can verify on Figure 3-b.



(a) Average error

(b) Processing time

(c) Failure rate

(d) Failure rate without Taylor results

Figure 3: Performance comparison: mean and processing time

Except for the Taylor method, Table 1 shows that increasing the number of reference nodes increases the processing time and decreases the average error.

[7]Not a number.
[8]Divided by zero

Nodes	*Bancroft*	*Bucher*	*Bakhoum*	*LeastSquares*	*Taylor*	*Quasi-Newton*	*GATOA*	*GATDOA*
4	0.1202	4.5136	3.7308	0.1269	22.5520	0.1092	0	0.3418
5	0.1145	2.1159	2.0789	0.1294	25.3050	0.1058	0	0.2057
6	0.0957	2.4586	2.4906	0.1236	27.5100	0.0879	0	0.1563
7	0.0889	2.7016	2.3469	0.1159	24.5190	0.0800	0	0.2255
8	0.0843	2.5896	2.5628	0.1109	21.8920	0.0766	0	0.2858
9	0.0812	0.5570	0.5014	0.1105	22.3410	0.0730	0	0.3258
10	0.0784	0.3472	0.3446	0.1128	27.4880	0.0710	0	0.2721
11	0.0751	0.2520	0.2984	0.1083	22.8340	0.0662	0	0.4306
12	0.0736	0.2581	0.2262	0.1085	23.7870	0.0648	0	0.3459
13	0.0736	0.1935	0.1944	0.1101	26.3310	0.0621	0	0.2983
14	0.0728	0.1876	0.1881	0.1133	27.8050	0.0612	0	0.2774
15	0.0701	0.1593	0.1652	0.1112	28.7340	0.0571	0	0.3263
16	0.0692	0.1610	0.1605	0.1112	29.6640	0.0561	0	0.4019

(a) Average Error (m)

Nodes	*Bancroft*	*Bucher*	*Bakhoum*	*LeastSquares*	*Taylor*	*Quasi-Newton*	*GATOA*	*GATDOA*
4	0.1051	0.1027	0.1003	0.09506	0.0998	1.2115	10.8520	7.0198
5	0.1062	0.1117	0.1076	0.09783	0.1004	1.4746	11.4110	7.4599
6	0.1040	0.1170	0.1180	0.09943	0.1035	1.6774	12.1100	7.9715
7	0.1060	0.1389	0.1382	0.09883	0.1019	1.8934	12.9160	9.1551
8	0.1072	0.1643	0.1602	0.0957	0.1061	2.1534	13.5630	9.0107
9	0.1097	0.1919	0.1945	0.0988	0.10657	2.4092	14.3050	9.5993
10	0.1112	0.2236	0.2215	0.1009	0.1066	2.651	15.0350	10.1060
11	0.1118	0.2641	0.2661	0.1024	0.1077	2.8595	15.9750	10.7230
12	0.1127	0.3218	0.3162	0.09976	0.1066	3.0821	16.5390	11.2090
13	0.1113	0.389	0.39	0.1003	0.1098	3.3217	17.454	11.8510
14	0.1164	0.4711	0.4669	0.1004	0.1096	3.5661	18.0860	13.7930
15	0.1165	0.5579	0.5511	0.1018	0.1134	3.8125	18.8140	12.7570
16	0.1165	0.6636	0.6604	0.1019	0.1102	4.0429	22.0110	15.0320

(b) Processing Time (ms)

Table 1: Mean absolute values obtained from the simulations

We can infer from the values above that Bancroft's and Quasi-Newton's algorithms present the best performance in terms of precision. Although, the Genetic Algorithm based on TOA offers an exact solution, it is also true that this algorithm is limited to a $1m$ of precision.

5. Conclusion

In this paper, four iterative and non-iterative algorithms are evaluated under an overdetermined circumstance. We verified how each algorithm react as the number of reference nodes increases. We were able to check the best and worst performances regarding average error, failure rate and processing time. Among the iterative and non-iterative algorithms, Bancroft and Quasi-Newton methods outperform the others. As future approaches, we consider optimizing the evolutionary algorithms in order to improve their performance. We may investigate the dilution of precision in order to find the best geometric positions to place the receivers and also verify qualitatively where and when the failure happens, i.e, the algorithm's singularity.

Acknowledgments

The authors wish to thank FINEP and CNPq for all the support, encouragement and the initiative to take this journey. This article is a result of the FINEP project 01.10.0492.00.

References

[1] D. Muñoz, et al. Position Location Techniques and Applications. Elvesier, 2009.

[2] N. Xu, et al. Performance Assessment of Multilateration Systems - A Solution to NextGen Surveillance. IEEE Integrated Communications Navigation and Surveillance (ICNS) Conference. May 2010.

[3] J.M. Abbud, G. Miguel and J. Besada. Correction of Systematic Errors in Wide Area Multilateration. IEEE Proceedings of Enhanced Surveillance of Aircraft and Vehicles. September 2011.

[4] G. Galati, et al. Multilateration Applied to Airport Vehicles Management Systems: The Agile Transponder. IEEE Proceedings of the 3rd European Radar Conference. September 2006.

[5] A. Moni, S. Rickard. Comparison of Location Algorithms Using Attenuation Estimates. IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. January 2009.

[6] A. Jasch, et al. Geometrical Siting Considerations for Wide Area Multilateratioin Systems. IEEE Position Location and Navigation Symposium (PLANS). 2010 IEEE/ION.

[7] R. Bucher, D. Misra. A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System. VLSI Design, 2002 Vol. 15 (2), pp, 507-520.

[8] Y.T. Chan, K.C. Ho. A Simple and Efficient Estimator for Hyperbolic Location. IEEE Transactions on Signal Processing, Vol. 42, NO. 8. August 1994

[9] B. Friedlander. A Passive Localization Algorithm and Its Accuracy Analysis. IEEE Journal of Oceanic Engineering, Vol. OE-12, NO. 1. January 1987.

[10] E.G. Bakhoum. Closed-Form Solution of Hyperbolic Geolocation Equations. IEEE Transactions on Aerospace and Eletronics Systems, VOL. 42, NO. 4. October 2006.

[11] M. Geyer, A. Daskalakis. Solving Passive Multilateration Equations Using Bancrofts' Algorithm. The AIAA/IEEE/SAE Digital Avionics Systems Conference. Proceeding 17th DASC. October 1998.

[12] G. Shen, R. Zetik, R.S. Thomä. Performance Comparison of TOA and TDOA Based Location Estimation Algorithms in LOS Environment. IEEE Proceedings of the 5th Workshop on Positioning, Navigating and Communication 2008.

[13] S. Forrest. Genetic Algorithms:Principles of Natural Selection Applied to Computation. Science, New Series, Vol. 261, No. 5123, pp 872-878. August 1993.

[14] Yu, et al. UWB location and tracking for wireless embedded networks. Signal Processing 86 (2006), pp 2153-2171.

[15] R. Fletcher. Practical Methods of Optimization, 2000. Chapters 1,2 and 3.

DGA 2013, pp. 83 – 88.

On the using of distances to measure goodness of fit in Item Response Theory models: a Bayesian perspective

Caio Lucidius Naberezny Azevedo[1] and Jose R. S. Santos[2]

[1,2]*Department of Statistics, University of Campinas, Brazil* cnaber@ime.unicamp.br,robertosilv258@yahoo.com.br

Abstract Many statistical tools for model validation and comparison are based on some suitable measures of distance (discrepance). Examples are: deviance residual, chi-squared type statistics, odds ratio and Mahalanobis distance. In Item Response Theory (IRT), which comprises a widely used set of psychometric models, some of these distance-type statistics can be used to verify the validity of many important assumptions such as: unidimensionality, the adequability of the item response function, the adequability of the latent trait distribution, the presence of DIF (Differential item functioning) among others. However, under a frequentist approach, the using of these statistics can be complicated because their distributions, under the null and alternative hypothesis, are usually not known. On the other hand, under the Bayesian paradigm, the obtaining of the so-called Bayesian p-values, related to these statistics, through MCMC algorithms, is feasible and straightforward. In this work, we explore the using of some of these statistics, under the Bayesian paradigm, to verify the validity of some usual assumptions for unidimensional IRT models for dichotomous responses. More specifically, through simulation studies, we intend to verify the relationship of some of these measures of distance with the departing of some assumptions. With the results we intend to understandig how suitable the aforementioned measures of distance are in detecting the departing of the aforementioned assumptions.

Keywords: Item response theory, Model fit assessmente, Bayesian Inference

1. Introduction

The Item Response Theory (IRT) comprises a set of widely used psychometric models. The most basic elements of this class of models, establish relationships between the so-called item parameters and the latent traits, through statistical models. In their turn, these models consider the probability of subjects get a certain score in each item, based on the responses of these subjects to these items. In general, the items are clustered in some measurement instrument, as a cognitive test or a psychiatric questionnaire. See [4] and [5], for more details. In this work, we will focus on our attention in dichotomous items, or item which are corrected as right/wrong. In addition, we will consider the most basic IRT models, that is, models that consideres only the item parameters and the latent traits.

As in any statistical model, it is necessary to consider some assumptions in order to estimate the parameters and to obtain results that are both interpretable and useful. Therefore, for a given data set, the model used to analyze it, must be fit to the data, properly. For the unidimensional IRT models to dichotomous responses, the usual assumptions are: the unidimensionality of the latent traits, the adequability of the item response function (IRF) and the adequability of the latent traits distribution. Several methods have been proposed in the literature for model fit assessment. The works of [9], [10], [11], [12] and [13], present some reviews concerning this

topic. Commonly, some type of distance (as the deviance residuals, chi-square type statistics and Mahalanobis's distance) are used, see [3] and [12]. Under a frequentist approach, the using of such distances can be complicated because their distribution, under both null hypothesis (the model assumption holds) and alternative hypothesis (the model assumption does not hold) are, in general, unknown. However, under the Bayesian paradigm, it is not necessary to know their distribution and the validity of the model assumptions can be checked by considering the so-called Bayesian p-values associated with these statistics (distances). Even though many works have explored the using of such statistics under the Bayesian paradigm, more detailed studies concerning the behaviour of them in identifying the lack of the aforementioned assumptions in the IRT model are necessary. The main goal of this work is to study the behavior of some distance-type statistics for undimensional IRT models for dichotomous responses. Our focus is to study the performance of these statistics in terms of identifing the lack of model fit by the violation of some of the aforementioned assumptions and in identifing when the assumptions hold. We will consider simulation studies for different situations in terms of number of subjects and size test. In the following subsections we will provide more details about the measures of distance and the simulation studies that will be considered.

2. Undimensional IRT models for dichotomous responses

Let Y_{ij} be a random variable which assumes the value 1 if the subject j answers the item i correctly and 0 otherwise. We assume that each subject is submitted to a test of I items and a response matrix of 0's and 1's is available. One of the most used IRT models for dichotomous responses is the three-parameter model:

$$P_{ij} = P(Y_{ij} = 1|\theta_j, \boldsymbol{\zeta}_i) = c_i + (1 - c_i)F\left[a_i\left(\theta_j - b_i\right)\right] , \tag{1}$$

where θ_j is the latent trait of the subject j (which can represent the knowledge level in Mathematics, the depression level, among other possibilities), a_i is the discrimination parameter of item i, b_i is the difficulty parameter of item i, c_i is the guessing parameter of item i, $\boldsymbol{\zeta}_i = (a_i, b_i, c_i)$ and $F(.)$ is a cdf (cumulative distribution function) of interest. Usual choices for $F(.)$ are the logistic and probit functions. For the first choice, the model (1) becomes

$$P_{ij} = P(Y_{ij} = 1|\theta_j, \boldsymbol{\zeta}_i) = c_i + (1 - c_i)\frac{1}{1 + e^{-a_i(\theta_j - b_i)}} . \tag{2}$$

For the second choice, we have:

$$P_{ij} = P(Y_{ij} = 1|\theta_j, \boldsymbol{\zeta}_i) = c_i + (1 - c_i)\Phi\left[a_i\left(\theta_j - b_i\right)\right] , \tag{3}$$

where $\Phi(x) = \displaystyle\int_{-\infty}^{z} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz$. Figures 1 and 2 show examples of curves, called item characteristic curve (ICC), for the model given by equation (2). We can see that, for a given value of parameter b, the higher is the value of parameter a the steeper is the ICC. On the other hand, for a given value of the parameter a, the higher is the value of the parameter b, the more shifted to the right is the ICC. In addition, the higher is the value of the parameter c the closer to 1 is the ICC, for the examinees with low values of the latent traits. More details can be found in [4] and [5]. Another interesting feature of the IRT models is that the scale of the latent traits is completely arbitrary (in our example we consider a latent trait scale with mean equal to 0 and standard deviation equal to 1).

For a $(n \times I)$ matrix of (0,1) responses, the parameters (θ, a, b, c) can be estimated by using several methods as the marginal maximum likelihood, marginal maximum a posterior, CADEM (Condicional Augmented Data EM) algorithm and by using a fully Bayesian approach trough MCMC (Monte Carlo Markov Chain) algorithms, see [4], [1], [8] and [2], for example.

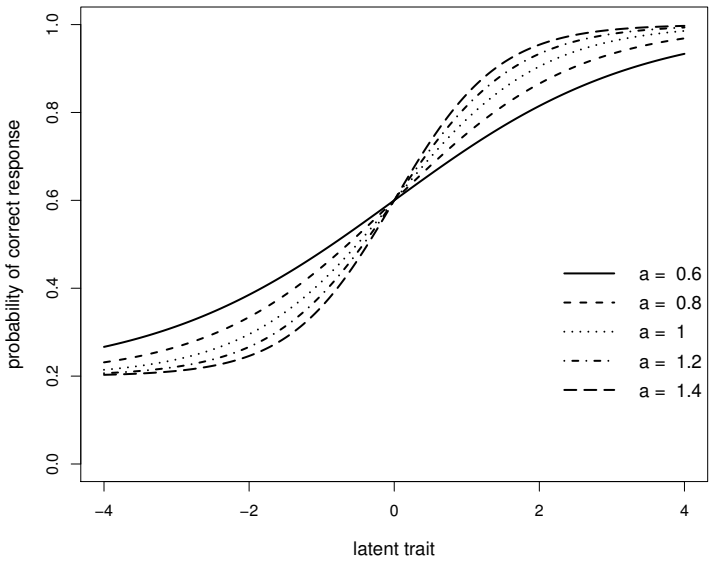


Figure 1: Examples of ICC's for the three-parameter logistic model for different values of discrimination parameter (b=0.0;c=0.20)

3. Measuring the Goodness of fit in IRT models through distance type statistics

3.1 Chi-square type distance

Let NC_k denote the number of examinees getting exactly k items correct, $k = 0, 1, 2, ..., I$. [6] and [7] suggest compare the observed and predicted score distributions to measure the overall model fit. To summarize the model fit to the observed score distribution in a single number, Beguin and Glas (2001) suggested the using of the discrepancy measure $\chi^2_{NC} = \sum_{k=1}^{I} \frac{(NC_k - E(NC_k))^2}{E(NC_k)}$, where $E(NC_k)$ is the expectation of NC_k. Although the statistic χ^2_{NC} does not follow a chi-square distribution, the Bayesian p-value provides a measure of overall goodness of fit. Clearly, the discrepancy measure χ^2_{NC} is a distance, once that measures the difference between the observed value NC_k and the expeceted value $E(NC_k)$.

3.2 Residual type distance

For each $Y_{ij}, i = 1, ..., I; j = 1, ..., n$ we can define the deviance residual, that is

$$DR_{ij} = \left[\sqrt{-2\ln\left(1 - P_{ij}\right)}\right]^{1-y_{ij}} \left[\sqrt{2\ln P_{ij}}\right]^{y_{ij}} .$$

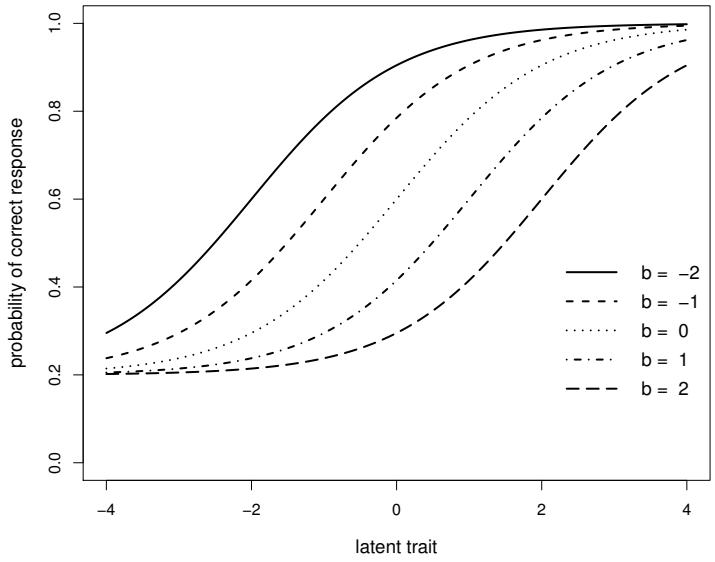


Figure 2: Examples of ICC's for the three-parameter logistic model for different values of difficulty parameter (a=1.0;c=0.20)

It is clear that, depending on the value of the response (y_{ij}), the value of the deviance residual changes. Therefore, it can be seen that the above quantity measures the difference beteween the observed response and the expected reponse.

3.3 Other type distance

Consider an item pair in the test. Let $n_{kk'}, k, k' \in \{0, 1\}$, the number of subjects scoring k on the first item and k'on the second item. The odds-ratio is defined as:

$$OR = \frac{\frac{n_{00}}{n_{01}}}{\frac{n_{10}}{n_{11}}} = \frac{n_{00}n_{11}}{n_{01}n_{10}}$$

in other words, we are measuring the distance between the proportion of incorrecet/correct answers among each pair of items.

4. Calculating Bayesian p-values for the distance type statistics

Let $\boldsymbol{y}^{obs}$ be the matrix of observed responses, and $\boldsymbol{y}^{rep}$ the matrix of replicated responses generated from its posterior predictive distribution. The posterior predictive distribution of the response data of group k is represented by

$$p\left(\boldsymbol{y}^{rep} \mid \boldsymbol{y}^{obs}\right) = \int p\left(\boldsymbol{y}^{rep} \mid \boldsymbol{\vartheta}\right) p\left(\boldsymbol{\vartheta} \mid \boldsymbol{y}^{obs}\right) d\boldsymbol{\vartheta},$$

where $\boldsymbol{\vartheta}$ denotes the set of model parameters considered in the discrepancy measure. Generally, given a discrepancy measure $D\left(\boldsymbol{y}, \boldsymbol{\vartheta}\right)$, the replicated data can be used to evaluate whether the

discrepancy value given the observed data is typical under the model. A p-value can be defined that quantifies the extremeness of the observed discrepancy value,

$$p_0\left(\boldsymbol{y}^{(obs)}\right) \quad = \quad P\left(D\left(\boldsymbol{y}^{(rep)},\boldsymbol{\vartheta}\right) \geq D\left(\boldsymbol{y}^{(obs)},\boldsymbol{\vartheta}_k\right) \mid \boldsymbol{y}^{(obs)}\right), \tag{4}$$

where the probability is taken over the joint posterior of $(\boldsymbol{y}^{(rep)},\boldsymbol{\vartheta})$. Consider, for example, the χ^2_{NC} distance. In this case $NC_k = f(\boldsymbol{y}^{obs})$,$E(NC_k) = \sum_{j=1}^{n} P_{ij}$ and $\vartheta = (\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})'$. Therefore, in this case, $D(\boldsymbol{y},\boldsymbol{\vartheta}) = \chi^2_{NC}$. In general, the distances can be defined in order to provide a overall, by item (adequability of the IRF) or by subject (adequability of the latent traits distribution) measure of model fit. The Bayesian p-value can be easily estimated by using the MCMC outputs as simply evaluating the observed proportion as defined in (4). The higher is the value of value of the Bayesian p-value, equation (4), better is the model fit.

5. Simulation study

We will generated a several number of replicas (matrix with responses of the subjects to the items) from different IRT models (one, two and three parameter), with one, two and three latent trait dimension and considering different latent trait distribution (normal, left skewed, right skewed and uniform). With these replicas we will estimate the parameters and calculate the bayesian p-values related to the aforementioned measures of distance, using the one, two an three onedimensional IRT models and a standard normal distribution for the latent traits. Therefore, we wil, study the behavior of the bayesian p-values in detecting the departing of the usual model assumptions (unidimensionality, the correct specification of the item response function and the correct specficitation of the latent traits distribution).

Acknowledgments

The authors wish to thank CNPq and CAPES (Brazilian agencies of financial support for research) for the finnancial support.

References

[1] Azevedo, C. L. N., Bolfarine, H. and Andrade, D. F. (2011). *Bayesian inference for a skew-normal IRT model under the centred parameterization*, Comput. Statist. Data Anal., 55, 353–365.

[2] Azevedo, C. L. N. and Andrade, D. F. (2011). *CADEM: A conditional augmented data EM algorithm for fitting one parameter probit models*, Brazilian Journal of Probability and Statistics, Accepted for publication.

[3] Azevedo, C. L. N., Andrade, D. F. and Fox, J.-P. (2012). *A Bayesian generalized multiple group IRT model with model-fit assessment tools*, Comput. Statist. Data Anal., 56, 4399–4412.

[4] Baker, F. B. and Kim, Seock-Ho. (2004). *Item Response Theory: Parameter Estimation Techniques*. New York, NY: Marcel Dekker.

[5] Linde, W. J. van der and Hambleton, R. K. (2010). *Handbook of Modern Item Response Theory*. New York, NY: Springer-Verlag.

[6] Ferrando, P. J. and Lorenzo-seva, U. (2001). *Checking the appropriateness of item response theory models by predicting the distribution of observed scores: The program EP-fit*, Educational and Psychological Measurement, 61, 5, 895—902.

[7] Hambleton, R. K. and Han, N. (2004). *Assessing the fit of IRT models: Some approaches and graphical displays*. Paper presented at the annual meeting of the National Council on Measurement in Education, San Diego, CA.

[8] Patz, R. J. and Junker, B. W. (1999). *A Straightforward Approach to Markov Chain Monte Carlo Methods for Item Response Models*, Journal of educational and behavioral Statistics, 24, 146–178.

[9] Sandip, S. H. and Stern, H. S. (2003). *Posterior predictive model checking in hierarchical models*, Journal of Statistical Planning and Inference, 111, 209-221.

[10] Sandip, S. H. (2002). *Practical Applications of Posterior Predictive Model Checking for Assessing Fit of Common Item Response Theory Models*, Technical Report, Educational and Testing Service.

[11] Sandip, S. H. (2005). *Assessing Fit of Unidimensional Item Response Theory Models Using a Bayesian Approach*, Journal of Educational Measurement, 42, 375–394.

[12] Sandip, S. H. (2006). *Bayesian item fit analysis for unidimensional item response theory models*, British Journal of Mathematical and Statistical Psychology, 59, 429–449.

[13] Sandip, S. H., Johnson, S. M. and Stern, H. S. (2006). *Posterior Predictive Assessment of Item Response Theory Models*, Applied Psychological Measurement, 30, 298–321.

DGA 2013, pp. 89 – 92.

A Formulation of Stellar Cluster Membership Assignment as a Distance Geometry Problem

Eduardo Bezerra[1], Leonardo Lima [2] and Alberto Krone-Martins [3]

[1]*Federal Center of Technological Education Celso Suckow da Fonseca, Informatics Department, Brazil* edubezerra@gmail.com

[2]*Federal Center of Technological Education Celso Suckow da Fonseca, Production Engineering Department, Brazil,* leolima.geos@gmail.com

[3]*Universidade de Lisboa, Faculdade de Ciências, Portugal,* algol@sim.ul.pt

Abstract Methods designed to solve the Distance Geometry Problem (DGP) aim at finding a valid embedding to a given weighted simple undirected graph, given a set of pairwise distances between points. DGP has been applied to several practical problems such as protein structure detection and Nuclear Magnetic Resonance Spectroscopy. On the other hand, the Stellar Cluster Membership Problem (SCMP) rises from astronomy domain and consists in segregating the field and cluster stars from catalogues which are generated from images taken from a telescope. In this paper, we sketch a formulation of SCMP as a variation of the DGP through the use of concepts from Spectral Graph Theory.

Keywords: molecular distance geometry, astronomy, star clustering

1. Introduction

Open clusters are stellar systems that share a common origin. They have widespread usage in Astronomy as they are key to understanding star formation and evolution as well as galaxy dynamics and structure (see for instance [3, 9, 10]). Certain photometric characteristics of stars in a cluster can point out that they formed from a same primordial cloud within a relatively short time scale. This is important, since all cluster member stars should have approximately the same age and share chemical properties. Furthermore, studies comprising several open clusters can help us to understand both the formation and kinematics of the Galactic disk due to the spread of the cluster age and mass distributions. Thus, the study of open cluster ages, distances, masses, luminosity and mass functions constitute important research topics in Astronomy. In this context, one of the most fundamental challenges is the problem of finding which are the stellar cluster members in a given stellar field.

This problem consists in segregating the field and cluster stars in a given field from catalogues which are generated from images of a telescope. Such procedure is known as membership assignment, or membership segregation (here we adopt *Stellar Cluster Membership Assignment*, or SCMP). Usually, the most widespread data available to solve the problem are the positions of each star as well as their photometric parameters. These parameters are related to the measurement of magnitude of each star in each wavelength passband.

The Distance Geometry problem has been intensively studied due to its applications to various real life problems. In particular, many of these problems are defined in three dimensional space and can be modeled as a Molecular Distance Geometry Problem (MDGP). In [8], ap-

plications in diverse domains are presented, such as in protein conformation, robotics, graph rigidity, data visualization and wireless sensor networks. Usually, the MGDP can be formulated as a minimization of an objective function consisting of a sum of error terms, which turns out to be a continuous Global Optimization problem. A survey of continuous methods to this specific problem can be found at [6]. It is also possible to model the MDGP in a discrete space. A very recent algorithm to solve the discrete version of the problem, called Branch and Prune, was developed by [5].

A common issue when solving SCMP is that there might exist some correlation between the random variables that represent the photometric measurements of each star. Therefore, dimensionality reduction is a step before applying some clustering algorithm. In this paper, we sketch a Spectral Graph Theory based dimensionality reduction approach that can be applied in the context of SCMP.

2. Problem Formulation

The positions projected on the celestial sphere and photometric measurements of astronomical objects are the most widespread type of information accessible to researchers about the Universe. These photometric measurements are based on filters which allow only a certain region of the electromagnetic spectrum to be observed, as different information can be obtained about the stars using different filters and different filter combinations (which are called colors). Hence, astronomical observations usually are performed using different sets of filters, and thus for each star different magnitude measurements are obtained. Examples of photometric filters are U (ultraviolet), B (blue), V (visible), R (red) and I (infrared). Even if usually there is some redundancy between some of these values, and their linear combinations (colors), they can be used to infer some properties of the stars (such as its temperature) or of the interstellar medium between the star and the Earth (the reddening).

In addition to this photometric data, each image also provides the position of the object in the image, which can be converted to a position on the celestial sphere. Although one could first guess that by considering the coordinates of two stars in the picture (i.e, their x-y position), that they are close in 3-dimensional space, this is not true due to the projection effect: in the 3-dimensional space these stars may be very distant. Hence, the challenge is to segregate the field and cluster stars, taking as input the positions and photometric data for each star.

According to the above description, our input data is a dataset of records, each one of them representing a single star in the original telescope image. This dataset can be modeled as a collection of $(m + 2)$-vectors, where m is the number of photometric bands used in the measurement, and they also can be modeled as an weighted undirected graph. Before presenting our proposed model, we introduce some relevant notation and concepts.

2.1 Notation and Problem definition

Let G be an undirected and simple graph. We denote by $V = V(G)$ and $E = E(G)$ its vertex and edge sets, respectively, such that the number of vertices is given by $|V(G)| = n$. Also, we consider G as a weighted graph with a weight function

$$w : V(G) \times V(G) \to \mathbf{R}^+$$

which assigns a real nonnegative weight $w(u, v)$ or w_{uv} to each pair u, v of vertices. It is required that $w_{uv} > 0$ if $uv \in E(G)$ and $w_{uv} = 0$ if $uv \notin E(G)$. Since G is undirected, $w_{uv} = w_{v,u}$ and, if $e = uv$, we write $w(e)$ instead of $w(u, v)$. An embedding of G in R^l is a function $x : V \to R^l$. For simplicity, we write $\mathbf{x}_v$ to $\mathbf{x}(v)$ for any vertex $v \in V$. Also, an embedding is valid for G if

$$\| x_u - x_v \| = w_{uv}$$

for every edge $uv \in E$, where $\| \cdot \|$ is the Euclidean norm. The *Distance Geometry Problem* consists of finding a valid embedding in R^l to a given weighted simple undirected graph G. That can also be thought as the following unrestricted optimization problem:

$$\min_{\mathbf{x}} \sum_{uv \in E} (\| \mathbf{x}_u - \mathbf{x}_v \|^2 - w_{uv}^2)^2,$$

where the distances w_{uv} are known and the vectors $\mathbf{x}_u = \left(x_u^1, \ldots, x_u^l\right)^T$ for each vertex $u \in V$ are unknown.

Let us define the position of a star u in 3-dimensional space by the vector $\mathbf{x}^u = (x^u, y^u, z^u)$. If the coordinates of the vectors $\mathbf{x}^u = (x^u, y^u, z^u)$ for each star u of the field were known, one could easily determine the open cluster, if any. The point is that, while the coordinates x^u and y^u are available, we do not have the coordinate z^u for every star u. On the other hand, for every star u, a collection of n photometric parameters denoted by $p_i,\ i = 1, \ldots, n$ are available. Besides, this photometric data encode information about the unknown coordinate. To better clarify the problem definition, we introduce some notation. Let S be the set of stars with cardinality k and let $m = n+2$ be the quantity of available parameters for each star. So, we write $\mathbf{P} \in R^{k \times m}$ as a matrix such that each row u is given by the vector $\mathbf{P}^u = \{x^u, y^u, p_1^u, \ldots, p_n^u\}$. Next, we define the SCMP.

Definition 1. *Let S be the set of stars and let $\mathbf{P}$ be the matrix of parameters associated to S. The* Stellar Cluster Membership Problem *consists in using the vectors P^u to determine the subset of stars $S' \subset S$ that corresponds to the open cluster.*

Observe that the main challenge here is to extract information from each row P^u of $\mathbf{P}$ in order to segregate the stars from the field, i.e., to determine which stars are in fact clustered in 3-dimensional space. In the following, we present our proposed formulation to the SCMP.

3. Proposed model

Consider each star u as a vertex of a graph G. Since each star u is related to a vector $P_u \in R^m$ we can easily obtain the matrix of weights W computing the Euclidian distances between every pair of vertices u and v, where $u \neq v$. Note that G is a complete undirected weighted graph on the vectorial space R^m. An interesting question that drives our work is the following:

Which is the smallest l such that there is a feasible embedding to G in R^l ?

Originally, we have a graph G in R^m and we want to reduce the dimensionality of this space. We do this in order to latter identify the stars of the open cluster in a lower dimensional space, by applying some clustering algorithm.

We now describe a spectral graph theory based approach to dimensionality reduction for the SCMP. Spectral graph theory studies properties of graphs by applying concepts from Algebra and Linear Algebra in a matrix related to a given graph. Thus, the use of eigenvalues and eigenvectors of a given graph matrix and finding their relation to graph invariants are two of the main topics in this area. One interesting graph matrix is the Laplacian. If G is a weighted undirected graph, the Laplacian matrix is defined by

$$L(G) = D(G) - W(G),$$

where $W(G)$ is the weight matrix and $D(G)$ is the diagonal weight matrix and its entries are row sums of W. Laplacian is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on vertices of G. A survey of this matrix can be found

at [7]. The authors in [1] proposed an algorithm to the dimensionality reduction and clustering using the eigenvectors and eigenvalues of the Laplacian matrix of G. This algorithm is based on four basic steps:

(i) build the graph (in our case the graph G is complete and it has been built already);

(ii) use the Heat kernel function to determine modified weights to the edges of G as a function of the W matrix entries;

(iii) compute the eigenvalues and eigenvectors to the generalized eigenvalue problem

$$L\mathbf{y} = \lambda D\mathbf{y},$$

where λ and $\mathbf{y}$ are the eigenvalue and the eigenvector of L, respectively.

(iv) choose some entries of the eigenvectors $\mathbf{y}_i$, for $i = 1, \ldots, m$ to obtain the dimensionality reduction.

By applying the above steps to the matrix $\mathbf{P} \in R^{k \times n}$, we find a new matrix $\tilde{\mathbf{P}} \in R^{k \times l}$. In particular, if we choose $l = 3$, and compute the Euclidean distance between all possible pairs of points, we end up with a variantion the classical Distance Geometry Problem, namely, we have a set of distances between points in R^3 for which we only know two of their three coordinates. Hence, some DG algorithm can be applied in order to find the complete embedding in R^3.

After applying one of the above described approach to reduce the dimensionality of the original space, some clustering algorithm, or more elaborated methods which enable taking measurement errors into account (such as [4]) can be applied to the embedded graph in order to segregate the stellar cluster from the field.

References

[1] Belkin, M., Niyogi, P. (2001) *Laplacian Eigenmaps and spectral techniques for embedding and clustering*, Advances in Neural Information Processing Systems 14, pp. 585–591, MIT Press.

[2] Cvetković, D., Rowlinson, P., Simić, S. (1997) Eigenspaces of graphs, Encyclopedia of Mathematics and its applications 66, Cambridge University Press.

[3] Friel, E.D. (1995) *The Old Open Clusters of the Milky Way*, Annual Review of Astronomy and Astrophysics, vol. 33, pp. 381-414.

[4] Krone-Martins, A., Moitinho, A. (2013) *UPMASK: Unsupervised Photometric Membership Assignment in Stellar Clusters*, submitted to Astronomy&Astrophysics.

[5] Lavor, C., Liberti, L. , Maculan, N., Mucherino, A. (2012) *Recent advances on the Discretizable Molecular Distance Geometry Problem* , European Journal of Operational Research, vol. 219, pp. 698—706.

[6] Liberti, L., Lavor, C., Mucherino, A., Maculan, N. (2010) *Molecular distance geometry methods: From continuous to discrete*, International Transactions in Operational Research 18, pp. 33-51.

[7] Merris, R. (1994) Laplacian matrices of graphs: a survey, Linear Algebra and its Applications, 197/198, 143-176.

[8] Mucherino, A., Lavor, C., Liberti, L., Maculan, N. (2013) *Distance Geometry: Theory, Methods and Applications.* Springer.

[9] Soderblom, D.R. (2010) *The Ages of Stars*, Annual Review of Astronomy and Astrophysics, vol. 48, pp. 581-629.

[10] Portegies Zwart, S.F., McMillan, S.L.W. and Gieles, M. (2010) *Young Massive Star Clusters*, Annual Review of Astronomy and Astrophysics, vol. 48, pp. 431-493.

DGA 2013, pp. 93 – 97.

The Hardness of the d-Distance Flow Coloring Problem

Manoel Campêlo[1], Cristiana G. Huiban[2] and Rudini Sampaio[1]

[1]*Universidade Federal do Ceará, Fortaleza, Brazil* {mcampelo,rudini}@lia.ufc.br

[2]*Universidade Federal de Pernambuco, Recife, Brazil,* cmngh@cin.ufpe.br

Abstract Let $G = (V, E)$ be a graph with a subset $V_s \subset V$ of *source nodes*, a *gateway* $g \in V \setminus V_s$ and a function $b : V_s \to \mathbb{N}$. A flow ϕ_u of a source node u is a multiset of $b(u)$ paths in G from u to g. A flow ϕ on G is a set with one flow for each source node. Every flow ϕ defines a multigraph G_ϕ with vertex set V and all edges in the paths on ϕ. A *d-distance edge coloring* of a flow ϕ is an edge coloring of G_ϕ such that edges with the same color are at distance at least d in G. The d-distance flow coloring problem (FCP_d) is the problem of obtaining a flow ϕ on G with a minimum d-distance edge coloring. We prove that FCP_d is NP-hard, for any fixed distance $d \geq 2$, even with just one source node on general graphs. We also study several cases of FCP_d proving their NP-hardness on bipartite graphs. Finally, we show that a list version of the problem is inapproximable by a factor of $O(\log n)$ even on paths for any distance $d \geq 1$.

Keywords: Flow coloring, distance on graphs, chromatic index, NP-hardness

1. Introduction

Flow and Coloring are two classical problems in Graph theory. Usually, simple flow problems are easy whereas coloring problems are hard. One could think of combining them in several ways. What happens, for instance, if we want to color the edges of a subgraph induced by the edges carrying flow in a network?

Before putting this question more precisely, let us say that a similar scenario has already been considered in the *Round Weighting Problem* - RWP [5]. Motivated by wireless network applications, the definition of RWP states that a flow must be sent, from sources to sink, through a network by rounds. A round is a set of links that can transmit simultaneously without interference, and therefore could share the same frequency. The distance between links is one of the parameters that are taken into account to define a round.

Recently, the idea of combining flow and coloring was formalized and the term *flow coloring* was adopted [3]. Instead of rounds, color classes are used to cover the links carrying flow. Counterparts of classical coloring parameters appear naturally. It is the case of the flow chromatic index [3]. Here, we consider more general color classes.

Let G be a simple graph with a special subset $V_s \subset V$ of vertices called *source nodes*, a special vertex $g \in V \setminus V_s$ called *gateway*, and a function $b : V_s \to \mathbb{N}$ which associates an integer demand $b(u)$ to every source node u.

A flow ϕ_u of a source node u is a multiset of $b(u)$ paths from u to the gateway g (these paths are not necessarily distinct nor disjoint). A flow ϕ on G is a set containing one flow ϕ_u for each source node u. Every flow ϕ defines a multigraph G_ϕ with vertex set V and all edges from the paths on ϕ (the number of times an edge from G appears in G_ϕ is the same as the number of paths on ϕ containing this edge).

Table 1: Our results for FCP_d

Problem	Distance	Graph type	Source	Complexity
FCP_d	$d \geq 3$	Bipartite	One source	NP-hard
FCP_d	$d = 2$	Bipartite	Multiple sources	NP-hard
FCP_d	$d = 2$	Bipartite	One source	Open
FCP_1	$d = 1$	General	Multiple sources	Open
List FCP_d	$d \geq 1$	Path	One source	NP-hard

A *d-distance edge coloring of a flow* ϕ is an edge coloring of G_ϕ such that edges with the same color are at distance at least d in the original graph G (the distance between two edges is the minimum distance between their end vertices). Let Φ stand for the set of all possible integer flows $\phi : E \to \mathbb{Z}_+$. The d-distance flow coloring problem (FCP_d) consists in obtaining a flow $\phi \in \Phi$ with a minimum d-distance edge coloring. The minimum number of used colors is called the *d-distance flow chromatic index* $\chi'_{\Phi,d}(G)$. If, for each edge, a list of possible colors is given, the d-distance list flow coloring problem ($LFCP_d$) is similarly defined.

The FCP_d as defined here is studied in [2, 4] as a variant of the RWP and, tools to obtain lower and upper bounds for general graphs were developed. A $\frac{d+1}{\lceil \frac{d+1}{2} \rceil}$-approximation algorithm for FCP_d was then presented. Exact and constructive results for grids are also obtained, in particular for the case of FCP_d with uniform demands. More recently, a polynomial time algorithm for FCP_d with $d = 1$ in any 3-connected graph and in several cases of 2-connected graphs was obtained [3] extending the results in [2, 4] related to FCP_1. However, the hardness of FCP_d, including the case $d = 1$, was still open.

Here, we address exactly the computational complexity of FCP_d. In Section 2, we prove that FCP_d is NP-hard, for any fixed distance $d \geq 2$, even with just one source node on general graphs. For $d \geq 3$, the same result is obtained even on bipartite graphs. In Section 3, we prove the NP-hardness on bipartite graphs with multiple sources and d even. In Section 4, we prove that $LFCP_d$ is inapproximable by a factor of $O(\log n)$ even on paths for any $d \geq 1$. Our results are summarized in the Table 1.

2. Hardness of FCP_d with one source node for distance $d \geq 2$

We prove that FCP_d with only one source node is NP-hard for $d \geq 2$ on general graphs and, if $d \geq 3$, on bipartite graphs. Let G be an instance of FCP_d with only one source node u and demand $b(u) = 2$. A cycle $C = v_1 \dots v_k$ is *interference free* $(d+1)$*-labeled* if k is a multiple of $d + 1$ and, the edges $v_i v_{i+1}$ and $v_j v_{j+1}$ are at distance $\geq d$ in G for $|i - j|$ multiple of $d + 1$.

Lemma 1. *Let G be a graph with only one source node u and $b(u) = 2$. For $d \geq 2$, $\chi'_{\Phi,d}(G) = d + 1$ if and only if G has an interference free $(d + 1)$-labeled cycle containing u and g.*

Let IFLC$_d$ be the problem of deciding if a graph has an *Interference free* $(d+1)$*-labeled* Cycle containing two given vertices. We prove that IFLC$_d$ is NP-hard by a reduction from 3SAT.

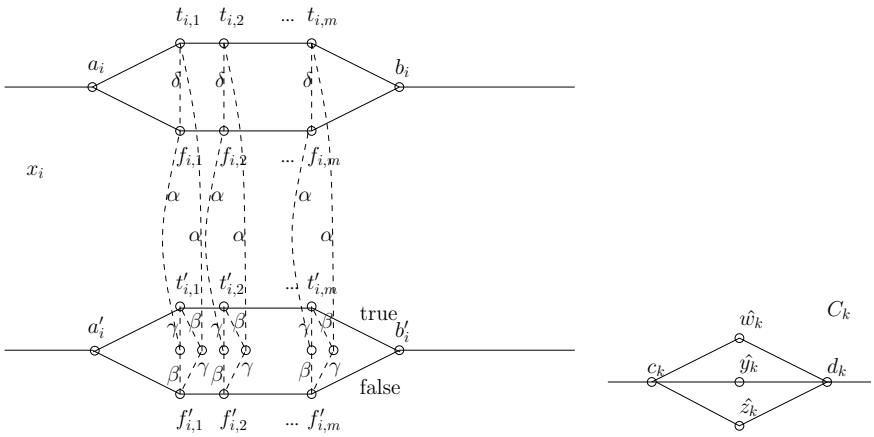
Theorem 2. *IFLC$_{d\geq 2}$ is NP-hard. Consequently, FCP$_{d\geq 2}$ with one source node is also NP-hard. If $d \geq 3$, they are NP-hard even on bipartite graphs.*

We sketch the main ideas of the proof. Given a 3SAT formula with variables $x_1, \dots, x_n$ and clauses $C_1, \dots, C_m$, create for each x_i a gadget shown in the Figure 1a, where the values $\alpha, \beta, \gamma, \delta$ (defined on Table 1) on the dashed lines are paths with $\alpha, \beta, \gamma, \delta$ edges, respectively.

It is important to notice that $\alpha + \beta = d$, $\alpha + \gamma + \delta = d$, $\beta + \gamma = d - 2$, if $d \geq 3$ is even, and $\beta + \gamma = d - 1$, if $d \geq 3$ is odd. This implies that the distance $d(t_{i,k}, t'_{i,k}) = d(f_{i,k}, f'_{i,k}) = d$, for

Table 2: The sizes of the paths in the dashed lines of Figure 1a

Path	$d = 2$	$d \geq 3$ even	$d \geq 3$ odd
α	1	$d/2$	$(d-1)/2$
β	1	$d/2$	$(d+1)/2$
γ	0	$(d-4)/2$	$(d-3)/2$
δ	1	2	2



(a) Variable gadget (b) Clause gadget

Figure 1: Gadgets for FCP_d with one source node.

every $i = 1, \ldots, n$ and $k = 1, \ldots, m$. In this reduction, the dashed lines represent paths, which we call *dashed paths* (we will prove that the cycle has no dashed path).

For each clause $C_k = (w_k \vee y_k \vee z_k)$, where w_k, y_k, z_k are literals, create 5 vertices $c_k, d_k, \widehat{w_k}, \widehat{y_k}, \widehat{z_k}$ and join them as described in the Figure 1b.

Also replace every non-dashed line in all gadgets by a path with $2(d+1)$ edges, except, for d odd, the lines $a_i t_{i,1}, a_i f_{i,1}$, which we replace by a path with $2d+1$ edges, and the lines $b_i t_{i,m}, b_i f_{i,m}$, which we replace by a path with $2d+3$ edges, for every $1 \leq i \leq n$.

Join b_i to a_{i+1} with a path of size $2(d+1)$ $(i < n)$. Join d_j to c_{j+1} with a path of size $2(d+1)$ $(j < m)$. Join b_n to c_1 with a path of size $2(d+1)$. Create special vertices u and v, join u to a_1 and a'_1 with paths of sizes $2(d+1)$ and join v to b'_n and d_m with paths of sizes $2(d+1)$.

Finally, if the literal w_k is x_i for some $1 \leq i \leq n$, then connect the vertex $\widehat{w_k}$ to $f_{i,k}$ and $f'_{i,k}$ with paths of sizes $d - \delta$ and δ, respectively (that is, with $d - \delta$ and δ edges, respectively). If w_k is $\overline{x}_i$ for some $1 \leq i \leq n$, then connect the vertex $\widehat{w_k}$ to $t_{i,k}$ and $t'_{i,k}$ with paths of sizes $d - \delta$ and δ, respectively. Analogously, we do the same for y_k and z_k, for every $1 \leq k \leq m$. Consider all paths of this paragraph as dashed (forbidden for the cycle). This finishes the reduction.

Given a 3SAT formula Γ, let (G, u, v) be an instance of $\text{IFLC}_{d \geq 2}$ obtained by the reduction above. With some effort, it is possible to prove that G is bipartite for $d \geq 3$. It is also possible to prove that Γ is satisfiable if and only if G has an *interference free* $(d+1)$*-labeled* cycle containing u and v. The main ideia is to prove that any *interference free* $(d+1)$*-labeled* cycle containing u and v cannot contain a dashed path.

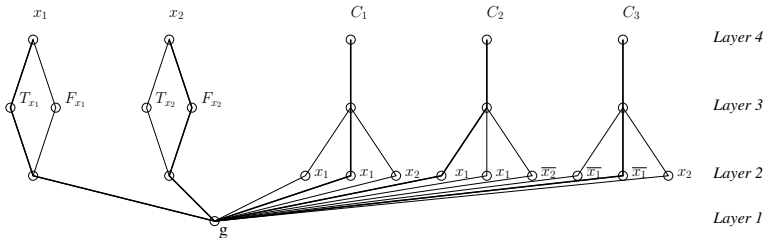


Figure 2: Reduction from 3SAT to FCP_d for $d = 2$ and multiple source nodes.

3. Hardness of FCP_d with multiple sources for distance $d \geq 2$

We show that $FCP_{d=2}$ is NP-hard on bipartite graphs by a reduction from $3SAT$ to FCP_2 inspired by [1]. Figure 2 shows an example for the formula Γ with clauses $C_1 = (x_1 \vee x_1 \vee x_2)$, $C_2 = (x_1 \vee x_1 \vee \overline{x_2})$ and $C_3 = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2})$. A truth assignment of Γ appears in bold.

In general, given a $3SAT$ instance Γ with n variables and m clauses, we construct a bipartite graph G with 4 layers of vertices. In layer 4 (the top layer), create $n + m$ vertices associated with the variables and clauses of Γ. In layer 3, create vertices F_x and T_x (representing *False* and *True*) for each variable x and connect them to the vertex x on layer 4. For each clause, create a vertex and connect it to the associated vertex on layer 4. In layer 2, create a vertex for each variable x and connect it to F_x and T_x on layer 3. For each clause $C = (z_1 \vee z_2 \vee z_3)$, create 3 new vertices z_1, z_2 and z_3 and connect them to the associated clause in layer 3. For $i \in \{1, 2, 3\}$, connect each literal $z_i = x$ (resp. $z_i = \overline{x}$) to the vertex F_x (resp. T_x) on layer 3. In layer 1, create the gateway g and connect it to every vertex on layer 2. Every vertice on layer 4 is a source node u with demand $b(u) = 1$.

Theorem 3. *If Γ is satisfiable, then $\chi'_{\Phi,2}(G) = n + m + 1$. Otherwise, $\chi'_{\Phi,2}(G) = n + m + 2$. Consequently, $FCP_{d=2}$ is NP-hard on bipartite graphs.*

4. Inapproximability of $LFCP_{d\geq 1}$ in path graphs

We prove that LFCP$_d$ is inapproximable by a factor of $O(\log n)$, with an approximation preserving reduction from the Set Cover Problem (SCP). Given a set $S = \{s_1, \ldots, s_n\}$ and a family $\mathcal{F}$ with m subsets of S, the objective of SCP is to obtain a minimum number of subsets in $\mathcal{F}$ that cover S (that is, their union is S). Raz and Safra [6] proved that SCP is $O(\log n)$-inapproximable. This holds even for instances where $|\mathcal{F}| \leq |S|$.

