Abstracts

IV ALIO/EURO Workshop on Applied Combinatorial Optimization

“Celebrating the 60th birthdays of Nelson Maculan and Andres Weintraub”

Edited by
Celso C. Ribeiro

Organized by
Universidad de Concepción, Chile
Catholic University of Rio de Janeiro, Brazil

Pucón, Chile
November 4-6, 2002
Program Committee:

Celso C. Ribeiro (Chair)
Martine Labbé
Abilio Lucena
Francesco Maffioli
Federico Malcocelli
Lorena Pradella
Catherine Roucairol
Gd C. de Souza
# Table of Contents

Preface .................................................................................................................. ix

Variable Neighborhood Search for the Optimization of the Scheduling of Terrestrial Production Drills  
DARIO J. ALOISE, DANIEL ALOISE, CAROLINE T. M. ROCHA, JORGE D. MELO, AND CELSO C. RIBEIRO ................................................................. 1

On the Location Problems Arising in UMTS Network Planning  
EDOARDO AMAULDI, ANTONIO CAPONE, AND FEDERICO MALUCELLI ......................... 3

Circular Arc Models and Algorithms for Packet Scheduling in Smart Antennas  
EDOARDO AMAULDI, ANTONIO CAPONE, AND FEDERICO MALUCELLI ........................... 5

A Mixed Integer Disjunctive Formulation for the Transmission Network Expansion Problem  
LAURA BAHNENSE, GERSON OLIVEIRA, MARIO V. PEREIRA, AND SERGIO GRANVILLE .......... 7

Performance Evaluation of Reliable Multicast Algorithm Based on Recovery Tree  
MARTA BARRIA AND REINALDO VALLEJOS ................................................................. 9

Tabu Search Approach to Sequencing DNA Chains  
JACEK BLAZEWICZ AND MARTA KASPRZAK ............................................................... 11

Computational Complexity of Isothermic DNA Sequencing  
JACEK BLAZEWICZ AND MARTA KASPRZAK ............................................................... 13

Randomized Heuristics for Scene Recognition by Inexact Graph Matching  
MARIA C. BOERES, CELSO C. RIBEIRO, AND ISABELLE BLOCH ................................. 15

Worst-case Performance of Wong’s Steiner Tree Heuristic  
HUGO BRAVO-AZLÁN AND ALFREDO CANDIA-VÉJAR .................................................. 19

A Specialized Version of Benders Decomposition Applied to a General Machine Sequencing Model  
RICARDO S. DE CAMARGO AND CARLOS ROBERTO V. DE CARVALHO .......................... 21

Models and Algorithms for Staff Scheduling Problems  
ALBERTO CAPRARA, MICHELE MONACI, AND PAOLO TOTH ....................................... 25

Greedy Algorithms for the 3-dimensional Assignment Problem  
MANUEL V. CENTENO AND JUAN JOSÉ SALAZAR ...................................................... 29


Application of Descents with Mutations to the Aggregation of Relations
Irene Charon and Olivier Hudry ........................................... 31

Approximate Combinatorial Optimization Models for Large-Scale Production Lot Sizing and Scheduling with Sequence-Dependent Setup Times
Alistair R. Clark ................................................................. 33

Optimal Procurement Decisions in the Presence of Total Quantity Discounts and Alternative Product Recipes
Yves Chama, Rodrigo Pascual, and Andrea Torres ......................... 35

Integrating Tabu Search and Constraint Programming
Mauro Dell’Amico and Andrea Lodi ........................................... 37

Clique-transversals and Clique-independent Sets on Helly Circular-arc Graphs
Guillermo Durán, Min Chhi Lin, Sergio Mera, and Jayme L. Szwarcfiter ........................................... 39

Discrete Capacity and Nonbifurcated Flow Assignment Algorithm in Communication Networks
Ricardo P. M. Ferreira, J. A. S. Monteiro, and Henrique P. L. Luna ........................................... 41

A Parallel ProtoG Algorithm to Determine the Optimum Configuration of Natural Gas Cogeneration Systems
Marco C. Goldbarg, Elizabeth F. Gouvea, and Francisco D. M. Neto ........................................... 43

A ProtoG Algorithm Applied to the Traveling Salesman Problem
Marco C. Goldbarg, Elizabeth F. Gouvea, and Cezar M. P. de Souza .............. 47

The Distribution System Service Restoration: An Exact Method and a Heuristic Search Method

Multi-item Capacitated Lotsizing Problems with Setup Times: Column Generation and Lagrangean Heuristics
Monique Guignard, Choonho Ryu, Hang Qian, and Lydia Dowlath ........................................... 53

Probabilistic Satisfiability
Pierre Hansen ................................................................. 57

An Approach to the Multiple Traveling Salesmen Problem with Limitant Factors
José G. Hernández, María J. García, Jonathan Benaim, and Simón Tache ........................................... 65
<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Hybrid Metaheuristic for the Multi-objective Knapsack Problem</td>
<td>SKANDER KORT</td>
<td>69</td>
</tr>
<tr>
<td>Exact Algorithms for the Job Sequencing and Tool Switching Problem</td>
<td>GILBERT LAPORTE, JUAN JOSÉ SALAZAR, AND FRÉDÉRIC SEMET</td>
<td>71</td>
</tr>
<tr>
<td>Relax and Cut Algorithms</td>
<td>ABILIO LUCENA</td>
<td>73</td>
</tr>
<tr>
<td>Column Generation Technique in Network Design</td>
<td>NELSON MACULAN, GÉRARD PLATEAU, AND IRENE LOISEAU</td>
<td>77</td>
</tr>
<tr>
<td>A Quadratic Algorithm for Projecting a Vector on the Intersection of Two Hyper-planes and a Box in ( \mathbb{R}^n )</td>
<td>NELSON MACULAN, C. P. SANTIAGO, AND MARIA HELENA C. JARDIM</td>
<td>83</td>
</tr>
<tr>
<td>On Chromatic Scheduling Polytopes Coming from the Bandwidth Allocation Problem in Point-to-Multipoint Radio Access Systems</td>
<td>JAVIER MARIENCO AND ANNIGRET WAGLER</td>
<td>85</td>
</tr>
<tr>
<td>Subclasses of ( k )-trees: Characterization and Recognition</td>
<td>LILIAN MARKENZON, CLAUDIA M. JUSTEL, AND NEWTON PACIORKI</td>
<td>87</td>
</tr>
<tr>
<td>The Simple Assembly Line Balancing Problem</td>
<td>FERNANDO MARVAL AND JUAN JOSÉ SALAZAR</td>
<td>89</td>
</tr>
<tr>
<td>The Lights Out Game and von Neumann Cellular Automata</td>
<td>JOSÉ P. DE MELO AND NEI Y. SOMA</td>
<td>91</td>
</tr>
<tr>
<td>An Exact Algorithm for Graph Coloring</td>
<td>ISABEL MÉNDEZ AND PAULA ZABALA</td>
<td>93</td>
</tr>
<tr>
<td>Three-dimensional On-line Packing with Rotations</td>
<td>FLAVIO K. MIYAZAWA AND Y. WAKABAYASHI</td>
<td>97</td>
</tr>
<tr>
<td>A Set-Covering Based Heuristic Approach for Bin-Packing Problems</td>
<td>MICHELE MONACI AND PAOLO TOTH</td>
<td>101</td>
</tr>
<tr>
<td>Ant Colony Optimization for Mine Planning</td>
<td>MICHAEL MOOSSEN AND MARÍA-CRISTINA RIFF</td>
<td>103</td>
</tr>
</tbody>
</table>
A Polyhedral Approach to Some Nonconvex Optimization Problems
GEORGE NEMHAUSER AND ISMAEL DE FARIAS ......................................................... 109

Graph Layering by Promotion of Nodes
NIKOLA S. NIKOLOV AND ALEXANDRE TARASSOV ............................................. 113

Ordering Graphs and Vulnerability of Networks
CARLA S. OLIVEIRA, NAIR MARIA M. DE ABREU, AND ADEMIR F. PAVOTO ........... 115

An Algorithm for Traffic Grooming in WDM Optical Networks
ZENILTON K. G. PATROCÍNIO JR. AND GERALDO R. MEDEU ..................................... 117

Probabilistic and Optimization Versions of Propositional Satisfiability
DANIELE PRETOLENI .................................................................................................. 119

Eliminating Redundant Solutions of Combinatorial Integer Programs by Exploiting Symmetry
PABLO A. REY AND OSCAR PORTO ................................................................. 121

Components Cutting Optimization by Memetic Algorithms
ANA MARIA RODRIGUES, LUÍS MIGUEL MOREIRA, AND JOSÉ SOEIRO FERREIRA .......... 123

Bicriteria Robustness versus Cost Optimization in the Generation of Aircrew Pairings
DAVID M. RYAN AND MATTHIAS EHRGOTT .............................................................. 125

Simulated Annealing Approach to Solve the Daily Crew Scheduling Problem
GUSTAVO P. SILVA, MARCONE J. F. SOUZA, AND JOSÉ MARIA DO C. B. ALVES ........ 127

Double Contraction, Double Probing, Short Starts for Mixed (0,1) Integer Programming
KURT SPIELBERG AND MONIQUE GUIGNARD ...................................................... 131

A Penalty-Evaporation Heuristic in a Decomposition Method for the Maximum Clique Problem
PATRICK ST-LOUIS, JACQUES A. FERLAND, AND BERNARD CENDRON .................... 133

Combinatorial Problems in Spatial Forest Planning
ANDRES WEINTRAUB ............................................................................................... 135

Geometric Steiner Tree Problems
PAWEL WINTER AND MARTIN ZACHARIASEN ....................................................... 137

Approximation Schemes for a Class-Constrained Knapsack Problem
EDUARDO C. XAVIER AND FLÁVIO MIYAZAWA .............................................. 141
An Improved Branch and Bound Method to Solve a Pattern Sequencing Problem Based on Partial Orderings
HORACIO H. YANASSE, JOSÉ CARLOS BECCENERI, AND NEI Y. SOMA .................. 145

Author Index ................................................................. 147
Preface

The IV ALIO/ EURO Workshop on Applied Combinatorial Optimization is the fourth of a series of (now) triennial meetings, organized with the goal of encouraging the cooperation and the interchanges between Latin-American and European researchers and practitioners in the field of Combinatorial Optimization. The first workshop in this series was held in Rio de Janeiro (Brazil) in 1989, followed by those in Valparaiso (Chile) and Erice (Italy), respectively in 1996 and 1999.

This volume presents a collection of abstracts of the contributions presented at the IV ALIO/EURO Workshop on Applied Combinatorial Optimization, which was held in Pucon (Chile) from November 4 to 6, 2002. It was attended by approximately 70 participants from Argentina, Brazil, Canada, Chile, England, France, Ireland, Italy, New Zealand, Poland, Portugal, Spain, United States, and Venezuela.

This workshop was also organized on behalf of the celebration of the 60th birthday of both our friends and colleagues Nelson Maculan and Andrés Weintraub. It was intended to be a well deserved special tribute of their colleagues to these two well known researchers. Their contribution to Latin America goes far beyond their teaching and research activities, but also to the creation of the Asociación Latino-Íbero-Americana de Investigación Operativa (ALIO) and mainly to the development of Operations Research and Combinatorial Optimization and to the spread of their use across the region.

We wish to acknowledge the Universidad de Concepción for the support to the organization of this meeting. We are particularly thankful to its staff for their willingness to help and their administrative support. We are also indebted to the invited lecturers, to the members of the Program Committee, and in particular to Francesco Maffioli, for their encouragement and support.

Rio de Janeiro and Concepción, November 2002

Celso C. Ribeiro (General Chair)

Lorena Prudêncio (Local Chair)
Variable Neighborhood Search for the Optimization of the Scheduling of Terrestrial Production Drills

Dario J. Aloise¹, Daniel Aloise¹, Caroline T. M. Rocha¹, Jorge D. Melo², and Celso C. Ribeiro³

¹Universidade Federal do Rio Grande do Norte, Departamento de Informática e Matemática Aplicada, Natal, RN 59072-970, Brazil
²Universidade Federal do Rio Grande do Norte, DCA, Natal, RN 59072-970, Brazil
³Department of Computer Science, Catholic University of Rio de Janeiro, R. Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
E-mail: celso@inf.puc-rio.br

There are many petroleum wells in Brazilian oil fields that use artificial lift methods. Therefore, maintenance services such as cleaning, reinstatement, stimulation and others are essential to these wells. These services are done by Terrestrial Production Drills (TPD), which, due to their high operation costs, endure economical constraints and are consequently few compared with the number of wells demanding service. In this context, a problem concerning the optimization of the TPDs services is configured, which consists of finding the best itinerary for the available drills, in order to minimize the production waste caused by the useless wells waiting for service. This budget waste can be very high depending on each well production and on the number of demanded services. The decision of which drill should be sent to perform a service is based on factors such as the potential of well productivity, the geographic location of the drill in relation to the demanding well, and the type of service to be done. The problem is NP-hard and belongs to the class of routing and scheduling problems.

In this work, we first present the problem of optimizing the interventions of TPDs in oil wells. The latter is formulated as an integer programming problem. An implementation of the Variable Neighborhood Search (VNS) metaheuristic is proposed for its solution, together with local search and path-relinking mechanisms. Computational experiments on real and synthetical instances are developed to assess the comparison of the results obtained by this heuristic with those obtained by other approaches.

Bibliography


On the Location Problems Arising in UMTS Network Planning

EDOARDO AMALDI¹, ANTONIO CAPONE¹, AND FEDERICO MALUCCELLI¹

¹ DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy
E-mails: {amaldi, capone, malucell}@elet.polimi.it

In UMTS (Universal Mobile Telecommunication System) networks with Wideband-CDMA (Code Division Multiple Access) air interface base station (BS) location cannot only rely on signal predictions, but it must also consider the traffic distribution, the power control mechanism as well as the signal quality requirements. Therefore the general two-phase approach adopted for second generation cellular systems such as GSM where the planning problem is subdivided into a coverage problem (in which BSs are located so that the signal is high enough in most of the service area) and a frequency allocation problem (in which a number of channels has to be assigned to each active BS taking into account interference) is no longer appropriate.

Service quality is usually measured in terms of Signal-to-Interference Ratio (SIR). Since the signal quality constraint \( SIR \geq SIR_{\text{min}} \) for each active connection imposes a limit on the overall system capacity, the problem of locating BSs so as to minimize installation costs and/or maximize traffic coverage clearly falls within the general class of capacitated facility location problems. However, due to the presence of service (signal) quality constraints the resulting models differ substantially from the capacitated facility location problems which have been extensively studied in the literature.

A key issue in third generation mobile telecommunication systems is the Power Control (PC) mechanism, which is used to adjust the transmission powers so that signals are high enough to be received but as low as possible to limit interference. The PC mechanism of the CDMA air interface can be modeled at different levels of detail [2]. Here we focus on a power-based PC mechanism in which the emitted power is adjusted so as to guarantee a given target power \( P_{\text{tar}} \) at the receiving end. Although an ideal PC mechanism would be \( SIR \)-based, our power-based PC model is more accurate than the one in [3] and it provides a good trade-off in terms of solutions quality and computational requirements, see [2].

Assuming a power-based PC mechanism, we present and investigate a mathematical programming model for the uplink (mobile to base station) direction which is the most stringent one in the presence of full-duplex balanced connections such as voice calls. Suppose that a set of candidate sites (CS) \( S = \{1, \ldots, m\} \) where a BS can be installed, is given and that an installation cost \( c_j \) is associated with each CS \( j \), \( j \in S \). A set of test points (TPs) \( I = \{1, \ldots, n\} \) is also given. Each TP \( i \in I \) can be considered as a centroid where a given number of active connections \( u_i \) is requested and where a certain level of service (measured in terms of \( SIR \)) must be guaranteed. Let \( g_{ij}, 0 \leq g_{ij} \leq 1 \), be the propagation factor of the radio link between TP \( i \), \( 1 \leq i \leq n \), and CS \( j \), \( 1 \leq j \leq m \), which is estimated according to approximate models or determined according to ray tracing techniques. In the W-CDMA UMTS base station location problem one wishes, given a pair of sets \( S \) and \( I \), a propagation matrix \( G \) and a demand vector \( u \), to select a subset of CSs within \( S \) where to install BSs, and an assignment of the TPs in \( I \) to the available BSs taking into account the traffic demand and the service quality requirements in terms of \( SIR \). In general one aims at a trade-off between minimizing installation costs and maximizing traffic coverage. By considering the decision variables \( y_j \) for \( j \in S \) (\( y_j = 1 \) if a BS is installed in CS \( j \) and \( y_j = 0 \) otherwise) and \( x_{ij} \) for \( i \in I \) and \( j \in S \) (\( x_{ij} = 1 \) if TP \( i \) is assigned to CS \( j \) and \( x_{ij} = 0 \) otherwise), the core of the integer programming model can be formulated as follows:
min \sum_{j=1}^{m} c_{j} y_{j} - \lambda \sum_{i=1}^{n} \sum_{j=1}^{m} u_{i} x_{ij} \quad (1)

s.t. \sum_{j=1}^{m} x_{ij} \leq 1 \quad i \in I \quad (2)

x_{ij} \leq y_{j} \quad i \in I; j \in S \quad (3)

x_{ij}, y_{j} \in \{0, 1\} \quad i \in I; j \in S, \quad (4)

where \lambda is a nonnegative trade-off parameter. In addition, for each CS \( j \in S \), we have to add the above-mentioned signal quality constraint:

\[
\frac{P_{tar}}{\sum_{h=1}^{n} u_{h} g_{hj} \sum_{t=1}^{m} P_{tar} x_{ht} - P_{tar}} \geq SIR_{min} y_{j}. \quad (5)
\]

Due to the power-based PC mechanism, the numerator \( P_{tar} \) corresponds to the power received at CS \( j \) for any single connection from TP \( i \) to \( j \). Since \( P_{tar} / g_{ht} \) is the power that needs to be emitted from \( h \) to guarantee \( P_{tar} \) in \( t \), the denominator amounts to the sum of the signals arriving in \( j \) because of all the other active connections in the system, i.e., the total interference with respect to any single connection to \( j \). This type of service quality constraints, which clearly limit the system capacity, accounts for the substantial difference with respect to classical capacitated facility location problems. Although these bilinear constraints can be linearized, they make the problem harder to decompose.

In this talk we discuss the special structure of this new class of location models as compared with known capacitated ones and we show that variants of these NP-hard problems admit a polynomial-time approximation scheme. We also report and evaluate the solutions obtained with some classical metaheuristics (Grasp and Tabu search) for realistic problem instances.

Bibliography


Circular Arc Models and Algorithms for Packet Scheduling in Smart Antennas

EDOARDO AMALDI\textsuperscript{1}, ANTONIO CAPONE\textsuperscript{1}, AND FEDERICO MALUCELLI\textsuperscript{1}

\textsuperscript{1} DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy
E-mails: \{amaldi, capone, malucell\}@elet.polimi.it

The use of adaptive antennas arrays, known as “smart antennas”, has been recently considered for third generation mobile telecommunication systems (see e.g. [2, 3]). The main advantages of smart antennas is that they are expected to bring higher capacity and enhanced signal quality as compared to traditional antennas.

If the antennas elements of the array are close enough to each other (i.e., less than half the wavelength) it is possible to view a smart antenna as a set of co-located directive antennas whose orientations can be adapted (via software) according to the mobile stations positions. It is as if a smart antenna simultaneously transmits to (or receive from) narrow “beams” (approximately 12 degree wide angles) within the same cell. The interference between transmissions in different non intersecting beams can be considered as negligible. Since the spatial separation of mobile stations is exploited to simultaneously receive signals transmitted on the same radio channel, this access scheme is usually referred to as Space Division Multiple Access (SDMA) in the telecommunications literature. To allow multiple transmissions within the same antenna beam, SDMA is usually combined with other access schemes like Code Division Multiple Access (CDMA), which is typical of third generation mobile telecommunication systems. Although smart antennas improve performance of any transmission service, from a combinatorial optimization point of view it is particularly interesting to investigate their use in packet services where transmission scheduling techniques may have a strong impact on the overall system performance. See [4, 5] for a general analysis and [1] for an assessment of the impact in GPRS.

In this context a cell can be thought of as a circular area, and the mobile stations can be represented as points within this area. Due to the power control mechanism which adjusts the emission power so as to guarantee a received power equal to a given target value $P_{tar}$, all mobile stations can be assumed to lie within the same distance from the smart antenna and all points can be projected on the unit circle $C$ representing the cell circumference.

Let $I = \{1, \ldots, n\}$ be the index set of the mobile stations falling in the cell of a given smart antenna. For each $i \in I$, denote by $r_i$ the position of the point on $C$ representing mobile station $i$. Moreover, let $w_i$ denote the weight associated to each $i \in I$, which is proportional to the number of packets to be transmitted.

A feasible schedule of the smart antenna transmissions during a single time frame can be formalized by a subset $\mathcal{T} \subseteq I$ of points, a set $\mathcal{A}$ of identical size circular arcs (corresponding to 12 degree angles) positioned around the unit circle $C$, and an assignment of the points in $\mathcal{T}$ to the circular arcs in $\mathcal{A}$ satisfying the following requirements. Each point in $\mathcal{T}$ must be assigned to exactly one circular arc in $\mathcal{A}$ among those containing that point. Each circular arc in $\mathcal{A}$ must contain exactly one point in $\mathcal{T}$. Notice that pairs of circular arcs in $\mathcal{A}$ may have a nonempty intersection but each assigned point, i.e., each point in $\mathcal{T}$, must be contained in exactly one circular arc in $\mathcal{A}$. Let us call independent point-arc assignment such an assignment from a subset $\mathcal{T} \subseteq I$ of points to an appropriate set $\mathcal{A}$ of circular arcs around the unit circle $C$. It is worth emphasizing that the circular arcs are not part of the input but must be determined.
From the smart antenna packet scheduling point of view, two main types of combinatorial optimization problems are of interest:

- Given any set \( I \) with positions \( \{r_i\}_{i \in I} \) and weights \( \{w_i\}_{i \in I} \), find an independent point-arc assignment which maximizes the total weight of the assigned points. This clearly amounts to maximize the total number of packets transmitted during a single time frame.

- Given any set \( I \) with positions \( \{r_i\}_{i \in I} \) and weights \( \{w_i\}_{i \in I} \), find a partition of the points in \( I \) into independent point-arc assignments which minimizes a suitable objective function such as, for instance, the number of independent point-arc assignments or the sum of the maximum weight in each independent point-arc assignment. The first objective function aims at minimizing the number of time frames needed to transmit all the given packets.

If SDMA is combined with other multiple access schemes, the assignment of points to the circular arcs can be many to one instead of one to one. This gives rise to generalized versions of the proposed problems in which at most \( K \) points can be assigned to each circular arc in \( A \). Application-wise both off-line and on-line settings are of interest.

Some variants of the above scheduling problems are investigated. Combinatorial optimization models will be presented together with exact and heuristic algorithms based on circular arc properties and shifting techniques.

**Bibliography**


A Mixed Integer Disjunctive Formulation for the Transmission Network Expansion Problem

Laura Bahiense¹, Gerson Oliveira², Mario V. Pereira², and Sergio Granville²

¹ Power Systems Research Inc.
E-mail: laura@psr-inc.com

² Power Systems Research Inc.

The problem of determining the optimal set of candidate circuit additions for a power transmission network so as to supply the forecasted loads with minimum cost is usually formulated as a mixed non-linear program. The non-linearity is due to constraints related to the linearized power flow equations, where bus voltage magnitude variables are multiplied by circuit investment binary decision variables. The system generation is supposed capable of supplying the forecasted load, and candidate circuits are informed for all possible network branches, called rights-of-way. The linearized power flow model is composed of Kirchoff first and second laws, which are linear equations relating node (bus) angles, generations and loads to circuit flows. The linearized power flow equations are usually used in planning studies of high voltage meshed networks, providing good approximations for the circuit flows, and avoids the need to iteratively solve the non-linear power flow equations. Inequality constraints are simple upper bounds on generations and circuit flows.

The classical non-linear mixed integer formulation of the transmission network expansion problem cannot guarantee finding the optimal solution due to its non-convex nature. The non-convex nature of the continuous relaxation of this formulation can be avoided in two ways: relaxing (dropping) 2nd Kirchoff law for candidate circuits or by linearizing this equation. The latter allows an implicit linearization scheme that arises when applying Benders decomposition [1,2,6]. The computational effort of this decomposition scheme is high due to the need to solve a mixed integer linear program (the master) for each iteration. In general, many Benders iterations are required until convergence. During the Benders iterations, the incorporation of cuts result in increasing ill conditioning of the master problem, and therefore slows down the solution time of the master subproblems. Although no proof can be given with respect to optimality of the resulting solution, this approach was successfully applied only to medium scale problem instances.

The formulation known as the standard disjunctive formulation was independently proposed by [3,4]. In this formulation, the non-linear constraints of the non-linear formulation are avoided by using a disjunctive form to which they are equivalent. Benders decomposition was also applied to this approach [5], using a very large penalty factor. It has been proven that if a candidate circuit $k$ is such that there is an existing circuit on the same branch, the minimum value of the penalty parameter $M_k$ is given by $f_k^{\text{max}}$ times the ratio of the candidate susceptance and the existing circuit susceptance. Also, if the candidate circuit is in a new right-of-way, its penalty parameter is the product of its susceptance times the solution value of a shortest path problem on the network between the branches terminal nodes, where the distance between each pair of nodes is measured by the ratio of the branches flow capacity and its reactance. By using these optimal penalties, ill conditioning can be alleviated during the solution of the linear relaxation along the B&B tree. The standard disjunctive formulation suffers from bad conditioning due to the use of the large penalties $M_k$ in the disjunctive constraints.

In this work we present an alternative disjunctive formulation which uses optimal penalty factors and a tighter representation of power flows on candidate circuits. This new formulation has better conditioning properties than the standard one, resulting in significant reduction of computational
effort, as shown in two large scale benchmark case studies coming from the brazilian transmission network. With the alternative formulation we were able to prove optimality for the first time for the two problems.

The new formulation has more continuous variables, but being tighter, is better than the previous standard disjunctive formulation. Note that, contrarily to the Benders decomposition approach, which is an iterative scheme, the mixed integer disjunctive model is solved only once. Since it has the same number of binary variables as the non-linear formulation, and also the due to the tighter formulation, the B&B solution processing effort is much lower.

Another ingredient was necessary to accelerate the B&B tree search, being commonly used in combinatorial optimization methods. It is an upper bound (UB) to the solution value, and the better it is, more effective is the effect of pruning the tree, avoiding solving many LP relaxations during the search. A natural UB is the solution value of a heuristic solution. The computational effort of the heuristic must be much lower than the one required by the B&B solver Xpress-MP[11], but also its solution quality must be such that the optimality gap (the difference between the heuristic solution value and the optimal one) is low. These requirements are satisfied by the solutions obtained using the GRASP [9,10] metaheuristic. Very small gaps are obtained after a few GRASP [7,8] iterations, consuming small computing time.

References


Performance Evaluation of Reliable Multicast Algorithm Based on Recovery Tree

Marta Barría¹ and Reinaldo Vallettos²

¹Computer Science Department, Universidad de Valparaíso, Chile
E-mail: mart.a.barria@uv.cl

²Electronic Engineering Department, Universidad Técnica Federico Santa María, Chile
E-mail: reinaldo@elo.utfsm.cl

Multicasting provides an efficient way of disseminating data from a sender to a group of receivers. Instead of sending a separate copy of the data to each individual receiver, the sender just sends a single copy to all the receivers. Broadly speaking, multipoint applications can be divided into applications that require that the transmission of information be reliable, that is, that the transmitted packets are guaranteed to arrive without error at their destinations, and applications that do not need such reliability. Applications such as video conferencing, audio over Internet, multimedia events, electronic kiosks, and so on, do not require reliable transmission, as a certain level of information loss may occur without the user detecting a significant deterioration in the service received. On the other hand, applications such as software distribution, shared white boards, interactive games, transmission of bank accounts, replication of databases, and so on, require reliable transmission, meaning that all transmitted packets must be received without any errors by all receivers (members of the group) [1].

A number of protocols have been proposed in the past to provide a reliable multicast service. Briefly, they can be distinguished into sender-initiated, receiver-initiated and tree-based approaches.

Multicast transport protocols use positive or negative acknowledgment schemes to ensure reliable message delivery. A positive acknowledgment returned by a receiver confirms correct message delivery. In this error recovery method, it is the source (or the transmitter) that is responsible for maintaining transmission reliability, and thus is classified as sender-initiated. In the case of mechanisms based upon the use of NACKs, the transmitter retransmits a packet only when it receives a NACK for it from at least one receiver. Each receiver is responsible for detecting any loss of information and requesting the retransmission of the corresponding packet. Losses are generally detected because of the appearance of gaps in the numbering sequence of the received packets. To ensure the retransmission of a packet after having sent a NACK, the receiver activates a timeout, which is associated with the specific retransmission request. If the timeout expires before the lost packet is received, the receiver again sends a NACK requesting the lost packet. The above-described method is classified as receiver-initiated [2], for it is the receiver that initiates the lost information recovery cycle. Although the error recovery mechanism just described appears very simple, in practice some difficulties arise since, if each one of the receivers affected by the loss of a packet sends a NACK, this may produce the synchronized arrival at the source of many NACKs. This phenomenon is known as “NACK implosion”, which is undesirable because it produces congestion at or around the source, thus causing degradation in network global performance measurements such as delays and throughput [2]. In order to reduce implosion size, the recovery mechanism must allow the subset of receivers affected by the error to agree (in some way) on which of them will send the NACK. Unfortunately, achievement of this agreement necessarily takes time, which in turn increases the total recovery time (latency). In other words, the reduction of implosion tends to cause an increase in latency, which is also undesirable. Ideally, the performance of a lost packet recovery mechanism should be such that the size of the implosion is equal to 1 and latency is at a minimum. For this reason, the mechanisms for the recovery of errors that have been proposed in the literature attempt to simultaneously decrease implosion and latency so as to
improve the global performance measurements (delay and throughput) of the network. Nevertheless, since these objectives are contradictory, in practice, the proposed algorithms try to achieve a good compromise between them. It has been shown [4] that tree-based multicast protocols scale better than other multicast schemes suggested in the literature. Tree-based approaches organize the receivers into a tree structure called ACK tree, which is responsible for collecting acknowledgments and sending retransmissions. In [3] a new reliable multicast algorithm based on recovery tree is proposed. The error recovery process is carried out only by members which are in the neighborhood of the failure. These members delimit a so-called recovery tree. A fundamental aspect of the algorithm is that the nodes of the recovery tree associated with the failure are the only nodes responsible for recovering the lost packet. The idea behind this is to keep the number of nodes responsible for recovering the error to a minimum, in order to achieve an algorithm that is highly scalable and that performs very well. The algorithm attributes greater probability of requesting the retransmission of a lost packet to the leaves furthest from the root of the tree, since they are the ones that have a greater probability of being affected by the loss of a packet. The proposed algorithm achieves a low latency and low implosion in error recovery for any given size of the multipoint group. This is due to the fact that error recovery is realized in a local manner and using a Bernoulli experiment which determines whether or not a recovery tree leaf affected by the failure requests a lost packet. The performance of the algorithm is mathematically evaluated for a typical situation. The results show that the algorithm simultaneously achieves low latency and low implosion in the error recovery process, for any given size of the multicast group. This implies that the algorithm has a very high scalability.

