IMM
DEPARTMENT OF MATHEMATICAL MODELLING

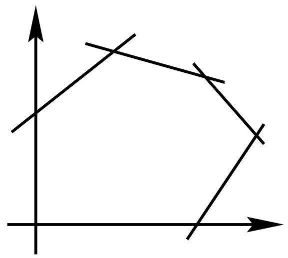
Technical University of Denmark
DK-2800 Lyngby – Denmark

J. No. LP99
6.9.1999
HBN/ms

THIRD SCANDINAVIAN WORKSHOP ON LINEAR PROGRAMMING

Extended Abstracts

Hans Bruun Nielsen (ed)



TECHNICAL REPORT

IMM-REP-1999-12

IMM

Foreword

This booklet contains the abstracts of the papers invited to the Third Scandinavian Workshop on Linear Programming. The workshop was organized and hosted by the Department of Mathematical Modelling of the Technical University of Denmark (IMM). Financial support was provided by the Danish Natural Sciences Research Council (SNF), MICA Foundation and IMM.

The organizing committee for the workshop consisted of the following individuals,

Erling D. Andersen, TU, Delft
Jens Clausen, IMM
Arne Drud, ARKI Consulting & Development A/S
Kaj Madsen, IMM
Oli B.G. Madsen, IMM
Hans Bruun Nielsen, IMM
Jørgen Tind, Copenhagen University

The plenary speakers were

E.D. Andersen, Technical University of Delft, Holland
R.E. Bixby, Rice University, USA
M.A.H. Dempster, University of Cambridge, England

The abstracts appear in alphabetical order of the presenting author's name.

Contents

Recent Work on Primal-Dual Interior-Point Methods for Linear Optimization and their Implementation

Erling D. Andersen

TU Delft, Faculty of ITS, TWI/SSOR
Mekelweg 4
2628 CD Delft. Holland
Ph.: +31(0)152787272 Fax: +31(0)152787255
e.d.andersen@its.tudelft.nl

During the last 15 years interior-point methods for linear programming and their implementation have been studied extensively. Although many different methods have been suggested then most implementations employ the so-called primal-dual method using Mehrotra's predictor-corrector scheme. A primal-dual method is essentially identical to Newton's method applied to the Karush-Kuhn-Tucker optimality conditions in a nontrivial way. This implies in each iteration of a primal-dual method a set of linear equations should be solved which is usually the computationally most expensive part of the method. In general all the implementations of the primal-dual method solves these linear equations using a variant of the Cholesky factorization. Hence, for linear programming there seem to be a consensus upon what is the best method and how it should be implemented. One consequence is that over the last couple of years not much new research about implementing the primal-dual method for linear programming has been reported.

Instead the attention has been directed towards generalizing the primal-dual method to other classes of problems. One particularly interesting class of problems is linear optimization over homogeneous and self-dual cones which includes linear programming as special case. Moreover, for this particular class of problems the good properties of the primal-dual method for the linear case are preserved.

The purpose of the present talk is to review the main ideas used when implementing a primal-dual method for linear programming and discuss

how these ideas are extended to the more general setting of homogeneous and self-dual cones.

Finally, we will dicuss one area where progress has been reported with respect to the implementation of interior-point methods for linear programming which is the matrix reordering step.

MIP: Closing the Gap

Robert E. Bixby

Department of Computational and Applied Mathematics
Rice University
6100 Main St. - MS 134
Houston, TX 77005-1892, USA
bixby@bob.caam.rice.edu

A mixed-integer programming (MIP) problem is a linear programming problem in which some or all of the variables are restricted to take integral values. The addition of these integrality requirements makes MIP difficult in general: The class is NP-hard. At the same time, the MIP modeling paradigm is extremely powerful, with seemingly unlimited applications. Given this wide applicability, combined with the difficulty of the class, one might well ask how are MIPs solved in practice?

The answer, at least for commerical solvers is linear-programming based branch-and-bound, with an emphasis on fast, efficient enumeration, good "branching" rules, clever "node selection," and other common sense ideas. At the same time a rich body of theory has developed for the solution of MIPs; a development largely ignored by the commercial solvers. These theoretical developments have created a backlog of promising computational ideas.

There will be two parts to this talk. In the first part, linear-programming computational advances will be discussed, with an emphasis on the developments of the last year, and, most-particularly, the improvements in the dual simplex method, a key "enabling technology" for MIP.

In the second part of the talk, the focus will be on the integrality part of MIP and the message will be the exploitation of the theory backlog. Using judicious choices of default settings and robust, efficient implementations that hopefully don't hurt when they don't help, we now have much more powerful, general-purpose tools for solving important, practical mixed-integer programming problems. The gap between theory and practice is indeed closing.

Constructing Elevation Maps From SAR Radar Images - an Application of Min Cost Flow Algorithms

Jens Clausen and Jesper Larsen

Department of Mathematical Modelling
Building 321, Technical University of Denmark
DK 2800 Lyngby, Denmark.
jc@imm.dtu.dk jla@imm.dtu.dk

Søren Nørvang Madsen and Johan Mohr

EMI, DTU

Using a technique called SAR interferometry it is possible to construct an image of a region, in which the value of each pixel corresponds to its height modulo an integer multiple of 2π. This is called the wrapped phase of the pixel. Under the assumptions that the surface of the region is smooth, that no noise appears, and that neighboring pixels have heights not differing substantially, it is relatively simple to construct the true height of each point in the region. The true height is called the unwrapped phase and the process is called phase unwrapping. These assumptions are, however, seldomly satisfied in practice. To overcome the difficulties arising, a non-linear optimization model can be applied. The model can be transformed to a very large Min Cost Flow model. We present the problem, discuss various modifications and the computational behaviour of the Min Cost Flow algorithm applied, and present an example of a map generated using the technique.

Semidefinite Programming Approaches for the (Maximum) Satisfiability Problem: Perspectives

E. de Klerk, H. van Maaren, J.P. Warners

Faculty of Technical Mathematics and Informatics,
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands.
e.deklerk@twi.tudelft.nl

SAT and MAX-SAT

The satisfiability problem of propositional logic is the original NP complete problem. An instance of the satisfiability (SAT) or maximum satisfiability (MAX-SAT) problem is defined by a collection of Boolean clauses $\{C_1, \ldots, C_k\}$, where each clause is a disjunction of literals drawn from a set of logical variables $\{x_1, \ldots, x_n\}$. A literal is either a variable x_i or its negation $\neg x_i$ for some i. An optimal solution to a MAX-SAT instance is an assignment of truth values to the variables which maximizes the number of satisfied clauses. MAX-p-SAT is a special case of MAX-SAT where each clause contains at most p literals. The satisfiability problem (SAT) is to determine whether or not all clauses can be satisfied simultaneously.