Given an instance $(S, \mathcal{F})$ of SCP, we construct a graph G, which will be a path. For every element $s_i \in S$, create two vertices x_i and y_i, connect them with an edge and set the list $L(x_i y_i)$ of the possible colors of the edge $x_i y_i$ to be all subset in $\mathcal{F}$ that contains s_i. For every $i < n$, connect y_i and x_{i+1} with a path of length d, where the list of possible colors of the k-th edge of each path has only the color k ($1 \leq k \leq d$). Graph G has only one source node, that is x_1 with demand $b(x_1) = 1$, and the gateway is the vertex y_n.

Given a solution (a flow coloring) for LFCP$_d$ in G, it is possible to prove that the colors used in the edges $x_1 y_1, \ldots, x_n y_n$ form a set cover of S. The number of colors in G is the size of the set cover of S plus d. Therefore, the size of the minimum set cover is equal to $\chi'_{\Phi,d}(G) - d$. Since d is constant, it is an AP-reduction, which, together with [6] implies the following:

Theorem 4. *If* P $\neq$ NP, *$LFCP_{d\geq 1}$ is $O(\log n)$-inapproximable even on paths.*

References

[1] J-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Perennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2):165–183, June 2006.

[2] J-C. Bermond, C. G. Huiban, and P. Reyes. Round Weighting Problem and gathering in wireless networks with symmetrical interference. Rapport de recherche hal inria-00408502, INRIA/UNSA, 2009.

[3] M. Campelo, R. Correa, C. G. Huiban, and D. Rodrigues. The flow coloring problem. In *Congreso Latino-Iberoamericano de Investigacion Operativa (CLAIO, SBPO)*, RJ, Brazil, September 2012.

[4] C. G. Huiban. *Radio Mesh Networks and the Round Weighting Problem.* PhD thesis, Université de Nice-Sophia Antipolis (UNS), Sophia Antipolis, France, December 2009.

[5] R. Klasing, N. Morales, and S. Pérennes. On the complexity of bandwidth allocation in radio networks. *Theoretical Computer Science*, 406(3):225–239, October 2008.

[6] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 475–484, New York, NY, USA, 1997. ACM.

DGA 2013, pp. 99 – 102.

On the Discretization of iDMDGP instances regarding Protein Side Chains with rings *

Virginia Costa[1], Antonio Mucherino[2], Luiz Mariano Carvalho[3] and Nelson Maculan[1]

[1]*COPPE, Federal University of Rio de Janeiro, Rio de Janeiro-RJ, Brazil,* {virscosta, maculan}@cos.ufrj.br

[2]*IRISA, University of Rennes 1, Rennes, France,* antonio.mucherino@irisa.fr

[3]*IME, State University of Rio de Janeiro, Rio de Janeiro-RJ, Brazil,* luizmc@ime.uerj.br

Abstract The *interval* Discretizable Molecular Distance Geometry (iDMDGP) consists in a subclass of distance geometry problems that can be discretized. Instances of the iDMDGP can be solved by employing an efficient *interval* Branch & Prune (iBP) algorithm. However, instances can belong to the iDMDGP class only if some particular assumptions are satisfied, that are mainly based on the order on which the atoms of the molecule are considered. In this short paper, we present 5 special orders for the side chains of 5 amino acids, the ones that contain rings in their structure.

Keywords: protein conformations, distance geometry, combinatorial optimization, Branch & Prune, side chains.

1. Introduction

We consider the *interval* Discretizable Molecular Distance Geometry Problem (iDMDGP) [3], which is the subclass of distance geometry problems where the distance information can be represented by suitable intervals and a discretization of the search space can be performed. We are particularly interested in problems arising in biology, and therefore our instances represent molecules, and specifically proteins, in the Euclidean three-dimensional space.

An instance of the iDMDGP can be represented by a weighted undirected graph $G = (V, E, d)$ where each vertex $v \in V$ represents an atom of a given molecule and each edge $(u, v) \in E$ represents the known distance between the vertices (atoms) u and v. The weight $d(u, v)$ associated to an edge (u, v) can correspond either to a precise distance or to a suitable interval where the actual distance is supposed to be contained. Supposing that there exists a total order relationship for the vertices of V, we consider iDMDGP instances that satisfy the following assumptions:

1. $(1, 2, 3) \subset V$ is a clique and all distances are precise,
2. $\forall v \in \{3, \ldots, n\}$, $(i-2, i)$ and $(i-1, i)$ correspond to precise distances,
3. $\forall v \in \{4, \ldots, n\}$, $\forall j, k \in \{v-3, \ldots, v\}$, $(j, k) \in E$,
4. $\forall v \in \{2, \ldots, n-1\}$, $d(v-1, v+1) < d(v-1, v) + d(v, v+1)$.

*The authors wish to thank CAPES, that funded a 4-month visit to Rennes for Virginia Costa (part of this work was performed during such a visit). We are also thankful to the French Embassy in São Paulo and to UNICAMP, which funded a 2-month visit (chaire) to UNICAMP for Antonio Mucherino. Finally, a special thank to Prof. Carlile Lavor for his fruitful comments.

We remark that only precise distances are concerned in the strict triangular inequalities. These assumptions allow to compute the possible positions for the generic atom v as the intersection among three Euclidean objects, which are related to the three immediate preceding atoms $v-3$, $v-2$ and $v-1$. Each Euclidean object can be either a sphere (when its radius is precise) or a spherical shell (when it is represented by an interval). The intersection among three spheres consists of, with probability one, two points in the three-dimensional space [2]. If one of the distances is represented by an interval, one of the spheres to be intersected is replaced by a spherical shell, so that the intersection generally consists of two disjoint curves.

The *interval* Branch & Prune (iBP) algorithm is based on the idea of building the search domain (a tree) recursively, and to verify the feasibility of its branches "on the fly", in order to prune the infeasible branches as soon as possible. In order to apply the algorithm, an order on the vertices of G must be available such that the above assumptions are satisfied. More details about iBP and the orders that allow for discretization can be found in [1].

As it is well known, proteins are chains of amino acids that fold in unique conformations, that imply a certain function for the molecule. Only 20 amino acids can be involved in the protein synthesis, and each of them has a different *side chain*. In a previous work [4], we proposed special orders that allow for the discretization to the 8 smallest side chains that can be part of an amino acid. In this short paper, we consider other 5 side chains, the ones that contain rings, i.e. local rigid conformations formed by 5 or 6 Carbon atoms that are bonded in a way to form this particular structure.

The rest of this paper is organized as follows. In Section 2 we remind the definition of *repetition order* (re-order) and present the 5 re-orders for the 5 considered side chains. Section 3 presents some computational experiments, while Section 4 concludes the paper.

2. Orders for side chains with rings

Let us consider that the set of edges E of G can be partitioned into those edges $\{u,v\} \in E'$ for which $d(u,v)$ is a real nonnegative number, and those edges $\{u,v\} \in E''$ for which $d(u,v)$ is a finite set of points belonging to a positive rational interval. Let $V' = V \cup \{0\}$. A *repetition order* (*re-order*) is a sequence $r : \mathbb{N} \to V'$ with length $|r| \in \mathbb{N}$ (for which $r_i = 0$ for all $i > |r|$) such that:

- $G[\{r_1, r_2, r_3\}]$ is a clique
- for all $i \in \{4, \ldots, |r|\}$ the sets $\{r_{i-2}, r_i\}$, $\{r_{i-1}, r_i\}$ are edges in E'
- for all $i \in \{4, \ldots, |r|\}$ the set $\{r_{i-3}, r_i\}$ is either a singleton (i.e. $r_{i-3} = r_i$) or an edge in $E' \cup E''$.

It is possible to easily verify that any re-order r represents an iDMDGP instance.

Fig. 1 shows the 5 re-orders that we hand-crafted for the 5 side chains containing rings. As in the previously proposed orders [1, 4], for artificially adding precise distances in our instances (distances between two copies of the same atom), Carbon atoms can be considered more than once in the orderings. Side chains with rings are generally larger than the others: the smallest we consider is the proline, where the re-order is composed by 18 vertices, while the largest is the tryptophan, whose re-order contains 40 vertices.

3. Computational Experiments

In this section, we present some computational experiments on iDMDGP instances, which were randomly generated considering the 5 side chains shown in Fig. 1. We suppose that all the covalent bond lengths are equal to 1.33 Å and the angles between two covalent bonds are equal to 110°. All codes were written in C programming language and all the experiments were

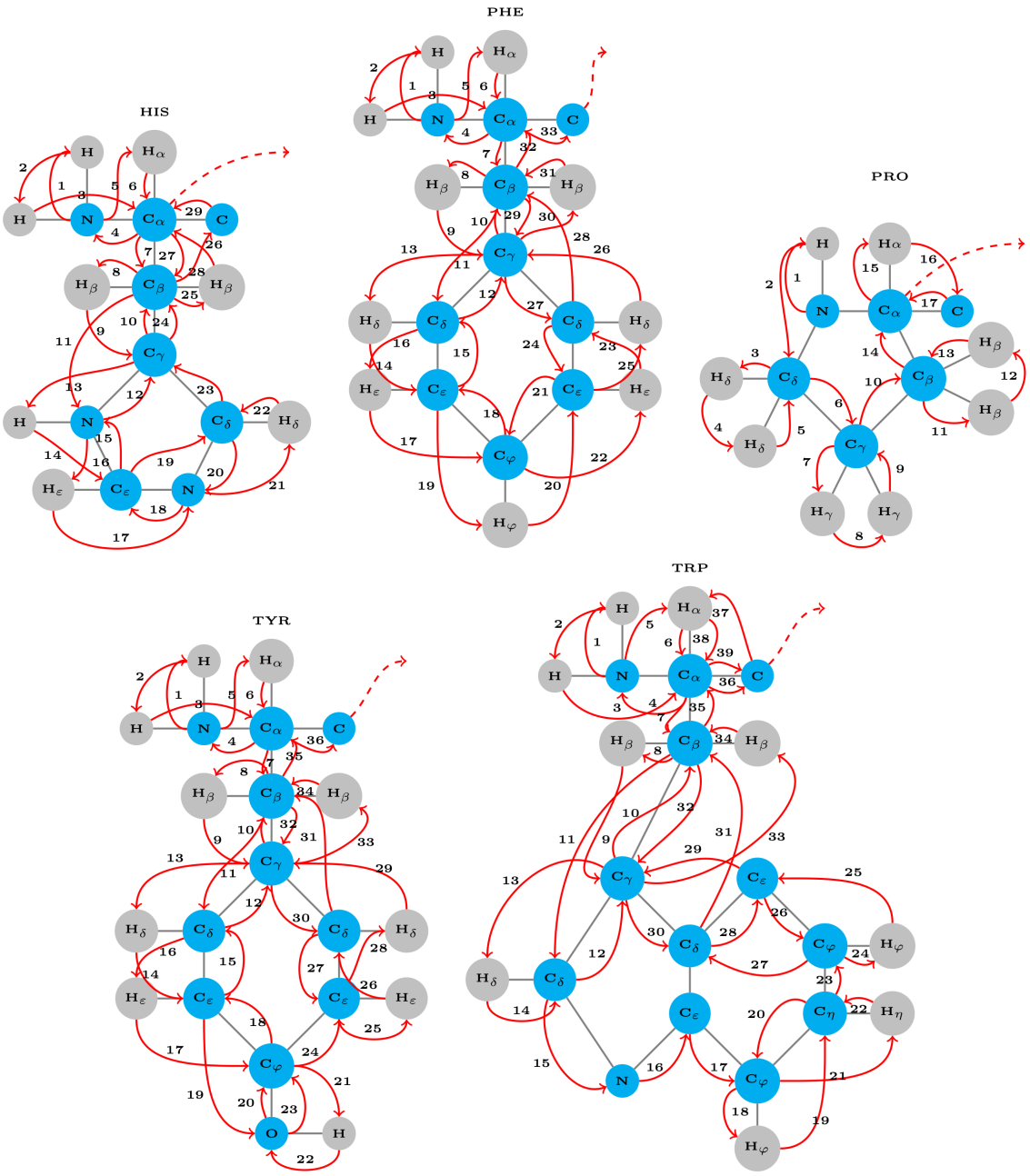


Figure 1: Hand-crafted orders for the side chains with rings: Histidine (HIS), Phenylalanine (PHE), Proline (PRO), Tryptophan (TRP) and Tyrosine (TYR).

Amino Acid Sequences	Number of Vertices	min(D)	CPU time
PRO-PHE-HIS-TRP	123	9	174.84
PHE-PRO-TYR-HIS	123	8	252.07
HIS-TYR-PRO-TRP	128	11	5.12
PRO-PHE-PRO-HIS	104	8	0.57
TYR-TYR-HIS-PRO	126	8	44.74
TRP-TRP-PHE-PRO	132	8	58.76

Table 1: Some computational experiments on the proposed orders for the 5 side chains

carried out on an Intel Core i7 2.30GHz with 8B RAM, running Linux. The codes have been compiled by the GNU C compiler v.4.7.2.

Table 1 shows some experiments on some small instances. In this table, we provide the total number of vertices forming our instances, the minimum number D of points selected from each interval $(i-3, i)$ for obtaining at least one solution [1], and the CPU time (in seconds) necessary for finding this solution. We can observe that the presented orders allow to discretize the considered instances, and that the CPU time ranges from about 1 second to about 4 minutes. It is important to observe that the *i*BP algorithm can return more than one solution and the number of found solutions, as well as the CPU time, is related to the distances between hydrogens and to the pruning techniques. The first can be supplied by NMR experiments, while the second one can be developed and improved using, for example, information about the protein structure.

4. Conclusions

We provided 5 new special orders for 5 side chains containing rings. These orders allow to discretize instances containing this kind of side chains. Only information about the chemical structure of the amino acids is exploited for the conception of such orders, while NMR distances are supposed to be used only for pruning purposes in the *i*BP algorithm. Preliminary computational experiments showed the effectivity of the proposed orders.

References

[1] C. Lavor, L. Liberti, A. Mucherino, *The interval Branch-and-Prune Algorithm for the Discretizable Molecular Distance Geometry Problem with Inexact Distances*, to appear in Journal of Global Optimization, 2013.

[2] C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, Computational Optimization and Applications **52**, 115–146, 2012.

[3] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, to appear on SIAM Review, 2013.

[4] V. Costa, A. Mucherino, C. Lavor, L.M. Carvalho, N. Maculan, *On Suitable Orders for Discretizing Molecular Distance Geometry Problems related to Protein Side Chains*, IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS12), Workshop on Computational Optimization (WCO12), Wroclaw, Poland, 397–402, 2012.

DGA 2013, pp. 103 – 107.

The monophonic convexity in bipartite graphs

Eurinardo R. Costa[1], Mitre C. Dourado[2] and Rudini M. Sampaio[1]

[1]*Universidade Federal do Ceará, Fortaleza, Brazil* {eurinardo,rudini}@lia.ufc.br

[2]*Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil,* mitre@nce.ufrj.br

Abstract We investigate several parameters of the monophonic convexity on graphs. In 2010, it was proved that the m-interval number and the m-convexity number are NP-hard on general graphs [4]. In this paper, we prove that deciding if the m-interval number is at most 4 and deciding if the m-percolation time is at most 1 are NP-Complete problems in bipartite graphs. We also prove that the m-Radon number and the m-convexity number are as hard to approximate as the maximum clique problem. Finally, we present a polynomial time algorithm to determine the m-convexity number on graphs with bounded clique number (as planar graphs and bipartite graphs).

Keywords: Monophonic convexity, bipartite graphs, NP-completeness, inapproximability

1. Introduction

Consider the following problem motivated by Distance Geometry applications: given a set U of points in the plane, obtain the minimum subset $S \subseteq U$ such that every point of U lies in the line segment between two points of S.

We can view the points of S as *infected*. Given two infected points x and y, we say that v is infected by x and y if v is in the line segment between them. We can also ask what is the minimum subset $S \subseteq U$ such that all vertex of U is infected after a finite number of sucessive infections. We can also ask what is the maximum proper subset $S \subset U$ which does not infect a vertex.

In this paper, we investigate these questions in a different structure. The points are vertices of a graph and the line segment between two vertices are the induced paths between them.

Such problems are intensively studied by the Theory of Convexity Spaces, which form a classical topic, studied in some different branches of mathematics. The study of convexities applied to graphs has started later, about 50 years ago. Then the convexity parameters motivated the definition of some graph parameters, whose study has been one of the central issues in graph convexities. In particular, complexity aspects related to the computation of these parameters has been the main goal of various recent papers.

Let G be a simple finite graph, with vertex set $V(G)$ and $\mathcal{C}$ a family of subsets of $V(G)$. The pair $(G, \mathcal{C})$ is a *graph convexity* when $\emptyset \in \mathcal{C}$, $V(G) \in \mathcal{C}$ and, if $S_1, S_2 \in \mathcal{C}$, then $S_1 \cap S_2 \in \mathcal{C}$. The subsets $C \in \mathcal{C}$ are called *convex sets*. The *convex hull* of a subset $S \subset V(G)$, denoted by $hull(S)$, is the minimum convex set which contains S. If $hull(S) = V(G)$, we say that S is a *hull set*.

Next, we describe some graph parameters related to a graph convexity. The *hull number* $hn(G)$ of G is the size of a minimum hull set. The *interval number* $in(G)$ is the size of the minimum subset $S \subseteq V(G)$ such that S is contained in no convex set, except $V(G)$. The

convexity number $cx(G)$ is the size of the maximum convex set distinct from $V(G)$. The *Radon number* $rd(G)$ is the minimum k such that every subset V' of $V(G)$ of size at least k has a Radon partition, which is a partition (V_1', V_2') such that $hull(V_1') \cap hull(V_2') \neq \emptyset$. Alternatively, $rd(G)$ is the size of a maximum anti-Radon set plus one, where a set is anti-Radon if it has no Radon partition.

Clearly, the computation of these parameters for a graph would depend on the particular convexity being considered. Among the existing convexities we can mention the following, whose convex sets are based on paths of the graph: monophonic, geodesic and P_3. They are defined by letting the convex sets be closed, respectively, under induced paths, shortest paths and paths of order 3.

Let the m-interval number be the interval number on the monophonic convexity. Analogously, we define the same for the other parameters.

In 2010, it was proved that the m-interval number and the m-convexity number are NP-hard on general graphs [4]. Interestingly, they obtained a polynomial time algorithm to compute the m-hull number of a graph.

In this paper, we extend some of these results. We prove that deciding if the m-interval number is at most 4 is NP-Complete in bipartite graphs. We also prove that the m-Radon number is as hard to approximate as the maximum clique problem. Finally, we present a polynomial time algorithm to determine the m-convexity number on graphs with bounded clique number (as planar graphs and bipartite graphs).

2. The interval number of the monophonic convexity

In 2010, it was proved the following theorem [5], which is very useful in this section.

Theorem 1 ([5])**.** *Given a bipartite graph G and three distinct vertices x, y, z, deciding whether there is an induced path from x to y passing through z is NP-complete.*

Given a subset $S \subseteq V(G)$, we define the *monophonic interval* $I(S)$ as the set with the vertices in S and all vertices in an induced path between two vertices of S. If $I(S) = V(G)$, we say that S is a monophonic set of G. The following corollary is a direct consequence of the Theorem 1.

Corollary 2. *Given a connected bipartite graph G and three distinct vertices x, y, z, deciding whether $z \in I(\{x, y\})$ is NP-complete.*

From the above corollary, the problem of determining the monophonic interval of a set X is NP-hard, even if X has only two elements and the graph is bipartite.

The following theorem proves that deciding if a set S of vertices is a monophonic set is NP-Complete, even if the graph is bipartite and S has at most 4 elements.

Theorem 3. *Given a connected bipartite graph G and a set S with at most 4 vertices of G, deciding whether S is a monophonic set is NP-complete.*

Sketch of the proof. A certificate that this problem belongs to NP is a set of at most $|V(G)|-|S|$ induced paths, each one beginning and finishing in distinct vertices of S, such that every vertex of $V(G) \setminus S$ appears in at least one of these paths.

We describe a reduction from the decision problem given in Corollary 2. Let H be a bipartite graph with bipartition (A, B) and let x, y, z be three distinct vertices of H. Without loss of generality, suppose that $z \in B$. Define a bipartite graph G by adding to H six new vertices $a_1, a_2, b_1, b_2, c_1, c_2$ such that a_1 and a_2 are adjacent to all vertices in $B \setminus \{z\}$, b_1 and b_2 are adjacent to all vertices in A. Also include the edges $a_1b_1, b_1c_1, a_2b_2, b_2c_2$. Clearly G is bipartite with bipartition $(A \cup \{a_1, a_2, c_1, c_2\}, B \cup \{b_1, b_2\})$. Set $S = \{x, y, c_1, c_2\}$.

We have to show that $z \in I(\{x, y\})$ in H if and only if S is a monophonic set of G. Notice that $A \subseteq I(\{c_1, c_2\})$, since, for every vertex $v \in A$, there is the induced path $c_1b_1vb_2c_2$. Also notice that $B \setminus \{z\} \subseteq I(\{c_1, c_2\})$, since, for every vertex $v \in B \setminus \{z\}$, there is the induced path $c_1b_1a_1va_2b_2c_2$. Thus $I(\{c_1, c_2\}) = V(G) \setminus \{z\}$, since every induced path containing z must have two neighbors of z, which are in A and are adjacent to b_1 and b_2. Finally, notice that there is no induced path containing z between a vertex in $\{x, y\}$ and a vertex in $\{c_1, c_2\}$, since every induced path containing c_1 must contain b_1, which is adjacent to all neighbors of z.

Assume that $z \in I(\{x, y\})$ in H. Since every induced path of H is also an induced path of G, then $z \in I(\{x, y\})$ in G. Consequently $I(\{x, y, c_1, c_2\}) = V(G)$.

Now assume that $I(\{x, y, c_1, c_2\}) = V(G)$. Since $I(\{c_1, c_2\}) = V(G) \setminus \{z\}$, then $z \in I(\{x, y\})$, since, as already mentioned, z is not in any induced path between a vertex in $\{x, y, c_1, c_2\}$ and a vertex in $\{c_1, c_2\}$. □

Finally, we prove that deciding if $in(G) \leq 4$ is NP-Complete, even if G is bipartite.

Theorem 4. *Given a bipartite graph G, deciding whether $in(G) \leq 4$ is NP-complete.*

Sketch of the proof. A certificate that this problem belongs to NP is a set S with at most 4 vertices and a set of at most $|V(G)| - |S|$ induced paths between two vertices of S, such that every vertex of $V(G) \setminus S$ belongs to at least one of these paths.

We now show a reduction from the decision problem of Theorem 2: deciding whether a given subset S is a monophonic set of a connected bipartite graph H with at most 4 vertices (we assume that H has at least two vertices). Let $S = \{x_1, \ldots, x_k\}$, where $2 \leq k = |S| \leq 4$, be a subset of vertices of a bipartite graph H.

Define a bipartite graph G by adding to H a set $S' = \{x'_1, \ldots, x'_k\}$ of k new vertices and k new edges $x_1x'_1, \ldots, x_kx'_k$. We have to prove that S is a monophonic set of H if and only if $m(G) \leq k$.

At first, suppose that S is a monophonic set of H. We claim that S' is a monophonic set of G and then $m(G) \leq k$. For this, let $z \in V(G) \setminus S$. Since $z \in I(S)$ in H, then z is in an induced path between two vertices x_i and x_j in H. By the construction, z belongs to an induced path between x'_i and x'_j in G. Now let $z = x_i \in S$. Let P be a minimum path in G between z and a vertex in $x_j \in S \setminus \{x_i\}$. Since G is connected, P exists and is induced. Then $z = x_i$ is in the induced path $x'_ix_iPx_jx'_j$.

Now suppose that $m(G) \leq k$. Since S' has k vertices of degree 1, then S' is the only monophonic set of G with k vertices. Consequently every vertex of G belongs to an induced path between two vertices in S'. This implies directly that every vertex of H belongs to an induced path between two vertices in S. □

3. The percolation time of the monophonic convexity

Given a graph G and a set S of vertices of G, let $t(S)$ be the minimum k such that $I^k(S) = I^{k+1}(S)$, where I^k is the k-th iterate of the function $I(\cdot)$. The m-percolation time $t(G)$ is the maximum $t(S)$ among all hull sets S in the monophonic convexity.

Regarding the percolation time of the P_3-convexity, it was proved that it is polynomial time solvable in grids [1, 2], it is polynomial time solvable to decide if it is at most 2 [3], but it is NP-Complete to decide if it is at most 4 [3]. The question about the percolation time 3 in the P_3-convexity is still open.

With some arguments similar to the ones in the proof of Theorem 4, we obtain the following.

Theorem 5. *Given a bipartite graph G, it is NP-hard to decide if the m-percolation time of the monophonic convexity is at most 1.*

4. The Radon number of the monophonic convexity

Given a maximization problem P, let $opt_P(I)$ denote the optimal solution value for some instance I of P and, for a solution S of I, let $val_P(I,S)$ denote the associated value.

Given two optimization problems P and Q, we say that P is L-reducible to Q ($P \leq_L Q$) (or that there is an L-reduction from P to Q) if there is a triple (f,g,α,β), where $\alpha,\beta \geq 1$, f and g are polynomial time computable functions such that f maps P-instances into Q-instances,

- given a P-instance I and a feasible solution S of $f(I)$, $g(I,S)$ is a feasible solution of I,
- $opt_Q(f(I)) \leq \alpha \cdot opt_P(I)$, and
- $\Big|opt_P(I) - val(I,g(I,S))\Big| \;\leq\; \beta \cdot \Big|opt_Q(f(I)) - val(f(I),S)\Big|$.

From this definition, it follows that the relative errors are linearly related:

$$\frac{|opt_P(I) - val_P(I,g(I,S))|}{opt_P(I)} \leq \alpha\beta \frac{|opt_Q(f(I)) - val_Q(f(I),S)|}{opt_Q(f(I))}.$$

Hence, the existence of a $\left(\frac{1}{1-\varepsilon}\right)$-approximation algorithm for Q implies the existence of a $\left(\frac{1}{1-\alpha\beta\varepsilon}\right)$-approximation algorithm for P.

Let *Clique* be the problem of compute $\omega(G)$: the size of a maximum complete subgraph of a given graph G.

Theorem 6. *The m-Radon number is NP-hard and there is an L-reduction from the clique number to the m-Radon number. Consequently, for every $\varepsilon > 0$, approximating the m-Radon number to within a factor $n^{1-\varepsilon}$ is NP-hard.*

Sketch of the proof. We prove that *Clique* $\leq_L$ *AntiRadon*, where *AntiRadon* is the maximization problem of return the size of a maximum anti-Radon set of a given graph plus one.

Let a graph G be an input instance of *Clique*. Let $G' = f(G)$ be the graph such that $V(G') = V(G) \cup \{x,y\}$, where x and y are new vertices, and $E(G') = E(G) \cup \{vx, vy : v \in V(G)\}$.

Given a feasible solution R of G' (that is, R is an anti-Radon set of G'), let $C = g(G,R) = R \setminus \{x,y\}$. Notice that R has no pair of non-adjacent vertices. Otherwise, if R has two non-adjacent vertices $\{u,w\}$, then the partition $(\{u,w\}, R \setminus \{u,w\})$ of R is a Radon partition, since $x,y \in hull(\{u,w\})$ and $V(G') \subseteq hull(\{x,y\})$. Consequently, R is a clique of G' and we can assume that R contains either x or y. Thus C is a clique of G. Moreover, $|C| = |R| - 1$. Recall that $val_{AntiRadon}(G,R) = |R| + 1$.

Furthermore, since every clique of G is an anti-Radon set of G', this implies that $\omega(G) \leq rd(G') - 2 \leq 2rd(G')$. Moreover, $\omega(G) - |C| = (rd(G') - 2) - (|R| - 1) = rd(G') - (|R| + 1)$. This proves that $(f,g,2,1)$ is an L-reduction.

In 2006, it was proved that, for every $\varepsilon > 0$, approximating the clique number to within a factor $n^{1-\varepsilon}$ is NP-hard [6]. Then, this is also true for the m-Radon number. □

5. The convexity number of the monophonic convexity

In [4], it was proved that the m-convexity number is NP-Complete. The same idea of the proof of Theorem 6 can be used to prove a stronger statement.

Theorem 7. *The m-convexity number is as hard to approximate as the maximum clique problem.*

In [4], it was obtained a polynomial time algorithm to compute the m-hull number of a graph. This algorithm applies a decomposition based on clique cutsets.

In our paper, we can use the main ideas of this algorithm to obtain a polynomial time algorithm to determine the m-convexity number on graphs with bounded clique number (as planar graphs and bipartite graphs). Roughly speaking, if G has no clique cutset, then $cx(G) = 1$. Otherwise, for every clique cutset C, let H_C be the smallest component of $G - C$. Then $cx(G)$ is close to $n - \min_C |H_C|$.

References

[1] Fabrício Benevides, Michal Przykucki. *On slowly percolating sets of minimal size in bootstrap percolation*, submited, 2012.

[2] Fabrício Benevides, Michal Przykucki. *Maximum percolation time in two-dimensional bootstrap percolation*, submitted, 2012.

[3] Fabrício Benevides, Victor Campos, Mitre Dourado, Rudini Sampaio, Ana Silva. *The percolation time of the P_3-convexity*. Submitted (2013).

[4] Mitre C. Dourado, Fábio Protti, Jayme L. Szwarcfiter. *Complexity results related to monophonic convexity.* Discrete Applied Mathematics **158** (12), 2010, 1268–1274.

[5] Mauro Mezzini. *On the complexity of finding chordless paths in bipartite graphs and some interval operators in graphs and hypergraphs.* Theoretical Computer Science **411** (7-9), 2010, 1212–1220.

[6] David Zuckerman. *Linear degree extractors and the inapproximability of max clique and chromatic number.* Proceedings of STOC'2006 (ACM symposium on Theory of computing), 2006, 681–690.

DGA 2013, pp. 109 – 114.

On graph coloring problems with distance constraints *

Bruno Dias[1], Rosiane de Freitas[1] and Jayme Szwarcfiter[2]

[1]*Institute of Computing, Federal University of Amazonas, Manaus - AM, Brazil,* {bruno.dias,rosiane}@icomp.ufam.edu.br

[2]*NCE, IM and COPPE, Federal University of Rio de Janeiro, Rio de Janeiro - RJ, Brazil,* jayme@nce.ufrj.br

Abstract Graph coloring composes a large and important class of combinatorial optimization problems, and has been extensively studied in the literature. One of its key applications is in the planning of resource allocation in mobile wireless networks, for which some models have been proposed, where the coloring should to respect certain geography and technological distance constraints. In this work, we show some coloring problems as the positioning of the vertices on the integer line ($\mathbb{Z}^+$), where the point where the vertex is placed equals to its color, according to the distances between adjacent vertices, and propose a branch-prune-and-bound algorithm for solving them. An empirical analysis was made considering equality and inequality distance contraints.

Keywords: Algorithms, combinatorial optimization, graph theory, telecommunications.

1. Introduction

Let $G = (V, E)$ be an undirected graph. A k-coloring of G is an assignment of colors $1, 2, \ldots, k$ to the vertices of G so that no two adjacent vertices share the same color [2]. The chromatic number χ_G of a graph is the minimum value of k for which G is k-colorable. The classic graph coloring problem (CP), which consists in finding the chromatic number of a graph, is one of the most important combinatorial optimization problems and it is known to be NP-hard [4].

There are several versions of this classic vertex coloring problem [13], involving additional constraints, in both edges as vertices of the graph, with a number of practical applications as well as theoretical challenges. One of the main applications of such problems involves the assignment of channels to transmitters in a mobile wireless network [12]. Each transmitter is responsible for the calls made in the area which it covers and the communication among devices is made through a channel consisting of a discrete slice of the electromagnetic spectrum. However, the channels cannot be assigned to calls in an arbitrary way, since there is the problem of interference among devices located near each other using approximate channels. There are three main types of interferences: co-channel, among calls of two transmitters using the same channels; adjacent channel, among calls of two transmitters using adjacent channels and co-site, among calls on the same cell that do not respect a minimal separation. It is necessary to assign channels to the calls such that interference is avoided and the usage of the spectrum is minimized [1, 6, 7].

The separation among channels is a type of distance constraint, so we can see the channel assignment as a type of geometry distance problem, since we have to place the channels in

*This work was supported by CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior)* and CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) - Brazil.

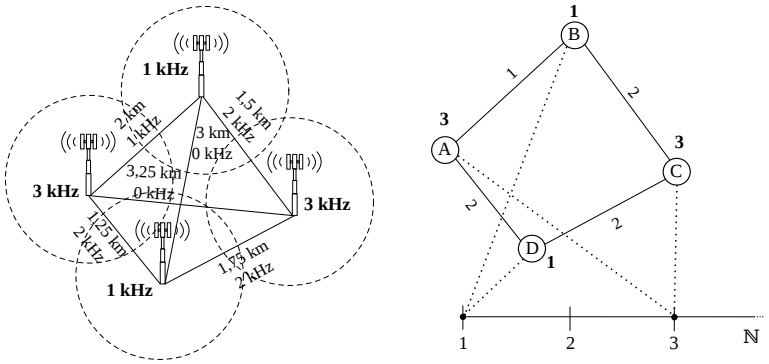


Figure 1: Example of channel assignment with distance constraints. In the left image, between two transmitters, if their geographical distance is between 0 and 2 km, the channels assigned to them must be apart at least 2 other channels. If the distance is between 2 and 3 km, the channels must be different and if the distance is greater than or equal 3 km, the transmitters can use the same channel. The distances (geographical and channel separation) are given in each edge. The bold number next to each transmitter is the channel assigned to it. The right image shows the network as an undirected graph and the projection of vertices in the natural number line.

the transmitters respecting some distances imposed in the edges, as can be seen on Figure 1. One method to solve GD problems is the branch-and-prune approach [8, 9], where a solution is constructed and, if at some point a distance constraint is violated, then we stop the building of the current solution (prune) and try another option in the search space.

The remainder of this paper is organized as follows. Section 2 states theoretical models for some channel assignment problems. Section 3 gives some coloring models stated as distance geometry problems. Section 4 formulates a branch-prune-and-bound (B&P&B) algorithm for the problems. Section 5 shows results of some experiments done with the B&P&B algorithm. Finally, Section 6 concludes the paper and states the next steps of ongoing research.

2. Preliminaries

In [5] various theoretical models for channel assignment problems are given. Some of these models are defined below.

In some instances of channel assignment problems,we have both frequency and distance constraints, that is, the channels attributed to calls must respect separation constraints according to channel proximity and geographic localization of the cells. We can formally define such scenarios as the following problem.

Definition 1. ***Frequency constrained minimum span assignment problem (F-CAP):*** *Let V be a set of labels representing the radio transmitters, and T a constraint matrix, where each element $T(u, v)$ consists of a set of positive integers where $u, v \in V$, $T(u, v) = T(v, u)$ and $T(u, u) = \varnothing$. A feasible channel assignment for this scenario is a vector a where $a(v)$ is the channel assigned to the point $v \in V$ and, for a pair of distinct points u and v, we have that $|a(u) - a(v)| \notin T(u, v)$. The span s of this assignment is equivalent to the maximum used channel ($s = \max_{v \in V} a(v)$). The problem consists of finding a feasible assignment a whose span is the minimum possible.*

F-CAP can be modelled as a graph coloring problem, as we show in following.

Definition 2. ***Generalized graph coloring problem (GCP):*** *Let $G = (V, E)$ be an undirected graph. For each edge $uv \in E$, there is a set $t(uv)$ of values (where $T = \bigcup_{uv \in E} t(uv)$) called the edge constraint $t(uv)$ of values (if there is no constraint associated to the edge, then $t(uv) = \emptyset$). A feasible coloring for G and T consists of a set a, where $a(v)$ is the color assigned to v and, for each edge uv, the condition $|a(u) - a(v)| \notin t(uv)$ holds. The objective is to find a feasible assignment whose coloring span is the minimum possible.*

It's possible to add some more constraints to the graph colorings to model some other scenarios. For example, in multicoloring, each vertex u in the graph has a weight c_u which is the number of different colors - in this situation, we have now that $t(uu) \neq \emptyset$, since there is a separation among multiple colors in the same vertex. Each element $a(u)$ of the assignment will be a set of c_u integers. In list coloring, each vertex u has a set $l(u)$ of colors and the color assigned to u must be in the set, that is, $a(u) \in l(u)$.

3. Graph colorings as distance geometry problems

An important case of the GCP occurs when the set $T(uv)$ of forbidden distances is composed of contiguous integers. Based on the definition of the Molecular Distance Geometry Problem given in [8], we can define the following problem.

Definition 3. ***Coloring Distance Geometry Problem (GCDGP):*** *Given a simple weighted undirected graph $G = (V, E, d)$, with $d : E \to \mathbb{Z}^+$, find an embedding $a : V \to \mathbb{N}$ such that $|a(u) - a(v)| = d_{uv}$ for each $uv \in E$ and $\max_{v \in V} a(v)$ is the minimum possible.*

When, for all $uv \in E$, $d_{uv} = 1$, we have the classic graph coloring problem. Also, coloring a complete graph can be done in linear time, since each vertex will have a different color. The distance geometry (DG) problem with a complete graph where all distances are known can also be solved in linear time [3]. The equality constraint of CDGP is applied in the context of channel assignment, when two transmitters are communicating between each other - one channel will be used for the downlink and the other for the uplink [11]. When the communication is only one-way, these distances are, instead of a value which of the difference among colors of adjacent vertices must be equal, a lower bound for that difference [10], as stated below.

Definition 4. ***Coloring Min-Distance Geometry Problem (CMDGP):*** *Given a simple weighted undirected graph $G = (V, E, d)$, with $d : E \to \mathbb{Z}^+$, find an embedding $a : V \to \mathbb{N}$ such that $|a(u) - a(v)| \geq d_{uv}$ for each $uv \in E$ and $\max_{v \in V} a(v)$ is the minimum possible.*

Since, in a wireless network, we can have multiple types of links, a more general model includes both equality and inequality distance constraints, defined in the following model.

Definition 5. ***Mixed Coloring Distance Geometry Problem (MCDGP):*** *Given a simple weighted undirected graph $G = (V, E, d)$, with $d : E \to \mathbb{Z}^+$, and a binary edge function $f : E \to \{0, 1\}$, find an embedding $a : V \to \mathbb{N}$ such that, for each $uv \in E$, $|a(u) - a(v)| \geq d_{uv}$ if, and only if, $f(uv) = 0$; or $|a(u) - a(v)| = d_{uv}$ if, and only if, $f(uv) = 1$, and where $\max_{v \in V} a(v)$ is the minimum possible.*

4. Branch-Prune-and-Bound for MCDCP-Multi-List

Consider the MCDCP problem stated in the previous section. By adding list coloring constraints and multicoloring demands, we have the MCDCP-Multi-List problem that is stated below.

Definition 6. ***Mixed List-Multicoloring Distance Geometry Problem (MCDCP-Multi-List):*** *Let $G = (V, E)$ be an undirected graph. For each edge $uv \in E$, there is an integer value*

$d(uv)$ and a binary value $f(uv)$. For each vertex, there is a weight c_v which is the number of different colors that must be assigned to v and a list $l(v)$ of possible colors that can be assigned to v. A feasible coloring for G, c and d consists of a set A, where $A(v)$ is the set of colors assigned to v such that $|A(v)| = c_v$; for all $1 \leq k \leq c_v$, $A(v,k)$ is the k-th assigned color of v, $A(v,k) \in l(v)$ and, for each edge uv, $|A(v,k) - A(u,h)| \geq d_{uv}$ if, and only if, $f(uv) = 0$; or $|A(v,k) - A(u,h)| \geq d_{uv}$ if, and only if, $f(uv) = 1$. The objective is to find a feasible assignment whose coloring span is the minimum possible.

Algorithm 1 Branch-Prune-and-Bound for optimization version of MCDCP-Multi-List

```
function BRANCH-PRUNE-AND-BOUND(V, c, t, l, A, B, Ub, Lb, Block)
    if all demands have been satisfied then
        if span of assignment A is less than Ub then
            B ← A; Ub ← assignment of span A
            if Ub = Lb then return B (optimal solution)
            end if
        end if
    else for all v ∈ V
        if demands of v have not yet been fully satisfied then
            for all k ∈ l(v) do
                if Block(v, k) = 0 then
                    Assign color k to v (and decrement current demand of v)
                    if current span of A is less than Ub then
                        for each vertex u such that c_vu > 0 do
                            for all m ∈ l(u) do
                                if |k − m| ≤ c_uv then Block(u, m) ← Block(u, m) + 1
                                end if
                            end for
                        end for
                        BRANCH-PRUNE-AND-BOUND(V, c, t, l, A, B, M, Block)
                        for each vertex u such that c_vu > 0 do
                            for all m ∈ l(u) do
                                if |k − m| ≤ c_uv then Block(u, m) ← Block(u, m) − 1
                                end if
                            end for
                        end for
                    end if
                end if
                Remove color k from v (and increment current demand of v)
            end for
        end if
    end if
    return B
end function
```

For solving this problem, we propose a branch-prune-and-bound algorithm. First, we choose a vertex with demands that have not yet been fulfilled. Then, a color from the list associated to the vertex is picked and, if it is not blocked for the vertex, it is added to the current assignment. If all colors from the list are blocked, the node is pruned. Then we check if the span of the current assignment is greater than or equal a given upper bound Ub. If it is, the node is cut, otherwise, the method is recursively applied. The algorithm is then recursively applied, and when all the demands are satisfied, we have a full feasible solution for the problem, and if its cost is lower than Ub, then it becomes the new upper bound. To check for the blocked colors, we use a matrix $Block$, with all elements initially set to 0, where $Block_{vk} > 0$ if color k is blocked for v and $Block_{vk} = 0$ otherwise. When the current upper bound is equal to the given lower bound Lb, we have the optimal solution for the instance. The algorithm can be executed

more efficiently when good bounds (a upper bound found by a heuristic and a lower bound found by a linear relaxation, for example) are given. Pseudocode for the B&P&B is given in Algorithm 1.

5. Computational experiments

The Branch-Prune-and-Bound algorithm was implemented in C language and executed in a computer equipped with a Intel Core i7 processor (3.4GHz) and 12GB of RAM. Some instances were generated with n vertices ($n \in [4, 10]$), where we derived scenarios for problems CMDGP-List, MCDGP-List (with distinct and equal lists), CMDGP-Multi-List and MCDGP-Multi-List.

The results of the experiments are given in Table 1 and Figure 2. The multicoloring instances are harder to solve (since, for each color demand, we are essentially duplicating the vertex), so the time needed to obtain the optimal solutions for $n \geq 7$ in these problems was too high. When equal distances are allowed in the problem, the runtime is lower, since there are less options for assigning colors that respect equality constraints. However, when we introduce equal lists, the time increases, since each vertex has more options now.

Table 1: Results for the Branch-Prune-and-Bound algorithm for generated instances of various types and sizes. Column |V| indicates the number of nodes in the graph; column SP gives the coloring span; column BND is the number of cuts by bounding; column PRN is the number of cuts by pruning; column SOL is the number of solutions and column T is the total CPU time for the algorithm.

	CMDGP-List					**MCDGP-List**					**MCDGP-List (Equal Lists)**				
|V|	**SP**	**BND**	**PRN**	**SOL**	**T**	**SP**	**BND**	**PRN**	**SOL**	**T**	**SP**	**BND**	**PRN**	**SOL**	**T**
4	7	1165	280	1	0.000	7	1165	280	1	0.000	4	6066	2301	2	0.000
5	8	10174	6385	1	0.000	8	10174	6385	1	0.000	5	147910	81783	2	0.020
6	10	305652	2875327	2	0.160	10	72589	109671	2	0.070	6	2175875	1731638	5	0.580
7	13	3759459	84146237	3	4.630	15	352338	4421436	1	2.390	9	97921297	164055357	4	42.950
8	13	73343425	1708180121	3	112.420	13	73343425	1708180121	3	86.890	9	535223979	915288021	4	301.170
9	13	557401766	450263944	3	877.640	-	-	-	-	-	-	-	-	-	-

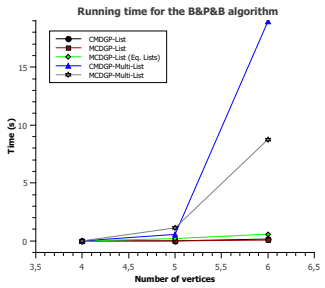


Figure 2: Number of vertices *times* running time for the B&P&B algorithm and problems used.

	CMDGP-Multi-List					**MCDGP-Multi-List**				
|V|	**SP**	**BND**	**PRN**	**SOL**	**T**	**SP**	**BND**	**PRN**	**SOL**	**T**
4	7	4151	1191	1	0.000	7	1907	42714	1	0.000
5	10	2722793	3400106	3	0.560	10	508274	11309799	3	1.210
6	10	18391637	136594416	4	18.940	10	3575337	91939693	4	8.750

6. Concluding remarks

In this work, we presented some graph coloring models with distance constraints which arise in channel assignment planning in cellular networks. These problems can be seen as the positioning of color points according to the distances of the vertices, so it's possible to solve them as distance geometry problems using branch-prune-and-bound method.

Ongoing research includes applying other algorithmic strategies to determine lower and upper bounds for the problem so the bounding occurs faster, analyzing the problem structure to make the pruning more effective, and applying these strategies to both real and artificial benchmark instances.

References

[1] G. K. Audhya, K. Sinha, S. C. Ghosh, and B. P. Sinha. A survey on the channel assignment problem in wireless networks. *Wireless Communications and Mobile Computing*, 2011.

[2] J. A. Bondy and U. S. R. Murty. *Graph Theory and its Applications.* MacMillan Press, 1976.

[3] Q. Dong and Z. Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22(1–4):365–375, 2002.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[5] W. K. Hale. Frequency assignment: theory and applications. *Proc. of IEEE*, 68(12):1497–1514, 1980.

[6] A. M. C. A. Koster. *Frequency assignment: models and algorithms.* Universiteit Maastricht, 1999.

[7] A. M. C. A. Koster and X. Muñoz. Graphs and algorithms in communication networks on seven league boots. In *Graphs and Algorithms in Communication Networks*, pages 1–59. 2010.

[8] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *European Journal of Operational Research*, 52(1):115–146, 2012.

[9] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. Recent advances on the discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 219(3):698–706, 2012.

[10] A. Lim, Y. Zhu, Q. Lou, and B. Rodrigues. Heuristic Methods for Graph Coloring Problems. In *Proc. of 2005 ACM Symposium on Applied Computing*, pages 933–939, 2005.

[11] R. A. Murphey, P. M. Pardalos, and M. G. C. Resende. Frequency Assignment Problems. In *Handbook of Combinatorial Optimization*, pages 295–377. Kluwer Academic Publishers, 1999.

[12] R. Rodrigues, B. Dias, and N. Maculan. Global and local optimization approaches for channel assignment in wireless networks. *Journal of Global Optimization*, 2013. Submitted (Special edition for Global Optimization Workshop 2012).

[13] R. Rodrigues, M. Dourado, and J. Szwarcfiter. Graph coloring and scheduling problems. 4th Latin American Workshop on Cliques in Graphs, 2010.

DGA 2013, pp. 115 – 118.

Optimality of Functionals on Delaunay Triangulations*

Nikolay P. Dolbilin,[1] Herbert Edelsbrunner,[2] Alexey Glazyrin,[3] and Oleg R. Musin[3]

[1]*Steklov Mathematics Institute, Moscow, Russian Federation.*

[2]*IST Austria (Institute of Science and Technology Austria), Klosterneuburg, Austria, Departments of Computer Science and of Mathematics, Duke University, Durham, North Carolina, and Geomagic, Research Triangle Park, North Carolina.*

[3]*Mathematics Department, University of Texas at Brownsville, Texas, USA.*

Abstract We study densities of functionals over uniformly bounded triangulations of a Delaunay set of vertices, and prove that the minimum is attained for the Delaunay triangulation if this is the case for finite sets.

Keywords: Delaunay sets, triangulations, Delaunay triangulations, uniformly bounded triangulations, functionals, densities.

1. Background

In this section, we introduce the background on Delaunay sets, their uniformly bounded triangulations, and functionals on such triangulations.

1.1 Delaunay Sets

$X \subseteq \mathbb{R}^d$ is a *Delaunay set* if there are positive constants $r < R$ such that (I) every open ball of radius r contains at most one point of X, and (II) every closed ball of radius R contains at least one point of X. Hence, X has no tight cluster and leaves no large hole.

1.2 Delaunay triangulations

Following the original idea of Boris N. Delaunay, we consider d-simplices with vertices from X such that the open ball bounded by the $(d-1)$-dimensional circumsphere contains no points of X. We call such d-simplices *empty*. Here, it is convenient to assume that X is *generic* in the sense that no $d+2$ points in X lie on a common $(d-1)$-sphere. Under this assumption, the empty d-simplices fit together without gap and overlap.