In this paper, we extended the previous performance evaluation and we calculate another performance measures such as mean number of NACKs sent to recover a packet, and the mean number of retransmissions. Also, we obtain new performance measures, that our actual acknowledgement do not still have published in the specialized literature. They are the Probability Distribution Function (PDF) of the NACKs sent until to transmit with success a packet, and the PDF of the necessary time to recover a packet lost. All theses measures permit to characterize the dynamic behavior of the system.

Bibliography


Tabu Search Approach to Sequencing DNA Chains

JACEK BLAZEWICZ\textsuperscript{1} AND MARTA KASPRZAK\textsuperscript{2}

\textsuperscript{1} Institute of Computing Science, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland
E-mail: blazewic@put.poznan.pl

\textsuperscript{2} Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznań, Poland
E-mail: martarcs.put.poznan.pl

The paper deals with the problem of DNA sequencing by hybridization. The first phase - a biochemical one, uses (in a standard version) libraries of equal lengths oligonucleotides [1]. They are hybridized with the unknown DNA chain of length \( n \). A computational phase of this approach, i.e. a construction of the DNA chain on the basis of the oligonucleotides that hybridized with it, is NP-hard in the strong sense in case of errors [3]. (It is easy in case of no errors [4].) Thus, since the last problem does not admit a polynomial time algorithm, a need arises to construct efficient heuristic algorithms. Such an algorithm based on tabu search (and on the model presented in [2]) is proposed. Computational tests have proved its low complexity and high accuracy for both types of errors: false negatives and false positives.

References


Computational Complexity of Isothermal DNA Sequencing

JACEK BLAZEWICZ and MARTA KASPRZAK

2 Institute of Computing Science, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland
E-mail: blazewicz@put.poznan.pl

3 Fellowship holder of the Foundation for Polish Science, Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznań, Poland
E-mail: marta@cs.put.poznan.pl

The DNA sequencing problem, one of the most important problems from the computational molecular biology domain, consists in determining sequences of bases of unknown DNA fragments. Its input data come from a biochemical hybridization experiment, and they can be viewed as a set of words over the alphabet \{A, C, G, T\}. The aim is to reconstruct the original DNA sequence of a known length on the basis of these overlapping words, with no additional knowledge (e.g., about partial order of the words). The reconstruction is further complicated by the presence of experimental errors within the set. We distinguish two types of the errors: negative ones, i.e., missing words in the set (the words which are parts of the original sequence but are not present in the set), and positive ones, i.e., additional words in the set (the words which are not parts of the original sequence but are present in the set). We do not have information which words are missing or erroneous within the set.

The traditional approach to the DNA sequencing by hybridization uses sets of words of equal lengths. It was proved that in this case the DNA sequencing problem assuming a presence of errors in the input data, is strongly NP-hard [BK02], while the problem without errors is polynomially solvable [Pev80]. The isothermal version of the DNA sequencing uses sets of words of equal temperatures (melting temperature of oligonucleotide duplexes) but different lengths [BFK+00]. Such modification of the hybridization experiment allows to reduce the number of errors in the sequencing data, therefore the sequencing results usually are more similar to the original ones. The present work concerns the computational complexity of several variants of the isothermal DNA sequencing.

In the case of ideal hybridization experiment, where no error is present in the input data, the isothermal DNA sequencing problem is polynomially solvable. A polynomial-time algorithm has been proposed, in which an isothermal DNA graph is constructed on the basis of the set of words, and where, after a number of transformations, the Eulerian path is searched for.

On the other side, the variants of the isothermal DNA sequencing problem assuming one type of errors in the data, only negative ones or only positive ones, have been proved to be strongly NP-hard. The proofs contain polynomial transformations from the directed Hamiltonian path problem between two vertices. Of course, the general isothermal DNA sequencing problem without the assumption about errors in the data, is also strongly NP-hard.

Concluding, several variants of the isothermal DNA sequencing problem have the same computational complexity as their traditional versions (with words of equal lengths). However, the hybridization experiment with isothermal libraries could produce fewer errors in the data than the traditional one, and thus, the isothermal DNA sequencing has an advantage over its traditional counterpart.

1The research has been supported by KBN grant 7T11P03621.
Bibliography


Randomized Heuristics for Scene Recognition by Inexact Graph Matching

MARIA C. BOERES\textsuperscript{1}, CELSO C. RIBEIRO\textsuperscript{2}, AND ISABELLE BLOCH\textsuperscript{3}

\textsuperscript{1}Universidade Federal do Espírito Santo, Department of Computer Science, R. Fernando Ferrari, Campus de Giauibeiras, Vitória, ES 29060-970, Brazil
E-mail: boeres@inf.ufes.br

\textsuperscript{2}Department of Computer Science, Catholic University of Rio de Janeiro, R. Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
E-mail: celso@inf.puc-rio.br

\textsuperscript{3}École Nationale Supérieure des Télécommunications, CNRS URA 820, 46 rue Barrault, 75634 Paris Cedex 13, France
E-mail: Isabelle.Bloch@enst.fr

1 Introduction

The recognition and understanding of complex scenes require not only a detailed description of the objects involved, but also of the spatial relationships between them. Graph based representations are often used for scene representation in image processing [1, 2, 10, 11]. Vertices of the graphs usually represent the objects in the scenes, while their edges represent the relationships between the objects. In model-based recognition, both the model and the scene are represented by graphs. Graph matching is one of the important problems to be solved whenever such representations are used. However, the assumption of a bijection between the elements in two instantiations of the same scene is too strong for many problems. In this case, scene recognition may be better expressed as an inexact graph matching problem. Usually, the model has a schematic aspect. Moreover, the construction of the image graph often relies on segmentation techniques that may fail in accurately segmenting the image into meaningful entities. Therefore, no isomorphism can be expected between both graphs and such problems call for inexact (i.e., non-bijective) graph matching.

The motivation for this work comes from an application in medical imaging, in which the goal consists in recognizing brain structures from 3D magnetic resonance images, previously processed by a segmentation method. Similar problems occur in other applications, such as aerial or satellite image interpretation using a map, face recognition or character recognition.

2 Optimal Graph Matching Formulation

Two graphs are used to represent the problem: $G_1 = (N_1, E_1)$ represents the model, while $G_2 = (N_2, E_2)$ represents the oversegmented image. We assume that $|N_1| \leq |N_2|$, which is the case when the image is oversegmented with respect to the model.

Each node of $G_2$ is assigned to a label corresponding to one node of $G_1$. These assignments are represented by binary variables: $x_{ij} = 1$ if nodes $i \in N_1$ and $j \in N_2$ are associated, $x_{ij} = 0$ otherwise. Thus, each solution of the Graph Correspondence Problem (GCP) can be represented by a bipartite graph $G' = (N_1 \cup N_2, E')$, with $E' = \{(i,j) : i \in N_1, j \in N_2 \mid x_{ij} = 1\} \subseteq N_1 \times N_2.$
Each edge \((i, j) \in E'\) corresponds to the association of vertex \(i \in N_1\) with vertex \(j \in N_2\). The set \(A(i) = \{ j \in N_2 \mid (i, j) \in E' \}\) denotes the subset of vertices of \(N_2\) associated with the vertex \(i \in N_1\). To ensure that the structure of \(G_1\) appears within \(G_2\), we favor solutions where a correspondence between edges also implies a correspondence between their extremities.

Similarity matrices are constructed from similarity values calculated from graph attributes. Let \(S^v\) (resp. \(S^e\)) denote an \([N_1] \times [N_2]\) (resp. \([E_1] \times [E_2]\)) vertex-similarity (resp. edge-similarity) matrix, whose elements \(s^v(i, j)\) (resp. \(s^e((i, i'), (j, j'))\)) \(\in [0, 1]\) represent the similarity between vertices (resp. edges) \(i \in N_1\) and \(j \in N_2\) (resp. \((i, i') \in E_1\) and \((j, j') \in E_2\)). A good matching is a solution to GCP in which the associations correspond to high similarity values.

We define the objective function value \(f_1(G')\) of each solution \(x\) associated with a correspondence graph \(G' = (N_1 \cup N_2, E')\) as

\[
f_1(x) = \max_{x \in X} \frac{\alpha}{|N_1| \cdot |N_2|} \left[ \sum_{i \in N_1, j \in N_2} (1 - |x_{ij} - s^v(i, j)|) \right] + \frac{(1 - \alpha)}{|E_1| \cdot |E_2|} \left[ \sum_{(i, j) \in E_1} \sum_{(j', j) \in E_2} (1 - \max\{|x_{ij}x_{j'j'}, x_{ij'}x_{j'j'}\} - s^e((i, i'), (j, j'))) \right],
\]

where \(\alpha\) is a parameter used to weigh each term of \(f_1\) and, for every pair \(i \in N_1\) and \(j \in N_2\), \(x_{ij} = 1\) if \((i, j) \in E'\); \(x_{ij} = 0\), otherwise. This function consists of two terms which represent, respectively, the vertex and edge contributions to the measure of the solution quality associated with each correspondence. Vertex and edge associations with high similarity values are privileged by \(f_1\), while those with low similarity values are penalized.

To reduce the solution space, we are restrained only to solutions which satisfy the constraint below, which states that each vertex of \(N_2\) has to be associated with exactly one vertex of \(N_1\):

**Constraint (1).** There exists exactly one node \(i \in N_1\) such that \((i, j) \in E'\) for every \(j \in N_2\), i.e. \(|A^{-1}(j)| = 1\).

The search for the best solution is restrained to correspondences for which each set \(A(i)\) of vertices induces a connected subgraph in \(G_2\), for every model vertex \(i \in N_1\):

**Constraint (2).** The subgraph induced in \(G_2(N_2, E_2)\) by \(A(i)\) is connected, \(\forall i \in N_1\).

Pairs of vertices with null similarity cannot be associated. Such associations are discarded by the constraint below:

**Constraint (3).** \(\forall (i, j) \in E' \Rightarrow s^e(i, j) > 0\).

To ensure that all objects of the model appear in the image graph, one more constraint is imposed:

**Constraint (4).** There exists at least one node \(j \in N_2\) such that \((i, j) \in E'\) (or, alternatively, \(|A(i)| \geq 1\), \(\forall i \in N_1\)).

### 3 Randomized Heuristics

**Construction:** The randomized construction heuristic is based on progressively building a bipartite graph \(G' = (N_1 \cup N_2, E')\) corresponding to a feasible solution to GCP. Its edge set \(E'\) is initialized as the empty set and a new edge is randomly selected and inserted at each iteration, until a feasible...
solution is built. If the algorithm fails to find a feasible solution, a new attempt is performed. The algorithm returns the best among the first MaxSolutions solutions built after at most MaxTrials attempts. The overall complexity of each attempt to build a feasible solution is \(O(|N_1|^2 \cdot |N_2|^2)\).

**Local search:** The solutions generated by the randomized construction algorithm are not necessarily optimal, even with respect to simple neighborhoods. Hence, it is almost always beneficial to apply local search to attempt to improve each constructed solution. A first-improving local search algorithm works in an iterative fashion by successively replacing the current solution by the first improving solution found in a neighborhood of the current solution. We use two neighborhoods associated to each solution of GCP. The first one is formed by all feasible solutions that can be obtained by the modification of \(A^{-1}(j)\) for some \(j \in N_2\). The second neighborhood is formed by all feasible solutions that can be obtained by the exchange of the nodes \(A^{-1}(j)\) and \(A^{-1}(k)\) associated with two nodes \(j, k \in N_2\). The second neighborhood is used whenever a local optimum with respect to the first neighborhood is found. The local search algorithm terminates when a local optimum with respect to both neighborhoods is found.

**GRASP:** The GRASP (Greedy Randomized Adaptive Search Procedure) metaheuristic [3, 5, 6, 7, 8, 9] is a multi-start or iterative process, in which each iteration consists of two phases: construction and local search. The construction phase builds a feasible solution, whose neighborhood is investigated until a local optimum is found during the local search phase. The best overall solution is kept as the result. We built a GRASP heuristic for the Graph Correspondence Problem using the construction and local search algorithms above outlined.

| Instance | \(|N_1|\) | \(|E_1|\) | \(|N_2|\) | \(|E_2|\) | \(f_1^*\) | Construction | LS | GRASP |
|----------|-----------|-----------|-----------|-----------|---------|-------------|-----|--------|
| GM-0     | 3         | 2         | 7         | 10        | 0.9045  | 0.9045      | 0.9045 | 0.9045 |
| GM-1     | 4         | 5         | 9         | 15        | 0.7698  | 0.7535      | 0.7698 | 0.7698 |
| GM-2     | 5         | 8         | 12        | 23        | 0.7465  | 0.7327      | 0.7462 | 0.7465 |
| GM-3     | 6         | 9         | 13        | 25        | 0.7147  | 0.6835      | 0.6937 | 0.7147 |
| GM-4     | 7         | 11        | 15        | 29        | 0.7761  | 0.7235      | 0.7577 | 0.7761 |
| GM-5     | 10        | 15        | 30        | 39        | -       | 0.5168      | 0.5474 | 0.5543 |
| GM-5a    | 10        | 15        | 30        | 41        | -       | 0.4981      | 0.5329 | 0.5532 |
| GM-6     | 12        | 42        | 95        | 1434      | -       | 0.4122      | 0.4251 | 0.4254 |
| GM-7     | 14        | 27        | 28        | 63        | -       | 0.6182      | 0.6338 | 0.6367 |
| GM-8     | 30        | 39        | 100       | 297       | -       | 0.4978      | 0.5183 | 0.5210 |
| GM-8a    | 30        | 42        | 100       | 297       | -       | 0.5014      | 0.5218 | 0.5245 |
| GM-9     | 50        | 88        | 250       | 1681      | -       | 0.5049      | 0.5184 | 0.5193 |

Table 1: Characteristics and results obtained for instances GM-0 to GM-9 for \(\alpha = 0.9\).

### 4 Computational Results

The algorithms described in the previous section were implemented in C. We have used a set of twelve test instances in the evaluation of the algorithms proposed in this work. The characteristics of these instances and the solution values found by the different algorithms are summarized in Table 1. For each instance, we first give the number of nodes and edges of the model and image graphs, together with the maximum value \(f_1^*\) for \(\alpha = 0.9\), whenever the latter is known (obtained by complete enumeration of all feasible solutions). Next, we consider \(\alpha = 0.9\) and we report the values of the solutions found by the construction algorithm (with the parameter values MaxTrials = 500 and MaxSolutions = 1), by the first-improving local search algorithm, and by the GRASP heuristic (after 200 iterations).
Instance GM-7 was created for the computational experiments reported in [4] from the 2D images in Figure 1. The image (Figure 1-(a)) was oversegmented in 28 regions (Figure 1-(c)) and compared with a model with only 14 well defined regions (Figure 1-(b)). The model graph $G_1$ has 14 vertices and 27 edges, while the oversegmented image graph $G_2$ has 28 vertices and 63 arestas. Grey levels were used in the computation of node similarities, while distances and adjacencies were used for the computation of edge similarities. We notice that the GRASP heuristic was able to correctly identify 26 out of the 28 regions of the oversegmented image.

![Images](a) (b) (c)

Figure 1: Cut of a muscle (instance GM-7): (a) original 2D image, (b) model, and (c) oversegmented image.

**Bibliography**


Pucon, Chile, November 4-6, 2002
Worst-case Performance of Wong’s Steiner Tree Heuristic

HUGO BRAVO-AZLÁN$^1$ AND ALFREDO CANDIA-VÉJAR$^2$

$^1$University of Tarapacá, Department of Mathematics, 18 de Septiembre 2222, Arica, Chile
E-mail: hbravo@uta.cl

$^2$University of Talca, Department of Systems Engineering, Merced 437, Curicó, Chile
E-mail: acandia@uta.cl

The Steiner problem in directed graphs (SPDG), a classic combinatorial optimization problem, is to find a minimum-cost subgraph of a given directed graph such that a directed path from a root node to every node of a set of specified node exists. More formally, let $D = (N, A)$ be a directed graph with a node set $N$ consisting of root $r$ and two disjoint sets $Z$ of demand nodes and $S$ of possible Steiner nodes and a set $A$ of arcs with nonnegative weights. SPDG is to find a minimum cost subgraph of $D$ including a directed path from $r$ to every demand nodes. In order to achieve this minimum subgraph additional nodes from the set $S$ may be used. An optimal solution which is a directed tree or $r$-arborescence is guaranteed since all arc weights are nonnegative.

It is known (Karp [1]) SPDG is \textit{NP}-hard optimization problem so many heuristics have been designed for its approximate solution.

An important group of heuristics use some combination of shortest paths and minimum spanning arborescence subproblems. Voss [2] analyzed the worst-case performance for some of these heuristics which are extensions of heuristics designed for the undirected case of SPDG. He found that the best bound for the worst-case error ratio is $O(|Z|)$. From this disencouraging result, Voss [2] suggested to study the performance of Wong’s algorithm [4]. Recently, Williamson [3] in his study of the primal-dual method applied for approximation algorithms noted the absence of performance guarantees for several dual-ascent methods including Wong’s algorithm.

Wong’s heuristic for SPDG consists in a dual ascent procedure for obtaining lower bounds to the optimal solution and the information gained in this phase is then used in a heuristic procedure for obtaining feasible solutions.

In this work, we design a particular class of directed graphs for which the worst-case performance is asymptotically 2. Therefore, we have found a lower bound for this performance.

Bibliography


A Specialized Version of Benders Decomposition Applied to a General Machine Sequencing Model

RICARDO S. DE CAMARGO$^{1}$ AND CARLOS ROBERTO V. DE CARVALHO$^{1}$

$^{1}$Universidade Federal de Minas Gerais, Departamento de Engenharia de Produção, Av. Antonio Carlos 6627, Ed. PCA, Campus UFMG, Caixa Postal 209, Belo Horizonte, MG 30161-970, Brazil
E-mails: {rcamargo, carlos}@deap.ufmg.br

In this paper, a specialized Benders Decomposition Algorithm is proposed to tackle a general machine sequencing model, formulated as the problem of finding a mini-maximal path in a disjunctive graph.

In a general machine sequencing problem ($GMSP$), a set $J$ of jobs is to be processed on a set $M$ of machines. The processing of a job in a machine is called an operation, and its duration is a given constant. (i) The order in which the various operations concerning one job are to be carried out is fixed by the technological process, while the order in which the various jobs are to be processed by a certain machine is not fixed. (ii) Each machine of $M$ can process only one job of $J$ at a time. The problem is then to find an optimal sequence of carrying out the various operations on the various machines that the time needed to complete all the jobs should be minimal.

The $GMSP$ can also be represented on a disjunctive potential-task graph, $G(N, A, E)$, with the introduction of two dummy operations, 0 and $n$, representing the “start” and “finish” operations of the problem, respectively. Where $N = 0, 1, 2, ..., n$ is the set of operations, or the node set; $A$ is the set of pairs of operations representing the precedence relations fixed by the technological process (condition (i)), or the directed arcs set; and $E$ is the set of subsets $E_k$, where $E_k$ is the set of pairs of operations to be processed on machine $k$. These pairs of operations cannot overlap in time. The set $E$ is also known as the undirected, but orientable edge set. Each edge of $E$ can be viewed as a pair of oppositely directed arcs, called disjunctive arcs. The length of an arc $(i, j) \in A$ or of an edge $(i, j) \in E$ is given by its orientation, $p_i$ if $i < j$ (operation $i$ precedes $j$) or $p_j$ if $i > j$ (operation $i$ succeeds $j$), where $p_i > 0$ and $p_j > 0$ are the processing times of operations $i$ and $j$, respectively.

Mann [10] stated this problem as a linear programming problem with integer numbers, using an integer variable, $y_{ij}$, to represent each disjunctive arc $E$, or each pair of operations that are to be processed by a certain machine:

\[
\begin{align*}
\text{(LP)} \quad \text{minimize} \quad & t_n, \\
\text{subject to :} \quad & t_j - t_i \geq p_i \quad \forall (i, j) \in A \\
& t_j - t_i \geq p_i - M(1 - y_{ij}) \quad \forall (i, j) \in E_k, \ k \in M \\
& y_{ij} + y_{ji} = 1 \quad \forall \{(i, j); (j, i)\} \in E_k, \ k \in M \\
& t_i \geq 0 \quad \forall i \in N
\end{align*}
\]

where $y_{ij} = 1$, if operation $i \prec j, \forall (i, j) \in E_k$, and $y_{ij} = 0$, if $i \succ j, \forall (i, j) \in E_k$; $t_i$ is the variable representing the starting time of operation $i$; and $M$ is a finite positive number sufficiently large to hold any value the variables $t_i$ and $t_j$ might take, an upper bound.

Any feasible solution to (LP) is called a schedule. A schedule is also defined as a directed cyclic
graph $D(N, A)$ obtained from $G$ by selecting exactly one member of each disjunctive arc pair of every $E_k$ and that does not contain a directed cycle. Furthermore, we denote $L(0, n)$ and $\mu(0, n)$ as the length and the path of the longest path from operation from 0 to $n$, respectively. So, the time needed to complete all jobs, $t_n$, is given by $t_n = L(0, n)$ on a graph $D$. Thus, the $GMS P$ can be that of finding an acyclic directed graph from $G$ that minimizes the length of the longest path.

For literature on $GMS P$, see Conway et al. [6], Baker [1], Lenstra [9], Rinnoy Kan [11], French [7], Blazewicz et al. [5].

Benders Decomposition ([4] and [8]) is a partitioning procedure for solving mixed integer programming problems. This procedure, applied to the $GMS P$, requires the alternate solution of two problems, one of which is a linear program, called Sub-Problem (SP); while the second one is a linear program with $p$ zero-one variables and one arbitrary continuous variable, called Master Problem (MP), where $p$ is the total number of edges in $E$. The SP is the problem that of finding the starting times of the operations, while the MP is the problem that of finding the orientation of the disjunctive arcs.

In the procedure of alternately solving SP and MP, each time we solve MP, either the procedure comes to an end, or we obtain a new directed acyclic graph $D$, resulting in a new SP; and each time we solve SP, either the procedure ends, or we obtain a new constraint that will be added to the MP. The procedure stops when the optimal solution of the SP is equal to the optimal solution of the MP.

Balas [2] was the first one to propose a specialized version of Benders Decomposition applied to a $GMS P$, but since then some questions were raised: Does the value of $M$ have some influence on the Benders Decomposition procedure when applied to a $GMS P$? What does “sufficiently large” mean? Is there a better value? Is it possible to improve the procedure by adding more constraints at a each iteration? Addressing these questions, we propose a specialized version of Benders Decomposition where we adopt an enumerative algorithm proposed by Balas [3] to enrich the MP at each iteration. Although limited computational experience is found, the results are encouraging.

Bibliography


Models and Algorithms for Staff Scheduling Problems

ALBERTO CAPRARÀ1, MICHELE MONACI1, AND PAOLO TOTH1

1 Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
E-mails: \{acaprara,monachi,ptoth\}@eis.unibo.it

Staff scheduling problems are frequently encountered in the Operations Research literature. Typically, these problems require the definition of the workload for employees working in companies (e.g., hospitals, fire departments, production plants, call centers, transportation companies, etc.) which are typically active from 16 to 24 hours every day. The workload has to be defined for a period of one or more weeks, so as to take care of the weekly rests for the employees. The problem is often approached by first defining the short-term assignments (typically daily assignments) for the employees (Phase 1) and then composing these short-term assignments into long-term assignments (typically monthly or weekly assignments) (Phase 2).

For the relevant case of traveling personnel for transportation companies, there are many possible ways of defining the short-term assignments and the rules that make such assignments feasible are quite complicated. This makes Phase 1 often the most important within these applications, also considering that, once the short-term assignments have been determined, Phase 2 is typically easy—in particular this applies to urban mass transit companies, for which almost all of the work has to be done during the day, whereas it does not apply to large airline and railway companies (see e.g., [3, 2, 4]).

In many other cases, there is a small list of possible short-term assignments, e.g., either from 6 AM to 2 PM, from 2 PM to 10 PM or from 10 PM to 6 AM. In these cases, Phase 1 is essentially already solved and the relevant part is Phase 2. In our case, all the short-term assignments we consider have a duration of less than one day (typically 8 hours) and are called daily assignments.

Several references on the existing literature for staff scheduling can be found in [1]. In most cases [1, 5], once Phase 1 has been solved, the number of employees being \( n \), the long-term assignment spans \( m \) weeks. The workload of each employee repeats cyclically every \( m \) weeks and, in each week, employee \( i + 1 \) has the same workload as employee \( i \) in the previous week. In this paper, we consider a different real-world problem, in which the workload of the employees may be different during the period, although the number of working days and the durations of the rest periods are the same for each employee.

The problem that we consider arises within an Italian call center. At present, the number and size of call centers in Europe is rapidly increasing, and call centers have become an important source of staff scheduling problems for Operations Research practitioners. The call center we consider belongs to an Italian company providing SOS electronic devices, and receives emergency calls from a pushbutton SOS telephone system, containing an automatic dialer. The problem is to schedule the employees that answer to emergency calls 24 hours a day.

In particular, in our application, we consider a period of \( n \) days, that are denoted by \( N := \{1, \ldots, n\} \). The daily assignments to be performed each day, i.e., the output of Phase 1 mentioned above, are actually already fixed by the company and are an input of our problem. In particular, there are \( n_{da} \) different daily assignments \( \{1, \ldots, n_{da}\} \) (generally having the same duration). For \( q = 1, \ldots, n_{da} \), daily assignment \( q \) requires \( e_{qj} \) employees for each day \( j \) of the period. We let \( req_{j} \)
denote the overall number of employees that must work on day \( j \), i.e., \( \text{req}_j := \sum_{q=1}^{n_d} c_{q,j} \).

Our aim is to solve Phase 2 mentioned before, i.e., to arrange the daily assignments into an \( n \)-day assignment, with the following constraints.

- Within the period, each employee works for blocks of consecutive days and has a rest after each block. Within each block, the employee performs the same daily assignment. If the daily assignment of each day of the block is \( q \), we say that the block is of color \( q \).
- There are \( n_t \) block types, the \( t \)-th having a duration (in days) equal to \( k_t \). Within the period, each employee must perform \( b_t \) blocks of type \( t \) for \( t = 1, \ldots, n_t \).
- There are \( n_r \) rest types, the \( p \)-th having a duration (in days) equal to \( d_p \). Within the period, each employee must have \( a_p \) rests of type \( p \) for \( p = 1, \ldots, n_r \).
- There is a list of infeasible sequences of the form \( \{(t_1, q_1), (t_2, q_2)\} \), with \( t_1, t_2 \in \{1, \ldots, n_t\}, q_1, q_2 \in \{1, \ldots, n_d\} \) and \( p \in \{1, \ldots, n_r\} \), meaning that it is not possible to have a block of type \( t_1 \) and color \( q_1 \) followed by a block of type \( t_2 \) and color \( q_2 \), with a rest of type \( p \) in between.
- Each employee works for at most \( s \) days among those in a given set \( S \) of special days, generally the Sundays and the other holidays within the period. (The week day corresponding to the first day of the period is specified on input.)

In addition, for each employee, a block or a rest can be split between the first and the last days of the period (e.g., a block of duration 3 can span days 1, \( n - 1 \), \( n \)). As some of the employees can be already working before the beginning of the period, for these employees initial conditions specify a block of type \( t \) to be performed by the employee for up to the first \( k_t - 1 \) days, along with the associated color, or the presence of a rest of type \( p \) that terminates within the first \( d_p - 1 \) days – in this case, the initial conditions specify also the type and color of the last block performed by the employee.

The natural objective for the problem is to minimize the global number of employees.

Our approach finds the solution of the problem in two steps. In the first one, we determine the minimum number of employees and the associated pattern for each employee, i.e., the days in which the employee works (without specifying the block colors) and the days in which he/she has a rest, so as to guarantee that at least \( \text{req}_j \) employees are working on each day \( j \in N \). This step is approached by formulating the problem as an Integer Linear Program (ILP) and finding an optimal solution by branch-and-bound. In the second step, we associate a color with each working block so as to ensure feasibility while balancing the workload among the employees. We solve heuristically this phase as a sequence of Transportation Problems, one for each day of the period.

Decomposition is motivated by the fact that the difficult part is the first one, whereas assigning block colors to the employees once the corresponding patterns have been fixed is relatively easy. On the other hand, an ILP formulation for the whole problem would be considerably larger than the one we use in the first step, and impractical to handle.

In order to define an ILP model for the first step, we let \( \mathcal{P} \) be the set of all feasible patterns for an employee. In particular, each pattern can be represented by a 0-1 vector with \( n \) entries, the \( j \)-th equal to 1 if the employee is working on day \( j \) and 0 otherwise. By defining

\[
g_P = \text{number of employees who are performing pattern } P \quad (P \in \mathcal{P}),
\]
we obtain the following ILP formulation:

\[
\begin{align*}
\min & \sum_{P \in \mathcal{P}} y_P \\
\sum_{P \in \mathcal{P}_j} y_P & \geq req_j \ (j \in N) \\
y_P & \geq 0, \text{ integer } \ (P \in \mathcal{P}),
\end{align*}
\]

where \( \mathcal{P}_j \subseteq \mathcal{P} \) denotes the set of feasible patterns that require working on day \( j \).

The explicit generation of all patterns in \( \mathcal{P} \) and the inclusion of all the corresponding variables into the model is possible only for small instances. Instead, when facing with real-world applications, we have to use a column generation technique in order to handle model (1)-(3). In particular, we develop three different methods to handle the model above when the number of variables is too large to have all of them explicitly in the model.

The first method is the explicit generation of all variables and the use of a pricing technique that computes the reduced costs for all variables not in the current model and adds (a subset of) those with negative reduced cost. This method can be applied in practice as long as the overall number of variables does not exceed, say, a few millions.

The second method generates columns with negative reduced costs (if any) by using a different ILP model in which the objective function is the maximization of the “dual” profit associated with the working days in a pattern and constraints impose the resulting pattern to be feasible.

The third method is based on dynamic programming. To this aim, we define a table giving the maximum profit that can be achieved in days from 1 to \( j \) by a pattern that, within these days, contains a certain number of blocks and rests of each type and works for a certain number of special days. Hence the maximum profit of a pattern can be detected by looking at the entries of the table, while the associated pattern can be reconstructed by storing the predecessor of each entry of the table in a standard way.

In order to achieve the optimal integer solution of the first step by using model (1)-(3) with the column generation methods, we implemented our own branch-and-bound method. As is often the case, branching changes the structure of the column generation problem. More specifically, all columns with negative reduced cost found by the dynamic programming technique may correspond to variables already fixed by branching. If this is the case, we resort to the ILP model based generation technique, adding suitable cardinality constraints to prevent the selection of these columns.

Given the patterns found in the previous step, in the second step we define the color of the blocks assigned to each employee. Here, the main objective is to find a feasible solution, called feasible color assignment. Note that there may be no feasible color assignment associated with a given feasible set of patterns, even if a feasible solution of the overall problem with the same number of employees exists. Moreover, our approach to this second step is heuristic in nature, in that it does not guarantee finding a feasible color assignment for any value of the parameters even if such an assignment exists. Nevertheless, it can be shown that in some relevant special cases, including our specific application, the proposed method surely finds a feasible assignment.

