Global Optimization: from Theory to Implementation

Leo Liberti
DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy

Nelson Maculan
COPPE, Universidade Federal do Rio de Janeiro, P.O. Box, 68511, 21941-972 Rio de Janeiro, Brazil

Preface

The idea for this book was born on the coast of Serbia-Montenegro, in October 2003, when we were invited to the thirtieth Serbian Conference on Operations Research (SYM-OP-IS 2003). During those days we talked about many optimization problems, going from discussion to implementation in a matter of minutes, reaping good profits from the whole "hands-on" process, and having a lot of fun in the meanwhile. All the wrong ideas were weeded out almost immediately by failed computational experiments, so we wasted little time on those. Unfortunately, translating ideas into programs is not always fast and easy, and moreover the amount of literature about the implementation of global optimization algorithm is scarce.

The scope of this book is that of moving a few steps towards the systematization of the path that goes from the invention to the implementation and testing of a global optimization algorithm. The works contained in this book have been written by various researchers working at academic or industrial institutions; some very well known, some less famous but expert nonetheless in the discipline of *actually getting global optimization to work.*

The papers in this book underline two main developments in the implementation side of global optimization: firstly, the introduction of symbolic manipulation algorithms and automatic techniques for carrying out algebraic transformations; and secondly, the relatively wide availability of extremely efficient global optimization heuristics and metaheuristics that target large-scale nonconvex constrained optimization problems directly.

The book is divided in three parts. The first part is about new global optimization methods. The chapters in the first part are rather theoretical in nature, although a computational experiments section is always present. The second part is oriented towards the implementation, focusing on description of existing solvers and guidelines about building new global optimization software. This part follows two main trends: the first four chapters deal with continuous methods, the last three with combinatorial ones. The third (and last) part presents two applications of global optimization in Data Mining and Molecular Conformation.

More specifically, a lot of work has been carried out on the application of Variable Neighbourhood Search to global optimization (Chapters 6, 8, 10 and 11). A MultiStart-type algorithm based on low-discrepancy sequences generated deterministically has also been thoroughly explored (Chapters 5, 8). A full description of an API for interfacing to metaheuristic codes is given in Chapter 11. Deterministic algorithms can be found in Chapters 1 (Branch-and-Bound algorithms), 3 (a cutting-plane algorithm), 4 and 8.

As has been mentioned, a particularly important development is the introduction of symbolic manipulation algorithms in optimization. Chapter 7 describes a modelling language by which it is possible to keep track of the convexity property of the optimization problem being described. Although Chapter 7 is about convex programming, the role of convexity is so important in Branch-and-Bound-type algorithms for global zoptimization that it was decided to include it in this book. In Chapter 8 the reader can find the description of some symbolic algorithms for differentiation, algebraic simplification and generation of convex relaxations. Chapter 3 introduces some effective convexity transformations for a large class of multilinear problems, as well as discussing some nonlinear cuts. Chapter 10 employs even more sophisticated symbolic techniques about automated theorem proving.

Chapter 1 and 2 are more theoretical than most other chapters. Chapter 1 considers global optimization problems where the objective functions and constraints are difference of monotonic functions, and proposes some deterministic solution methods; Chapter 2 reports on a special local search method for reverse convex problems. In both chapters, a section on computational results is presented, discussing the efficiency of different solution approaches.

Chapter 4 describes one of the very few existing implementations of a deterministic global optimization software targeting robust nonconvex programming. In order to face the huge computational resources needed to solve multi-scenario nonconvex problems, the author proposes a Branch-and-Bound approach where the lower bounds are computed by solving a nonconvex Lagrangian relaxation through a standard global optimization algorithm. This multi-level solution method requires careful software design to obtain a working implementation.

Chapters 9 and 12 describe working implementations of commercial-grade software. In particular, Chapter 9 is about the Lipschitz Global Optimization (LGO) solver suite, and its embedding within the Mathematica software framework; Chapter 12 describes a solver for Mixed-Integer Linear Programming problems (sold by Process Systems Enterprise, Ltd.): this software relies on CORBA techniques to automate the parallelization and distributed running of the solver.

As far as the applications are concerned, Chapter 13 describes an extremely interesting class of problems arising in Data Mining and Nonlinear Classification. Chapter 14 describes a new way to generate instances for the Molecular Distance Geometry Problem, which is one of the hardest problems in Molecular Conformation.