Theorem 1 (Delaunay Triangulation Theorem [2])**.** *Let X be a generic Delaunay set in $\mathbb{R}^d$. The collection of empty d-simplices together with their faces form a triangulation of X, commonly known as the* Delaunay triangulation*, $Del(X)$.*

*This research is partially supported by the Russian Government under the Mega Project 11.G34.31.0053, RFBR grant 11-01-00735, DMS 1101688, and the European Science Foundation (ESF) under the Research Network Programme.

1.3 Uniformly Bounded Triangulations

Let X be a generic Delaunay set in $\mathbb{R}^d$, and let T be a triangulation of X. This means that T is a simplicial complex with vertex set X whose underlying space is $\mathbb{R}^d$. T is called *uniformly bounded* if there is a real number $q = q(T)$ such that the radius of the circumsphere of every d-simplex in T is smaller than or equal to q. It follows that no edge of T is longer than $2q$. Note that the Delaunay triangulation of X is uniformly bounded with $q = R$.

1.4 Functionals

Let $\mathcal{S}_d$ be the set of all d-simplices in $\mathbb{R}^d$, including degenerate ones. We are interested in functionals that have constant upper and lower bounds for the simplices that arise in uniformly bounded triangulations of Delaunay sets. For other degenerate simplices we also allow infinity as a value.

Definition 2. *Let $\mathcal{E}$ be the class of functionals $F : \mathcal{S}_d \to \mathbb{R}$ for which there are constants $e = e(r, q, d)$ and $E = E(r, q, d)$ such that $e \le F(\sigma) \le E$ for all d-simplices σ with edges of length at least $2r$ and radius of the circumsphere at most q.*

1.5 Densities

We define the density of a functional on a triangulation by taking the lower limit over a growing ball, of the sum of values over all d-simplices in the ball divided by the volume of the ball:

$$f(T) \quad = \quad \liminf_{\alpha\to\infty} \frac{1}{Vol(\mathbb{B}_\alpha)} \sum_{\mathbb{B}_\alpha \supseteq \sigma \in T} F(\sigma). \tag{1}$$

1.6 Subclasses

We are interested in two subclasses of functionals, $\mathcal{G} \subseteq \mathcal{F} \subseteq \mathcal{E}$, which we now introduce. To define $\mathcal{F}$, let Y be a generic set of $d+2$ points in $\mathbb{R}^d$ such that no point lies inside the convex hull of the others. The non-degenerate d-simplices spanned by the points cover the convex hull twice; see Radon [10]. Indeed, we can split them into two collections such that each forms a triangulation of Y: the Delaunay triangulation, $D = Del(Y)$, and the other triangulation, T. Changing one triangulation into the other is a *flip*, a name motivated by the planar case in which it replaces one diagonal of a convex quadrilateral with the other. We give the flip a direction, leading from T to D. Let now F be a functional, let Σ_T be the sums of $F(\sigma)$ over all d-simplices in T, and define Σ_D similarly.

Definition 3. *The class $\mathcal{F}$ consists of all functionals $F \in \mathcal{E}$ for which $\Sigma_D \le \Sigma_T$.*

In $\mathbb{R}^2$, the extra property of functionals in $\mathcal{F}$ suffices to prove our main result. In $\mathbb{R}^d$, for $d \ge 3$, we need more structure. The reason is the existence of triangulations that cannot be turned into the Delaunay triangulation by a sequence of directed flips; see [5] for finite examples in $\mathbb{R}^3$. Such examples do not exist in $\mathbb{R}^2$; see [7].

Let now Y be a finite set of points in $\mathbb{R}^d$. As before, we assume that Y is generic. Let T' be a simplicial complex with vertex set Y, but note that we do not require that T' be a triangulation of Y. For example, we could start with a triangulation of Y and construct T' as the subset of d-simplices that do not belong to the Delaunay triangulation together with their faces. Let D' be the subset of simplices in $Del(Y)$ contained in the underlying space of T'. Finally, let $\Sigma_{T'}$ be the sum of $F(\sigma)$ over all d-simplices in T', and define $\Sigma_{D'}$ similarly.

Definition 4. *The class $\mathcal{G}$ consists of all functionals $F \in \mathcal{E}$ for which $\Sigma_{D'} \le \Sigma_{T'}$.*

The condition for F to belong to $\mathcal{G}$ is at least as strong as that for F to belong to $\mathcal{F}$, which implies $\mathcal{G} \subseteq \mathcal{F}$.

2. Results

2.1 Main Theorem

The main result of this paper is an extension of optimality results for Delaunay triangulations from finite sets to Delaunay sets, which are necessarily infinite.

Main Theorem 5. *Let X be a Delaunay set in $\mathbb{R}^d$.*

(i) In $\mathbb{R}^2$, $F \in \mathcal{F}$ implies $f(Del(X)) \leq f(T)$ for all uniformly bounded triangulations T of X.

(ii) In $\mathbb{R}^d$, $F \in \mathcal{G}$ implies $f(Del(X)) \leq f(T)$ for all uniformly bounded triangulations T of X.

2.2 Implications in the Plane

There are many functionals on triangles that are known to be in $\mathcal{F}$. Applying the Main Theorem thus gives many optimality results for Delaunay triangulations of Delaunay sets.

Corollary 6. *Let σ be a triangle in $\mathbb{R}^2$, with edges of length a, b, c, let $c_1 > 0$ and $c_2 \geq 1$ be constants, and consider the following list of functionals:*

$F_1(\sigma) = Circumradius^{c_1}(\sigma)$; $F_2(\sigma) = Circumradius^{c_2}(\sigma) \cdot Area(\sigma)$; $F_3(\sigma) = -Inradius(\sigma)$; $F_4(\sigma) = (a^2 + b^2 + c^2)/Area(\sigma)$; $F_5(\sigma) = (a^2 + b^2 + c^2) \cdot Area(\sigma)$; $F_6(\sigma) = ||Centroid(\sigma) - Circumcenter(\sigma)||^2 \cdot Area(\sigma)$.

Then $f_i(Del(X)) \leq f_i(T)$ for every Delaunay set $X \subseteq \mathbb{R}^2$, for every uniformly bounded triangulation T of X, and for $1 \leq i \leq 6$.

2.3 Implication in d Dimensions

We have one example of a functional on d-simplices that is in $\mathcal{G}$, namely the extension of F_5 to three and higher dimensions. Writing a_1 to a_k for the lengths of the $k = \binom{d+1}{2}$ edges of a d-simplex σ, we define $F_R(\sigma) = Vol(\sigma) \sum_i a_i^2$; see also [1]. Rajan proved that for finite sets in $\mathbb{R}^d$, the density of F_R attains its minimum for the Delaunay triangulation. We extend his proof to show that F_R belongs to $\mathcal{G}$. With this, we get another consequence of the Main Theorem.

Corollary 7. *We have $f_R(Del(X)) \leq f_R(T)$ for every Delaunay set $X \subseteq \mathbb{R}^d$ and for every uniformly bounded triangulation T of X.*

References

[1] A. V. Akopyan. Extremal properties of Delaunay triangulations. *Trudy ISA RAS* **46** (2009), 174–187.

[2] B. N. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk* **7** (1934), 793–800.

[3] B. N. Delone. Geometry of positive quadratic forms. *Uspekhi Mat. Nauk* **3** (1937) 16–62.

[4] N. P. Dolbilin, O. R. Musin and H. Edelsbrunner. On the optimality of functionals over triangulations of Delaunay sets *Uspekhi Mat. Nauk* 67 (2012), 189–190.

[5] B. Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Statist. Comput.* **10** (1989), 718–741.

[6] T. Lambert. The Delaunay triangulation maximizes the mean inradius. *In* "Proc. 6th Canad. Conf. Comput. Geom., 1994" 201–206.

[7] C. L. Lawson. Software for C^1 surface interpolation. In *Mathematical Software III*, Academic Press, New York, 1977, 161–194.

[8] O. R. Musin. Properties of the Delaunay triangulation. *In* Proc. 13th Ann. Sympos. Comput. Geom., 1997" 424–426.

[9] O. R. Musin. About optimality of Delaunay triangulations. *Geometry, Topology, Algebra and Number Theory, Applications,* Internat. Conf. dedicated to 120th anniversary of B. N. Delone, 2010, 166–167.

[10] J. Radon. Mengen konvexer Körper, die einen gemeinschaftlichen Punkt enthalten. *Math. Ann.* **83** (1921), 113–115.

[11] V. T. Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. *Discrete Comput. Geom.* **12** (1994), 189–202.

[12] G. F. Voronoi. Nouvelles applications des parametres continus a la theorie des formes quadratiques. *J. Reine Angew. Math.* **34** (1908), 198–287.

DGA 2013, pp. 119 – 124.

Quaternions as a tool for merging multiple realization trees*

Felipe Fidalgo[1] and Jaime Rodriguez[2]

[1]*Department of Applied Mathematics, IMECC - UNICAMP, Campinas, Brazil,* felipefidalgo@ime.unicamp.br

[2]*Department of Mathematics, UNESP, Ilha Solteira, Brazil,* jaime@mat.feis.unesp.br

Abstract This work uses Quaternion Algebra as a tool to improve the resolution of the Multiple Realization Trees method which solves the Discretizable Molecular Distance Geometry Problem (DMDGP) by dividing the instances into smaller pieces producing more than one binary tree of realizations. Quaternion Rotations are used here to merge such trees, saving positions of memory and causing a decreasing in the number of operations.

Keywords: Molecular Geometry, Quaternion Algebra, Distance Geometry, Branch-and-Prune Algorithm

1. Introduction

From known distance values for pairs of atoms in a molecule, it is possible to formulate an inverse problem called *Molecular Distance Geometry Problem (MDGP)* [1, 5] which consists of finding a 3-D conformation for the molecule such that it satisfies the distance constraints. Such data usually come from chemical knowledge (like atomic bond lengths and bond angles) combined with a physical experimental method called *Nuclear Magnetic Resonance (NMR)* [3]. With additional assumptions, Lavor et. al [5] proposed a discrete formulation for a subclass of the MDGP, which is called *Discretizable Molecular Distance Geometry Problem (DMDGP)*. In addition, an efficient method was also proposed for solving this problem, the Branch-and-Prune (BP) algorithm, which generates a binary tree with all possible solutions [5]. To make this method faster, Nucci et. al [6] proposed another one which uses the BP algorithm more than once, generating more than one tree, as shown in Section 2. Finally, Section 3 shows how to use quaternions in order to make the method, proposed by Nucci et. al, even more efficient, comparing it with the rotation matrix approach.

2. Multiple Realization Trees

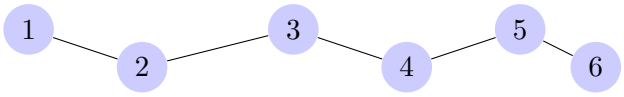
Given a molecule M in a backbone-chain shape, with an order $<$ on its set of atoms $\{1, \ldots, n\}$, we can split it into an union of intervals in an increasing order like

$$M = M_1 \cup M_2 \cup \ldots \cup M_k, \tag{1}$$

where $M_j = [a_j, b_j]$, $a_1 = 1$, $b_k = n$, $1 \leq a_j \leq a_{j+1} \leq n$, $b_j - a_{j+1} \geq 2$ and $j = 2, \ldots, k-1$. The whole molecule can be represented as the interval $M = [1, n]$. For example: let $M = [1, 6]$

*The authors would like to thank to the brazilian research agencies CNPq and FAPESP for the financial support.

be a molecule. So, it can be splitted into the union $M = M_1 \cup M_2$, where $M_1 = [1, 4]$ and $M_2 = [2, 6]$.



This division motivates what Nucci et. al [6] called as the method of *Multiple Realization Trees (MRT)*. Before describing it, we give some important definitions. A *Valid Realization* of a DMDGP instance M corresponds to a bijective embedding of it in $\mathbb{R}^3$ which satisfies the distance constraints, i.e., a three-dimensional conformation for M. A *Realization Tree*, in our case, is a binary-tree graph which represents, in a depth-first fashion, *all* the valid realizations which solves the DMDGP. A *Feasible Branch* is the name we give for each of the branches of a Realization Tree, i. e., each feasible branch represents one embedding of M in $\mathbb{R}^3$.

The MRT method applies the BP algorithm in each interval M_j, producing k realization trees T_j. We denote by T the realization tree which represents all the feasible realizations for throughout the DMDGP instance M. Back to our example, the BP method provides the trees T_1 and T_2, as in the figures below.

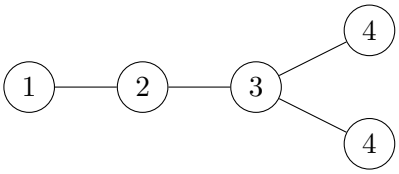


Figure 1: Tree T_1: 4 levels.

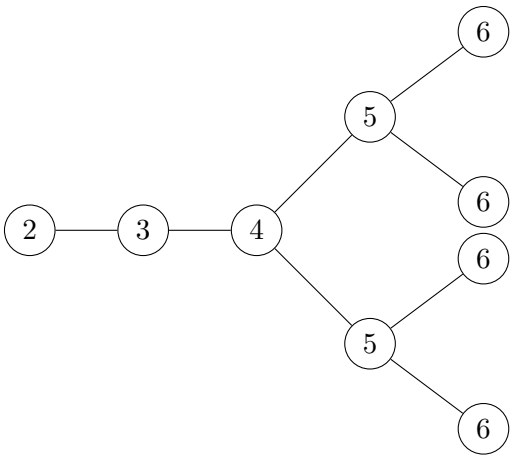


Figure 2: Tree T_2: 5 levels.

It is necessary to merge all the trees following the same order of split (1), aiming to get realizations of the whole molecule. The procedure is merging each branch from one tree to all branches from the other, one at a time. We denote the t^{th} branch of a tree T_p as $T_{p,t}$.

In order to produce mergeable trees, one has to assume that two consecutive intervals M_p and M_{p+1} have, at least, three atoms in the intersection [6]. Consider, then, the two consecutive trees T_p and T_{p+1} relative to the previously mentioned intervals. As they have three levels of intersection, we consider the tree T_p to be fixed, calling it *Base Tree*, and we move the other tree T_{p+1}, which we name *Sliding Tree*, towards T_p using Euclidean transformations in order to preserve lenghts and angles. Assume that the last three atoms of the base interval are i, j and k, respectively ordered, and let $T_{x,y}(z)$ be the generic notation for the position of the atom $z \in \{1, 2, \ldots, |T_{x,y}|\}$ in the branch y of the binary tree x. Also, if the number of feasible branches in a tree T is denoted by $|T|$, then the final number of feasible realizations of M, provided by the MRT method, is r, which is defined by the multiplication

$$r = |T_1||T_2| \ldots |T_k| \leq |T|.$$

Three Euclidean transformations are necessary to merge the arbitrary branches $T_{p+1,t}$ and $T_{p,q}$. The first one is a *translation* that makes $T_{p+1,t}(i) \to T_{p,q}(i)$, shown in Figure 2.

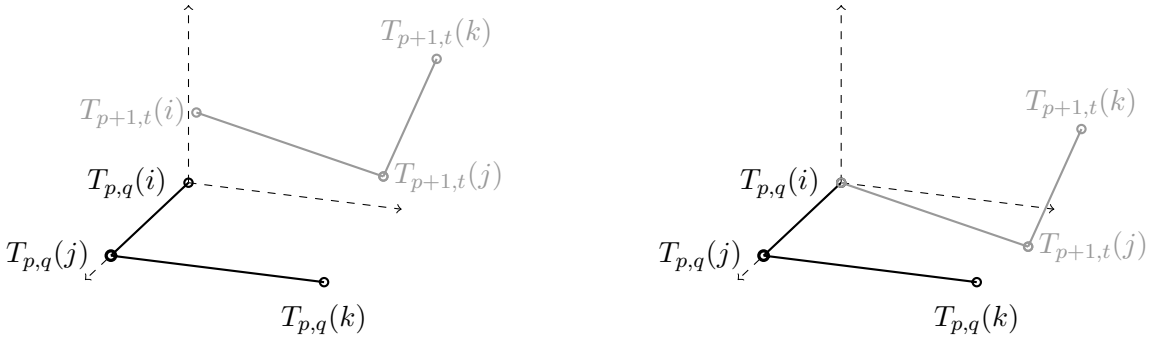


Figure 3: First Euclidean transformation: a translation of both realized branches.

We also want to make $T_{p+1,t}(j) \to T_{p,q}(j)$, without losing what we have built with the translation. Let us denote $E_p = T_{p,q}(j) - T_{p,q}(i)$ and $E_{p+1} = T_{p+1,t}(j) - T_{p+1,t}(i)$ and let θ be the angle between E_p and E_{p+1}. We apply a *plane rotation* of θ in the branch $T_{p+1,t}$, as one can see in Figure 4.

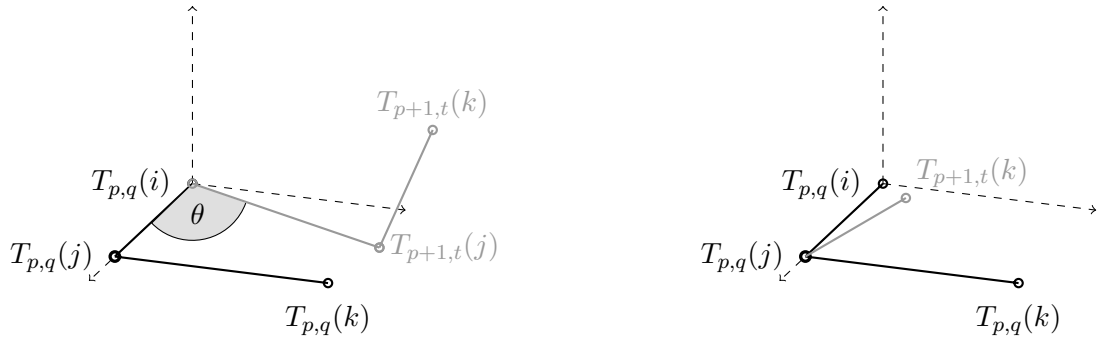


Figure 4: Second Euclidean transformation: a plane rotation in terms of θ.

Finally, after one translation and one rotation, consider $F_p = T_{p,q}(k) - T_{p,q}(j)$ and $F_{p+1} = T_{p+1,t}(k) - T_{p+1,t}(j)$. We want to move the sliding branch $T_{p+1,t}$ such that it satisfies $T_{p+1,t}(k) \to T_{p,q}(k)$, without moving anything else which has been already transformed previously. We define the rotation axis, whose attitude L is spanned by $T_{p,q}(j) - T_{p,q}(i)$, and consider the plane $\mathbb{P}$, orthogonal to this axis. Let $\mathbf{P} = I_3 - LL^T$ be the matrix that gives the orthogonal projection to $\mathbb{P}$.

Then, the projections of F_p and F_{p+1} in $\mathbb{P}$ are, respectively,

$$P_p = \mathbf{P}F_p \qquad \text{and} \qquad P_{p+1} = \mathbf{P}F_{p+1}.$$

Now, let φ be the angle between P_p and P_{p+1}. Thus, we rotate F_{p+1} towards F_p in φ about the axis spanned by L, as it is shown in Figure 5. So we do with the remaining structure. Therefore, both realizations are connected and supposed to respect all original distance and angle constraints.

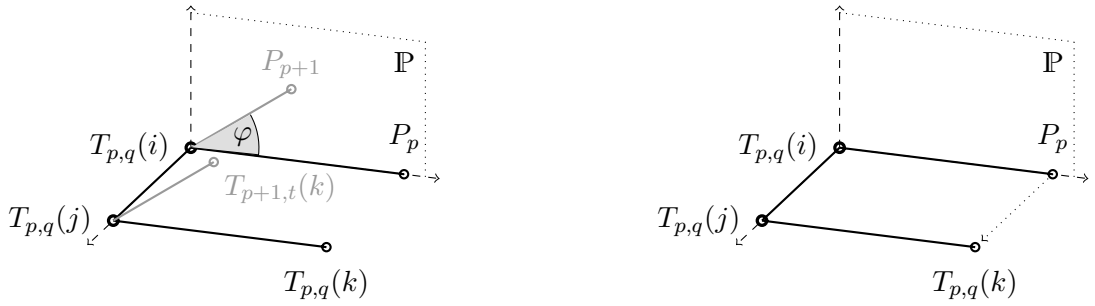


Figure 5: Third transformation: a spatial rotation in terms of the projected angle φ.

Following this outline, all the trees are connected and their branches consist of feasible points that solve the DMDGP. In addition, we remark that all rotations are computed using matrices.

3. Merging Trees with Quaternion Rotations

All general rotations in real 3-D space can be represented by an axis, spanned by a unitary vector $\mathbf{n}$, and an angle θ. Using this information, one can build the matrix $R_{\mathbf{n},\theta}$ which carries out a general rotation and can be determined by using the matrix form of *Rodrigues' Rotation Formula* [7]

$$R_{\mathbf{n},\theta} = I + \sin(\theta)J(\mathbf{n}) + (1 - \cos(\theta))J(\mathbf{n})^2, \tag{2}$$

where $J(\mathbf{n})$ is a skew-symmetric 3×3 - matrix generated by the elements of $\mathbf{n}$ as

$$J(\mathbf{n}) = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

Such rotation matrices need 37 arithmetic operations to be determined, according to Equation (2), and 9 positions of memory to be stored. In addition, it is necessary to use more 15 operations to multiply it for a vector $\mathbf{v}$ we want to rotate, totalizing 52 arithmetic operations.

This work aims to propose a theoretical modification on the tools which are used to make rotations on three-dimensional structures in order to decrease the storage space and, consequently, the number of operations to accelerate this process. Our approach uses the Quaternion Algebra $\mathbb{H}$ [4] to do that.

Consider the unit quaternion $q = q_0 + \mathbf{q_v}$, where $q_0 \in \mathbb{R}$ and $\mathbf{q_v} \in \mathbb{R}^3$. It is possible to prove that there is an unique angle $0 \leq \theta \leq \pi$ such that $q_0 = \cos(\theta)$ and $\|\mathbf{q_v}\| = \sin(\theta)$. Then, we can rewrite $q = \cos(\theta) + \mathbf{u}\sin(\theta)$, where $\mathbf{u} = \frac{\mathbf{q_v}}{\|\mathbf{q_v}\|}$ [4]. The conjugate of q can be written as $q^* = \cos(\theta) - \mathbf{u}\sin(\theta)$ [4]. Now, the following outcome characterizes a quaternion rotation by means of a linear operator[4].

Theorem 1 (Quaternion Rotation Operator)**.** *For a unit quaternion $q = \cos(\theta) + \mathbf{u}\sin(\theta)$, the operator $R_q : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$, whose action on the vector $\mathbf{v} \in \mathbb{R}^3$ is given by $R_q(\mathbf{v}) = q\mathbf{v}q^*$, is a*

rotation operator which rotate vectors about the axis spanned by the unit vector $\mathbf{u}$ *through an angle* 2θ *in clockwise sense.*

Explicitly, the action of R_q in a vector $v \in \mathbb{R}^3$ can be derived as the *Rodrigues' Rotation Formula*

$$R_q(\mathbf{v}) = \cos(2\theta)\mathbf{v} + (1 - \cos(2\theta))\mathbf{p_u}(\mathbf{v}) + \sin(2\theta)(\mathbf{u} \times \mathbf{v}), \tag{3}$$

where $\mathbf{p_u}(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})\mathbf{u}$ is the orthogonal projection of $\mathbf{v}$ in the direction of $\mathbf{u}$.

First of all, one improvement we can realize on the use of quaternions is that it is easier to encode a rotation in a unit quaternion than in an orthogonal matrix. Moreover, Equation (3) shows that each quaternion rotation creates a local frame $(\mathbf{v}, \mathbf{p_u}(\mathbf{v}), \mathbf{u} \times \mathbf{v})$ to represent the rotated vector and, also, gives us a route on how to localize it in the space more easily.

On computational complexity, quaternion rotations require the storage of only four positions instead of nine, necessary in the use of three-dimensional matrices. In addition, 25 arithmetic operations are used to compute such rotations, fashioned such as in Equation (3). As we can see, this number of operations is reasonably less than the one used in the matrix approach. When considering more than one rotation, it seems to be even more efficient by saving space and floating-point-arithmetic operations.

Therefore, we apply these ideas as efficient tools in the merging of BP-trees. According to the MRT procedure described previously, we have to carry out two rigid rotations in the sliding structure. For each one of them, it is necessary first to determine the cosine of the rotation angle by using the usual dot product in $\mathbb{R}^3$, restricting the domain of the cosine function to the range $[0, \pi]$ in order not to allow it to reach the position determined by the angle $2\pi - \theta$ since both angles have the same cosine value. Moreover, the unitary vector which spans the oriented rotation axis can be chosen by applying the usual cross product in 3-D Euclidian Space, whose signal induces the orientation of the rotation, following the so-called *Right-Hand Rule*. Thus, the order in the cross product really matters: indeed, the axis for a rotation of a vector $\mathbf{x}$ towards another vector $\mathbf{y}$ is spanned by the vector $\mathbf{x} \times \mathbf{y}$, while the axis for the rotation of $\mathbf{y}$ towards $\mathbf{x}$ is spanned by the vector $\mathbf{y} \times \mathbf{x}$. As they satisfy the relation of anticommutativity $\mathbf{y} \times \mathbf{x} = -(\mathbf{x} \times \mathbf{y})$, the direction of the rotation is, therefore, encoded in the sign of the cross product.

The first rotation is supposed to take E_{p+1} into E_p, since they have the same origin. Thus, the cosine of the angle and the unitary vector which spans the correspondent axis are, respectively,

$$\cos(\theta) = \frac{\langle E_{p+1}, E_p \rangle}{\|E_{p+1}\| \, \|E_p\|} \quad \text{and} \quad \mathbf{n} = \frac{E_{p+1} \times E_p}{\|E_{p+1} \times E_p\|}. \tag{4}$$

Using the parameters developed above, we associate the following quaternion element to such rotation

$$q_{\theta,\mathbf{n}} = \cos(\tfrac{\theta}{2}) + \mathbf{n}\sin(\tfrac{\theta}{2}).$$

Then, employing the result displayed in Theorem 1, we apply the rotation in the sliding structure $T_{p+1,t}$

$$T_{p+1,t}(u) \leftarrow R_{q_{\theta,\mathbf{n}}}(T_{p+1,t}(u)), \qquad \text{for } u = 2, \ldots, |T_{p+1,t}|, \tag{5}$$

where $|T_{p+1,t}|$ is the number of vertices in this feasible branch of the realization tree T_{p+1}.

Further, assume $L = E_p / \|E_p\|$, $F_p = T_{p,q}(k) - T_{p,q}(j)$ and $F_{p+1} = T_{p+1,t}(k) - T_{p+1,t}(j)$. The orthogonal projection matrix, associated to the plane $\mathbb{P}$, is given by $M = I_3 - LL^T$, as we have seen. Then, the projections are given by the vectors $P_p = MF_p$ and $P_{p+1} = MF_{p+1}$. Analogously to (4), the second rotation is generated by the parameters

$$\cos(\varphi) = \frac{\langle P_{p+1}, P_p \rangle}{\|P_{p+1}\| \, \|P_p\|} \quad \text{and} \quad \mathbf{m} = \frac{P_{p+1} \times P_p}{\|P_{p+1} \times P_p\|}. \tag{6}$$

After calculating them, we define the associated quaternion to the respective rotation by

$$q_{\varphi,\mathbf{m}} = \cos(\tfrac{\varphi}{2}) + \mathbf{m}\sin(\tfrac{\varphi}{2}).$$

Therefore, we rotate the sliding structure, again as in Theorem 1, by making

$$T_{p+1,t}(u) \leftarrow R_{q_{\omega,\mathbf{m}}}(T_{p+1,t}(u)), \qquad \text{for } u = 3, \ldots, |T_{p+1,t}|, \tag{7}$$

concluding the merging of the two structures $T_{p,q}$ and $T_{p+1,t}$.

There are two rotations in this approach. The first one fixes the first vertex of the sliding structure and the second one fixes both the first and the second vertices. Then, compounding both the rotations in only one and applying it in the sliding structure leads us to reach the same resulting structure. It is reasonably easier and computationally cheaper to compose two quaternion rotations than multiplying two rotation matrices [4, 2]. Using this, we can save half of the storage space and carry out less than the half of arithmetic operations for the transformation of each point.

As a conclusion, using quaternion rotations, instead of matrices, can bring improvements either about computational time or about simplifying the method. We are in the process of implementing these ideas in order to illustrate computationally the theoretical improvements.

References

[1] G. Crippen and T. Havel. *Distance Geometry and Molecular Conformation.* Research Studies Press, U.K., 1988.

[2] D. Eberly. Rotation Representation and Performance Issues. *A summary of representations of rotations and performance available in http://www.geometrictools.com/Documentation/RotationIssues.pdf*, 2002.

[3] T. Havel. Distance Geometry. In D. Grant and R. Harris, editors, *Encyclopedia of Nuclear Magnetic Resonance, pp. 1701–1710.* Wiley, New York. 1995.

[4] J. Kuipers. *Quaternions and Rotation Sequences.* Princeton University Press, Princeton, 1998.

[5] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[6] P. Nucci, L. Nogueira, and C. Lavor. Solving the Discretizable Molecular Distance Geometry Problem by multiple realization trees. In Mucherino, A., Lavor, C., Liberti. L., and Maculan, N., editors, *Distance Geometry: Theory, Methods and Applications.* Springer, New York, 2013.

[7] C. Taylor and D. Kriegman. Minimization on the lie group SO(3) and related manifolds. *Technical Report, Yale University*, 1994.

DGA 2013, pp. 125 – 129.

Updated T Algorithm for the resolution of Molecular Distance Geometry Problems by means of linear systems *

Felipe Fidalgo, Douglas Maioli and Eduardo Abreu

Department of Applied Mathematics, IMECC - UNICAMP, Campinas, Brazil,
felipefidalgo@ime.unicamp.br, douglasmaioli@bol.com.br, eabreu@ime.unicamp.br

Abstract The Molecular Distance Geometry Problem (MDGP) has several attempts for its resolution, such as those from the class of Geometric Build-up methods. This work deals with a new approach named *Updated T Algorithm*. It consists on solving linear systems, with LU factorization, together with the so-called *re-initialization* and a sequence of Euclidian transformations in order to build linear systems with better-condition-number properties. Numerical experiments with PDB (Protein Data Bank) intances are shown, comparing this method with one from the literature, aiming to bear out this approach.

Keywords: Molecular Distance Geometry, UT Algorithm, Root-Mean-Square Deviation, Geometric Build-Up Algorithms

1. Introduction

It is possible to obtain distance values corresponding to pairs of atoms in a molecule M from a combination of chemical knowledge (such as bond angles and bond lengths) with Nuclear Magnetic Resonance (NMR) data [5]. From them, we can formulate the Molecular Distance Geometry Problem (MDGP) as MDGP Given an n - atom molecule M, consider the set S_M of known distances d_{ij} between pairs of atoms $(i, j) \in \{1, \ldots, n\}^2$. Is it possible to find a 3 - D conformation $\mathbf{x}_1, \ldots, \mathbf{x}_n$ for M? This problem is NP-hard [4]. In this work, we present a method to solve it with relatively low CPU time that brings evidences of the possibility of treating uncertainties on distance data, which is still under investigation for a future work. It is called *Updated T (UT) Algorithm*. Numerical experiments of tests with proteins are shown, using the Root-Mean-Square Deviation (RMSD) as error estimator.

2. Updated T Algorithm

Let M be a molecule and S_M its associate pairwise-distance set. The method starts by choosing four non-coplanar atoms whose *all* distances between each other are known. It is reasonably simple to determine coordinates for this subset, which are called *Base Atoms* (see, e.g., [2], p. 325). If they are not found, stop. Let F be the set of the *positioned* atoms. Then, the iterating process starts: let j be the undetermined atom and $\mathbf{x}_j \in \mathbb{R}^3$ the position the method wants to find. It looks for a subset with four base atoms $B_j = \{\mathbf{x}_{j1}, \mathbf{x}_{j2}, \mathbf{x}_{j3}, \mathbf{x}_{j4}\} \subseteq F$ whose distances to j are in S. If it is not found, stop. The next step is called *re-initialization*: the

*We want to thank to CNPq, CAPES and FAPESP (2011/11897-6) for financial support.

four determined base atoms of the iteration have their positions changed in order to depend only on the distances. This is done following the same procedure for the first four atoms and it aims to avoid error accumulation from the iterative process of solving linear systems, as done in [3]. The new base atoms are $B_j^t = \{\mathbf{y}_{j1}, \mathbf{y}_{j2}, \mathbf{y}_{j3}, \mathbf{y}_{j4}\}$, which are ilustrated in the figure.

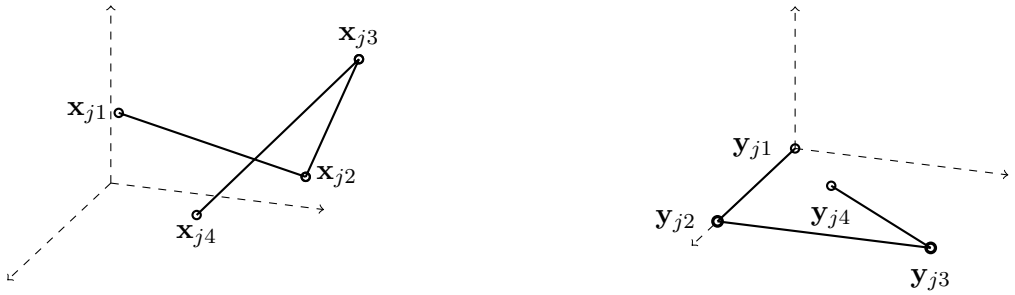


Figure 1: The re-initialization process: changing the coordinates of the Base Atoms.

These new positions, together with the distances in S_M, establish a quadratic system of equations

$$\|\mathbf{y}_{j1} - \mathbf{y}_j\| = d_{j1}, \quad \|\mathbf{y}_{j2} - \mathbf{y}_j\| = d_{j2}, \quad \|\mathbf{y}_{j3} - \mathbf{y}_j\| = d_{j3} \text{ and } \|\mathbf{y}_{j4} - \mathbf{y}_j\| = d_{j4}, \tag{1}$$

where $\mathbf{y}_j$ is the position of the undetermined atom in the transformed framework. The next result is the core of the method: $\mathbf{y}_j$ is calculated by the solution of a linear system to be described in what follows.

Theorem 1 (Fidalgo, [3])**.** *Let $B_j^t = \{y_{j1}, y_{j2}, y_{j3}, y_{j4}\}$ be a set of base atoms for j whose distances to it are all known. If y_j is a solution for the quadratic system* (1)*, then $\mathbf{x} = \begin{bmatrix} t_j & \mathbf{y}_j^T \end{bmatrix}^T$, where $t_j = -\frac{\|\mathbf{y}_j\|^2}{2}$, is the unique solution of the linear system $A\mathbf{x} = b$ with*

$$A = \begin{bmatrix} 1 & \mathbf{y}_{j1}^T \\ 1 & \mathbf{y}_{j2}^T \\ 1 & \mathbf{y}_{j3}^T \\ 1 & \mathbf{y}_{j4}^T \end{bmatrix} \qquad \textit{and} \qquad b = \begin{bmatrix} d_{j1}^2 - \|\mathbf{y}_{j1}\|^2 \\ d_{j2}^2 - \|\mathbf{y}_{j2}\|^2 \\ d_{j3}^2 - \|\mathbf{y}_{j3}\|^2 \\ d_{j4}^2 - \|\mathbf{y}_{j4}\|^2 \end{bmatrix}.$$

After calculating y_j, the method ought to put it back to its position x_j in the original framework. For this, we work with rigid Euclidean transformations, also used by Wu et. al [1, 6]. Consider the matrices X and Y, whose rows consists on the positions of the base atoms $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively. Thus, the geometric center of both structures, represented by these matrices, can be determined by

$$\mathbf{x}_c(k) = \frac{1}{4}\sum_{i=1}^{4} X(i,k) \qquad \text{and} \qquad \mathbf{y}_c(k) = \frac{1}{4}\sum_{i=1}^{4} Y(i,k) \qquad (k = 1, 2, 3).$$

Then, we work out the translation on Y below so that both structures have the same geometric center

$$Y(i,j) = Y(i,j) - [\mathbf{y}_c(j) - \mathbf{x}_c(j)], \qquad (j = 1, 2, 3). \tag{2}$$

The same translation is applied to the position $\mathbf{y}_j$ we want to transform.

Finally, following Wu et. al [6], we have to find an orthogonal matrix Q so that the structure of Y is rotated into X, in order to achieve the RMSD value between both matrices. This is done by the resolution of the *Orthogonal Procrustes Problem*

$$\min_{\text{s. to } Q^TQ=I} \|X - YQ\|_F. \tag{3}$$

The matrix $Q = UV^T$ solves this problem. U and V are the orthogonal matrices of the Singular Value Decomposition of the matrix Y^TX [3]. Applying the same rotation to $\mathbf{y}_j$, we find the position $\mathbf{x}_j$ and include it in F. It follows a picture illustrating this and the outline of the UT algorithm.

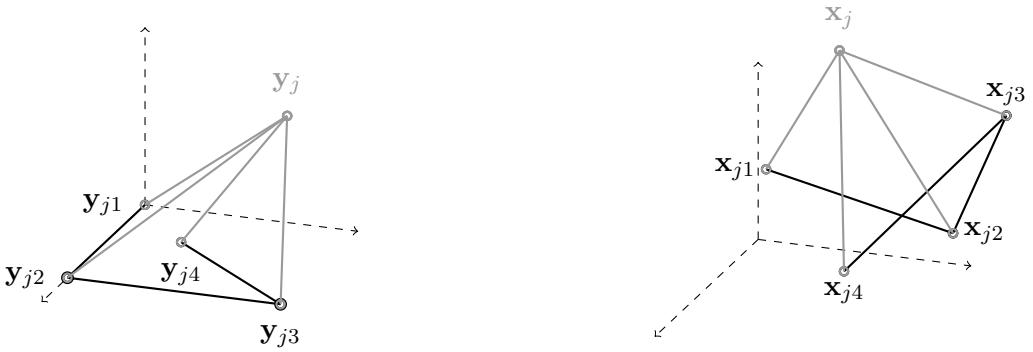


Figure 2: After a sequence of Euclidian transformations, the atom j is put back on its original position.

Algorithm 1 Updated T Algorithm

1: Find four base atoms, determine $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$ and let F be the set of determined atoms;
2: **if** They are not found **then**
3: **return**
4: **end if**
5: **loop**
6: For each unpositioned atom j, find four base atoms $\mathbf{x}_{j1}, \mathbf{x}_{j2}, \mathbf{x}_{j3}$ and $\mathbf{x}_{j4}$;
7: Re-initialize the base atoms into $\mathbf{y}_{j1}, \mathbf{y}_{j2}, \mathbf{y}_{j3}$ and $\mathbf{y}_{j4}$ and find the position $\mathbf{y}_j$ (Theorem 1);
8: Put back $\mathbf{y}_j$ to the original structure into $\mathbf{x}_j$ and include $\mathbf{x}_j$ in F;
9: **end loop**

3. Computational Issues and Numerical Experiments

The implementation and the computational tests have been done using MATLAB in a computer Intel Core i3, 3.07 GHZ processor and RAM memory with 4 GB. All the linear systems were solved by using LU factorization with partial pivoting. We used a subgroup of the proteins tested by Wu et al. [6] and Davis et. al [1], from the Protein Data Bank (PDB), and the numerical experiments with the UT algorithm are compared to the *Updated Geometric Build-Up (UGB) Algorithm* [1, 6] results. Both methods have similar outlines, but their cores consist on resolutions of different linear systems. In addition, UT solves only one system per iteration instead of four ones, as it is in UGB. For the tests, we used the RMSD as an error estimator

$$RMSD(X,Y) = \min_{\text{s. to } Q^TQ=I} \frac{\|X - YQ\|_F}{\sqrt{n}}, \tag{4}$$

where the original and the calculated instances are stored as rows in the matrices X and Y, respectively.

A subset of the results for these instances is shown in Table 1. Such tests were carried out using 6 Å as the cut-off value for the distances. The first and second columns bring,

respectively, the name of the tested PDB instance and its number of atoms, which are also displayed in Figure 3. The third and fifth columns store CPU time for the UT and the UGB methods, respectively. One can see that the first method exhibits better performance in all the cases. Having less systems may save more time. Figure 3 also shows this fact. The last column shows a relative time, when comparing the time values: it indicates an improvement average of 60.78% for the UT. Finally, the fourth and sixth columns bring RMSD values for both the methods. These values are precisely close to each other in all the tests. Then, UT is able to solve an MDGP with the same accuracy order of UGB, but demanding less CPU time, what is promising desirable.

PDB Name	# atoms	UT time (s)	UT RMSD	UGB Time (s)	UGB RMSD	Time(UT/UGB)
1ID7	189	4,74E-02	3,12E-09	8,27E-02	3,12E-09	57,32%
1FW5	332	7,19E-02	1,18E-08	1,19E-01	1,18E-08	60,42%
1JAV	360	7,66E-02	1,38E-07	1,26E-01	1,38E-07	60,70%
1MEQ	405	8,39E-02	6,39E-11	1,36E-01	6,42E-11	61,51%
1AMB	438	8,86E-02	8,22E-06	1,45E-01	8,22E-06	61,15%
1R7C	532	1,07E-01	8,39E-07	1,68E-01	8,38E-07	63,83%
1HLL	540	1,06E-01	1,59E-07	1,71E-01	1,59E-07	61,96%
1VII	596	1,12E-01	9,19E-07	1,92E-01	9,19E-07	58,55%
1HIP	617	1,13E-01	3,15E-09	1,95E-01	3,14E-09	57,98%
1ULR	677	1,24E-01	2,82E-09	2,09E-01	2,81E-09	59,20%
1KVX	954	1,61E-01	1,65E-06	2,77E-01	1,65E-06	58,19%
1VMP	1166	2,04E-01	1,97E-07	3,63E-01	1,97E-07	56,25%
1RGS	2015	3,11E-01	1,37E-08	5,63E-01	1,37E-08	55,23%
1BPM	3671	5,94E-01	5,08E-06	1,04E+00	5,08E-06	57,11%

Table 1: Numerical experiments of tests with PDB instances - 6 Å cut-off

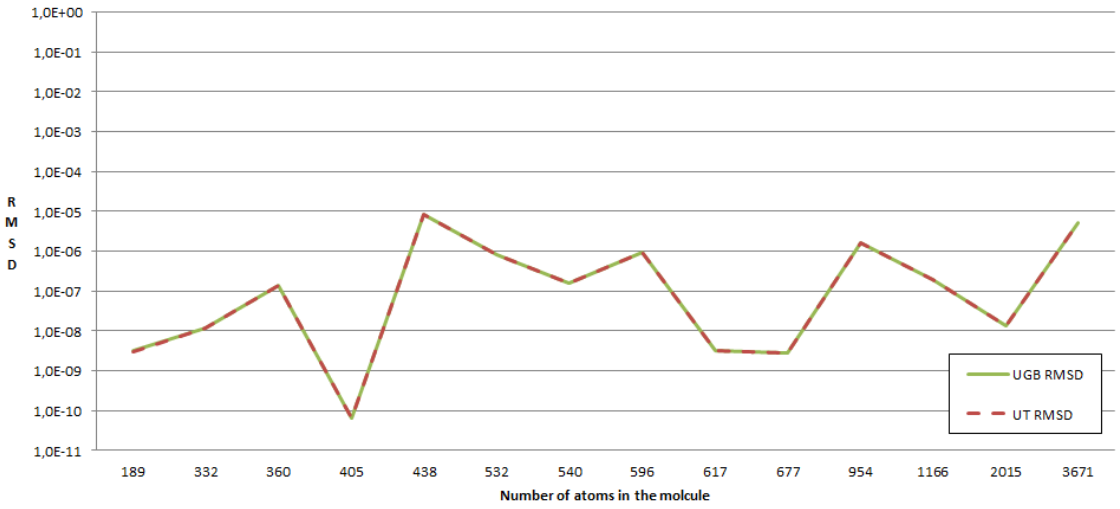


Figure 3: Graphic representation of the RMSD results for the computational tests with proteins from the PDB.

4. Conclusions

Concluding, this work deals with the Updated T (UT) Algorithm which solves the Molecular Distance Geometry Problem (MDGP) by applying a sequence of linear-system resolutions and Euclidean rigid transformations based on Root-Mean-Square Deviation (RMSD) techniques. It has shown as good numerical stability and accuracy as the Updated Geometric Build-Up method (from the literature) does and has taken less CPU time to determine protein structures. Such instances have been extracted from the Protein Data Bank (PDB). For future work, our

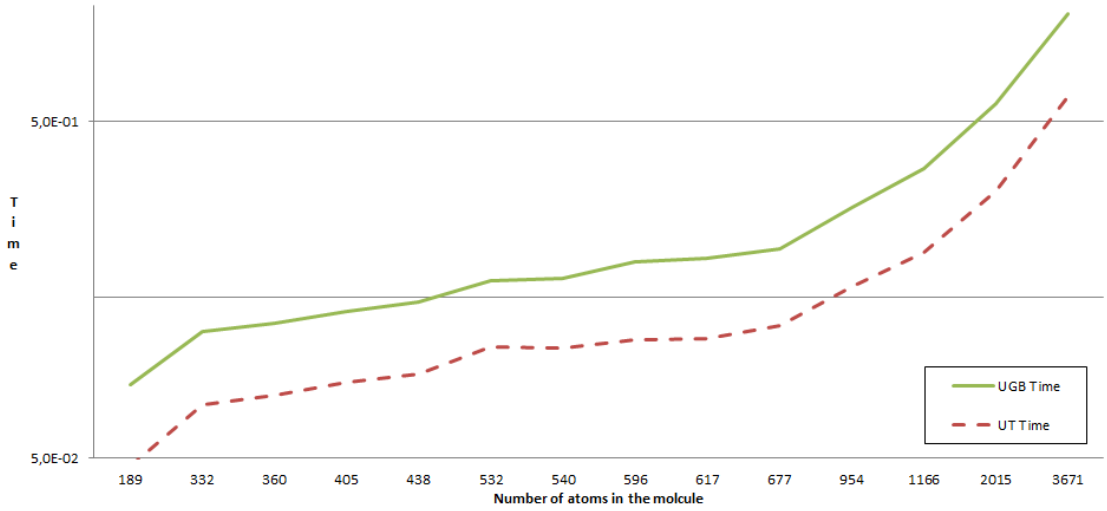


Figure 4: Graphic representation of the time results for the computational tests with proteins from the PDB.

main outlook is the aplication of our numerical method in order to treat noisy distances, i.e., sparse and inexact ones, through a stochastic modeling of the Molecular Distance Geometry Problem (MDGP) by means of a Monte Carlo approach. In addition, we also need to make a numerical complexity analysis and understand better the advantages and limitations of the variable t_j, specially in connection with error estimators aiming to quantify uncertainties in UT, since there are some evidences that it would be suitable.

References

[1] Davis, R., Ernst, C. and Wu, D. Protein structures determination via an efficient geometric build-up algorithm. *BMC Structural Biology*, 10:1–10, 2010.

[2] Q. Dong and Z. Wu. A Geometric Build-up Algorithm for solving the Molecular Distance Geometry Problem. *Journal of Global Optimization*, 26:321–333, 2003.

[3] F. Fidalgo. Algorithms for problems of molecular geometry. Master's thesis, University of Campinas, Campinas, 2011.

[4] J. Saxe. Embeddability of weighted graphs in k - space is strongly NP - Hard. In *Proceedings of 17th Allerton Conference in Communications, Control and Computing, Monticello*, pages 480–489, 1979.

[5] T. Schlick. *Molecular modeling and simulation: an interdisciplinary guide*. Springer, New York, 2002.

[6] D. Wu and Z. Wu. An Updated Geometric Build-up Algorithm for solving the Molecular Distance Geometry Problem with sparse distance data. *Journal of Global Optimization*, 37:661–673, 2007.

DGA 2013, pp. 131 – 135.