SDP relaxations: bounds and perspectives

There are many algorithms and heuristics available for the SAT and MAX-SAT problems, even though the 3-SAT and MAX-2-SAT problems respectively are already NP-complete; an extensive review is given in [7]. Semidefinite programming (SDP) became a powerful tool for MAX-2-SAT problems when Goemans and Williamson [6] proved that a 0.87856 approximation algorithm could be obtained by rewriting the MAX-2-SAT problem as a Boolean quadratic programming problem, and solving a convex SDP relaxation of the resulting problem, followed by a randomized rounding procedure. This rounding procedure may be seen as a heuristic

which gives a solution with a quality guarantee, while the optimal value of the SDP gives a bound on the sub-optimality of this heuristic solution. Feige and Goemans [5] have shown that the addition of so-called *triangle inequalities* to the SDP relaxation improves the quality guarantee from 0.87856 to 0.93109. This line of research was extended to MAX-3-SAT by Karloff and Zwick [10] to obtain a 7/8 approximation algorithm (the 7/8 guarantee is optimal unless P = NP).

In subsequent, related work, Zwick [11] obtained improved results for MAX-NAE-SAT[1], and Halperin and Zwick [8] have obtained an 'almost' 7/8 approximation for MAX-4-SAT. Thus much progress has been booked towards obtaining an optimal 7/8 approximation scheme for general MAX-SAT.

For the SAT problem, De Klerk *et al.* [2] have investigated the idea of relaxing a given SAT problem to a semidefinite feasibility problem; the authors showed that SDP can be used in this way to detect unsatisfiability of several classes of polynomially solvable formulas, including 2-SAT, doubly balanced formulas, and a large class of unsatisfiable covering problems. The last class include the so-called *pigeonhole* and *mutilated chessboard* problems which are hard for many conventional SAT-algorithms.

Computational perspectives

In spite of these collected theoretical results, it remains unclear whether semidefinite programming will remain a theoretical tool only, or whether these relaxations also offer computational possibilities in branch and cut algorithms. Until recently, the bottleneck for such an approach has been that it is hard to exploit sparsity in the solution procedure of the SDP relaxations. In this regard, it has recently been shown by Benson, Ye and Zhang [1] that one can solve the dual of the SDP relaxations efficiently, using a dual interior point method. The efficiency of this approach opens new computational perspectives, and De Klerk and Warners [3] have obtained encouraging preliminary results for solving MAX-2-SAT problems of moderate sizes to optimality.

[1]MAX-NAE-SAT is a generalization of MAX-SAT with the additional requirement that the satisfied clauses must have at least one literal which is 'FALSE'.

Aims of the presentation

In the presentation we will review these recent theoretical and computational advances, and give our perspectives on future developments on both fronts.

References

[1] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. Working paper, Computational Optimization LAb, Dept. of Management Science, University of Iowa, Iowa City, USA, 1997.

[2] E. de Klerk, H. van Maaren, and J.P. Warners. Relaxations of the satisfiability problem using semidefinite programming. CWI Report SEN-R9903, CWI, Amsterdam, The Netherlands, 1999.

[3] E. de Klerk and J.P. Warners. Semi-definite programming approaches for max-2-sat and max-3-sat: computational perspectives. Technical Report 98–34, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, 1998.

[4] D. Du, J. Gu, and P.M. Pardalos, editors. *Satisfiability problem: Theory and applications*, volume 35 of *DIMACS series in Discrete Mathematics and Computer Science*. American Mathematical Society, 1997.

[5] U. Feige and M. Goemans. Approximating the value of two prover proof systems with applications to MAX 2SAT and MAX DICUT. In *Proc. Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[6] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[7] J. Gu, P.W. Purdom, J. Franco, and B.W. Wah. Algorithms for the satisfiability (SAT) problem: a survey. In Du et al. [4], pages 9–151.

[8] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. Technical report, Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel, 1998.

[9] S. Joy, J. Mitchell, and B. Borchers. Solving max-sat and weighted max-sat problems using branch-and-cut. Manuscript, 1998.

[10] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? Manuscript, 1997.

[11] U. Zwick. Outward rotations: a new tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. Technical Report, Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel, 1998.

Pricing Financial Options with Linear Programming

M.A.H. Dempster

Centre for Financial Research
Judge Institute of Management
University of Cambridge
Trumpington Street, Cambridge CB2 1AG, England
Phone: +44 (0) 1223 339641 Fax: +44 (0) 1223 339652
`mahd2@cam.ac.uk`

This talk reports the results of several papers concerned with very fast pricing of American style financial options using linear programming techniques. We first present the underlying theoretical results for the American vanilla put option which relate the option valuation problem to equivalent formulations in terms of a free boundary value problem for the Black Scholes parabolic partial differential operator, a variational inequality, an order complementarity problem and an abstract linear programming problem. Our main theoretical result is an extension to the parabolic case of a known equivalence for elliptic coercive partial differential operators of type Z, namely that the order complementarity formulation is equivalent to a least element problem and hence to an abstract LP.

We go on to develop a time stepping UL decomposition LP algorithm for the finite difference discretization of the continuous problem which exploits both the tridiagonal structure of the discrete Black Scholes operator and the problem structure to require only one linear equation solve per simplex iteration and result in linear space complexity.

This new algorithm is first applied to price vanilla American put options and then exotic American options such as lookbacks and Asians which are path dependent and thus involve an extra space variable. Next a theory of local volatility is developed which allows the pricing of such options fitting the volatility smile prevalent in real options markets by contrast to the Black Scholes assumption of constant volatility.

All algorithms developed run in times less – often far less – than their competitors, such as algorithms based on binomial lattice, PDE or Monte Carlo methods, for a given level of numerical accuracy. Numerical results will be given to support these claims and some directions of current extensions and related research indicated.

Problem Structure and Solution Strategies in Hydroelectric Power System Operations Scheduling

Anders Gjelsvik, Olav B. Fosso

Norwegian Electric Power Research Institute (formerly EFI)
7034 TRONDHEIM, Norway
Anders.Gjelsvik@energy.sintef.no
Olav.Fosso@energy.sintef.no

1. Problem overview

Linear programming can be used to find operating schedules in hydroelectric power systems. In the case of short-term scheduling (typically 168 h) the linearised models for such systems contain network substructures. We point out two such cases. We then consider whether advantage may be taken of the network part in the (more general) case where there are other constraints in addition to the network flow part.

Suppose the system is to be analysed over T time intervals. There are two main types of constraints; the reservoir water balances

$$V_t = V_{t-1} + Bu_t + q_t, \qquad t = 1, \ldots, T, \tag{1}$$

and the power balance equations

$$G_t u_t + E_t y_t = d_t, \qquad t = 1, \ldots, T. \tag{2}$$

V_t is a vector of reservoir storages at the end of time step t. The vector u_t contains water releases and spill from the reservoirs, and q_t is the inflow. The vector y_t contains power exchanges in the grid and buying and selling of power. The connections in the hydraulic system are given by the matrix B, which is an incidence matrix with elements of 1 or -1. The power demand at each node in the power grid is d_t, and G_t is a matrix so that the outputs from the power stations are given by

$G_t u_t$. The matrix E_t describes the transmission grid and the allocation of dealing options. It has network structure with non-unit multipliers in transmission arcs to account for losses. Both E_t and G_t are derived by linearisation of non-linear relationships. The components of V_t, u_t and y_t all have upper and lower limits.

In addition to the above there may be a range of other more special constraints, for instance ramp rate constraints (on the difference $u_t - u_{t-1}$).