In our heuristic, we solve a sequence of Transportation Problems, one for each day of the given period. We consider days \( 1, \ldots, n \) in this order. For each day \( j \), the daily assignment for some of the employees working on day \( j \) (i.e., starting a block of type \( t \) on days \( j, j - 1, \ldots, j - k + 1 \)) is specified either by the initial conditions or by the choices made for days \( 1, \ldots, j - 1 \). In the solution of the problem, we introduce an objective function aimed at balancing, for each employee, the number of blocks of each color as well as the number of days between each block color. In particular, the cost
of assigning a color to an employee if set to $+\infty$ if the employee cannot perform the daily assignment (because of the infeasible sequence); otherwise, this cost is set to a suitable value, favoring blocks of a color that was not assigned to the employee for longer time, balancing the number of blocks of each color assigned to the employees, and penalizing "undesired" sequences of consecutive block colors.

Computational experiments show the results of our branch-and-bound algorithm for finding the optimal patterns (step 1). Indeed, for all the instances considered, the heuristic algorithm used to find a feasible color assignment always succeeded, and ran in a time much smaller than the one needed to find the patterns. We used real-world test instances from our application with $n$ ranging from 42 to 168 days (i.e., from 6 to 24 weeks), and ran our branch-and-bound algorithm with a time limit of 4000 CPU seconds on a Digital Alpha 533 MHz for each instance.

For what concerns the solution of the LP relaxation of model (1)-(3), the column generation methods outperform the solution of the entire model, as may be expected. Moreover, generating columns by means of dynamic programming seems to be better than both the pricing technique (even for small instances) and the ILP model based generation technique. As to the branch-and-bound algorithm mentioned above, it was able to solve all instances to optimality within the time limit.

In order to test the flexibility of our branch-and-bound algorithm, we considered two additional classes of test instances obtained from the previous one by changing the block and rest durations, and the daily workload, respectively. As for the first class of instances, the branch-and-bound algorithm was able to determine, within the time limit, the optimal solution of almost all the instances. The same happened for instances with a workload randomly generated for each day, which turned out to be easier than those having a regular workload.

**Bibliography**


Greedy Algorithms for the 3-dimensional Assignment Problem

MANUEL V. CENTENO\textsuperscript{1} AND JUAN JOSÉ SALAZAR\textsuperscript{2}

\textsuperscript{1} Universidad de Oriente, Núcleo de Sucre, Dpto. Matemáticas, Cumaná Edo, Sucre, Venezuela
E-mail: mceneto@sucre.udo.edu.ve

\textsuperscript{2} Universidad de La Laguna, Facultad de Matemáticas, Departamento de Estadística, Investigación Operativa y Computación, La Laguna, Tenerife, Spain
E-mail: jjaisalaza@ull.es

The combinatorial optimization problem which is the subject of this work is the multidimensional assignment problem (mAP). The mAP, with $m = 3$, this is, the 3AP, could be defined as: given three sets of $n$ elements and a weight associated with each selection containing one element from each set, the 3-dimensional assignment problem is that of finding $n$ disjoint selections with minimum weight. The problem is an immediate generalization of the well-known 2-dimensional assignment problem. There are several versions in the literature and that here presented is NP-hard and corresponds to the so-called “axial 3-AP”, see e.g. the references in [1]. Our contribution is the development new metaheuristic algorithms (greedy) for finding good solutions to randomly generated instances with up to $n = 50$, that can be used as good initial solutions in other metaheuristic algorithms. In particular, we have implemented five greedy algorithms and showed the results from our computational experiments.

Bibliography

Application of Descents with Mutations to the Aggregation of Relations

Irene Charon\textsuperscript{1} and Olivier Hudry\textsuperscript{1}

\textsuperscript{1}École Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris cedex 13, France
E-mails: \{Irene.Charon, Olivier.Hudry\}@ens.t.ad

We study here the application of a new metaheuristic issued from the noising methods that we call descents with mutations (DWM) to two problems arising in the field of the aggregation of relations:

\begin{itemize}
  \item The clique partitioning of a graph: given a complete graph $G = (X, E)$ of which the edges are weighted by negative or positive integers, partition $X$ into $k(G)$ subsets $X_1, X_2, \ldots, X_k(G)$ in order to minimize the sum of the weights of the edges with both extremities in a same subset. This problem arises for example when a set of equivalence relations must be aggregated into a unique equivalence relation.
  \item The linear ordering problem: given a tournament $T = (X, A)$ of which the arcs are weighted by non-negative integers, find a linear order $O = (X, B)$ in such a way that the sum of the weights of the arcs which belong to $A$ but not to $B$ is minimum. This problem occurs in different contexts; for instance, in voting theory, when a set of linear orders defined on a set $X$ of candidates (representing the individual preferences of voters) should be summarized by one linear order (representing the collective preference of the voters).
\end{itemize}

DWM constitutes a local search metaheuristic. As the other metaheuristics, DWM is not designed to be applicable to only one combinatorial problem, but to many of them. DWM is based on elementary transformations. In a usual descent, the process is not blind in the sense that the elementary transformations are adopted only if they involve an improvement of the value taken by the function $f$ that we want to optimize. In DWM, we also apply the basic process of a descent but, from time to time, instead of applying the acceptance criterion, we apply and accept the considered elementary transformation, whatever its effect on $f$: we say that we have a blind elementary transformation, or simply a mutation, since it is the word commonly used in genetics (and in genetic algorithms) to denote such an operation. Thus, the design of DWM is quite simple: the only thing to precise in order to apply DWM (in addition to what must be defined in order to apply a descent, like the elementary transformation) is when a mutation is adopted. In this communication, we explain how this can be done for the problems depicted above. In order to study the efficiency of DWM, we compare it with another very efficient metaheuristic, which is a classic simulated annealing improved by the addition of some ingredients coming from the noising methods. These experiments show that the descents with mutations are at least as efficient for the studied problems as this improved simulated annealing, sometimes a little better, while it is much easier to apply.
Approximate Combinatorial Optimization Models for Large-Scale Production Lot Sizing and Scheduling with Sequence-Dependent Setup Times

Alistair R. Clark

Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, BS16 1QY, England
E-mail: alistair.clark@uwe.ac.uk

Several Enterprise Resource Planning (ERP) software vendors now include optimization facilities that make use of powerful Operational Research approaches, such as mixed integer programming (MIP), to help improve the quality of operations planning and scheduling. The onus is, however, still on the user to formulate an optimizing model that suitably reflects the organization’s planning objectives and constraints. The formulation of an appropriate model is by no means a simple task and if carried out naively can result in models that, if too large, take an impossible amount of computing time to optimize and will need to be solved via a heuristic method [6].

A related but alternative approach to heuristic optimization is to use approximate models that are simpler to solve, but still reflect the objectives and constraints. As a case in point, this paper develops three MIP models and associated solution methods to assist in identifying a capacity feasible master production schedule in MRP systems. The initial exact model represents setup times that are sequence-dependent and permits multiple setups within a planning period. This makes for a combined lot sequencing and sizing problem, recognised as a difficult topic [9, 2, 7, 8]. The resulting huge number of binary variables causes great computational intractability for non-trivial problems.

Such complexity is partially overcome in this paper by substituting the vast majority of the binary variables and constraints with continuous ones. In a previous paper [1] which formulated an exact MIP that allowed multiple setups per planning period, fast approximate models and solution methods were developed and found to produce solutions of reasonable quality. The current paper extends this approach to multi-level systems with a revised method of estimating key parameters and refined additional approximate modeling.

A first approximate model and solution method, to be used with larger product structures, suboptimally schedules setups and lots on a period-by-period basis, estimating the capacity usage of future setups through the use of linear rather than integer variables. A second model and method, developed from the first, greatly accelerates computing time by sequencing setups gradually within each period, but again suboptimally. The trade-offs between schedule quality and computing time are analyzed in computational tests. The second model is able to schedule setups of up to 100 products on 10 machines over 5 periods in viable computing time. It can be used operationally to make complex allocation and sequencing decisions of reasonable quality, particularly if applied on a rolling horizon basis to quickly schedule only the most immediate setups, a common practice given the prevalent reality of imperfect demand forecasts.

A further advantage of the above approach is that no special algorithms need to be developed. Rather, the accelerated solutions can be achieved by using powerful modelling approximations whose parameters are calculated from the specific demand and production data. As a result, the models are easily implemented with high-level mathematical programming tools [3, 4], making use of the good default search strategies within industrial strength MIP solvers [5].
Bibliography


Optimal Procurement Decisions in the Presence of Total Quantity Discounts and Alternative Product Recipes

Yves Crama\textsuperscript{1}, Rodrigo Pascual\textsuperscript{2}, and Andrea Torres\textsuperscript{3}

\textsuperscript{1}University of Liège, Boulevard du Rectorat 7 (B31), 4000 Liège, Belgium
E-mail: y.crama@ulg.ac.be

\textsuperscript{2}Department of Mechanical Engineering, Universidad de Chile, Beaucheff 850, Santiago, Chile
E-mail: rpascual@cec.uchile.cl

\textsuperscript{3}Ferrocarrières del Pacífico, Alfredo Barros Errazuriz 1960, Providencia, Santiago, Chile
E-mail: andrea.torres@entelchile.net

We describe the cost-minimization problem faced by the purchasing department of a multi-plant company when its suppliers offer discounts based simultaneously on plant and on corporate purchases, when discount schedules depend on the total quantity (rather than cost) of ingredients purchased, and when alternative production recipes exist for each final product. We formulate the problem as a nonlinear mixed 0-1 programming problem and we propose various ways to linearize this formulation. The quality of these models is evaluated on real-world data.
Integrating Tabu Search and Constraint Programming

MAURO DELL’AMICO\(^1\) AND ANDREA LODI\(^2\)

\(^1\)DISMI, University of Modena and Reggio Emilia, Viale A. Allegri 13, 42100 Reggio Emilia, Italy
E-mail: dellamico@unimore.it

\(^2\)DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
E-mail: alodi@deis.unibo.it

Introduction

In a recent paper, Focacci, Laburthe and Lodi [4] surveyed the integration between Local Search and Constraint Programming which seems to be suitable to address real-world combinatorial optimization problems. In this paper, we focus on the integration of the machinery developed in the Tabu Search context into incomplete global search algorithms based on CP.

Local Search (LS) methods are based on the simple and general idea of: (i) starting from a feasible solution, say \( s \), (ii) defining a neighborhood of such a solution, say \( \mathcal{N}(s) \), as the set of solutions which can be reached from \( s \) through a move (a simple manipulation of \( s \) whose effect is the transition to another solution \( s' \in \mathcal{N}(s) \)), and finally: (iii) exploring this neighborhood \( \mathcal{N}(s) \) in order to find another solution, say \( s^* \), which is better than \( s \) with respect to the objective function.

If such a \( s^* \) exists, the process is iterated by using \( s^* \) as starting solution. Otherwise (no solution better than \( s \) exists in the neighborhood), \( s \) is a local optimum and several possible escape mechanisms of metaheuristic flavor can be applied (see, Aarts and Lenstra [1] for a complete treatment of LS and metaheuristic algorithms).

Constraint Programming (CP) is a programming paradigm exploiting Constraint Satisfaction techniques (see, Mackworth [6]), and in the following we restrict our attention to CP on Finite Domains (CP(FD)) which is the case of all constraint tools for discrete optimization. CP(FD) models combinatorial optimization problems with a set of variables in the domain of integers, and a set of constraints on these variables. Constraints are both mathematical and symbolic, where symbolic (or global) means that they model well-defined subparts of the overall problem. The most classic example of symbolic constraints is the \texttt{all different}(\( X_1, \ldots, X_n \)) constraint which imposes that variables \( X_1, \ldots, X_n \) must assume different values in a feasible solution. To each constraint is associated a propagation algorithm aimed at deleting from variable domains the values that cannot lead to feasible solutions. Constraints interact through shared variables, i.e., as soon as a constraint has been propagated (no more values can be removed), and at least a value has been eliminated from the domain of a variable, say \( X \), then the propagation algorithms of all the other constraints involving \( X \) are triggered (see again Mackworth [6]).

Since in the general case a full propagation of the constraints is as difficult as the problem itself, propagation algorithms are usually incomplete in the sense that they are possibly stopped with some inconsistent values in the domains.

There are two main directions for LS and CP integration. From one side, the possibility of using constraint-reasoning and techniques based on CP within a classical LS algorithm so as to produce a constrained local search framework. From the other side, the transformation of the classical global
search algorithm provided by CP tools into an incomplete global search one through the use of LS techniques. This second direction seems to be very powerful and promising, typically leading to flexible algorithms whose great advantage is moving through a decision tree. Indeed, the classical exploration methods of a decision tree must be rigid enough to guarantee optimality. However, in a heuristic context, relaxing this rigidity but still remaining inside the tree allows using all the CP machinery in terms of problem reductions and efficiency by obviously accelerating the process.

In this paper, we concentrate on the second direction, and in particular, we focus on the integration of the machinery developed in the Tabu Search (TS) context (see, Glover and Laguna [5] for a complete treatment of the subject) into incomplete global search algorithms based on CP.

The main issue of such an integration is the following. The techniques originating in TS context usually have a complete solution at hand (most of the times feasible, sometimes infeasible) and iteratively improve it (where ‘improving’ includes the fact of moving an infeasible solution toward feasibility). Within a global search algorithm, instead, we want to apply LS techniques (and in particular TS techniques) to partial solutions, i.e., to internal nodes of the branching decision tree. The aim of the paper is to re-interpret some of the techniques developed in the former case to the latter one. The treatment is neither methodologically exhaustive nor provides a complete description of implementation issues, but it represents a kick-off for the research into this new and promising area.

It is worth mentioning that incomplete global search algorithms do not necessarily require an implementation by means of CP tools. There are several effective ways of devising such kind of algorithms in the Integer Programming context (e.g., GRASP [3], recovering beam search [2]).

In the development of the research we will specifically discuss methods for:
1. limiting the size of the tree ‘a priori’;
2. limiting the size of the tree ‘based on history’;
3. restructuring the tree.

Bibliography

Clique-transversals and Clique-independent Sets on Helly Circular-arc Graphs

Guillermo Durán\textsuperscript{1}, Min Chih Lin\textsuperscript{2}, Sergio Mera\textsuperscript{2}, and Jayme L. Szwarcfiter\textsuperscript{3}

\textsuperscript{1}Partially supported by UBACyT Grants X036 and X127, and PID Conicet Grant 644/98. Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina
E-mail: willy@dc.uba.ar

\textsuperscript{2}Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina
E-mails: \{oscarlin, smera\}@dc.uba.ar

\textsuperscript{3}Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, FAPERJ, Instituto de Matemática, NCE and COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 2324, Rio de Janeiro, RJ 20001-970, Brazil
E-mail: jayme@nce.ufrj.br

A complete subgraph in a graph $G$ is a set of pairwise adjacent vertices of $G$. A clique is a maximal complete subgraph. A family of subsets $S$ satisfies the Helly property when every subfamily of it consisting of pairwise intersecting subsets has a common element.

A graph $G$ is \textit{circular-arc (CA)} if there exists a family $\sigma$ of arcs around a circle and a one-to-one correspondence between vertices of $G$ and arcs in $\sigma$, such that two distinct vertices are adjacent in $G$ if and only if their corresponding arcs intersect in $\sigma$. Such a family of arcs is called an \textit{arc model} for $G$. This class has been very studied and has many applications, ranging from genetics to traffic control. A graph $G$ is a \textit{Helly circular-arc (HCA)} graph if there exists an arc model for $G$ such that the arcs verify the Helly property.

Helly circular-arc graphs have been characterized and efficiently recognized by Gavril [3]. In that work, Gavril proved that it is possible to build an arc model that verifies the Helly property in $O(n^2)$ time, if it exists.

A set of vertices that meets all the cliques of a graph $G$ is a \textit{clique-transversal} of $G$. A \textit{clique-independent set} is a collection of pairwise vertex-disjoint cliques of $G$. The \textit{clique-transversal number} and \textit{clique-independence number} of $G$, denoted by $\tau_C(G)$ and $\omega_C(G)$, are the sizes of the minimum clique-transversal and maximum clique-independent set of $G$. The decision problems associated to these parameters are NP-Hard [2] and NP-Complete [1], respectively.

Clique-transversal and clique-independent set problems have been studied in the last years. Two algorithms that solve these problems in HCA graphs are presented in [4]. Both algorithms run in $O(n^2)$ time, if the Helly model is given.

In this paper, we present greedy algorithms for finding the \textit{clique-transversal number} and the \textit{clique-independence number} on Helly circular-arc graphs. Both algorithms run in $O(n\log n)$ time if the Helly circular-arc model is given, improving the algorithms of [4].

The weighted version of the clique-transversal and the clique-independent set problems are studied
too. The vertices of the graph have positive weights attached to them and we seek a clique-transversal which has a minimum total weight, or a collection of independent cliques which has a maximum total weight, respectively (the weight of a clique is the sum of the weights of its vertices).

The weighted clique-transversal problem has been solved in polynomial time for strongly chordal graphs, chordal graphs of bounded clique size and cographs [4], while the weighted clique-independent set problem has not yet been studied in the literature. We describe here polynomial time algorithms for both problems on Helly circular-arc graphs.

Bibliography


Discrete Capacity and Nonbifurcated Flow Assignment Algorithm in Communication Networks

RICARDO P. M. FERREIRA¹, J. A. S. MONTEIRO², AND HENRIQUE P. L. LUNA¹

¹Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, Belo Horizonte, MG 31270-010, Brazil
E-mails: {poley, pacca}@dcc.ufmg.br

²Universidade Salvador (UNIFACS), Salvador, Bahia, Brazil
E-mail: suruagy@unifacs.br

Multicommodity flow problems such as the traffic assignment and the capacity assignment problems are well-known for their relevance in planning congested network systems. The scope of applications is large, but we retain here the interpretation that have been addressed to (store-and-forward) packet-switched computer network discrete capacity allocation and nonbifurcated routing problems [1, 3, 4, 6]. A classical point of view is that the design of communication networks can be modeled as an optimization problem where, on the one hand, the link capacity must be assigned at the lowest cost and, on the other hand, the communication flow must be routed to get the highest quality of service [7]. Normally capacity costs refers to leasing discrete levels of capacity for the links, and quality of service is measured in terms of average packet delay or with the end-to-end average buffer overflow probability.

The network discrete capacity allocation and routing problem is a special issue of a general network design problem. Specifically, the problem assumes that an initial network topology is given, such that there exists at least one path between each origin-destination pair [1, 3]. If the messages in the network follow static and nonbifurcated routes the network will be denominated nonbifurcated. The nonbifurcated multicommodity routing problem is a NP-Hard problem [2]. The network problem studied here searches how to select link capacities and a single route to be used by all kinds of traffic (data, image, voice) between each communicating node in the network, such as to minimize the sum of link allocation costs and nonlinear separable congestion (quality) costs associated with the total flow in each link.

We have implemented a new modelling and algorithmic framework to integrate design and operation in computer communication networks. The integrated approach used associates to the packet delay a congestion cost function, in such a way that the whole problem can be seen in terms of a unique cost criterion [5]. The result is that both the adopted continuous model and the optimization heuristics deal simultaneously with the two conflicting criteria of the problem. On the one hand, the link capacities are assigned at the lowest cost and, on the other hand, the communications flows are routed to get the highest quality of service.

A good convex approximation of each arc cost function is one of the main results that we have explored in our integrated framework. The separability of the new objective function was used to obtain an approximated convex function with an explicit gap. We have showed how to get lower bounds applying efficient algorithms for the solution of the resultant nonlinear bifurcated convex multicommodity flow problem. Economies of scale in capacity allocation induces concavity in leasing costs, but convex congestion costs make the integrated objective function a pointwise infimum of a series of convex functions. The application concerns the simultaneous use of routing implicit capacity assignment on the links, which are expanded as far as induced by congestion costs in order to ensure an acceptable performance level at a minimum total cost. Small execution times obtained to solve
the test problems indicate that the methodology can be applied to solve large scale problems. In fact, our computational experiments suggest that the procedure is both efficient and effective in identifying good solutions for practical problems. Better multicommodity nonbifurcated routing methods could yet be used to improve the execution time and the quality of the solutions.

Bibliography


A Parallel ProtoG Algorithm to Determine the Optimum Configuration of Natural Gas Cogeneration Systems

MARCO C. GOLDBARG$^1$, ELIZABETH F. GOUEVA$^1$, AND FRANCISCO D. M. NETO$^1$

$^1$Universidade Federal do Rio Grande do Norte, Campus Universitário Lagoa Nova, Natal, RN 59072-970, Brazil
E-mails: {gold, beth, fdantas}@dimap.ufrn.br

1 Introduction

Cogeneration is the simultaneous production of electricity and thermal energy from a single energy source such as natural gas, although a variety of fuels can be used. The heat produced from the electricity generating process is captured and used to produce high and low level steam. The steam can be used as a heat source for both industrial and domestic purposes and can be used in steam turbines to generate additional electricity. Cogeneration can be one of the most cost-effective and efficient uses of resources. Industries such as pulp and paper, chemical and food processing, pharmaceutical manufacturing, hotel operations, universities, hospital complexes and many others are among those that can benefit from this type of generation arrangement. The technological issues necessary to determine which components compose a certain cogeneration system are very complex. The technical adjustments needed to optimize a given configuration of cogeneration equipments are, in general, hard problems to be practically solved. Furthermore, there are a huge number of possible configurations to compose a theoretical feasible configuration to supply a given demand. This work presents and compares the performance of a parallel transgenic algorithm with two evolutionary algorithms to solve the problem of determining the best configuration for a cogeneration system that uses natural gas as its single energy source. Two main factors must be taken into account when designing a cogeneration system: its architecture and the demand requirements. To determine a specific cogeneration system architecture one must specify all the components that will be part of the system. The usual components of an energy cogeneration system are: motors, turbines, chillers, boilers and generators. The cogeneration system architecture depends on the system purpose. For instance, systems developed to produce cooling as its output are, in general, composed of a motor that starts a generator and a chiller that can be fired with engine exhaust gases or directly with natural gas. There are a number of possible architectures depending on the purpose of the cogeneration system. There are five usual cogeneration cycles: Combined Cycle, Combined Cycle and Vapor from Process, Cogeneration - Compression and Electricity, Cogeneration - Drying, Cogeneration - Cooling Production-Absorption. To supply the demand requirements, the components of a cogeneration system must match certain specifications. For example, some sorts of motors are not compatible with some types of turbines and some chillers are designed to work with specific components. Due to the great number of equipments commercially available to build cogeneration systems, there are a huge number of possible configurations that can supply a client's demand. Thus, the enumeration of all possible configurations is prohibitive and a heuristic approach must be used to solve the problem.

2 Algorithms

Computational Transgenetics is an approach that develops the evolution of computational information with the interaction of a memetic and a genetic context. By doing so, Computational Transgenetics
redefines the genetic information evolution process and creates a process to evolve memetic information. The approach is based on agents that manipulate genetic information. These agents are also responsible for translating and matching memetic and genetic information and for providing feedback to the evolutionary process. Three main ideas support Computational Transgenetics. At first, to infiltrate information in an evolutionary process, according to a plan, to inform and guide the process. In this way, Computational Transgenetics imitates Genetic Engineering (Dudley, 1990). Second, the manipulation is accomplished by agents and follows the epigenetic paradigm. Thus, the idea of cultural evolution (Boyd and Richerson, 1985) is incorporated to the metaphor. Finally, the prokaryotic recombination paradigm is used as a complementary tool to preserve genetic information. The transgenic agents are logical entities that integrate memetic information with the information encoded on chromosomes. A transgenic agent is composed of one or more memes (Plotkin, 1996) and an operating method.

This work presents a memetic algorithm (Radcliffe and Surry, 1994), that is a genetic algorithm with a local search procedure, and two transgenic algorithms. The transgenic algorithms presented here belong to the ProtoG class. The work presents a sequential and a parallel ProtoG to solve the problem. ProtoG algorithms are uniquely based on prokaryotic recombination to accomplish their evolutionary process (Goldbarg et al., 2002). The agents used to implement the transgenic algorithm were the mobile genetic particles. The population had 1000 chromosomes that were initially randomly generated. The mobile genetic particles' length was two. The meme base was composed of ten heuristic solutions. The heuristic method used to generate the initial solutions of the meme base built feasible solutions by choosing the equipments at random with a higher probability for those equipments with lower costs. When the algorithm finds a solution better than all the others, then this solution takes the place of the worst solution in the meme base. The transgenetic algorithm stop criterion was 2000 iterations with no improvement or any chromosome. The memetic algorithm stop criterion was 2000 iterations with no improvement of the best individual.

3 Computational Experiments and Conclusions

The algorithms were applied to 35 instances randomly generated - 7 instances were generated for each type of cogeneration cycle. The cogeneration system’s sizes varied in a range from 100 to 1000 parts, admitting repetition of equipments and formation of subsystems. The work reports results of 10 independent runs of each algorithm. The results are compared in terms of the best, the average and the worst solutions found by each algorithm. A comparison of runtimes is also reported. The conclusions explore the strong and weak points presented by each approach when applied to the designed set of instances.

4 References


Centre, King’s Buildings, University of Edinburgh.
A ProtoG Algorithm Applied to the Traveling Salesman Problem

MARCO C. GOLDBARG$^1$, ELIZABETH F. GOUVEA$^1$, AND CEZAR M. P. DE SOUZA$^1$

1Universidade Federal do Rio Grande do Norte, Campus Universitário Lagoa Nova, Natal, RN 59072-970, Brazil
E-mails: {gold, beth, cezar}@dimap.ufrn.br

1 Introduction

Biological evolution has been a source of inspiration to solve hard computational problems. Evolutionary algorithms have been applied to a number of different areas and problem domains. Computational Transgenetics is an approach to design evolutionary algorithms based on recent theories for species evolution. This technique enlarges the scope of the classical evolutionary computation by considering two new contexts: the epigenetic and the intracellular. Some genetic operators of Computational Transgenetics are based on the prokaryotic recombination paradigm. The prokaryotic paradigm allows the transgenic algorithms to manipulate the code of a chromosome directly, with no need of crossover operations. One of the classes of algorithms of Computational Transgenetics is called ProtoG. ProtoG algorithms are uniquely based on prokaryotic recombination to accomplish their evolutionary process (Goldberg et al., 2002). This work presents a ProtoG algorithm to solve the Traveling Salesman Problem, TSP. The TSP is one of the most studied classical combinatorial optimization problems. Given a weighted graph $G=(N,A)$ where $N=1,\ldots,n$ and $A=1,\ldots,m$ denote, respectively, the vertex and the edge sets of $G$, the TSP consists in finding the shortest hamiltonian cycle of $G$. Junger et al. (1995) present a detailed investigation of the TSP. Evolutionary algorithms have found good approximate solutions for this problem (Larrainga et al., 1999). The algorithm is applied to 30 instances of the benchmark TSPLIB (TSPLIB95). The solutions found by ProtoG are compared to the ones found by a memetic algorithm presented by Krasnogor and Smith (2000).

2 The Algorithm ProtoG

Computational Transgenetics, CT, is a metaheuristic that aims at infiltrating pieces of information, according to a plan, in an evolutionary process in order to improve it. CT uses information agents to insert pieces of information into the chromosomes. CT can use a number of sources to obtain information to manipulate chromosomes. The method used to encode information mimics the epigenetic format. The non-genetic pieces of information used to guide the evolutionary process are called memes (Plotkin, 1996) in this work. CT uses intracellular agents, such as viruses and mobile genetic particles (Goldberg and Gouvea, 2002) to express epigenetic rules in the computational context. The interaction between the populations of chromosomes and agents results on a co-evolutionary process. To act in the genetic context, manipulating chromosomes, an information agent uses mechanisms similar to the biological horizontal transfer ones - assimilations - and also similar to prokaryotic recombination mechanisms. A chromosome can resist to the manipulation of a transgenic agent. This phenomenon is called immunologic resistance. The immunologic resistance may guide the evolution of the population of agents reinforcing, changing or withdrawing epigenetic rules.

The non-genetic pieces of information constitute an information repository, called meme base. This cultural repository can contain information from any result related to the problem or a given instance.
The information agents used to implement the transgenetic algorithm presented in this work are the Mobile Genetic Particle (MGP) and Viruses. A ProtoG description is given in the following.

ProtoG Algorithm

```
Begin
Load a Meme Base
Generate and evaluate an initial population
Repeat
Generate an agent competing memes
For each chromosome do
If the chromosome is sensitive to manipulation then
Begin
Manipulate it
Evaluate it
If the chromosome satisfies the immune criterion
then include its memes in the Meme Base
End
Until stop criterion be satisfied
End
```

3 Computational Experiments and Conclusions

The algorithms were applied to 30 instances of TSPLIB. The results comparing the performance of the memetic and the transgenetic algorithms were obtained from 10 independent runs for each algorithm. The work presents minimum, average and maximum solutions found by each algorithm and the runtime for each instance. The memes of ProtoG were mainly based on lower bounds, being constituted by fragments of Minimum Spanning Trees and subtours generated from the solution of the Assignment Problem. The work investigates the strong and weak points of the transgenetic approach.

4 References


TSPLIB95. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
The Distribution System Service Restoration: An Exact Method and a Heuristic Search Method

F. A. Grupelli Junior\textsuperscript{1}, Maria T. A. Steiner\textsuperscript{2}, J. Y. Yuan\textsuperscript{3}, C. Carnieri\textsuperscript{3}, N. M. P. Volpi\textsuperscript{3}, V. E. Wilhelm\textsuperscript{3}, N. H. Musit\textsuperscript{3}, C. F. Antonio\textsuperscript{3}, E. M. Queirós\textsuperscript{3}, E. L. Andretta Filho\textsuperscript{3}, E. M. Kalinowski\textsuperscript{3}, A. C. Alves\textsuperscript{3}, C. Gulin\textsuperscript{3}, V. C. Zambenedetti\textsuperscript{4}, and M. Klimkowski\textsuperscript{3}

\textsuperscript{1}COPEL, Curitiba, Paraná, Brazil

\textsuperscript{2}Universidade Federal do Paraná, Departamento de Matemática, Centro Politécnico, CP 19081, Curitiba, PR 81531-990, Brazil
E-mail: tere@mat.ufrpr.br

\textsuperscript{3}Universidade Federal do Paraná, Departamento de Matemática, Centro Politécnico, CP 19081, Curitiba, PR 81531-990, Brazil

\textsuperscript{4}LACTEC

An exact method and a heuristic search method are used and compared for service restoration of a distribution system. The purpose is to develop a restoration plan after the location of a fault has been located and the faulted zone has been isolated. This study was done considering that the restoration service must be executed in a very short time (minimizing the required number of switching operations) and that we have to restore as much load as possible within the out-of-service area (maximizing the customers' satisfaction).

It is analyzed, firstly, an Exact Method for the presented problem, through a Mixed Integer Programming Model [Ciric and Popovic, 2000] for which it was used the LINGO (Language for Interactive General Optimizer) software, through a host program developed using the MATLAB software.