A geometric trigraph model for unit disk graph recognition*

Guilherme da Fonseca[1], Vinícius Pereira de Sá[2], Raphael Machado[3] and Celina de Figueiredo[4]

[1]*Universidade Federal do Estado do Rio de Janeiro, Brazil* fonseca@uniriotec.br

[2]*DCC/IM, Universidade Federal do Rio de Janeiro, Brazil* vigusmao@dcc.ufrj.br

[3]*Inmetro — Instituto Nacional de Metrologia, Qualidade e Tecnologia, Brazil* rcmachado@inmetro.gov.br

[4]*COPPE, Universidade Federal do Rio de Janeiro, Brazil* celina@cos.ufrj.br

Abstract A unit disk graph G is a graph whose vertices can be mapped to points on the plane and whose edges are defined by pairs of points within unitary Euclidean distance from one another. The recognition of unit disk graphs is no easy feat. Indeed, the fastest known algorithm to decide whether a given graph is a unit disk graph is doubly exponential. In this paper, we introduce a practical algorithm to produce certified answers to the question "is G a unit disk graph?" in either way, for any given graph G. By imposing that the points' coordinates belong to discrete sets of increasing granularity, our method builds a sequence of trigraphs G', i.e. graphs with *mandatory* and *optional* edges, until either some G' is found possessing properties which certify that G is a unit disk graph, or the sequence of trigraphs has to be interrupted, certifying that G is not a unit disk graph. The proposed method was actually implemented, and we were able to obtain our first certificates for some small graphs.

Keywords: unit disk graphs, graph recognition, trigraphs, geometric algorithms

1. Motivation

A ***unit disk graph*** (UDG) is a graph whose n vertices can be mapped to points on the plane and whose m edges are defined by pairs of points within Euclidean distance at most 1 from one another. Alternatively, one can regard the vertices of a UDG as mapped to coplanar congruent closed disks, so that two vertices are adjacent whenever the corresponding disks intersect. Unit disk graphs have been widely studied in recent times due to their applications to wireless sensor networks [1].

In the present paper, we consider the problem of recognizing unit disk graphs. Though a YES answer can be verified in polynomial time assuming the Real RAM model, the size of certificates comprising the coordinates of the disk centers may not be polynomially bounded under the classic model of computation over finite strings [4]. Indeed, it is not known for the time being whether the problem belongs to NP, and the fastest known recognition algorithm is doubly exponential [5]. Since no practical algorithm is available, there are graphs with as few as ten vertices which have never been proved as being (or not being) UDG [6].

*Research partially supported by FAPERJ and CNPq.

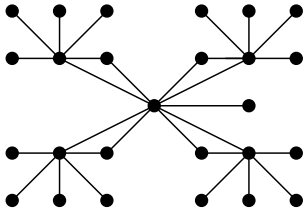


Figure 1: Graph conjectured [6] not to be a UDG.

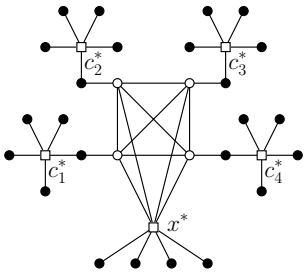


Figure 2: Graph that corresponds to the lower bound for the approximation factor of the algorithm introduced in [2] for minimum (independent) dominating sets in unit disk graphs.

A practical method to certify whether a graph is a UDG is of utmost importance. Indeed, many of the existing bounds for approximation factors of algorithms for hard problems on unit disk graphs are based on the fact that certain graphs are (or are not) UDG, but each one of those graphs demanded their own ad-hoc geometric proof. For an example, [6] conjectures that the graph in Figure 1 is not a UDG. The correctness of their conjecture would imply a decrease from 3.8 to 3.6 in the maximum ratio (except for an additive constant) between the size of a maximal independent set and the size of a connected dominating set in any given UDG, and that would immediately tighten the approximation factor of algorithms that estimate the size of minimum connected dominating sets by computing maximal independent sets.

Another example was obtained in [2]. Denote by $G_{p,q}$ the graph that has one p-clique such that one of its vertices is adjacent to q pendant vertices, and each of the other $p-1$ vertices is adjacent to a degree-2 vertex that in turn is a pendant vertex of an induced $K_{1,5}$. The graph $G_{5,4}$ of Figure 2 is known to be a UDG (a geometric model with only integral coordinates is available [3]) and is the worst known instance for an algorithm that approximates the minimum (independent) dominating set of a unit disk graph, establishing a lower bound of 4.8 for the approximation factor of that algorithm. On the other hand, the graph $G_{9,4}$ is known *not* to be a UDG (the proof is based on numerous geometric lemmas), and this fact is central in the proof of the (upper bound for the) approximation factor of $44/9 = 4.888\ldots$ of such algorithm. Further knowledge about the family $G_{p,q}$, closing the gap between what is currently known to be a UDG (graph $G_{5,4}$) and what is known not to be a UDG (graph $G_{9,4}$), would immediately tighten the existing bounds on the approximation factor of the aforementioned algorithm.

The difficulty in developing a certifier for unit disk graphs, even a "brute force" one, comes from the fact that the solution space — namely $(\mathbb{R}^2)^n$ — is uncountable. In the present paper, we formulate a strategy to reduce the solution space to a countable, finite set, whose granularity is subsequently refined, leading to a YES/NO certificate in many cases. An inconclusive answer, however, may possibly be obtained.

2. The proposed model

The central idea of our strategy is to discretize the solution space by defining an enumerable set of 2-dimensional coordinates where the points associated to the input graphs' vertices may be placed at. For a positive $\epsilon \in \mathbb{R}$, consider the set $N_\epsilon := \{x \in \mathbb{R} \mid x = d\epsilon,\ d \in \mathbb{N}\}$, and let $C_\epsilon := N_\epsilon \times N_\epsilon$ be a discrete set of 2-dimensional coordinates. We call such C_ϵ a ***mesh*** and we say C_{ϵ_1} is *thinner* than C_{ϵ_2} if $\epsilon_1 < \epsilon_2$. Clearly, any subset of points $M_\epsilon \subseteq C_\epsilon$ determines a unit

disk graph G whose vertices are pairwise adjacent whenever their corresponding points in M_ϵ are within unitary distance of one another. We say M_ϵ is an ***ϵ-discrete model*** for G.

Trigraph embodiments. Given a mesh C_ϵ and a set $M_\epsilon \subseteq C_\epsilon$ of n points, we define the trigraph $G_{M_\epsilon} = (V, E_1 \cup E_2)$ as the graph whose vertex set V corresponds to the points in M_ϵ, and whose edges can be partitioned into E_1, the set of ***mandatory*** edges, and E_2, the set of ***optional*** edges. A mandatory edge is associated to a pair of points $v, w \in M_\epsilon$ that are at distance $d(v, w) < 1 - \epsilon\sqrt{2}$ from one another. An optional edge, on its turn, is associated to a pair of points $v, w \in M_\epsilon$ satisfying $1 - \epsilon\sqrt{2} \leq d(v, w) \leq 1 + \epsilon\sqrt{2}$. We say G_{M_ϵ} is a ***trigraph embodiment*** of graph $G(V, E)$ if, and only if, $E \subseteq E_1 \cup E_2$ and $E_1 \setminus E = \emptyset$, i.e. all edges of G are either mandatory or optional edges in G_{M_ϵ}, and no edge that does not belong to G appears as a mandatory edge in G_{M_ϵ}.

If G_{M_ϵ} is a trigraph embodiment of G, and G_{M_ϵ} has no optional edges, then M_ϵ is a unit disk model for G, hence G is certainly a UDG. Moreover, if G_{M_ϵ} does have optional edges, but all optional edges in G_{M_ϵ} correspond to pairs of adjacent vertices in G, then G is a UDG as well. (The same goes for the case where all optional edges in G_{M_ϵ} correspond to pairs of non-adjacent vertices in G.) This is the core of the YES certificates produced by our method.

It can be shown that, if G is a UDG, then G admits a trigraph embodiment G_{M_ϵ}, for all $\epsilon > 0$. Conversely, if, for some ϵ, there is no possible trigraph embodiment G_{M_ϵ} for G, then G is *not* a UDG. Our NO certificates come from this fact.

Our strategy to recognize unit disk graphs can therefore be summarized in the following steps:

INPUT: A connected graph $G = (V, E)$
OUTPUT: YES, if G is a UDG; NO, if it is not a UDG; or INCONCLUSIVE.

1. Choose a value for ϵ and consider the corresponding mesh C_ϵ.
2. For each possible discrete model $M_\epsilon \subseteq C_\epsilon$ with $|M| = |V|$, obtain the respective trigraph $G_{M_\epsilon} = (V, E_1, E_2)$.
   (a) If $E = E_1$ then a disk model was found for G, hence G is a UDG. Return YES.
   (b) If $E \subseteq E_1 \cup E_2$ and $E_1 \setminus E = \emptyset$, then G_{M_ϵ} is a trigraph embodiment for G.
3. If a trigraph embodiment was found for G, then let $\epsilon \leftarrow \epsilon/2$. If ϵ is still greater than some previously defined constant ϵ_{min}, then restart the algorithm with the new value for ϵ; otherwise, return INCONCLUSIVE.
4. If no trigraph embodiment was found for G, then G is not a UDG. Return NO.

Note that, in spite of the apparent infinite number of possible discrete models, we may assume that G is connected[1], so any model of G must be enclosed in a disk of diameter $2|V|$.

Notice also that, whenever the algorithm produces a conclusive answer, then an appropriate certificate has been found. However, as discussed in Section 4, the input graph may not be a UDG, but still be such that, no matter how thin the mesh is, a trigraph model can always be found, leading the algorithm to an inconclusive answer.

3. Results

To validate our proposed model, we implemented it using the Python language. Our implementation includes some nice refinements aimed at reducing the number of candidate placements of each vertex in the considered mesh, such as

[1]Trivially, a graph is a UDG if and only if all its connected components are UDG.

(i) taking the maximum and minimum distances between pairs of vertices as input;

(ii) taking the maximum and minimum angle between two vertices with respect to a third one as input;

(iii) allowing the imposition of a fixed circular order of vertices around a reference point.

Naturally, such features can only be used if some previous geometric analysis determines such distances and angles constraints. With this preliminary implementation, we could already correctly classify some small graphs as being (or not being) UDG.

4. Future directions

In spite of the nice results it has enabled us to obtain, the proposed method does presents some limitations, one of which is disclosed by the following "pathological" example.

Let G be the $K_{1,6}$ graph, which is known (by geometric methods) not to be a UDG. Our procedure is doomed to give an inconclusive answer for G no matter how thin the mesh is. The reason is that, for all $\epsilon > 0$, there is always a trigraph embodiment G_{M_ϵ} for G, in which the center of the star and one of the leaves coincide (see Figures 3, 4 and 5).

A second weakness of the method is its worst-case time complexity, since the time demanded to produce a certificate for certain graphs may be as long as unforeseeable.

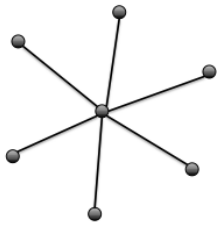


Figure 3: Graph $K_{1,6}$.

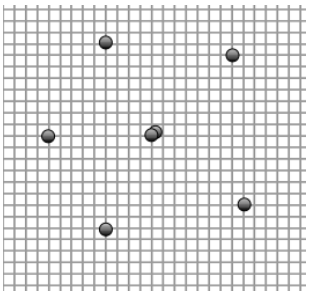


Figure 4: Discrete model for graph $K_{1,6}$. The circles are not the unit-diameter disks themselves, but rather represent their centers. The two overlapping circles represent the centers of coincident disks.

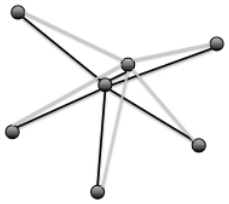


Figure 5: Trigraph corresponding to Figure 4.

The previous observations lead to the following open questions, which are currently under investigation.

1. Is it possible to characterize such "pathological" graphs, those which deny our method any chance of recognizing them in either way?

2. Is it possible to modify our method so that it always stop with a conclusive question within a reasonable, predetermined time?

Notwithstanding the open questions above, there seem to be several promising ways our method can be improved upon. We list some of them below.

- The exhaustive enumeration of possible trigraph embodiments for G can be achieved by a backtracking-based approach. First, a sequence $v_1, \ldots, v_n$ of vertices of G must be determined, in such a way that the subgraph G_k of G induced by $v_1, \ldots, v_k$ is connected for all $k \in \{1, \ldots, n\}$. Each vertex v_k is then positioned, one at a time, at some point of the mesh, in such a way that the set of already occupied points of the mesh (including the one assigned to v_k) defines a trigraph embodiment for G_k. By doing so, the search space for trigraph embodiments for G shall decrease considerably.
- By the end of the k-th iteration of the algorithm, after some trigraph embodiments were found, the value of ϵ is halved, so each former grid point p gives rise to four grid points p_1, p_2, p_3, p_4 to be considered (as possible vertex locations) during the $(k+1)$-th iteration. It shall now be possible to eliminate at once from the list of candidate locations for a vertex v all points p_i corresponding to a point p that was not occupied by v in any trigraph embodiment obtained in the k-th iteration. By so doing, the search for trigraph embodiments on the thiner mesh becomes limited to "refining" previously obtained trigraph embodiments, instead of a search that would otherwise have begun from scratch.
- Proving geometric results such as "if G is a UDG, then G admits a disk model where no two vertices are either vertically aligned, or horizontally aligned, or coincident" may allow for the earlier elimination of a considerable number of discrete models, therefore also speeding up the algorithm.

References

[1] Marathe, M.V., H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz (2005). Simple heuristics for unit disk graphs. *Networks* **25** (2): 59–68.

[2] Fonseca, Guilherme D., Celina M. H. de Figueiredo, Vinícius G. P. de Sá, and Raphael Machado (2012). Linear Time Approximation for Dominating Sets and Independent Dominating Sets in Unit Disk Graphs. *Proc. Workshop on Approximate and Online Algorithms (WAOA 2012)*.

[3] Fonseca, Guilherme D., Celina M. H. de Figueiredo, Vinícius G. P. de Sá, and Raphael Machado (2012). Linear-time sub-5 approximation for dominating sets in unit disk graphs. `http://arxiv.org/abs/1204.3488`.

[4] McDiarmid, Colin, and Tobias Mueller (2013). Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B* **103** (1): 114–143, `http://arxiv.org/abs/1111.2931`

[5] Spinrad, J. (2003). *Efficient Graph Representations*. Fields Inst. monographs. AMS.

[6] Zou, F., Y. Wang, X.-H. Xu, X. Li, H. Du, P. Wan, and W. Wu (2011). New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science* **412** (3): 198–208.

DGA 2013, pp. 137 – 142.

A rotation-invariant image processing operation transformed into the k-nearest neighbours problem

L. R. Foulds, H. A. D. do Nascimento*, H. Longo

Instituto de Informática, Universidade Federal de Goiás, Goiânia–GO, Brasil, {lesfoulds,hadn,longo}@inf.ufg.br.

Keywords: Image processing, non-Euclidean metrics, problem transformation, k-nearest neighbours, approximation algorithms, hierarchical tree decomposition, locality-sensitive hashing.

1. Introduction

Taking keypoints of an object in a reference image and searching for similar keypoints in a collection of candidate images is an important operation in image processing. The number of keypoints in the reference image is usually quite small ($10\,..\,100$), but the number of keypoints among the candidates can be very large ($10^5\,..\,10^7$). Hence it is important to develop efficient identification methods for neighbours of reference keypoints. This is the subject of the present abstract. A recently developed similarity measure (based on dual trees and an oriented complex-valued wavelet transform) has proved to be highly beneficial for multi-dimensional signal processing [10]. This measure has the advantage over a commonly used former measure [9] of the efficient matching of keypoint pairs in a rotationally invariant way. In order to preserve rotational invariance, each reference keypoint must be represented by a set of m cyclic vectors of m dimensions, where m is usually about 200. This multiplicity of vectors means that a non-Euclidean metric must be used to calculate distances between reference and candidate keypoints, which is computationally burdensome for image processing projects of practical size. We explain how this rotationally invariant operation can be transformed into the well-known k-nearest neighbours problem (KNN) with the Euclidean distance metric. This is of interest since there exist fast approximation algorithms for KNN, some of which we describe and discuss.

Let m be the dimension of both the reference and candidate keypoints, both having real-valued elements. Let Y denote the set of initial reference keypoints, with elements denoted by $y^p = (y^p_1, y^p_2, \ldots, y^p_m)^T$, $p \in \{1, 2, \ldots, r\}$, where $r = |Y|$. Let $y^{p,q}$, $q = 1, 2, \ldots, m$; denote the complete set of m keypoints that represent the p^{th} reference, being cyclic versions of y^p:

$$y^{p,q} = (y^p_{s(q,0)}, y^p_{s(q,1)}, \ldots, y^p_{s(q,m-1)})^T, \quad p \in \{1, 2, \ldots, r\}, \tag{1}$$

where, for any $t, u \in \mathbb{Z}^+$, $s(t, u) = t + u$, if $t + u \leq m$ and $= t + u \pmod m$, otherwise.

Let X' denote the set of candidate keypoints, with elements denoted by $x^i = (x^i_1, x^i_2, \ldots, x^i_m)^T$, $i \in \{1, 2, \ldots, n'\}$, where $n' = |X'|$. The metric for calculating the distance between any candidate keypoint $x^i \in X'$ and any initial reference keypoint $y^p \in Y$ is:

$$d'_{ip} = \min_{q \in \{1,2,\ldots,m\}} d(x^i, y^{p,q}), \tag{2}$$

*Hugo do Nascimento is partially sponsored by a Scholarship of Research Productivity from CNPq (309463/2009-2).

where $d(a,b) = ||a-b||_2 = (\sum_{i=1}^{m}(a_i - b_i)^2)^{\frac{1}{2}}$, the ℓ_2 norm (Euclidean distance) between a and b.

Let $Y' = \{y^{p,q} \mid p = 1,2,\ldots,r;\ q = 1,2,\ldots,m\}$. Suppose X' and Y are given and Y' has been constructed from Y. The original image processing operation is denoted by $P_k(X',Y')$. For given positive integer $k \leq n'$, $P_k(X',Y')$ involves finding the k nearest neighbours in X' for each reference keypoint in Y according to the metric d'_{ip}.

The main contribution of the present abstract is to suggest that $P_k(X',Y')$ can be solved by formulating it as KNN: given a collection of candidate keypoints, build a data structure which, given any reference keypoint, reports the k candidate keypoints that are closest to the reference keypoint, with all data points being in a given metric space [8]. The metric in the space need not necessarily be Euclidean distance although it is the one commonly used and it is used here from now on. KNN is of major importance in similarity searching and has signification application in many areas including: image processing, statistical measure estimation, machine learning, data mining, data compression, information retrieval and pattern recognition.

We now explain a transformation that enables $P_k(X',Y')$ to be performed by any KNN algorithm with Euclidean distance. The basic idea is to construct a set of "candidate keypoints", based on cyclic versions of each candidate keypoint, in the same way that Y' was constructed from Y. The motivation for this is that (2) is an awkward metric to evaluate, as it involves m reference keypoints as well as a candidate keypoint. To avoid this, we deal with Y, the set of initial reference keypoints rather than with Y'. We augment X' by element cycling. The augmentation means that each distance calculation involves just one initial reference keypoint and one candidate keypoint. Let $x^{i,j}$, $j = 1,2,\ldots,m$; denote the m keypoints that are constructed from x^i, $i \in \{1,2,\ldots,n'\}$, for the augmentation. The $x^{i,j}$'s are cyclic versions of x^i and are defined in the same vein as (1). That is, let $x^{i,j} = (x^i_{s(j,0)}, x^i_{s(j,1)}, \ldots, x^i_{s(j,m-1)})^T$, $j = 1,2,\ldots,m$, with s defined as in (1).

Let $X = \{x^{1,1}, x^{1,2}, \ldots, x^{1,m}, x^{2,1}, \ldots, x^{2,m}, \ldots, x^{n',1}, \ldots, x^{n',m}\}$ denote the augmented set of keypoints constructed from X'. The metric for calculating the distance between any candidate keypoint $x^{i,j} \in X$ and any reference keypoint $y^p \in Y$ is:

$$d_{ijp} = d(x^{i,j}, y^p),\ i \in \{1,2,\ldots,n'\}; \quad p \in \{1,2,\ldots,r\}, \tag{3}$$

where $d(a,b)$, once again, denotes the ℓ_2 norm, the Euclidean distance between vectors a and b.

As usual, when minimising relative distance, the "1/2" power of Euclidean distance can be neglected. Calculating d_{ijp} requires $\mathcal{O}(m^2)$ time. (The m distances computed each require approximately $2{\cdot}m$ additions/subtractions and m multiplications and the sort of the distances requires $\mathcal{O}(m{\cdot}\log m)$ time.) Furthermore, the fact that the candidate keypoints are rotated leads to additional savings in computation.

The newly transformed image processing operation is denoted by $P_k(X,Y)$. It involves finding the k nearest neighbours in X for element in Y according to the metric d_{ijp}. Note that the neighbours returned must arise from distinct elements of X.

2. Problem Transformation

The transformation $P_k(X,Y)$ is the k-nearest neighbours problem (KNN). We show that $P_k(X',Y')$ and $P_k(X,Y)$ are equivalent.

Lemma 1. *If x is a nearest neighbour of some $y^p \in Y'$ for $P_1(X',Y')$, then solving problem $P_1(X,Y)$ will produce a nearest neighbour that is the same distance from y^p as x.*

Proof. Suppose for some $i \in \{1,2,\ldots,n'\}$, $x^i = (x^i_1, x^i_2, \ldots, x^i_m)^T \in X'$ is a nearest neighbour of Y' produced by $P_1(X',Y')$. Suppose further, for some $q \in \{1,2,\ldots,m\}$, $y^{p,q} = (y^p_q, y^p_{q+1},$

$\ldots, y^p_m, y^p_1, \ldots, y^p_{q-2}, y^p_{q-1})^T \in Y'$ induces the minimisation in (2). That is,

$$d(x^i, y^{p,q}) = ((x^i_1 - y^p_q)^2 + \cdots + (x^i_{m-q+1} - y^p_m)^2 + (x^i_{m-q+2} - y^p_1)^2 + \cdots + (x^i_m - y^p_{q-1})^2)^{\frac{1}{2}}. \quad (4)$$

The expression in (6) can be rearranged so the y values appear in the order $y^p_1, y^p_2, \ldots, y^p_m$:

$$d(x^i, y^{p,q}) = ((x^i_{m-q+2} - y^p_1)^2 + (x^i_{m-q+3} - y^p_2)^2 + \cdots + (x^i_{m-q+1} - y^p_m)^2)^{\frac{1}{2}}. \quad (5)$$

Let $j = m - q + 2$ and $x^{i,j} = (x^i_{m-q+2}, x^i_{m-q+3}, \ldots, x^i_m, x^i_1, x^i_2, \ldots, x^i_{m-q+1})^T$. Then $x^{i,j} \in X$ is a neighbour of y^p that is distance $d(x^i, y^{p,q}) = d(x^{i,j}, y^p)$ from y^p. Thus, $x^{i,j}$ is a nearest neighbour of y^p that will be produced by solving $P_1(X, Y)$. □

Lemma 2. *If x is a nearest neighbour of some $y^p \in Y$ for $P_1(X, Y)$, then solving $P_1(X', Y')$ will produce a nearest neighbour that is the same distance from y^p as x.*

Proof. Suppose for some $i \in \{1, 2, \ldots, n'\}$ and $j \in \{1, 2, \ldots, m\}$, $x^{i,j} = (x^i_j, x^i_{j+1}, \ldots, x^i_m, x^i_1, \ldots, x^i_{j-2}, x^i_{j-1})^T$ is a nearest neighbour of y^p found by solving $P_1(X, Y)$. That is,

$$d(x^{i,j}, y^p) = ((x^i_j - y^p_1)^2 + \cdots + (x^i_m - y^p_{m-j+1})^2 + (x^i_1 - y^p_{m-j+2})^2 + \cdots + (x^i_{j-1} - y^p_m)^2)^{\frac{1}{2}}. \quad (6)$$

The expression in (6) can be rearranged so the x values appear in the order $x^i_1, x^i_2, \ldots, x^i_m$:

$$d(x^i, y^{p,q}) = ((x^i_1 - y^p_{m-j+2})^2 + (x^i_2 - y^p_{m-j+3})^2 + \cdots + (x^i_m - y^p_{m-j+1})^2)^{\frac{1}{2}}. \quad (7)$$

Let $q = m - j + 2$ and $y^{p,q} = (y^p_{m-q+2}, y^p_{m-q+3}, \ldots, y^p_m, y^p_1, y^p_2, \ldots, y^p_{m-q+1})^T$. Then $x^i \in X'$ is a neighbour of $y^{p,q} \in Y'$ that is distance $d(x^{i,j}, y^p) = d(x^i, y^{p,q})$ from y^p. Furthermore, x^j is a nearest neighbour of y^p that will be produced by solving problem $P_1(X', Y')$. □

Theorem 3. *$P_k(X', Y')$ and $P_k(X, Y)$ are equivalent.*

Proof. In order to find the k nearest neighbours (for $k > 1$), the procedures in Lemmas 1 and 2 can be repeated k times as follows. Whenever $x^{i,j} \in X$ is established as a nearest neighbour, the keypoints $x^{i,j}$, for all $j \in \{1, 2, \ldots, m\}$, are removed from X. Then the procedures to find a new nearest neighbour are repeated for the set $X \backslash \{x^{i,j} \mid j = 1, 2, \ldots, m\}$. This avoids the alias problem of two cyclic version of a vector in X being accidentally identified as two separate nearest neighbours. Thus, the 1-neighbour (closest neighbour) procedure is performed k times. The above arguments can be repeated for each $p \in \{1, 2, \ldots, r\}$. □

3. Algorithmic Solutions

Note that X is independent of r, the number of reference keypoints. Once X has been created, it remains fixed for all future reference keypoints and its construction cost can be amortised over Y. This will be advantageous whenever r is relatively large and new reference keypoints are inserted. One way to perform $P(X, Y)$ is by "brute force", using the following exhaustive search algorithm: The time complexity of $ES(X, Y)$ is $\mathcal{O}(m{\cdot}n{\cdot}r + n{\cdot}r{\cdot}\log n)$, where $n = m{\cdot}n' = |X|$

ES$(\mathbf{X}, \mathbf{Y})$: Create the augmented candidate keypoint set X'. For each reference keypoint $p \in Y$: Calculate d_{ijp} for all $i \in \{1, 2, \ldots, n'\}$ and $j \in \{1, 2, \ldots, m\}$; Sort the distances just calculated; Select the first k distinct candidate keypoints in X based on these distances.

---

is the number of candidate keypoints. Due to the special structure of X' the algorithm can be

speeded up by parallel computing. We shall discuss data structures that can reduce this huge time complexity.

When m is relatively small, the metric space (X, d_{ijp}) can be fruitfully partitioned by using k-d trees [3, 6] in order to compute distances only within specific nearby volumes. However, the performance of k-d tree-based algorithms declines as m increases. If the metric is non-Euclidean, or if m is relatively large, ball trees [12] often provide more useful results in practical situations [5]. Although distance sorting is not an issue with the hierarchical tree decomposition algorithms discussed so far, the fact that their space requirements are exponential in m is a major concern. Cover trees have been developed to address this difficulty and to enable fast approximate KNN searches. Indeed, cover tree-based algorithms use implicit representation to keep track of repeated points and thus require only $\mathcal{O}(n)$ space, independent of any assumptions regarding m [2].

Like the other hierarchical tree decomposition algorithms mentioned, cover trees allow for KNN searches in $\mathcal{O}(b{\cdot}\log n)$ time where b is a constant derived from m. However, when m is relatively large, b is of significant size and must be taken into account in complexity analysis, implying that performance declines with increasing m. But cover tree algorithms are unique among tree-based methods in that a theoretical bound on b is available. This bound is c^{12}, where c is an expansion constant for exact algorithms and a doubling constant for approximation algorithms [2], leading to a bound on search time of $\mathcal{O}(c^{12}{\cdot}\log n)$. Although cover trees provide reasonably fast KNN searches, the speed comes with the additional cost of maintaining the data structure. In exhaustive search, the time to add a new point to the dataset can be neglected because order does not need to be preserved, but in a cover tree it can take up to $\mathcal{O}(c^6{\cdot}\log n)$ time. Samet [11] provided a survey of hierarchical tree decomposition algorithms for KNN.

Practical data for $P_k(X', Y')$ often has dimensions: $m \approx 200$, $n \approx 10^5 \,..\, 10^7$, $k \approx 50$ and $r \approx 10 \,..\, 100$. Clearly, the strategy of transforming the image processing operation and solving $P_k(X, Y)$ with such dimensions will be computationally effective only if approximation algorithms are used, possibly in conjunction with parallel computing. "Approximation" in the context of KNN implies, given $y \in Y$ and an approximation parameter $\varepsilon > 0$, find elements $x_1, x_2, \ldots, x_k \in X$ such that $d(x_i, y) \leq (1+\varepsilon){\cdot}d(X, y)$, $i \in \{1, 2, \ldots, k\}$ and x_i is the i^{th} nearest neighbour of y. The notion of approximation is appealing here as it has been found for many practical datasets that the approximately nearest neighbours identified are very close to the exact ones and the differences are often unimportant [4]. Tree cover approximation algorithms show promise in this regard. Indeed, Beygelzimer et al. [2] provide such an algorithm that is of practical interest. It has a time requirement at most $c^{12} \cdot \log \Delta + (1/\varepsilon)^{\mathcal{O}(\log c)}$, where c is the doubling constant and Δ is the aspect ratio (the ratio of the largest to the smallest interpoint distance). The space bound is $\mathcal{O}(n)$, which is independent of c.

Locality-sensitive hashing (LSH) is another suitable KNN approximation method where m is probabilistically reduced when it is relatively large [7]. The basic idea is to hash the input items using several hash functions so that similar items are mapped to the same buckets with much higher probability than for dissimilar items (the number of buckets being significantly smaller than n). When the reference keypoints are included, one can then determine their near neighbours by hashing a reference keypoint and retrieving the elements in its buckets.

Gionis et al. [4] and Andoni and Indyk [1] have developed fast LSH approximation algorithms that can be used to solve $P_k(X, Y)$ when m and n are both relatively large. Their methods are based on LSH families that are simple, easy to use and can accommodate the situation where new reference keypoints are inserted dynamically. The Gionis et al. algorithm is $\mathcal{O}(m{\cdot}d^{1/(1+\varepsilon)})$. The authors solved particular KNN problems with $n = 270,000$, $m = 64$ and $k = 10$, in better than an order of magnitude faster than tree-based algorithms, requiring $\mathcal{O}(n)$ space and with less than 4% error. Andoni and Indyk [1] describe an LSH-based algorithm for m-dimensional

Euclidean space that is provably near-optimal in the class of the LSH algorithms regarding the separation of collision probabilities of close and far points.

4. Computing

Suppose for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, $x^{i,j} = (x^i_{j(1)}, x^i_{j(2)}, \ldots, x^i_{j(m)})^T$ is one of the candidate keypoints and for some $p \in \{1, 2, \ldots, m\}$, one of the reference keypoints is represented by the vector $y^p = (y^p_1, y^p_2, y^p_3, \ldots, y^p_{m-1}, y^p_m)^T$. Suppose further, that for a given non-negative finite real number ε:

$$|x^i_{j(q)} - y^p_q| \leq \varepsilon, \; \forall \, q = 1, 2, \ldots, m. \tag{8}$$

This relationship implies $d_{ijp} \leq \sqrt{m} \cdot \varepsilon$. Rather than performing all the m calculations necessary to calculate d_{ijp}, one might instead perform element-to-element comparisons of $x^{i,j}$ and y^p based on (8). During the comparison process, whenever (8) does not hold $x^{i,j}$ could be eliminated from further consideration as a potential k-nearest neighbour of y^p. As with the doubling constant for cover trees, the bound ε could be increased until k candidate keypoints have been identified.

5. Conclusion

The transformation introduced above may be viewed as a means of linking the two classes of k-nearest neighbours problems, so that theoretical results for $P(X, Y)$ (for which relatively fast approximation algorithms exist) can be extended to $P(X', Y')$. No claims are made, however, as to the computational utility of this transformation. The authors are in the process of investigating its usefulness for rotation-invariant image processing operations of practical size via approximation algorithms and parallel computing.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–128, 2008.

[2] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104, 2006. `http://hunch.net/~jl/projects/cover_tree`.

[3] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(4):209–226, 1977.

[4] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th Very Large Database (VLDB) Conference*, Edinburgh, UK, 1999.

[5] A. Gray and A. Moore. n-body problems in statistical learning. *Advances in Neural Information Processing Systems (NIPS)*, 13:266–272, 2000.

[6] P. Indyk. Nearest neighbors in high-dimensional spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 39. CRC Press, London, UK, 2004.

[7] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of 30th Symposium on Theory of Computing*, pages 604–613, 1998.

[8] L. Jiang, Z. Cai, D. Wang, and S. Jiang. Survey of improving k-nearest-neighbor for classification. In *Fourth International Conference on Fuzzy Systems and Knowledge*, pages 679–683, 2007.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] J. Nelson and N. G. Kingsbury. Enhanced shift and scale tolerance for rotation-invariant polar matching with dual-tree wavelets. *IEEE Transactions on Image Processing*, 20(3):814–821, 2011.

[11] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Elsevier, Holland, 2006.

[12] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40:175–179, 1991.

DGA 2013, pp. 143 – 148.

Using Correspondence Analysis And its Distance To Evaluate The Components of A Naming Test For Studying Aphasia

Gastão Coelho Gomes,[1] Sergio Camiz,[2] Christina Abreu Gomes[3] and Fernanda Duarte Senna[4]

[1]*DME– IM– UFRJ, Brazil* gastao@im.ufrj.br

[2]*Dipartimento di Matematica–Sapienza Università di Roma, Italia* sergio.camiz@uniroma1.it

[3]*Departamento de Linguística, UFRJ, Cidade Universitária, Brazil* christina-gomes@uol.com.br

[4]*Doutoranda Programa de Pós-Graduação de Linguística, UFRJ, Brazil* fonofernandasenna@gmail.com

Abstract Exploratory Multidimensional Data Analyses were used to manage two components of a naming test for studying lexical access in aphasic patients, i.e., the naming agreement of images and age of acquisition of the names themselves from an original international test. In order to be reliable the images should be easily and unequivocally named by any subject using the same word. Theoretical assumptions about word learning states that words acquired later tend to be the first to be lost due to brain damage in aphasia. Thus, these two variables are important predictors of the patient's word retrieval. We first selected the images according to normal judges recognition agreement; then, to range them based on their primitiveness, these images were submitted to two sets of judges, that had to answer according to two different scales. Data were analyzed with several exploratory multidimensional techniques, including Simple and Multiple Correspondence Analyses, Principal Component and Multiple Factor Analyses. A comparison suggested that no mayor differences existed due to the two scales' differences.

Keywords: Chi-square Distance, Correspondence Analysis, Factor Analysis, Linguistics, Aphasia

1. Introduction

This study is the continuation of a previous one [4] and concerns the evaluation of a set of 260 images [8] internationally used to test the lexical access in aphasia, i.e. the loss of some abilities related to language production and/or comprehension due to brain damage. The test, that aims at measuring to what extent the disease affects the word retrieval, is based on the recognition of familiar objects submitted as images to the patients and their consequent verbalization. In order to be reliable, we considered that a selection of these images ought to be done to suit the Brazilian reality and we based it on two criteria: *i)* the images should be easily and unequivocally recognizable, and *ii)* the *primitiveness* of the word, say its age of acquisition, should be measured. Indeed, it is theoretically assumed (based on word learning) that words acquired later tend to be the first to be lost in aphasic patients affected by brain damage. Thus, these two characters are important predictors of the patient's word retrieval. Thus, we selected randomly three groups of non-affected people to act as judges, and asked one to identify the images and the other two to estimate the degree of primitivity of the corresponding names. As

our study was based mainly both on judges and scale evaluation, we show in the following how we dealt with the judges' reliability and the scale definition.

2. The data

- For the selection of the images, all of them (260) were submitted to a panel of 38 judges randomly selected among non-affected people. The answers have been coded as 1 = recognized, 0 = not recognized. From this selection, 161 images resulted.

- To measure primitiveness, we asked 128 non-affected judges to estimate how primitive were the 161 represented objects, according to their personal experience. This estimation was based on two different scales: *i)* the first panel, with 60 judges, labelled *E*, has been asked to measure the age of acquisition on a scale from 1 to 7 according to how early in their life each word was first known, but without specifically mentioning the age; here 1 corresponds to very early in life and 7 to most late; *ii)* the second panel, with 68 subjects, labelled *I*, has been asked a measure based on a scale 1-7 as well, but this time based on age classes: the classes are: 1=0-2 years, 2=2-4 years, 3=4-6 years, 4=6-8 years, 5=8-10 years, 6=10-12 years and 7=13 and further.

3. Theoretical Framework

The consistency of judges is of high importance in several frameworks, as in sensorial analysis. For this task, specific estimation methods have been developed (see, e.g. [7]). Here, we preferred to consider the problem on another point of view, as no objective primitiveness may be measured, but only identify a central tendence stastistics. Thus, we only removed those judges whose results appeared clearly far from all others. For what concerns the scale definition, we tried to compare two possible scales: a free one and one based on age intervals, and we studied their agreement. Thus, we considered of interest to use for our study *exploratory multidimensional analysis methods*, since their graphical representations allowed a visual inspection of most of the questions that we might ask.

An interesting feature of the analyses that we adopted is that they are all based on the same principle, the *Singular Value Decomposition* (*SVD*, [2]) of some transformation of the original data matrix $T : X \to A = T(X)$. The *SVD* of a matrix A is given by $A = U\Lambda^{1/2}V'$, with U and V the matrices of the (vertical) eigenvectors of $A'A$ and AA' respectively, and Λ the (diagonal) matrix of their corresponding eigenvalues, sorted in descending order. The theorem states the highest importance, in terms of represented inertia, of the first generated axes in respect to the following ones.

According to the data at hand, the analyses have been submitted to *Simple Correspondence Analysis* (*SCA*, [1], [5]) to identify both judges and items with critical recognition behavior, and *Multiple Correspondence Analysis* (*MCA, ibid.*) to identify those judges with biased evaluation of primitiveness in respect to others. *Multiple Factor Analysis* (*MFA*, [3]) has been used to compare the primitiveness of the words given by the two panels of judges according to the two different scales, and eventually *Principal Component Analysis* (*PCA, ibid.*, see also [6]) has been used to define the primitivity index of our interest.

The data transformations, according to the different methods may be described as follows:

PCA $x_{ij} \to z_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{n}\sigma_j}$ standardization

MFA $x_{ij_k} \to z_{ij_k} = \frac{x_{ij_k} - \overline{x}_{j_k}}{\sqrt{\lambda_k^1}\sqrt{n}\sigma_{j_k}}$ std. adjusted to group's coherence

SCA $x_{ij} \rightarrow s_{ij} = \frac{x_{ij}}{\sqrt{x_{i.}x_{.j}}} - \frac{\sqrt{x_{i.}x_{.j}}}{x_{..}}$ deviation from independence

MCA $x_{ij_q} \rightarrow s_{ij_q} = \frac{1}{\sqrt{Q}}\left(\frac{x_{ij_q}}{\sqrt{x_{i.}}} - \frac{\sqrt{x_{i.}}}{x_{..}}\right)$ deviation from average profile

4. Selecting Images

The results of first submission reported 10 images that no judge could identify, so that we withdrew them immediately. On the other side, 66 images have been recognized by all judges, thus automatically included. Therefore, we applied *SCA* to the remaining images to get a graphical representation of the pattern of both judges and images on factor planes. According to Figure 1(a) below, six judges, P13, P17, P25, P32, P34, and P36, appear further from the origin than all others, whose central pattern seems homogeneous, thus they have been withdrawn.

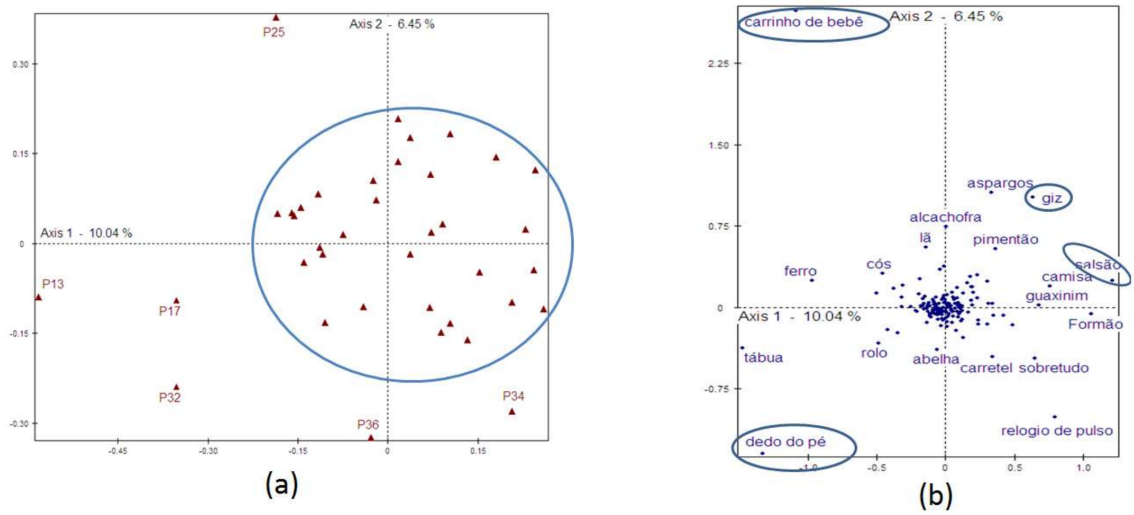


Figure 1: Analysis for the selection of the images. The items on the first factor plane of SCA: (a) The judges, (b) The names.

As well, some items, such as *baby stroller, toe, celery,* and *chalk*, were identified by no more than 5 judges. They are located at the border of the cloud as can be seen on Figure 1(b) above. We re-ran *SCA* with only 32 judges and also all the items whose frequency of correct identification was lower than 50%. From the homogeneous results we could conclude that no further removal of judges seemed necessary. Eventually, we decided to keep all the images that were correctly identified by at least 90% of judges. Based on 32 judges: 97 images were identified by all of them, 26 by only 31 (97%), 24 by 30 (94%), 14 by 29 judges (91%), summing up to 161 images.

5. Defining word primitiveness

In order to examine first the homogeneity of the judges, we started by running *MCA*s on each of two tables. Their behavior is represented by a trajectory that connects the seven levels of the scale. Observing the two graphics in Figure 2, one may observe that the trajectories of the judges that measured freely (Figure 2(a)) are much longer than those of judges based on age (Figure 2(b)). This may be explained by a reduced use of the first levels by the latter.

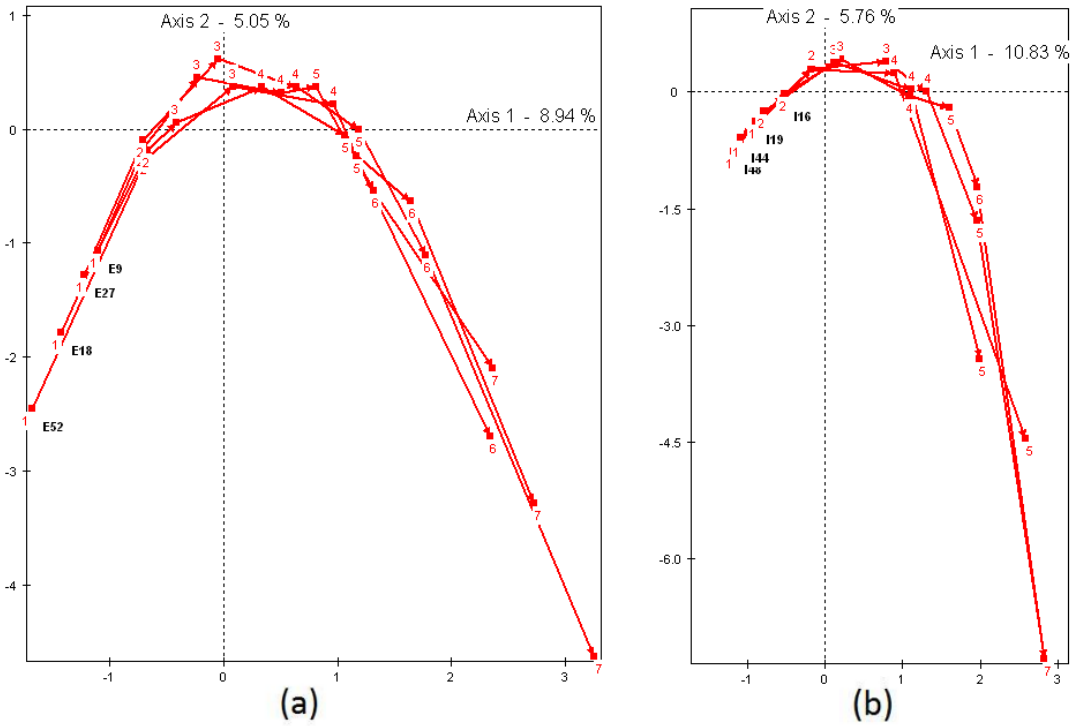


Figure 2: Analysis of the primitiveness judgements. The judges' trajectories represented on the first factor plane of *MCA*: (a) free judgements, (b) judgements based on age intervals.

The pattern of trajectories on both tables on the first factor plane is very homogenous among both sets of judges: Only five of them (*E12, E23, E59, I2*, and *I58*) showed very strange trajectories (see them in Figure 3), thus were removed.

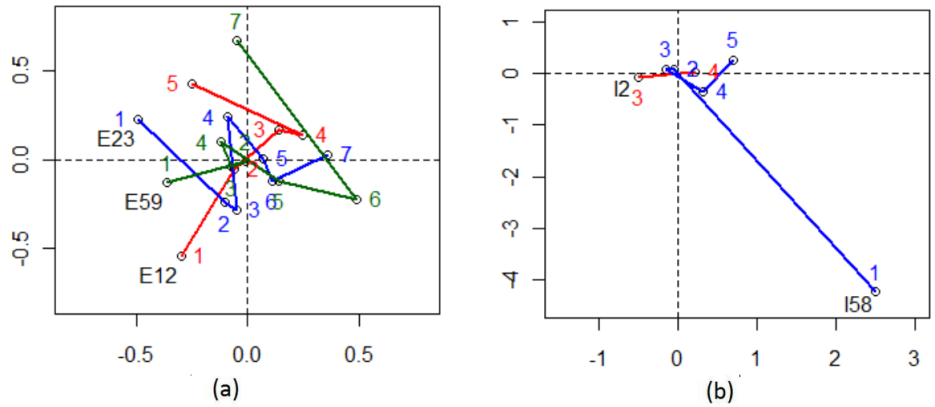


Figure 3: Analysis of the primitiveness judgements. The outlier judges' trajectories represented on the first factor plane of the respective *MCA*: (a) free judgements, (b) judgements based on age intervals.

Then, we ran a *MFA*, considering the two groups of reduced judges (57 that used the free scale (E) and 66 with age-scale (I). A specific advantage of *MFA* in respect to *PCA* it its ability to represent on factor planes not only the global units, but also the partial ones, that is, in our case, the projection of the words seen by either group of judges. Indeed, the total word is situated on the centroid of the two partial words. Therefore, distances between partial words

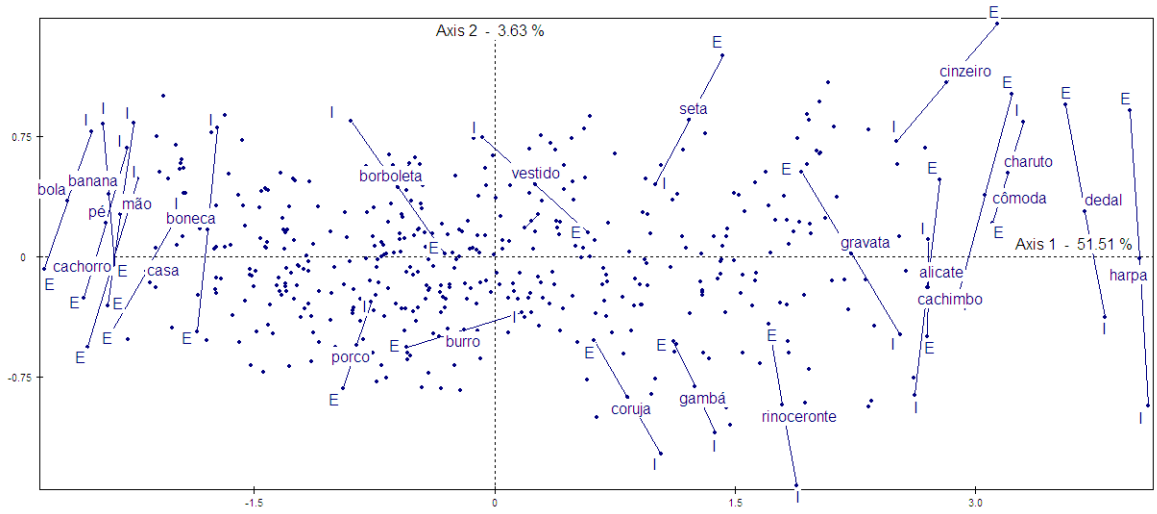


Figure 4: Analysis of the age of acquisition judgements. All the words represented on the plane spanned by the first two factors of *MFA*. Only the words with the largest trajectories are labelled, with the word (the compromise) and either *E* or *I* the partial ones.

are a measure of their dissimilarity according to the two sets of measurements and they may be decomposed according to the different axes. The words with highest negative differences along the first factor are *burro, gravata, lâmpada, mala*, and *patins* and those with highest positive ones are *borboleta, cigarro, cinzeiro, escada, galinha, ônibus*, and *vestido*. Thus, the first might be words judged more primitive by the free-scale judges, whereas the second might be judged more primitive by the age-scale ones. Here, we deal only with the first axis that clearly represents primitivity of words (51.51% of total inertia), since the following explain too little inertia to deserve being taken into account (the second only 3.64%). In Figure 4 all words are represented both totally and partially, with the total units at the centroid of the respective partials. Looking at the extreme of the first axis it is interesting to find the words with the largest differences on the second axis and in particular a reverse behaviour: this reflects the small rotation of the first factors of partial tables, but does not deserve a true interest for our purposes.