The objective function to be minimised is

$$\sum_{t=1}^{T} c_t^T y_t - \Phi(V_T), \tag{3}$$

where c_t contains marked price estimates and $\Phi(V_T)$ is the value of water at the final time, a convex function given by hyperplanes.

The resulting linear model has an L-shaped form. The upper part contains the water balances (1) and the lower portion consists of the power balances (2); the constraints that represent $\Phi(\cdot)$, and possibly other constraints.

2. Special cases with network structure

2.1. Non-cascaded reservoirs

We now assume that there are no extra special constraints. In general, a component of u_t has two entries in the matrix B, one for the upstream reservoir and one for the downstream reservoir; and then one element in the power balance. If, however, the reservoirs are not cascaded, and if $\Phi(V_T)$ is separable, each release enters only one water balance and the power balance, and so the problem becomes a minimum cost network flow problem, with gains in the arcs due to the conversion factors in the matrix G_t.

2.2. Price given at all grid nodes

This case is reasonable in a deregulated system like the Norwegian, where the generation is coordinated by prices. It corresponds to a relaxation of the power balances, and in this case, the constraint matrix corresponds to a pure network with arc gains one.

In cooperation with the Norwegian University of Science and Technology in Trondheim (Dag Augland), a generalised network flow code was developed, based largely on [1], that has been successfully applied to these special cases [2].

3. Solution strategies for the general case

In general, one has to deal with the full set of constraints. If the system has 20 reservoirs and is connected to three grid nodes, the ratio of network constraints to other constraints will be 20:3, or about 13 per cent "complicating constraints." For 168 time steps , there will be 3864 constraints, plus the number of constraints required to represent $\Phi(V_T)$. A typical number of variables will be around 25000. This is not a very large problem by today's standards, but it has to be solved frequently, and often repeated solutions with different inflow and price forecasts are required. At our institute, a program named SHOP has been developed for short-term scheduling. If necessary, it uses successive linearisations, leading to subproblems as described above. A commercial solver (CPLEX) is used.

In SHOP, various solution strategies from CPLEX can be used:

1. First solve the network part of the problem, ignoring the power balances. Then add the other constraints and solve with a dual simplex algorithm.
2. Use primal simplex from the beginning
3. Use of an interior point algorithm

If the problem is solved from scratch, experience shows that 1) is much better than 2). In a typical case, the number of dual simplex iterations after the network part was solved in 1) was around 500, while 2) required about 2700 iterations.

If a close-to-optimal basic solution is available (as after a relinearisation of the problem) 2) is the more efficient, since the network flow phase

of 1) leads the solution away from the optimum.

The above remarks apply to the use of the simplex algorithm. For the largest models that we have run, the barrier solver in CPLEX has turned out to be significantly faster than the simplex method (see also [3]). In this case the network substructure is not used. However, in some cases the simplex algorithm may still be advantageous, for instance if warm start can be used, or if a basic solution is wanted.

References

[1] F. Glover, J. Hultz, D. Klingman, and J. Stutz. Generalized networks: A fundamental computer-based planning tool. *Management Science*, 24(12):1209–1220, 1978.

[2] A. Hjertenæs, A. Gjelsvik, and Ø. Skarstein. Implementation of an integrated system for short term scheduling in Alta power plant. In Einar Broch and Dagfinn K. Lysne, editors, *Hydropower '92*, Rotterdam, 1992. A. A. Balkema.

[3] Claus Jørgensen. *Driftsoptimering på kraftvarmesystemer*. PhD thesis, Institut for Matematisk Modellering, Danmarks Tekniske Universitet, 1997.

Deriving Duality for l_p-Norm Optimization Using Conic Optimization*

François Glineur

Faculté Polytechnique de Mons,
Service de Mathématique et de Recherche Opérationnelle,
Rue de Houdain, 9, B-7000, Mons, Belgium.
Francois.Glineur@fpms.ac.be http://image.fpms.ac.be/~glineur/

Abstract. In this paper, we formulate the l_p-norm optimization problem as a conic optimization problem and derive its standard duality properties.
We first define an *ad hoc* closed convex cone $\mathcal{L}$ and derive its dual. We express then the standard l_p-norm optimization primal problem as a conic problem involving $\mathcal{L}$. Using conic duality, we derive the dual of this problem and prove the well-known regularity properties of this primal-dual pair, i.e. zero duality gap and dual attainment.

Key words: l_p-norm optimization, conic optimization, duality.

1. Introduction

l_p-norm optimization problems are an important class of convex problems. This class includes as special cases linear programming, quadratic programming, quadratically constrained quadratic programming and l_p-norm approximation problems.

Some duality results are known for l_p-norm optimization. Namely, a pair of feasible primal-dual l_p-norm optimization problems satisfies the weak duality property, has a zero duality gap and has its dual optimum objective attained by a feasible solution.

These results were first obtained by Peterson and Ecker [2], [1], [3] and later simplified by Terlaky [5]. The aim of this paper is to derive

*This work is supported by a grant from the F.N.R.S. (Belgian National Fund for Scientific Research).

these results using the machinery of conic duality, which simplifies further the proofs.

In order to use this setting, we define an appropriate convex cone that allows us to express l_p-norm optimization problems as conic programs. In order to analyze duality for this problem, we first study some properties of this cone (closedness, interior) and determine its dual cone, which happens to have nearly the same structure.

We apply then the general duality theory for conic optimization to our programs [4], to prove the above mentioned duality results about l_p-norm optimization.

This leads in our opinion to clearer proofs, the specificity of the problem being confined to the convex cone we use in our formulation. Moreover, the reason why this problem has better duality properties than a general conic problem becomes clear: this is essentially due to the existence of a strictly interior primal solution. However, such an interior solution does not always exist and a reduction procedure involving an equivalent regularized problem has to be introduced to prove the results in the general case.

2. The $\mathcal{L}$ cone

Definition 1. *Let $p \in \mathbb{R}^n$ with $p_i > 1$. We define the following set*

$$\mathcal{L}^p = \left\{ (x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \le \kappa \right\}$$

using in the case of a zero denominator the following convention:

$$\frac{|x|}{0} = \begin{cases} +\infty & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases} .$$

This means that $(x, 0, \kappa) \in \mathcal{L}^p$ implies $x = 0$.

Theorem 1. *$\mathcal{L}^p$ is a convex cone.*

The interior of this cone is

$$\operatorname{int} \mathcal{L}^p = \left\{ (x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}_{++} \mid \sum_{i=1}^n \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} < \kappa \right\} .$$

In the special case where $p_i = 2$ for all i, our cone $\mathcal{L}^p$ becomes

$$\mathcal{L}^{\{2,\cdots,2\}} = \left\{ (x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \sum_{i=1}^n x_i^2 \leq 2\theta\kappa \right\} ,$$

which is the *hyperbolic* or *rotated* second-order cone. To determine the dual cone of $\mathcal{L}^p$, we will use the following inequality, which lies at the heart of duality for the $\mathcal{L}$ cones.