Some of these papers have inter-relations and cross-references, due both to collaborations among the authors and to emergence of new trends in global optimization. Most of these inter-relations have been emphasized by means of footnotes, which have all been added by the editors.

We hope that the reader will find this book interesting and enlightening, and that it will serve as a source of ideas as well as a desktop companion for people who need to implement global optimization software.

Milano, Rio de Janeiro
June 2005

Leo Liberti
Nelson Maculan

Contents

Optimization under Composite Monotonic Constraints and Constrained Optimization over the Efficient Set
Hoang Tuy, N.T. Hoai-Phuong 3
1 Introduction 3
2 Some basic concepts and results of monotonic optimization 5
3 Problems With Composite Monotonic Constraints 7
4 Constrained Optimization over the Efficient Set 11
5 Solution Method for Problem (Q) 15
6 Improvements for problems (OWE) and (OE) 19
7 Problems With a Composite Monotonic Objective Function 25
8 Illustrative Examples and Computational Results 26
References 29

On a Local Search for Reverse Convex Problems
Alexander Strekalovsky 33
1 Introduction 33
2 Some features of RCP 34
3 Local search methods 36
4 Computational testing 40
5 Conclusion 42
References 42

Some transformation techniques in Global Optimization
Tapio Westerlund 45
1 Introduction 45
2 The MINLP Problem 46
3 The transformation approach 47
4 Examples of transformations 52
5 The GGPECP algorithm 55
6 Convergence to the globally optimal solution 57
7 A numerical example 59

8 Some aspects on the numerical solution approach 64
9 Conclusions 71
References 71

Solving nonlinear mixed integer stochastic problems: a global perspective
Maria Elena Bruni 75
1 Introduction 76
2 Motivations 76
3 SMINLP: state of the art 77
4 Problem formulation 84
5 The two-phase solution approach 86
6 Illustrative Application: The Stochastic Trim Loss Problem 98
7 Concluding Remarks 104
References 106

Application of Quasi Monte Carlo Methods in Global Optimization
Sergei Kucherenko 111
1 Introduction 111
2 Analysis of Quasirandom Search Method 114
3 Single linkage and multilevel single linkage methods 117
4 Computational experiments 120
5 Conclusion 131
References 131

GLOB – A new VNS-based software for global optimization
M. Drazić, V. Kovacevic–Vujcić, M. Cangalović, N. Mladenović 135
1 Introduction 135
2 VNS methodology 136
3 Software package GLOB 137
4 Numerical experiments 141
5 Conclusion 147
References 148

Disciplined Convex Programming
Michael Grant, Stephen Boyd, Yinyu Ye 155
1 Introduction 155
2 Motivation 156
3 Convex programming 162
4 Modeling frameworks 169
5 Disciplined convex programming 171
6 The convexity ruleset 172
7 The atom library 183
8 Verification 188
9 Creating disciplined convex programs 191

10 Implementing atoms 193
11 Conclusion 199
References 200

Writing Global Optimization Software
Leo Liberti 211
1 Introduction 211
2 Global Optimization algorithms 214
3 Global Optimization software 223
4 Optimization software framework design 232
5 Symbolic manipulation of mathematical expressions 240
6 Local solvers 247
7 Global solvers 248
8 Conclusion 257
References 258

Mathoptimizer Professional: Key features and illustrative applications
János D. Pintér, Frank J. Kampas 263
1 Introduction 263
2 Global Optimization 266
3 LGO Solver Suite 267
4 Mathoptimizer Professional 268
5 Illustrative applications: solving sphere packing models 271
6 Conclusions 276
References 277

Variable Neighborhood Search for Extremal Graphs 14: The AutoGraphiX 2 System
M. Aouchiche, J.M. Bonnefoy, A. Fidahoussen, G. Caporossi, P. Hansen, L. Hiesse, J. Lacheré, A. Monhait 281
1 Introduction 281
2 AGX 2 Interactive Functions 283
3 Algebraic Syntax Used in AutoGraphiX 291
4 Optimization Using Variable Neighborhood Search 294
5 AutoGraphiX Tasks 299
6 Automated Proofs 301
7 Some Examples 305
8 Conclusion 308
References 308

From Theory to Implementation: Applying Metaheuristics.
I.J. García del Amo, F. García López, M. García Torres,, B. Melián Batista, J.A. Moreno Pérez, J.M. Moreno Vega 311
1 Introduction 311
2 Class Hierarchy 316