This exact method is compared to a heuristic search method [Hsu et al., 1991], [Nagata et al., 2000], which analysis, firstly, feasible switching operations located inside of the faulted zone, that is, it searches the possibility to solve the problem through some(s) loop(s). After that, if necessary, all the links between the faulted zone (with one or more feeders) with its supporting feeders are verified. It is also verified if the spare capacity margin of the supporting feeder with greatest load value will be wide enough to supply the faulted zone. If that is the case, the problem is solved and just one switching operation (close the feeder tie switch between the interrupted feeder and the referred supporting feeder) will be necessary; otherwise, it is verified if the spare capacity margin, for each group of two supporting feeders, verifying the combination that better attends the faulted feeder. If two supporting feeders can solve the problem, the next step will be search which switch between them must be opened in order to avoid a cycle formation. In this case, it will be necessary three switching operations (one switch, with better solution, has to be opened, and two feeder tie switches between the interrupted feeder, referred above, have to be closed). If two aggressed feeders are not enough, we continue, searching now for three supporting feeders and so on, until a solution that satisfies total or partially the faulted zone, is achieved. For partial solutions, a list of them is presented to the operator of the Power Company for that he/she takes the best choice, considering subjective aspects that are not considered in the algorithm.

References:

CIRIC, R.M.; POPOVIC, D.S. Multi-Objective Distribution Network Restoration using Heuristic


Pucon, Chile, November 4-6, 2002
Multi-item Capacitated Lotsizing Problems with Setup Times: Column Generation and Lagrangean Heuristics

MONIQUE GUIGNARD\textsuperscript{1}, CHOONHO RYU\textsuperscript{2}, HANG QIAN\textsuperscript{3}, AND LYDIA DOWLATH\textsuperscript{4}

\textsuperscript{1}Department of OPIM, Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6366, United States
E-mail: guignard@wharton.upenn.edu

\textsuperscript{2}Hong Ik University, College of Business Administration, Seoul, 121-971, Korea
E-mail: ryu@vow.hongik.ac.kr

\textsuperscript{3}Credit Suisse-First Boston, 302 Carnegie Center, 2nd Floor, Princeton, NJ, United States
E-mail: Harry.Qian@cmfbconnect.com

\textsuperscript{4}Electrical and Systems Engineering, SEAS, University of Pennsylvania, Philadelphia, PA, United States
E-mail: ldowlath@seas.upenn.edu

1 Problem Description

Lot-sizing models are used to support optimal timing and sizing production decisions. They are characterized by the fact that production lots are determined based on a trade-off between production costs (basic production costs and inventory related costs) and customer service (e.g., backorder costs) [15]. Here, we consider the Multi-Item Capacitated Lot-Sizing Problem (MCLP), which assumes one-level production (no assembly) on a single machine, finite planning time horizon, discrete time scale, dynamic demand (deterministic), sequence-independent setup times and costs, uniform production costs, and capacity limits.

2 Formulations

The following constraints must be satisfied: production and setups cannot use more time than is available in each period; the demand in each period must be satisfied; a setup must occur if item \( i \) is produced in period \( t \); production levels are nonnegative, and setup variables are binary. The objective is to minimize the sum of carrying costs, backorder costs, and setup costs.

The formulation we use deals with the backorder (BO) case. The no-backorder case is a very well studied problem [6], [9], [10], [17], [12], [14]. MCLP with backorders however has received relatively little attention. Production systems with BO’s, however, are very much a reality and there is a need for effective approaches to solve production problems in which they are permitted [13]. MCLP with BO’s is NP-hard. Heuristics and mathematical programming approaches developed for MCLP (see [2], [11] and [7]) cannot be easily modified for MCLP with BO’s. [11] illustrated the inability of the Dixon-Silver [6] and the Lambrach-Vanderveken [10] algorithms to handle shelf-life restrictions adequately. These features are easily handled by the formulation considered here.

We also consider setup times. With setup times, even the feasibility problem is NP-complete [18].

53
Additional complexities have been studied, like multi-level production patterns [16], or multiple machines [1] or both. We are limiting ourselves here to single-level single-machine production processes.

3 Aggregate vs. Disaggregate Models

Mathematical models for dynamic production scheduling problems fall into two categories: (1) aggregate models, which decide how much and when a product must be produced, and (2) disaggregate models, which decide what fraction of the demand for a product in what period must be produced in each period. Aggregate models need inventory balance equations while disaggregate models do not. Most authors used aggregate (smaller) models and reported large gaps for MCLP. Eppen and Martin [8] using variable redefinition and [17] using Lagrangean relaxation produced considerably better lower bounds. Disaggregate models used for more general production planning problems ([4],[9],[6]) were adapted by [12] for MCLP, and Lagrangean decomposition produced good lower bounds.

Later research has introduced strong cutting planes and specialized branch-and-cut methods [3].

4 A Lagrangean Approach

We report here on an approach that incorporates (1) a disaggregate model with setup times based on [nil.yan 93], (2) a Lagrangean relaxation of the capacity constraints that decomposes the problem into a UFLP for each product, (3) a column generation scheme for solving the Lagrangean dual problem, and (4) Lagrangean heuristics that modify infeasible solutions to make them feasible and modify feasible solutions to make them more economical.

The column generation scheme converges so quickly (often in around ten iterations) that the Lagrangean heuristics have only few opportunities for finding good feasible solutions. More work is needed and justified to refine them and make them more efficient.

We will present computational results showing that (1) most data sets from the literature can be solved quickly using CPLEX 7.5, and (2) the Lagrangean approach can usually obtain quickly a tight bracket around the optimal value. For some data sets however, either the Lagrangean duality gap is large and then of course, the gap between the Lagrangean bound and the best feasible value produced by the heuristic is even larger, or (2) the duality gap is small but CPLEX cannot solve the problem to optimality in less than 10 to 20 thousand seconds, and in that case a quick Lagrangean approach does make practical sense.

5 References


Probabilistic Satisfiability

PIERRE HANSEN

GERAD and HEC Montréal, Canada

Among many models recently proposed for the treatment of uncertainty in knowledge-based systems, probabilistic satisfiability plays a prominent role. This is largely due to the facts that (i) it agrees with the standard theories of probabilities and logic; (ii) it allows for many extensions and thus has considerable expressive power; (iii) large instances can be solved exactly using the column-generation technique of linear programming, together with nonlinear 0−1 programming to solve the subproblem of finding an entering column; (iv) medium-sized instances can be solved analytically with a computer by enumeration of vertices and extreme rays of polyhedra, which entails a system for automated theorem-proving in probability; (v) there are applications in various fields, from reliability and computer-aided diagnosis to quantum physics.

Problem Statement

The probabilistic satisfiability problem in decision form may be defined as follows: Consider m logical sentences $S_1, S_2, \ldots, S_m$ defined on n logical variables $x_1, x_2, \ldots, x_n$ with the usual Boolean operators $\lor$ (logical sum), $\land$ (logical product) and $\neg$ (negation, or complementation). Assume probabilities $\pi_1, \pi_2, \ldots, \pi_m$ for these sentences to be true are given. Are these probabilities consistent?

There are $2^n$ complete products $w_j$, for $j = 1, 2, \ldots, 2^n$, of the variables $x_1, x_2, \ldots, x_n$ in direct or complemented form. These products may be called, following Leibniz, possible worlds. In each possible world $w_j$ any sentence $S_i$ is true or false. The probabilistic satisfiability problem may then be reformulated: is there a probability distribution $p_1, p_2, \ldots, p_{2^n}$ on the set of possible worlds such that the sum of the probabilities of the possible worlds in which sentence $S_i$ is true is equal to its probability $\pi_i$ of being true, for $i = 1, 2, \ldots, m$. Defining the $m \times 2^n$ matrix $A = (a_{ij})$ by

$$a_{ij} = \begin{cases} 1 & \text{if } S_i \text{ is true in possible world } w_j \\ 0 & \text{otherwise} \end{cases}$$

the decision form of probabilistic satisfiability may be written:

$$\begin{align*}
1_p &= 1 \\
Ap &= \pi \\
p &\geq 0
\end{align*}$$

(1)

where $1$ is a $2^n$ unit row vector, $p$ and $\pi$ are the column vectors $(p_1, p_2, \ldots, p_{2^n})^T$ and $(\pi_1, \pi_2, \ldots, \pi_m)^T$ respectively. The answer is yes if there is a vector $p$ satisfying (1) and no otherwise. Note that not all columns of $A$ need be different. Moreover, not all $2^n$ possible different column vectors of $A$ need, or in most cases will, be present. This is due to the fact that some subset of sentences being true will force other sentences to be true or prohibit them from being so.

Considering one more sentence $S_{m+1}$, with an unknown probability $\pi_{m+1}$ leads to the optimization form of probabilistic satisfiability. Usually the constraints (1) do not impose a unique value for the probability $\pi_{m+1}$ of $S_{m+1}$. As shown by de Finetti [22, 23, 24] this is the case if and only if the line-vector $A_{m+1} = (a_{m+1,j})$ where $a_{m+1,j} = 1$ if $S_{m+1}$ is true in possible world $w_j$ and $a_{m+1,j} = 0$
if not, is a linear combination of the rows of \( A \). Otherwise, the constraints (1) imply bounds on the probability \( \pi_{m+1} \). The satisfiability problem in optimization form is to find the best possible such bounds. It can be written

\[
\begin{align*}
\min / \max \quad & A_{m+1}p \\
\text{subject to:} \quad & \mathbb{1}_p = 1 \\
& A_p = \pi \\
& p \geq 0.
\end{align*}
\] (2)

Nilsson [42] calls (1) and (2) probabilistic logic and probabilistic entailment. However, while (1) and (2) are very useful inference tools they do not properly constitute a logic, i.e., a set of axioms and inference rules. The name of probabilistic satisfiability, proposed by Georgakopoulos, Kavvadias and Papadimitriou [29], appears better suited as it stresses the relationship of (1) with the satisfiability problem, which is the particular case where \( \pi = \mathbb{1} \) and a solution with a single positive \( p_j \) is required (which can be easily deduced from any other solution of (2)).

Both problems (1) and (2) have their origin in the work of Boole [4, 5, 6, 7, 8], where they are called “con- ditions of possible experience” and “general problem in the theory of probabilities”. Boole proposed algebraic methods for their solution (discussed below). Criticized by Wilbraham [46], and later by Keynes [37], Boole’s work in probability was long forgotten in English-speaking countries. It seems however to have strongly influenced de Finetti [21, 22, 23, 24], through Medolghi [41], in the development of his theory of subjective probabilities.

Boole’s work was revived by Hailperin [32, 33, 34] who wrote a seminal paper explaining it with the help of linear programming, and a book-length study of Boole’s logic and probability [33, 34]. Hailperin [32, 33, 34] also obtained several new results and proposed extensions of probabilistic satisfiability, discussed below. Due to its basic character probabilistic satisfiability was often independently rediscovered, sometimes in particular cases or variants, i.e., by Adams and Levine [1], Kounias and Marin [38], Nilsson [42], Čirnokov [15], Gelembe [28] and probably others.

**Probability Intervals (or Imprecise Probabilities)**

Hailperin [32] noted that the use of intervals instead of point values for probabilities is often more realistic and more general than Boole’s “general problem”. Then problem (2) becomes:

\[
\begin{align*}
\min / \max \quad & A_{m+1}p \\
\text{subject to:} \quad & \mathbb{1}_p = 1 \\
& \pi \leq A_p \leq \pi \\
& p \geq 0.
\end{align*}
\] (3)

If bounded variables are used, an equivalent expression in which the number of constraints remains equal to \( m + 1 \) is obtained:

\[
\begin{align*}
\min / \max \quad & A_{m+1}p \\
\text{subject to:} \quad & \mathbb{1}_p = 1 \\
& A_p + s = \pi \\
& p \geq 0 \\
& 0 \leq s \leq \pi - A_p.
\end{align*}
\] (4)

This problem is also discussed in Lad, Dicky and Rahman [39], Jaumard, Hansen and Poggi de Aragão [36], Andersen and Hooker [2]. An extensive study of statistical reasoning with imprecise probabilities, using (3) and various extensions, is due to Walley [45].

Pucón, Chile, November 4-6, 2002
Conditional Probabilities

Another important extension of probabilistic satisfiability is to consider conditional probabilities instead of, or in addition to, unconditioned ones. Indeed, in many cases probabilistic knowledge is only precise when some conditions hold. Use of conditional probabilities was already discussed by Boole [5] for particular examples. It is connected with his idea of independence. Other authors addressing conditional probabilities in the context of probabilistic satisfiability are Hailperin [34], Chesnokov [15], Jaumard, Hansen and Poggi de Aragão [36] and Coletti [19].

Two cases arise: conditionals may be in the constraints of (4) or in the objective function. Several ways of representing the conditional probability \( \Pr(S_k|S_{\ell}) = \frac{\Pr(S_k \land S_{\ell})}{\Pr(S_{\ell})} = \pi_{k\ell} \) in (2) have been proposed. Introducing a variable \( \pi_{\ell} \) for the unknown probability \( \Pr(S_{\ell}) \) leads to the two constraints (Jaumard et al. [36]):

\[
\begin{align*}
A_{k\ell}P - \pi_{k\ell}\pi_{\ell} &= 0 \\
A_{\ell}P - \pi_{\ell} &= 0
\end{align*}
\]

where \( A_{k\ell} = (a_{k\ell,j}) \) with \( a_{k\ell,j} = 1 \) if both \( S_k \) and \( S_{\ell} \) are true in possible world \( w_j \) and 0 otherwise. This way to express conditional probabilities is close to that of Boole [4] who also introduces an unknown parameter. A more compact expression is obtained by eliminating \( \pi_{\ell} \) (Hailperin [34]):

\[
A_{k\ell}P = (A_{k\ell} - \pi_{k\ell}A_{\ell})p = 0
\]

i.e., \( A_{k\ell}' = (a_{k\ell,j}') \) where \( a_{k\ell,j}' = 1 - \pi_{k\ell} \) if \( S_k \) and \( S_{\ell} \) are true, \( -\pi_{k\ell} \) if \( S_k \) is false and \( S_{\ell} \) true and 0 if \( S_k \) is false in possible world \( w_j \).

Adding \( \pi_{k\ell} \) 1 to both sides of (6) gives an equation

\[
A_{k\ell}'P = \pi_{k\ell}
\]

where \( A_{k\ell}' = (a_{k\ell,j}') \) is such that \( a_{k\ell,j}' = 1 \) if \( S_k \) and \( S_{\ell} \) are true, 0 if \( S_k \) is false and \( S_{\ell} \) true and \( \pi_{k\ell} \) if \( S_{\ell} \) is false. Observe that these three values coincide with those given by de Finetti [23, 24] in his definition of the probability of a conditional event in terms of a bet won, lost or cancelled. If the conditional probability \( \Pr(S_k|S_{\ell}) \) is in the objective function, the problem becomes one of hyperbolic (or fractional) programming:

\[
\begin{align*}
\min / \max & \quad \frac{A_{k\ell}'P}{A_{\ell}P} \\
\text{subject to:} & \quad Pp = 1 \\
& \quad A_{\ell}P = \pi \\
& \quad p \geq 0.
\end{align*}
\]

As noted by Hailperin [34] and by Chesnokov [15], a result of Charnes and Cooper [14] may be used to reduce the problem (8) to a linear program with one more variable:

\[
\begin{align*}
\min / \max & \quad A_{k\ell}'P \\
\text{subject to:} & \quad A_{k\ell}P = t \\
& \quad Pp = t \\
& \quad A_{\ell}P = \pi t \\
& \quad p \geq 0, \quad t \geq 0,
\end{align*}
\]

and the same optimal value; the corresponding solution is obtained by dividing the optimal solution \( p^* \) of (9) by \( t^* \).
Additional Linear Constraints

Fagin, Halpern and Megiddo [27] note that if some of the \( \pi_i \) are not fixed they may be subject to \( v \geq 1 \) further linear inequalities. This leads to another extension:

\[
\begin{align*}
\text{min} & / \text{max} & A_{m+1}p \\
\text{subject to:} & & 1p = 1 \\
& & Ap + s = \pi \\
& & \underline{\pi} \leq \pi \leq \overline{\pi} \\
& & B\pi = b 
\end{align*}
\]  

(10)

where \( B \) and \( b \) are a \((v \times m)\)-matrix and a \(v\)-column vector of real numbers. This includes the problem of coherence of qualitative probabilities studied by, among others, Coletti [17, 18, 19] where only order relations between probabilities are given (with an arbitrarily small approximation if some or all of the inequalities are strict). Qualitative conditional probabilities, also studied by Coletti [16, 19], Coletti and Scozzafava [20] lead to a more complex nonlinear model.

Imprecise conditional probabilities can be treated similarly to imprecise probabilities. If \( \overline{\pi}_M \leq \pi_M \leq \underline{\pi}_M \) the corresponding lines in the linear program are

\[
\begin{align*}
A_{k \wedge \ell} p - \overline{\pi}_M A_{\ell} p & \geq 0 \\
A_{k \wedge \ell} p - \underline{\pi}_M A_{\ell} p & \leq 0 
\end{align*}
\]  

(11)

Logical Operations on Conditional Events and their Probabilities

Conditional probabilities \( P(S_1|S_2) \) may be viewed as probabilities of conditional events \( (S_1|S_2) \) which have three truth values: true if \( S_1 \) and \( S_2 \) are true, false if \( S_1 \) is false and \( S_2 \) true and undetermined if \( S_2 \) is false. Such conditional events, implicit in Boole [4] were defined by de Finetti [22, 23, 24] and rediscovered recently by many authors. Proposals for building an algebra of conditional events were made, more or less systematically, by Reichenbach [43], Schay [44], Adams [1], Halperin [33, 34], Dubois and Prade [25], Bruno and Gilio [9], Calabrese [10, 11, 12, 13], Goodman, Nguyen and Walker [31]. Several definitions, often justified on intuitive grounds, were given for conjunction and disjunction operations. Difficulty is largely due to the fact that as shown by Lewis’ Triviality Result [40], there is no expression \( S \) for \( (S_1|S_2) \) in boolean algebra such that \( P_S = P(S_1|S_2) \) except in very particular cases. Goodman, Nguyen and Walker [31] show that the space of conditional events is a Stone algebra, generalizing Boolean algebras. Moreover, they show that different ways to define conjunction and disjunction correspond to different three-valued logics.

Schay [44] proposes two systems:

\[
\begin{align*}
(S_1|S_2) \wedge (S_3|S_4) & = ((S_1 \vee S_4)(S_2 \vee S_3)|S_2 \vee S_4) \\
(S_1|S_2) \vee (S_3|S_4) & = (S_1 S_2 \vee S_1 S_4|S_1 \vee S_4) 
\end{align*}
\]  

(12)

and

\[
\begin{align*}
(S_1|S_2) \wedge (S_3|S_4) & = (S_1 S_2|S_2 S_4) \\
(S_1|S_2) \vee (S_3|S_4) & = (S_1 \vee S_2|S_2 S_4) 
\end{align*}
\]  

(13)

Goodman and Nguyen [30] propose another one:

\[
\begin{align*}
(S_1|S_2) \wedge (S_3|S_4) & = (S_1 S_2|S_1 S_2 \vee S_3 S_4 \vee S_2 S_4) \\
(S_1|S_2) \vee (S_3|S_4) & = (S_1 \vee S_2|S_1 S_2 \vee S_3 S_4 \vee S_2 S_4) 
\end{align*}
\]  

(14)

All three systems have negation defined by

\[
(S_1|S_2) = (S_1 \overline{S}_2|S_2). 
\]  

(15)

Pucón, Chile, November 4-6, 2002
Truth tables for $\overline{S}_1, S_1 \lor S_2$ and $S_1 \land S_2$ as a function of $S_1$ and $S_2$ deduced from rules (12) (15) and (13) (15) are those of Sobociński's and Bochvar's 3-valued logics. Those for the system (14)–(15) correspond to Lukasiewicz and Kleene's 3-valued logics (as well as to Heyting's 3-valued logic concept for $\overline{S}_1$). These results show that any algebraic expression of conditional events can be reduced (in several ways) to a single conditional event. Probabilities of such compound expressions can thus be expressed in probabilistic satisfiability models as usual conditional probabilities.

Iterated conditionals have also been reduced to conditionals in various ways. For instance, Calabrese [13] proposes the relation

$$
\left((S_1|S_2|S_4)\right) = (S_1|S_2) \land (S_3 \lor \overline{S}_4)
$$

(16)

The subject is also discussed in detail in Goodman, Nguyen and Walker [31].

Further results

In the full version of this paper [35] we present a more detailed survey of probabilistic satisfiability and its extensions, as well as ways to solve the resulting programs, analytically and numerically. These topics will be discussed.

Very recent results will also be presented at this conference. They bear upon probabilistic satisfiability with qualitative probabilities and conditional events (for which column generation must be combined with a recent algorithm for nonconvex quadratic programming (Audet et al. [3]) and acceleration of the column generation algorithm through stabilization (du Merle et al. [26]). It appears from these last results that rule-based heuristics and mathematical programming algorithms for probabilistic satisfiability are complementary: the latter ensure the results of the former are correct, while the former give an exploration of why this is so, and provide estimates of optimal values of dual variables for stabilization.

Bibliography


An Approach to the Multiple Traveling Salesmen Problem with Limitant Factors

JOSÉ G. HERNÁNDEZ¹, MARÍA J. GARCÍA², JONATHAN BENAÍM, AND SIMÓN TACHE

¹Universidad Metropolitana, Escuela de Ingeniería de Sistemas, Caracas, Venezuela
E-mail: j.hernandez@unimet.edu.ve

²Minimax Consultores C.A., Gerencia General, Caracas, Venezuela
E-mail: m.garcia.minimax@mercaglob.com

The Traveling Salesman Problem (TSP) is an exciting subject, and although optimal solutions already exist for a big number of cities, many publications about it, and its variations, are still found in the specialized literature. One of its variations, of limited study but great applicability, is the multiple traveling salesmen problem. In this research approaches to the solution of the multiple traveling salesmen problem are presented, with limitant factors focus.

Establishing as objectives: building a tool able to manhole some solution models for the multi traveling salesmen problem, so as that it would minimize the total expenses of the travels, when limitant factors exist.

From the extensive literature about TSP, we quote: Bryant (2000), Applegate et al. (2001), Ríos & González (2000), and a previous work (Hernández & García, 1986). With all this, the TSP is stated, as the starting point for the multiple traveling salesmen problem.

As the TSP, for one agent, since a long time ago (Rao, 1980), the multiple traveling salesmen problem is being studied, and can be expressed as:

$$\min \left\{ z_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} X_{ij} \right\}$$  \hspace{1cm} (1)
subject to:

\[ \sum_{i=2}^{n} X_{ij} = 1 \text{ or } 0 \text{ to } j = 2, \ldots, n \] (2)

\[ \text{if } \sum_{i=2}^{n} X_{ij} = 0 \text{ Then } X_{1j} = 1 \] (3)

\[ \sum_{j=2}^{n} X_{ij} = 1 \text{ or } 0 \text{ to } i = 2, \ldots, n \] (4)

\[ \text{if } \sum_{j=2}^{n} X_{ij} = 0 \text{ Then } X_{il} = 1 \] (5)

\[ \sum_{j=2}^{n} X_{1j} = K \] (6)

\[ \sum_{i=2}^{n} X_{il} = K \] (7)

\[ X_{ij} = 0 \text{ or } 1, \text{ for every } i, \text{ for every } j \] (8)

develop K circuits one for each agent K

The equation (1), is the objective function that minimizes all the travels. Equations (2) and (4), guarantee arrival and departure, one time, for and from each different node that the origin, from where there are more than one departure and more than one arrival, making necessary the conditional equations (3) and (5). These equations point that if the summations are zero, it is because that node is reached from the origin \( X_{1j} = 1 \) or the origin is reached from it \( X_{il} = 1 \). Equations (2) and (7), represent, respectively, that the origin is left \( K \) times and \( K \) times the origin is reached, with \( K \) being the number of necessary traveling agents. The expression (8) controls that variables only take value (one) if the city \( j \) is visited from \( i \), and expression (9) guarantees the presence of as many \( K \) circuits as needs.

Three possible solution modifications to the multiple salesmen problem are stated according the use of one criterion or multiples criterions.

The presence of only one limitant factor force to use the next agent, when the current agent is overflowed by the mentioned factor (charge, distance, time, or any other). If the problem is feasible, there will be circuits for the different agents, as it is cost and total cost.

Use of multiple criteria. The way of solving it is similar to that used for one criterion, two cases can be distinguished: multiple criteria with ponderations and multiple criteria with hierarchy.

Multiple criteria with ponderations. In this case, each criterion has a different weight on the desired solution, but all criteria are evaluated simultaneously, respecting each of these weights. Therefore, the matrix is modified trough a normalization process, accomplishing with this two points, handle different criterions on one base, and the best values are transformed into minor quantities since the shortest route is being searched.

Multiple criteria with hierarchy. In this case the initial matrix is not modified, and like the previous cases, the origin node is the starting point, feasible nodes are being searched, and from this nodes the evaluation proceeds to the hierarchy criterion matrix and the node with the smallest value is visited. Once chosen a node the procedure is repeated from this node until the limitant factors left no feasible nodes, in which case the circuit is closed and the procedure is repeated from the origin to all the

Pucón, Chile, November 4-6, 2002
unvisited nodes.

References


A Hybrid Metaheuristic for the Multi-objective Knapsack Problem

Skander Kort

1Concordia University, 1455 De Maisonneuve Blvd. W., Montreal, PQ H3G 1M8, Canada
E-mail: kort@ece.concordia.ca

1 Introduction

Many engineering problems involve the optimization of several competing objectives. For instance, a
digital systems designer might be interested in maximizing the circuit's performance while minimizing
its area. Formally, the multi-objective optimization problem is defined as follows[1]. Let $f_1, \ldots, f_n$
be $n$ functions defined over a discrete domain $D$. Then, the aim is to maximize these $n$ objectives
subject to a set of constraints.

Due to the contradictory nature of the objectives, these problems do not admit, in general, a single
optimum solution. The aim is rather to find the set of the best trade-off solutions, also called the
pareto optimal solutions. A solution is said to be pareto optimal when it is impossible to improve one
of the objectives without decreasing at least one of the others. Let $x$ and $y$ be in $D$. Solution $x$ is
said to be dominated by $y$ iff $\forall i \in \{1, \ldots, n\}$, $f_i(x) \leq f_i(y)$ and $\exists j \in \{1, \ldots, n\}$, $f_j(x) < f_j(y)$.

This paper describes a hybrid algorithm to solve the multi-objective knapsack problem. This
problem is defined as follows. A knapsack of capacity $W$ and $n$ items are given. Every item $i$ has a
weight $w_i$ and $K$ utilities $c_i^1, \ldots, c_i^K$. The objective is to find out which items should be packed in the
knapsack such that the total utility is maximized and the capacity constraint satisfied.

2 The Hybrid Algorithm

We follow a hybrid approach in solving the multi-objective knapsack problem. This approach combines
a genetic algorithm (GA) and a local search procedure. The GA allows the exploration of the search
space through the application of the genetic operators. The local search procedure is expected to
improve the solutions provided by the GA. It is invoked on every generation just after applying the
genetic operators.

Many studies were undertaken to evaluate the performance of multi-objective GAs. Despite the
difficulties inherent to evaluating such algorithms, most of the studies made the following recommenda-
tions [3]. First, it is necessary to archive the non-dominated solutions found so far by the algorithm.
Second, elitism should be used to guide the GA toward better solutions. We followed both recom-
endations in designing our GA. However, we use a different archiving strategy. The non-dominated
solutions found so far by the GA are still added to the archive. Moreover, the archive contains now
some dominated solutions. A dominated solution $x$ is added to the archive iff the following rules are
fulfilled:

R1 All the solutions dominating $x$ do not belong to the neighborhood of $x$.

R2 The rank of $x$ is less than some fixed value.
The neighborhood of $x$ is defined by all the solutions within some distance (also called niche radius) from $x$. Rule R1 aims at preserving interesting, though dominated, solutions. This is expected to help exploring some low density regions in the search space. NSGA ranking was applied in rule R2. The aim of this rule is to maintain a minimum quality of the archived solutions.

The local search procedure is applied after the genetic operators. This procedure computes a neighbor of the current archive. This is achieved by calculating a neighbor for every archived solution. Both sets are then merged. The resulting set is reduced by applying rules R1 and R2.

3 Evaluation

Many experiments were conducted to evaluate the efficiency of the hybrid approach. The benchmarks from [2] were used. These are bi-objective randomly generated instances. The number of items ranges between 50 and 500. Moreover, the pareto optimal solutions of all these instances are already known. The solutions provided by the genetic algorithm, the local search procedure and the hybrid algorithm are compared. Three performance metrics are used in this study. These metrics are related to the distance between the pareto optimal set and the set provided by the metaheuristic.

Bibliography


Exact Algorithms for the Job Sequencing and Tool Switching Problem

GILBERT LAPORTE\textsuperscript{1}, JUAN JOSÉ SALAZAR\textsuperscript{2}, and FRÉDÉRIC SEMET\textsuperscript{3}

\textsuperscript{1}École des Hautes Études Commerciales, Montréal, Canada
E-mail: gilbert@crct.umontreal.ca

\textsuperscript{2}Universidad de La Laguna, Facultad de Matemáticas, Departamento de Estadística, Investigación Operativa y Computación, La Laguna, Tenerife, Spain
E-mail: jjsalaza@ull.es

\textsuperscript{3}LAMH, Université de Valenciennes, Valenciennes, France
E-mail: frederic.semet@univ-valenciennes.fr

The Job Sequencing and Tool Switching Problem (SSP) arises in the following flexible manufacturing context. We are given a set of jobs $J = \{1, \ldots, n\}$ to be processed sequentially without preemption on a single flexible machine, and a set of tools $T = \{1, \ldots, m\}$. Denote by $J_t$ the set of jobs requiring tool $t$. Each job $j$ requires a subset of tools $T_j$ to be loaded on a magazine capable of holding at most $c$ tools at a time, where $c \geq \max_j(|T_j|)$. In most practical situations, the magazine cannot hold all tools at once, so that some tool switches may be necessary when performing two jobs in succession. A tool switch consists of removing a tool from the magazine and inserting another one in its place. The order of the tools in the magazine is irrelevant. The SSP consists of simultaneously determining the processing sequence of jobs and the subsets of tools present in the magazine for each job in order to minimize the total number of tool switches.

Oreelmas [5] has proved that the SSP is $\mathcal{NP}$-hard for $c \geq 2$. Also, minimizing the number of tool switches for a given job sequence can be solved in polynomial time by means of a “Keep Tool Needed Soonest” (KTNS) policy (Bard [1], Tang and Denardo [6]). This policy states when tool changes are necessary, the tools required for the upcoming job should be kept first in the magazine.

Several heuristics have been proposed for the SSP (Bard [1], Tang and Denardo [6], Orelmam [5], Crausa et al. [2], Hertz and Widmer [3], Hertz et al. [4]). However, to our knowledge, no exact algorithm is available if one excludes an Integer Linear Programming (ILP) formulation proposed by Tang and Denardo [6]. The purpose of this article is to fill this gap.

We compare some formulations and exact algorithms for the SSP, and propose an integer linear programming formulation that improves that of Tang and Denardo [6]. We also devise two enumeration algorithms, the first based on branch-and-cut, the second based on branch-and-bound. We show extensive computational results, solving to optimality instances up to 25 jobs.