Lemma 1. *Let $a, b \in \mathbb{R}_+$, $\alpha, \beta > 1$ and $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. We have the inequality*

$$\frac{a^\alpha}{\alpha} + \frac{b^\beta}{\beta} \geq ab ,$$

with equality if and only if $a^\alpha = b^\beta$.

For ease of notation, we also introduce the *switched* cone $\mathcal{L}_s$ as the $\mathcal{L}$ cone with its last two components exchanged, i.e.

$$(x, \theta, \kappa) \in \mathcal{L}_s^p \Leftrightarrow (x, \kappa, \theta) \in \mathcal{L}^p .$$

We are now ready to describe the dual of $\mathcal{L}^p$.

Theorem 2 (Dual of $\mathcal{L}$). *Let $p, q \in \mathbb{R}^n$ such that $p_i, q_i > 1$ and $\frac{1}{p_i} + \frac{1}{q_i} = 1$ for each i. The dual of $\mathcal{L}^p$ is $\mathcal{L}_s^q$.*

The dual of a $\mathcal{L}$ cone is thus equal, up to a permutation of two variables, to another $\mathcal{L}$ cone with a dual set of exponents.

Corollary 1. *We also have $(\mathcal{L}_s^p)^* = \mathcal{L}^q$, $(\mathcal{L}^q)^* = \mathcal{L}_s^p$ and $(\mathcal{L}_s^q)^* = \mathcal{L}^p$.*

Corollary 2. *$\mathcal{L}^p$ and $\mathcal{L}_s^q$ are closed.*

The next theorem describes the cases where two vectors from $\mathcal{L}^p$ and $\mathcal{L}_s^q$ are orthogonal, which will be used in the study of the duality properties.

Theorem 3. *Let $v = (x, \theta, \kappa) \in \mathcal{L}^p$ and $v^* = (x^*, \theta^*, \kappa^*) \in \mathcal{L}_s^q$. We have $v^T v^* = 0$ if and only if*

$$\kappa^*(f_p(x, \theta) - \kappa) = 0, \quad \theta(f_q(x^*, \kappa^*) - \theta^*) = 0,$$
$$\kappa^* \tfrac{|x_i|^{p_i}}{\theta^{p_i - 1}} = \theta \tfrac{|x_i^*|^{q_i}}{\kappa^{*\, q_i - 1}}, \quad x_i x_i^* \leq 0 \text{ for all } i .$$

3. Duality for the l_p-norm optimization problem

Let $K = \{1, 2, \ldots, r\}$ and $I = \{1, 2, \ldots, n\}$. Let $\{I_k\}_{k \in K}$ be a partition of I into r components. Moreover, let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times r}$, $\eta \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^r$. We will also use the following notation: v_I (resp. M_I) denotes the restriction of column vector v (resp. matrix M) to the components (resp. rows) whose indices belong to set I. The primal l_p-norm optimization problem is the following

$$\text{(P)} \qquad \inf\ \psi(x, z) = c^T x + d^T z + \textstyle\sum_{k=1}^r z_k \sum_{i \in I_k} \frac{1}{p_i} \left| \frac{x_i}{z_k} \right|^{p_i}$$
$$\text{s.t.} \quad Ax + Bz = \eta \text{ and } z \geq 0 .$$

This problem can be shown to be equivalent to the following conic optimization problem using $\mathcal{L}$ cones:

$$(\mathrm{P_C}) \qquad \inf c^T x + d^T z + e^T v$$
$$\text{s.t.} \quad Ax + Bz = \eta, \quad (x_{I_k}, z_k, v_k) \in \mathcal{L}^{p_{I_k}}\ \forall k \in K .$$

This conic problem being in standard form, we can easily derive its dual and find

$$(\mathrm{D_C}) \qquad \sup \eta^T y \quad \text{s.t.} \quad A^T y + x^* = c, \quad B^T y + z^* = d,$$
$$v^* = e, \quad (x_{I_k}^*, z_k^*, v_k^*) \in \mathcal{L}_s^{q_{I_k}}\ \forall k \in K .$$

After some simplifications, we obtain the usual l_p-norm optimization dual:

$$\text{(D)} \qquad \sup \eta^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \frac{1}{q_i} \left| c_i - a_i^T y \right|^{q_i} \leq d_k - b_k^T y \quad \forall k \in K .$$

The following theorem deals with weak duality for the l_p-norm optimization problem.

THEOREM 4 (Weak duality). *If (x,z) is feasible for (P) and y is feasible for (D), we have $\eta^T y \le \psi(x,z)$. Equality occurs if and only if for all $k \in K$ and $i \in I_k$*

$$z_k\left(\sum_{i\in I_k} \tfrac{1}{q_i}\left|c_i - a_i^T y\right|^{q_i} + b_k^T y - d_k\right) = 0,$$
$$x_i(c_i - a_i^T y) \le 0, \quad \frac{|x_i|^{p_i}}{z_k^{p_i-1}} = z_k\left|c_i - a_i^T y\right|^{q_i}.$$

In the case of a general pair of primal and dual conic problems, the duality gap is not always equal to zero, neither are the primal or dual optimum objective values always attained by feasible solutions. However, in the special case of linear optimization, we always have a zero duality gap and attainment of both optimum objective values. The status of l_p-norm optimization lies somewhere between these two situations: the duality gap is always zero but attainment of the optimum objective value can only be guaranteed for the dual problem, as described by the following theorem:

THEOREM 5. If (P) and (D) are feasible, the dual optimum objective is attained with a zero duality gap, i.e.

$$\begin{aligned} d^* &= \max \eta^T y \quad \text{s.t.} \quad \textstyle\sum_{i\in I_k} \frac{1}{q_i}\left|c_i - a_i^T y\right|^{q_i} \le d_k - b_k^T y \quad \forall k \in K \\ &= \inf \psi(x,z) \quad \text{s.t.} \quad Ax + Bz = \eta \text{ and } z \ge 0 . \end{aligned}$$

This theorem is proved by introducing a regularized couple of primal-dual conic problems

$$(\mathrm{P_r}) \qquad \inf c_{I_N}^T x_{I_N} + d_N^T z_N + e^T v_N \quad \text{s.t.} \quad \begin{array}{l} A_{I_N} x_{I_N} + B_N z_N = \eta, \\ (x_{I_k}, z_k, v_k) \in \mathcal{L}^{p_{I_k}} \ \forall k \in N , \end{array}$$

$$(\mathrm{D_r}) \qquad \sup \eta^T y \quad \text{s.t.} \quad \begin{array}{l} A_{I_N}^T y + x_{I_N}^* = c_{I_N}, \quad B_N^T y + z_N^* = d_N, \\ v_N^* = e, \ (x_{I_k}^*, z_k^*, v_k^*) \in \mathcal{L}^{q_{I_k}} \ \forall k \in N , \end{array}$$

where N is the set of indices k such that z_k is not identically zero on the feasible region. This pair has a zero duality gap and the dual optimum is attained, because its primal problem has always a strictly interior solution. The idea of the proof is then to show that $\inf(\mathrm{P}) = \inf(\mathrm{P_r}) =$

$\max(\mathrm{D_r}) = \max(\mathrm{D})$, the last equality being the most delicate part.