3 Implementation: The p-Median Problem . . . 333
4 Conclusions . . . 338
References . . . 339

ooMILP – **A C++ Callable Object-oriented Library and the Implementation of its Parallel Version using CORBA**
Panagiotis Tsiakis, Benjamin Keeping . . . 353
1 Introduction . . . 353
2 *ooMILP* Overview . . . 356
3 C++ objects and pre-CORBA serial implementation . . . 357
4 Initial CORBA Version . . . 361
5 Partially decomposable MILPs . . . 366
6 Parallel solution software architecture . . . 368
7 CONCLUSIONS . . . 375
References . . . 375

Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy
R. Andreani, J.M. Martinez, M. Salvatierra, F. Yano . . . 379
1 Introduction . . . 379
2 Local Algorithm . . . 381
3 Lissajous motions . . . 382
4 Global algorithm . . . 384
5 Hidden patterns . . . 387
6 Numerical experiments . . . 388
7 Conclusions . . . 394
References . . . 397

On generating instances for the Molecular Distance Geometry Problem
Carlile Lavor . . . 405
1 Introduction . . . 405
2 Moré-Wu instances . . . 406
3 New instances . . . 407
4 Conclusion . . . 413
References . . . 414

Index . . . 415

Part I: Methods

Optimization under Composite Monotonic Constraints and Constrained Optimization over the Efficient Set

Hoang Tuy and N.T. Hoai-Phuong

Institute of Mathematics, VAST, 18 Hoang Quoc Viet, 10307 Hanoi, Vietnam
{htuy,htphuong}@math.ac.vn

Summary. We present a unified approach to a class of nonconvex global optimization problems with composite monotonic constraints. (By composite monotonic function is meant a function which is the composition of a monotonic function on $\mathbb{R}^n$ with a mapping from $\mathbb{R}^n \to \mathbb{R}^p$ with $p \leq n$.) This class includes problems with constraints involving products of linear functions, sums of ratio functions, etc., and also problems of constrained optimization over efficient/weakly efficient points. The approach is based on transforming the problem into a monotonic optimization problem in the space $\mathbb{R}^p$, which can then be efficiently solved by recently developed techniques. Nontrivial numerical examples are presented to illustrate the practicability of the approach.

Key words: Global optimization. Monotonic optimization, difference of monotonic (d.m.) optimization. Composite monotonic constraint. Nonconvex optimization. Branch-reduce-and-bound method. Constrained optimization over the efficient/weakly efficient set. Multiplicative constraint. Sum-of-ratio constraint.
Mathematics Subject Classification 90C26, 65K05, 90C20, 90C30, 90C56, 78M50

1 Introduction

Convexity is essential to modern optimization theory. However, it is not always the natural property to be expected from many nonlinear phenomena. Another property, perhaps at least as pervasive in the real world as convexity, is monotonicity.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *increasing* on a box $[a, b] = \{x \in \mathbb{R}^n \mid a \leq x \leq b\}$ if $f(x) \leq f(x')$ whenever $a \leq x \leq x' \leq b$ (throughout this paper, inequalities between vectors are understood in the componentwise sense); it is said to be decreasing if $-f(x)$ is increasing, monotonic if it is either increasing or decreasing. A function which can be represented as a difference

On a Local Search for Reverse Convex Problems

Alexander Strekalovsky

Institute of System Dynamics and Control Theory SB of RAS, 134 Lermontov St., Irkutsk-33, 664033 Russia strekal@icc.ru

Summary. In this paper we propose two variants of Local Search Method for reverse convex problems with the first based on well-known theorem of H. Tuy as well as on Linearization Principle. The second variant is due to the idea of J. Rosen. Also we demonstrate the practical effectiveness of the proposed methods by a computational testing.

Key words: Nonconvex optimization, reverse convex problem, local search, computational testing.

1 Introduction

The present situation in Continuous Nonconvex Optimization may be viewed as dominated by methods transferred from other sciences [1, 3], as Discrete Optimization (Branch&Bound, cuts methods, outside and inside approximations, vertex enumeration and so on), Physics, Chemistry (simulated annealing methods), Biology (genetic and ant colony algorithms) etc.

On the other hand the classical method [11] of convex optimization have been thrown aside because of its inefficiency [1, 6]. As well-known the conspicuous limitation of convex optimization methods applied to nonconvex problems is their ability of being trapped at a local extremum or even a critical point depending on a starting point [1, 3]. So, the classical apparatus shows itself inoperative for new problems arising from practice.