As already observed by Tang and Denardo, the SSP is a very difficult combinatorial optimization problem for which strong lower bounds are hard to derive, irrespective of the algorithm used. Our results indicate that a traditional branch-and-bound approach seems superior to mathematical programming for this type of problem.

Acknowledgements: This research was partly supported by the Canada Research Chair in Distribution Management, by the Canadian Natural Sciences and Engineering Research Council under grant OGP00039682, and by “Ministerio de Ciencia y Tecnología”, Spain.
Bibliography


Relax and Cut Algorithms

Abilio Lucena

1 Universidade Federal do Rio de Janeiro, Departamento de Administração, Av. Pasteur 250, Rio de Janeiro 22290-240, Brazil
E-mail: lucena@openlink.com.br

1 Introduction

Attempts to allow exponentially many inequalities to be candidates to Lagrangian dualization date from the early 1980’s. A (possibly incomplete) list of contributions in this area is initiated with the Restricted Lagrangian Approach of Balas and Christofides [1] for the Traveling Salesman Problem. Later on Gavish [10] suggested an Augmented Lagrangian Approach to a Centralized Network Design Problem. Then Lucena [12, 13] proposed a scheme to dualize violated inequalities on the fly. The scheme is very much akin to cutting planes generation, in polyhedral solution approaches to integer and combinatorial optimization problems. Finally, Escudero, Guignard, and Malik [7] proposed an algorithm to solve the Sequential Ordering Problem with Precedence Constraints and used the very adequate term Relax and Cut to name their approach. More recently, Barahona and Ladányi [3] proposed the use of the Volume Algorithm [2] as an alternative to the Simplex or to Interior Point algorithms in Branch and Cut [16].

For the past five years we have been using the term Relax and Cut to denote the whole class of Lagrangian relaxation algorithms where inequalities are dualized as they become violated at the solution to a Lagrangian relaxation subproblem. In this extended abstract we review, in the following section, the basic ideas behind the Relax and Cut algorithm of Lucena [12, 13]. For the oral presentation, details of the application of that algorithm, together with the associated computational results, will be discussed for the Steiner Problem in Graphs [12, 13], the Linear Ordering Problem [3], and a problem in Computational Geometry denoted the Rectangular Partition Problem [6]. In that presentation, we also intend to highlight the use of Lagrangian heuristics which benefit from the dual bounds generated by Relax and Cut algorithms.

2 Relax and Cut

Assume that a formulation for a NP-hard combinatorial optimization problem is given. Assume as well that exponentially many inequalities may be included in it. Such a formulation can be generically described as:

$$\min \left\{ cx : \ Ax \leq b, \ x \in X \right\},$$

(1)

where $x \in \mathbb{B}^n$ is an array of variables, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ and, finally, $X \subseteq \mathbb{B}^n$. Assume, as it is customary in Lagrangian relaxation, that

$$\min \left\{ cx : \ x \in X \right\}$$

(2)

is an easy (polynomial time) problem to solve. On the other hand, in what is unusual for the application of Lagrangian relaxation, let $m$ be exponential in $n$. In spite of that, assume one wishes to dualize

$$\left\{ a_i x \leq b_i : \ i = 1, 2, \ldots, m \right\}$$

(3)
in a Lagrangian fashion and let \( \lambda \in \mathbb{R}_+^m \) be the corresponding array of Lagrangian multipliers. Subgradient Optimization (SO) could then be used to solve

\[
\max_{\lambda \geq 0} \{ \min \{ (c + \lambda A)x - \lambda b : x \in X \} \}. \tag{4}
\]

Optimization is typically conducted here in an interactive way with multipliers being updated so that the optimal value of (4) is attained. For the sake of completeness, let us briefly review the Subgradient Method (SM) of [9] as implemented in [8], which is used to produce our computational results.

At any given iteration of the SM, for given feasible values of Lagrangian multipliers \( \lambda \), let \( \overline{\lambda} \) be an optimal solution to (4). Denote by \( z_{ab} \) the value of this solution and let \( z_{ab} \) be a known upper bound on (1). Additionally, let \( g \in \mathbb{R}^m \) be an array of subgradients associated with the relaxed constraints. For the current solution, \( \overline{\lambda} \), \( g \) is evaluated as

\[
g_i = (b_i - a_i \overline{\lambda}), \quad i = 1, 2, \ldots, m. \tag{5}
\]

In the literature (see [8], for instance) Lagrangian multipliers are usually updated by firstly determining a step size \( \theta \),

\[
\theta = \frac{w(z_{ab} - z_{ab})}{\sum_{i=1}^m g_i^2}, \tag{6}
\]

where \( w \) is a real number assuming values in \((0, 2]\). One would then proceed to computing

\[
\lambda_i \equiv \max \{0; \lambda_i - \theta g_i\}, \quad i = 1, \ldots, m, \tag{7}
\]

and then move on to the following iteration of the SM.

Under the conditions imposed here, the straightforward use of updating formulas (5)–(7) is not as simple as it might appear. The reason being the exceedingly large number of inequalities that would, typically, be dualized.

Inequalities in (3), for every SM iteration, may be divided into three groups. The first one contains inequalities that are violated by \( \overline{\lambda} \). The second group is for those inequalities that have nonzero multipliers currently associated with them. Notice that an inequality may be, simultaneously, in the two groups just defined. Finally, the third group consists of the remaining inequalities and evaluating their subgradients would account for most of the computational burden at a SM iteration.

One should notice that, under the classification proposed above, inequalities may change groups from one SM iteration to another. It should also be noticed that the only multipliers that may directly contribute to Lagrangian costs \((c + \lambda A)\), at any given SM iteration, are the ones associated with inequalities in groups one and two. These inequalities are thus denoted active inequalities. Conversely, group three inequalities are denoted inactive inequalities. Finally, a point to be made is that, from (7), multipliers for inactive inequalities will not change their present null values at the end of the current SM iteration.

Clearly, inactive inequalities do not directly contribute to Lagrangian costs (at a current SM iteration). On the other hand, they do play a decisive role in determining the value of \( \theta \). Typically, for the application being described, the number of strictly positive subgradients, amongst inequalities in group three, tends to be huge. Consequently, the value of \( \theta \) would result extremely small, leaving multiplier values virtually unchanged from iteration to iteration. Bearing this in mind, one may choose to apply (5)–(7) exclusively to subgradients and multipliers associated with active inequalities. Such a choice results in a dynamic scheme where the set of active multipliers may continuously change. Notice, in association, that a multiplier may become active at one given SM iteration, then become inactive at a subsequent one and become, yet again, active at a later iteration.
The scheme above is, in a sense, very much akin to cutting planes generation. It has been firstly proposed and successively used for the Steiner Problem in Graphs [12, 13]. Later on, it has been used for the Edge-Weighted Clique Problem [11], the Quadratic Knapsack Problem [15], the Traveling Salesman Problem [4], the Vehicle Routing Problem [14], the Linear Ordering Problem [5], the Rectangular Partition Problem [6] and the Capacitated Minimum Spanning Tree Problem [17].

Bibliography


Column Generation Technique in Network Design

NELSON MACULAN$^1$, GÉRARD PLATEAU$^2$, AND IRENE LOOSEAU$^3$

$^1$ Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação, C.P. 68511, Rio de Janeiro, RJ 21945-970, Brazil
E-mail: maculan@cos.ufrj.br

$^2$ Université Paris 13, Institut Galilée, LIPN, UMR 7030, 99 Avenue J. B. Clément, 93430 Villetaneuse, France
E-mail: gp@lipn.univ-paris13.fr

$^3$ Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación, Buenos Aires, Argentina
E-mail: irene@dc.uba.ar

The techniques of column-generation were first presented in the early sixties as part of methods to solve linear programming problems with a huge number of variables. When the size of the problem doesn’t allow to store the entire matrix of a LP problem, columns to enter the basis in the Simplex Method are generated by means of an auxiliary problem. Of course this is possible for some problems with a special structure. In [16] Dantzig and Wolfe introduced those techniques in the context of their very well known decomposition algorithm, developed originally to deal with the limited storage capacity computers had at that time. In their seminal works [17], [18], Gilmore and Gomory used this approach to solve the cutting stock problem. They generate columns by solving in each step a knapsack problem. The original problem is an Integer Programming (IP) problem and they used column generation to solve the linear programming relaxation. This relaxation had shown to provide very good lower bounds in practice. Two decades after, Marcotte [24] showed that the optimal value is very often the rounded optimal LP relaxation.

Since then several other applications of column generation to obtain tighter relaxations to IP problems were developed and can be found in the literature.

Column generation methods to solve exactly integer programming problems appeared more recently. So called Branch and Price methods combine branch and bound with column generation methods to solve the LP relaxation at each node, and ad-hoc branching rules designed to maintain the original structure of the pricing problem tractable, all along the branch and bound tree. Sometimes exact methods derive into heuristics when only a subset of the feasible columns in the nodes of the branch and bound tree is used or only the ones generated in the root node are kept.

Several successful applications of column generation techniques to vehicle routing problems are described in [15], [11], [35], [12], [5], [23].

One of the first Branch and Price methods appearing on the literature is the one presented in [15] for the vehicle routing problem with time windows. Desrosiers et al. modeled the problem as a set partitioning problem where columns are generated using a modified shortest path algorithm which takes into account time windows. Branching rules that conserve this shortest path structure of the subproblem are used. In the cases it was no possible to enumerate the entire branch and bound tree, they were able to obtain good lower bounds. A similar approach is used in [11] by Desrochers et al. Ribeiro and Soumis [35] present also a column generation approach for solving the linear programming relaxation of the multiple depot vehiclescheduling problem. They obtained a better bound that those
Loebel [23] developed a column generation method for the LP relaxation of vehicle scheduling problems in public transit that provides very good solutions for huge real cases. In their comprehensive survey on constrained scheduling and vehicle routing problems Desrosiers et al. [12] stated that optimal algorithms based on Dantzig-Wolfe decomposition and column generation schemes have shown to be the most powerful solution methodologies for that kind of problems.

Branel and Simchi-Levi [5] intend to formalize an empirical conclusion that comes out of applications to VRPTW problems: when column generation techniques are applied to problems modeled as set covering ones, the method works well when the gap between the LP relaxation and the IP solution is small. They show that with some assumptions about the distribution of the customers, in the case of VRPTW (vehicle routing problem with time windows) this gap decreases to zero as the number of customers increases.

A column generation approach to the Crew Scheduling Problem in Urban Transit that uses the set covering and shortest path with constraints problems as subproblems is presented in [14]. In [42] Vance et al. present a decomposition algorithm for airline crew scheduling problem improved bounds provided in previous work. Gamache et al. [19] present also a column generation heuristic method for solving a aircrew rostering problem that was able to obtain very good results on real large scale problems.

In [34] an algorithm for a traffic assignment problem arising in a satellite switching systems which can be modeled as a large scale set partitioning problem is presented. The algorithm combines column generation at each node with a ranking procedure of the columns in order to ensure optimality of the final integer solution. Vance et al. [43] implemented an exact algorithm for the one dimensional binary cutting stock problem. They propose branching rules that keeps the structure of the subproblem tractable at each node. (this rules had been previously proposed by Ryan and Foster [33]). In [39] the author compares two algorithms for the same problem based on two different formulations of the master problem. She presents appropriated branching rules for each one. Carvalho presents in [10] a different approach for the general, not necessary binary, case of the cutting stock problem. An arc-flow formulation with side constraints is proposed and solved by column generation techniques. The author shows that the algorithm works well when the gap between the first LP relaxation and the optimal integer value is less than one. He also concludes that it's sensitive to the width of the rolls. In [40] Vandenberg is based on linear programming relaxations for the graph partitioning problem.

Mehrotra and Trick, in [28], develop a method for solving the graph coloring problem using the independent set formulation, that avoids to certain extend symmetry. They use customized branching rules that are similar to the ones in [43]. This approach leads to an heuristic when columns are generated only in the root node. In [21] column generation was used in order to produce heuristics based on linear programming relaxations for the graph partitioning problem.

Bourjolly et al [3] obtain lower bounds for the maximum stable set problem by means of column generation in the frame of a branch and bound method that doesn’t uses linear programming explicitly.

Savelebergh [36] presents a branch and price algorithm to obtain optimal integer solutions for the generalized assignment problem using again a set partitioning formulation. He also shows that truncating the search tree in his procedure gives very good approximation algorithms.

In [44] Van Den Akker et al. describe a column generation method for solving the LP relaxation of a parallel machine scheduling problem. They obtain a very good lower bound and are able to solve some problems to optimality. Hansen et al [20] propose primal and dual algorithms for mixed integer programming and their use to solve the probabilistic maximum satisfiability problem.
Barnhart et al [2] present an overview of column generation techniques for solving integer problems to optimality and a review of several classes of problems that were successfully solved in this way. They intend to generalize ideas that were successful in solving special problems by column generation.

Vanderbeck and Wosley [38], and [45], present an exact method that combines branch and bound with column generation. They developed an ad-hoc branching scheme and tested the algorithm in three types of problems. In [37] the same approach is applied to solve a problem arising in the design of telecommunications networks. In [41] the author goes further in this direction and proposes a Dantzig-Wolfe decomposition based on the discretization of the integer polyhedron associated to a group of constraints and branching schemes appropriated to this. He tests his ideas on the cutting stock and cutting strip problems.

Although the list of articles on column generation we had mentioned above is by no means exhaustive, it is representative enough to conclude, as several authors already did ([29],[2],[45]), that most successful applications of column generation techniques happen in IP problems that are able to be modeled as set partitioning (or set covering) ones. In most of the quoted examples columns of the set partitioning problem have a well defined structure and it has been possible to develop pricing algorithms (exact or heuristic) to identify them. This formulation also allows usually to handle good branching rules compatible with pricing algorithms and able to keep balanced the search tree.

Some articles are also found at the literature that report methods that combine column generations with branch and cut. In [30] Nemhauser and Park use a matching formulation of the edge coloring problem and propose an algorithm with combines a simple separation routine for recognizing odd circuit constraints with a pricing algorithm for the weighted matching problem.

In [1] a multicommodity flow problem is solved combining column and row (cuts) generation. They develop an algorithm which combines a pricing algorithm, branching rules and cut adding that are mutually compatible along the branch and bound tree. Column generation and branching rules generalized previous work, and lifted cover inequalities are added to LP’s in each node.

In this talk we present a general scheme that proved to be successful in solving a problem appearing in network design and could be applied in other problems. We present a special way of getting ride of columns that we don’t want to enter the basis when working with bounded variables and we show how the columns generated at the previous nodes of the branch and bound tree can be reused. We also describe how to use a column-generation technique in the frame of the simplex algorithm for linear programming problems with bounded variable constraints following the ideas of the COLGEN algorithm presented in [26]. We briefly report computational results of a network design problem solved with this method.

Bibliography


Pucón, Chile, November 4-6, 2002


Pucón, Chile, November 4-6, 2002


A Quadratic Algorithm for Projecting a Vector on the Intersection of Two Hyperplanes and a Box in $\mathbb{R}^m$

NELSON MACULAN$^1$, C. P. SANTIAGO$^1$, AND MARIA HELENA C. JARDIM$^2$

1Unversidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação, C.P. 68511, Rio de Janeiro, RJ 21945-970, Brazil
E-mails: \{maculan, prata\}@cos.ufrj.br

2Universidade Federal do Rio de Janeiro, Instituto de Matemática, Rio de Janeiro, Brazil
E-mail: mhelena@cos.ufrj.br

The aim of this work is to present an $O(n^2 \log n)$ time algorithm for the projection of a vector on the intersection of two hyperplanes and a box. We propose a extension of the result of Maculan et al. [5] and others (see Maculan and de Paula [3], Maculan, Mino and Plateau [4], and Michelot [6]) concerning the projection of a vector on a intersection of only one hyperplane with a box, for which we presented an $O(n)$ time algorithm. The extension of the approach makes the problem more difficult.

In this paper, we define this new problem as a quadratic programming problem and we state some new results using the Karush-Kuhn-Tucker optimality conditions. The approach of the proof complexity is based on the efficient median-finding algorithm to determine the median of components of the vector to be projected (see Cormen, Leiserson and Rivest [1]).

Some applications favorables to our development are subproblems of optimization methods (see Held, Wolfe and Crowder [2]). For example, the solution of the dual lagrangian problem in integer programming using $\varepsilon$-subgradients techniques instead of subgradients techniques yields a projection on the intersection of two hyperplanes and a box. The formulation $\varepsilon$-subgradients is interesting because it guarantees to improve the bound provide that $\varepsilon$-subgradients methods are descent methods.

Bibliography


On Chromatic Scheduling Polytopes Coming from the Bandwidth Allocation Problem in Point-to-Multipoint Radio Access Systems

JAVIER MARENCO\textsuperscript{1} AND ANNEGRET WAGLER\textsuperscript{2}

\textsuperscript{1}Departamento de Computación, FCEN, Universidad de Buenos Aires, Pabellón I, Ciudad Universitaria, (1428) Buenos Aires, Argentina
E-mail: jmarenco@dc.uba.ar

\textsuperscript{2}Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, 14195 Berlin, Germany
E-mail: wagler@zib.de

\textbf{Motivation.} The bandwidth allocation in Point-to-Multipoint Radio Access Systems can be modeled as a chromatic scheduling problem \cite{1}. Such problems are NP-hard in general \cite{4} and cannot be approximated in polynomial time with a guaranteed quality. Small instances can be solved by suitable heuristics \cite{1} but, in order to tackle real world instances, more powerful tools as, e.g., cutting plane methods are needed. For that, the polytopes representing the convex hull of all feasible solutions have to be studied. We present several valid and facet-inducing inequalities.

\textbf{Problem description.} The purpose of a Point-to-Multipoint Radio Access System is to supply wireless access to voice/data communication networks. Base stations form the access points to the backbone network, and costumer terminals are linked to base stations by means of radio signals. In contrast to cellular phone networks, the costumers are not mobile but assigned to a fixed base station and have individual communication demands. A central problem is that a link connecting a costumer terminal and a base station may be subject to interference from another link, provided that the same frequency is used.

\textbf{Bandwidth allocation} is the problem of assigning, to each costumer $i$, a frequency interval $[l_i, r_i]$ covering its communication demand $d_i$ within the available radio frequency spectrum $[0, \lambda]$. The goal is to identify interference-free frequency plans, taking into account several further technical and legal restrictions.

This problem can be seen as a chromatic scheduling problem \cite{2} or as a consecutive or interval coloring problem \cite{3, 4} on weighted graphs $G = (V, E, d)$ where $V = \{1, \ldots, n\}$ represents the set of customers, $E$ the set of interfering links $ij \in V \times V$, and $d = (d_1, \ldots, d_n)$ the demand vector and where $s$ can be interpreted as maximal allowed makespan.

\textbf{Integer programming formulation.} We present a model in terms of variables $l_i, r_i \in \mathbb{Z}^+$ for each node $i \in V$ (which represent the frequency interval assigned to customer $i$) and ordering variables $x_{ij} \in \{0, 1\}$ for $ij \in E$ with $x_{ij} = 1$ if $r_i < l_j$, $x_{ij} = 0$ otherwise.

\textbf{Chromatic scheduling polytope.} We study the polytope $P(G, d, s)$ representing the convex hull of the feasible solutions of this problem.

For every integer solution $y$, there is a feasible solution which is symmetric to $y$ with respect to the available spectrum $[0, \lambda]$. The polytope $P(G, d, s)$ reflects the symmetry of the frequency assignments. This symmetry enables us to obtain a symmetric valid (facet-defining) inequality of any valid (facet-defining) inequality.

We give several examples of valid inequalities. In order to prove whether an inequality is facet-
inducing (i.e. a best possible cutting plane), one has to know the dimension of the polytope. We prove that the problem of calculating the dimension of $P(G, d, s)$ is NP-complete. We obtain a bound $\gamma(G, d)$ such that $P(G, d, s)$ is full-dimensional if $s \geq \gamma(G, d)$.

Moreover, we show that constraints $l_i + d_i \leq r_i$ (for $i \in V$) are always facet defining for $P(G, d, s)$. We also show that the bounding constraints $0 \leq x_{jk} \leq 1$ (for $jk \in E$) and a further type of valid inequalities always define facets of $P(G, d, s)$ if it is nonempty. Finally, we present a number of valid inequalities which are facet-defining if $s \gg 0$.

We address the issue of the complexity of the separation problem for some of these inequalities, proving that these problems are NP-complete.

Bibliography


Pucon, Chile, November 4-6, 2002
Subclasses of $k$-trees: Characterization and Recognition

LILIAN MARKENZON$^1$, CLAUDIA M. JUSTEL$^2$, AND NEWTON PACIORNİK$^3$

$^1$Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, CP 2324, Rio de Janeiro 20001-970, Brazil
E-mail: markenzon@nce.ufrj.br

$^2$Instituto Militar de Engenharia, Praça General Tibúrcio 80, Praia Vermelha, Rio de Janeiro 22290-270, Brazil
E-mail: cjustel@ime.eb.br

$^3$Ministério da Ciência e Tecnologia, Praia do Flamengo 200 sala 2401, Rio de Janeiro 22210-901, Brazil
E-mail: npaciornik@met.gov.br

A $k$-tree is either a complete graph on $k$ vertices or a graph $G = (V, E)$ that contains a vertex whose neighbourhood in $G$ induces a complete graph on $k$ vertices and whose removal results in a $k$-tree [5]. This family is a subclass of chordal graphs, for which there is a well known recognition algorithm, using the Lexicographic Breadth First Search (lex-BFS) [6]. The lex-BFS has been also applied to solve other problems; Brandstädt et al. [1] surveyed these results. Partial results of the lex-BFS were used in [2] to efficiently recognize a $k$-tree.

In this paper, we introduce two new subclasses of $k$-trees and their properties. First, we present the definition and characterization of $k$-path graphs, based on the concept of $k$-paths, as defined by Proskurowski [4], that generalizes the classic concept of paths. For $k = 1$, there is only one $k$-path graph with $n$ vertices, that is the same as the path graph $P_n$. For $k \neq 1$, there are several $k$-path graphs with $n$ vertices. We show how the graph $P_n^k$ (the $k$th power of a path graph) is related with the generalized concept. The class of $k$-path graphs with $n$ vertices is denoted $kP_n$. The recognition algorithm, based on the Characterization Theorem, is linear.

We also introduce a new subclass of $k$-trees, the simple-clique $k$-trees ($SC$ $k$-trees). This family contains the $kP_n$ family and it provides a nested hierarchy of new and well-known graph families, as the maximal outerplanar graphs ($mops$) and maximal planar chordal graphs.

Le [3] defined the $k$-line graph of a graph $G$ as a graph whose vertices are the cliques of size $k$ in $G$. Two distinct such vertices are adjacent in the $k$-line graph if and only if they have, in $G$, $k - 1$ vertices in common. The Characterization Theorem of SC $k$-trees shows that a $k$-tree $G = (V, E)$ with $n > k$, is a SC $k$-clique if and only if the $(k + 1)$-line graph of $G$ is a tree.

We can summarize the hierarchy established in this paper as follows.

For a fixed $k > 3$:

$$P_n^k \subset kP_n \subset SC \; k\text{-trees} \subset k\text{-trees}, \; \text{for} \; n > k.$$ 

For $k = 2$ and $k = 3$:

$$2P_n \subset mops = SC \; 2\text{-trees} \subset 2\text{-trees}, \; \text{for} \; n > 2,$$ 

87
3P₃ ⊆ maximal planar chordal graphs = SC 3-trees ⊆ 3-trees, for n > 3.

Bibliography


The Simple Assembly Line Balancing Problem

FERNANDO MARVAL$^1$ AND JUAN JOSÉ SALAZAR$^2$

$^1$ Universidad de Oriente, Núcleo de Sucre, Dpto. Matemáticas, Cumaná Edo, Sucre, Venezuela
E-mail: fmarval@sucre.UDO.EDU.VE

Universidad de La Laguna, Facultad de Matemáticas, Departamento de Estadística, Investigación Operativa y Computación, La Laguna, Tenerife, Spain
E-mail: jj.salaza@ull.es

Let us consider a set of identical machines and a set of jobs to be performed. Each job requires some resource (e.g., seconds) from a machine, and the jobs are related by precedence constraints. The Simple Assembly Line Balancing Problem of type I consists in finding the minimum number of necessary machines to perform all the jobs within a given cycle time (i.e., the maximum available time for a machine). This NP-hard problem has been investigated by several authors, see e.g. [1].

We have introduced and experimented with an Integer Linear Programming model and with some heuristic algorithms. Some results are shown on classical testbed instances, and compared with other algorithm from literature.

Bibliography

The Lights Out Game and von Neumann Cellular Automata

José P. de Melo\textsuperscript{1} and Nei Y. Soma\textsuperscript{2}

\textsuperscript{1}Universidade de Taubaté, Taubaté, São Paulo, Brazil

\textsuperscript{2}Instituto Tecnológico de Aeronáutica, CTA/ITA/IEC, Praça Mal. Eduardo Gomes 50, Vila das Acácias, São José dos Campos, São Paulo 12228-900, Brazil

The Turning the Lights Out (TLO) is a retailed combinatorial game consisting on a square array with 25 lighted buttons where each light can either be in a state on or off. Any move occurs after pushing a single button and it also alters the states of its vertical and horizontal neighbors. Pushing a button twice has the same effect on a lamp as not pushing it and the order in which the buttons are pushed does not influence the final states of the lamps. The objective of the game is to turn off all lights, given an initial set of states. This problem can be solved by a 125 × 125 system of linear equations over GF(2), Anderson and Fei (1998).

For the general case however, i.e. \( n \times n \) square array, different approaches for solving the problem are needed, since the resulting linear system \( (n^2) \times (n^2) \), although polynomial, becomes time and memory consuming. Sutner (1989) showed that the solution to the TLO is associated with the conditions in which the well known von Neumann Cellular Automata has inverse configurations. Sutner, Anderson and Fei (1998) and Soma and Melo (1996), independently of each other and by the same time, showed that the TLO can be related to Chebyshev second order polynomials. These associations imply in further interesting combinatorial properties, such as the Lucas’ numbers, Allouche \textit{et al.} (1996).

To determine if a von Neumann Cellular Automata has an inverse configuration – or what it is the same if the TLO has a general solution – remains a major open question, Barua and Ramakrishnan (1996). It is addressed here a general sufficient condition for the case when the TLO cannot have solution.

Additionally, from that general sufficient condition for the problem, some new arithmetic proofs for classic results from Combinatorial Optimization are also presented, e.g. the generation of the Sierpiński’s Gasket by the Pascal’s triangle coefficients.

References


R. Barua, S Ramakrishnan (1996) \( \sigma \)-game, \( \sigma^+ \)-game, and two dimensional cellular automata, \textit{Theoretical Computer Science}, 154(2), 349-366.


An Exact Algorithm for Graph Coloring

ISABEL MÉNDEZ and PAULA ZABALA

Universidad de Buenos Aires, FCEyN, Departamento de Computación, Buenos Aires, Argentina
E-mails: \{imendez, pzabala\}@dc.uba.ar

Given an undirected graph $G = (V, E)$, a coloring of $G$ is an assignment of colors to each vertex such that the endpoints of any edge have different colors. The chromatic number of $G$ is the smallest number of colors needed to color $G$ and it is denoted by $\chi(G)$. The graph coloring problem (GCP) is to determine $\chi(G)$. GCP is known to be NP-hard for arbitrary graphs [6]. Like most optimization problems on graphs, GCP can be formulated as a linear integer programming problem. Few work using this approach is found in the literature (see for instance [1],[7]). In [8], we studied the facet structure of $\mathcal{C}_P$, the 0/1-polytope associated with our integer programming formulation and some families of facet-defining inequalities were identified: clique, p-color clique, path and block color inequalities. Based on these theoretical results, in this paper we present a Branch-and-Cut algorithm.

Branch-and-Cut Algorithm

LP-based branch-and-cut algorithms are currently the most important tool to deal with linear integer programming problems computationally. To reduce computational time, it is important to have good lower and upper bounds, good rules to partition the feasible set, good strategies to search on the tree and a good strengthening of the linear relaxations. In the following we describe the different aspects we consider in our implementation for GCP.

Preprocessing: A simple heuristic algorithm finds a maximal clique which size, $n_{\text{clique}}$, is set as a lower bound of the chromatic number and we color the vertices in the clique with the first $n_{\text{clique}}$ colors. Then, we eliminate vertices having a no-adjacent vertex on the clique that it is adjacent to any vertex of its neighborhood. Finally, vertices with degree less than $n_{\text{clique}} - 1$ are deleted. From any optimal coloring of the new graph follows an optimal coloring of the original graph. Besides that, we generate a feasible initial coloring applying a partial enumeration heuristic based on DSATUR [2]. This solution gives an upper bound of the chromatic number ($\bar{\chi}$) and allows us to eliminate variables of the model. A limit of 5 seconds is specified for both heuristics but in many instances the complete run time was less than 20% of this limit.

Variable Selection: Following the idea of Brélaz [2] we choose a fractional vertex adjacent to the largest number of differently colored vertices. In case of ties, the vertex with highest degree in the uncolored subgraph is chosen. For each feasible color for the vertex out of the used colors in the subproblem, a new subproblem is created. In addition, a subproblem is created with the vertex receiving the next color.

Enumeration Strategy: We use a depth first search strategy in choosing the node to evaluate. When the number of still uncolored vertices is "small" it was useful to implement the implicit enumeration scheme. This level is a parameter of our implementation. We fix it to 60 for graphs with more than 60 vertices, otherwise the complete enumeration begins on level 2 of the Branch-and-Cut tree.

Cutting plane generation: We limit to 18 the iterations of the cutting plane algorithm on the

---

1This research was partially supported by UBACYT grant X036 and PID CONICET grant 644/98.
root node and 3 iterations on others nodes. A maximum of 1500 cuts per iteration can be added to the formulation. In order to make the clique inequality, the p-color clique inequality and the path inequality separation, we have developed heuristic procedures. The block color inequalities are handled by brute-force.

Computational Experiments

We report the computational experiences with our Branch-and-Cut code on 73 DIMACS benchmark instances [4]. The code was implemented in C++ using the ABACUS framework [5] and CPLEX 6.0 LP solver [3]. We have performed the experiments on a Sun ULTRA workstation and the times are reported on seconds. We use a CPU time limit of two hours. To compare our results, we use the DSATUR code available on Tricks’s page. (mat.gsia.cmu.edu/COLOR/solvers/trick.c). On table 1 we report the instances that our algorithm was able to solve within the CPU time limit. Table 2 shows instances where CPU time was exceeded but lower or upper bounds were improved. The rest of 31 instances do not need Branch-and-Cut phase because the lower and upper bounds obtained by the initial heuristics are equal. Our algorithm solves instances that DSATUR was not able to. In many cases, DSATUR finds the optimal solution very early in the enumeration process but requires too much time to conclude that there is not better solution. Branch-and-Cut was able to obtain the optimal certification faster than DSATUR. Moreover, for instances not solved within the time limit, Branch-and-Cut reduces significantly the initial gap between the lower and upper bounds provided by n\textsuperscript{2}\text{clique} and χ. The advantage of our approach is that provide good lower bounds. So, if the upper bound is good enough, the algorithm has good chances to find the optimal solution. Our results suggest that our algorithm is a promising solution strategy and it has potential improvements.