As the partial first factors of the two tables where most correlated among each other (.98) and with the *MFA* one (over .99), we decided to merge the two data sets , so that as measure of the words' primitivity was taken the first principal component of this unified table's *PCA*.

6. Conclusions

The study aiming at both selecting images with high naming agreement and measuring the degree of primitiveness of their correspondent words, has been carried out using only exploratory multidimensional data analyses. This allowed to withdraw judges with a clearly biased behavior in respect with the others and select a set of words that have been recognized by nearly the totality of judges. The free scale resulted a little better performing than the other, since it allowed a more instinctive estimate.

References

[1] Benzécri J.P. and coll. (1973–82), *L'analyse des données*, 2 voll., Paris, Dunod.

[2] Eckart C. and G. Young (1936), The approximation of one matrix by another of lower rank, *Psychometrika*, Vol. 1: pp. 211–218.

[3] Escofier B. and J. Pagés (1998). *Analyses factorielles simples et multiples*, 3e ed., Paris, Dunod.

[4] Camiz S., G.C. Gomes, F.D. Senna, and C.A. Gomes (2010). Correspondence Analysis in a Study of Aphasic Patients, XLII SBPO 2010, Bento Gonçalves (RS) Brazil, 30/8-3/9. http://www.sobrapo.org.br/sbpo2010/xliisbpo_pdf/72251.pdf

[5] Greenacre M. (1983), *Theory and Applications of Correspondence Analysis*, London, Academic Press.

[6] Jolliffe I.T. (2002), *Principal Components Analysis*, Berlin, Springer.

[7] Rust R.T. and B. Cooil (1994), Reliabilty Measures for Qualitative Data: Theory and Implications. *Journal of Marketing Research*, Vol. 31: pp. 1–14.

[8] Snodgrass J.G. and M. Vanderwart (1980), A standardized set of 260 pictures: Norms for name agreement, familiarity and visual complexity. *Journal of Experimental Psychology: Human Learning & Memory*, Vol. 6(2): pp. 174–215.

DGA 2013, pp. 149 – 152.

A new algorithm to finding discretizable orderings for Distance Geometry

Warley Gramacho[1], Douglas Gonçalves[2], Antonio Mucherino[2] and Nelson Maculan[3]

[1]*Federal University of Tocantins, Palmas-TO,Brazil,* wgramacho@uft.edu.br

[2]*IRISA, University of Rennes 1, Rennes, France,* { douglas.goncalves,antonio.mucherino } @irisa.fr

[3]*COPPE, Federal University of Rio de Janeiro, Rio de Janeiro-RJ,Brazil,* maculan@cos.ufrj.br

Abstract We present a new algorithm to finding suitable orderings for instances of the Distance Geometry Problem (DGP) that can allow for discretization. We present some preliminary computational results showing that the new algorithm outperforms a previously proposed one.

Keywords: Distance Geometry, Discretizing Vertex Order Problem, Sensor Network Localization

1. Introduction

Given an integer $K > 0$ and a weighted undirected graph $G = (V, E, d)$, with $d : E \longrightarrow \mathbb{R}+$, the Distance Geometry Problem (DGP) asks whether there exists a function $x : V \longrightarrow \mathbb{R}^K$ such that:

$$\forall (u, v) \in E \qquad ||x_u - x_v|| = d(u, v),$$

where $x_u = x(u)$ and $x_v = x(v)$ [6].

The DGP is usually formulated as a continuous optimization problem, but, under some assumptions, it can be formulated as a combinatorial problem.

Definition 1.1. *The Discretizable Distance Geometry Problem (DDGP) [7].*
Let $G = (V, E, d)$ be a weighted undirected graph associated to a DGP instance. Let us suppose that there is a partial order relation on the vertices of V. The DDGP in dimension K consists in all the DGP instances satisfying the following two assumptions:

Assumption 1: *there exists a subset V_1 of V such that*

- $|V_1| = K$;
- V_1 *is a clique;*
- *the order relation on V_1 is total;*
- $\forall v_0 \in V_1 \quad \forall v \in V \setminus V_1, \quad v_0 < v$.

Assumption 2: $\forall v \in V \setminus V_1, \exists u_1, u_2, \ldots, u_K \in V$ *such that:*

- $u_1 < v, u_2 < v, \ldots, u_K < v$;
- $\{(u_1, v), (u_2, v), \ldots, (u_K, v)\} \in E$;

- *the Cayley-Menger determinant of the distance matrix related to* $\{u_1, u_2, \ldots, u_K\}$ *is not* 0.

We say that an ordering for the vertices of V is a discretizing ordering if it makes the assumptions of the DDGP satisfied.

DDGP instances can be solved by employing a branch-and-prune (BP) algorithm [5] that is potentially able to enumerate the whole solution set. This is a major difference between the BP algorithm and other algorithms for the DGP. However, in order to apply BP, the DDGP assumptions have to be satisfied. Finding an order for the vertices in V such that these assumptions are satisfied represents an important pre-processing step for the solution of DDGPs [7]. We refer to this problem as the Discretizing Vertex Order Problem (DVOP) [4].

The rest of the paper is organized as follows. In Section 2, we introduce the DVOP and present the new algorithm. Some preliminary computational results are presented in Section 3.

2. Suitable orderings for the DDGP

Let $G = (V, E, d)$ be a weighted undirected graph related to a DGP instance, and let us suppose that a total order is associated to the vertices in V (it is known that, from any partial order on V, a total order can be derived). For referring to an order, we will consider the usual symbol $<$, and we will add subscripts when it will be necessary to distinguish among different orders (e.g. $<_1$ or $<_2$). Similarly, the symbol $(u, v)_{<_1}$ will refer to the arc involving the vertices u and v in the order $<_1$. We will refer to an order $<$ for which the assumptions in Def. 1.1 are satisfied in dimension K as a DDGP K-order.

Let $\alpha_<(v)$ be, for $v \in V$, the number of adjacent predecessors of v in the order $<$, that is:

$$\alpha_<(v) = card\{u \in V : (u, v)_< \in E\}.$$

Equivalently, let $\beta_<(v)$, for $v \in V$, be the number of adjacent successors of v, in the order $<$:

$$\beta_<(v) = card\{u \in V : (v, u)_< \in E\}.$$

Definition 2.1. *The Discretizing Vertex Order Problem (DVOP).*
Given an undirected graph $G = (V, E)$ and a positive integer K, establish whether there is an order $<$ on V such that: (a) the first vertices in the order form a K-clique, and (b) for each $v \in V$, $\alpha_<(v) \geq K$.

We observe that the DVOP does not verify whether the order satisfies the assumption on Cayley-Menger determinant given in Def. 1.1. This is because the set of distance matrices yielding Cayley-Menger determinant having value exactly zero has Lebesgue measure zero within the set of all possible (real) distance matrices [4]. The probability for this to happen is therefore 0 in a mathematical sense. The NP-completeness of the DVOP follows from NP-completeness of the K-clique problem, because finding a DDGP K-order implies finding K vertices forming a clique in G. When K is fixed, however, as in real applications, the DVOP can be solved in polynomial time [4].

Proposition 2.1. *Given a weighted undirected graph $G = (V, E, d)$ and an order $<$ on V, there do not exist DDGP K-orders if some vertex has degree less than K.*

Note that Prop. 2.1 cannot be inverted, i.e. there can exist instances that do not admit any DDGP K-order even if, for all $v \in V$, $\alpha_<(v) + \beta_<(v) \geq K$.

2.1 The new algorithm

Let us consider that an order $<_1$ for the vertices in G is already available. We suppose that this order is not a DDGP K-order, and, for each $v \in V$, $\alpha_{<_1}(v) + \beta_{<_1}(v) \geq K$, for guaranteeing

that such an order may exist. The basic idea behind this algorithm is to select all v for which $\alpha_{<_1}(v) < K$, and to modify their position so that, in the new order $<_2$, we have $\alpha_{<_2}(v) = K$ and $\beta_{<_2}(v) = \beta_{<_1}(v) + \alpha_{<_1}(v) - K$.

By considering the order $<_1$, let us suppose that v' is such that $\alpha{<_1}(v') < K$. Let $h = K - \alpha_{<_1}(v')$, and $\Xi = \{u \in V : (v', u)_{<_1} \in E\}$. From the order $<_1$, an order on the vertices of Ξ can be obtained, so that the h^{th} element can be selected, say v''. In this new order $<_2$, we can move v' just after v'', implying that $\alpha_{<_2}(v') = K$. The vertices between the old and the new position for v' can be affected by this change, whereas the situation remains unchanged for all others. If a vertex v is between the old and new position for v', then the value of $\alpha_{<_1}(v)$ might decrease. In such a case, the position of the vertex in the order needs to be modified, and this can be simply done by applying the procedure above to the vertices following the old position of v' in the order $<_1$. A sketch of the new algorithm we propose is in Alg. 1. This algorithm requires an order $<_1$ in input; as a consequence, the performances of this algorithm are dependent on the given initial order.

Algorithm 1 Algorithm for finding suitable orders for the DDGP

```
1: reorder(G, <_1)
2: copy order <_1 in <_2;
3: define ordered set B such that each v ∈ V is in the order <_2
4: for each v ∈ B, in order <_2 do
5:     if α_{<_2}(v) < K then
6:         let Ξ = {u ∈ V : (v, u)_{<_2} ∈ E};
7:         let h = K − α_{<_2}(v);
8:         let w = h^{th} element, in the order <_2, in Ξ;
9:         move, in the set B, v just after w;
10:        update order <_2 (from updated B);
11:    end if
12: end for
```

We remark that this algorithm could cycle. When there is a subset of vertices that are selected in repetition, it means that they form a subset of vertices having less than K connections with the rest. When the algorithm cycles, we can stop the execution, and no DDGP K-orders may exist.

3. Computational experiments

In this section, we present some computational results on a set of instance of the Wireless Sensor Network Localization (WSNL) problem [1, 2, 8]. It is supposed that $K = 2$ and that all distances are precisely known. The instances were generated in similar way as in [3]: on a square in $\mathbb{R}^2$ having side 1, all distances between randomly placed points, that are closer than a predefined radio range distance R, are supposed to be known.

We compared the running time of Alg. 1 to the greedy algorithm proposed in [4]. All codes have been written in C and compiled with the gcc compiler by GNU, version 4.7.1, under Linux on an Intel(R) Core(TM) i3-2120 CPU@3.30GHz with 8Gb RAM.

Table 1 shows some computational experiments for different sizes n and different radio ranges R. It can be easily remarked that the greedy algorithm is strongly dependent on the size n and on the cardinality of E, because the computational experiments are more expensive when the values of n and $|E|$ are larger. On the other side, Alg. 1 shows this behavior only in relation with n, while it improves its performances when $|E|$ is larger.

Table 1: Comparison between Alg 1 and Greedy algorithm proposed in [4] on a set of WSNL instances

Instances			Alg. 1	Greedy	Instances			Alg. 1	Greedy
n	R	$\|E\|$	CPU time	CPU time	n	R	$\|E\|$	CPU time	CPU time
4000	0.05	60351	1.23	0.98	8000	0.05	241590	3.35	5.83
4000	0.06	85815	0.76	1.13	8000	0.06	343873	1.21	7.59
4000	0.07	115511	0.47	1.33	8000	0.07	466346	1.07	9.71
4000	0.08	149606	0.32	1.62	8000	0.08	601909	0.57	11.96
4000	0.09	187789	0.25	1.91	8000	0.09	750550	0.43	14.71
4000	0.10	230116	0.19	2.29	8000	0.10	918520	0.36	17.01
6000	0.05	136532	1.71	2.64	10000	0.05	378545	3.77	11.09
6000	0.06	195323	0.98	3.39	10000	0.06	536711	2.70	14.61
6000	0.07	262742	0.73	4.24	10000	0.07	723071	0.97	17.81
6000	0.08	337476	0.40	5.07	10000	0.08	936524	0.68	22.18
6000	0.09	426764	0.32	5.27	10000	0.09	1182242	0.44	27.5
6000	0.10	518907	0.29	7.48	10000	0.10	1440175	0.49	32.69

In future works, we plan to develop in more details the theory behind the new proposed algorithm. Moreover, we will work for extending this new algorithm for solving instances of the DVOP where not all the available distances are precise. We will also explore the possibility to combine the two algorithms compared in this paper in the attempt of bringing their best properties into a hybrid one.

Acknowledgments

The authors would like to thank Professor Carlile Lavor by his encouragement, suggestions and help all the time.

References

[1] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye, Semidefinite programming approaches for sensor network localization with noisy distance measurements, *IEEE Transactions on Automation Science and Engineering*, 3:360–371, 2006.

[2] S. Kim, M. Kojima, and H. Waki, Exploiting sparsity in SDP relaxation for Sensor Network Localization, *SIAM Journal of Optimization*, 20:192–215, 2009.

[3] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20:2679–2708, 2010.

[4] C. Lavor, L. Jon, J. Lee S., L. Liberti, A. Mucherino, and M. Sviridenko. Discretization Orders for Distance Geometry Problems. *Optimization Letters*, 6:783–796, 2012.

[5] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[6] L. Liberti, C. Lavor, A. Mucherino, and N.n Maculan. Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18:33–51, 2011.

[7] A. Mucherino, C. Lavor, and L. Liberti. The discretizable distance geometry problem. *Optimization Letters*, 6:1671–1686, 2012.

[8] Y. Yemini. The positioning problem - a draft of an intermediate summary. In *Proceedings of the Conference on distributed Sensor Network*, pages 137–145. Carnegie-Mellon, Pittsburgh, 1978.

DGA 2013, pp. 153 – 155.

Ab-initio nanostructure determination

Saurabh R. Gujarathi [1] and Phillip M. Duxbury [2]

[1] *Dept. of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA.* saurabh@msu.edu

[2] *Dept. of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA.* duxbury@pa.msu.edu

Abstract Many complex materials at the nanoscale do not have periodic long range order and hence their structures cannot be solved by traditional crystallographic methods. The "nanostructure problem" is determining, with high precision, the arrangement of atoms in such irregular nanostructures. Our approach to this inverse problem is the use of distance geometry methods, which can reconstruct structures using only the interatomic distances obtained from the atomic pair distribution function, which is generated from scattering data.

Keywords: Structure determination, Nanoparticles, Inverse problem

Reconstruction of complex structures using pair distance information is an inverse problem that occurs in many branches of science and engineering [1–5]. Given a set of inter atomic distances we need to find the location of the atoms, up to global rotations and translations of the structure. This pair distance inverse problem may be interpreted as a complex network reconstruction problem where the edge weights are equal to the Euclidean distances between nodes in the network [6, 7].

In material physics, crystallography is the gold standard for structure determination. When crystals are not available, other methods are used. Determination of protein structure in solution has been successfully done by using the pair distance information extracted from NOESY NMR data [3, 4, 8–10]. In proteins, the list of residues or the sequence of a protein is known, enabling experiments to be carried out to specify the points between which each distance lies. This leads to the assigned case of the inverse problem. Algorithms for solving this type of problem are known to be easy, being of order the number of atoms in the structure (N). However, the NMR data has large uncertainties in the experimentally obtained interatomic distances, with imprecisions typically of the order of 25% or higher [11], making the problem computationally hard [12, 13].

In contrast, for problems concerning materials and most heterogeneous media, the pair distances are not assigned, as we do not know which nodes lie at the end of each distance. This makes reconstruction significantly harder and is the unassigned case of the inverse problem. The pair distribution function (PDF) method is used for the analysis of the local structure of nanoparticles and complex materials. In many complex materials, such as high performance thermoelectric materials [14], high temperature superconductors [15] and manganites [16], crystalline order and heterogeneous local distortions co-exist so that crystallographic and PDF methods are complementary. Crystallography finds the average structure and the PDF gives the local structure [17, 18]. The PDF gives a direct measure of the list of interatomic distances arising in the local structure, however the end points of the distances are not known. We face a computationally challenging problem known as the "nanostructure problem"[19].

In collaboration with Professor Billinge and his group at Columbia University, we developed an efficient algorithm for reconstructing structures which have a high symmetry, such as C_{60}

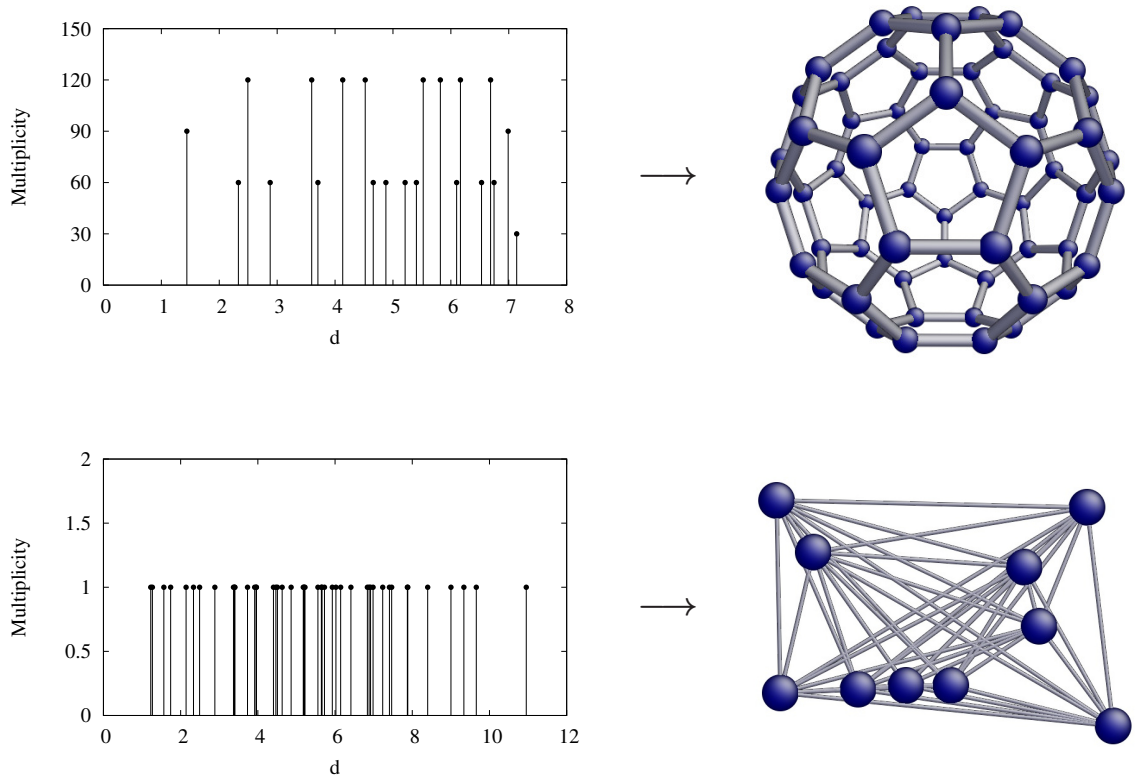


Figure 1: Some examples of the different types of structures we have reconstructed using Euclidean distance lists. The figures on the left are the distance lists while those on the right are the reconstructed structures. The plot on the top left is for C_{60} fullerene molecule that has a degenerate distance list and at the bottom left is that for a random set of 10 points in the plane that has a non-degenerate distance list. The multiplicity is on the Y axis while the distance is on the X axis (in arbitrary units). The structure of the C_{60} fullerene (top right) was found using the Liga algorithm and the structure of the random point set (bottom right) was found using the Tribond algorithm from the given distance lists.

and a range of crystal structures. The novel "Liga algorithm" [20–22] is inspired by the Spanish soccer league and is based on tournaments and promotion and relegation. Although this method works well for structures with relatively high symmetry, it fails miserably for low symmetry cases such as random point sets, due to the fact that there are a large number of unique pair distances in random structures. Thus, they fail for the general problem of complex Euclidean networks.

To overcome this problem we came up with the Tribond algorithm, which is specifically designed for solving structures which have low symmetry (Fig. 1). It makes use of the fact that any over-constrained cluster (core) will be very likely part of the final structure. Tribond finds such a cluster and then does the remaining buildup, all in polynomial time. We have successfully implemented the algorithm using C++ in two and three dimensions. In 2D, we have been able to reconstruct low symmetry structures consisting of a thousand atoms in about 24 hours on a desktop computer. Our Tribond algorithm solves the unassigned case of the inverse problem problem given precise distances and we also have some success in solving the problem when given imprecise distances. A hybrid algorithm that combines Tribond and Liga

algorithm would be able to solve structures which fall in between those having high symmetry and low symmetry.

References

[1] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation.* Wiley and Sons, New York, 1988.

[2] G. Crippen, "Chemical distance geometry: current realization and future projection," *Journal of mathematical chemistry*, vol. 6, no. 1, pp. 307–324, 1991.

[3] K. Wuthrich, "The development of nuclear magnetic resonance spectroscopy as a technique for protein structure determination," *Accounts of Chemical Research*, vol. 22, pp. 36–44, Jan. 1989.

[4] K. Wuthrich, "Protein structure determination in solution by nuclear magnetic resonance spectroscopy," *Science*, 1989.

[5] M. Li, Y. Otachi, and T. Tokuyama, "Efficient algorithms for network localization using cores of underlying graphs," *Algorithms for Sensor Systems*, pp. 101–114, 2012.

[6] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, "Molecular distance geometry methods: from continuous to discrete," *International Transactions in Operational Research*, vol. 18, no. 1, pp. 33–51, 2011.

[7] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *ArXiv e-print, arXiv:1205.0349v1*, May 2012.

[8] J. Yoon, Y. Gad, and Z. Wu, "Mathematical modeling of protein structure using distance geometry," tech. rep., 2000.

[9] J. C. Kendrew, Dickerson R. E., B. E. Strandberg, R. G. Hart, D. R. Davies, D. C. Phillips, and V. C. Shore, "Structure of Myoglobin," *Nature*, vol. 185, pp. 422–427, 1960.

[10] M. F. Perutz, M. Rossmann, A. Cullis, H. Muirhead, G. Will, and A. C. T. North, "Structure of Haemoglobin," *Nature*, vol. 185, pp. 416–422, 1960.

[11] M. Nilges and S. I. O'Donoghue, "Ambiguous NOEs and automated NOE assignment," *Progress in Nuclear Magnetic Resonance Spectroscopy*, vol. 32, pp. 107–139, Apr. 1998.

[12] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM Journal on Optimization*, vol. 5, no. 4, pp. 835–857, 1995.

[13] B. Berger, J. Kleinberg, and T. Leighton, "Reconstructing a three-dimensional model with arbitrary errors," *Journal of the ACM (JACM)*, pp. 1–16, 1999.

[14] H. Lin, E. Božin, S. Billinge, E. Quarez, and M. Kanatzidis, "Nanoscale clusters in the high performance thermoelectric AgPbmSbTem+2," *Physical Review B*, vol. 72, pp. 1–7, Nov. 2005.

[15] L. Malavasi, G. a. Artioli, H. Kim, B. Maroni, B. Joseph, Y. Ren, T. Proffen, and S. J. L. Billinge, "Local structural investigation of SmFeAsO(1-x)F(x) high temperature superconductors.," *Journal of physics. Condensed matter*, vol. 23, p. 272201, July 2011.

[16] T. Proffen and S. Billinge, "Probing the local structure of doped manganites using the atomic pair distribution function," *Applied Physics A*, vol. 74, pp. 1770–1772, 2002.

[17] S. J. Billinge, "Nanoscale structural order from the atomic pair distribution function (PDF): There's plenty of room in the middle," *Journal of Solid State Chemistry*, vol. 181, pp. 1695–1700, July 2008.

[18] S. J. L. Billinge and M. G. Kanatzidis, "Beyond crystallography: the study of disorder, nanocrystallinity and crystallographically challenged materials with pair distribution functions.," *Chemical communications (Cambridge, England)*, pp. 749–60, Apr. 2004.

[19] S. J. L. Billinge and I. Levin, "The problem with determining atomic structure at the nanoscale.," *Science (New York, N.Y.)*, vol. 316, pp. 561–5, Apr. 2007.

[20] P. Juhás, D. M. Cherba, P. M. Duxbury, W. F. Punch, and S. J. L. Billinge, "Ab initio determination of solid-state nanostructure.," *Nature*, vol. 440, pp. 655–8, Mar. 2006.

[21] P. Juhás, L. Granlund, P. M. Duxbury, W. F. Punch, and S. J. L. Billinge, "The Liga algorithm for ab initio determination of nanostructure.," *Acta crystallographica. Section A, Foundations of crystallography*, vol. 64, pp. 631–40, Nov. 2008.

[22] P. Juhas, L. Granlund, S. R. Gujarathi, P. M. Duxbury, and S. J. L. Billinge, "Crystal structure solution from experimentally determined atomic pair distribution functions," *Journal of Applied Crystallography*, vol. 43, pp. 623–629, 2010.

DGA 2013, pp. 157 – 161.

Distance Eigenvalue Location in Threshold Graphs*

David P. Jacobs[1], Vilmar Trevisan[2], and Fernando C. Tura[3]

[1] *School of Computing, Clemson University, USA,* dpj@clemson.edu

[2] *Instituto de Matemática, UFRGS, Brazil,* trevisan@mat.ufrgs.br

[3] *Campus Alegrete, UNIPAMPA, Brazil,* fernandotura@unipampa.edu.br

Abstract Let G be a threshold graph of order n with distance matrix $\mathbf{\Theta}$. We give an $O(n)$ algorithm for constructing a diagonal matrix congruent to $B_x = \mathbf{\Theta} + xI$ for any real x. An application using Sylvester's Law of Inertia can determine, in linear-time, how many eigenvalues of $\mathbf{\Theta}$ lie in any interval, allowing fast divide-and-conquer approximation. We also show that any distance eigenvalue $\lambda \neq -1, -2$ must be simple.

Keywords: eigenvalue, distance matrix, threshold graph

1. Introduction

Distance in graph theory is a simple but powerful idea, upon which many parameters depend, including diameter, radius, average distance and Wiener index. A *path* in a graph is a sequence of distinct vertices, such that adjacent vertices in the sequence are adjacent in the graph. The *length* of a path is the number of edges on the path. For connected graphs, the *distance* between two vertices u and v, denoted $d(u, v)$, is the length of a shortest $u - v$ path.

The *diameter* of a connected graph G, denoted $\mathrm{diam}(G)$, is the maximum distance between two vertices. The *eccentricity* of a vertex is the maximum distance from it to any other vertex. The *radius*, denoted $\mathrm{rad}(G)$, is the minimum eccentricity among all vertices of G.

The *average distance* of a graph G of order n, denoted $\mu(G)$, is the expected distance between a randomly chosen pair of distinct vertices. The study of the average distance began with the chemist Wiener [14], who noticed that the melting point of certain hydrocarbons is proportional to the sum of all distances between unordered pairs of vertices of the corresponding graph. This sum, denoted by $W(G)$, is called the *Wiener index* of G. Clearly,

$$W(G) = \binom{n}{2}\mu(G).$$

The Wiener index and its applications to chemistry have received much attention (See, for example, [1, 2, 4, 10, 12, 13]).

The *distance matrix* $\mathbf{\Theta}$ of a connected graph G is the matrix whose rows and columns are indexed by its vertices such that its (u, v)-entry is equal to $d(u, v)$. If $\mathbf{1}$ denotes the all 1's

*The second author was partially supported by CNPq (Grants 309531/2009-8 and 473815/2010-9) and FAPERGS (Grant 11/1619-2). The third author was on leave from UNIPAMPA and supported by CAPES (Grant 0283/12-6).

column vector, the Wiener index may be written in the form

$$W = \frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Theta}\mathbf{1}}{2}.$$

The eigenvalues of $\boldsymbol{\Theta}$ are called the *distance eigenvalues* of G, form the *distance spectrum*, and have several real-world applications. Distance eigenvalues were first studied by Graham and Pollack in 1971 to solve a data communication problem [6]. The distance matrix contains information on various walks in chemical graphs. It is useful in the computation of topological indices and thermodynamic properties such as pressure and temperature coefficients. It contains more structural information than the adjacency matrix [7]. In the chemistry literature, the largest eigenvalue of $\boldsymbol{\Theta}(G)$ helps to model the boiling point of alkanes [1]. In addition to chemistry, distance matrices find applications in music theory, ornithology, molecular biology, psychology, archeology etc. (See [3] and the papers cited therein.)

This paper is concerned with the distance eigenvalues of *threshold graphs*. Threshold graphs have several applications in psychology, scheduling, and synchronization of parallel processes [11]. They can be characterized in many ways, but a simple way of obtaining a threshold graph is through an iterative process which starts with an isolated vertex, and where, at each step, either a new isolated vertex is added, or a vertex adjacent to all previous vertices (*dominating vertex*) is added. We represent the graph with a binary sequence $(b_1, \ldots, b_n)$, ordering the vertices in the way they are created. The adjacency matrix A and distance matrix $\boldsymbol{\Theta}$ of the threshold graph represented by $(0, 1, 0, 1)$ are

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad \boldsymbol{\Theta} = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 2 & 1 \\ 2 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{1}$$

If $\boldsymbol{\Theta} = [a_{ij}]$ is the distance matrix of a threshold graph G represented with $(b_1, \ldots, b_n)$, then it is easy to see that if $b_i = 1$, $a_{ij} = a_{ji} = 1$, for $j < i$. And if $b_i = 0$, $a_{ij} = a_{ji} = 2$, for $j < i$.

2. Diagonalizing $\boldsymbol{\Theta} + xI$

Recall that two matrices R and S are *congruent* if there exists a nonsingular matrix P such that $R = P^T SP$. Our main result is an $O(n)$ algorithm for constructing a *diagonal* matrix congruent to $B_x = \boldsymbol{\Theta} + xI$, where $\boldsymbol{\Theta}$ is the distance matrix of a threshold graph, and x is an arbitrary scalar. The algorithm proceeds in $n-1$ stages and works bottom-up, and right-to-left. At each stage, adjacent rows and columns m and $m-1$ participate in operations. *Diagonalization* is achieved because at the end of this stage, all entries of row and column m, will be zero except the diagonal element. For $\boldsymbol{\Theta}$ in (1) and $x = 1$, the algorithm would proceed as follows:

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

Congruence is guaranteed because at each stage of the algorithm we use the *same* pair of elementary row and column operations. For example, in transforming the first matrix above to the second, we employed the row and column operations

$$R_4 \leftarrow R_4 - \frac{1}{2}R_3, \quad C_4 \leftarrow C_4 - \frac{1}{2}C_3$$

```
Algorithm Diagonalize(G, x)
   initialize d(i) ← x, for all i
   for  m = n to 2
      α ← d(m)
      if b_m = 0
         α ← α/4
      if b_{m−1} = 1
         if α + x ≠ 2
            d(m − 1) ← (αx − 1)/(α + x − 2)
            d(m) ← α + x − 2
         else if x = 1
            d(m − 1) ← 1
            d(m) ← 0
         else
            d(m − 1) ← 1
            d(m) ← −(1 − x)^2
            m ← m − 1
      else if b_{m−1} = 0
         if α + x/4 ≠ 1
            d(m − 1) ← (αx − 1)/(α + x/4 − 1)
            d(m) ← α + x/4 − 1
         else if x = 2
            d(m − 1) ← 2
            d(m) ← 0
         else
            d(m − 1) ← 2
            d(m) ← −(1/2)(1 − x/2)^2
            m ← m − 1
   end loop
```

Figure 1: Diagonalizing $\mathbf{\Theta} + xI$.

and then

$$R_3 \leftarrow R_3 - 2R_4, \quad C_3 \leftarrow C_3 - 2C_4$$

What is remarkable is that we do not need to store the entire matrix, only the diagonal and the representation $(b_1, \ldots, b_n)$ of G. Our $O(n)$ time and space algorithm is shown in Figure 1. The proof of correctness appears in the full-length version of our paper.

Theorem 1. *For inputs G and x, where G is a threshold graph with distance matrix $\mathbf{\Theta}$, algorithm* `Diagonalize` *computes a diagonal matrix D, which is congruent to $\Theta + xI$.*

3. Finding distance eigenvalues

We seek the eigenvalues of $\mathbf{\Theta}$, the distance matrix of a threshold graph G. The proof of the following theorem, which depends on Sylvester's Law of Inertia, may be found in [8, 9].

Theorem 2. *Let D be a diagonal matrix congruent to $\mathbf{\Theta} - \alpha I$, where $\mathbf{\Theta}$ is real symmetric.*

i. The number of eigenvalues of $\mathbf{\Theta}$ greater than α is the number of positive entries in D.

ii. The number of eigenvalues of $\mathbf{\Theta}$ less than α is the number of negative entries in D.

iii. The multiplicity of eigenvalue α is the number of zero entries in the diagonal of D.

Corollary 3. *Counting multiplicities, the number of eigenvalues of* $\mathbf{\Theta}$ *in* $(\alpha, \beta]$*, is the number of positive entries in the diagonalization of* $\mathbf{\Theta} - \alpha I$*, minus the number of positive entries in the diagonalization of* $\mathbf{\Theta} - \beta I$*.*

This observation shows that we may determine the number of eigenvalues in an interval by making *two* calls to algorithm `Diagonalize`. As an example, consider G represented by $(0, 1, 0, 1)$ and $x = 1$. After initialization, when $m = 4$, we will have $\alpha = x = 1$, $b_4 = 1$ and $b_3 = 0$, so the first step will assign $d_3 \leftarrow \frac{\alpha x - 1}{\alpha + \frac{x}{4} - 1} = 0$ and $d_4 \leftarrow \alpha + \frac{x}{4} - 1 = \frac{1}{4}$. Next, when $m = 3$, we will have $\alpha = 0$, $x = 1$, $b_2 = 1$ and $b_3 = 0$. This second step will assign $\alpha \leftarrow \frac{\alpha}{4} = 0$, $d_2 \leftarrow \frac{\alpha x - 1}{\alpha + x - 2} = 1$, $d_3 \leftarrow \alpha + x - 2 = -1$. Finally, when $m = 2$, we will have $\alpha = 1$, $x = 1$, $b_1 = 0$ and $b_2 = 1$, so we assign: $d_1 \leftarrow \frac{\alpha x - 1}{\alpha + \frac{x}{4} - 1} = 0$, $d_2 \leftarrow \alpha + \frac{x}{4} - 1 = \frac{1}{4}$. The following table illustrates the sequence of states.

b_i	d_i
0	1
1	1
0	1
1	1

initial

b_i	d_i
0	1
1	1
0	0
1	$\frac{1}{4}$

after m=4

b_i	d_i
0	1
1	1
0	-1
1	$\frac{1}{4}$

after m=3

b_i	d_i
0	0
1	$\frac{1}{4}$
0	-1
1	$\frac{1}{4}$

after m=2

By Theorem 2, this means that $x = -1$ is an eigenvalue of G of multiplicity 1, there are two eigenvalues greater than -1 and one eigenvalue is smaller than -1. When applying the algorithm to the same graph and $x = 0$, the final sequence is given by $d = (7/3, -3/7, -7/4, -1)$. This implies that 3 eigenvalues are negative and 1 is positive. We conclude that there is a single distance eigenvalue $\lambda \in (-1, 0]$. This technique allows fast divide-and-conquer approximation: Letting $x = -\frac{1}{2}$ will locate λ in $(-1, -.5]$ or $(-.5, 0]$.

An eigenvalue is *simple* if its multiplicity is one. Using our algorithm, we can prove:

Theorem 4. *In the distance matrix of a threshold graph, an eigenvalue* λ *is simple if* $\lambda \neq -1, -2$*.*

4. Research Problem

In [6] it was shown that $\det(\mathbf{\Theta}) = (-1)^{n-1}(n-1)2^{n-2}$, where Θ is the distance matrix of a tree of order n. We seek a formula for $\det(\mathbf{\Theta})$, for distance matrices of threshold graphs G, in terms of the representation of G.

References

[1] A.T. Balaban, D. Ciubotariu and M. Medeleanu, Topological indices and real number vertex invariants based on graph eigenvalues and eigenvectors, J. Chem. Inf. Comput. Sci. 31 (1991) 517–523.

[2] A.T. Balaban and M.V. Diudea, Real number vertex invariants: regressive distance sums and related topological indices, J. Chem. Inf. Comput. Sci. 33 (1993), 421–428.

[3] K. Balasubramanian, Computer generation of distance polynomials of graphs, Journal of Computational Chemistry 11 (1990), 829–836.

[4] A.A. Dobrynin, R. Entringer and I. Gutman, Wiener index of trees: theory and applications, Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications, 66(3), (2001), 211–249.

[5] W. Goddard and O.R. Oellermann. Distance in graphs, In Structural Analysis of Complex Networks, pages 49–72. Birkhäuser, 2011.

[6] R. L. Graham and H. O. Pollak, On the addressing problem for loop switching, Bell System Tech. J. 50 (1971) 2495–2519.

[7] G. Indulal, Distance spectrum of graph compositions, Ars Mathematica Contemporanea 2 (2009) 93–100.

[8] D. P. Jacobs and V. Trevisan, Locating the eigenvalues of trees, Linear Algebra and its Applications 434 (2011) 81–88.

[9] D. P. Jacobs, V. Trevisan and F. Tura, Eigenvalue location in threshold graphs, 2012, manuscript.

[10] D.J. Klein, Z.Milanic, D. Plavsic and N. Trinajstic, Molecular topological index: a relation with the Wiener index, J. Chem. Inf. Compu. Sci. 32 (1992), 304–305.

[11] N. V. R. Mahadev and U. N. Peled, Threshold graphs and related topics, Elsevier, 1995.

[12] B. Mohar, A novel definition of the Wiener index of trees, J. Chem. Inf. Comp. Sci. 33 (1993), 153–154.

[13] M. Randic, A.F. Kleiner and L.M. DeAlba, Distance matrices, J. Chem. Inf. Comp. Sci. 34 (1994), 277–286.

[14] H. Wiener, Structural determination of paraffin boiling points. J. Amer. Chem. Soc., 69(1), (1947), 17–20.

DGA 2013, pp. 163 – 167.

A Space Filling Global Optimization Algorithm to Solve Molecular Distance Geometry Problems

Mario Salvatierra Junior[1]

[1]*UFAM, Manaus, AM,Brazil,* mario@icomp.ufam.edu.br

Abstract In this work we consider a global optimization algorithm based on a space filling curve, the Lissajous Curve, for the Molecular Distance Geometry Problem (MDGP). We will deal with the problem through its continuous nature.

Keywords: Global optimization, Distance geometry, Space filling curve.

1. Introduction

The problem of determining molecular structures has attracted great interest due to its application in relevant areas such as medicine, pharmacy, biology, design of materials, and chemistry [1]. The Molecular Distance Geometry Problem (MDGP) consists on estimate relative positions of all atoms of a molecule, given a subset of all the pair-wise distances between the atoms. Then, the MDGP can be defined as determine positions for m points $x_1, x_2, \ldots, x_m \in \mathbb{R}^3$ such that, for a given set of pairs D and given bounds l_{ij}, u_{ij} :

$$l_{ij} \leq \| x_i - x_j \| \leq u_{ij}, \quad \forall \{i,j\} \in D.$$

Where each point x_i, $i \in \{1, \ldots, m\}$ represents (the center of) an atom and $D \subseteq \{1, \ldots, m\} \times \{1, \ldots, m\}$ is a set that identifies the available lower l_{ij} and upper u_{ij} bounds for the pair-wise Euclidean distances.
So, the MDGP can be reformulated as a mathematical programming problem:

$$\begin{aligned} min f(x) = \quad & \frac{1}{2} \sum_{(i,j) \in D} max^2 \left(\frac{l_{ij}^2 - \| x_i - x_j \|^2}{l_{ij}^2}, 0 \right) + max^2 \left(\frac{\| x_i - x_j \|^2 - u_{ij}^2}{u_{ij}^2}, 0 \right) \\ & where \quad x = (x_1, \ldots, x_m) \in \mathbb{R}^{3m}, \\ & x_k \in \mathbb{R}^3, \quad \forall k \in \{1, \ldots, m\} \end{aligned} \tag{1}$$

It is easy to see that $f(x_1, \ldots, x_m) = 0$ if and only if all the restrictions $l_{ij} \leq \| x_i - x_j \| \leq u_{ij}$ are satisfied. Thus, as the function $f(x) \geq 0$, $\forall x \in \mathbb{R}^{3m}$, our goal is to find a global minimum of f.

2. Regularized hessians

Different approaches to the MDGP (1) have been explored and someones are about smoothing techniques [6].

The problem (1) has the following general form:

$$min \quad f(x) = \frac{1}{2}\sum_{i=1}^{N} max^2\left(0, g_i(x)\right) + max^2\left(0, h_i(x)\right) \tag{2}$$

It is easy to see that $f(x)$ has continuous first (but not second) derivatives. The second derivatives of $f(x)$ are, in general, discontinuous at the points where $g_i(x) = 0$. This is an disadvantage for minimization algorithms based on quadratic models, like Newton's method, which enjoys good convergence properties.

Consider the associated problem:

$$min \quad \Psi(x, z, w) = \frac{1}{2}\sum_{i=1}^{N}\left[g_i(x) + z_i^2\right]^2 + \left[h_i(x) + w_i^2\right]^2 \tag{3}$$

In the following lemma we prove that problems (2) and (3) are equivalent. Problem (3) has continuous second derivatives but depends on the additional variables $z_1, \ldots, z_N$ and $w_1, \ldots, w_N$.

Lemma 1. *The point $\overline{x} \in \mathbb{R}^{3m}$ is a global minimizer of (2) if, and only if, there exists $\overline{z}, \overline{w} \in \mathbb{R}^N$ such that $(\overline{z}, \overline{w})$ is a global minimizer of (3). Moreover $f(\overline{x}) = \Psi\left(\overline{x}, \overline{z}, \overline{w}\right)$.*

Proof. Firstly, note that given any $x \in \mathbb{R}^{3m}$, define for each $i = 1, \ldots, N$

$$z_i = z_i(x) = \begin{cases} \sqrt{-g_i(x)}, & \text{if} \quad g_i(x) \leq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

and

$$w_i = w_i(x) = \begin{cases} \sqrt{-h_i(x)}, & \text{if} \quad h_i(x) \leq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Then, $f(x) = \Psi(x, z, w)$, for z, w as in equations (4) and (5). If $\overline{x} \in \mathbb{R}^{3m}$ is a global minimizer of f, set $\overline{z} = \overline{z}(\overline{x}), \overline{w} = \overline{w}(\overline{x})$ as (4) and (5). Given another $x \in \mathbb{R}^{3m}$, let be $z' = z'(x), w' = w'(x)$ as (4) and (5). Then for all z, w, $\Psi\left(\overline{x}, \overline{z}, \overline{w}\right) = f(\overline{x}) \leq f(x) = \Psi(x, z', w') \leq \Psi(x, z, w)$. So, $(\overline{x}, \overline{z}, \overline{w})$ is a global minimizer of Ψ. Conversely, if $(\overline{x}, \overline{z}, \overline{w})$ is a global minimizer of Ψ, we have that $\Psi\left(\overline{x}, \overline{z}, \overline{w}\right) = \Psi(\overline{x}, z^*, w^*) = f(\overline{x})$, for $z^* = z^*(\overline{x}), w^* = w^*(\overline{x})$ as (4) and (5). For another $x \in \mathbb{R}^{3m}$, let be $z' = z'(x), w' = w'(x)$ as (4) and (5). Thus, $f(\overline{x}) = \Psi(\overline{x}, z^*, w^*) = \Psi\left(\overline{x}, \overline{z}, \overline{w}\right) \leq \Psi(x, z', w') = f(x)$. So, $\overline{x}$ is a global minimizer of f.

This equivalence motivates us to study Newton-like minimization methods for solving (3). Computing the gradient and the Hessian matrix of Ψ, we get

$$\nabla\Psi(x, z, w) = \begin{bmatrix} \sum_{i=1}^{N}\left[g_i(x) + z_i^2\right] \cdot \nabla g_i(x) + \left[h_i(x) + w_i^2\right] \cdot \nabla h_i(x) \\ 2\left[g_1(x) + z_1^2\right] z_1 \\ \vdots \\ 2\left[g_N(x) + z_N^2\right] z_N \\ 2\left[h_1(x) + w_1^2\right] w_1 \\ \vdots \\ 2\left[h_N(x) + w_N^2\right] w_N \end{bmatrix}, \tag{6}$$

and

$$\nabla^2\Psi = \left[\begin{array}{c|c|c} \begin{array}{c} \sum_{i=1}^{N} \nabla g_i(x)\cdot\nabla g_i(x)^T \\ +\sum_{i=1}^{N}\left[g_i(x)+z_i^2\right]\cdot\nabla^2 g_i(x) \\ +\sum_{i=1}^{N}\nabla h_i(x)\cdot\nabla h_i(x)^T \\ +\sum_{i=1}^{N}\left[h_i(x)+w_i^2\right]\cdot\nabla^2 h_i(x) \end{array} & 2z_1\nabla g_1(x)\cdots 2z_N\nabla g_N(x) & 2w_1\nabla h_1(x)\cdots 2w_N\nabla h_N(x) \\ \hline \begin{array}{c} 2z_1\cdot\nabla g_1(x)^T \\ \vdots \\ 2z_N\cdot\nabla g_N(x)^T \end{array} & \begin{array}{cc} 6z_1^2+2g_1(x) & 0 \\ \ddots & \\ 0 & 6z_N^2+2g_N(x) \end{array} & 0 \\ \hline \begin{array}{c} 2w_1\cdot\nabla h_1(x)^T \\ \vdots \\ 2w_N\cdot\nabla h_N(x)^T \end{array} & 0 & \begin{array}{cc} 6w_1^2+2h_1(x) & 0 \\ \ddots & \\ 0 & 6w_N^2+2h_N(x) \end{array} \end{array}\right]. \tag{7}$$

Definition 1. *We shall call good triplet (x,z,w) to those triplets such that $z_i^2 = -g_i(x)$ when $g_i(x)\le 0$ and $z_i = 0$ when $g_i(x) > 0$,$w_i^2 = -h_i(x)$ when $h_i(x)\le 0$ and $w_i = 0$ when $h_i(x) > 0$. In other words $z_i^2 = max\{\,0,-g_i(x)\}$ and $w_i^2 = max\{\,0,-h_i(x)\}$.*

Theorem 2. *Assume that (x,z,w) is a good triplet. Assume that $\overline{\Delta x}\in\mathbb{R}^{3m}$ satisfies*

$$\begin{aligned} &\left[\sum_{i=1}^{N}\nabla g_i(x)\cdot\nabla g_i(x)^T + \sum_{\{i|g_i(x)\ge 0\}} g_i(x)\cdot\nabla^2 g_i(x)\right. \\ &\left.+\sum_{i=1}^{N}\nabla h_i(x)\cdot\nabla h_i(x)^T + \sum_{\{i|h_i(x)\ge 0\}} h_i(x)\cdot\nabla^2 h_i(x)\right]\overline{\Delta x} = \\ &-\left[\sum_{\{i|g_i(x)\ge 0\}} g_i(x)\cdot\nabla g_i(x) + \sum_{\{i|h_i(x)\ge 0\}} h_i(x)\cdot\nabla h_i(x)\right] \end{aligned} \tag{8}$$

Then, there exists $\overline{\Delta z},\overline{\Delta w}\in\mathbb{R}^N$ such that

$$\nabla^2\Psi(x,z,w)\begin{pmatrix}\overline{\Delta x}\\ \overline{\Delta z}\\ \overline{\Delta w}\end{pmatrix} = -\nabla\Psi(x,z,w) \tag{9}$$

Remarks.

(i) The theorem above shows that, essentially, a Newtonian iteration for the minimization of $\Psi(x,z,w)$ followed by a restoration $z_i^2 \leftarrow max\{\,0,-g_i(x)\}$ and $w_i^2 \leftarrow max\{\,0,-h_i(x)\}$ is equivalent to a Newton iteration for minimizing $f(x)$ provided that we define

$$\begin{aligned} \nabla^2 max\{0,g_i(x)\}^2 &= \nabla^2\left[g_i(x)^2\right], \quad \text{if} \quad g_i(x)=0, \\ \nabla^2 max\{0,h_i(x)\}^2 &= \nabla^2\left[h_i(x)^2\right], \quad \text{if} \quad h_i(x)=0. \end{aligned} \tag{10}$$

(ii) The singularity of $\nabla^2\Psi(x,z,w)$ corresponds to the discontinuity of $\nabla^2 f(x)$.