In such a situation it seems very probable to create an approach for finding just a global solution to nonconvex problems in particular to Reverse Convex Problem (RCP) on one side connected with Convex Optimization Theory and secondly using the methods of Convex Optimization.

Nevertheless we ventured to propose such an approach [12] and even to advance the following principles of Nonconvex Optimization.

Some transformation techniques in Global Optimization

Tapio Westerlund

Process Design Laboratory, Åbo Akademi University, Biskopsgatan 8, FIN-20500 ÅBO, Finland tapio.westerlund@abo.fi

Summary. In this chapter some transformation techniques, useful in deterministic global optimization, are discussed. With the given techniques, a general class of nonconvex MINLP (mixed integer non-linear programming) problems can be solved to global optimality. The transformations can be applied to signomial functions and the feasible region of the original problem can be convexified and overestimated by the transformations. The global optimal solution of the original nonconvex problem can be found by solving a sequence of convexified MINLP sub-problems. In each such iteration a part of the infeasible region is cut off and the algorithm terminates when a solution point is sufficiently close to or within the feasible region of the original problem. The principles behind the algorithm are given in this chapter and numerical examples are used to illustrate how the global optimal solution is obtained with the algorithm.

Key words: Transformation techniques, reformulation, mixed integer nonlinear programming, signomial functions.

1 Introduction

The transformations, discussed in this chapter, are applicable to signomial functions and can be applied to problems where the objective function or some of the constraints are composed of a convex and a signomial function. The transformations are made in two steps. Single variable transformations are first applied term-wise to convexify every signomial term. Secondly, the transformations are selected such that the signomial terms are not only convexified but also underestimated. The latter property is important when developing a global optimization approach and this property is obtained by carefully selecting the transformations such that they can be applied together with piecewise linear approximations of the inverse transformations. This allows us not only to convexify and to underestimate every generalized signomial constraint but also to convexify the entire nonconvex problem and to overestimate the feasible region of it. When generalized signomial constraints are

Solving nonlinear mixed integer stochastic problems: a global perspective

Maria Elena Bruni

Dipartimento di Elettronica, Informatica e Sistemistica, Università degli Studi della Calabria, 87030 Rende (CS), Italy mebruni@deis.unical.it

Summary. In this paper, we present a novel approach for solving nonlinear mixed integer stochastic programming problems. In particular, we consider two stage stochastic problem with nonlinearities both in the objective function and constraints, pure integer first stage and mixed-integer second stage variables. We formulate the problem by a scenario based representation, adding linear nonanticipativity constraints coming from splitting the first stage decision variables. In the separation phase we fully exploit the partial decomposable structure of SMINLPs. This allows to deal with a separable nondifferentiable problem, which can be solved by Lagrangian dual based procedure. In particular, we propose a specialization of the Randomized Incremental Subgradient Method- proposed by Bertsekas(2001)- which takes dynamically into account the information relative to the scenarios. The coordination phase is aimed at enforcing coordination among the solutions of the scenario subproblems. More specifically, we use a branch and bound in order to enforce the feasibility of the relaxed nonanticipativity constraints. In order to make more efficient the overall method, we embed the Lagrangian iteration in a branch and bound scheme, by avoiding the exact solution of the dual problem and we propose an early branching rule and a worm start procedure to use within the Branch and Bound tree. Although SMINLPs have many application contexts, this class of problem has not been adequately treated in the literature. We propose a stochastic formulation of the Trim Loss Problem, which is new in the literature. A formal mathematical formulation is provided in the framework of two-stage stochastic programming which explicitly takes into account the uncertainty in the demand. Preliminary computational results illustrate the ability of the proposed method to determine the global optimum significantly decreasing the solution time. Furthermore, the proposed approach is able to solve instances of the problem intractable with conventional approaches.

Key words: Stochastic Programming, MINLP, Trim Loss, Lagrangian decomposition, Branch-and-Bound.