References

[1] E. L. Peterson and J. G. Ecker, *Geometric Programming: Duality in quadratic programming and l_p approximation II,* SIAM Journal on Applied Mathematics **13** (1967), 317-340 .

[2] E. L. Peterson and J. G. Ecker, *Geometric Programming: Duality in quadratic programming and l_p approximation I,* Proceedings of the International Symposium of Mathematical Programming (Princeton, New Jersey) (H. W. Kuhn and A. W. Tucker, eds.), Princeton University Press, 1970.

[3] E. L. Peterson and J. G. Ecker, *Geometric Programming: Duality in quadratic programming and l_p approximation III,* Journal on Mathematical Analysis and Applications **29** (1970), 365-383.

[4] J. F. Sturm, *Primal-Dual Interior-Point Approach to Semidefinite Programming,* Ph.D. thesis, Erasmus Universiteit Rotterdam, The Netherlands, 1997.

[5] T. Terlaky, *On l_p programming,* European Journal of Operations Research **22** (1985), 70-100.

On the Solution of Pseudomonotone Variational Inequalities by Nonlinear and Cutting Plane Methods

Jean-Louis Goffin

Faculty of Management, McGill University
1001 Sherbrooke Street West, Montréal, Québec, H3A 1G5
Canada
Ph: 1-514-398-4003 FAX: 1-514-398-3876
Ma56@musica.mcgill.ca

The analytic center cutting plane method can be applied to the numerical solution of pseudomonotone variational inequalities, working on a reformulation of the variational inequality as a convex feasibility problem. Interior point methods which deal directly with a nonlinear functional have been studied by Nesterov and Nemirovskii, and by B. Jansen.

We intend to report computational and complexity results on two algorithms:
1) a primal-dual analytic center cutting plane method which assumes that both the functional and the constraints are defined by cutting planes
2) a method that will mix these the two strategies, i.e., a method that uses a nonlinear representation of the functional and a cutting plane representation of the constraints.

In the case of the primal-dual analytic center cutting plane method which assumes that the functional is strongly monotone, and possibly mulltivalued, we also study a weighing scheme that uses the primal variables as weights. This greatly improve computational results and a complexity analysis appears possible. are defined by cutting planes

We also consider a quadratic cut method based on analytic centers for the convex quadratic feasibility problem defined by a finite yet large number of convex quadratic inequalities. We extend the quadratic cut algorithm of Luo and Sun for solving such problems by placing or translating the quadratic cuts directly through the current approximate center.

We show that the total number of iterations is at most $O\left(N \ln \frac{1}{\epsilon}\right)$, where N is the total number of convex quadratic inequalities and ϵ is the radius of the largest ball contained in Γ. Moreover, we show that, at each iteration, the total number of (damped) Newton steps required to update from one approximate analytic center to another is bounded by a constant plus $O\left(\ln\left(\frac{N}{\epsilon^2}\right)\right)$.

We expect to report on the case where the number of underlying cuts may be infinite.

This is, in parts, joint work with Z. Q. LUO and my former doctoral students Michel DENAULT and Faranak SHARIFI-MOKHTARIAN.

The Use of Linear and Integer Programming in Airline Crew Scheduling

Niklas Kohl

Carmen Systems AB
Odinsgatan 9
SE-411 03 Gothenburg / Sweden
niklas@carmen.se

1. Introduction

This is an extended abstract for a presentation about the use of linear and in particular linear integer programming embedded in a system for solving large scale real-world optimization problems in the airline industry.

We define the airline crew scheduling problem as the problem of assigning indivisible tasks (generally referred to as flight legs or just *legs*) to named or anonymous crew members. In the literature as well as in practice this problem is decomposed into the crew pairing problem and the crew assignment or crew rostering problem. In the crew pairing problem, legs are grouped into larger pieces of work (known as pairings or trips), usually with a duration of some days. In the crew assignment problem the anonymous pairings - as well as other tasks - are grouped in to anonymous or (the usual case) personalized lines of work or *rosters*. Typically the planning horizon of the crew assignment problem is one month.

2. The Carmen systems for crew scheduling

Carmen Systems develop optimization systems for crew scheduling. The systems are installed and used at SAS, Lufthansa, British Airways, Air France, Alitalia, Iberia, KLM, Aero Mexico and a number of smaller airlines. The Carmen crew pairing and crew assignment systems are based on a tool-box of methods of which the most, somewhat simplified,

can be described by the following algorithm. The algorithm assumes the existence of an initial solution.

1. Fix some of the decisions in the current solution (for example the assignment of some pairings to crew members)
2. Generate all possible pairings/rosters, given the decisions taken in step 1.
3. Solve a linear integer program
4. Update the solution and go to step 1.

This presentation focuses on the mixed integer program involved in step 3 above. However, it is important to note that as well step 1 as step 3 are interesting from an optimization point of view. In our experience the quality of the final solution depends more on the algorithms (heuristics) employed to limit the solution space in step 1 than the algorithms employed to solve the integer program. Also the majority of the total solution time (typically between 60% and 90%) is spent in the generator. Also, note the due to the iterative nature of the algorithm the integer program must be solved fast. Often the time limit imposed is between 5 and 30 minutes. In some cases this doesn't permit to prove optimality.

3. Linear integer programming models for crew scheduling

The Carmen crew pairing and crew assignment systems work on different linear integer programming models. Here we introduce the basic model and outline some extensions. In the presentation we will discuss these in more detail. A simple linear integer programming model of the crew pairing problem is the set covering model displayed below.

$$\begin{aligned} &\textit{minimize} \quad \sum_{i\in\mathcal{I}} c_i x_i, && \textit{s.t.} \\ &\sum_{i\in\mathcal{I}} a_{ij} x_i && \geq 1 \quad \forall j \in \mathcal{J} \\ &x_{ij} \in \{0,1\} \end{aligned}$$

$\mathcal{J}$ is the set of flight legs whereas $\mathcal{I}$ is the set of pairings. Without step

1 in the algorithm above $|\mathcal{I}|$ would almost be infinitely large. x_i is a binary decision variable which determines whether pairing i is used in the solution or not. a_{ij} is the binary coefficient which determine whether pairing i covers leg j. We don't make any assumptions about the objective except that it must be the sum of the costs of the pairings used in the solution. The model is a bit misleading because we are interested in a partitioning (=1) solution, not a covering (≥ 1) solution. The covering formulation is justified by the fact, that a covering solution often can be "transformed" to a partitioning solution with the same or lower costs, by letting one of the crew members fly as passenger.

In practice the set covering model doesn't capture all important aspects of the crew pairing problem. Therefore one can introduce other constraints of the form $\sum_i a_i^1 x_i \leq b$. Some of the additional constraints introduced in the Carmen crew pairing system are

- Base constraints are used to ensure that the distribution or pairings match with the distribution of crew. Here a_i^1 could be the number of working days (or some other attribute) of pairing i and b the availability of crew at the corresponding base. Often there will be one base constraint per (crew base, calendar day) combination. In some cases there is also a lower bound on the left hand side, imposing a minimum utilization of crew.
- Global constraints are used for a number of purposes: limit the number of pairings longer than n days, ensure the existence of pairings with some special properties which makes them useful for training and limiting the number of nightstops in particular unpopular stations.
- Variable crew complement is used if all flight legs don't require the same number of crew members. Then the right hand side of the covering constraint must reflect this, i.e. 1 is replaced by b_j, where b_j is the needed number of crew members on leg j.