Theorem 3. *Assume that (x, z, w) is a good triplet. Given $\epsilon > 0$, assume that $\overline{\Delta x} \in \mathbb{R}^{3m}$ satisfies*

$$\begin{aligned}
&\left\{ \sum_{\{i|g_i(x)<0\}} \frac{\epsilon}{\epsilon - 2g_i(x)} \nabla g_i(x) \cdot \nabla g_i(x)^T + \sum_{\{i|g_i(x)\geq 0\}} \left[\nabla g_i(x) \cdot \nabla g_i(x)^T + g_i(x) \cdot \nabla^2 g_i(x) \right] \right. \\
&\left. + \sum_{\{i|h_i(x)<0\}} \frac{\epsilon}{\epsilon - 2h_i(x)} \nabla h_i(x) \cdot \nabla h_i(x)^T + \sum_{\{i|h_i(x)\geq 0\}} \left[\nabla h_i(x) \cdot \nabla h_i(x)^T + h_i(x) \cdot \nabla^2 h_i(x) \right] \right\} \overline{\Delta x} = \\
&- \left[\sum_{\{i|g_i(x)\geq 0\}} g_i(x) \cdot \nabla g_i(x) + \sum_{\{i|h_i(x)\geq 0\}} h_i(x) \cdot \nabla h_i(x) \right]
\end{aligned} \tag{11}$$

Then, there exists $\overline{\Delta z}, \overline{\Delta w} \in \mathbb{R}^N$ such that

$$\nabla^2 \Psi(x, z, w) \begin{pmatrix} \overline{\Delta x} \\ \overline{\Delta z} \\ \overline{\Delta w} \end{pmatrix} = -\nabla \Psi(x, z, w) \tag{12}$$

Remarks.

(i) Unlike Theorem 2, in Theorem 3 we see that the $(\overline{\Delta z}, \overline{\Delta w})$-part of the solution of (12) is uniquely determined. This is due to the regularizing perturbation. Defining as (10), the system (11) can be written as

$$\begin{aligned}
&\left\{ \nabla^2 \left[\frac{1}{2} \sum_{i=1}^{N} max \left\{0, g_i(x)\right\}^2 \right] + \sum_{\{i|g_i(x)<0\}} \frac{\epsilon}{\epsilon - 2g_i(x)} \nabla g_i(x) \cdot \nabla g_i(x)^T \right. \\
&\left. + \nabla^2 \left[\frac{1}{2} \sum_{i=1}^{N} max \left\{0, h_i(x)\right\}^2 \right] + \sum_{\{i|h_i(x)<0\}} \frac{\epsilon}{\epsilon - 2h_i(x)} \nabla h_i(x) \cdot \nabla h_i(x)^T \right\} \overline{\Delta x} = \\
&- \left\{ \nabla \left[\frac{1}{2} \sum_{i=1}^{N} max \left\{0, g_i(x)\right\}^2 \right] + \nabla \left[\frac{1}{2} \sum_{i=1}^{N} max \left\{0, h_i(x)\right\}^2 \right] \right\},
\end{aligned} \tag{13}$$

or, equivalently,

$$\begin{aligned}
&\left[\nabla^2 f(x) + \sum_{\{i|g_i(x)<0\}} \frac{\epsilon}{\epsilon - 2g_i(x)} \nabla g_i(x) \cdot \nabla g_i(x)^T \right. \\
&\left. + \sum_{\{i|h_i(x)<0\}} \frac{\epsilon}{\epsilon - 2h_i(x)} \nabla h_i(x) \cdot \nabla h_i(x)^T \right] \overline{\Delta x} = -\nabla f(x).
\end{aligned} \tag{14}$$

The reasoning above leads us to define the Regularized Hessian of f, given $\epsilon > 0$, as

$$\nabla^2 f(x, \epsilon) = \nabla^2 f(x) + \sum_{\{i|g_i(x)<0\}} \frac{\epsilon}{\epsilon - 2g_i(x)} \nabla g_i(x) \nabla g_i(x)^T + \sum_{\{i|h_i(x)<0\}} \frac{\epsilon}{\epsilon - 2h_i(x)} \nabla h_i(x) \nabla h_i(x)^T \tag{15}$$

The Regularized Hessian in (15) do not exhibit discontinuities on the boundaries $g_i(x) = 0$ and $h_i(x) = 0$. Since the perturbation is positive semidefinite, the perturbed Hessian is positive semidefinite provided that $\nabla^2 f(x)$ is. This is an advantage for minimization algorithms based on quadratic models.

3. The Global Optimization Algorithm

The global optimization algorithm presented here is based on idea to cover the domain Ω through a dense curve. Suppose that $\Omega \subset \mathbb{R}^n$ is a closed box with nonempty interior. This is:

$$\Omega = \{x \in \mathbb{R}^n | l \leq x \leq u\}.$$

The Lissajous curve is:

$$\gamma(t) = (\cos(\theta_1 t + \varphi_1), \ldots, \cos(\theta_n t + \varphi_n)). \tag{16}$$

Given $x_0 \in \Omega$ and choosing appropriately $\varphi_1, \ldots, \varphi_n$ we can find a Lissajous curve such that $\gamma(0) = x_0$. Clearly, the Lissajous curves are smooth. Under certain conditions in the coefficients $\theta_1, \ldots, \theta_n \in \mathbb{R}$, the image of a Lissajous curve is dense on $[-1, 1]^n$ and under linear transformations it is dense in Ω [5].

The strategy to solve the problem (1) is to use Newton-like minimization local methods using the regulrized hessians and then use the Lissajous curve to try to escape from local minimizer to a better one.

4. Computational experiments

The instances that will be considered are generated from data in the Protein Data-Bank [2, 4] and we will adopt some instances from the work by Moré and Wu [3].

References

[1] T. Schlick. *Molecular Modeling and Simulation: an Interdisciplinary Guide*. Springer, 2002.

[2] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. *The protein data bank. Nucleic Acids Research*, 28:235–242, 2000.

[3] J. Moré, and Z. Wu. *Distance geometry optimization for protein structures. J. Glob. Optim.*, 15:219–234, 1999.

[4] www.pdb.org

[5] L.T. Santos, F. Yano, M. Salvatierra, J.M. Martínez, R. Andreani and M. Tygel. *A global optimization algorithm applied to the common reflection surface (CRS) problem. Journal of Seismic Exploration*, 14:217–233,2005.

[6] M. Souza, A.E. Xavier, C. Lavor and N. Maculan. *Hyperbolic smoothing and penalty techniques applied to molecular structure determination. Operations Research Letters*, 39:461–465,2011.

DGA 2013, pp. 169 – 174.

From Star Configuration to Minimum Length Spanning Tree: The Role of Distances in Optimal Access Networks

Henrique P. L. Luna[1]

[1]*Instituto de Computação, Universidade Federal de Alagoas, 57072-970, Maceió, AL Brasil,* henrique.luna@pq.cnpq.br

Abstract Given a set of $n+1$ points on the two-dimensional plane, indexed by $i = 0, 1, 2, ..., n$, suppose that point 0 is the source of a single commodity that must be delivered to the other n points. For each of these customer points a specific demand q_i is required. Local access design concerns the problem of finding the minimum cost network that connects all points, using only lines joining pairs of points from the given set. Fixed (structural) and variable (operational) costs are taken into account in the connecting network. For each used line both the fixed and the variable cost to install and to use link (ij) are directly proportional to the distance d_{ij} between points i and j. A spanning tree is a natural candidate for the underlined structure of an optimal local access design. Two specific solutions plays a major role to calculate a lower bound for the total cost. A star configuration, with degree n at the origin node 0, that minimizes the sum of variable costs. And a minimum length spanning tree, that minimizes the sum of fixed costs. The lower bound is expressed is terms of the used line distances d_{ij}, the customer demands q_i and the parameters β and γ, where β express the fixed cost per unit of distance and γ is the variable cost per unit of flow and per unit of distance. Necessary and sufficient conditions are derived to verify the optimality of a feasible solution. For given demands and line distances among the points, there exists an open interval for the fixed over variable cost ratio β/γ for which neither a star configuration nor a minimum length spanning tree corresponds to an optimal network.

Keywords: network design, distance in fixed cost, minimum length spanning tree, distance in variable cost, shortest paths, computer networks, geometric structures , graph theory

1. Introduction

The spatial nature and hierarchical organization of telecommunication and transportation systems can be found in several real world applications, such as the location of switching centres or postal offices, and plays a major role in operations research and management science models. Minimum distances are crucial for the objective of cost minimization in these public systems. Together with values of the spatially distributed demands, the influence of distances must be taken into account to find optimized levels of customer concentrations, what enables the economies of scale of aggregating the flows in the related networks. The main differences among the models concern the hierarchical level of network design, typically backbone versus local access network, and how the relevant aspects of connectivity, capacity, reliability, demand patterns, routing, pricing, performance and quality of service are considered for such networks ([6] [3]). Depending on the context or application, hub nodes are called switches, warehouses, water sources, facilities, concentrators or access points. Likewise, backbones may be referred to as hub-level networks and local access networks may be called distribution or tributary networks. Normally, backbone links carry larger volumes of traffic than tributary links. Traffic originating at a specific customer location can pass through a local access network to get to

a hub node. After passing through the backbone network, the traffic again uses a local access network to travel from a hub to its final destination at another location. The influence of distance in cost occurs in all levels of an hierarchical network, but plays a major role in the access networks.

For each level of network design, a wide range of model formulations has been covered by the related literature, which may be classified as having a deterministic or stochastic character, and also according to the continuous or discrete nature of the model. Because of the complexity of these problems, local access network design and backbone design are often considered independently. The papers [1], [4] and [7] are some examples that treat specifically local access network design problems. This is the class of problems of our interest in this paper, in such a way that we discard any reference to a considerable body of literature that treats topological design, capacity planning and flow assignment questions at the hub-level of transportation or telecommunication networks.

We also focus on a deterministic problem in which a single hub location is chosen from among a *continuous* set of points. Most of the literature on public utility networks has only a half of century, but the $min - sum$ location problems originated in the 17th century, when Fermat posed the question of, given three points in a plane, find a median point in the plane such that the sum of the distances from each point to the median point is minimized. In the last century, many studies have addressed extensions of the Fermat problem. A remarkable contribution has been done by Alfred Weber, that studied the problem for a general number n of points, also adding weights q_i on each point i to consider customer demand qt that point. The Weber problem locates facilities (medians) at *continuous* locations in the Euclidian plane. We assume in this paper that, given the n customer points, the location of the source node 0 is an optimal solution of the Weber problem. The idea behind the Fermat problem has been introduced in graphs by (Hakimi, 1964), who defined the absolute median as the point on a graph that minimizes the sum of the weighted distances between that point and the vertices of the graph. He allowed this point to lie anywhere along the edges of the graph, but proved that an optimal absolute median is always located at a node of the graph, thus providing a discrete representation of a continuous problem.

Given a set of points on the two-dimensional plane, the problem of finding the shortest connecting network that connects all the points, using only lines joining pairs of points from the given set is one of the nicest and simplest problems in network optimization, and arises in many applications. If the length between every pair of points is positive, the shortest connecting network is clearly a spanning tree, that is called a minimum length spanning tree (*MLST*). This problem is closely related to the local access network design *LAND* problem. This paper shows that, for a sufficiently high value of the fixed over variable cost ratio β/γ, an optimal topology for the *LAND* problem is also an optimal topology for the *MLST* problem. On the other hand, the paper also shows that, for a sufficiently low value of the fixed over variable cost ratio β/γ, a star centered in in source node 0 is an optimal topology for the *LAND* problem. The next two sections formalizes the *LAND* problem and provides theoretical results concerning cost and distance relationships in the problem, while a summary section concludes the paper.

2. A Flow Formulation for Tree Network Design

The problem is to find a minimum cost tree over a graph $G(N, E)$, where N is a set of $n + 1$ nodes and E is a set of m edges. Unless stated otherwise, the graph is complete, with a number of $m = n(n + 1)/2$ edges. The number of selected edges in an optimal tree connecting network is n. The model parameters are provided by one square matrix, D, of order $n + 1$, a demand vector q, of order n, and the two scalars β and γ, both indicating cost per unit of distance. Each element d_{ij} of the symmetric matrix D refers to the distance between nodes i and j, that

is assumed to be equal to the distance between nodes j and i. The diagonal of D has elements $d_{ii} = 0$ and if nodes i and j are not linked by an edge in E then $d_{ij} = \infty$.

Consider the binary variables $x_{(ij)}$, for $i = 0, 1, ..., n-1$ and $j = i+1, ..., n$, such that $x_{(ij)} = 1$ if and only if edge $(ij) \in E$ belongs to an optimal tree design. Consider also the directed flow variables $f_{ij} \geq 0$, for $i = 0, 1, ..., n$ and $j = 1, ..., n$ with $i \neq j$, which specify a single commodity flow between nodes i and j.

A mixed integer linear program for the local access network design (*LAND*) problem is

$$\min \beta \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} d_{ij}\, x_{(ij)} + \gamma \sum_{i=0}^{n} \sum_{j=1}^{n} d_{ij}\, f_{ij} \tag{1}$$

subject to the constraints

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} x_{(ij)} = n \tag{2}$$

$$\sum_{j=1}^{n} f_{0j} = \sum_{h=1}^{n} q_h \tag{3}$$

$$\sum_{h=0}^{n} f_{hi} - \sum_{j=1}^{n} f_{ij} = q_i \;\; \forall\, i = 1, ..., n \tag{4}$$

$$f_{0j} \leq (\sum_{h=1}^{n} q_h)\; x_{(0j)} \;\; \forall\, (0j) \in E \tag{5}$$

$$f_{ij} \leq (\sum_{h=1}^{n} q_h)\; x_{(ij)} \;\; \forall\, (ij) \in E \tag{6}$$

$$f_{ji} \leq (\sum_{h=1}^{n} q_h)\; x_{(ij)} \;\; \forall\, (ij) \in E \tag{7}$$

$$x_{(ij)} \in \{0, 1\} \;\; \forall\, (ij) \in E \tag{8}$$

$$f_{ij} \geq 0 \;\; \forall\, i = 0, ..., n,\; j = 1, ..., n \tag{9}$$

This single-commodity flow formulation is a simplified version of more elaborated models concerning spanning trees or local access network problems ([5],[2], [4], [7]). Remark that the linear programming relaxation of some of these multi-commodity versions provides integer solutions for the problem, but any computational issue concerning this class of problems is out of scope in this paper. Apart notation questions, the essential results that follows neither are dependent from the (*LAND*) problem formulation nor from the used algorithm to solve the problem.

3. Cost and Distances Relationships

3.1 Cost and length definitions

For any feasible solution (x^t, f^t) in the mixed integer linear programming model (1-9) we identify two parts of the objective function:

$$z^t = \beta \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} d_{ij}\, x^t_{(ij)} \tag{10}$$

$$v^t = \gamma \sum_{i=0}^{n} \sum_{j=1}^{n} d_{ij}\, f^t_{ij} \tag{11}$$

in such way that $z^t + v^t$ is the total cost of the designed network. Let $T(N, E^t)$ be the spanning tree corresponding to this solution and linking the origin 0 to all demand nodes $h = 1, ..., n$ ($x_{(ij)} = 1 \; \forall \, (ij) \in E^t$ and $x_{(ij)} = 0 \; \forall \, (ij) \in E - E^t$). Assume that L^t is the total length of the spanning tree $T(N, E^t)$. Let P^t_{0h} be the set of edges in the path from the origin 0 to the demand node h, with l^t_{0h} being the correspondent length, obtained by summing the distances d_{ij} across the edges of P^t_{0h}. Then equations (10) and (11) can be rewritten in terms of trees and paths lengths:

$$z^t = \beta\, L^t \tag{12}$$

$$v^t = \gamma \sum_{h=1}^{n} q_h\, l^t_{0h} \tag{13}$$

3.2 Lower bounds for costs

Two specific feasible solutions, for which we use the indices $t = 0$ and $t = 1$, provides information to determine a lower bound for any feasible solution of the *LAND* problem. We say that $T(N, E^0)$ is the trivial solution of a star configuration, with the source node 0 being directly linked to each of the other nodes $i = 1, ..., n$. And we define $T(N, E^1)$ as being an optimal topology for the *MLST* problem over the graph $G(N, E)$. We call any of the spanning trees $T(N, E^0)$ or $T(N, E^1)$ an extremal solution for the problem.

For $T(N, E^0)$, by definition, we have $l^0_{0h} = d_{0h} = l^{min}_{0h}$ for all $h = 1, ..n$, where l^{min}_{0h} indicates the length of a shortest path from the source 0 to the customer point h, that is of course the straight line between points 0 and h. On the other hand, by definition, we have $L^1 = L^{min}$, where L^{min} indicates the minimum length spanning tree. The following results, that are not proved here to reduce space, are easily shown:

Lemma 1. *Every feasible solution (x^t, f^t), related to an enumerated spanning tree $T(N, E^t)$, has a variable cost not smaller than that of the star configuration $T(N, E^0)$, that is*

$$v^0 \leq v^t \;\; \forall \;\; T(N, E^t).$$

Lemma 2. *Every feasible solution (x^t, f^t), related with an enumerated spanning tree $T(N, E^t)$, has a fixed cost not smaller than that of the minimum length spanning tree $T(N, E^1)$, that is*

$$z^1 \leq z^t \;\; \forall \;\; T(N, E^t).$$

Theorem 3. *The spanning tree minimum length $L^1 = L^{min}$ and the shortest path $l^0_{0h} = d_{0h} = l^{min}_{0h}$ from the origin 0 to each demand node h are such that*

$$\beta\, L^{min} + \gamma \sum_{h=1}^{n} q_h\, d_{0h}$$

is a **lower bound** *for the total cost of any enumerated tree $T(N, E^t)$ related to a feasible solution (x^t, f^t) .*

3.3 Necessary optimality conditions

Let (x^*, f^*), related with a spanning tree $T(N, E^*)$, be an optimal solution for the $(LAND)$ problem (1-7). The following results hold:

Theorem 4. *The total length L^* of a spanning tree associated with an optimal solution (x^*, f^*) must satisfy*

$$L^1 = L^{min} \leq L^* \leq L^0 = \sum_{h=1}^{n} d_{0h}$$

Theorem 5. *If $z^* + v^*$ is the objective function value of an optimal solution (x^*, f^*) then*

$$v^* \leq v^1$$

3.4 Sufficient optimality conditions for extremal solutions

A star configuration centered in 0 is better then any given tree $T(N, E^t)$ if

$$\beta \sum_{h=1}^{n} d_{0h} + \gamma \sum_{h=1}^{n} q_h \, d_{0h} < \beta \, L^t + \gamma \sum_{h=1}^{n} q_h \, l^t_{0h}$$

In particular, for any non-trivial case where $L^0 > L^1$, that is $\sum_{h=1}^{n} d_{0h} > L^{min}$, we can determine the value β' for which the shortest path solution has the same cost of the minimum length spanning tree. This kind of consideration leads to a series of results concerning sufficient optimality conditions for extremal solutions. A detailed specification of these interesting properties is left for our workshop presentation and for a complete version of this paper.

4. Summary

The concept of distance is essential to the objective of cost minimization in public utility networks. This paper puts this concept as the main object to find an optimal geometric structure of a local access network. For a given set of n customer points in a two-dimensional plane, with known distances between all pairs of these points, the fundamental problem addressed here concerns the optimal location of a single source node 0 and an adequate choice of a tree connecting network in order to minimize the total cost to install and to use the distribution network.

Besides the beauty of the mathematical theory associated to cost and distance relationships, the interest in this research topic is explained by the richness and variety of its applications. Among the best known we find the examples of computer and telecommunication networks, logistics of distribution systems, water supply management and electrical energy distribution. The influence of distance plays a major role to optimize the geometric structure and the operational access to all these public utility networks.

Acknowledgments

The author wish to thank for the financial support of CNPq, the brazilian council of scientific research and technological development.

References

[1] Gavish, B. (1982). "Topological design of centralized computer networks – formulations and algorithms", *Networks* 12, 355-377.

[2] Gouveia, L. (1996). "Multicommodity flow models for spanning trees with hop constraints", *European Journal of Operational Research* 95, 178-190.

[3] Luna, H. P. L. Network planning problems in telecommunications. In: Mauricio G. C. Resende; Panos M. Pardalos. (Org.). Handbook of Optimization in Telecommunications. 1ed.New York: Springer, 2006, p. 213-240.

[4] Luna, H. P. L., Ziviani, N. and Cabral, R. (1987). "The telephonic switching centre network problem: Formalization and computational experiments", *Discrete Applied Mathematics* 18, 199-210.

[5] Maculan, N. (1986). "A new linear programming formulation for the shortest s-directed spanning tree problem", *Journal of Combinatorics, Information & Systems Sciences* 11, 53-56.

[6] Magnanti, T. and Wong, R. (1984). "Network design and transportation planning: Models and algorithms", *Transportation Science* 18, 1-55.

[7] Randazzo, C. and Luna, H. P. L. (2001). "A comparison of optimal methods for local access uncapacitated network design", *Annals of Operations Research* 106, 263-286.

DGA 2013, pp. 175 – 179.

A new algorithm for efficient computation of Hausdorff distance in evaluation of digital image segmentation

R. S. Marques,[1] D. A. Machado,[2] G. Giraldi,[2] and A. Conci[1]

[1] *Universidade Federal Fluminense - UFF,* {rmarques,aconci}@ic.uff.br

[2] *Laboratório Nacional de Computação Científica - LNCC,* {danubiad,gilson}@lncc.br

Abstract

Large ground truth databases are necessary to evaluate and validate computer-aided diagnosis systems. Images used for diagnosis purposes usually have their regions of interests segmented as a first step of the patterns recognition procedures. Automatic segmentation of medical images is an open issue in image processing where there is an important need for validation and comparison among new results with available image databases. The limits of such regions of interests are frequently very irregular and to verify the adequacy of several approaches numerical and not only visual techniques must be used. In this paper, a new algorithm for the Hausdorff distance computation in discrete curves and areas is presented. The proposed algorithm finds the exact result in much less computational time than the traditional method. Results of its use on comparison of two automatic segmentation methods for breast infrared images are presented to illustrate the algorithm.

Keywords: Discrete imaging, Hausdorff distance, Digital image segmentation, Ground Truth

1. Introduction

The use of new technologies (as infrared and electrical impedance tomography) can improve early and correct diagnosis, especially if considered in computer-aided diagnosis (CADx) systems [1]. These systems uses artificial intelligence (AI) and mining techniques to improve early diagnosis and it needs classified databases for knowledge acquisition [2]. To accomplish this, the development of databases with proven cases is fundamental and it is the main goal of the project on execution at the Hospital of Fluminense Federal University (HUAP/UFF) aiming to improve the breast diseases detection using infrared (IR) images [3]. This research aims to assist the development of CADx based on a fusion of exams (mammography, ultrasound, MRI, thermography) considering a new way to compare the results of different discrete image segmentation approaches. In this paper we present a new method that allow numerical comparisons among breast segmentations by improving Hausdorff distance calculation algorithm, considering particular aspects of discrete objects (curves and areas) [5]. Moreover, this article describes experiments on real breast images used to evaluate the segmentations methodologies. Results compares two forms of achieve this information using the proposed algorithm and the traditional one. The result of this research aims to assist the generation of diagnostic systems or at least to be a tool to compare biological segmentation results based on discrete images. This work is divided in more three parts: Description of the Hausdorff distance in continuous geometry and its new formulation for discrete geometry (DG); The developed algorithm and some aspects of its implementation; Its results on infrared exams (thermographic images) and

conclusion about this new technique on the 2D discrete curves of the breast boundaries detected from the IR acquisition. Validations were made comparing the real breasts of volunteers submitted at same time to at least two segmentations considered very satisfactory visually and that must be numerically compared.

2. Preliminaries

We begin by describing *Hausdorff metric or distance*. Let D be a closed subset of R^n (continuous space) and S denote the class of all non-empty compact subset of D ([6] p.113). There are some alternative (equivalent) ways of defining *Hausdorff metric* on all non-empty compact subset S, of closed subset D, of R^n. The one presented by Falconer ([6] p.114) considering the δ-*parallel body* of $A \in S$ is very adequate to transform S to a *discrete* $Z_\triangle$ subspace and it is the one used here. We define A_δ , i.e. *the parallel body* of $A \in S$ as:

$$A_\delta = \{x \in D : |x - a| \leq \delta \text{ for some } a \in A\}, \tag{1}$$

We make S into a metric space by defining the distance $d(A, B)$ between two sets A, B to be the least δ such that the δ-*parallel body* of A, A_δ, contains B and the δ-*parallel body* of B, B_δ, contains A (see first image on Figure 1):

$$d(A, B) = \inf\{\delta : A \subset B_\delta \text{ and } B \subset A_\delta\}, \tag{2}$$

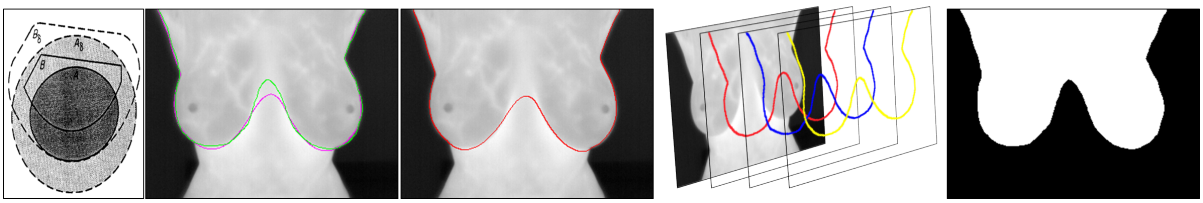


Figure 1: Hausdorff distance between two sets A, B and the δ-parallel body ([6] p.114). Result of a spline based segmentation [5] (pink) for $IR0100$ overlapped with the ground truth (green). Refined result for the same image [11]. Manual segmentation, ground truth composition [9] and ROI defined in a binary black and white version.

Then the two sets A, B is now subsets of the metric space S. Considering the definition of the closure of A and B, denoted by A^- and B^- ([7] p.114) we have

$$d(A^-, B^-) = \inf\{\delta : A^- \subset B_\delta{}^- \text{ and } B^- \subset A_\delta{}^-\} = \inf\{\delta : A \subset B_\delta \text{ and } B \subset A_\delta\} = d(A, B), \tag{3}$$

Moreover by definition of the boundary of A and B : ∂A and ∂B ([7] p.181) we have:

$$d(\partial A, \partial B) = d(A^-, B^-) = d(A, B), \tag{4}$$

It is plausible to consider a discrete version of (4) by replacing A with a discrete version of it, say $A_\triangle$. Allowing a slight abuse of notation, let $\partial A_\triangle$ denote the intersection of $A_\triangle$ with ∂A. Carefully note that this is not the boundary of $A_\triangle$ (in fact, it makes no sense to talk about the boundary of a discrete subset of R^n). The same argument applied to the set B and yields a discrete version $B_\triangle$ of it. In this way, we can think of $A_\triangle$ and $B_\triangle$ as subsets of a discrete version of R^n , say $Z_\triangle$, and so we proceed to compute a discrete version of the *Hausdorff distance* between $A_\triangle$ and $B_\triangle$ based on following discrete version of (2):

$$d(A_\triangle, B_\triangle) = d(\partial A_\triangle, \partial B_\triangle), \tag{5}$$

Although this new expression (5) for discrete images follows from equation (2) and various definitions, we not yet have known it from elsewhere in the literature. In next section, this new expression for Hausdorff distance is used to create a new algorithm to compare binary digital images. Although few examples are presented here, it has been extensively tested for us in the IR segmentation evaluation as will be commented [3].

3. Proposed Algorithm

Using equation (5) instead of the equation (2), we greatly improve any traditional algorithm for calculating the distance between two sets. To illustrate this statement a very common algorithm (brute force) was applied in these two forms and used for images of 2 resolutions, the results can be seen in Table 1. Moreover, this section presents some considerations to (5) that can turn its computation on discrete images even faster. Let $A_\triangle$ be an image of the automatic segmented region of interest (ROI) and let $B_\triangle$ be its ground truth respectively (for instance the pink and green curves in the second image of Figure 1). These images can be represented as binary images, i.e. with white (value 1) representing ROI's pixels and black (value 0) representing the background (Figure 1, right image). In this way, the boundary points of A and B (pink and green curves in the right image of Figure 1) can be defined by a list of connected pixels in a given resolution, i.e.:

$$\partial A_\triangle = \{a_i \in A_\triangle : \exists p_j \in A^c, p_j \in N_8(a_i)\} \text{ and } \partial B_\triangle = \{b_i \in B_\triangle : \exists p_m \in B^c, p_m \in N_8(b_i)\}, \tag{6}$$

where $N_8(p)$ denotes the 8-neighborhood of the discrete point p (or pixel p) ([8] p. 210) and A^c, B^c denotes the complement of sets A and B.

To find $d(\partial A_\triangle, \partial B_\triangle)$ through an exhaustive algorithm, it must be calculated for each $a_k \in A_\triangle$ the distance $|a_k - b_i|$ (for $i = 1...n$, where $n = |B|$). It is possible to simplify this search by testing when the pixel a_k of coordinates (x_k, y_k) has a corresponding pixel in same coordinates in the image $\partial B_\triangle$, that is (x_k, y_k) is equal to value in $\partial B_\triangle$. In this case, the distance of a_k and B equals zero: this means $n - 1$ comparisons will not be accomplished, simplifying the search. The proposed algorithm to calculate (5) is presented in algorithm below (The Hausdorff distance algorithm ([5] *p.*109). To facilitate this calculation the index of the pixels (i.e. x_k, y_k) can be used to compute the distance between them.

$1 : \partial A_\triangle := \{a_i \in A_\triangle : \exists p_j \in A^c, p_j \in N_8(a_i)\}$ and $\partial B_\triangle = \{b_i \in B_\triangle : \exists p_m \in B^c, p_m \in N_8(b_i)\}$
$2 : h := 0;$
$3 : for\ each\ a_k = (x_k, y_k) \in \partial A_\triangle\ do$
$4 :$ $\quad if\ value(b_k) \equiv value(a_k)\ then$
$5 :$ $\quad\quad h := 0$
$6 :$ $\quad else$
$7 :$ $\quad\quad h^2 := |b_i - a_k|^2 = |x_i - x_k|^2 + |y_i - y_k|^2$
$8 :$ $\quad end\ if$
$9 : end\ for$
$10 : return\ h$

This algorithm can be used for the entire discrete binary ROI (Figure 1 the rightmost image) if they are presented on this way. Discrete images in two resolutions (320×240 and 640×480) were tested. The Hausdorff distance was calculated using the entire image ($d(A, B)$) and only its boundary ($d(\partial A_\triangle, \partial B_\triangle)$) with the proposed algorithm and the traditional method. Although the result was the same (exact Hausdorff Distance), the computation time was very different as can be seem in line 2 of Table 1 ([5] *p.*110).

Table 1: Execution time in seconds for same images using force brute and the proposed algorithm [5].

Time	$d(\partial A_{\triangle}, \partial B_{\triangle})320 \times 240$	$d(A, B)320 \times 240$	$d(\partial A_{\triangle}, \partial B_{\triangle})640 \times 480$	$d(A, B)640 \times 480$
Traditional algorithm	9.57	37.78	76.51	610.65
Proposed algorithm	0.75	1.49	3.98	20.6

4. Results on Infrared Exams and Conclusions

Numerical results on ten real breast segmentation using two different approaches were considered for illustrative purposes and presented in Table 2. For the ground truth generation three manual segmented images for each patient were used. They were manually defined using a Samsung Galaxy $P7510$ tablet with stylus pen by a specialist in breast radiology and two trained users (Figure 1). A specialized software was developed for it [9].To facilitate, all three manual segmentations results were combining to a unique ground truth using the voting policy proposed by Li et al. [10]. The first segmentation technique evaluated is based on Quadratic Uniform B-Splines presented by Marques [5]. The results of the second approach were obtained by a refinement of the previous approach using Level Set [11] (post-processing). More details about these techniques can be found in [5, 11]. The second and third image of Figure 1 shows the results of the segmentation methods that was evaluated by the Hausdorff distance. There is not a visual significant difference between the automatic segmentations used, this illustrate the need of numerical comparison. They are then compared by the proposed algorithm (Table 2).

Table 2: Hausdorff distance between the Ground-Truth and segmented images.

Images IR	0100	0149	0213	0756	0973	0990	3416	3743	3748	3825
Splines	12.76	5.00	9.22	15.56	16.97	7.07	16.28	7.28	7.81	28.28
Posproces.	12.73	5.00	9.22	14.87	16.97	7.07	16.28	7.28	7.81	29.00

5. Summary

This paper is concerned with the application of *Hausdorff metric or distance* for imaging. It should be of interest to a broad readership involved in segmentations methods and its validation by using a typical strategy of comparison with ground truth of available databases. This work shows how *Hausdorff metric* computation could have its computation time reduced by using theoretical and numerical techniques. Summing up, the steps are: Binarize the segmented images (continuous closed sets); Subtract the inside, yielding the image boundaries; Discretize the boundaries and calculate the distance between the sets. Validations are made by comparing results of images in two resolutions, using well-known algorithm and ten real breast infrared exams of volunteer who accepted to have their data included in the public database development by the projects that support this work.

Acknowledgments

The authors thank to CAPES for financial support (projects PROENG PE021/2008 and ProCad no. 540/2009) and to UFF-Telemedicine Group an Associated Laboratory of INCT-MACC.

References

[1] Ng, E. Y-K and N. M. Sudharsan. *Computer Simulation in Conjunction with Medical Thermography as an Adjunct Tool for Early Detection of Breast Cancer. BMC Cancer*, 4(17):6, 2004.

[2] M. Moghbel and S. Mashohor. A review of Computer Assisted Detection/Diagnosis (CAD) in breast thermography for breast cancer detection. *Artificial Intelligence Review*, 2011.

[3] PROENG. *Image processing and image analyses applied to mastology.* http://visual.ic.uff.br/en/proeng/.

[4] I. Cheng, C. Flores-Mir, P. Major and A. Basu. *Measuring and evaluating ground truth for boundary detection in medical images. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5889–92, 2008.

[5] R.S. Marques. *Segmentação automática das mamas em imagens térmicas.* M.Sc. thesis, IC-UFF, 2012.

[6] K.J. Falconer. *Fractal Geometry: Mathematical foundations and applications.* Wiley, 1990.

[7] C.S. Kubrusly. *Elements of operator theory.* MIT Press, Cambridge, MA, 1994.Birkhauser, 2001.

[8] A. Conci, E. Azevedo and F.R. Leta. *Computação Gráfica.* Campus/Elsevier, v.2, 2008.

[9] GTMAKER. *Ground Truth Maker.* http://visual.ic.uff.br/en/proeng/software.php, 2012.

[10] X. Li, B. Aldridge, R. Fisher and J. Rees. *Computer Simulation in Conjunction with Medical Thermography as an Adjunct Tool for Early Detection of Breast CancerEstimating the ground truth from multiple individual segmentations incorporating prior pattern analysis with application to skin lesion segmentation. IEEE International Symposium on Biomedical Imaging From Nano to Macro*, pages 1438–1441, 2011.

[11] D.A. Machado. *Segmentação de Imagens via Método dos Conjuntos de Níveis e Derivada Topológica.* M.Sc. thesis, LNCC, 2012.

DGA 2013, pp. 181 – 185.

Does the packing radius depend on the distance? The Case for Poset Metrics

Rafael Gregorio Lucas D'Oliveira[1] and Marcelo Firer[2]

[1]*IMECC-UNICAMP, Universidade Estadual de Campinas, CEP 13083- 859, Campinas, SP, Brazil,* rgldoliveira@gmail.com

[2]*IMECC-UNICAMP, Universidade Estadual de Campinas, CEP 13083- 859, Campinas, SP, Brazil,* mfirer@gmail.com

Abstract Until this work, the packing radius of a poset code was only known in the cases where the poset was a chain, a hierarchy, a union of disjoint chains of the same size, and for some families of codes. Our objective is to approach the general case of any poset and any code. To do this, we will divide the problem into two parts.

The first part consists in finding the packing radius of a single vector. We will show that this is equivalent to a generalization of a famous NP-hard problem known as "the partition problem". Then, we will review the main results known about this problem giving special attention to the algorithms to solve it. The main ingredient to these algorithms is what is known as the differentiating method, and therefore, we will extend it to the general case.

The second part consists in finding the vector that determines the packing radius of the code. For this, we will show how it is sometimes possible to compare the packing radius of two vectors without calculating them explicitly.

Keywords: Error Correction Codes, Packing Radius, Partitioning Problems, Poset Codes

1. Introduction

Let (V, d) be a finite metric space and let $C \subset V$ be a nonempty subset. The minimal distance of C is

$$d(C) = \min\{d(x, y) : x, y \in C, x \neq y\}$$

and the packing radius of C is

$$R_d(C) = \max\{R : B(x, R) \cap B(y, R) = \emptyset, x, y \in C, x \neq y\}$$

where $B(x, R)$ is the ball centered at x with radius R, i.e.

$$B(x, R) = \{y \in V : d(x, y) \leq R\}.$$

The question we pose in this work is to analyze the relation between these quantities, the minimum distance and the packing radius, in a specific context, that of the Theory of Error Correcting Codes, where $V = \mathbb{F}_q^n$ is a $n-$vector space over a finite field with q elements, C is a linear subspace and d belongs to a family of metrics that assumes the integer values between 0 and $n = \dim V$. If we denote by $\lfloor x \rfloor$ the floor function, it is straightforward to show that

$$\left\lfloor \frac{d(C) - 1}{2} \right\rfloor \leq R_d(C) \leq d(C) - 1$$

(the floor function is only a consequence of the values of $d(\cdot, \cdot)$ being integers).

Due to the importance of this question for the Theory of Error Correcting Codes, we keep the notation used in this context. The usual metric used in this context, the Hamming metric d_H, is defined as the cardinality $|\{i; x_i \neq y_i\}|$, where $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are elements in $\mathbb{F}_q^n$. In this case, it is well known that

$$R_{d_H}(C) = \left\lfloor \frac{d_H(C) - 1}{2} \right\rfloor.$$

In this work we will consider the problem of finding the packing radius in the case of poset metrics. These metrics where first introduced by Brualdi et al. [1] generalizing on the work of Niederreiter [9]. An interesting property of these metrics is that the packing radius is not necessarily determined by the minimum distance. Until this work, to the authors' knowledge, the packing radius of a poset code was only known in the following cases: chain posets [3], hierarchical posets [2], disjoint union of chains of the same size (L. Panek, M. Muniz, M. Firer, personal communication), and for some families of codes [4]. We will approach the general poset case. To do this we will divide our problem in two.

The first part consists in determining the packing radius of a single vector. We will see that this is equivalent to solving a generalization, which we will call "the poset partition problem", of a famous NP-hard problem known as "the partition problem". We will then take a look at one of the fastest known algorithms for solving the partition problem (in some cases) and generalize it to the poset partition problem. The first time the problem of finding the packing radius of a poset code was identified, in some sense, as a partitioning problem was in [5].

The second part consists in finding which code-word determines the packing radius of the code. To do this we will show how sometimes it is possible to compare the packing radius of two vectors without calculating them explicitly.

2. The Poset Metric

Let $[n] = \{1, 2, \ldots, n\}$ be a finite set and $\preceq$ be a partial order on $[n]$. We call the pair $P = ([n], \preceq)$ a poset and often identify P with $[n]$. An ideal in P is a subset $J \subseteq P$ with the property that if $x \in J$ and $y \preceq x$ then $y \in J$. The ideal generated by a subset $X \subseteq P$ is the smallest ideal containing X and is denoted by $\langle X \rangle$. A poset is called a chain if every two elements are comparable, and an anti-chain if none are. The length of an element $x \in P$ is the cardinality of the largest chain contained in $\langle \{x\} \rangle$.

Let q be the power of a prime, $\mathbb{F}_q$ the field with q elements and $\mathbb{F}_q^n$ the vector space of n-tuples over $\mathbb{F}_q$. We denote the coordinates of a vector $x \in \mathbb{F}_q^n$ by $x = (x_1, x_2, \ldots, x_n)$.

A poset $P = ([n], \preceq)$ induces a metric d_P, called the P-distance, in $\mathbb{F}_q^n$ defined as

$$d_P(v, w) = |\langle supp(v - w) \rangle|$$

where $supp(x) = \{i \in [n] : x_i \neq 0\}$. The distance $\omega_P(v) = d_P(v, 0)$ is called the P-weight of v.

Note that if P is an anti-chain then d_P is the Hamming distance. Because of this, when P is an anti-chain we will denote it by H.

Given a linear code (subspace) $C \subseteq \mathbb{F}_q^n$ and a poset $P = ([n], \preceq)$, we denote the minimum distance of C as $d_P(C)$ and the packing radius of C as $R_{d_P}(C)$. We remark that, since d_P is translation invariant, if we define the P-weight as $\omega_P(x) = d_P(x, 0)$ then $d_P(C) = min\{\omega_P(v) : v \in C - \{0\}\}$. Since C is linear, $z = x - y \in C$ and therefore the packing radius is the largest positive integer such that

$$B_P(0, R_{d_P}(C)) \cap B_P(z, R_{d_P}(C)) = \varnothing$$

for every $z \in C - \{0\}$.

3. The Packing Radius of a Vector

We begin this section by defining the packing radius of a vector.

Definition 1. *Let $x \in \mathbb{F}_q^n$ and d be a metric over $\mathbb{F}_q^n$. The **packing radius of** x is the largest integer r such that*

$$B(0,r) \cap B(x,r) = \emptyset$$

and is denoted by $R_d(x)$.

Next, we show that the packing radius of a linear code is the smallest of the packing radii of its code-words.

Proposition 1. *Let $C \subseteq \mathbb{F}_q^n$ be a linear code and d a metric over $\mathbb{F}_q^n$. Then,*

$$R_d(C) = \min_{x \in C - \{0\}} R_d(x).$$

Thus, to find the packing radius of a linear code, we need to find the code-word with the smallest packing radius, which we will call the packing vector of the code. We then approach the problem of finding the packing radius of a vector proving the following result:

Theorem 1. *Let P be a poset and $v \in \mathbb{F}_q^n$. Then,*

$$R_{d_P}(v) = \min_{A,B \subseteq M_{supp(v)}} \{max\{|\langle A \rangle|, |\langle B \rangle|\}\} - 1,$$

where (A,B) is a partition of $M_{supp(v)}$, the set of maximal elements of $supp(v)$.

Therefore, the packing radius of a vector is a property of its support. We then show that the problem can be interpreted as a poset partitioning problem.

Definition 2. *Let P be a poset and M_P be the set of its maximal elements. We define the **packing radius of the poset** P as*

$$R(P) = \min_{A,B \subseteq M_P} \{max\{|\langle A \rangle|), |\langle B \rangle|\}\} - 1,$$

where (A,B) is a partition of M_P.

Applying Theorem 1 to the definition we have that the packing radius of a vector v is

$$R_{d_P}(v) = R(\langle supp(v) \rangle).$$

The problem of finding the packing radius of a vector is then equivalent to the problem of finding the packing radius of a poset, which we will call the **poset partition problem**. This problem is a generalization of the famous NP-hard problem known as "the partition problem".

4. The Partition Problem

The **partition problem** is defined as follows: Given a finite list S of positive integers, find a partition (S_1, S_2) of S that minimizes

$$\max \left\{ \sum_{x \in S_1} x, \sum_{y \in S_2} y \right\}.$$

This is equivalent to minimizing the **discrepancy**

$$\Delta(S_1, S_2) = \left| \sum_{x \in S_1} x - \sum_{y \in S_2} y \right|.$$

This problem is of great importance both from the practical and theoretical point of view. In [7], Karp proves that it is NP-hard.

5. The Poset Partition Problem

In the poset partition problem we must minimize not the discrepancy, but what we call the discordancy.

Definition 3. *Let P be a poset and (A, B) a partition of M_P, the maximal elements of P. We define the* ***discordancy*** *between A and B as*

$$\Lambda(A, B) = ||\langle A\rangle| - |\langle B\rangle|| + |\langle A\rangle \cap \langle B\rangle|,$$

and the ***minimum discordancy*** *of P as*

$$\Lambda^*(P) = \min_{X \sqcup Y = M_P} \Lambda(X, Y).$$

The packing radius of a poset can then be written in terms of its minimum discordancy.

Theorem 2. *Let P be a poset of size n. Then, the packing radius of P is*

$$R(P) = \frac{n}{2} + \frac{\Lambda^*(P)}{2} - 1.$$

One of the main heuristics used in solving the partition problem is known as the KK (Karmarkar-Karp) heuristic [6], or as the differencing heuristic. We can generalize this heuristic for the poset partition problem and also generalize one of the best known algorithms, for some cases, that heavily uses the KK heuristic known as the CKK (Complete Karmarkar-Karp) algorithm [8].

6. Finding the Packing Vector

To find the packing radius of a poset code we need to find its packing vector, the code-word with minimum packing radius. One way to do this would be to calculate the packing radius of each code-word, but as we have seen that would be a big problem since we would have to solve a poset partition problem for each code-word. We can show some ways in which we can sometimes compare the the packing radius of two posets without explicitly determining them.

References

[1] Richard A. Brualdi, Janine S. Graves, and K. Mark Lawrence. Codes with a poset metric. *Discrete Mathematics*, 147:57–72, 1995.

[2] Luciano V. Felix and Marcelo Firer. Canonical-systematic form of hierarchical codes. Advances in Mathematics of Communication, 2011. to appear.

[3] Marcelo Firer, Luciano Panek, and Marcelo Muniz Silva Alves. Classification of niederreiter-rosenbloom-tsfasman block codes. *IEEE Transactions on Information Theory*, 56:5207–5216, 2010.

[4] Marcelo Firer, Luciano Panek, and Laura Rifo. Coding in the presence of semantic value of information: Unequal error protection using poset decoders, 2011.

[5] Jong Yoon Hyun and Hyun Kwang Kim. The poset structures admitting the extended binary hamming code to be a perfect code. *Discrete Mathematics*, 288:37–47, 2004.

[6] Narendra Karmarkar and Richard M. Karp. The differencing method of set partitioning. Technical report, Computer Science Division (EECS), University of California, Berkley, 1982.

[7] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[8] Richard E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106:181–203, 1998.

[9] Harald Neiderreiter. A combinatorial problem for vector spaces over finite fields. *Discrete Mathematics*, 96:221–228, 1991.

DGA 2013, pp. 187 – 191.

Distance-Based Imputation on Classification Problems with Missing Features

Mirlem R. Ribeiro[1] and Eulanda M. Dos Santos[1]

[1]*Federal University of Amazonas, Manaus, Brazil,*
mirlem@ifam.edu.br
emsantos@icomp.ufam.edu.br

Abstract This paper presents a comparison between two different imputation methods, mean and k Nearest Neighbors (kNN), applied to classification problems with missing features. This comparison is conducted using two classification methods: Decision Tree and kNN classifier. The former is an unstable classifier, while the latter is stable. It is shown that the distance-based imputation method (kNN) is less prone to introduce data distortion, leading to higher recognition rates.

Keywords: Missing Features, Imputation Methods, Classification Problems.

1. Introduction

Classification methods are learning algorithms used to solve tasks for which the design of software using traditional programming techniques is difficult. Biometric recognition, filter for electronic mail messages and DNA recognition are examples of these tasks. Several different learning algorithms have been proposed in the literature such as Decision Tree, kNN, Neural Networks, Support Vector Machines, etc. Considering a supervised classification problem with the following set of class labels $\Omega = \{\omega_1, \omega_2 \ldots, \omega_c\}$, samples $\mathbf{x}_{i,t}$ contained in a training dataset are used by learning algorithms to the design of a robust well-suited classifier to the problem concerned. Then, this classifier is used to predict the label of the test samples $\mathbf{x}_{i,g}$ contained in a test dataset, focusing on estimating the generalization performance of the trained classifier. Each training sample $\mathbf{x}_{i,t}$ is an n-dimensional vector $\mathbf{x}_{i,t} = [x_1, ..., x_n]^T \in R^n$, where the real space R^n is called feature space. Traditionally, it is assumed that the test samples are also n-dimensional vectors $\mathbf{x}_{i,g} = [x_1, ..., x_n]^T \in R^n$. Nonetheless, several real-world applications may be prone to missing features on test data set due to bad sensors, data corruption, refusal of respondents to answer certain questions, failed pixels, and others.