Application of Quasi Monte Carlo Methods in Global Optimization

Sergei Kucherenko

CPSE, Imperial College London, SW7 2AZ, UK s.kucherenko@imperial.ac.uk

Summary. It has been recognized through theory and practice that uniformly distributed deterministic sequences provide more accurate results than purely random sequences. A quasi Monte Carlo (QMC) variant of a multi level single linkage[1] (MLSL) algorithm for global optimization is compared with an original stochastic MLSL algorithm for a number of test problems of various complexities. An emphasis is made on high dimensional problems. Two different low-discrepancy sequences (LDS) are used and their efficiency is analysed. It is shown that application of LDS can significantly increase the efficiency of MLSL. The dependence of the sample size required for locating global minima on the number of variables is examined. It is found that higher confidence in the obtained solution and possibly a reduction in the computational time can be achieved by the increase of the total sample size N. N should also be increased as the dimensionality of problems grows. For high dimensional problems clustering methods become inefficient. For such problems a multistart method can be more computationally expedient.

Key words: stochastic methods, low-discrepancy sequences, multi level single linkage method

1 Introduction

The motivation for this paper is to develop further efficient and robust optimization methods. Let $f(\boldsymbol{x}) : R^n \to R$ be a continuous real valued objective function. A nonlinear global optimization problem is defined as follows:

$$\min f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n \tag{1}$$

subject to

[1] Also see Chapter 8, Sections 2.1, 7.1. In particular, the SobolOpt solver within the *oo$\mathcal{OPS}$* software framework shares the same code as the software implementation proposed in this paper.

Part II: Implementations

GLOB – A new VNS-based software for global optimization

M. Drazić[1], V. Kovacevic–Vujcić[2], M. Cangalović[2], and N. Mladenović[3][4]

[1] Faculty of Mathematics, University of Belgrade, Belgrade mdrazic@sezampro.yu
[2] Faculty of Organizational Sciences, University of Belgrade, Belgrade {verakov,canga}@fon.bg.ac.yu
[3] Mathematical Institute, Serbian Academy of Arts and Sciences, Belgrade nenad@mi.sanu.ac.yu
[4] GERAD and Ecole des Hautes Commerciales, Montreal

Summary. We describe an application of Variable Neighbourhood Search (VNS) methodology to continuous global optimization problems with box constraints. A general VNS algorithm is implemented within the software package GLOB. The tests are performed on some standard test functions and on a class of NP–hard global optimization problems arising in practice. The computational results show the potential of the new software.

Key words: Metaheuristics, variable neighborhood search.

1 Introduction

Global optimization problems have the form

$$\operatorname*{global\,min}_{x \in X} f(x)$$

where $f : R^n \to R$ is a continuous function on an open set containing X and X is a compact set. In most cases of practical interest global optimization is very difficult because of the presence of many local minima, the number of which tends to grow exponentially with the dimension of the problem. Besides, in general it is only possible to design methods that offer an ε-guarantee to find the global minimum. Nevertheless, a number of methods for global optimization problems have been proposed, both deterministic and nondeterministic (for a comprehensive bibliography see [15, 16, 18]).

There are two common approaches to finding the global minimum. The first, so called Multistart[5] Local Search (MS) consists of generating a set of

[5] Also see Chapters 5, 8 (Sections 2.1, 7.1).

Disciplined Convex Programming

Michael Grant[1], Stephen Boyd[1], and Yinyu Ye[1,2]

[1] Department of Electrical Engineering, Stanford University
{mcgrant,boyd,yyye}@stanford.edu
[2] Department of Management Science and Engineering, Stanford University

Summary. A new methodology for constructing convex optimization models called *disciplined convex programming* is introduced. The methodology enforces a set of conventions upon the models constructed, in turn allowing much of the work required to analyze and solve the models to be automated.

Key words: Convex programming, automatic verification, symbolic computation, modelling language.

1 Introduction

Convex programming is a subclass of nonlinear programming (NLP) that unifies and generalizes least squares (LS), linear programming (LP), and convex quadratic programming (QP). This generalization is achieved while maintaining many of the important, attractive theoretical properties of these predecessors. Numerical algorithms for solving convex programs are maturing rapidly, providing reliability, accuracy, and efficiency. A large number of applications have been discovered for convex programming in a wide variety of scientific and non-scientific fields, and it seems clear that even more remain to be discovered. For these reasons, convex programming arguably has the potential to become a ubiquitous modeling technology alongside LS, LP, and QP. Indeed, efforts are underway to develop and teach it as a distinct discipline [29, 21, 115].