To model the crew assignment problem we use a generalization of the set partitioning model.

$$
\begin{array}{llll}
\mathit{minimize} & \sum_{i\in\mathcal{I}} c_i x_i, & s.t. & \\
& \sum_{i\in\mathcal{I}} a_{ij} x_i & = 1 & \forall j \in \mathcal{J} \\
& \sum_{i\in\mathcal{I}} a_{ik} x_i & = b_k & \forall k \in \mathcal{K} \\
& x_{ij} \in \{0,1\}, & x_{ik} \in \{0,1\} &
\end{array}
$$

Here $\mathcal{J}$ is the set of crew members, $\mathcal{K}$ is the set of pairings and $\mathcal{I}$ is the set of rosters. a_{ij} is one if roster i was generated for crew member j and 0 otherwise. a_{ik} is one if it contains pairing k and 0 otherwise. The right hand side b_k gives the number of times pairing k must be assigned. Here we use a partitioning model because crew members must be assigned exactly one roster (the $\mathcal{J}$ constraint set) and because one would never consider to overcover an entire pairing.

Some of the extensions of the this model are:

- Qualification constraints are used to ensure the presence of certain qualifications in all or some pairings. An example is, that 5 of 7 crew members must speak Spanish on a pairing including legs to Spanish speaking countries. A qualification constraint can also model that a recently promoted captain is only permitted to fly together with an experienced first officer.
- Global constraints can be used to enforce a certain quality (some other criteria than the one modeled by the objective) of the solution.
- Route instruct constraints enforce two (or more) crew members to fly the same pairings - for training purposes - within some time period. This is modeled with one constraint per pairing, +1 coefficients for the one crew member, -1 coefficients for the other, and a right hand side of 0.

Even though we are mainly interested in the integer solution to problems presented above, the solution to the relaxed (LP) problem gives useful dual information which can be exploited to generate columns (pairing or rosters depending on the application) in the next iteration. This idea can be exploited in a number of ways. In all cases the idea is to generate columns with negative reduced cost with respect to the current dual solution. Since the rules and regulations of crew pairing and assignment problems are often very complex, the column generating subproblem cannot be solved optimally. Instead we can relax the rules, generate the best pairing / roster and test the rules in a subsequent filter. Alternatively we can use the fact that interesting parings / rosters have negative reduced costs to prune the search in step 2 of the algorithm above.

In the presentation we will give examples of size and computational results using commercial solvers on an array of linear integer programming problems derived from real-world crew scheduling problems. The examples will show that the solution time depends on as well size as structure of the problems.

We will also discuss our own algorithmic work. This work mainly focuses on preprocessing and heuristic problem reductions. Also we present results on an integer programming optimizer designed especially for crew pairing problems.

Acknowledgments

This presentation is partly based on results obtained by my colleagues Mattias Grönkvist, Dr. Curt Hjorring, Dr. Stefan Karisch as well as partners in our Esprit projects. This research is partly funded by the Commission of the European Union through Esprit projects 24 960 Parrot and 20 115 Paros.

Speeding up Column Generation in VRPTW

Jesper Larsen

Department of Mathematical Modelling
Building 321, Technical University of Denmark
DK 2800 Lyngby, Denmark.
jla@imm.dtu.dk http://www.imm.dtu.dk/~jla

Routing with time windows (VRPTW) has been an area of research that have attracted many researchers within the last 10 – 15 years. In this period a number of papers and technical reports have been published on the exact solution of the VRPTW.

The VRPTW is a generalization of the well-known capacitated routing problem (VRP or CVRP). In the VRP a fleet of vehicles must visit (service) a number of customers. All vehicles start and end at the depot. For each pair of customers or customer and depot there is a cost. The cost denotes how much is costs a vehicle to drive from one customer to another. Every customer must be visited exactly ones. Additionally each customer demands a certain quantity of goods delivered (know as the customer demand). For the vehicles we have an upper limit on the amount of goods that can be carried (known as the capacity). In the most basic case all vehicles are of the same type and hence have the same capacity. The problem is now for a given scenario to plan routes for the vehicles in accordance with the mentioned constraints such that the cost accumulated on the routes, the fixed costs (how much does it cost to maintain a vehicle) or a combination hereof is minimized.

In the more general VRPTW each customer has a time window, and between all pairs of customers or a customer and the depot we have a travel time. The vehicles now have to comply with the additional constraint that servicing of the customers can only be started within the time windows of the customers. It is legal to arrive before a time window "opens" but the vehicle must wait and service will not start until the time window of the customer actually opens.

For solving the problem exactly 4 general types of solution methods have evolved in the literature: dynamic programming, Dantzig-Wolfe (column generation), Lagrange decomposition and solving the classical model formulation directly.

Presently the algorithms that are based on the Dantzig-Wolfe method performs best (Desrochers, Desrosiers and Solomon, and Kohl). In the Dantzig-Wolfe method the problem is split into two problems: a "master problem" and a "subproblem". The master problem is a relaxed set partitioning problem that guarantees that each customer is visited exactly ones, while the subproblem is a shortest path problem with additional constraints (capacity and time window). Using the master problem the reduced costs are computed for each arc, and these costs are then used in the subproblem in order to generate routes from the depot and back to the depot again. The best (improving) routes are then returned to the master problem and entered into the relaxed set partitioning problem. As the set partitioning problem is relaxed by removing the integer constraints the solution is seldomly integral therefore the Dantzig-Wolfe method is embedded in a separation-based solution-technique.

In the Dantzig-Wolfe process a large number of columns may be generated, but a significant fraction of the columns introduced will not be interesting with respect to the master problem.

We report on positive results that are achieved by stopping the route-generation process prematurely in the case of time-consuming shortest path computations. Often this leads to stopping the shortest path subroutine in cases where the information (from the dual variables) leads to "bad" routes. The premature exit from the shortest path subroutine restricts the generation of "bad" routes significantly. This produces solutions significantly faster than before and has made it possible to solve problem instances not solved to optimality before.

Application of Constraint Logic Programming to the Detection of Embedded Network Structures

James Little, Nalân Gülpınar

Department of Mathematics and Statistics, Brunel University, Middlesex, UB8 3PH, UK.

Abstract not available. This contribution was due to a last minute change. It is based on a paper with the same title, which can be found at http://www.brunel.ac.uk/ masrjjl/pubs/netdetect.zip

Stochastic Modelling of Logistic and Hedging Strategies in the Petroleum Industry

E.A. Medova

Judge Institute of Management Studies
University of Cambridge
Trumpington Street, Cambridge CB2 1AG , England
Phone: +44 (0) 1223 339593 Fax: +44 (0) 1223 339652
`eam28@cam.ac.uk`

A strategic level planning model of major activities such as supply, storage, transportation, transformation together with trading on the commodity market is formulated as a dynamic stochastic linear programme. We investigate the influence of random prices and demands on scheduling decisions at the tactical level of planning which is viewed as the implementable first stage solution of the stochastic model. These decisions are analysed regarding the costs of the corresponding stochastic formulations: two-stage and multistage. From the computational view point the multistage problem formulation shows significant reduction in memory requirements and allows investigation of a large number of scenarios. The model is implemented using a suite of existing and newly developed software and constitutes a case study of a very large-scale stochastic problem involving decisions in both space and time.