Even though discarding all instances with missing features may be used to cope with such a problem, this is not a suitable solution since is not frequently possible to reject to take a decision in real applications. Consequently, several methods have been proposed in the literature to treat missing features. According to Garcia-Laencina et al. [2], the methods for pattern classification with missing features may be divided into three groups: (1) model-based procedures; (2) machine learning-based methods; and (3) missing data imputation. In the first group, a model defining the distribution of the data is constructed using strategies like expectation-maximization. In the second group, missing features are dealt with directly by a classifier, for instance Decision Tree [3]. Finally, imputation methods focus on substituting missing values with meaningful estimates.

Imputation methods are divided into statistical and machine learning-based imputation[2]. Mean and multiple imputation are examples of statistical methods, while kNN and Neural Networks are examples of imputation based on machine learning. The second group is assumed to outperform statistical methods [4], especially mean. However, imputation introduces data distortion, whatever the method used to perform it. In this paper, we show that a distance-based imputation method (kNN) introduces less data distortion than a statistical method (mean). This comparison is conducted on five different databases, that represent five different classification problems, and using two different classifiers: kNN and Decision Tree (DT), stable and unstable classifier respectively. It is important to take into account the distinction between *unstable* or *stable* classifiers [1]. The first group, for instance DT and Neural Networks, is strongly dependent on the training samples, while the second group, classifiers like kNN and Fischer linear discriminant, is less sensitive to changes on the training dataset.

This paper is organized as follows. Section 2 presents research work related to this paper. Then, the parameters employed in the experiments and the results obtained are presented in section 3. Conclusions and suggestions for future work are discussed in section 4.

2. Related Work

In [4], Batista and Monard investigated four imputation methods, namely mean, kNN and internal missing features treatment strategies used by two DT algorithms. Missing data was inserted completely at random (MCAR) in the following percentages: 10%, 20%, 30%, 40%, 50% and 60%. The authors concluded that kNN can outperform all the other three methods. However, only DT was used as a classifier to measure the impact of imputation methods in its performance.

Ding and Ross [5] compared the following four groups of imputation methods: kNN, likelihood-based methods, Bayesian-based methods and multiple imputation, applied to the biometric fusion problem. Using MCAR, 10% and 25% of missing rates were generated for the test set. Their results indicated that kNN was better than the other methods investigated. It is important to mention that a technique for score-level fusion, instead of a classifier, was employed to classify samples from the test set.

In [2], four missing features estimation techniques have been compared: kNN imputation, self-organizing map (SOM) imputation, Multylayer Perceptron (MLP) imputation, and the expectation-maximization algorithm. The following missing rates were inserted based on MCAR: 5%, 10%, 20%, 30% and 40%. Taking into account that three different databases were investigated, the authors concluded that there was not a unique best method for all classification domain tested. Again, only one classifier was used to measure the effect of missing features estimation techniques, that is an artificial neural network.

Finally, Branden and Verboven [6] have compared original kNN, a modified version of kNN, an iterative procedure imputation method, a Bayesian-based method and a sequential imputation technique in three real databases. They have also proposed an imputation method. Their objective was to evaluate how all these methods handle outliers in the data set. They tested seven different percentages of missing rates introduced by MCAR: 1%, 3%, 5%, 10%, 15% , 20% and 30%. The results showed that their proposed method outperformed the other imputation methods employed, due to the fact that this method was designed to be robust to outliers. Only one classifier (distance-based) was used in their experiments, similar to previous works.

In this paper, two imputation methods are investigated, mean and kNN, using two different classification methods: DT (unstable classifier) and kNN (stable classifier). Mean consists of replacing the missing feature by the mean or mode, of all known values of that feature. In a classical kNN imputation, each missing feature is completed by taking an average (or mode) of

the corresponding values of the k nearest samples. A distance function, for instance Euclidean distance, is considered as the similarity measure.

3. Experiments and Discussion

Experiments have been carried out to verify whether or not the distance-based imputation method helps to reduce data distortion in both unstable and stable classifiers. We used five databases in our experiments. It is important to note that the number of features was taken into account when selecting the databases for our experiments, since the chosen databases range from relatively high-dimensional feature spaces to small feature spaces, as it is shown in Table 1.

Even though problems with missing features are frequently detected in real applications, few databases containing real classification problems with missing features are available in the literature. Due to this limitation, missing features were artificially implanted into the test sets of the databases investigated in our experiments. The following percentages of missing features were introduced based on MCAR: 2.5%, 5%, 7.5%, 10%, 15%, 25%, 35% and 45%. The smallest missing rates were not used in databases with small feature spaces, for instance, Feltwell does not have 2.5%, 5%, while Ship, has not 2.5%, 5%, 7.5% of missing features, since there was not enough features to obtain all missing rates.

Table 1: Specifications of the databases used in the experiments.

Dataset	**Number of classes**	**Number of features**	**Training Dataset**	**Validation Dataset**	**Test Dataset**
Dna	3	180	1300	700	1186
Feltwell	5	15	4376	2188	4380
NIST	10	132	5000	10000	68089
Ship	8	11	1020	508	1017
Texture	11	40	3080	1100	1320

Table 2: Error rates obtained using the unstable DT classifier on comparing mean and kNN imputation methods.

Database	**Imputation**	**0%**	**2.5%**	**5%**	**7.5%**	**10%**	**15%**	**25%**	**35%**	**45%**
DNA	mean	22.30	23.50	25.50	25.70	25.70	28.80	32.90	34.80	37.40
	kNN	22.30	23.20	25.50	25.50	25.70	28.40	32.80	34.80	37.50
Feltwell	mean	17.50	-	-	23.60	28.50	34.50	39.10	43.00	47.90
	kNN	17.50	-	-	17.60	17.80	18.20	18.90	19.10	20.00
NIST	mean	10.30	18.40	24.10	28.90	35.30	43.20	55.50	65.70	68.80
	kNN	10.30	10.30	10.40	10.30	10.40	10.50	10.60	10.90	11.10
Ship	mean	10.90	-	-	-	21.60	31.40	40.30	49.00	54.90
	kNN	10.90	-	-	-	12.40	13.60	17.20	18.50	22.90
Texture	mean	9.70	13.80	18.90	22.70	25.50	30.20	41.60	53.80	61.70
	kNN	9.70	9.10	9.30	9.60	9.30	9.50	8.20	8.30	8.50

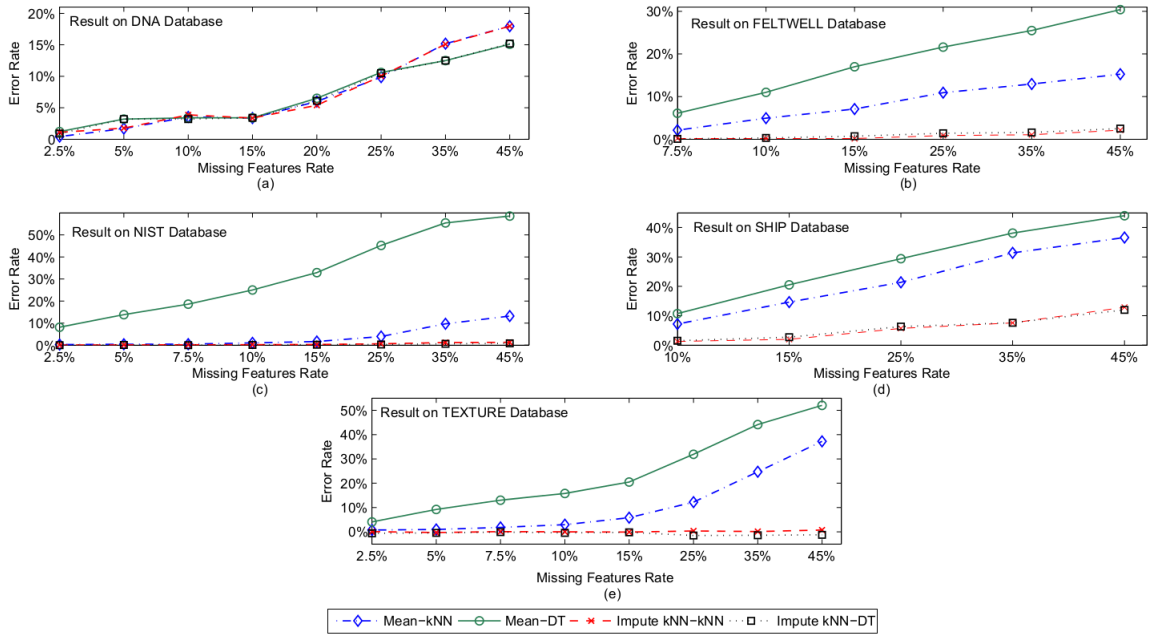


Figure 1: Error rates obtained on experiments comparing kNN and mean imputation methods using DT and kNN classifiers.

Especially noteworthy is the fact that the kNN imputation method may be critically affected by values of its parameter k (number of neighbors), and distance functions. We used k = 5 for kNN imputation by fine-tuning this parameter using the validation data sets. Euclidean distance was employed as distance measure. Experimental tests were also conducted to set up the k value to kNN classifier. The best results were obtained when using k = 1. Finally, DT does not need any parameter to be set.

The obtained results are summarized in Table 2 for DT and in Table 3 for kNN classifier. These tables show the error rates attained when varying the missing rates. In Figure 1, plots of the error rates reached by both kNN and DT classifiers versus missing rates are illustrated to better compare the investigated methods.

Based on these results, it may be observed that:

1. Both imputation methods introduced high level of data distortion, since error rates increased as the missing features rates increased. However, mean introduced more data distortion than kNN impute, especially when DT was employed as classification method. These results were expected since unstable classifiers are very sensitive to small changes on the data.

2. The distance-based imputation method is more stable as the missing rates increase.

3. Although this paper is not focused on comparing classification methods in terms of performance, our results indicate that kNN is better than DT, as well as more robust to missing features.

Table 3: Error rates obtained using the stable kNN classifier on comparing mean and kNN imputation methods.

Database	**Imputation**	**0%**	**2.5%**	**5%**	**7.5%**	**10%**	**15%**	**25%**	**35%**	**45%**
DNA	mean	15.35	15.77	17.03	18.89	18.72	21.33	25.21	30.52	33.31
	kNN	15.35	16.44	17.12	19.22	18.72	20.74	25.38	30.44	33.31
Feltwell	mean	16.53	-	-	18.65	21.48	23.63	27.44	27.47	31.78
	kNN	16.53	-	-	16.74	16.74	16.64	17.40	17.60	18.65
NIST	mean	4.57	4.80	4.96	5.02	5.57	6.22	8.51	14.22	17.74
	kNN	4.57	4.60	4.65	4.68	4.47	4.88	5.19	5.73	5.83
Ship	mean	12.39	-	-	-	19.57	27.04	33.73	43.76	48.97
	kNN	12.39	-	-	-	13.67	14.36	18.09	19.96	25.27
Texture	mean	1.52	2.27	2.50	3.33	4.47	7.35	13.71	26.21	38.71
	kNN	1.52	1.59	1.21	1.59	1.52	1.44	1.89	1.67	2.27

4. Conclusion

In this paper we have presented an experimental study on comparing two imputation methods, mean and kNN, using two different classification methods: DT (unstable classifier) and kNN (stable classifier). The experiments demonstrated that the distance-based imputation method (kNN), introduces less data distortion since both classifiers present higher performance when using kNN imputation. Moreover, the unstable classifier is more prone to data distortion introduced by imputation.

References

[1] L.I. Kuncheva and M.Skurichina and R.P.W. Duin An Experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, 3(4):245-258, 2002.

[2] P.J. Garcia-Laencina and J.L. Sancho-Gomez and A.R. Figueiras-Vidal Pattern classification with missing data: a review. Neural Computing & Applications, 12(2):263-282, 2010.

[3] M. Saar-Tsechansky and F. Provost Handling Missing Values when Applying Classification Models *Journal of Machine Learning Research*, 8:1625-1657, 2007.

[4] G. E. Batista and M. C. Monard An analysis of four missing data treatment methods for supervised learning *Applied Artificial Intelligence*, 17(5-6):519-533, 2003.

[5] Y. Ding and A. Ross A comparison of imputation methods for handling missing scores in biometric fusion *Pattern Recognition*, 45(2012):919-933, 2012.

[6] K.V. Branden and S. Verboven Robust data imputation *Computational Biology and Chemistry*, 33(2009):07-13, 2009.

DGA 2013, pp. 193 – 197.

Proteins Structure Determination with Imprecise Distances*

Ivan Sendin[1] and Siome Klein Goldenstein[2]

[1]*Dept. of Computer Science-CAC, Federal University of Goias, Catalao, Brazil,* sendin@catalao.ufg.br

[2]*Institute of Computing, IC/Unicamp, Campinas, Brazil,* siome@ic.unicamp.br

Abstract The Molecular Distance Geometry Problem is related to protein structure determination using Nuclear Magnetic Resonance information which is imprecise distances of some proteins atoms. Most current methods available to solve this problem work with exact distances. We propose three new methods to propagate uncertainty: using particles, using affine forms and hybrid affine-particles. We use these new methods to propagate uncertainty and determine the protein backbone using NMR like information.

Keywords: proteins structure, uncertainty propagation affine arithmetic, particles

1. Introduction

Proper knowledge of three-dimensional protein structure is a major step in many bioinformatic tasks. On important method to obtain protein structure is the Nuclear Magnetic Resonance(**NMR**)[15] . This process can detect the interaction between pairs of atoms near to each other. So the information given by an NMR experiment is an imprecise distance of some pairs of atoms [6].

The computational problem to determine a protein structure from inter-atomic distances is the **Molecular Distance Geometry Problem (MDGP)**[10]. Usually, we view this problem as a graph problem, where each atom is mapped to one vertex and the edges are the known inter-atomic distances, so this problem is also called **graph embedding problem**. The problem to decide if a graph can be embedded in some k-dimensional space is known to be NP-Complete, even for one dimensional case [13, 12]. For a complete graph with exact distances, this is a trivial problem. Dong and Wu[4] presented the Geometric Build-Up(GBU) algorithm that uses a sufficient dense graph with exact distances to iteratively build a solution for the problem in polynomial time.

Most current methods used to address MDGP use exact distances or an optimization process, like Simulated Annealing [9]. In this work, we introduce the use of particles and present a new hybrid method to propagate uncertainty. Applied to GBU, we can reconstruct a protein using a sparse graph with intervalar distances.

*This research was partly supported by CAPES and FAPESP.

1.1 Uncertainty Propagation

Uncertainty representation and propagation is an important field in information theory [8].In this work uncertainty means imprecise information, i.e. the unknown true value lies in an interval. An uncertainty propagation method should represent the uncertainty of each system state and control the uncertainty growth: if the uncertainty grows too much the information can be useless.

We will use two well known methods for uncertainty propagation: Particles and Affine Forms. Also, we will introduce a new hybrid method. All three methods will be applied to the GBU algorithm and tested in protein structure determination.

Particles. Particles is a non-parametric uncertainty representation method [14]. Modelling with particles is straightforward, a set of samples - called particles - is created for each unknown value and computation is applied on these particles.

This approach is interesting because the computational framework is the same as that used on exact values, the selection, filtering and optimization already available can be applied over particles. In this work, we will use two methods to control particles:

Selection To control the amount of uncertainty to be propagated a subset of particles is selected to represent one state. This selection is performed using Mahalanobis Distance [11].

Sample Importance Ressample Using a problem dependent scoring function, the score of each particle is calculated and this score is used to determine the propagation probability of each particle [2].

Affine Forms. A partially known value $\hat{x}$ is defined by its central value and symbolic sum of noise terms

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i e_i,$$

with $x_i \in \mathcal{R}$ and $e_i \in [-1, 1]$. The unknown terms e_i models the uncertainty of one affine form. To measure the uncertainty of one affine form $\hat{x}$, one can use the *range* function:

$$range(\hat{x}) = \sum_{i=1}^{n} |x_i|.$$

In [3], an Affine Arithmetic (**AA**) is defined, arithmetical operations with real numbers are trivial, other mathematical operations require approximations that create new unknowns values and enlarge the range. One important feature of AA is that noise terms can be cancelled:

$$\hat{x} - \hat{x} = 0.$$

Affine arithmetic ensures that the resulting affine form contains the true value provided that the operands contain the true value. This property, useful for reliable computing, in general is not desirable because it causes the growth of noise range, because unlikely regions are reached by an affine form.

Another drawback of affine representation is its distance to exact representation, that makes the optimization process harder to design and implement.

One can create an exact representation from affine forms sampling values for unknown terms and replacing the sampled values in all affine forms. As the affine correlation is held on unknown sharing, this sampling process can create consistent values for a set of affine forms. Also, it is possible to create a particles representation using this method.

Hybrid Method. Here, we introduce a hybrid method for uncertainty propagation. Like in [7] and in [5], the uncertainty is represented both in parametrical and non-parametrical forms. Our method starts with an affine representation of the problem. Then a exact representation is obtained sampling values for the unknown. The sampling process is repeated and a set of exact instances is created. Now these instances are filtered and optimized (as seen on Section 1.1). We expect that this process will produce narrower limits and use those particles to control the affine forms.

2. Computational Experiments

Three versions of the GBU were created to propagate uncertainty: particles-GBU, affine-GBU and a hybrid-GBU. For particles and hybrid method, at each GBU step we create 60 particles for each state. The uncertainty is controlled as follows: the SIR process uses a quadratic penalty function, and is repeated until the average score is stabilized, and a range that contains 85% of the particles is propagated. After the proteins is determined an interval version of Stochastic Embedding Proximity [1] improves the final structure.

2.1 Dataset and Distances Determination

We obtained all NMR proteins structures available in October 2012 at the PDB bank. As the proposed method uses covalent and C_α distances (see below), we are able to use only proteins whose distances were well defined in PDB bank, making 367 proteins.

The distances used in the tests were determined as follows:

1. **RMN-like distances** With atoms separated up to 5 Å , we use a intervalar distance: 2 to 3 Å , 3 to 4 Å and 4 to 5 Å ,in accordance with the observed real distance;
2. **Molecular Geometry distances** For atoms separated by one or two covalent bonds and for consecutive C_α its exact distance is used;

2.2 Results

The affine GBU method did not work: this method does not control the uncertainty, the range grows too fast and the computation does not work. The results for particles and hybrid methods are summarized on Table 1. The results are grouped by the size of the protein backbone, and we show the percentage of distance restraints satisfied and the RMSD to the original protein. In Figure 1 we show the result for the 2gp8 protein.

Table 1: First the average percentage of distance restraints satisfied by the reconstructed protein and, in parentheses, the average RMSD to the original protein, in Angströms.

Method/Backbone Size	50	100	150	200	>200
Particles	65,9 (2,8)	68,4 (4,6)	63,0 (7,2)	64,5 (8,4)	63,2 (10,5)
Hybrid	73,2 (3,2)	73,8 (4,6)	62,3 (6,7)	63,7 (8,1)	64,2 (9,9)

3. Conclusions

In this work we presented three methods to build protein structures using imprecise inter-atomic distances. The pure affine approach did not work. The statistical uncertainty propagation -

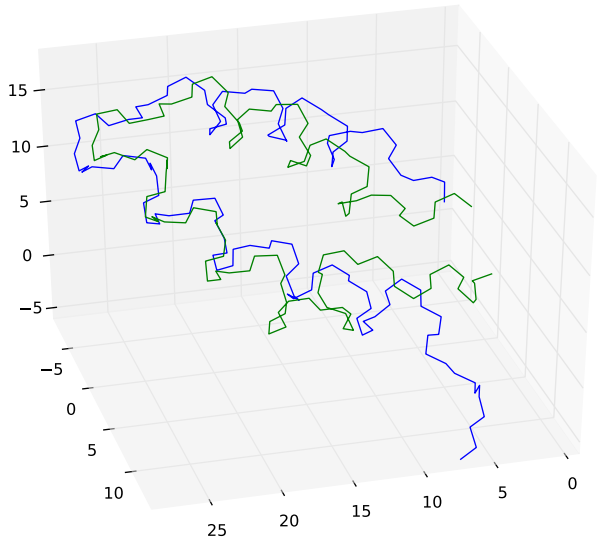


Figure 1: The alignment of 2gp8 protein. In blue, the reconstructed protein using the hybrid method, aligned with the original one, in green.

provided by particles selection and SIR filtering - is efficient to control the uncertainty enabling the particles and the hybrid methods to determine the protein structure.

References

[1] D. Agrafiotis. Stochastic Proximity Embedding. *Journal of Computational Chemistry*, 24(10):1215–1221, 2003.

[2] J. Carpenter, P. Clifford, and P. Fearnhead. An Improved Particle Filter for Non-linear Problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[3] Luiz H. de Figueiredo and Jorge Stolfi. *Self-Validated Numerical Methods and Applications*. Brazilian Mathematics Colloquium monographs. IMPA/CNPq, Rio de Janeiro, Brazil, 1997.

[4] Qunfeng Dong and Zhijun Wu. A Geometric Build-Up Algorithm for Solving the Molecular Distance Geometry Problem with Sparse Distance Data. *Journal of Global Optimization*, 26:321–333, 2003.

[5] Leyza Baldo Dorini and Siome Klein Goldenstein. Unscented feature tracking. *Computer Vision and Image Understanding*, 115(1):8–15, 2011.

[6] P. Guntert. Structure calculation of biological macromolecules from NMR data. *Quarterly reviews of biophysics*, 31(2):145–237, 1998.

[7] Simon J. J. and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *SPIE*, 1997.

[8] G.J. Klir. *Uncertainty and information: foundations of generalized information theory*. Wiley-IEEE Press, 2006.

[9] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. Recent advances on the discretizable molecular distance geometry problem. *European Journal of Operational Research*, 219:698–706, 2012.

[10] C. Lavor, L. Liberti, and a. Mucherino. On the solution of molecular distance geometry problems with interval data. *BIBMW*, pages 77–82, 2010.

[11] P.C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[12] J. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proceedings of the 17th Allerton Conference on Communication, Control, and Computing*, pages 480–489, 1979.

[13] J. Saxe. Two Papers on Graph Embedding Problems. Technical Report 10-102, Department of Computer Science, Carnegie-Mellon University, 1980.

[14] L. Wasserman. *All of nonparametric statistics*. Springer-Verlag New York Inc, 2006.

[15] D.M. Webster. *Protein Structure Prediction: Methods and Protocols*. Methods in Molecular Biology. Humana Press, 2000.

DGA 2013, pp. 199 – 203.

Multicoloring of cannonball graphs

Petra Šparl,[1,2] Rafał Witkowski,[3] and Janez Žerovnik[4,2]

[1]*FOV, University of Maribor, Slovenia,* petra.sparl@fov.uni-mb.si

[2]*Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia*

[3]*Adam Mickiewicz University, FMCS, Poznań, Poland,* rmiw@amu.edu.pl

[4]*FME, University of Ljubljana, Slovenia,* janez.zerovnik@fs.uni-lj.si

Abstract In the frequency allocation problem, we are given a cellular telephone network whose geographical coverage area is divided into cells, where phone calls are serviced by assigned frequencies, so that none of the pairs of calls emanating from the same or neighboring cells is assigned the same frequency. The problem is to use the frequencies efficiently, i.e. minimize the span of frequencies used. The frequency allocation problem can be regarded as a multicoloring problem on a weighted hexagonal graph, where each vertex knows its position in the graph. We can generalize this problem into higher dimension. In this paper we present algorithm for multicoloring so called cannonball graphs.

1. Introduction

A fundamental problem that appeared in the design of cellular networks is to assign sets of frequencies to transmitters in order to avoid unacceptable interferences. The number of frequencies demanded at a transmitter may vary between transmitters. The problem appeared in the sixties and was soon related to multicoloring of graphs (see [2]). Besides the mobile telephony there are several applications of frequency assignment including radio and television broadcasting, military applications, satellite communication and wireless LAN (see [1]). A sizable part of theoretical studies is concentrated on the simplified model when the underlying graph which has to be multicolored is a subgraph of triangular lattice. This is a natural choice because it is well known that hexagonal cells provide a coverage with optimal ratio of the distance between centers compared to the area covered by each cell. Such graphs are called hexagonal graphs [7–12]. Although the multicoloring of hexagonal graphs seems to be a very simplified optimization problem, some interesting mathematical problems were asked at the time that are still open. An example is the Reed McDiarmid conjecture saying that the multichromatic number of any hexagonal graph G is between $\omega(G)$ and $9\omega(G)/8$, where $\omega(G)$ is the weighted clique number [5]. On the other hand, the hexagonal graph model is known to be practically useless in urban areas, where high concrete buildings on one hand prevent propagation of radio signals and on the other hand allow very high concentration of users. Loosely speaking, a three dimensional model may be needed in contrast to the hexagonal graphs that are good model for two dimensional networks. In this paper we discuss a generalization of the multicoloring problem on hexagonal graphs from planar case to three dimensions, where the situation is much more interesting as in two dimensions. Obviously, optimal cells would be nearly balls, and the question is how to position centers of the balls to achieve an optimal

diameter to volume ratio. The famous Kepler conjecture [1] was a longstanding conjecture about ball packing in three-dimensional Euclidean space. It says that no arrangement of equally sized balls filling space has greater average density than that of the cubic close packing (face-centered cubic) and hexagonal close packing arrangements.

Recently Thomas Hales, following an approach suggested by Fejes Toth, published a proof of the Kepler conjecture (see [3, 4]). Given an optimal arrangement of balls, we define a graph by taking the balls (or centers of balls) as vertices and connect touching balls with edges. We call these graphs *cannonball graphs*, as Keplers motivation for studying the arrangements of balls was optimal arrangements of cannonballs. Nonnegative weights are assigned to each vertex and we are interested in multicoloring of the graph induced on vertices of positive weight. Loosely speaking, we generalize the problem of multicoloring of hexagonal graphs from two dimensions to three dimensions.

More formally, we are interested in multicoloring of weighted graphs $G = (V(G), E(G), d)$ where $V = V(G)$ is the set of vertices, $E = E(G)$ is the set of edges, and d assigns a positive integer $d(v)$ to a vertex $v \in V$. A *proper multicoloring* of G is a mapping f from $V(G)$ to subsets of integers such that $|f(v)| \geq d(v)$ for any vertex $v \in V(G)$ and $f(v) \cap f(u) = \emptyset$ for any pair of adjacent vertices u and v in the graph G. The minimal cardinality of a proper multicoloring of G, $\chi_m(G)$, is called the *multichromatic number*. Another invariant of interest in this context is the *(weighted) clique number*, $\omega(G)$, defined as follows: The weight of a clique of G is the sum of weights on its vertices and $\omega(G)$ is the maximal clique weight on G. Clearly, $\chi_m(G) \geq \omega(G)$.

No approximation algorithm and no upper bound was previously known for multichromatic number of cannonball graphs. Here we give an upper bound using some structural properties of the cannonball graphs as well as constructions of polynomial approximation algorithms. The main result of this paper that gives the first answer to the problem asked in [6] is

Theorem 1.1. *There is an approximation algorithm for multicoloring cannonball graphs which uses at most* $\frac{11}{6}\omega(G) + O(1)$ *colors. Time complexity of the algorithm is polynomial.*

2. Basic definitions and useful facts

First we formally define hexagonal and cannonball graphs. Recall the definition of hexagonal graphs: the position of each vertex is an integer linear combination $x\vec{p} + y\vec{q}$ of two vectors $\vec{p} = (1, 0)$ and $\vec{q} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ and the vertices of the triangular lattice are identified with pairs (x, y) of integers. Put an edge if the points representing the vertices are at distance one in this grid. To construct a hexagonal graph G, positive weights are assigned to a finite subset of points in the grid and G is the subgraph induced on $V(G)$, the set of grid vertices with positive weights. Cannonball graphs are constructed in a similar way. However, we have many possibilities already when constructing the underlying grid, which consists of tetrahedrons and will be called a *tetrahedron grid* T. Optimal arrangement of balls in one layer is to put the centers of balls in points of triangular grid. Then, there are exactly two possibilities to put a second layer on the top of the first layer. These two arrangements are obviously symmetric, however, when choosing a position for the third layer, there are two possibilities that give rise to different arrangements (see figure 1).

Consequently, we have an infinite number of tetrahedron grids, that all came from optimal ball arrangements. One of the arrangements (see case (a) of figure 1), can be described nicely by introducing a third vector $\vec{r} = (\frac{1}{2}, \frac{\sqrt{3}}{6}, \frac{\sqrt{6}}{3})$ in addition to $\vec{p} = (1, 0, 0)$ and $\vec{q} = (\frac{1}{2}, \frac{\sqrt{3}}{2}, 0)$.

[1] The solution of Kepler's conjecture is included as a part of 18th problem in the famous list of Hilbert's problem list back in 1900 [13].

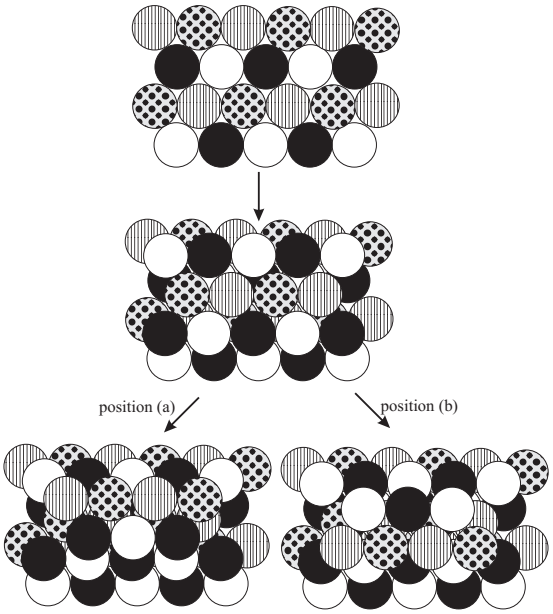


Figure 1: Two different arrangements of the third layer.

Now the position of each vertex is an integer linear combination $x\vec{p}+y\vec{q}+z\vec{r}$ and vertices of the triangular lattice may be identified with a triplet (x, y, z) of integers. For other arrangements (case (b) of figure 1) there is no such an easy extension of the notation from hexagonal graphs. A cannonball graph G is obtained by assigning integer weights to the points of the tetrahedron grid T, taking as $V(G)$ the vertices in the grid with positive weights, and introducing edges between vertices at euclidean distance one. Clearly, from the construction it follows that any layer of a cannonball graph is a hexagonal graph (maybe not connected).

Formally, *cannonball graph* is a graph induced on vertices of positive weight.

There are natural basic 4-colorings of the (unweighted) cannonball graphs. Start with any layer and call it the base layer. Introduce coordinates $(x, y, 0)$ in this layer and define a base coloring by the formula $bc(v) = x \bmod 2 + 2(y \bmod 2)$. Colors of vertices of the next layers are then determined exactly as follows. It is obvious that whenever we store a new layer on (or under) the previous one with fixed coloring, we know that each ball from the new layer is connected to exactly three balls from the previous layer, and all of those balls have different colors. Thus there is exactly one extension of the four coloring to the next layer (see Figure 1). It is easy to see that this rule gives a proper coloring of the next layers.

The cliques in the cannonball graphs can have at most four vertices. The *(weighted) clique number*, $\omega(G)$, is the maximal clique weight on G, where the weight of a clique is the sum of weights on its vertices. We can define invariants $\omega_i(G)$ which denote the maximal weight of a clique of size at most i on G.

It was proved in [5] that for any weighted bipartite graph H, $\chi_m(H) = \omega(H)$, and it can be optimally multicolored by the following procedure:

Procedure 2.1. *[9] Let $H = (V', V'', E, d)$ be a weighted bipartite graph. We get an optimal multicoloring of H if to each vertex $v \in V'$ we assign a set of colors $\{1, 2, \ldots, d(v)\}$, while with each vertex $v \in V''$ we associate a set of colors $\{m(v) + 1, m(v) + 2, \ldots, m(v) + d(v)\}$, where $m(v) = \max\{d(u) : \{u, v\} \in E\}$.*

In a graph $G = (V, E)$, we call a coloring $f : V \to \{1, \ldots, k\}$ *k-good* if for every odd cycle in G and for every i, $1 \leq i \leq k$, there is a vertex $v \in V$ in the cycle such that $f(v) = i$. A graph is

k-good if such coloring exists. The notions of k-good colorings and graphs was first defined in [12]. We can give a procedure for $\frac{k}{k-1}\omega(G)$-coloring of any k-good graph in the following way:

Procedure 2.2. *[12] Since for $1 \leq i \leq k$ we know that every odd cycle in G has at least one vertex assigned color i, the graph remaining after the removal of vertices of color i can be two-colored. Repeating this for $i = 1...k$, and using procedure 2.1, we get $\frac{k}{k-1}\omega(G)$-coloring of G.*

For each vertex $v \in G$, define a *base function* κ as $\kappa(v) = \max\{a(v,u,t) : \{v,u,t\} \in \tau(T)\}$, where $a(u,v,t) = \left\lceil \frac{d(u)+d(v)+d(t)}{3} \right\rceil$, is an average weight of the triangle $\{u,v,t\} \in \tau(T)$.

Clearly, the following fact holds.

Fact 2.1. *For each $v \in G$, $\kappa(v) \leq \left\lceil \frac{\omega_3(G)}{3} \right\rceil \leq \left\lceil \frac{\omega(G)}{3} \right\rceil$*

We call vertex v *heavy* if $d(v) > \kappa(v)$, otherwise we call it *light*. If $d(v) > 2\kappa(v)$ we say that the vertex v is *very heavy*.

To color vertices of G we use colors from an appropriate *palette*. For a given color c, its palette is defined as a set of pairs $\{(c,i)\}_{i\in\mathbb{N}}$. A palette is called a *base color palette* if $c \in \{0,1,2,3\}$ is one of the base colors, and it is called *additional color palette* if $c \notin \{0,1,2,3\}$.

If a vertex v does not have a neighbor of color i in G, we call such color a *free color* of v.

3. Algorithm for multicoloring cannonball graphs

Input: Weighted cannonball graph $G = (V, E, d)$.

Output: A proper multicoloring of G, using at most $\frac{11}{6} \cdot \omega(G) + O(1)$ colors.

Step 0 For each vertex $v \in V$ compute its base color $bc(v)$ and its base function value

$$\kappa(v) = \max\left\{ \left\lceil \frac{d(u)+d(v)+d(t)}{3} \right\rceil : \{v,u,t\} \in \tau(T) \right\},$$

where $\tau(T)$ is a set of all triangles in tetrahedron grid T.

Step 1 For each vertex $v \in V$ assign $\min\{\kappa(v), d(v)\}$ colors from its base color palette to v. Construct a new weighted triangle-free cannonball graph $G_1 = (V_1, E_1, d_1)$ where $d_1(v) = \max\{d(v) - \kappa(v), 0\}$, $V_1 \subseteq V$ is the set of vertices with $d_1(v) > 0$ (heavy vertices in G) and $E_1 \subseteq E$ is the set of all edges in G with both endpoints from V_1 (G_1 is induced by V_1).

Step 2 For each vertex $v \in V_1$ with $d_1(v) > \kappa(v)$ (very heavy vertices in G) assign the first unused $\kappa(v)$ colors of the base color palettes of its neighbors in tetrahedron grid T. Construct a new graph $G_2 = (V_2, E_2, d_2)$ where $d_2(v)$ is the difference between $d_1(v)$ and the number of colors assigned in this step, $V_2 \subseteq V_1$ is the set of vertices with $d_2(v) > 0$ and $E_2 \subseteq E_1$ is the set of all edges in G_1 with both endpoints from V_2 (G_2 is induced by V_2).

Step 3 For each vertex $v \in V_2$ with $deg(v) = 4$ assign unused colors from the free color base palette. Construct a new 3-colorable graph $G_3 = (V_3, E_3, d_3)$ where $d_3(v)$ is the difference between $d_2(v)$ and the number of colors assigned in this step, $V_3 \subseteq V_2$ is the set of vertices with $d_3(v) > 0$ and $E_3 \subseteq E_2$ is the set of all edges in G_2 with both endpoints from V_3 (G_3 is induced by V_3).

Step 4 Apply Procedure 2.2 for graph G_3 by using colors from new additional color palettes.

4. Conclusion

In this paper we provide an algorithm for a proper multicoloring of cannonball graphs that uses at most $\frac{11}{6}\omega(G) + C$ colors. We believe that further improvements can be done. The interesting problems that remain open are, improvement of the competitive ratio 11/6, finding some distributed algorithms for multicoloring cannonball graphs or finding some k-local algorithms for some k, similarly as in 2D case for hexagonal graphs. We already mentioned that in 2D case better bounds were obtained for triangle-free hexagonal graphs. It is very likely that also for cannonball graphs exist some "forbidden" subgraphs H, maybe tetrahedrons, such that better bounds can be obtained for H-free cannonball graphs.

References

[1] K. Aardal, S van Hoesel, A koster, C. Mannino, A.Sassano, *Models and solution techniques for frequency assignment problems*, Annals of Operations Research 153, 79-129, 2007.

[2] Hale, W.K. *Frequency assignment: theory and applications*, Proceedings of the IEEE, vol 68(12), 1497-1514, 1980.

[3] T. Hales, Cannonballs and honeycombs, Notices of the American Mathematical Society 47, 2000, 440—449.

[4] T. Hales, A proof of the Kepler conjecture, Annals of Mathematics. Second Series 162, 2005, 1065—1185.

[5] McDiarmid, C., Reed, B. *Channel assignment and weighted coloring*, Networks, vol. 36(2), 114-117, 2000.

[6] Algorithmische Graphentheorie, Oberwolfach, December 8-14, 2002. (Report No. 55/2002). Oberwolfach: Mathematisches Forschungsinstitut, 2002. (seminar page http://www.mfo.de/occasion/0250/www_view, link to report www.mfo.de/document/0250/Report55_2002.ps).

[7] Sau, I., Šparl, P., Žerovnik, J. *7/6-approximation Algorithm for Multicoloring Triangle-free Hexagonal Graphs*, Discrete Mathematics, vol. 312, 181-187, 2012.

[8] Šparl, P., Witkowski, R. Žerovnik, J. *A Linear Time Algorithm for* $7-[3]$*-coloring Triangle-free Hexagonal Graphs*, Information Processing Letters, vol. 112, 567-571, 2012.

[9] Šparl, P., Witkowski, R. Žerovnik, *1-local 7/5-Competitive Algorithm for Multicoloring Hexagonal Graphs*, Algorithmica, vol. 64(4), 564-583, 2012.

[10] Šparl, P., Žerovnik, J. *2-local 4/3-competitive Algorithm for Multicoloring Hexagonal Graphs*, Journal of Algorithms, vol. 55(1), 29-41, 2005.

[11] Šparl, P., Žerovnik, J. *2-local 5/4-competitive algorithm for multicoloring triangle-free hexagonal graphs*, Information Processing Letters, vol. 90(5), 239-246, 2004.

[12] Sudeep, K.S., Vishwanathan, S. *A technique for multicoloring triangle-free hexagonal graphs*, Discrete Mathematics, vol. 300, 256-259, 2005.

[13] Benjamin H. Yandell, The Honors Class. Hilbert's Problems and Their Solvers, A K Peters, 2002.

DGA 2013, pp. 205 – 208.

Geometric distances in relative astrometry

Ramachrisna Teixeira,[1] Alberto Krone-Martins, [2] Christine Ducourant[3] and Phillip A.B. Galli[1]

[1]*IAG - Universidade de São Paulo, São Paulo, Brasil,* teixeira@astro.iag.usp.br, galli@astro.iag.usp.br

[2]*SIM - Universidade de Lisboa, Lisboa, Portugal,* algol@sim.ul.pt

[3]*LAB - Université de Bordeaux, Bordeaux, France,* ducourant@obs.u-bordeaux1.fr

Abstract In this work we present a brief introduction on the adoption of geometric distances in relative astrometry. We quickly describe the observational and data-reduction principles, as well as the precision that has been obtained in contemporary studies.

Keywords: Astronomy, Relative Astrometry, Distance Measurements

1. Introduction

Although the word Astrometry has a broad meaning, in daily astronomical research it represents the branch of Astronomy concerned with the position of celestial bodies in space, and associated variations in time. This includes from the definition and materialization of a reference system to the study of the movements of the observer, the astronomical source's intrinsic kinematics, the Galactic structure and the measurement of the most fundamental quantity in Astronomy: the stellar trigonometric parallax. It is this quantity that provides the best estimation of stellar distances from the Solar System, and thus it is the first step in a cosmic distance ladder.

Usually, the word "position of an astronomical object" means the direction in which we are able to observe this object: in other words, it is a projection of the object's spatial position at the surface of an unitary sphere centered at the observer, the celestial sphere. In most astronomical studies, this position is estimated from the measurement of angular distances between the target object and several other objects with known celestial positions – in a certain way, this is not dissimilar to the adoption of anchor nodes in the Distance Geometry formulation of the sensor network location problem (e.g. [1]), using euclidean distances in the case of images coordinates or spherical distances in the case of the objects at the celestial sphere.

Most of contemporary relative astrometric works are based in observations with CCD cameras or infrared detectors. These observations result in image matrices that must be analyzed in order to allow the determination of the coordinates of the photocenters. Depending on the adopted data reduction system, the photocenter determination may be based on brightness momenta (e.g. [2]), profile fitting (e.g. [3]), or point-spread-function analyses (e.g. [4]). As an example, the data reduction system employed in Valinhos and Bordeaux CCD meridian circles, adopts a profile fitting method. For each source, the method determines the rectangular coordinates (x_0, y_0) of their photocenters, relative to an arbitrary origin, using a bivariate gaussian function:

$$\phi(x,y) = \frac{\Phi}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp -\frac{1}{2(1-\rho^2)}[\left(\frac{x-x_0}{\sigma_x}\right)^2 + \left(\frac{y-y_0}{\sigma_y}\right)^2 - 2\rho\left(\frac{x-x_0}{\sigma_x}\right)\left(\frac{y-y_0}{\sigma_y}\right)]$$

where Φ is the object's total flux, (x_0, y_0) are the photocenter coordinates, σ_x and σ_y are the standard deviations, and ρ is the correlation coefficient.

Then, in order to obtain celestial positions of astronomical objects from the rectangular coordinates of the photocenters of all the detected objects in an image, it is necessary to identify which are those with already known positions from some reference catalogue. Adopting these objects with already known celestial coordinates, a least-squares problem is setup (e.g. [5], [6], [7], [8], [9]), enabling the determination of instrumental parameters and the conversion between the rectangular coordinates of all the target objects into celestial coordinates. In this way, the link established by the reference objects materializes a fraction of the celestial sphere whose orientation is defined by their known positions. Thus the target objects' celestial positions may be determined with respect to this materialization.

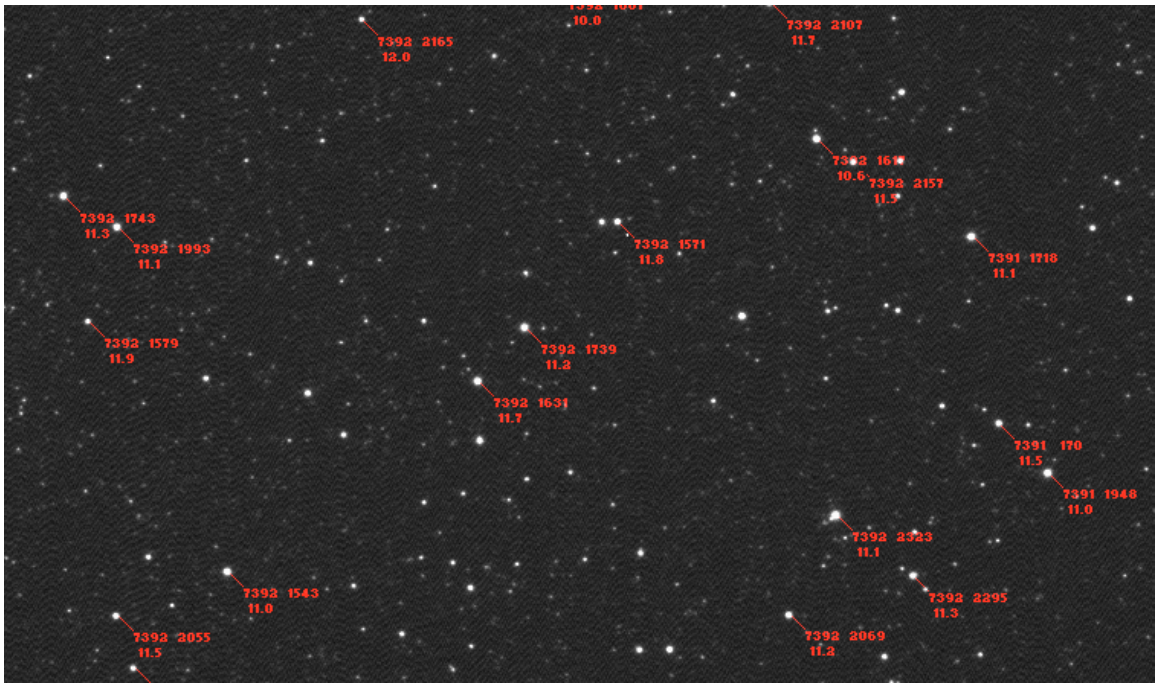


Figure 1: A field observed with the CCD meridian circle of the Observatório Abrahão de Moraes at IAG/USP – Valinhos. The objects adopted as references are marked in red.

In Fig. 1 a typical observation of a stellar field performed by the Valinhos CCD meridian circle is represented. In this image, the objects with red identifiers have known coordinates from a reference catalogue – in this case, the Tycho 2 catalogue [10], that was constructed from observations performed by the ESA Hipparcos satellite. This catalogue provides a reliable materialization of the International Celestial Reference System [11] in the optical wavelengths.

Naturally, the scientific contribution and even the meaning of the astrometric observations are directly linked to the precision attained by their measurements. However, the precision and accuracy of these measurements depend on several factors. First, the atmosphere and optics play a major role, limiting the possible attainable resolution at each observation (e.g. [12]).

Then, the noise sources[1] as well as the apparent brightness of the target, limits the photocenter determination. Afterwards, the quality of the adopted reference system also play a role – by its on, the quality of a reference system is dependent on how it is materialized, or what (and how many) are the reference sources present in the field. Finally, the precision and accuracy of these measurements depend on the rigidity of the link between all the individual images together – and this is ruled by the determination of geometric distances between the sources.

Using the CCD meridian circles such as Valinho's or Bordeaux's, for instance, it is possible to obtain positions from ~ 6 observations with mean precisions of ~ 50 mas (or milliarcseconds). If more observations are performed, it is feasible to obtain positions with mean precisions of ~ 10 to 30 mas, depending on the target's magnitude (the best range is V ~ 9 to 14 mag). Proper-motions can be obtained by these instruments with precisions better than 5 mas/yr, using observation baselines of several years (e.g. [13]).

If instruments with bigger optics are adopted, and thus greater spatial resolution, such as the ESO NTT telescope, it is possible to obtain positions and proper motions with more than ten times the above quoted precisions. Also, these instruments enable reliable determinations of stellar trigonometric parallaxes, which are the first steps towards the determination of physical distances in the Universe.

Acknowledgments

R.T and P.A.B.G. thank the Brazilian agencies FAPESP and CAPES for financial support. A.K.M. thanks the Portuguese agency Fundação para Ciência e Tecnologia, FCT (SFRH/ BPD/ 74697/ 2010) for financial support. C.D. thanks the French agency COFECUB.

References

[1] Caoa, M., Anderson, B.D.O., Morsea, A. S. (2006). "Sensor network localization with imprecise distances", Systems & Control Letters, v. 55, p. 887.

[2] Bertin, E. and Arnouts, S. (1996). "SExtractor: Software for source extraction". Astronomy & Astrophysics Supplement Series, v. 317, p. 393.

[3] Viateau, B., Réquième, Y., Le Campion, J.F., Benevides-Soares, P., Teixeira, R., et. al. (1998). "The Bordeaux and Valinhos CCD meridian circles". Astronomy & Astrophysics Supplement Series, v. 134, p. 173.

[4] Stetson, P.B. (1987). "DAOPHOT - A computer program for crowded-field stellar photometry". Publications of the Astronomical Society of the Pacific, v. 99, p. 191.

[5] Eichhorn H. (1960)."Über die Reduktion von photographischen Sternpositionen und Eigenbewegungen". Astronomische Nachrichten, v. 285, p. 233.

[6] Jefferys, W.H. (1987). "Quaternions as astrometric plate constants". Astronomical Journal, v. 93, p. 755.

[7] Brosche, P., Wildermann, E., Geffert, M. (1989). "Astrometric plate reductions with orthogonal functions". Astronomy and Astrophysics, v.211, n.1, p. 239.

[8] Benevides-Soares, P., Teixeira, R. (1992). "On the relationship between conventional and overlap reduction techniques in positional astronomy". Astronomy and Astrophysics, v. 253, n. 1, p. 307.

[9] Teixeira, R., Requieme, Y., Benevides-Soares, P., Rapaport, M. (1992). "Global treatment of the Bordeaux meridian observations". Astronomy and Astrophysics, v. 264, n. 1, p. 307.