Nevertheless, there remains a significant impediment to the more widespread adoption of convex programming: the high level of expertise required to use it. With mature technologies such as LS, LP, and QP, problems can be specified and solved with relatively little effort, and with at most a very basic understanding of the computations involved. This is not the case with general convex programming. That a user must understand the basics of convex analysis is both reasonable and unavoidable; but in fact, a much deeper understanding is required. Furthermore, a user must find a way to transform

Writing Global Optimization Software

Leo Liberti

DEI, Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy
liberti@elet.polimi.it

Summary. Global Optimization software packages for solving Mixed-Integer Nonlinear Optimization Problems are usually complex pieces of codes. There are three main difficulties in coding a good GO software: embedding third-party local optimization codes within the main global optimization algorithm; providing efficient memory representations of the optimization problem; making sure that every part of the code is fully re-entrant. Finding good software engineering solutions for these difficulties is not enough to make sure that the outcome will be a GO software that works well, of course. However, starting from a sound software design makes it easy to concentrate on improving the efficiency of the global optimization algorithm implementation. In this paper we discuss the main issues that arise when writing a global optimization software package, namely software architecture and design, symbolic manipulation of mathematical expressions, choice of local solvers and implementation of global solvers.

Key words: MINLP, symbolic computation, multistart, variable neighbourhood search, branch-and-bound, implementation, software design.

1 Introduction

The object of Global Optimization (GO) is to find a solution of a given nonconvex mathematical programming problem. By "solution" we mean here a *global* solution, as opposed to a *local* solution; i.e., a point where the objective function attains the optimal value with respect to the whole search domain. By contrast, a solution is local if it is optimal with respect to a given neighbourhood. We require the objective function and/or the feasible region to be nonconvex because in convex mathematical programming problems every local optimum is also a global one. Consequently, any method solving a convex problem locally also solves it globally.

In this paper we address Mixed-Integer Nonlinear Programming (MINLP) problems in their most general setting:

Mathoptimizer Professional: Key features and illustrative applications

János D. Pintér[1] and Frank J. Kampas[2]

[1] Pintér Consulting Services, Inc., Halifax, Nova Scotia, Canada
jdpinter@hfx.eastlink.ca, http://www.pinterconsulting.com
[2] WAM Systems, Inc., Plymouth Meeting, PA, USA fkampas@wamsystems.com,
http://www.wamsystems.com

Summary. Integrated scientific-technical computing (ISTC) environments play an increasing role in advanced systems modeling and optimization. MathOptimizer Professional (MOP) has been recently developed to solve nonlinear optimization problems formulated in the ISTC system *Mathematica*. We introduce this software package, and review its key functionality and options. MOP is then used to solve illustrative circle packing problems, including both well-frequented models and a new (more difficult) model-class.

Key words: Integrated computing systems, *Mathematica*, LGO solver suite, MathOptimizer Professional, circle packings, illustrative results.

1 Introduction

Operations Research (O.R.) provides a consistent quantitative framework and techniques, to assist analysts and decision-makers in finding "good" (feasible) or "best" (optimal) solutions in a large variety of contexts. For an overview of prominent O.R. application areas, consult e.g. the 50^{th} anniversary issue of the journal *Operations Research* (2002).

A formal procedure aimed at finding optimized decisions consists of the following key steps.

- Conceptual description of the decision problem at a suitable level of abstraction that retains all essential attributes, but omits secondary details and circumstances.
- Development of a quantitative model that captures the key elements of the decision problem, in terms of decision variables and functional relationships among them.
- Development and/or adaptation of an algorithmic solution procedure, in order to explore the set of feasible solutions, and to select the best decision.

Variable Neighborhood Search for Extremal Graphs 14: The AutoGraphiX 2 System

M. Aouchiche[1], J.M. Bonnefoy[2], A. Fidahoussen[2], G. Caporossi[3], P. Hansen[3], L. Hiesse[4], J. Lacheré[5], and A. Monhait[6]

[1] Ecole Polytechnique de Montréal, Canada mustapha.aouchiche@gerad.ca
[2] ISIMA, Clermont-Ferrand, France
[3] GERAD and HEC Montréal, Canada
{gilles.caporossi,pierre.hansen}@gerad.ca
[4] Institut d'Informatique d'Entreprise, Clamart, France
[5] Ecole Polytechnique, Univ. de Nantes, France
[6] CUST, Univ. Blaise Pascal, Clermont-Ferrand, France

Summary. The AutoGraphiX (AGX) system for computer assisted or, for some of its functions, fully automated graph theory was developed at GERAD, Montreal since 1997. We report here on a new version (AGX 2) of that system. It contains many enhancements, as well as a new function for automated proof of simple propositions. Among other results, AGX 2 led to several hundred new conjectures, ranking from easy ones, proved automatically, to others requiring longer unassisted or partially assisted proofs, to open ones. Many examples are given, illustrating AGX 2's functions and the results obtained.