On the Numerical Issues of Interior Point Methods

Csaba Mészáros

Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1518 Budapest, P.O. Box 63. Hungary
`mcsaba@novell.oplab.sztaki.hu`

The fundamental step of interior point implementations is solving systems of linear equations of the form:

$$AD^{-2}A^T\alpha = \beta,$$

where $D \in R^{n\times n}$ positive definite, $A \in R^{m\times n}$ of full row rank, $\alpha, \beta \in R^m$. The method considered to be the practically most effective is the Cholesky decomposition of the matrix on the left hand of the equation:

$$AD^{-2}A^T = LL^T,$$

where $L \in R^{m\times m}$ lower triangular. Our investigation focuses on two numerical techniques which showed to be especially important for the Cholesky decompositions in interior point methods.

The first is the modified Cholesky factorization which increases the robustness of decompositions performed during interior point iterations when the optimization problem is degenerate [4]. The modified Cholesky factorization skips factors whose diagonal value is "small" and considers the underlying row in AD^{-1} to be dependent on the set of preceding rows:

$$
\begin{aligned}
&\text{For } i = 1, ..., m \\
&\quad c_i = \sum_{k=1}^{n} \left(AD^{-1}\right)_{ik}^2 - \sum_{k=1}^{i-1} L_{ki}^2 \\
&\quad L_{ii} = \begin{cases} \infty & \text{if } c_i < \epsilon_i, \\ \sqrt{c_i} & \text{otherwise} \end{cases} \\
&\quad \text{For } j = i+1, ..., m \\
&\qquad L_{ij} = \frac{\sum_{k=1}^{n} \left(AD^{-1}\right)_{ik} \left(AD^{-1}\right)_{jk} - \sum_{k=1}^{i-1} L_{kj} L_{ki}}{L_{ii}} \quad .
\end{aligned}
$$

The principles of the Cholesky factorization suggest using adaptive pivot tolerances in a natural way, i.e. ϵ_i may be different for different indices. An attractive strategy for choosing the pivot tolerance is

$$\epsilon_i = \epsilon \sum_{k=1}^{n} \left(AD^{-1}\right)_{ik}^2 \quad \text{for } i = 1, ..., m,$$

since the resulting numerical tolerance is independent of the row scaling of A. Furthermore, it is easy to see that such rows of AD^{-1} are considered to be dependent whose relative distance to the linear space of the preceding rows is less than $\sqrt{\epsilon}$ [2].

The second technique is called regularization, in which small diagonal values in D are enlarged to "reasonably" large ones. Such a regularization has the important property that the error introduced in the solution of the system is of same order of magnitude than the regularization itself. Furthermore, the factorization of the regularized system can be used as preconditioner in an iterative refinement process which always converges to the solution of the unregularized system [1].

In our study we highlight the connection of the discussed techniques and give their generalization to the decomposition of quasidefinite matrices. We show how nonconvex optimization problems can be handled where indefinite matrices have to be regularized to positive definite ones [3].

The techniques discussed so far in the literature of numerical techniques of IPMs cover the case when a "small" positive pivot arises during

the Cholesky factorization. Present approaches are capable to reliable handle such cases. An even more serious situation is when a negative diagonal value arises whose absolute value is larger than the tolerances used. Whereas, this case is rare in the practice, it can result in serious problems for the implementations. We will show how this case can be handled with the techniques discussed.

References

[1] Cs. Mészáros. On free variables in interior point methods. *Optimization Methods and Software*, 9:121–139, 1997.

[2] Cs. Mészáros. On a property of the Cholesky factorization and its consequences in interior point methods. Working paper WP 98–7, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[3] R.J. Vanderbei and D.F. Shanno. An interior–point algorithm for nonconvex nonlinear programming. Technical report, SOR 97-21, Princeton University, 1997.

[4] S.J. Wright. Modified Cholesky factorization in interior–point algorithms for linear programming. Technical Report P600-0596, Mathematics and Computer Science Division, Argonne National Laboratory, 1996.

Analysis of the structure of Production Frontiers - The Extended Facet Approach in DEA

Niels Christian Petersen and Ole B. Olesen

Department of Management
USD, Odense University
Denmark
`ncp@sam.sdu.dk` `ole@sam.sdu.dk`

DEA studies based upon the so-called Extended Facet Approach have been carried out. However, focus has been on the pricing of slacks along with an estimation of efficiency indices and not on the structure of the estimated frontier technology as described by e.g. elasticities of various kinds, and the problem concerned with the identifications of all so-called Full Dimensional Efficient Facets (FDEFs) has not been given full attention. Focus in this study is on the identification of all FDEFs in real life data sets and the development of tools for the practical implementation of the extended facet approach under different conditions w.r.t. scale. In addition, preliminary results in structural frontier analyses for the Danish hospital sector is provided along with computational results for the Danish agricultural sector.

A Scaled Gauss-Newton Primal–Dual Search Direction for Semidefinite Optimization

E. de Klerk, J. Peng, C. Roos, T. Terlaky

Faculty of Information Technology and Systems
Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands.
`C.Roos@its.tudelft.nl` http://ssor.twi.tudelft.nl/~roos/

Interior point methods for semidefinite optimization (SDO) became a popular research area when it became clear that the known interior-point algorithms for linear optimization (LO) can often be extended to the more general SDO case. Following the trend in LO, primal–dual algorithms soon enjoyed the most attention. Unlike in the LO-case, however, there are many possibilities to obtain primal–dual search directions. The usual scheme is to apply linearization in conjunction with symmetrization to the perturbed optimality conditions of the SDO problem. Symmetrization is necessary since the linearized system is overdetermined. Different directions arise when the perturbed optimality conditions are linearized and subsequently symmetrized; a quite comprehensive survey of the search directions obtained this way may be found in [2]. A recent idea by Kruk *et al.* [1] was to avoid symmetrization by solving a least squares problem by the Gauss-Newton method. Kruk *et al.* obtained a numerically robust search direction in this way, but did not give convergence proofs for their search direction. The work in our paper was inspired by their approach: here we show that, by using scaling and a different (local) norm in the definition of the least squares problem, a direction is obtained which allows a polynomial time convergence analysis.

Key words: semidefinite optimization, interior-point method, polynomial-time.

References

[1] S. Kruk, M. Muramatsu, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. The Gauss–Newton direction in semidefinite programming. Research report CORR 98-16, University of Waterloo, Dept. Combinatorics and Optimization, Waterloo, Canada, 1998.

[2] M.J. Todd. On search directions in interior-point methods for semidefinite programming. Research report, School of OR and Industrial Engineering, Cornell University, Ithaca. New York 1485-3801, 1997.