[10] Hog, E., Fabricius, C., Makarov, V. V., Urban, S. et. al. (2000). "The Tycho-2 catalogue of the 2.5 million brightest stars". Astronomy and Astrophysics, v. 355, p. L27.

[11] Arias, E.F., Charlot, P., Feissel, M. and Lestrade, J.F., (1995). "The extragalactic reference system of the International Earth Rotation Service, ICRS". Astronomy and Astrophysics, v.303, 604-608.

[1]The noise sources can be physically intrinsic due to poissonian photon processes and/or generated by detector electronics.

[12] Lindegren, L. (1980). “Atmospheric limitations of narrow-field optical astrometry”. Astronomy and Astrophysics, v. 89, n. 1-2, p. 41.

[13] Teixeira, R., Galli, P.A.B, Benevides-Soares, P. et al. (2011). "Proper motion and densification of the International Celestial Reference Frame in the direction of the Galactic bulge". Astronomy and Astrophysics, v. 534, A91.

DGA 2013, pp. 209 – 214.

Influence Analyses of Skew–Normal/Independent Linear Mixed Models

Filidor Vilca,[1] Camila Borelli Zeller,[2] and Victor Hugo Lachos[1]

[1] *University of Campinas, Brazil,* fily@ime.unicamp.br

[2] *Universidade de Juiz de Fora, Brazil,* camilaestat@yahoo.com.br

[2] *University of Campinas, Brazil* hlachos@ime.unicamp.br

Abstract Linear mixed models were developed to handle clustered data, and these models have increased significantly in the last fifty years. In general, the normality (or symmetry) of the random effects is a common assumption, but this kind of assumption may be unrealistic, obscuring important features of among-subjects variations.

We have extended the classical linear mixed model herein, allowing the random effects and the random errors to jointly follow a multivariate skew–normal/independent distribution, and we consider diagnostic analyses following the ideas from Cook's well–known approach which is based on the likelihood displacement. We developed local influence measures according to Zhu and Lee's (2001) approach for skew–normal/independent linear mixed model (SNI-LMM). Perturbations schemes are discussed as well as the use the Mahalanobis distance for identifying potential outlying observations. Finally, a real data set has been analyzed in order to illustrate the usefulness of the proposed methodology.

Keywords: Mahalanobis distance, local influence, outliers

1. Introduction

Estimating the distance between two points or more general between objects of interest are of fundamental concern different area as well as in statistical applications, for example, the distance between two observations. In statistics, is usual to find applications based on the Mahalanobis distance is a metric which is better adapted than the usual Euclidean distance to settings involving non spherically symmetric distributions. It is more particularly useful when multivariate distributions are involved.

Influence diagnostics techniques consist in evaluating the sensitivity of the parameter estimates of a particular model when perturbation occurs in the data set or in the assumptions of the model. Case deletion (Cook, 1977) is a common approach to analyze one or more fitted models after excluding observations that is direct assessed by some metrics such as the likelihood displacement and the Cook's distance. This method is also known as the global influence method. The influence of the ith observation on the parameter estimate can be assessed by studying the difference between $\widehat{\theta}$ and $\widehat{\theta}_{(i)}$, where $\widehat{\theta}_{(i)}$ denotes the maximum likelihood(ML) estimate of θ obtained from the sample of size $n-1$ excluding the ith observation. To assess the influence of the ith case on the ML estimate $\widehat{\boldsymbol{\theta}}$, the basic idea is to compare the difference between $\widehat{\boldsymbol{\theta}}_{(i)}$ and $\widehat{\boldsymbol{\theta}}$. The generalized Cook's distance is defined as a standardized norm of

$\widehat{\boldsymbol{\theta}}_{(i)} - \widehat{\boldsymbol{\theta}}$, i.e.,

$$GD_i = (\widehat{\boldsymbol{\theta}}_{(i)} - \widehat{\boldsymbol{\theta}})^\top \mathbf{M} (\widehat{\boldsymbol{\theta}}_{(i)} - \widehat{\boldsymbol{\theta}}), \tag{1}$$

where $\mathbf{M}$ is a non-negative definite matrix. Another measure of distance between $\widehat{\boldsymbol{\theta}}_{(i)}$ and $\widehat{\boldsymbol{\theta}}$ is the likelihood displacement defined as $LD_i(\boldsymbol{\theta}) = 2\{\ell(\widehat{\boldsymbol{\theta}}) - \ell(\widehat{\boldsymbol{\theta}}_{(i)})\}$, where $\ell(.)$ is the log-likelihood function.

Cook (1986) proposed an unified approach for assessment of local influence in minor perturbations of a statistical model and it can be viewed as a generalization of the robustness concept to study and detect the influential subsets of data. A alternative approach was proposed by Zhu and Lee (2001) that is based on the EM algorithm, that requires the evaluation of $Q(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}) = \mathrm{E}[\ell_c(\boldsymbol{\theta}|\mathbf{y}_c)|\mathbf{y}, \widehat{\boldsymbol{\theta}}]$. To evaluate the departure of the models, Cook (1986) proposed to use the likelihood displacement that is defined below: let $\boldsymbol{\omega}$ be a $g \times 1$vector of perturbation restricted to some open subset of $\mathbf{R}^g$. The perturbations are made in the likelihood function such that it takes the form $\ell(\boldsymbol{\theta}|\boldsymbol{\omega})$, and consider $\boldsymbol{\omega}_0$ such that $\ell(\boldsymbol{\theta}|\boldsymbol{\omega}_0) = \ell(\boldsymbol{\theta})$. To asses the influence of the perturbations on the ML estimate, one may consider the likelihood displacement

$$LD(\boldsymbol{\omega}) = 2\{\ell(\widehat{\boldsymbol{\theta}}) - \ell(\widehat{\boldsymbol{\theta}}_\omega)\},$$

where $\widehat{\boldsymbol{\theta}}$ is the ML estimate of $\boldsymbol{\theta}$ under the proposed model and $\widehat{\boldsymbol{\theta}}_\omega$ denotes the ML estimate under the perturbed model.

The multivariate skew-normal/independent (SNI) distribution (Branco and Dey, 2001) is defined through the probability density function (pdf)

$$f(\mathbf{y}) = 2 \int_0^\infty \phi_p(\mathbf{y}|\boldsymbol{\mu}, u^{-1}\boldsymbol{\Sigma})\ \Phi(u^{1/2}\boldsymbol{\lambda}^\top \boldsymbol{\Sigma}^{-1/2}(\mathbf{y} - \boldsymbol{\mu}))\, dH(u; \boldsymbol{\nu}), \quad \mathbf{y} \in R^p, \tag{2}$$

where $\phi_p(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the pdf of the p-variate normal distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$, $\Phi(\cdot)$ denotes the cumulative distribution function (cdf) of the standard normal distribution, U is a positive random variable with a cdf $H(u; \boldsymbol{\nu})$, where $\boldsymbol{\nu}$ is a scalar or parameter vector indexing the distribution of U. The distribution defined in (2) in denoted by $\mathrm{SNI}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}; H)$. When $\boldsymbol{\lambda} = \mathbf{0}$, the SNI distribution in (2) reduces to the normal/independent (NI) distribution. That is, $\mathbf{Y} \sim \mathrm{NI}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}; H)$; see Lange and Sinsheimer (1993).

2. The skew–normal/independent linear mixed model

In this section, we consider the skew–normal/independent linear mixed model (SNI-LMM). In general, a normal linear mixed effects model (N-LMM hereafter) is defined as (Arellano–Valle et al, 2005)

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\epsilon}_i, \quad ,, \tag{3}$$

where $\mathbf{Y}_i$ is a $(n_i \times 1)$ vector of observed continuous responses for sample unit i, $\mathbf{X}_i$ of dimension $(n_i \times p)$ is the design matrix corresponding to the fixed effects, $\boldsymbol{\beta}$ of dimension $(p \times 1)$ is a vector of population-averaged regression coefficients called fixed effects, $\mathbf{Z}_i$ of dimension $(n_i \times q)$ is the design matrix corresponding to the $(q \times 1)$ random effects vector $\mathbf{b}_i$, and $\boldsymbol{\epsilon}_i$ of dimension $(n_i \times 1)$ is the vector of random errors. It is assumed that the random effects $\mathbf{b}_i$ and the residual components $\boldsymbol{\epsilon}_i$ are independent with $\mathbf{b}_i \overset{\text{iid}}{\sim} N_q(\mathbf{0}, \mathbf{D})$ and $\boldsymbol{\epsilon}_i \overset{\text{ind}}{\sim} N_{n_i}(\mathbf{0}, \boldsymbol{\Sigma}_i)$. The $q \times q$ covariance matrix $\mathbf{D}$ may be unstructured or structured. The $n_i \times n_i$ covariance matrices $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_i(\boldsymbol{\gamma})$, $i = 1, \ldots, m$, are typically assumed to depend on i through their dimension, being parameterized by a fixed, generally small, parameter set $\boldsymbol{\gamma}$ as, for instance, with an AR(1) covariance structure. As in Lachos et al. (2009), the SNI-LMM is defined by considering

$$\mathbf{b}_i \overset{\text{iid}}{\sim} SNI_q(\mathbf{0}, \mathbf{D}, \boldsymbol{\lambda}; H) \text{ and } \boldsymbol{\epsilon}_i \overset{\text{ind}}{\sim} NI_{n_i}(\mathbf{0}, \sigma_e^2\mathbf{R}_i; H),\ i = 1, \ldots, m. \tag{4}$$

From (**??**), we can write the SNI-LMM as follows

$$\mathbf{Y}_i|\mathbf{b}_i, U_i = u_i \overset{\text{ind}}{\sim} N_{n_i}(\mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i, u_i^{-1}\sigma_e^2\mathbf{R}_i), \tag{5}$$

$$\mathbf{b}_i|T_i = t_i, U_i = u_i \overset{\text{ind}}{\sim} N_q(\boldsymbol{\Delta}t_i, u_i^{-1}\boldsymbol{\Gamma}), \tag{6}$$

$$T_i|U_i = u_i \overset{\text{ind}}{\sim} HN_1(0, u_i^{-1}), \tag{7}$$

$$U_i \overset{\text{iid}}{\sim} H(u_i; \boldsymbol{\nu}), \tag{8}$$

for $i = 1, \ldots, m$, all independent, $\boldsymbol{\Delta} = \mathbf{D}^{1/2}\boldsymbol{\delta}$, $\boldsymbol{\Gamma} = \mathbf{D} - \boldsymbol{\Delta}\boldsymbol{\Delta}^\top$, with $\boldsymbol{\delta} = \boldsymbol{\lambda}/\sqrt{1 + \boldsymbol{\lambda}^\top\boldsymbol{\lambda}}$, and $\mathbf{D}^{1/2}$ is the square root of $\mathbf{D}$ containing $q(q+1)/2$ distinct elements, say $\boldsymbol{\alpha}$, and $HN_1(0, \sigma^2)$ is the half-$N_1(0, \sigma^2)$ distribution. When $U_i = 1$ ($i = 1 \ldots, m$), the SNI–LMM reduces to the SN–LMM as defined in Arellano–Valle et al. (2005), and if $\boldsymbol{\lambda} = \mathbf{0}$, the SNI-LMM reduces to the usual NI-MLM which has been discussed quite extensively in the the literature. The EM-type algorithm requires the evaluation of $Q(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}) = \mathrm{E}[\ell_c(\boldsymbol{\theta}|\mathbf{y}_c)|\mathbf{y}, \widehat{\boldsymbol{\theta}}] = \sum_{i=1}^m Q_i(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}})$, where the expectation is taken with respect to the joint conditional distribution of $\mathbf{b}$, $\mathbf{u}$ and $\mathbf{t}$, given $\mathbf{y}$ and $\widehat{\boldsymbol{\theta}}$. Thus, we have that $Q_i(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}) = Q_{1i}(\boldsymbol{\beta}, \sigma_e^2|\widehat{\boldsymbol{\theta}}) + Q_{2i}(\boldsymbol{\alpha}, \boldsymbol{\lambda}|\widehat{\boldsymbol{\theta}})$, where $Q_{1i}(\boldsymbol{\beta}, \sigma_e^2|\widehat{\boldsymbol{\theta}}) = -\frac{1}{2}\log|\sigma_e^2\mathbf{R}_i| - \frac{\widehat{u}_i}{2\sigma_e^2}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^\top\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) + \frac{1}{\sigma_e^2}(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^\top\mathbf{R}_i^{-1}\mathbf{z}_i\widehat{\mathbf{ub}}_i - \frac{1}{2\sigma_e^2}\mathrm{tr}\left(\mathbf{R}_i^{-1}\mathbf{z}_i\widehat{\mathbf{ub^2}}_i\mathbf{z}_i^\top\right)$, $Q_{2i}(\boldsymbol{\alpha}, \boldsymbol{\lambda}|\widehat{\boldsymbol{\theta}}) = -\frac{1}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Gamma}^{-1}\widehat{\mathbf{ub^2}}_i\right) + \boldsymbol{\Delta}^\top\boldsymbol{\Gamma}^{-1}\widehat{\mathbf{utb}}_i - \frac{\widehat{ut^2}_i}{2}\boldsymbol{\Delta}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\Delta}$, with $\widehat{u}_i$, $\widehat{\mathbf{ub}}_i$, $\widehat{\mathbf{ub^2}}_i$, $\widehat{\mathbf{utb}}_i$ and $\widehat{ut^2}_i$ $i = 1, \ldots, m$, are all as given in Lachos et al. (2009).

3. Local influence

Consider a perturbation vector $\boldsymbol{\omega}$ in an open region $\boldsymbol{\Omega}$. To apply the local influence approach, we consider $Q(\boldsymbol{\theta}, \boldsymbol{\omega}|\widehat{\boldsymbol{\theta}}) = \mathrm{E}[\ell_c(\boldsymbol{\theta}, \boldsymbol{\omega}|\mathbf{Y}_c)|\mathbf{y}, \widehat{\boldsymbol{\theta}}]$, where $\ell_c(\boldsymbol{\theta}, \boldsymbol{\omega}|\mathbf{Y}_c)$, $\boldsymbol{\theta} \in \mathbf{R}^h$, be the complete-data log-likelihood of the perturbed model. We consider the following perturbation schemes:
1. Perturbation of case weights: The complete-data log-likelihood function (perturbed Q-function) is given by $Q(\boldsymbol{\theta}, \boldsymbol{\omega}|\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^m w_i Q_{1i}(\boldsymbol{\beta}, \sigma_e^2|\widehat{\boldsymbol{\theta}}) + \sum_{i=1}^m w_i Q_{2i}(\boldsymbol{\alpha}, \boldsymbol{\lambda}|\widehat{\boldsymbol{\theta}})$, where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)^\top$ is an $m \times 1$ vector.
2. Perturbation of the scale matrix $\mathbf{D}$:To study the effects of departure from the assumption regarding the scale matrix $\mathbf{D}$ of the random effects, we consider the following perturbation $\boldsymbol{\Delta}(\omega_i) = \omega_i^{-1/2}\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}(\omega_i) = \omega_i^{-1}\boldsymbol{\Gamma}$.
3. Perturbation of explanatory variables: Here is perturbed explanatory matrix $\mathbf{X}_i(\boldsymbol{\omega}) = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iu}(\omega_i), \ldots, \mathbf{x}_{ip})$, where $\mathbf{x}_{iu}(\omega_i) = \mathbf{x}_{iu} + \omega_i\mathbf{1}_{n_i}, u = 1, \ldots, p$, $\mathbf{x}_{iu}$ is the uth column of the matrix $\mathbf{X}_i$, and $\mathbf{1}_{n_i}$ is an $n_i \times 1$ vector of ones.
4. Perturbation of response variables: A perturbation of the response variables $(\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top)^\top$ is introduced by replacing $\mathbf{y}_i$ by $\mathbf{y}_i(\boldsymbol{\omega}) = \mathbf{y}_i + \omega_i\mathbf{1}_{n_i}$, $i = 1, \ldots, m$.

4. Application

We illustrate the developed method with the Framingham cholesterol data set from Zhang and Davidian (2001). We fit a LMM model to the data as specified by Zhang and Davidian (2001)

$$Y_{ij} = \beta_o + \beta_1\mathrm{sex}_i + \beta_2\mathrm{age}_i + \beta_3 t_{ij} + b_{0i} + b_{1i}t_{ij} + \epsilon_{ij}\,, \tag{9}$$

where Y_{ij} is the cholesterol level, divided by 100, at the j-th time for subject i; t_{ij} is (time $-$ 5)/10, with time measured in years from the start of the study; age_i is age at the start of the study; sex_i is the gender indicator (0 = female, 1 = male). Thus, $\mathbf{x}_{ij} = (1, \mathrm{sex}_i, \mathrm{age}_i, t_{ij})^\top$,

$\mathbf{b}_i = (b_{0i}, b_{1i})^\top$ and $\mathbf{Z}_{ij} = (1, t_{ij})^\top$, $i = 1, \ldots, 200$. First, we have fitted the SNI-LMM for the Framingham cholesterol data set. Although not being formal tests, as in Zhang and Davidian (2001), we compare the SNI-LMM and the NI-LMM (specifically, models normal, t-Student, slash and contaminated normal) by inspecting some information criteria. Next, we have identified influential observations for the Framingham cholesterol data set.

Now, we revisit the Framingham cholesterol data in order to study the local influence approach in the context of SNI-LMM. In our analysis we will assume SN, ST, SSL and SCN distributions from the SNI class for comparative purposes. In order to detect outlying observations, we use the Mahalanobis distance

$$d_i^2(\widehat{\boldsymbol{\theta}}) = \frac{1}{\widehat{\sigma}_e^2}\widehat{\mathbf{e}}_i^\top \mathbf{R}_i^{-1}\widehat{\mathbf{e}}_i + \widehat{\boldsymbol{\mu}}_{b_i}^\top \widehat{\mathbf{D}}^{-1}\widehat{\boldsymbol{\mu}}_{b_i} = \widehat{d_{\mathbf{e}_i}^2} + \widehat{d_{\mathbf{b}_i}^2}, \tag{10}$$

We can use as cutoff points the quantile of the distribution of d_i^2. Figure 1 displays these distances for the four fitted models. The cutoff lines correspond to the quantile $\upsilon = \chi_4^2(\xi)$, with $\xi = 0.99$. We can see from these figures that observations $8, 15, 26, 69, 74, 90, 111, 122, 138, 146, 160, 162, 174, 175$ and 187 appear to be outliers.

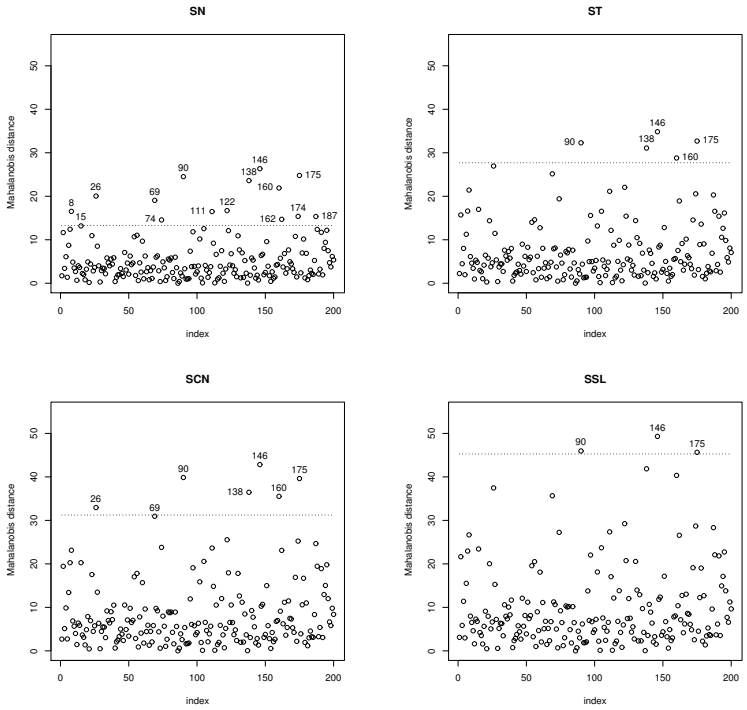


Figure 1: Index plots of the Mahalanobis distances for the four fitted models.

The estimated distances $d_{\mathbf{e}_i}^2$ (Error) and $d_{\mathbf{b}_i}^2$ (Random Effect–R.E.), obtained from (10), provide useful diagnostic statistics for identifying subjects with outlying observations. Figure 2 presents these diagnostic statistics for SN-LMM. The observations $69, 90, 138, 145$ and 175 presents large value of $d_{\mathbf{e}_i}^2$, suggesting an **e**-outlier. Moreover, observations $8, 26$ and 160 present large values of $d_{\mathbf{b}_i}^2$ suggesting a **b**-outlier. The $d_{\mathbf{b}_i}^2$ plots gives some indication that observations $2, 131$ and 172 are possibly a **b**-outliers, which cannot be concluded from Figure 1. For SNI distributions with heavy tails, we observed the same results and so they are not shown here.

Perturbation of case weights: From Figure 3 is noted that under for the four fitted models, the observation 39 is identified as influential. As expected, the influence of such observation is reduced when we consider distributions with heavier tails than the skew-normal ones.

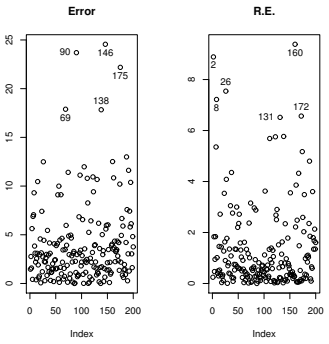


Figure 2: Estimated $d^2_{\mathbf{e}i}$ (error) and $d^2_{\mathbf{b}_i}$ (R.E.) to the skew-normal fit.

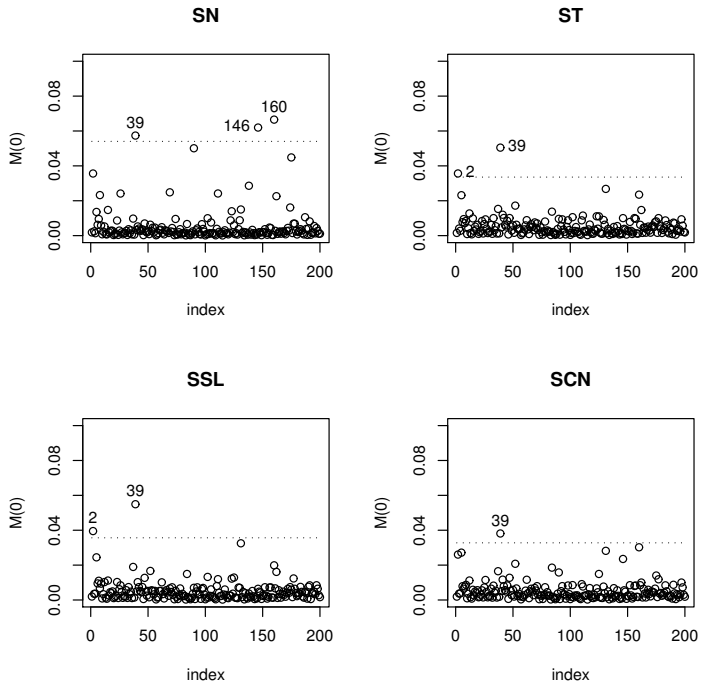


Figure 3: Index plots of M(0) under case weights perturbation for the four fitted models. The horizontal lines delimit the Lee and Xu (2004) benchmark for M(0) with $c* = 5$.

References

[1] Arellano-Valle, R. B., Bolfarine, H. and Lachos, V. H. (2005). Skew-normal linear mixed models, *Journal of Data Science,* **3**, 415-438.

[2] Branco, M. D. and Dey, D. K. (2001). A general class of multivariate skew-elliptical distribution. *Journal of Multivariate Analysis* **79**: 93-113.

[3] Cook, R. D. (1977). Detection of influential observations in linear regression. *Technometrics*, **19**, 15-18.

[4] Cook, R. D. (1986). Assessment of local influence (with discussion), *Journal of the Royal Statistical Society, Series B*, **48**, 133-169.

[5] Lachos, V. H., Ghosh, P. and Arellano–Valle, R. B. (2009). Likelihood based inference for skew–normal/independent linear mixed models, *Statistica Sinica.*

[6] Lange, K. and Sinsheimer, J. S. (1993). Normal/independent distributions and their applications in robust regression, *Journal of Computational and Graphical Statistics,* **2**, 175-198.

[7] Zhang, D. and Davidian, M. (2001). Linear mixed models with flexible distributions of random effects for longitudinal data, *Biometrics*, **57**, 795-802.

[8] Zhu, H. and Lee, S. (2001). Local influence for incomplete-data models, *Journal of the Royal Statistical Society, Series B*, **63**, 111-126.

DGA 2013, pp. 215 – 217.

Solving the Distance Geometry Problem by the Hyperbolic Smoothing Approach

Adilson Elias Xavier[1] and Helder Manoel Venceslau[2]

[1]*Federal University of Rio de Janeiro - Brazil,* adilson@cos.ufrj.br

[2]*Federal University of Rio de Janeiro - Brazil,* heldermv@cos.ufrj.br

Abstract The geometrical distance problem in graphs is characterized by determining the positions of the nodes in a Euclidian space, according to the given distances associated with the arcs. It is a non-convex and non-differentiable problem, having a myriad of local minima. The presented methodology addopts a smoothing strategy named Hyperbolic Smoothing Technique. Computational results obtained in the resolution of large instances of a difficult canonic problem show the efficiency and robustness of the method. The geometrical distance problem has a relevant application in the determination of geometrical structures of proteins.

Keywords: Protein Folding, Nondifferentiable Programming, Smoothing

1. Solving the DGP by the HS Approach

The presented methodology adopts a smoothing strategy named Hyperbolic Smoothing Technique. By a smoothing approach, we fundamentally mean the substitution of an intrinsically non-differentiable problem by a C^∞ differentiable alternative. In the Hyperbolic Smoothing (HS) methodology, the solution is obtained by solving a sequence of smooth problems which gradually approaches the original one.

First, we will consider the smoothing of the absolute value function $|\,u\,|$, where $u \in \mathbb{R}$. For this purpose, for $\gamma > 0$, let us define the function

$$\theta(\,u\,,\,\gamma\,) \;=\; \sqrt{u^2 \,+\, \gamma^2} \tag{1}$$

The considered Distance Geometry problem has the following specification. Let $G = (V, E)$ denote a graph, in which for each arc $(i, j) \in E$, it is associated a measure $a_{ij} > 0$. The problem consists of associating a vector $x_i \in \mathbb{R}^n$ for each knot $i \in V$, basically addressed to represent the position of this knot into a $n-$dimensional space, so that Euclidean distances between knots, $\|x_i - x_j\|$, corresponds appropriately to the given measures a_{ij} :

$$minimize \;\; f(x) \;=\; \sum_{(i,j)\in E} (\,\|x_i - x_j\| \,-\, a_{ij})^2. \tag{2}$$

This formulation presents the non-differentiable property due the presence of the Euclidean norm term. Moreover, the objective function is non-convex, so the problem has a large number of local minima. For solving the problem (2) by using the HS technique it is only necessary to use the function $\theta(\,u\,,\,\gamma\,)$ and to take $u = \|x_i - x_j\|$:

$$minimize \;\; f_s(x) \;=\; \sum_{(i,j)\in E} (\,\theta(\|x_i - x_j\|, \gamma) \;-\; a_{ij})^2. \tag{3}$$

Besides its smoothing properties, Xavier [6] shows the important convexification power of the function θ :

Proposition 1: There is a number $\bar{\gamma}$ such that, for all values $\gamma > \bar{\gamma}$, the Hessian matrix $\nabla^2 f_s(x)$ will be positive definite. ∎

Souza [4] and Souza et al [5] considers an alternative formulation, where the distances $\|x_i - x_j\|$ must be inside given intervals $[l_{i,j}, u_{i,j}]$:

$$minimize\; f(x) = \sum_{(i,j)\in E} max[(l_{ij} - \|x_i - x_j\|\,)\,,\, 0] + \sum_{(i,j)\in E} max[(\|x_i - x_j\| - u_{ij})\,,\, 0]$$

By using function $\phi(y,\tau) = \left(y + \sqrt{y^2 + \tau^2}\,\right)/2$ in the place of function $max(0,y)$, and by using function $\theta(u)$ in the place of the Euclidean distance $u = \|x_i - x_j\|$, it is possible to obtain the smooth formulation:

$$minimize\; f_s(x) = \sum_{(i,j)\in E} \phi(l_{ij} - \theta(\|x_i - x_j\|\,, \gamma), \tau) \;+\; \sum_{(i,j)\in E} \phi(\theta(\|x_i - x_j\| - u_{ij}\,, \gamma), \tau)$$

Souza [4] and Souza et al [5] extends the previous theoretical result of Proposition 1 showing the convexification of the above problem for all values $\gamma > \max_{(i,j)\in E}\, u_{ij}$.

In order to show the computational properties of the HS methodology, we took a traditional test problem considered in: Moré and Wu [3], Hoai and Tao [1], Macambira [2] and Xavier [6]. This instance is a synthetic problem, where the knots are located on the intersection of s planes that cut a cube in the three principal directions in equal intervals.

Following Moré and Wu [3], the knot positions are refereed by their coordinates indexes $\{(i_1, i_2, i_3),\, 0 \le i_1 \le s, 0 \le i_2 \le s, 0 \le i_3 \le s\}$. The relative position i of the knot x_i is given by the rule $i = 1 + x_1 + si_2 + s^2 i_3$. The distances a_{ij} associated to the arcs (i,j) are exactly given by $a_{ij} = \| x_i - x_j \|_2$ for each arc $(i,j) \in S$, where $S = \{(i,j) \;||\; i - j \;|< s^2.\}$ The set of the $m = s^3$ knots is represented by $x = (x_1, \ldots, x_m) \in \mathbb{R}^{3s^3}$. So, the problem has $n = 3s^3$ components and $p = s^5 + s^3 + s$ arcs.

Table 1 presents the computational results produced by the HS methodology. The numerical experiments have been carried out on a Intel Core i7-2620M Windows Notebook with 2.70GHz and 8 GB RAM. The programs are coded with Intel(R) Visual Fortran Composer XE 2011 Update 7 Integration for Microsoft Visual Studio* 2010. The unconstrained minimization tasks were carried out by means of a conjugate gradients algorithm employing the Fletcher & Reeves updating formula from the Harwell Library, routine VA08ad, obtained in the site:
(www.cse.scitech.ac.uk/nag/hsl/).

The initial smoothing parameter γ^1 was fixed $\gamma^1 = 10$. In all experiments, the decreasing rate parameter ρ of the parameter γ was fixed $\rho = (10)^{1/32}$ and the number of iterations assumed the value equal to 108. Ten different randomly chosen start points were used.

The columns of Table 1 show the number of splits of the cube (s), the number of knots $(m = s^3)$, the number of variables of the problem $(m = 3\,s^3)$, the number of arcs (p), the occurrences of correct solutions obtained in 10 tentative solutions $(Occur.)$, the average value of the correct solutions (f_{Med}), the mean CPU time $(Time)$ given in seconds associated to 10 tentative solutions, and, whenever considered relevant, the occurrences of correct solutions obtained in 100 tentative solutions using a non smoothed version $(Occ.n.s.)$.

s	$m = s^3$	$n = 3s^3$	p	$Occur.$	f_{Med}	$Time$	$Occ.n.s.$
3	27	81	198	0	-	0.1	8
4	64	192	888	6	0.27E-6	0.7	5
5	125	375	2800	8	0.29E-5	2.8	7
6	216	648	7110	8	0.19E-4	7.6	4
7	343	1029	15582	5	0.16E-4	19	5
8	512	1536	30688	8	0.29E-3	45	3
9	729	2187	55728	6	0.86E-3	97	0
10	1000	3000	94950	7	0.95E-3	45	1
11	1331	3993	153670	6	0.17E-2	81	0
12	1728	5184	238392	8	0.15E-1	143	0
13	2197	6591	356928	7	0.32E-1	222	-
14	2744	8232	518518	8	0.18E-1	380	-
15	3375	10125	733950	6	0.65E-1	543	-
16	4096	12288	1015680	7	0.42E-1	835	-
17	4913	14739	1377952	6	0.16E0	1270	-
18	5832	17496	1836918	7	0.21E0	1853	-
19	6859	20577	2410758	8	0.24E0	2335	-
20	8000	24000	3119800	8	0.59E0	3187	-

Table 1: Results of HS Technique applied to Moré-Wu Instance

In view of the computational results obtained, where the proposed HS methodology performed efficiently and robustly solving large instances, in comparison with Moré and Wu [3] or Hoai and Tao [1], we believe that it, alone or in combination with another algorithm, can represent a possible approach for dealing with real applications involving large geometric distance problems, such as protein folding problems.

References

[1] Hoai An, L. T. and Tao, P. D. (2000). “Large-Scale Molecular Conformation Via the Exact Distance Geometry Problem", Lecture Notes in Economics and Mathematical Systems, Vol. 481, pp. 260-277.

[2] Macambira, A.F.U.S. (2003). “Determinação de Estruturas de Proteínas via Suavização e Penalização Hiperbólica", M.Sc. Thesis COPPE UFRJ.

[3] Moré, J. J. and Wu, Z. (1997). “Global Continuation for Distance Geometry Problems", SIAM J. Optimization Vol. 7, no 3, pp. 814-836.

[4] Souza, M.F. (2010). “Suavização Hiperbólica Aplicada à Otimização de Geometria Molecular", D.Sc. Thesis COPPE UFRJ.

[5] Souza, M.F., Xavier, A.E., Lavor, C. and Maculan, N. (2011). “Hyperbolic Smoothing and Penalty Techniques Applied to Molecular Structure Determination", Operations Research Letters, Vol. 39, pp. 461-465, 2011, doi:10.1016/j.orl.2011.07.007.

[6] Xavier, A.E. (2003). “Convexificação do Problema de Distância Geométrica através da Técnica de Suavização Hiperbólica", Workshop em Biociências COPPE UFRJ.

DGA 2013, pp. 219 – 223.

Tetrahedra Determined by Volume, Circumradius and Face Areas *

Lu Yang[1,2] and Zhenbing Zeng[1]

[1] *Shanghai Key Laboratory of Trustworthy Computing, East China Normal University*
200062 Shanghai, China {lyang,zbzeng}@sei.ecnu.edu.cn

[2] *Chengdu Institute of Computer Application, Chinese Academy of Sciences*
200062 Chengdu, China, cdluyang@casit.ac.cn

Keywords: Tetrahedron, Volume, Circumradius, Area, Polynomial Equation.

1. The Problem and the Background

A tetrahedron is a polyhedron in the three dimensional space $\mathbb{R}^3$ composed of four triangular faces, three of which meet at each vertex. It is clear that the freedom of a tetrahedron is six, and therefore, given four appropriate positive numbers there may exist infinitely many non-isometric tetrahedra which four face areas are the given numbers. M. Mazur asked in [5] whether or not a tetrahedron is uniquely determined by its volume V, circumradius R and face areas A_1, A_2, A_3, A_4. A negative answer to this question was given by P. Lisoněk and B. Israel in [4] through constructing two or more non-congruent tetrahedra that have the same volume, circumradius and face areas. In [7] L. Yang and Z. Zeng showed that for the case $A_2 = A_3 = A_4$ a family of infinitely many non-congruent tetrahedra $T_{(x,y)}$ can be constructed, where (x, y) varies over a component of a cubic curve, such that all tetrahedra $T_{(x,y)}$ share the same volume, circumradius and face areas, and conjectured that for any six given positive constants V, R, A_1, A_2, A_3, A_4 where A_1, A_2, A_3, A_4 are pairwise distinct there are at most nine non-congruent tetrahedra can be constructed from the given parameters. One of the referees to that paper investigated the problem and observed that in this case it always leads to an equation $R(u) = 0$ of degree nine with at least one negative real root, where u is an edge of tetrahedron, which means that the number of the tetrahedra satisfying the given parameters is at most eight. In this paper, we present a proof to this fact by using the metric equations of tetrahedra and symbolic algebra. Our main result is the following theorem.

Theorem 1. *Given six positive numbers V, R, A_1, A_2, A_3, A_4. Then there are at most eight tetrahedra with volume V, circumradius R and four face areas A_1, A_2, A_3, A_4, except in the case that three of the values A_1, A_2, A_3, A_4 are equal.*

2. A Sketch of Proof of the Main Theorem

The proof of this theorem relies on the following five known results concerning metric invariants of tetrahedra and one new result on a necessary and sufficient condition for a tetrahedron to

*Supported by the 973 Program No. 2011CB302402 of China, NNSFC Grant No. 61021004, and the MOE Project No. 20110076110010 of China. Corresponding author: Zhenbing Zeng.

have a right angle dihedron. To state the following lemmas we first introduce the notation for metric invariants of tetrahedra. Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$, $A_i(1 \le i \le 4)$ the area of the face F_i opposite with vertex P_i, that is,

$$F_1 = P_2P_3P_4, F_2 = P_3P_4P_1, F_3 = P_4P_1P_2, F_4 = P_1P_2P_3,$$

V the volume and R the circumradius of the tetrahedron, respectively. Let $\theta_{i,j}(1 \le i, j \le 4)$ the dihedral angle formed by F_i and F_j, $d_{i,j}$ the distance between vertices P_i and P_j, as shown in the Fig. 1.

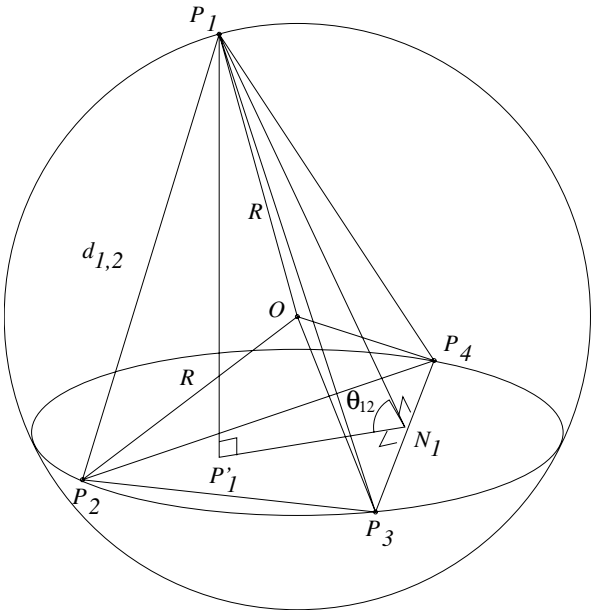


Figure 1: A tetrahedron $T = P_1P_2P_3P_4$ with circumcenter at O and circumradius R.

The following lemma shows that the algebraic sum of the projections of the three faces meet at vertex P_i on the face F_i equals to A_i.

Lemma 1. *Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$. Then*

$$\begin{aligned}
&A_2 \cdot cos(\theta_{1,2}) + A_3 \cdot cos(\theta_{1,3}) + A_4 \cdot cos(\theta_{1,4}) = A_1,\\
&A_3 \cdot cos(\theta_{2,3}) + A_4 \cdot cos(\theta_{2,4}) + A_1 \cdot cos(\theta_{2,1}) = A_2,\\
&A_4 \cdot cos(\theta_{3,4}) + A_1 \cdot cos(\theta_{3,1}) + A_2 \cdot cos(\theta_{3,2}) = A_3,\\
&A_1 \cdot cos(\theta_{4,1}) + A_2 \cdot cos(\theta_{4,2}) + A_3 \cdot cos(\theta_{4,3}) = A_4.
\end{aligned} \tag{1}$$

Note that $\theta_{i,j} = \theta_{j,i}$ in (1) for all i, j. The next lemma is a formula for computing volume of tetrahedra through face areas and dihedra (cf. Lee [2]).

Lemma 2. *Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$ and V, A_i $(1 \le i \le 4)$, $\theta_{i,j}, d_{i,j}$ $(1 \le i, j \le 4)$ as described before. Then*

$$\begin{aligned}
&3\,d_{1,2}\,V = 2\,A_3\,A_4\,\sin(\theta_{3,4}), \quad 3\,d_{3,4}\,V = 2\,A_1\,A_2\,\sin(\theta_{1,2}),\\
&3\,d_{1,3}\,V = 2\,A_2\,A_4\,\sin(\theta_{2,4}), \quad 3\,d_{2,4}\,V = 2\,A_1\,A_3\,\sin(\theta_{1,3}),\\
&3\,d_{1,4}\,V = 2\,A_2\,A_3\,\sin(\theta_{2,3}), \quad 3\,d_{2,3}\,V = 2\,A_1\,A_4\,\sin(\theta_{1,4}).
\end{aligned} \tag{2}$$

The following result is the well-known Cayley-Menger determinant (cf. [1] and[3]) for computing the volume of a tetrahedron through its six edges.

Lemma 3. *Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$. Let $g_{ij} = d_{i,j}^2\,(1 \le i, j \le 4)$ and*

$$M_V = \begin{bmatrix} 0 & g_{12} & g_{13} & g_{14} & 1 \\ g_{12} & 0 & g_{23} & g_{24} & 1 \\ g_{13} & g_{23} & 0 & g_{34} & 1 \\ g_{14} & g_{24} & g_{34} & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Then

$$V^2 = \det(M_V)/288, \tag{3}$$

where $\det(\cdot)$ *is the determinant of a square matrix.*

Let M be any square matrix and $\det(\mathrm{minor}(M, i, j))$ the determinant of the submatrix of A obtained by removing its i-th row and j-th column. The following formula (cf. [2]) is a generalization of the law of cosines to the tetrahedron.

Lemma 4. *Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$ and M_V the matrix defined in Lemma 3. Then*

$$16\,A_i\,A_j\,\cos(\theta_{i,j}) + \det(\mathrm{minor}(M_V, i, j)) = 0 \tag{4}$$

The circumradius R of a tetrahedron is connected with the six edges by the following result, see [1], [3] and [6].

Lemma 5. *Let $T = P_1P_2P_3P_4$ be a tetrahedron in $\mathbb{R}^3$ and M_V the matrix defined in Lemma 3. Then*

$$2\,R^2 = -\det(\mathrm{minor}(M_V, 5, 5))/\det(M_V). \tag{5}$$

The following lemma is a new result to be proved in this paper. Note that we need only to assure the existence of the polynomial F in the proof of Theorem 1.

Lemma 6. *Let V the volume, R the circumradius, A_1, A_2, A_3, A_4 face areas, and $\theta_{i,j}(1 \le i, j \le 4)$ dihedral angles of the tetrahedron $T = P_1P_2P_3P_4$. Then there exists a polynomial $F(x_{-1}, x_0, x_1, x_2, x_3, x_4)$ of real coefficients satisfying that*

$$\theta_{1,2} = \pi/2 \Leftrightarrow F(V, R, A_1, A_2, A_3, A_4) = 0.$$

The proof of Theorem 1 can be sketched very briefly as following. Assume that $V, R, A_1, \cdots, A_4$ are given positive real numbers and $T = P_1P_2P_3P_4$ is a tetrahedron which volume, circumradius and face area are $V, R, A_1, \cdots, a_4$ with respectively. If

$$F(V, R, A_1, A_2, A_3, A_4) = F(V, R, A_1, A_3, A_2, A_4) = F(V, R, A_1, A_4, A_3, A_4) = 0,$$

then $\theta_{1,2} = \theta_{1,3} = \theta_{1,4} = \pi/2$ according to lemma 6. It is clear that no tetrahedron in $\mathbb{R}^3$ satisfies this condition, so without loss of generality we may assume that $F(V, R, A_1, A_2, A_3, A_4) \neq 0$. Let $x = \cos(\theta_{1,2}), y = \cos(\theta_{1,3})$. In the first step, we express all other $\cos(\theta_{i,j})$ into rational fractions of x, y, A_1, A_2, A_3, A_4 by applying Lemma 1. Secondly, we use Lemma 2 to construct six equations that connect the edges with face areas and cosine of dihedral angles, namely

$$\begin{aligned} g_{i,j} \;=\; & \frac{4A_k^2A_l^2(1-\cos^2(\theta_{k,l}))}{9V^2} = \wp_{i,j}(x, y, A_1, A_2, A_3, A_4), \\ & (1 \le i < j \le 4, k, l \in \{1,2,3,4\} \setminus \{i, j\}, k < l). \end{aligned} \tag{6}$$

In the third step, we construct the Cayley-Menger determinant and construct an equation $p_1(x, y, V, A_1, A_2, A_3, A_4)$ that connects the cosine of dihedral angles with volume and face areas according to Lemma 4, and construct an equation that relates the circumradius to edges and hence to face areas and cosine of dihedral angles, according to Lemma 5, called $p_2(x, y, V, R, A_1, A_2, A_3, A_4)$. Under the assumption $-1 < x < 1, x \neq 0$, these two equations can be simplified to the following triangular form by symbolic computation.

$$r_1 := e_0 + e_1\, x + e_2\, x^2 + \cdots + e_9\, x^9 = 0, \quad r_2 := A(x) + B(x)\, y = 0, \tag{7}$$

where $e_0, e_1, \cdots, e_8, e_9$ are polynomial of V, R, A_1, A_2, A_3, A_4,

$$e_9 = -512 A_2^9 A_1^9 (A_4^2 - A_1^2)(A_3^2 - A_1^2)(A_4^2 - A_2^2)(A_3^2 - A_2^2),$$

and $A(x), B(x)$ are polynomials of V, R, A_1, A_2, A_3, A_4. In the final step, we prove the following facts:

1. $r_1(1) = (A_2 - A_1)^2 \cdot r_{11}^2 \cdot r_{12}^2 \geq 0, \quad r_1(-1) = (A_2 + A_1)^2 \cdot r_{21}^2 \cdot r_{22}^2 \geq 0,$
2. If $\neg \diamondsuit(A_1, A_2, A_3, A_4)$, then $\deg(r_1, x) \geq 1$ and $e_0 \neq 0$.

where $r_{11}, r_{12}, r_{21}, r_{22}$ are polynomials of V, R, A_1, A_2, A_3, A_4, and $\diamondsuit(A_1, A_2, A_3, A_4)$ stands for $(A_2 = A_3 = A_4) \vee (A_1 = A_3 = A_4) \vee (A_1 = A_2 = A_4) \vee (A_1 = A_2 = A_3)$, This immediately implies that $r_1(x) = 0$ has at most eight roots in the interval $(-1, 1)$. After getting $x = \cos(\theta_{1,2}), y = \cos(\theta_{1,3})$, the six edges $d_{i,j}$ of the tetrahedra can be obtained from (6).

3. An Unsolved Problem

Substituting randomly selected $V, R, A_1, \cdots, A_4$ into $r_1(x) = 0$ one may search the maximal number of real roots of (7) with numerical computation. However, we have not found any example of $V, R, A_1, \cdots, A_4$ so that $r_1(x) = 0$ has eight real roots in $(-1, 1)$ yet. Fig. 3 shows the record of a Monte Carlo experiment of this computation.

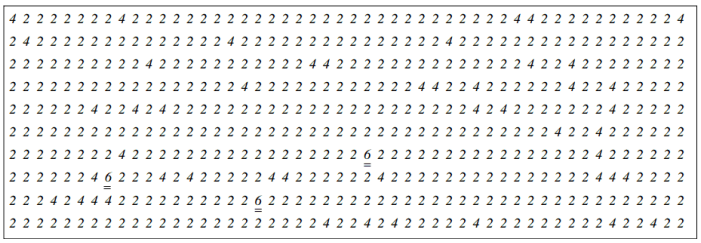


Figure 2: The numbers of real roots of $r_1(x) = 0$ in $(-1, 1)$ for 500 randomly generated examples.

We conjecture that the maximal number of tetrahedra in Theorem 1 is six.

References

[1] L. M. Blumenthal, Theory and Applications of Distance Geometry, Chelsea, New York, 1970.

[2] J. R. Lee, The Law of Cosines in a Tetrahedron. J. Korea Soc. Math. Ed. Ser. B: Pure Appl. Math. 4, 1-6, 1997.

[3] D. Michelucci, S. Foufou, Using Cayley-Menger determinants for geometric constraint solving. Proceedings of the ninth ACM symposium on Solid modeling and applications, pp.285-290, 2004.

[4] Petr Lisoněk, Robert B. Israel, Metric invariants of tetrahedra via polynomial elimination, Proceedings of the 2000 international symposium on Symbolic and algebraic computation, p.217-219, July 2000, St. Andrews, Scotland

[5] Problem 10717 (proposed by M. Mazur). Amer. Math. Monthly 106 (February 1999), 167.

[6] Lu Yang, Distance coordinates used in geometric constraint solving, in Automated Deduction in Geometry, F. Winkler (Ed.), LNAI 2930, pp. 216–229, Springer-Verlag 2004.

[7] Lu Yang, Zhenbing Zeng, An open problem on metric invariants of tetrahedra, Proceedings of the 2005 international symposium on Symbolic and algebraic computation, p.362-364, July 24-27, 2005, Beijing, China

Author Index