<table>
<thead>
<tr>
<th>Problem</th>
<th># vertices</th>
<th># edges</th>
<th>n\textsuperscript{2}\text{clique}</th>
<th>χ</th>
<th>χ</th>
<th>Branch-and-Cut</th>
<th>DSATUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSATUR</td>
<td>126</td>
<td>536</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2.00</td>
<td>0.10</td>
</tr>
<tr>
<td>f6002__1</td>
<td>496</td>
<td>11654</td>
<td>55</td>
<td>5</td>
<td>5</td>
<td>40.00</td>
<td>0.30</td>
</tr>
<tr>
<td>f6002__2</td>
<td>451</td>
<td>8690</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td>9.00</td>
<td>0.20</td>
</tr>
<tr>
<td>f6002__3</td>
<td>425</td>
<td>8698</td>
<td>20</td>
<td>30</td>
<td>30</td>
<td>9.00</td>
<td>0.20</td>
</tr>
<tr>
<td>miles1000</td>
<td>128</td>
<td>3216</td>
<td>30</td>
<td>3</td>
<td>3</td>
<td>14.00</td>
<td>0.10</td>
</tr>
<tr>
<td>miles1000</td>
<td>128</td>
<td>5198</td>
<td>71</td>
<td>73</td>
<td>73</td>
<td>25.00</td>
<td>0.10</td>
</tr>
<tr>
<td>ash631GPIA</td>
<td>662</td>
<td>4185</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3180.0</td>
<td>1.70</td>
</tr>
<tr>
<td>1-FullIn__2</td>
<td>93</td>
<td>593</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>2.00</td>
<td>*****</td>
</tr>
<tr>
<td>2-FullIn__3</td>
<td>52</td>
<td>201</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3.00</td>
<td>2558</td>
</tr>
<tr>
<td>3-FullIn__3</td>
<td>80</td>
<td>346</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>1.00</td>
<td>*****</td>
</tr>
<tr>
<td>4-FullIn__3</td>
<td>114</td>
<td>541</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>3.00</td>
<td>*****</td>
</tr>
<tr>
<td>5-FullIn__3</td>
<td>151</td>
<td>792</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>3.00</td>
<td>*****</td>
</tr>
<tr>
<td>queen8__2</td>
<td>64</td>
<td>728</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>37.00</td>
<td>46.00</td>
</tr>
<tr>
<td>mugs88__1</td>
<td>86</td>
<td>146</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>381.00</td>
<td>*****</td>
</tr>
<tr>
<td>mugs88_25</td>
<td>88</td>
<td>146</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1228.0</td>
<td>*****</td>
</tr>
<tr>
<td>mugs100__1</td>
<td>100</td>
<td>166</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4020.0</td>
<td>*****</td>
</tr>
<tr>
<td>mugs100_25</td>
<td>100</td>
<td>166</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5888.0</td>
<td>*****</td>
</tr>
<tr>
<td>1 Insertion__4</td>
<td>67</td>
<td>232</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>3671.0</td>
<td>*****</td>
</tr>
<tr>
<td>3 Insertion__4</td>
<td>56</td>
<td>110</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>12.00</td>
<td>12.70</td>
</tr>
</tbody>
</table>

Bibliography


Pucon, Chile, November 4-6, 2002

<table>
<thead>
<tr>
<th>Problem</th>
<th># vertices</th>
<th># edges</th>
<th>n-Lique</th>
<th>(\chi)</th>
<th>(\chi)</th>
<th>Branch-and-Cut</th>
<th>DNA/TUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIC125</td>
<td>125</td>
<td>3890</td>
<td>9</td>
<td>20</td>
<td>?</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>DSIC125</td>
<td>125</td>
<td>3890</td>
<td>9</td>
<td>20</td>
<td>?</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>DSIC250</td>
<td>250</td>
<td>3218</td>
<td>4</td>
<td>9</td>
<td>?</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>DSIC250</td>
<td>250</td>
<td>3218</td>
<td>4</td>
<td>9</td>
<td>?</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>DSIC250</td>
<td>250</td>
<td>15668</td>
<td>11</td>
<td>30</td>
<td>?</td>
<td>13</td>
<td>36</td>
</tr>
<tr>
<td>DSIC250</td>
<td>250</td>
<td>15668</td>
<td>11</td>
<td>30</td>
<td>?</td>
<td>13</td>
<td>36</td>
</tr>
<tr>
<td>DSIC500</td>
<td>500</td>
<td>121275</td>
<td>72</td>
<td>87</td>
<td>?</td>
<td>76</td>
<td>88</td>
</tr>
<tr>
<td>DSIC500</td>
<td>500</td>
<td>121275</td>
<td>72</td>
<td>87</td>
<td>?</td>
<td>76</td>
<td>88</td>
</tr>
<tr>
<td>queen90</td>
<td>81</td>
<td>2112</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>queen90</td>
<td>81</td>
<td>2112</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>myci166</td>
<td>95</td>
<td>755</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>myci17</td>
<td>191</td>
<td>2360</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1-Inserts</td>
<td>202</td>
<td>1227</td>
<td>2</td>
<td>6</td>
<td>?</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>1-Inserts</td>
<td>607</td>
<td>6237</td>
<td>2</td>
<td>7</td>
<td>?</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>2-Inserts</td>
<td>149</td>
<td>541</td>
<td>2</td>
<td>5</td>
<td>?</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2-Inserts</td>
<td>597</td>
<td>3936</td>
<td>2</td>
<td>6</td>
<td>?</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>3-Inserts</td>
<td>281</td>
<td>1046</td>
<td>2</td>
<td>5</td>
<td>?</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3-Inserts</td>
<td>1406</td>
<td>9935</td>
<td>2</td>
<td>6</td>
<td>?</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4-Inserts</td>
<td>79</td>
<td>156</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4-Inserts</td>
<td>1475</td>
<td>1795</td>
<td>2</td>
<td>5</td>
<td>?</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>1-FullIn</td>
<td>282</td>
<td>3247</td>
<td>3</td>
<td>6</td>
<td>?</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2-FullIn</td>
<td>212</td>
<td>1621</td>
<td>4</td>
<td>6</td>
<td>?</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2-FullIn</td>
<td>852</td>
<td>12201</td>
<td>4</td>
<td>7</td>
<td>?</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>3-FullIn</td>
<td>405</td>
<td>3524</td>
<td>5</td>
<td>7</td>
<td>?</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3-FullIn</td>
<td>2030</td>
<td>33751</td>
<td>5</td>
<td>8</td>
<td>?</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>4-FullIn</td>
<td>690</td>
<td>6650</td>
<td>6</td>
<td>8</td>
<td>?</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Pucón, Chile, November 4-6, 2002
Three-dimensional On-line Packing with Rotations

Flávio K. Miyazawa\textsuperscript{2} and Y. Wakabayashi\textsuperscript{2}

\textsuperscript{2} Instituto de Computação, Universidade Estadual de Campinas, Caixa Postal 6176, Campinas, SP 13084-971, Brazil
E-mail: fkm@ic.unicamp.br

\textsuperscript{3} Universidade de São Paulo, Rua do Matão 1010, Cidade Universitária, São Paulo, SP 05508-900, Brazil
E-mail: yu@ime.usp.br

1 Introduction

We study a three-dimensional packing problem, which is a special case of the problem \( TP^2 \) (z-oriented three-dimensional packing problem), defined as follows. Given a list \( L \) of rectangular boxes, pack the boxes in \( L \) into a unique box \( B = (l, w, \infty) \), orthogonally and oriented in the z-axis, in such a way that the height of the packing is minimized. In this problem the boxes in \( L \) are allowed to be rotated (by ninety degree) around the z-axis, but are not allowed to be turned down. We consider the special case of \( TP^2 \) in which \( l = w \), that is, the box \( B \) has square bottom and infinite height. In this case we may clearly assume that \( l = w = 1 \) and any box in \( L \) has each of its dimensions bounded by 1. We denote this special problem by \( stTP^2 \).

Given a packing algorithm \( A \) and an input list \( L \) of boxes, we denote by \( A(L) \) the height of the packing generated by \( A \) when applied to \( L \); and by \( OPT(L) \) the height of an optimum packing of \( L \). We say that \( w \) is an asymptotic performance bound of algorithm \( A \) if there exists a constant \( \beta \) such that for all lists \( L \) the following holds: \( A(L) \leq w \cdot OPT(L) + \beta \).

Li and Cheng [1] were the first to present an approximation algorithm for \( TP^2 \). They presented this problem as a model for a job scheduling problem in partitionable mesh connected systems. In this problem a set of jobs \( J_1, J_2, \ldots, J_k \) must be processed in a partitionable mesh connected system that consists of \( l \times w \) processing elements connected as a rectangular mesh. Each job \( J_i \) is specified by a triplet \( J_i = (x_i, y_i, t_i) \) indicating that a submesh of size either \((x_i, y_i)\) or \((y_i, x_i)\) is required by job \( J_i \), and \( t_i \) is its processing time. The objective is to assign the jobs to the submeshes so as to minimize the total processing time. On-line algorithms for this kind of scheduling problems are desirable as most of the time, the operational system does not know about further jobs users can start. Moreover, parallel computer with square mesh topology are also very common [2]. Thus this special case of three-dimensional packing problem we treat here is very attractive for this application.

In [6] we describe an off-line asymptotic approximation algorithm for \( stTP^2 \) with performance bound 2.528; this is the best bound known so far. We present here an on-line algorithm for \( stTP^2 \) with an asymptotic performance bound that can be made as close to 2.6875 as desired. To our knowledge, this is the best bound known for this problem.

A special case of this problem is the two-dimensional bin packing problem, where the items are rectangles and the bins are two-dimensional unit squares. Sieden and See [7] showed that 1.621 is a

\textsuperscript{1}This research was partially supported by MCT/CNPq through the PRONEX program (proc. 664107/97-4) and CNPq (procs. 470008/01-3, 464114/00-4, 300301/98-7, and 304527/88-4).
lower bound for the on-line square packing into unit two-dimensional bins. Therefore, 1.621 is also a lower bound for any on-line algorithm for sTPP $^2$.

The well-known variant of TPP $^2$, in which the boxes are not allowed to be rotated, has been more studied. In [5] we describe an off-line algorithm with asymptotic performance bound 2.67 (the best bound known). Li and Cheng [4] presented an on-line algorithm for this problem with asymptotic performance bound $w_{r,s}$, where $\lim_{r,s \to \infty} w_{r,s} = 2.89$, also the best bound known for this case.

2 The algorithm

In this section we present a family of algorithms, denoted by $R_{k,p}$ where $0 < p < 1$ and $k \geq 1$ ($k$ integer). Each algorithm uses a rounding technique applied to the height of the boxes. Each box has an upper height rounded up to the nearest power of $p$. Then, each box with height $p^i$, which is called $i$-box, is packed into a level of height $p^i$, which is called $i$-level.

The algorithm $R_{k,p}$ uses the technique of rounding up the height of the boxes to the nearest power of $p$ and sorting them by types according to their length and width. A specialized algorithm is then applied to pack the boxes of each different type.

Let $1/2 = r_0 > r_1 > \cdots > r_{k+1} = 1/3$, $s_i = 1 - r_i$, be numbers such that $r_is_{i-1} = r_{i+1}s_i$, $i = 1, \ldots, k+1$. Denote by $C[tx,ux;ty,uy]$ the set of boxes $c$ with $tx < x(c) \leq ux$ and $ty < y(c) \leq uy$. Let $C_k := \bigcup_{i=1}^{k} C[r_{i-1},r_i;1/2,s_i]$.

The following algorithms are used as subroutines in the algorithm $R_{k,p}$: MC, FF and NF. The subroutine MC packs boxes of $C_k$ and is a modification of the algorithm COLUMN presented in [5]. The algorithm FF is an adaptation of the algorithm FF$^{(2)}$ presented by Li and Cheng in [3] for the two-dimensional bin packing problem. It is used to pack boxes with small dimensions. The algorithm NF packs restricted type of boxes into levels, like the well-know Next Fit algorithm: when a box cannot be packed side by side in the $x$ direction, it is packed in the $y$ direction at width $p$, and whenever a box cannot be packed in a same level, it is packed in a new level.

Due to space limitations, we do not give here a detailed description of these subroutines (this will be done in the full paper).

**Algorithm R$_{k,p}$**

*Input:* List of boxes $L$.

1. For each box $b \in L$ do
2. Rotate $b$ such that $y(b) \geq x(b)$.
3. For $i = 1, \ldots, 10$, let $T_i$ (set of boxes) and $A_i$ (algorithm) be defined as follows:

   - $T_1 := C_k$ and $A_1 := \text{MC}_k$;
   - $T_2 := C[0,1;1,1]$ and $A_2 := \text{NF}_1$;
   - $T_3 := C[0,1/2;1,1] \setminus C_k$ and $A_3 := \text{NF}_1$;
   - $T_4 := C[0,1/2;0,1/2]$ and $A_4 := \text{NF}_1$;
   - $T_5 := C[0,1/3;0,1]$ and $A_5 := \text{NF}_1$;
   - $T_6 := C[0,1/3;0,1/3]$ and $A_6 := \text{NF}_1$;
   - $T_7 := C[0,1/3;0,1/3]$ and $A_7 := \text{NF}_2/3$;
   - $T_8 := C[0,1/3;0,1/3]$ and $A_8 := \text{NF}_2/3$;
   - $T_9 := C[0,1/3;0,1/3]$ and $A_9 := \text{FF}_4$;
   - $T_{10} := C[0,1/3;0,1/3]$ and $A_{10} := \text{FF}_4$;

4. Let $j$ be such that $b \in T_j$, and let $N_j$ be the set of levels generated by the algorithm $A_j$ applied to boxes of $T_j$.
5. Pack $b$ into $N_j$ using subroutine $A_j$: if necessary, generate a new level of height $p^m$ where $p^{m+1} < z(b) \leq p^m$.
6. Return the generated packing.

The following result holds for this algorithm.

Pucon, Chile, November 4-6, 2002
**Theorem 1** Let $L$ be an instance for sTPP. Then, $R_{k,p}(L) \leq w_{k,p}\text{OPT}(L) + \left(\frac{k+14}{1-p}\right)$, where $\lim_{k\to\infty,p\to1} w_{k,p} \leq 2.6875$.

3 Conclusion

The algorithm we described can be implemented to run in polynomial time. To our knowledge, it is an on-line algorithm with the best asymptotic performance bound that has been obtained for sTPP.

Bibliography


A Set-Covering Based Heuristic Approach for Bin-Packing Problems

MICHELE MONACI\(^1\) AND PAOLO TOTH\(^1\)

\(^1\)Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
E-mails: \{mmonacci, ptoth\}@deis.unibo.it

Several combinatorial optimization problems can be formulated as large-size Set-Covering (or Set-Partitioning) Problems. This is the case of many problems in the Cutting & Packing area, as well as of other well-known and relevant problems in combinatorial optimization, such as the Vehicle Routing Problem, the Graph Coloring Problem or the Crew Scheduling Problem.

In this work, we use the \textit{Set-Covering formulation} for obtaining a general heuristic algorithm for combinatorial optimization problems which can be formulated as mentioned above, and describe our implementation of the algorithm for solving two variants of the well-known (One-Dimensional) Bin Packing Problem: the \textit{Two-Dimensional Vector Packing Problem (2-DVP)}, in which we are required to pack a set of items (each having a positive weight and a positive volume) into the minimum number of identical bins having a given weight and volume capacity, and the \textit{Two-Dimensional Bin Packing Problem (2BP)}, in which we have to pack a given set of rectangular items (each having a positive width and a positive height) into the minimum number of identical rectangular bins having a given width and height. For the considered problems, each \textit{column} of the associated Set-Covering formulation corresponds to a feasible (and inclusion maximal) item set (i.e., to a feasible “filling” of a bin).

In our approach, both the "column-generation" and the "column-optimization" phases are heuristically performed. In particular, in the first phase, we do not generate the entire set of columns, but only a small subset of it, by simply using some heuristic algorithms from the literature. Each feasible solution corresponds to a set of columns, and our Set-Covering matrix is obtained by considering all the solutions obtained by the iterative application of simple heuristic algorithms (possible identical columns are removed by means of a hashing procedure). Thus, the aim of the first phase is twofold: on the one hand, we want to obtain a good feasible solution, while on the other hand this phase is aimed at generating a set of different columns (possibly all the "good" feasible columns). In the second phase, we heuristically solve the associated Set-Covering instance by means of the Lagrangian-based heuristic algorithm proposed by Caprara, Fischetti and Toth [2].

A similar approach has already been proved to be quite effective for the Crew Scheduling Problem (see Caprara, Monaci and Toth [3]). The main aim of this work is to show that this technique gives good results also for different kind of problems, such as the packing problems mentioned above. Indeed, for both problems, we use a large set of instances from the literature in order to test the effectiveness of the proposed algorithms. In particular, for 2-DVP we compare our results with those obtained by the heuristic and exact algorithms of Caprara and Toth [4], while for 2BP we perform a comparison with the results obtained by the exact algorithm of Martello and Vigo [7], the Tabu Search algorithm of Lodi, Martello and Vigo [6], the Guided Local Search algorithm of Fero, Pisinger and Zachariasen [5], and the constructive heuristic by Boeschetti and Mingozzi [1]. The computational results show that, for both problems, the new approach is competitive with the best algorithms proposed so far in the literature.
Bibliography


Ant Colony Optimization for Mine Planning

Michael Mooszen and María-Cristina Riff

1 Computer Science Department, Universidad Técnica Federico Santa María, Valparaíso, Chile
2 Computer Science Department, Universidad Técnica Federico Santa María, Valparaíso, Chile
E-mail: mcriff@inf.utfsm.cl

Chile is the biggest copper-producing country, and its economy is principally related to the price of this mineral. Therefore, the gain is directly related to an efficient extraction process of the mine. In this context, we are involved in a project where our goal is to propose a new way to schedule the extraction, taking into account the most of the related constraints. The problem has a lot of different constraints types, where the hardest one is the technical constraint. This constraint requires to use a bottom-up extraction technique. The others ones are operational constraints like capacity and timing, as well as physical and geological ones.

In the last ten years, many efforts have been made in order to solve binary linear programming copper mine models. However, we are not able to solve them in a polynomial time of computing because of their complexity. We have used CPLEX relaxing some constraints, but the problem is still hard for these kind of techniques.

Our copper mine scheduling extraction problem is a Constraint Optimisation Satisfaction Problem.

Because of the success obtained solving complex problems like timetabling problem, scheduling [1], vehicle routing problem [2], travel salesman problem [3], constraint satisfaction [8], [9], and real-world applications [4], [5], [7], [6] using genetic algorithms, tabu search, simulated annealing, local search techniques, ant colonies, metaheuristics and hybrid algorithms, we have been motivated to tackle our problem by these techniques. Our goal is to design an ant colony based algorithm that uses ideas coming from constraint satisfaction and local search communities to solve it. In this research report, we propose an algorithm based on ants colonies which takes into account all the constraints.

Using these techniques we are able to solve very complex instances of the problem. We have obtained better results applying an Ants Colonies based technique than the linear programming relaxed model. The algorithm found a better optimal value for our real world problem. On the other hand, we have also studied the behaviour of the algorithm solving other kind of mine configurations, with more levels than the real one, and with some instances hardest than the real one, obtaining very good results comparing with complete traditional approaches.

Bibliography


1Research supported by the USM Project, the FONDEF Project Complex Systems, and CNRS-CONICYT Project France.


Exploring Alternative Network Flow Models for the Capacitated Vehicle Routing Problem

DANIEL NEGRÓTTO², JULIÁN ARÁOZ³, ELENA FERNÁNDEZ³, AND IRENE LOISEAU²

²Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación, Buenos Aires, Argentina
E-mails: {dnegrotto, irene}@dc.uba.ar

³Departamento de Estadística e Investigación Operativa, Universidad Politécnica de Catalunya, Spain
E-mails: {julian.araoz, elena}@upc.es

In the Capacitated Vehicle Routing Problem (CVRP), we need to find a set of routes, each one starting at the central depot and ending at the same place after visiting a subset of the n clients. Each client has to be visited exactly once, and capacity of vehicles has not to be surpassed.

The objective is to minimize the total distance traveled by the fleet of m vehicles. CVRP is known to be NP-Hard and the most promising approaches to solve it exactly come from branch and cut methods. Flow models have been proposed for some classes of VRP, although we found only few references of this kind of model for CVRP. Some classical formulations can be seen this way (see, e.g. Toth and Vigo [7], Laporte, Nobert and Desrochers [3] and Conneuols and Harche [1]). Ralphs and Hartman [5] also introduce a model with $O(n^2)$ two-index binary variables. Fischetti, Salazar González and Toth present in [2] a three-index model using $O(n^2m)$ binary variables providing more flexibility than two-index formulations. In this work we present a family of three-index flow formulations using only $O(n^2)$ variables but having the same flexibility. The underlying idea is to represent the routes inside the client nodes in contrast to the classical arc route multi-commodity representation (notated by the 3rd. index). Preliminary results using these models showed to be promising.

Mathematical Model

Let $M = \{1 \ldots m\}$ represent the set of vehicles, $Q = \{1 \ldots n\}$ the set of clients, $s$ an artificial source node, $t$ an artificial target node and then we form the set of nodes $V = M \cup Q \cup \{s, t\}$. We define the set of edges $A = \{(s, m) | m \in M\} \cup \{(m, q) | m \in M, q \in Q\} \cup \{(q, j) | q, j \in Q, i \neq j\} \cup \{(q, t) | q \in Q\} \cup \{(t, s)\}$ and finally the directed graph of the CVRP model like $G = (V, A)$. We assume the graph is complete but is possible to eliminate some arcs according to the application. Every edge has an associated cost $c_e$ function (representing the travel cost or distance $c_{ij}$ if $e$ represents the link between the clients $i$ and $j$, the distance $a_{0j}$ if $e$ represents the connection between the depot and a client node, the distance $c_{0i}$ if $e$ represents the travel end, i.e., last client visited before returning to depot and the value 0 on all other cases). We split the client and vehicle nodes in order to represent on each one the associated route. We will describe how to split the nodes (we will see afterwards that vehicle nodes splitting could be omitted in order to minimize the number of nodes):

1. Client nodes (now renamed meta-client nodes) are split in 3 nodes, $q_{ia}, q_{ib}$ and $q_{ic}$ corresponding to the original client node $q_i$. We connect all the original $q_i$ entering arcs to $q_{ia}$ and all exiting arcs to $q_{ic}$. Besides, we add one new arc connecting the nodes $q_{ib}$ and $q_{ic}$ with lower and upper

¹This research was partially supported by UBACyT Grant X006, PID CONICET Grant 644/98 and PICT Grant 11-9112

105
bounds equal 1 to assure each node is visited exactly once (we will see afterwards that this arc and the $q_{ik}$ node could be omitted, but on this early model they were included to clarify the exposition). Finally, we add $m$ new multi-arcs between $q_{ia}$ and $q_{ib}$ each one representing a vehicle route with lower bound 0 and upper bound 1.

2. Every vehicle node (now renamed meta-vehicle node) is split on 2 nodes: $a$ and $b$ notated $m_{ia}$ and $m_{ib}$ when they represents to the original vehicle node $m_i$. All the original $m_i$ entering arcs are connected to $m_{ia}$ and all exiting arcs to $m_{ib}$. We add $m$ new multi-arcs connecting $m_{ia}$ and $m_{ib}$ with lower and upper bounds equal 0, excluding the $i$ arc which contains upper and lower bounds equal 1. This will be simplified on the other models.

With all split nodes and the new arcs we obtain the node set $V'$ ($Q'$ and $M'$ for clients and vehicles respectively) and the arc set $A'$ building the associated directed graph $G' = (V', A')$ and cost $c$ function with takes the 0 value on all new arcs. Therefore, the new model contains simultaneously two-index variables and three-index variables (multi-arcs).

Taking a subset $S$ of $V'$, is necessary to denote by $\delta^+(S)$ the set of edges of $G'$ that have exactly one tail in $S$ and head not in $S$, and by $\delta^-(S)$ the set of arcs of $G'$ having head in $S$ and tail not in $S$. Also, we denote by $r(S)$ the minimum number of vehicles needed to serve all clients in $S$. This value may be determined by solving the Bin Packing Problem (BPP) associated with the CVRP (see, e.g., Martello and Toth [4]). Like BPP is NP-Hard in the strong sense, $r(S)$ is generally replaced by the trivial BPP lower bound $\left\lceil \frac{\sum_{a \in S} p_a}{\max_{k=1, \ldots, m} k} \right\rceil$ where $p_i$ denote the client $i$ demand and $k_k$ the vehicle $k$ capacity.

We are ready to present a first mathematical programming formulation. The binary variable $X$ is used to indicate if an arc is active in the optimal solution. In other words, variable $X_{i,j}$ takes value 1 if arc $(i, j) \in A'$ belongs to the optimal solution and takes value 0 otherwise (equivalently with three-index variables). Sometimes we use a one-index notation to denote the arcs (e.g. $X_a$ denotes at the same time two-index arcs and three-index arcs). On other places we use directly the two-index notation and three-index notation (e.g. $X_{i,j}$ and $X_{i,j,k}$, respectively). Finally the integer variable $X_{t,s}$ represents the arc $(t, s)$ and his value represents the number of vehicles used on the optimal solution.

$$\min \sum_{a \in A'} c_a X_a$$  \hspace{1cm} (1)

subject to

$$\sum_{a \in \delta^+(i)} X_a = \sum_{a \in \delta^-(i)} X_a \quad i \in V'$$  \hspace{1cm} (2)

$$\sum_{q \in Q} p_i X_{q_i,q_1,k} \leq k \quad k = 1, \ldots, m$$  \hspace{1cm} (3)

$$X_{q_i,q_1,k} - X_{q_j,q_2,k} \leq 1 - \left( X_{q_i,q_1} + X_{q_j,q_2} \right) \quad q_i, q_j \in Q \quad i \neq j \quad k = 1, \ldots, m$$  \hspace{1cm} (4)

$$X_{m_i,m_1,k} + X_{q_j,q_2} \geq 2X_{m_i,m_1} \quad m_i \in M \quad q_j \in Q$$  \hspace{1cm} (5)

$$\sum_{(q_i,q_j) \in A'(S)} X_{q_i,q_j} \leq |S| - r(S) \quad S \subseteq Q' \quad 2 \leq |S| \leq n - 1$$  \hspace{1cm} (6)

$$X_{m_i,m_1,k} = 0 \quad m_i \in M \quad i \neq k \quad k = 1, \ldots, m$$  \hspace{1cm} (7)

$$X_{q_i,q_j} = 1 \quad q_i \in Q$$  \hspace{1cm} (8)

$$r(Q) \leq X_{t,s} \leq m$$  \hspace{1cm} (9)

$$X_a \in \{0,1\} \quad a \in A' \setminus (t, s)$$  \hspace{1cm} (10)

$$X_{t,s} \text{ integer}$$  \hspace{1cm} (11)

Pucón, Chile, November 4-6, 2002
Note that we use here the sets $Q$ and $M$ that represents the original nodes before the splitting operation (meta-clients and meta-vehicles nodes), in order to simplify the notation. Constraints (2) are the classical flow conservation restrictions. Constraints (3) are the capacity restriction for each vehicle. Constraints (4) and (5) determines the routes and the corresponding logic on the node multi-arcs. Here we establish that each node visited by the $k$ vehicle must contain one flow unit on $k$ multi-arc. In addition, constraints (4) are also used to warrant one single travel by arc (only in one direction). Constraints (6) are the well known generalized subtour elimination constraints. We use $A(S)$ to denote the set of arcs with both ends on the node set $S$. Constraints (7) and (8) fixes some flow arcs as were mentioned before. Finally, constraint (9) establishes lower and upper bounds to the number of vehicles used in any solution. The above formulation uses exactly $m + mn + n^2$ two-index binary variables, $m^2 + mn + n$ three-index binary variables and one integer variable. This summarize $1 + m^2 + 2mn + m + n + n^2$ variables. In most real cases we have $n \geq m$ so the model uses $O(n^2)$ variables. We improve this model and propose two others, one with decision variables using $1 + mn + m + 2n + n^2$ variables, and the other one without decision variables and using $1 + 2mn + n + n^2$ variables.

Preliminary tests have been carried on some small instances taken from TSPLIB [6]. We are working on devising facet structure and valid inequalities for the polytope associated with our integer programming formulations as a first step for developing of a Branch-and-Cut algorithm.

Bibliography


A Polyhedral Approach to Some Nonconvex Optimization Problems

GEORGE NEMHAUSER$^1$ AND ISMAEL DE FARIAS$^2$

$^1$ School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, United States

$^2$ Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, United States

Many optimization models that arise in practical applications are very difficult to solve because of combinatorial constraints that yield nonconvex feasible regions and/or nonconvex objective functions. We call these problems Nonconvex Combinatorial Optimization Problems or NCOPs for short. This talk is about solving NCOPs by linear programming based branch-and-bound algorithms.

Linear programs with integrality restrictions on some of the variables, which are known as mixed-integer programs, or MIPs, are the best-known and most studied NCOPs. In the last decade, very powerful branch-and-bound algorithms have been developed for solving both general and structured MIPs. These algorithms typically use preprocessing, cuts, primal heuristics and specialized branching to control the size of the branch-and-bound enumeration tree.

Most NCOPs can be reformulated as MIPs by the addition of auxiliary binary variables. For this reason the study of NCOPs other than MIPs has been very limited. However, reformulation may have many disadvantages including increasing the size of the problem significantly and losing the problem structure.

There are some NCOP structures that arise naturally in many practical applications and therefore merit study in their own right. These include:

- semi-continuous: if a nonnegative variable is positive, it must be at least some positive constant;
- $k$-cardinality: no more than $k$ variables from a set of $n$ nonnegative variables may be positive; note that for $k = 2$, we get the special case of complementarity where the product of 2 variables equals zero.
- special ordered set of type 2: no more than 2 variables from a sequence of $n$ nonnegative variables may be positive, and if 2 variables are positive, they must be adjacent in the sequence.

The primary objective of this talk is to discuss the polyhedral structure of linear programs subject to some of these NCOP constraints. We incorporate these polyhedral results as cuts in branch-and-cut algorithms that also have specialized preprocessing and branching procedures and primal heuristics.

We are using these ideas to develop branch-and-cut algorithms for linear programs with piecewise linear nonconvex objectives, complementarity problems such as nonconvex unconstrained quadratic programs, and some other NP-hard problems for which an MIP approach has not been very successful.

We will begin the talk by presenting some models for which this approach appears to be particularly useful and then we will focus on cardinality constrained linear programs. Here our direct approach to NCOPs without binary variables has yielded an algorithm that runs nearly an order of magnitude
faster than the venerable CPLEX system on some classes of problems. After developing the polyhedral
theory, we will present computational results that demonstrate the effectiveness of the algorithm.

Our polyhedral study of cardinality constrained linear programs uses knapsack relaxations to
develop cover and lifted cover inequalities. This approach is similar to the well-known theory of cover
inequalities for binary variables. The basic idea is as follows.

Let $N = \{1, \ldots, n\}$, $x_1, \ldots, x_n \geq 0$ be the variables, of which at most $k$ can be positive, and
$\sum_{j \in N} a_j x_j \leq b$ the knapsack constraint. Let $PS$ be the convex hull of the set of feasible solutions of
the resulting $k$-cardinality knapsack problem.