Key words: Graph theory, automated system, computer-assisted, AGX, automated proof, conjecture, refutation.

1 Introduction

Computers have been extensively used in graph theory and its applications to various fields since the fifties of the last century. The main use was computation of the values of graph invariants, *i.e.,* quantities such as the independence and chromatic numbers, the radius or the diameter of a graph, which do not depend on the labelling of its vertices or edges. In addition to such tasks of intelligent number-crunching (which imply the design of exact algorithms or heuristics as well as their efficient implementation with well-chosen data-structures [34, 35, 40, 41]), computers can also be used for graph drawing [17, 18] and for advancing the theory itself, *i.e.,* finding in a computer-assisted or sometimes fully automated way conjectures, proofs and refutations. See [27] for a survey and discussion of systems designed for that purpose, focussed on

From Theory to Implementation: Applying Metaheuristics.

I.J. García del Amo, F. García López, M. García Torres,, B. Melián Batista, J.A. Moreno Pérez, and J.M. Moreno Vega

Departamento de Estadística, I.O. y Computación
Universidad de La Laguna, 38271 La Laguna, Spain
{igdelamo,fcgarcia,mgarciat,mbmelian,jamoreno,jmmoreno}@ull.es

Summary. Metaheuristics are strategies to design heuristic procedures to find high quality solutions to an optimization problem. This chapter focuses on the implementation aspects of heuristic algorithms based on metaheuristics, using an object oriented approach. This programming paradigm takes advantage of the common parts shared by codes that implement different metaheuristic procedures. We give a class hierarchy for metaheuristics that permits quickly generate algorithms from existing metaheuristic codes for specific problems by extending a few classes and adding the problem functionality. It also allows the development of new metaheuristic algorithms without programming from scratch the basis of the procedure. It consists of selecting an appropriate class with the closest functionality, and extending it to add the core of the algorithm. The purpose of this hierarchy is thus to provide an extensible model for a quick implementation of metaheuristics and the problem structures associated with them.

Key words: Metaheuristic, implementation, API, variable neighbourhood search, genetic algorithms.

1 Introduction

Metaheuristics are strategies to design heuristic procedures. Since the first time the word *metaheuristic* appeared in the seminal paper of Tabu Search by Fred Glover in 1986 [8], there have been a lot of papers, reviews and books on Metaheuristics [29, 37, 31, 2, 12]. The classification of metaheuristics is usually based on the kind of procedures for which they are designed. For example, there are constructive metaheuristics like GRASP [30], evolutive metaheuristics like Genetic Algorithms [28] or neighborhood metaheuristics like the classical greedy local search. However, other possible classifications of metaheuristics are given by the computational tool or technique considered fundamental for the procedure, like Neural Networks [27] or Ant Colony Systems [3]. Some of the proposed algorithms are designed following not only one

oo$\mathcal{MILP}$ – A C++ Callable Object-oriented Library and the Implementation of its Parallel Version using CORBA

Panagiotis Tsiakis and Benjamin Keeping

Process Systems Enterprise Ltd, 107a Hammersmith Bridge Rd, London W6 9DA, United Kingdom {p.tsiakis,b.keeping}@psenterprise.com

Summary. Process systems engineering is one of the many areas in which mixed integer optimisation formulations have been successfully applied. The nature of the problems requires specialised solution strategies and computer packages or callable libraries able to be extended and modified in order to accommodate new solution techniques. Object-oriented programming languages have been identified to offer these features. Process system applications are usually of large scale, and require modelling and solution techniques with high level of customisation. *oo*$\mathcal{MILP}$ is a library of C++ callable procedures for the definition, manipulation and solution of large, sparse mixed integer linear programming (MILP) problems without the disadvantages of many existing modelling languages. We first present a general approach to the packaging of numerical solvers as software components, derived from material developed for the CAPE-OPEN project. The presentation is in the context of construction and solution of Mixed Integer Linear Programming (MILP) problems. We then demonstrate how this package, based on the use of CORBA interfaces for synchronous execution within a single process, can be adapted with a minimum of problem-specific changes to provide a distributed solution.