**Definition 1** Let $C$, $N_0$, and $N_1$ be three disjoint subsets of $N$ with $N = C \cup N_0 \cup N_1$ and $|C| = k -
|N_1|$. If $\sum_{j \in C} a_j > b - \sum_{j \in N_1} a_j$, we say that $C$ is a cover for $PS \cap \{x : x_j = 0 \forall j \in N_0\} \cap \{x : x_j = 1 \forall j \in N_1\}$, and that

$$\sum_{j \in C} a_j x_j \leq b - \sum_{j \in N_1} a_j$$

is a cover inequality for $PS \cap \{x : x_j = 0 \forall j \in N_0\} \cap \{x : x_j = 1 \forall j \in N_1\}$.

We then have,

**Proposition 1** Inequality (1) is valid and facet-defining for $PS \cap \{x : x_j = 0 \forall j \in N_0\} \cap \{x : x_j = 1 \forall j \in N_1\}$.

The cover inequalities introduced above are valid for the set of feasible solutions of the LP-
relaxation, and therefore cannot be used as cuts. However, they can be lifted to derive inequalities
that can be used as cuts.

**Example 1** Let $n = 5$, $a_1 = 8$, $a_2 = 6$, $a_3 = 5$, $a_4 = 4$, $a_5 = 1$, $b = 16$, and $k = 3$. Take $C =
\{1, 2, 3\}$, $N_0 = \{4, 5\}$, and $N_1 = \emptyset$. Then,

$$8x_1 + 6x_2 + 5x_3 \leq 16$$

is a cover inequality. We now lift (2) with respect to $x_5$. Because $k = 3$, when $x_5$ is positive, at
least one of $x_1, x_2$, or $x_3$ must be 0. Therefore, the left-hand-side of (2) is at most 14, and the lifting
coefficient is 2. Thus, the inequality

$$8x_1 + 6x_2 + 5x_3 + 2x_5 \leq 16$$

is valid. Inequality (3) cuts off, for example, the point $\tilde{x}$, given by $\tilde{x}_1 = \tilde{x}_2 = \tilde{x}_5 = 1$, $\tilde{x}_3 = \frac{1}{5}$, $\tilde{x}_4 = 0$,
which is a basic feasible solution of the LP-relaxation.

By lifting cover inequalities in a specific order, we have derived the following family of facet-
defining inequalities valid for the convex hull of the set of feasible solutions of knapsack problems with
$k$-cardinality constraints.
Theorem 2 Suppose \( a_1 \geq \cdots \geq a_n \), \( \sum_{j=n-k+1}^{n} a_j < b \), \( \sum_{j=n-k}^{n-1} a_j \leq b \), and \( a_1 + \sum_{j=n-k+2}^{n} a_j > b \). Then,

\[
a_1 x_1 + \sum_{j=2}^{n-k-1} \max\{a_j, b - \sum_{i=n-k+2}^{n} a_i\} x_j + (b - \sum_{i=n-k+2}^{n} a_i) \sum_{j=n-k}^{n} x_j \leq k (b - \sum_{j=n-k+2}^{n} a_j)
\]

defines a facet of \( P_S \).

These inequalities are used in our branch-and-cut algorithm.

Bibliography


Graph Layering by Promotion of Nodes

Nikola S. Nikolov¹ and Alexandre Tarassov²

¹CSIS Department, University of Limerick, Ireland
E-mail: {nikola.nikolov,alexandre.tarassov}@ul.ie

We present a new improvement heuristic for decreasing the sum of edge spans in a layered directed acyclic graph (DAG). Formally, layer a DAG \( G = (V, E) \) to partition its node set \( V \) into \( h \) subsets \( V_1, V_2, \ldots, V_h \), called layers, such that if \( (u, v) \in E, u \in V_i, \) and \( v \in V_j \) then \( i > j \). The height of a layering is the number of layers, \( h \); the width of a layering is \( \max \{|V_i| : 1 \leq i \leq h\} \); and the span of edge \( (u, v) \in E \) with \( u \in V_i \) and \( v \in V_j \) is \( i - j \). We assume that when \( G \) is drawn in a hierarchical manner the layers are drawn on parallel horizontal levels by placing \( V_1 \) at the bottom level, \( V_2 \) at the level above it, and so on.

The problem of finding a layering of a DAG is known to be NP-hard when both the height and the width of the layering are bounded from above. This is the central problem in scheduling causally related tasks for execution on a limited number of parallel processors [1, 4]. It is also a problem to be solved at one of the phases of the algorithmic framework of Sugiyama et al. [5, 2], which is probably the most popular heuristic for automated drawing of directed graphs. From the graph drawing point of view a layering is expected to satisfy a number of additional aesthetic requirements. These usually include upper bounds on the dimensions of the layering, minimum sum of edge spans and even edge density between adjacent layers.

A recent empirical study [3] of the behavior of various layering algorithms as drawing algorithms has shown that the superior among them is the algorithm of Gansner et al., which involves solving a linear programming model for minimizing the sum of edge spans without imposing any bounds on the layering dimensions. The layering problem solved by the algorithm of Gansner et al. is in \( P \), however, there is no known algorithms to solve it without employing a linear programming solver. Layerings with minimum sum of edge spans are on average as compact as layerings produced by algorithms which impose upper bounds on the layering dimensions. Thus, they might be used for efficient multiprocessor scheduling.

The new heuristic, Promote Layering, which we present here is a polynomial-time algorithm that decreases the sum of edge spans in an already layered DAG. Promote Layering does not minimize the sum of edge spans, but when applied after the longest-path layering algorithm it leads to a sum of edge spans which is very close to the minimum. The longest-path layering algorithm is a linear-time algorithm which constructs layerings of height equal to the number of nodes in the longest directed path in a DAG. Promote Layering consists of two main parts:

(i) A recursive function PromoteNode for promoting a single node from layer \( V_k \) to layer \( V_{k+1} \), which returns the difference between the sum of edge spans before and after the promotion; and

(ii) A loop over all the nodes (in no particular order) of a layered DAG in which each node gets promoted once if that reduces the total sum of edge spans. The loop goes on until there is no node whose promotion reduces the total sum of edge spans.

The worst-case time complexity of PromoteNode is \( O(|E|) \), and we have estimated that the worst-case time complexity of Promote Layering in total is \( O(S|E||V|) \), where \( S \) is the sum of edge spans in the initial layered DAG. However, the tests we have run on more than 8000 DAGs show that the
running time of *Promote Layering* applied after the longest-path layering algorithm is very close to the running time of the algorithm of Gansner et al. Our empirical study has also shown that layerings computed by *Promote Layering* applied after the longest-path layering algorithm are more compact and have lower edge density than layerings with minimum sum of edge spans.

Thus, we conclude that *Promote Layering* applied after the fast and simple longest-path layering algorithm is a powerful alternative to the linear programming algorithm of Gansner et al. from the graph drawing point of view and possibly it can be also applied for efficient multiprocessor scheduling.

References


Ordering Graphs and Vulnerability of Networks

CARLA S. OLIVEIRA¹, NAIR MARIA M. DE ABREU², and ADEMIR F. PAZOTO³

¹Escola Nacional de Ciências Estatísticas e Programa de Engenharia de Produção, COPPE/UFRJ, Rio de Janeiro, Brazil
E-mail: carlas@uol.com.br

²Programa de Engenharia de Produção, COPPE/UFRJ, Rio de Janeiro, Brazil
E-mail: nair@pop.ufrj.br

³Instituto de Matemática, UFRJ, Rio de Janeiro, Brazil
E-mail: ademir@adm.ufrj.br

The concept of vulnerability arises in the study of a network N(G) modelled by a graph G. By vulnerability we mean the susceptibility of the network to attack. It may be considered destroyed if after removal of some vertices or edges the resulting graph is not connected. Thus vulnerability of a network is related to vertex, edge and algebraic connectivities [2], [3], [4], [8] and [9]. Vertex (or edge) connectivity of G, \( \kappa_v(G)(\kappa_e(v)) \), is the minimum number of vertices (or edges) whose removal from G results in a disconnected graph or a trivial one. The algebraic connectivity, a spectral parameter, is given in function of Laplacian of G, \( L(G) = \Delta - A \), where \( A \) is the adjacency matrix of G and \( \Delta \) is a diagonal matrix whose diagonal elements are the vertex degrees of G. The characteristic polynomial of Laplacian is \( p_{L(G)}(\lambda) = \text{det}(L(G) - \lambda I) \) and \( \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n \) are its eigenvalue in increasing order. According to Godsil and Royle, [5], Fiedler called \( \lambda_2 \) the algebraic connectivity of G and, in this work, we denote \( \lambda_2 = \kappa_v(G) \).

A network \( N(G_1) \) is considered more vulnerable than \( N(G_2) \) if the vertex connectivity of \( G_1 \) is less than that of \( G_2 \), that is, \( \kappa_v(G_1) < \kappa_v(G_2) \) and \( N(G_1) \) is as vulnerable as \( N(G_2) \) if \( \kappa_v(G_1) = \kappa_v(G_2) \), [1] and [9]. The null algebraic connectivity characterizes disconnected graphs, [2] and [6]. Because of that, we say that \( N(G_1) \) is more algebraic-vulnerable than \( N(G_2) \) if \( \kappa_a(G_1) < \kappa_a(G_2) \) and that \( N(G_1) \) is as algebraic-vulnerable as \( N(G_2) \) if \( \kappa_a(G_1) = \kappa_a(G_2) \). Consider a path \( P_n \) and a bipartite complete graph \( K_{1,n-1} \). We have \( N(P_n) \) as vulnerable as \( N(K_{1,n-1}) \) and, for \( n \geq 4 \), \( N(P_n) \) is more algebraic-vulnerable than \( N(K_{1,n-1}) \).

However, the vulnerability of network cannot be related only by the minimum number of vertices which can be removed from its underlying graph. The quantity of subsets of these vertices can be taken into account, when we want to avoid the network destruction. So, cardinal-connectivity of G, \( \alpha_v(G) \), is the number of subsets of \( V(G) \) with cardinality \( \kappa_v(G) \), which are able to disconnect G, when their vertices are removed from it. Therefore, \( N(G_1) \) is more cardinal-vulnerable than \( N(G_2) \) if \( \alpha_v(G_2) < \alpha_v(G_1) \) and \( N(G_1) \) is as cardinal-vulnerable as \( N(G_2) \) if \( \alpha_v(G_2) = \alpha_v(G_1) \). It is easy to see that \( N(P_n) \) is more cardinal-vulnerable than \( N(K_{1,n-1}) \).

Consider now a network sequence \( (N(G_n))_{n \in \mathbb{N}^*} \), with respect to the underlying graph sequence \( (G_n)_{n \in \mathbb{N}^*} \). For all \( n \in \mathbb{N}^* \), let every vertex connectivity of \( G_n \) be equal to \( \kappa_v(G_n) = c \). If \( (\alpha_v(G_n))_{n \in \mathbb{N}^*} \), the cardinal-connectivity sequence of \( G_n \), is increasing (or decreasing) we say that \( (N(G_n))_{n \in \mathbb{N}^*} \) is an increasing (or decreasing) cardinal-vulnerable network sequence. An analogous definition is considered when we have an algebraic connectivity sequence of \( G_n \).

Finally, we prove that in some classes of graphs, [7], increasing cardinal-vulnerable sequences correspond to decreasing algebraic-vulnerable sequences and reciprocally. This result is not true in
the set of all graphs.

Bibliography


An Algorithm for Traffic Grooming in WDM Optical Networks

ZENILTON K. G. PATROCÍNIO JR. 1 AND GERALDO R. MATEUS 1

1 Computer Science Department, Federal University of Minas Gerais, P.O. Box 702, Belo Horizonte, MG 30123-970, Brazil
E-mails: {zenilton, mateus}@dcc.ufmg.br

Wavelength division multiplexing (WDM) has been pointed out as a powerful tool that allows an enlargement of the transmission capacity over optical links. Speed limitations of electronics can be bypassed using different wavelengths to establish independent channels at the same fiber [1]. Nevertheless, due to a limited number of wavelengths available in current technology, it is not possible to assign an exclusive wavelength for each traffic stream. So in order to attain a feasible and less costly solution one should try to combine low speed traffic streams into high speed ones, e.g., using time division multiplexing (TDM). Moreover this has to be done in such a way to minimize the need and use of electro-optical conversions at the intermediate nodes. Actually there are two main reasons that have been used as guidelines for recent research efforts in this area [2]. First, the cost of network components, specially electro-optical converters (EOCs) and add/drop multiplexers (ADMs), has proved to be a better criterion (on which network topology and routing decisions may be based) rather than the number of allocated wavelengths. The other important issue is related to the fact that bandwidth requirements of independent traffic streams are always smaller than available bandwidth in a single wavelength.

The problem of combining low speed traffic streams into high speed ones has been called Traffic Grooming Problem (TGP) in the literature and addressed by many authors, specially problems involving ring topology [2, 3, 4, 5, 6, 7]. Most works explore the use of heuristic techniques in order to solve this problem. More recently some works have started to deal with different topologies, like meshes [8, 9, 10]. It is also worth to mention that the problem of logical topology design is NP-hard even for a physical ring topology [1]. As a consequence of this, tight upper and lower bound values are very useful not only for evaluating the performance of those heuristic approaches but also for assessing the quality of a specific problem formulation.

In this paper, TGP in WDM optical networks is explored regardless of underlying physical topology. Our main concern is to investigate the optimal solution for this problem despite difficulties of solving it. To do that, TGP was then formulated as an IP (integer program) that captures technological restrictions like wavelength continuity constraint. We also introduce a layered graph representation of the problem which is used to reformulated the original TGP model into a simple and compact one which can be seen as an extension of network loading problem [11, 12] and/or minimum cost capacity installation problem [13]. In order to solve the final model, Lagrangian relaxation was used to obtain lower bounds that are used in a branch-and-bound procedure along with valid cut inequalities that are added to strengthen the formulation and to achieve tight lower bounds. We also develop an heuristic that uses information of lower bound solutions in order to generate "good" upper bounds.

Preliminary results are very promising and have shown so far that our approach can cope very well with standard and "real-life" problem patterns like rings and ring chains. Actually, it was able to attain very small duality gaps, or even to solve optimally many instances of those problems without any need of branching. Despite these first results, an extensive test procedure is now been conducted to assess the real performance of proposed procedure to deal with different topologies, demand patterns and sizes.
Bibliography


Probabilistic and Optimization Versions of Propositional Satisfiability

DANIELE PRETOLANI

1 Dipartimento di Matematica e Informatica, Università di Camerino, via Madonna delle Carceri, Camerino, I-62032, Italy
E-mail: daniele.pretolani@unicam.it

Both the probabilistic satisfiability problem (PSAT) and the check of coherence of probabilistic assessment (CPA) [2] can be considered as probabilistic counterparts of the classical propositional satisfiability problem (SAT). Despite their similarities (see e.g. the unified treatment given by Hansen et al. [5]) the two problems show opposite features. In CPA, probabilities are associated with variables, while clauses define logical relations between events, and have no associated uncertainty. In PSAT, probabilities are associated with clauses, thus logical relations between variables hold with uncertainty.

Interestingly, a quite similar relation holds between two optimization counterparts of SAT. In the Min SAT problem, each variable is assigned a weight, and the goal is to minimize the sum of the weights of the variables set to true in a satisfying assignment. In the well known Max SAT problem, weights are associated with clauses, and it is assumed that some clauses may not be satisfied.

Our aim here is to exploit the relations between probabilistic and optimization counterparts of SAT; in particular, we apply Combinatorial Optimization techniques and results to deal with some computational complexity issues.

It is well known that the classical linear programming formulation of PSAT can be dealt with by column generation techniques [4], where the pricing problem is Max SAT. To begin with, we present a simplified LP formulation for CPA, leading to a Min SAT pricing problem.

Then we consider the generalized set covering formulation of PSAT proposed in [1]. We devise a simplified formulation for CPA, and we show that CPA for ‘ideal formulas’ [1] can be solved in linear time. Note that PSAT can be solved in linear time [1] only for a fragment of ‘ideal formulas’.

Finally we prove that Max SAT with unrestricted weights is NP-hard for the class of graph formulas (another fragment of ‘ideal formulas’) where Min SAT is polynomially solvable. In light of a well known result in polyhedral combinatorics we conclude that PSAT for ideal formulas is NP-complete.

Notably, CPA is at the core of probabilistic inference methods for conditional events (see e.g. [3]). This motivates research on effective algorithms, that recently led to the development of Davis-Putnam-like procedures [2]. The extension of this approach to PSAT is currently under investigation, and shall be briefly discussed here, in comparison to other techniques such as column generation and pseudo-Boolean optimization.

Bibliography


Eliminating Redundant Solutions of Combinatorial Integer Programs by Exploiting Symmetry

PABLO A. REY\textsuperscript{1} AND OSCAR PORTO\textsuperscript{2}

\textsuperscript{1}Departamento de Ingeniería Industrial, Universidad de Chile, República 701, Santiago, Chile
E-mail: prey@dii.uchile.cl

\textsuperscript{2}Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, Rua Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
E-mail: oscar@ele.puc-rio.br

This work presents a way to eliminate redundant solutions (due to geometrical symmetries) of linear integer programs by adding appropriate inequalities to the formulation. In the examples considered, this approach leads to an impressive reduction of the search tree of enumerative algorithms.

1 Introduction

One important drawback in solving linear integer programs (IP) is the presence of “symmetry.” In the seventies Jeroslow noted that some IP, even small ones, were difficult to solve by enumerative methods [Jer74]. He attributed this difficulty to the indistinguishability of the variables. Many problems share this pattern. Most of them come from IP formulations of combinatorial problems.

This work presents a theoretical framework within which redundant feasible solutions can be eliminated by exploiting the symmetry of the problem. The technique is applied to several examples.

It is worth to mention that this kind of considerations are very sparse in the literature of IP. To the best of our knowledge the unique attempt to tackle this difficulty by means of introducing inequalities in the formulation is the recent paper [SS]. Considering symmetry in the search phase in IP context it has been considered just recently, see [Mar01, Mou98].

2 Indistinguishable objects and permutations

In order to formalize our approach consider the IP

$$\min \left\{ \sum_{i=1}^{n} x_i : Ax \geq b, x \in \{0, 1\}^n \right\}$$

(1)

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$. Denote by $P_I$ the convex hull of the feasible set of (1). Suppose further, that in (1) the variables $x$ are indistinguishable. The following holds:

**Proposition 2** Let $n \geq 2$ be an integer number and define

$$F := \{ x \in \mathbb{R}^n : x_i \leq x_{i+1}, \ i = 1, \ldots, n-1 \}.$$  

Consider the programs $\min \{ e^\top x : x \in P_I \}$ (*) and $\min \{ e^\top x : x \in P_I \cap F \}$ (**), where $P_I$ is convex hull of the feasible set of (1). Then,
(a) any feasible solution of (*) can be obtained from a feasible solution of (**) by permuting the values of the variables;  
(b) two distinct feasible solutions of (**) can be obtained one from the other by permuting the values of its variables only if they lay on the boundary of $P_f \cap F$.

Often, variables of combinatorial integer programs have more than one index. Sometimes, for a subset of the indices, variables resulting from permuting indices in that particular subset are indistinguishable. entities represented by some of the indices are indistinguishable. In this work is also presented the result similar to Proposition 2 for the multiple indices case.

2.1 Groups and symmetry information

The general pattern of symmetry correspond to a set of feasible permutations on the indices of the variables (or the variables themselves). It is not difficult to see that these permutations must form a group. With the equivalence relation defined by that permutations a set $F$ (like in Proposition 2) can be constructed. Then, the inequalities defining $F$ can be added to the formulation of the problem cutting off redundant solutions.

We also consider as a case study the well-known class of set covering problems that arises from Steiner triple systems (STS) introduced by Fulkerson, Nemhauser, and Trotter in [FNLT74].

Bibliography


Components Cutting Optimisation by Memetic Algorithms

ANA MARIA RODRIGUES\textsuperscript{2}, LUI\textsuperscript{S} MIGUEL MOREIRA\textsuperscript{2}, AND JOSÉ SOEIRO FERREIRA\textsuperscript{1,2}

\textsuperscript{1} Faculdade de Engenharia, Universidade de Porto, Porto, Portugal
\textsuperscript{2} Instituto de Engenharia de Sistemas e Computadores do Porto, Porto, Portugal
E-mail: jsoeiro@inescporto.pt

A class of complex Path Optimisation problems rising in the manufacturing of irregular components for woodworking tools is considered. The discussion is essentially based on an important Portuguese industry of that sector.

Components of small size and of irregular shapes must be cut from circular (expensive) plates made of tungsten with a thin diamond layer, using an electrified copper string-cutting tool. This procedure must be carried out in a continuous way, the cutting tool never leaves the cutting surface, and it is particularly important to minimise the cutting path/time. Other complicate restrictions must be taken into account. Moreover, there is an obvious relation to: nesting (to determine a good layout) and visibility checking (to create bridges between the components), which must be considered before the Path Optimisation problems. Nesting uses the methods of Oliveira et al [7], and Gomes et al [3]. The solution of the inter-visibility checking problem is described in Galvão and Ferreira [2], and Moreira [6].

Which ways to follow to cut all the components, now that they are all connected? These optimisation problems are modelled as a Rural Postman Problem (RPP). The RPP is NP-hard. Mathematical programming techniques could have been considered (see Eiselt et al [1]), but it has been decided to resort to a more flexible approach - Memetic Algorithms (MA). The authors are not aware of any applications of MA to the RPP.

MA may be seen as Genetic Algorithms with a local search operator to aggregate memetic information - after recombination and mutation, local search is applied to the resulting solutions (Moscato [4]). The MA method to solve the RPP, includes a population of 13 agents, with initial solutions, as a ternary tree of three levels, i.e., subpopulations of four individuals each, where each node represents one agent, thus: each subpopulation is composed by one "leader" node and three supporters (follower). The leader of a subpopulation (or cluster) is always better fitted than its supporters.

At any moment, each agent of the optimising population will be handling two tours. They are: Pocket Tour - the best tour found or received by that agent; Current Tour - the one actually being optimised, by the heuristic assigned to that agent.

The best tour of each agent is updated each time a new "current" tour, of a smaller length, is obtained (see Moscato [5]). To preserve the organized structure of the tree and to avoid the loss of diversity, after a series of recombinations and local search steps, crucial specific steps are followed.

Computational results illustrate the application of MA to 'simple' cases (RPP's) taken from the literature and to real industrial applications. Results obtained by Monte Carlo Methods, Galvão and Ferreira [2], and Constructive algorithms, Moreira [6] are also included. The authors may conclude, based on the industrial validation, that the integrated system (nesting, visibility checking and path optimisation) is effective in the factory environment. And that was a main objective of this work.
Bibliography


Bicriteria Robustness versus Cost Optimisation in the Generation of Aircrew Pairings

DAVID M. RYAN$^1$ AND MATTHIAS EHRIOTT$^2$

$^1$ Department of Engineering Science, University of Auckland, Auckland, New Zealand

Optimisation based computer systems are used by many airlines to solve crew planning or pairings problems by constructing minimal cost tours of duty. These systems are almost always based on the set partitioning or set covering combinatorial optimization models. However, airlines do not only require cost effective solutions, but they are also very interested in robust solutions. A more robust solution is understood to be one in which disruptions in the flight schedule (due to delays) are less likely to be propagated into the future, causing the delay of subsequent flights. Current scheduling systems based solely on cost do not automatically provide robust solutions. In fact minimum cost solutions tend not to be particularly robust.

These considerations lead to a multiobjective framework, as the maximisation of robustness will be in conflict with the minimisation of cost. We propose a possible robustness measure for a tour of duty and show how to develop a linear robustness objective function. We then present a bicriteria optimisation formulation of the problem. Solutions of the bicriteria problem are called Pareto optimal. A Pareto optimal solution is one which does not allow an improvement in cost and robustness at the same time.

We discuss several approaches for solving the bicriteria model and show that a weighted sum scalarization is not appropriate. We also show that treating the cost objective as a constraint while maximizing robustness leads to very difficult integer programming problems. This situation can be overcome by treating the cost objective as an elastic constraint and penalizing violations of the constraint in the robustness objective.

We have tested our model and solution method with actual airline data for both technical and cabin crew. Our results show that a considerable gain in robustness can be achieved with a small increase in cost. A feature of robust solutions is that aircraft change within a duty period is discouraged if inadequate ground time is provided to remove the threat of disruption due to a late arriving inward flight. The additional cost of robust solutions is mainly due to a small increase in overnights, which allows for a reduction in the number of aircraft changes.
Simulated Annealing Approach to Solve the Daily Crew Scheduling Problem

Gustavo P. Silva¹, Marcone J. F. Souza¹, and José Maria do C. B. Alves¹

¹Federal University of Ouro Preto, Ouro Preto, MG 35420-000, Brazil
E-mails: gpsilva@em.ufop.br, marcone@iceb.ufop.br, jmalves@uai.com.br

1 Introduction

The Crew Scheduling Problem (CSP) of a public transportation system consists of determining the minimum number of crews in order to achieve a successful vehicles' programming. The solution of this problem also involves the sequencing of the crews' activities and the generation of a set of work days with minimum operational cost. This problem becomes particularly complex due to its magnitude, operational restrictions, and labor laws. In addition, the conditions of the transport systems are in continuous transformation and they demand increasingly an efficient management of the available resources [5], [6]. The development of the metaheuristic Genetic Algorithms, Tabu Search, Simulated Annealing and others [3] have offered new possibilities for the resolution of NP-hard problems like CSP. Although such methods do not guarantee the obtaining of the optimum global they allow to easily include any restriction. Among the works that have applied metaheuristic to solve CSP problems the followings must be cited: Wren and Wren [7], Kwan et al. [2] and Shen and Kwan [4]. Although CSP has widely been studied, its main restrictions are related to the compliance of labor laws, and the effective operational norms of companies. Such factors do not allow that models, which were successful developed in other countries, be applied in Brazil. This work proposes a heuristic procedure to solve CSP based on the metaheuristic Simulated Annealing and taking into account the operational restrictions of the Brazilian reality. A study case of a public transportation company operating in the city of Belo Horizonte, southeastern Brazil, is presented.

2 Daily Crew Scheduling

The crew's scheduling is usually done after the vehicles' programming, which contains Relief Opportunities (RO). The RO's are time intervals, which are necessary for the shift of two crews. Each RO is associated to the Relief Time (RT) and the Relief Point (RP), which represent the hour and the place, respectively, where the change can occur. A Task is a set of trips between two successive RO. The daily crew scheduling is formed by a set of tasks also called workload. The Brazilian workdays can be a Straight or a Split. In the first type the tasks are accomplished with interruption for the meal. If the meal interval is larger than two hours, the workday is of the Split type. In this case, that interval is not paid. When defining workdays the following restriction should be taken into account: a) the crew's change can occur at preset points, only; b) the crew can operate vehicles from the same group of lines, only; c) the workdays duration is 7 hours and 10 minutes, and a maximum of 2-hour overtime can be added; d) a crew that does not follow the split system is entitled to a 30-minute rest during the workday. This 30-minute period can be divided into smaller intervals if one of them is larger or equal to 15 minutes. Restrictions of this type turn real problems untreatable by computational means and justify the application of metaheuristic to their resolution.
3 Simulated Annealing Application to Daily Crew Scheduling

Simulated Annealing [3] is a method of local search that accepts worsening movements to avoid becoming trapped early in a local optimum. This method can be characterized as a search form in a neighborhood, where each \( s \in S \) solution has an associated set of neighbors, \( N(s) \subseteq S \), which is called \( s \) neighborhood. Any \( s' \in N(s) \) solution can be reached directly from \( s \), starting from a movement, which is probabilistically accepted as a function of the temperature parameter. The developed algorithm works with the movement Insert Task, which attributes a task from an \( i \) crew to a \( j \) crew. In order to generate such a movement any two crews \( i \) and \( j \) are drawn obeying the condition that the \( i \) crew must have at least one task. Next, a task of the \( i \) crew is raffled, on which the movement is done. The objective function to be minimized, which evaluates the crew scheduling, is a penalty function that has the following goals: a) to eliminate unfeasibility situations such as task superposition, insufficient meal interval, etc., b) to reduce costs that are related to the number of crews, overtime, etc., and c) to maximize the use of labor by minimizing crew idleness.

4 Computational Results

The algorithm was successful tested using data from a Belo Horizonte public transportation company. Table 1 display the characteristics of the best solution. This solution was obtained from several tests, which used different seeds of random numbers. The algorithm was run in an Intel Celeron (400 MHz clock, 64 MB RAM memory) microcomputer.

<table>
<thead>
<tr>
<th></th>
<th># of Crews</th>
<th>Idleness (hh:mm)</th>
<th>Line Changes</th>
<th>Split (hh:mm)</th>
<th>Overtime (hh:mm)</th>
<th>CPU time (hh:mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current solution</td>
<td>218</td>
<td>188:22</td>
<td>2</td>
<td>21</td>
<td>116:00</td>
<td>-</td>
</tr>
<tr>
<td>Best solution</td>
<td>206</td>
<td>113:54</td>
<td>59</td>
<td>67</td>
<td>133:47</td>
<td>07:30</td>
</tr>
</tbody>
</table>

Table 1 shows a 5.5% reduction in relation to the current number of crews (in the Current solution). This is the most important factor affecting the objective function. The idleness reduction indicates a better labor use. This fact can be explained by the increase of crews that are responsible for line changes. Line changes allow a flexibility and, as a consequence, a reduction of operational costs. These can only occur in Split situations, the number of this workday's type was increased. In this case, the idleness reduction does not generate enough time for the execution of all tasks, demanding, as a consequence, an overtime increase. A cost analysis must be required because the increase of the number of crews must provoke an overtime reduction. The proposed algorithm does not deal with monthly scheduling. This will be the focus of future research.

5 Conclusions

This work presents an optimization heuristic model based on the metaheuristic Simulated Annealing for the CSP resolution of a public transportation company in Belo Horizonte, Minas Gerais, Brazil. Despite the inherent difficulties of the CSP approach, the obtained results showed significant operational labor cost reductions of the daily scheduling. Such a result emphasizes the metaheuristic Simulated Annealing potential and incentives the method application to generate monthly crew scheduling.
Bibliography


Double Contraction, Double Probing, Short Starts for Mixed
(0,1) Integer Programming

KURT SPIELBERG\(^1\) AND MONIQUE GUIGNARD\(^2\)

\(^1\)2005 Market Stret, Philadelphia, PA 19103, United States
\(^2\)Department of OPIM, Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6366,
United States
E-mail: guignard@wharton.upenn.edu

We describe three new tools for potential use in the solution of difficult and large Mixed (0,1)
Integer Programming problems. They are related to early work of the authors in Logical Processing
(Probing and alike).

1 Double Contraction

Double Contraction (DC) goes back to the creation of Minimal Preferred Inequalities (minimal covers)
of smallest degree, mostly of degree 2. A variable is double-contracting if setting it to 0 or 1 reduces
the feasibility range of some other variable(s). DC characterizes the constraint set of a (0,1) MIP
model as having strong interactions (to be exploited).

By associating the best DC columns with highest priorities in a BB production run, one can obtain
striking improvements in performance. We can point to two classes of problems for which we have
(in cooperation with Ph. D. students) attained improvement factors of up to thousand in number of
nodes and/or time; in two other cases the improvements were smaller but still very good.

Selecting such models did not prove difficult; we had a high ratio of success. It seems to us that
this is one of the most promising features which should be built into modern (0,1) MIP solvers.

2 Double Probing

In normal probing one “sets” one variable and can often fix variables by following its consequences.
We have tested the possibility of setting two variables at a time (guided by patterns in a DC matrix).
It is stronger than probing, but one will have to test whether the extra work involved is justified by
the results. This is still an open question.

3 Short Starts

We have a procedure for using industrial solvers to construct BB tree fragments of “ones” and “zeros”
which represent a given solution almost minimally. From a Short Start (SS) we can then search for
new solutions more efficiently than normally. We have, for one large problem, improvement factors of
about 20. We typically can represent solutions of a problem with 1803 (0,1) variables by a SS with
some 35 branches only.
4 References


A Penalty-Evaporation Heuristic in a Decomposition Method for the Maximum Clique Problem

PATRICK ST-LOUIS, JACQUES A. FERLAND, AND BERNARD GENDRON

In this paper, we introduce two approaches to solve the maximum unweighted clique problem. The first is a general decomposition method restricting the search for a maximum clique to subgraphs, but also performing an exhaustive search of the feasible domain. Any algorithm, exact or heuristic, that provides feasible cliques can be used to search in the subgraphs. If the embedded algorithm is exact, then the decomposition algorithm provides exact solution. Otherwise, if the embedded algorithm is a heuristic method, then the decomposition algorithm improves significantly the quality of the solution found by using the heuristic method alone. The second approach is a greedy-type heuristic method based on the concept of penalty and evaporation, which identifies a large clique in a graph requiring very short computing time. Numerical results indicate that the heuristic method alone is very efficient and reliable, and the gain in the quality of the solutions obtained by embedding it into the decomposition algorithm is worthy of the additional computing time required.
Combinatorial Problems in Spatial Forest Planning

Andres Weintraub

1 University of Chile, Department of Industrial Engineering, República 701, Santiago, Chile

The use of models to represent forest problems has been in use successfully for several decades now. The first models where linear models dealing basically with decisions related to harvesting and replanting. Forests can be divided into units called stands which can be considered approximately homogeneous in terms of species planted, quality of soil, tree age and other silvicultural characteristics. It was vital to be able to define these homogeneous areas which could be considered as black boxes. Linear models typically indicate how many hectares of each stand should be harvested in each period. In the last two decades there has been an increased importance given to environmental issues, in particular wildlife, vegetation and water protection, as well as scenic beauty. Different models have been formulated to capture, at least approximately, these issues. Environmental problems lead inevitably to spatially related formulations and thus to often complex combinatorial problems.

The first formulations, in the 80's, called the adjacency problem, was based on the notion that stands would be subdivided into cutting units, of up to 40 hectares. These cutting units can be harvested in any period completely or not at all, leading thus to 0-1 formulations. The adjacency constraints state that in each period if one cutting unit is harvested, none of the neighboring cutting units can be harvested in the same period (or maybe in the next one or two periods also). The idea behind this restriction is to protect mainly wildlife. Several species will not dare go out and feed on a clearing provided by a recently harvested stand unless they are relatively close to the protection given by grown trees. Other motives for this type of constraint are scenic beauty (so that recreationers don't have a view of a large bare area), control of erosion. To solve this problem several approaches have been proposed. Heuristics include Monte Carlo simulations, Tabu search, simulated annealing. Exact formulations, albeit solved approximately in some cases include column generation (where each subproblem is a stable set), dynamic programming (limited to smaller problems), strengthening of the LP formulation. Adjacency constraints are now a standard use in Europe, North America, New Zealand and increasingly in South America. As a mathematical problem, in most cases heuristics are used, but the exact formulation can reasonably deal with this problem.

Experts have seen that in order to represent better the environmental questions, the simple adjacency requirement is not enough, so more complex 'proxi' representations have been formulated. In this presentation we will show how to solve more complex representations of these problems.

The above discussed cutting units are formed by planners using GIS and are based on basic cells (or the minimal indivisible area) which are likely to be between 5 to 10 hectares. Planners group these cells into cutting units using their knowledge. It has been recently noted that by forming the cutting units a priori, and inputting these blocks into models, a significant suboptimality is introduced. So, new models are being proposed in which the forming of blocks or cutting units is integrated into the problem. While leading to better solutions, these formulations are far more complex combinatorially, and so far have only been solved using heuristics.

In addition, another direction of increasingly more complex representation has been the inclusion of additional conditions. These include: considering that between two openings there must be a corridor of grown trees, so that animals can move between the openings, consideration of the value of edge, that is, the perimeter between grown trees and openings, which has value for some species. In general these more complex problems have been approached through heuristic approaches even for the
case were cutting units are formed a priori. We discuss as a representative example a model which considers both adjacency as well as the need to have blocks of mature trees (old growth patches) of minimal dimensions, which favor some animal species. This is done in a format of forming within the problem the cutting units based on basic cells that are given in a GIS. We show a Tabu search solution approach for this problem in work carried out with Felipe Caro, Miguel Constantino and Isabel Martins. We present computational results both for representative problems as well as a real one. In this case it can be seen that to obtain 'good' solutions in reasonable CPU time, considerable experimentation needs to be carried out to determine the best definition of neighbors, intensification, diversification, probabilistic choices, allowing for infeasible intermediate solutions.

We next consider an exact formulation of the simple adjacency problem but integrating the forming of the cutting unit blocks based on basic cells. This is work carried out with Marcos Goycoolea, Alan Murray, Francisco Barahona and Rafael Epstein. In essence, this problem corresponds to the adjacency problem, where part of the problem is to form clusters of up to 40 hectares, starting with the basic cells, and so that these clusters, or cutting units satisfy adjacency in the final solution. A direct graph representation considering the basic cells proved to be impossible to solve. A cluster representation proved more amenable, in particular as it allowed through projected constraints, a much stronger formulation. While cliques, odd cycles, and their lifting could be projected, in the implementation cliques alone proved the best approach. We show computational results for both representative problems as well as two real problems. With this approach medium sized problems could be solved well. Extension to multiple periods proved to be particularly limiting for large scale problems.

We finally discuss what seem to be still open problems in this area.
Geometric Steiner Tree Problems

Paweł Winter¹ and Martin Zachariasen¹

¹Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark
E-mails: {pawel, martinz}@diku.dk

1 Introduction

We study the geometric Steiner tree problem (STP): Find a shortest network, called a Steiner minimal tree (SMT) interconnecting a set Z of n terminals in the plane under a given distance function. The Euclidean STP with the $L_2$ distance metric is the most studied geometric STP. It has its roots more than three centuries back in the works of Fermat and Gauss. The rectilinear STP with the $L_1$ distance metric was first considered by Hanan [2] in the 1960s. He proved that this problem reduces to that of determining SMTs in grid graphs (consisting of horizontal and vertical lines through terminals and with Steiner points at their intersections). The interest in the rectilinear STP comes mainly from applications in VLSI design. The uniformly-oriented STP is a relative newcomer to the fascinating family of STPs; the edges are permitted to have a limited number $\lambda$, $\lambda \geq 2$, of equally-spaced orientations. Permissible orientations are defined by straight lines making angles $i\omega$, $i = 0, 1, \ldots, \lambda-1$, with the positive x-axis, where $\omega = \pi/\lambda$. Finding such a tree, called a $\lambda$-SMT (Fig. 1), is in general NP-hard. The rectilinear STP is in fact a special case of the uniformly-oriented STP for $\lambda = 2$. Also the Euclidean STP can be considered as a special case for $\lambda = \infty$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{$\lambda$-SMTs with varying $\lambda$ for the same set of 15 points. Note that the topologies are not identical.}
\end{figure}

The interest in $\lambda$-SMTs, and in 4-SMTs in particular, is motivated by recent developments in VLSI technology. It will soon be possible to manufacture chips with wires having more than two orientations [9]. Note that a $\lambda$-SMT for $\lambda = 4$ can be almost 30% shorter than one with $\lambda = 2$.

The first exact algorithms for the Euclidean Steiner tree problem (see Hwang et al. [3] for references) were based on a straightforward common framework. Subsets of terminals were considered one by
one. For each subset, all its full Steiner trees (where all terminals have degree 1 and all edges meet at 120°) were determined one by one, and the shortest was retained. Several tests were applied to these shortest FSTs in order to prune away those not in any SMT. Surviving FSTs were then concatenated in all possible ways to obtain trees spanning all terminals. The shortest of them was an SMT.

The bottleneck of this approach was the generation of FSTs. Winter [8] suggested (for the Euclidean variant) a departure from the above general framework. He observed that substantial improvements are available if FSTs are generated across various subsets of terminals. Retained FSTs are not necessarily minimal. However, pruning tests are so powerful that only very few FSTs survive. A similar strategy was later applied to the generation of rectilinear FSTs by Zachariasen [10]. Most recently, a similar approach was applied to the uniformly-oriented STP [4]. Spectacular speed-ups have been achieved also in these cases.

Warne [5] noticed that the concatenation of FSTs can be formulated as a problem of finding a minimum spanning tree in a hypergraph with terminals as vertices and subsets spanned by FSTs as hyperedges. He solved this problem using branch-and-cut. Instances of the STP (all variants) with as many as 10,000 terminals can today be solved in a reasonable amount of time.

2 Generation of FSTs

The goal of the generation phase is to enumerate FSTs included in at least one λ-SMT. The strength of the approach stems from the fact that most FSTs can be discarded; they can be shown to be in no λ-SMT.

Before discussing how FSTs are generated, we need to introduce the notion of a half FST. It is a tree with one “dangling” extension leg. The extension leg is a ray extending from a root vertex of the half FST. The size of a half FST is the number of terminals spanned by it. The smallest half FST is a terminal with an extension leg. In this case, the terminal also acts as the root of the half FST. There are 2λ different half FSTs with a given terminal as the root. Note that the number of half FSTs with one terminal for λ = ∞ is infinite.

A half FST T of size k, k ≥ 2, is obtained by combining a half FST T₁ of size j with a half FST T₂ of size k − j, j = 1, ..., k − 1. Most combinations are infeasible as the extension legs L₁ of T₁ and L₂ of T₂ do not meet at appropriate angles. More specifically, let α denote the smaller of two angles formed at the intersection point r of L₁ and L₂ (Figure 2). This r will act as the root of T. Assume that the extension leg L of T makes angles β and γ with L₁ and L₂. For λ = ∞, we have α = β = γ = \(\frac{2\pi}{3}\). For λ ≡ 0 (mod 3), α, β, γ must be \((2\lambda/3 - i)\pi\), i = -1, 0, 1. For all other finite values of λ, the angles α, β, γ must be \([2\lambda/3]\pi\) or \([2\lambda/3]\pi\) (see [1]).

The generation of FSTs is carried out along with the generation of half FSTs. A pair of half FSTs forms a valid FST if their extension legs overlap (from opposite directions) or if they meet at the \(\pi - \omega\) angle (for finite values of λ). Furthermore, every FST either is a union of two half FSTs or can be transformed by length-preserving transformations to such a union [1]. The reader is referred to [6] for more details.

3 FST Concatenation

In contrast to the FST generation, the FST concatenation is purely combinatorial and essentially metric-independent. Given a set of FSTs \(\mathcal{F} = \{F₁, F₂, ..., Fₘ\}\), known to contain a subset whose union is an SMT for Z, the problem is to identify a subset \(\mathcal{F}^* \subseteq \mathcal{F}\) such that the FSTs in \(\mathcal{F}^*\) interconnect Z and have minimum total length. This problem can be formulated as that of finding
Figure 2: Two half FSTs $T_1$ and $T_2$ which are combined into a new tree $T$. The dashed arrow is a possible extension of $T$. 

a (constrained) minimum spanning tree (MST) in a hypergraph. This problem is NP-hard. But it can be formulated as the following integer programming problem. Let $c$ be a vector in $\mathbb{R}^m$ whose components are $c_i = |E_i|$. Denote by $x$ an $m$-dimensional binary vector.

\[
\begin{align*}
\min c x \\
\text{s.t.} \quad \sum_{E_i \in F} (|Z_i| - 1)x_i &= n - 1 \quad (1) \\
\sum_{E_i \in F} \max(0, |Z_i \cap S| - 1)x_i &\leq |S| - 1, \quad \forall S \subseteq Z, \ 2 \leq |S| < n \quad (2)
\end{align*}
\]

The objective (1) is to minimize the total length of selected FSTs subject to the following constraints: Constraint (2) enforces the correct number and cardinality of hyperedges to construct a spanning tree. Constraints (3) eliminate cycles by extending the standard notion of subtour elimination constraints. These constraints also ensure, in conjunction with equation (2), that the chosen FSTs interconnect $Z$.

This IP is solved by branch-and-cut. Lower bounds are provided by linear programming (LP) relaxation, i.e., by relaxing integrality of every component $x_i$ of $x$ to $0 \leq x_i \leq 1$. The reader is referred to [7] for details.

4 Computational Results

We summarize the computational results for the exact algorithms on the OR-Library instances. We have also performed experiments on VLSI design instances and other instances; these results are presented in [4].

All experiments were made on a 1.4 GHz Pentium III (Linux) machine with 1 Gb of memory. The running times of the exact algorithm are presented in Table 1. These running times include FST concatenation, but this is only a small fraction of the total running time for this problem size range. The number of generated FSTs is highest for the rectilinear problem and decreases with increasing $\lambda$. Approximately $4n$ FSTs are generated for $\lambda = 2$ and approximately $2n$ for $\lambda = \infty$, where $n$ is the number of terminals.

The average running time growth is reasonable, in particular for small values of $\lambda$, and we are (on average) able to solve 100 terminal problem instances in less than 10 seconds. For $\lambda = 2$, $\lambda = 4$ and $\lambda = \infty$ we also solved the 100000 terminal instance from the OR-Library. However, this required substantial computing effort, in particular for the rectilinear problem.
Bibliography


<table>
<thead>
<tr>
<th>Size</th>
<th>$\lambda = 2$</th>
<th>$\lambda = 3$</th>
<th>$\lambda = 4$</th>
<th>$\lambda = 8$</th>
<th>$\lambda = \infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sec</td>
<td>Impr</td>
<td>Sec</td>
<td>Impr</td>
<td>Sec</td>
</tr>
<tr>
<td>10</td>
<td>0.005</td>
<td>10.656</td>
<td>0.006</td>
<td>4.680</td>
<td>0.008</td>
</tr>
<tr>
<td>20</td>
<td>0.021</td>
<td>11.798</td>
<td>0.029</td>
<td>3.744</td>
<td>0.043</td>
</tr>
<tr>
<td>30</td>
<td>0.044</td>
<td>11.552</td>
<td>0.111</td>
<td>4.296</td>
<td>0.115</td>
</tr>
<tr>
<td>40</td>
<td>0.069</td>
<td>10.913</td>
<td>0.277</td>
<td>4.599</td>
<td>0.273</td>
</tr>
<tr>
<td>50</td>
<td>0.157</td>
<td>10.867</td>
<td>0.309</td>
<td>4.053</td>
<td>0.400</td>
</tr>
<tr>
<td>60</td>
<td>0.147</td>
<td>11.862</td>
<td>0.599</td>
<td>4.221</td>
<td>0.660</td>
</tr>
<tr>
<td>70</td>
<td>0.154</td>
<td>11.387</td>
<td>0.881</td>
<td>4.267</td>
<td>0.884</td>
</tr>
<tr>
<td>80</td>
<td>0.237</td>
<td>11.301</td>
<td>1.545</td>
<td>4.092</td>
<td>1.385</td>
</tr>
<tr>
<td>90</td>
<td>0.293</td>
<td>11.457</td>
<td>1.922</td>
<td>3.833</td>
<td>1.459</td>
</tr>
<tr>
<td>100</td>
<td>0.366</td>
<td>11.720</td>
<td>2.363</td>
<td>4.685</td>
<td>2.301</td>
</tr>
</tbody>
</table>

Table 1: Running times in seconds and average improvements compared to MST in percent (OR-library instances). Each entry is an average of 15 instances. GeoSteiner [7] was used for $\lambda = 2$ and $\lambda = \infty$. 

Pucón, Chile, November 4-6, 2002
Approximation Schemes for a Class-Constrained Knapsack Problem

Eduardo C. Xavier and Flávio K. Miyazawa

Instituto de Computação, Universidade Estadual de Campinas, Caixa Postal 6176, Campinas, SP 13084-971, Brazil
E-mails: {eduardo.xavier, fkm}@ic.unicamp.br

1. Introduction. We consider the following generalization of the knapsack problem, which we denote by Generalized Knapsack (G-Knapsack): Given an integer $K$, a set of items $S = \{1, \ldots, n\}$, each item $i$ with value $v_i$, size $s_i$ and a class $c_i$, two wall divisions of size $d$, find a set $M \subseteq S$ of maximum total value and a partition of $M$ into compartments $C_1, \ldots, C_k$, each compartment with size $\sum_{c \subseteq C_j} s(c)$ where the following conditions are valid: (i) items in the same compartment have the same class, (ii) two subsequent compartments are separated by a wall, (iii) the total size used for compartments and walls is at most $K$, (iv) each compartment has size at least $d_{\text{min}}$ and at most $d_{\text{max}}$.

This problem has a practical motivation in the problem of roll cutting in iron and steel industry. Ferreira et al. [3] present a cutting problem where a raw material roll must be cut into final rolls grouped by certain properties after two cutting phases. The rolls obtained after the first phase, called primary rolls, are submitted to different processing operations (tensioning, tempering, laminating, hardening etc.) before the second phase cut. Due to technological limitations, primary rolls have a maximum and minimum allowable width and each cut generate a loss in the roll width.

Given a family of algorithms $A$, for any $\varepsilon > 0$, and an instance $I$ for some problem $P$ we denote by $A(\varepsilon)$ the value of the solution returned by $A$ when executed on instance $I$ and by opt($I$) the value of an optimal solution for this instance. We say that $A$ is a polynomial time approximation scheme (PTAS) if for any $\varepsilon > 0$ and any instance $I$ we have $A(\varepsilon) \geq (1-\varepsilon) \text{opt}(I)$. If the algorithm is also polynomial in $1/\varepsilon$ we say that $A$ is a fully polynomial time approximation scheme (FPTAS).

In this paper we present approximation algorithms for two restricted versions of the G-Knapsack problem. The first algorithm is a FPTAS for the problem with the restriction that each class have a limited number $k$ of different item sizes. The second algorithm is a PTAS for the problem with the restriction that $d_{\text{min}} = 0$.

Related Work: The knapsack problem is a well studied problem and a FPTAS is presented by Ibarra and Kim[4]. There are other generalizations like the multiple knapsack problem (MKP), class-constrained multiple knapsack problem (CCMK) and class-constrained bin-packing problem (CCBP). Chekuri and Khanna [1] present a PTAS for MKP and Shachnai and Tamir [5] present PTAS for CCMK and CCBP.

2. A generic algorithm. Consider the following problem, denoted by SMALL: Given an instance of G-Knapsack, and a value $W$, find a solution with value $W$ that has smallest size for each class. First we show how a generic algorithm works given that we know how to solve problem SMALL. For each class $i$ we have an array $v_i$ indexed by all possible values $W$ that a solution can take. Each position stores the smaller solution that gets the corresponding value $W$. Using these arrays for every class, we can compute the optimal solution using the following algorithm. We generate a table $T$ with columns indexed with all possible values of $W$ and rows corresponds to classes. The entry $(i, j)$ of table $T$
contains the smaller packing using items of classes $1, \ldots, i$ that gets the value $j$. The entries of the table can be computed as follows: The first line of the table is a copy of the array $v_1$ of the first class. The other entries are obtained using the following recurrence:

$$(i, j) := \min\{ (i-1, j), v_i(j), \min_{1 \leq k < j} \{ (i-1, k) + v_i(j-k) \} \}$$

Let $m$ be the last line of the table. The optimal solution is given by the biggest value $j$ such that $(m, j) \leq K$. Notice that if we consider all possible values of $W$ we do not have a polynomial time algorithm. By the other side, we can use the same rounding strategy used by Ibarra and Kim [4]. This leads to a polynomial time algorithm with only $\epsilon$ loss on the total value of the solution find by the algorithm.

Both algorithms we present work as above and differ only in the way problem SMALL is solved. In the next two sections, we show how this problem is tackled in each algorithm.

3. The FPTAS. In this section, we consider the G-knapsack problem restricted to $k$ different item sizes in each class. First, consider the problem, denoted by $k$-Pack, of packing $n$ items with at most $k$ different item sizes into the minimum number of bins of size $d_{\max}$, each bin filled by at least $d_{\min}$.

**Lemma 3** Problem $k$-Pack can be solved in polynomial time.

The above algorithm computes a table with all possible entries $(q_1, \ldots, q_k)$, where $q_j$ is the number of items used of size type $j$. Each entry has the maximum value given by an instance with this size specification. The array $v_i$ is computed using the size of the packed items plus $dt$ where $t$ is the minimum number of bins used to pack the corresponding items. Notice the strict relationship between the minimal bin packing of maximal value and the packing in the knapsack of minimal size that have a given value.

**Theorem 4** There is a polynomial time algorithm that computes the entries of the array $v_i$.

4. The PTAS. In this section, we present an algorithm to solve the problem with the restriction that $d_{\min} = 0$. The algorithm uses some nice ideas proposed by Chekuri et al. [1].

For each possible value $W$ we use the following algorithm. First find a polynomial number of item subsets $U \subseteq S$ such that at least one of these sets has a feasible pack and have value very close to $W$. We obtain this set using another number in the item values in such a way that only $h \geq O(\log n)$ different values are used. We partition the set $S$ in sets $S_1, \ldots, S_h$ and find the number of items of each set we will put in $U$. We obtain items by increasing order of size in such a way that we have a feasible packing of them.

**Lemma 5** There is a polynomial time algorithm that generates a polynomial number of subsets $U \subseteq S$, such that at least one is feasible and has value $p(U) \geq (1 - O(\epsilon))W$.

Each of these sets is packed using known algorithms for bin packing like the one presented by Fernandez de la Vega and Lueker [2] which can be modified to use $(1 + \epsilon)OPT + 1$ bins. If $OPT \leq 1/\epsilon + 2$, we can develop a PTAS to pack the items[6]. In the other case we will take the $OPT$ bins with largest profit generated by the algorithm of Lueker and Fernandez de la Vega. In any case we will lose only $O(\epsilon)$ value of the optimal $W$. We will take the set with the smallest size that have value at least $(1 - O(\epsilon))W$. The entry of the array will take as value the size of the items of the set plus $dOPT$ where $OPT$ is the number of bins used.
Theorem 6 In polynomial time we can compute the array $v_i$ in such a way that for each entry $W$ we get a packing of value $(1 - O(e))W$ that have size no more than the optimal packing.

Conclusions. In this paper we present approximation schemes for two restricted versions of the G-Knapsack problem. It would be interesting to obtain an approximation scheme for the general case. Another interesting direction, is to consider a more generalized problem where in each compartment we could pack items of $h$ different classes.

Bibliography

An Improved Branch and Bound Method to Solve a Pattern Sequencing Problem Based on Partial Orderings

HORACIO H. YANASSE\(^1\), JOSÉ CARLOS BECCENERI\(^1\), AND NEI Y. SOMA\(^2\)

\(^1\) INPE/CTE/LAC, Av. dos Astronautas 1758, P.O. Box 515, São José dos Campos, SP 12227-010, Brazil
E-mails: {horacio, becce}@lac.inpe.br

\(^2\) CTA/ITA/IEC, Praça Marechal Eduardo Gomes 50, Vila das Acácias, São José dos Campos, SP 12228-900, Brazil
E-mail: nysoma@comp.ita.cta.br

The sequence in which the patterns are cut in some cutting settings may be important. Limited storage space, delivery due dates, material handling problems etc. may impose restrictions on the sequence in which the patterns are to be cut.

In this work we focus on one of these sequencing problems, the minimization of open stacks problem (MOSP) which arises, for instance, in wood cutting settings where panels (small items) are to be cut from large boards (objects). A stack is opened for each different panel type cut and it is closed only after the last part of that type is cut. Closed stacks can be removed prior to the cutting of the next pattern. Due to limited storage space around the saw machine, the number of stacks must be kept as low as possible. We seek a cutting sequence of the patterns that minimizes the maximum number of open stacks during the production run.

MOSP is NP-hard (see Linhares and Yanasse (02]) and has been studied in Lins (89), Yuen (91, 95), Yanasse (96, 97a, 97b), Limeira (98), Forster and Wässcher (98), Limeira (98), Faggioni and Bentivoglio (98), Fink and Voß (99), Yanasse et al (99), Becceneri (99) to name a few. Relations of this problem with other combinatorial optimization problems can also be found in Linhares and Yanasse (02).

In Yanasse (97b), it is shown how the MOSP can be put as a graph traversing problem. Let us refer to the graph to be traversed as the MOSP graph. We look after a sequence to traverse all the arcs in the MOSP graph in such a way that the maximum number of open nodes is minimised. There is a one-to-one relation between nodes in the graph and items. Each different item type corresponds to a node in the MOSP graph.

An open node is a node that has been visited but has not been labelled. A node is visited if at least one arc incident to it has already been traversed; a node is labelled only if all arcs incident to it have been traversed. Therefore, an open node corresponds to an open stack; a labelled node corresponds to a closed stack.

Any arc traversing sequence of a MOSP graph defines an order in which the nodes of the graph will be labelled, hence, our ultimate goal is to determine one of these nodes labelling order that minimises the maximum number of open nodes.

Our proposal is to perform a pre-analysis of the MOSP graph to search for special conditions of the nodes that guarantee that some labeling ordering of the nodes can be assumed in some optimal solution to the MOSP. We then show how this partial ordering can be used to reduce the solution search space in a branch and bound method. Computational experiments are presented comparing

145
this new approach with previous ones proposed in the literature.

Bibliography


Author Index

A
Nair Maria M. de Abreu 115
Daniel Aloise 1
Dario J. Aloise 1
A. C. Alves 51
José Maria do C. B. Alves 127
Edoardo Almaldi 3, 5
E. L. Andretta Filho 51
C. F. Antonio 51
Julián Aráoz 105

B
Laura Bahiense 7
Marta Barriá 9
José Carlos Beccenieri 145
Jonathan Benaim 65
Jacek Blazewicz 11, 13
Isabelle Bloch 15
Maria C. Boeres 15
Hugo Bravo-Azilán 19

C
Ricardo S. de Camargo 21
Alfredo Candia-Véjar 19
Antonio Capone 3, 5
Alberto Caprara 25
C. Carnieri 51
Carlos Roberto V. de Carvalho 21
Manuel V. Centeno 29
Irene Charon 31
Alistair R. Clark 33
Yves Crama 35

D
Mauro Dell’Amico 37
Lydia Dowlati 53
Guillermo Durán 39

E
Matthias Ehrigott 125

F
Ismael de Farias 109
Jacques A. Ferland 133
Elena Fernández 105
José Soeiro Ferreira 123
Ricardo P. M. Ferreira 41

G
María J. García 65

Bernard Gendron 133
Marco C. Goldbarg 43, 47
Elizabeth F. Gouvea 43, 47
Sérgio Granville 7
F. A. Grupelli Junior 51
Monique Guignard 53, 131
C. Gulin 51

H
Pierre Hansen 57
José G. Hernández 65
Olivier Hudry 31

J
Maria Helena C. Jardim 83
Claudia M. Justel 87

K
E. M. Kalinowski 51
Marta Kasprzak 11, 13
M. Klimkowskii 51
Skander Kort 69

L
Gilbert Laporte 71
Min Chih Lin 39
Andrea Lodi 37
Irene Loiseau 77, 105
Abilio Lucena 73
Henrique P. L. Luna 41

M
Isabel Méndez 93
Nelson Maculan 77, 83
Federico Malucelli 3, 5
Javier Marencó 85
Lilian Markenzon 87
Fernando Marval 89
Geraldo R. Mateus 117
Jorge D. Melo 1
José P. de Melo 91
Sérgio Mera 39
E. Miqueles 51
Flávio K. Miyazawa 97, 141
Michele Monaci 25, 101
J. A. S. Monteiro 41
Michael Moossen 103
Luís Miguel Moreira 123
N. H. Mussi 51

147
<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Daniel Negrotto</td>
<td>105</td>
</tr>
<tr>
<td>George Nemhauser</td>
<td>109</td>
</tr>
<tr>
<td>Francisco D. M. Neto</td>
<td>43</td>
</tr>
<tr>
<td>Nikola S. Nikolov</td>
<td>113</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Carla S. Oliveira</td>
<td>115</td>
</tr>
<tr>
<td>Gerson Oliveira</td>
<td>7</td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Newton Pacornik</td>
<td>87</td>
</tr>
<tr>
<td>Rodrigo Pascual</td>
<td>35</td>
</tr>
<tr>
<td>Zenilton K. G. Patrocínio Jr.</td>
<td>117</td>
</tr>
<tr>
<td>Ademir F. Pazoto</td>
<td>115</td>
</tr>
<tr>
<td>Mario V. Pereira</td>
<td>7</td>
</tr>
<tr>
<td>Gérard Plateau</td>
<td>77</td>
</tr>
<tr>
<td>Oscar Porto</td>
<td>121</td>
</tr>
<tr>
<td>Daniele Pretolani</td>
<td>119</td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>Hang Qian</td>
<td>53</td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Pablo A. Rey</td>
<td>121</td>
</tr>
<tr>
<td>Celso C. Ribeiro</td>
<td>1, 15</td>
</tr>
<tr>
<td>María-Cristina Riff</td>
<td>103</td>
</tr>
<tr>
<td>Caroline T. M. Rocha</td>
<td>1</td>
</tr>
<tr>
<td>Ana Maria Rodrigues</td>
<td>123</td>
</tr>
<tr>
<td>David M. Ryan</td>
<td>125</td>
</tr>
<tr>
<td>Choongho Ryu</td>
<td>53</td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Juan José Salazar</td>
<td>29, 71, 89</td>
</tr>
<tr>
<td>C. P. Santiago</td>
<td>83</td>
</tr>
<tr>
<td>Frédéric Semet</td>
<td>71</td>
</tr>
<tr>
<td>Gustavo P. Silva</td>
<td>127</td>
</tr>
<tr>
<td>Nei Y. Soma</td>
<td>91, 145</td>
</tr>
<tr>
<td>Cezar M. P. de Souza</td>
<td>47</td>
</tr>
<tr>
<td>Marcone J. F. Souza</td>
<td>127</td>
</tr>
<tr>
<td>Kurt Spielberg</td>
<td>131</td>
</tr>
<tr>
<td>Patrick St-Louis</td>
<td>133</td>
</tr>
<tr>
<td>Maria T. A. Steiner</td>
<td>51</td>
</tr>
<tr>
<td>Jayme L. Szwarecfiter</td>
<td>39</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Simón Tache</td>
<td>65</td>
</tr>
<tr>
<td>Alexandre Tarassov</td>
<td>113</td>
</tr>
<tr>
<td>Andrea Torres</td>
<td>35</td>
</tr>
<tr>
<td>Paolo Toth</td>
<td>25, 101</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Reinaldo Vallejos</td>
<td>9</td>
</tr>
</tbody>
</table>