Key words: Object-oriented programming, callable library, optimization, parallel computing, interface, branch-and-bound

1 Introduction

A wide variety of engineering, industrial, and business applications are formulated as Mixed Integer Linear Programming (MILP) problems. The high utilisation of MILPs in all these areas requires flexible integer optimisation techniques and supporting software, able to be customised for different applications. These features are provided by object-oriented languages. These languages are suitable for creating reusable software in an error-free manner.

A modelling language is expected to provide the ability to write a model of the formulation in a manner that can be manipulated in many ways. Currently,

Part III: Applications

Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy

R. Andreani, J.M. Martinez, M. Salvatierra, and F. Yano

Department of Applied Mathematics, IMECC-UNICAMP, University of
Campinas, CP 6065, 13081-970 Campinas SP, Brazil
{andreani,martinez,otita,yano}@ime.unicamp.br

Summary. The OVO (Order-Value Optimization) problem consists in the minimization of the Order-Value function $f(x)$, defined by $f(x) = f_{i_p(x)}(x)$, where $f_{i_1(x)}(x) \leq \ldots \leq f_{i_m(x)}(x)$. The functions $f_1, \ldots, f_m$ are defined on $\Omega \subset \mathbb{R}^n$ and p is an integer between 1 and m. When x is a vector of portfolio positions and $f_i(x)$ is the predicted loss under the scenario i, the Order-Value function is the discrete Value-at-Risk (VaR) function, which is largely used in risk evaluations. The OVO problem is continuous but nonsmooth and, usually, has many local minimizers. A local method with guaranteed convergence to points that satisfy an optimality condition was recently introduced by Andreani, Dunder and Martínez. The local method must be complemented with a global minimization strategy in order to be effective when m is large. A global optimization method is defined where local minimizations are improved by a tunneling strategy based on the harmonic oscillator initial value problem. It will be proved that the solution of this initial value problem is a smooth and dense trajectory if Ω is a box. An application of OVO to the problem of finding hidden patterns in data sets that contain many errors is described. Challenging numerical experiments are presented.

Key words: Order-Value optimization, local methods, harmonic oscillator, tunneling, hidden patterns.

1 Introduction

Given m continuous functions $f_1, \ldots, f_m$, defined in a domain $\Omega \subset \mathbb{R}^n$ and an integer $p \in \{1, \ldots, m\}$, the $p-$Order-Value (OVO) function f is given by

$$f(x) = f_{i_p(x)}(x)$$

for all $x \in \Omega$, where $i_p(x)$ is an index function such that

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \ldots \leq f_{i_p(x)}(x) \leq \ldots \leq f_{i_m(x)}(x).$$

On generating instances for the Molecular Distance Geometry Problem

Carlile Lavor

Universidade do Estado do Rio de Janeiro (IME-UERJ), Rua S. Francisco Xavier
524, sala 6018-D, Rio de Janeiro - RJ, 20550-900, Brazil carlile@ime.uerj.br

Summary. The molecular distance geometry problem can be stated as the determination of the three-dimensional structure of a molecule using a set of distances between pairs of atoms. It can be formulated as a global minimization problem, where the main difficulty is the exponential increasing of local minimizers with the size of the molecule. The aim of this study is to generate new instances for the molecular distance geometry problem that can be used in order to test algorithms designed to solve it.

Key words: Molecular distance geometry problem, instance generation, NMR spectroscopy.

1 Introduction

The molecular distance geometry problem (MDGP) can be defined as the problem of finding Cartesian coordinates $x_1, ..., x_N \in \mathbb{R}^3$ of the atoms of a molecule such that

$$||x_i - x_j|| = d_{i,j} \qquad ([i,j] \in S), \tag{1}$$

where S is the set of pairs of atoms $[i,j]$ whose Euclidean distances $d_{i,j}$ are known. If all distances are given, the problem can be solved in linear time [3]. Otherwise, the problem is NP-hard [8].

The distances $d_{i,j}$, in (1), can be obtained, for example, with nuclear magnetic resonance (NMR) data and with knowledge on bond lengths and bond angles of a molecule. Usually, NMR data only provide distances between certain close-range hydrogen atoms [1].

The MDGP can be formulated as a global minimization problem, where the objective function can be given by

$$f(x_1, ..., x_N) = \sum_{[i,j] \in S} \left(||x_i - x_j||^2 - d_{i,j}^2 \right)^2 . \tag{2}$$