Optimization Algorithms and Applications at SOL

Michael Saunders

Systems Optimization Laboratory
Department of EESOR
Stanford University
mike@sol-michael.stanford.edu

The Systems Optimization Laboratory was formed by George Dantzig and Richard Cottle 25 years ago to foster research on algorithms for linear and nonlinear programming. Some important topics at SOL are mathematical models, sparse matrix methods, numerical reliability, portable software, and collaboration with Philip Gill at UC San Diego.

We review the main solvers (MINOS, NPSOL, SQOPT, SNOPT) and their use within larger systems (GAMS, AMPL, OTIS, DECIS). Excerpts from "Life in New Zealand" will illustrate some of the key concepts in optimization.

LP in Forestry

Peter Tarp
The Royal Veterinary and Agricultural University
Department of Economics and Natural Resources, Unit of Forestry
23, Rolighedsvej
DK-1958 Frederiksberg C, Denmark
Phone +45 35 28 22 21. Fax +45 35 28 26 71.
E-mail: peta@kvl.dk

Linear programming (LP) is extensively used within economic planning and is one of the most suitable methods for this purpose. Forest and nature resource management planning is viewed as economic planning but with focus on the forest sector in terms of resources, means and goals.

LP is suitable for forest planning both through application of primal and dual model versions. Primal LP may serve the purpose of clarification of traditional management goals such as computation of production flows, cash-flows, net present value, and sustainability issues. Dual LP may fulfil the need for evaluation of non-market outputs from forestry which are usually not explicitly considered in a traditional forest planning context.

LP may be applied as a so-called stratified model where strata are composed of age class areas representing the forest resource. This is the traditional data structure applied for longterm forest planning, termed periodic planning. In periodic forest planning the length of the planning periods is usually equal to the width of age classes applied in the stratified model structure. The reason for selection of this model structure is linked to the need for long-term analyses in forest planning. The investment period under consideration in Danish forestry is usually from 50-150 years, that is the so-called rotation length. So, the age class structure of a forest resource is a very important and fundamental resource attribute.

The age class structure is, however, revealed through analyses of a so-called stand list. That is, a database containing area specific information. A stratum, which is diverse in an area context, often consist of a set of

area specific members, which may be located geographically. For example, age class 1 (0-9 years old) of beech may consist of all members in the stand list represented by stands of beech, aged 0-9 years. Age class 2 of beech consists of beech stands, aged 10-19 years, and so on. So, the basic information needed for preparation of a long-term periodic forest management plan is by tradition formulated as a database, a so-called stand list. Database management is required for utilisation of this information in a stratified LP model. Database management is, therefore, a central component of long-term forest management planning. The geographical link between the stand list and the forest map is, however, not expressed when a stratified planning model is applied.

Long-term forest planning requires consideration of a planning horizon of multiple decades. Here, strategic and tactical planning goals are often in focus. However, a forest plan also requires delineation of operational goals which should be measurable and controllable in terms of quantity, time scheduling, and geographical location. In this context, strategic planning is viewed as the frame for tactical planning which again functions as the frame for operational planning. Similarly, long-term planning functions as a frame for short-term planning. Short-term planning is in a forestry context synonymous with annual planning. Both long- and short-term planning may include strategic, tactical, and operational goals. Short-term planning is a prerequisite for implementation of a forest plan and functions as a guide for management decisions.

Short-term planning in forestry is area specific. Therefore, a socalled area based model is applied. The planning horizon may be one or multiple years. A planning model covering multiple years will facilitate achievement of model equilibrium between long-term and shortterm goals. +p-p-p-The database structure of a forest stand list is determined by historical traditions, but the structure is suitable as input to short-term planning. In practice, the stand list may be applied as input in a slightly modified or reduced version dependent on the purpose of the planning task. In this case, a number of treatment alternatives is formulated for each area unit. Each alternative is viewed as a so-called local decision variable. In addition, a number of so-called global decision variables is formulated for the whole forest estate or strata of this. The local decision variables are operational because they are measurable and controllable in terms of quantity, time scheduling, and geography.

The traditional database structure of a forest stand list corresponds to other industrial sectors' information storage. Information databases within other sectors of society have been utilised for development of decision support systems, some of which are based on application of geographical information systems (GIS). Many of these decision support systems are very powerful planning tools. However, data management and application of multiple planning criteria may be more efficiently integrated into models based on traditional operations research methods such as LP.

The dominating tool applied in forest planning in Denmark is GIS combined with simulation. Optimisation is not yet applied in practice, probably due to lack of know-how and education in the past decades. Optimisation is, however, expected to be integrated in practical forest planning. Therefore, it is relevant to explore the potential benefits associated with combined use of optimisation (for example LP) and GIS. Short-term forest planning applying either LP or GIS is based on use of the stand list in the form of a database. GIS provides a possibility for specific geographical analyses combined with information themes linked to the database. Information themes may be established by formulation of criteria. Various planning themes may be analysed by preparation of so-called theme maps, represented by members of the database which fulfil the criteria (for example stands with a mean tree diameter above 60 cm). In this way, the GIS system may function as a basis for formulation of management goals. An LP based short-term forest planning system may outline a plan for activity scheduling in forest stands. This optimisation may be subject to a set of constraints. If the short-term planning model is an area based model it is natural to use the GIS as a medium for geographical evaluation of the optimal plan. In other words, it is likely that a planning system based on combined and iterative use of GIS and optimisation could improve the planning results. We could utilise the eminent data and criteria management capabilities of optimisation in combination with the strong database and geographical capabilities of GIS. Development of a suitable simulation-optimisation interface is an inherent problem when LP is applied. The column generation phase of planning using LP is dependent on the available data structure and the purpose of the planning

process. Similarly, the most suitable GIS-LP interface is dependent on the data structure, data content, and the geographical constraints of the planning problem.

It is here the purpose to illustrate some perspectives of combined use of GIS and LP in forest and nature resource management planning, from a qualitative point of view. A small forest is selected as a case study area to illustrate the potential advantages of this model structure. Some of the primary quantitative planning goals within forest planning are net present value, cash-flow, volume production and increment, age class structure, diameter increment, wood quality, and these decision variables may be constrained by fulfilment of other biological, ecological, social, economic, technological, environmental, sustainability, geographical, political, or psychological goals. It is, therefore, possible to illustrate the need for incorporation of decision variables using both planning tools. The next step will be to develop a well functioning computer version of the interface. It is the intention to address this problem with the purpose to discuss possible solutions.

Polarity in DEA on Relaxed Convexity Assumptions

Jørgen Tind

Department of Operations Research
University of Copenhagen
tind@math.ku.dk

Per Agrell and Peter Bogetoft

Department of Economics
Royal Agricultural University, Copenhagen

This presentation discusses DEA-models under relaxed convexity conditions. Usually convexity is assumed for the entire production possibility set, and an efficiency measure is calculated by linear programming.

Here, convexity is only required for input and output projections of the possibility set. The projections are polyhedral and a polarity correspondence compares feasible input-output polyhedra in the original space as well as in a dual space. The polarity has roots in the blocking and anti-blocking theory originally developed for a linear programming investigation of certain problems in extremal combinatorics.

A mixed integer programming model is proposed for the calculation of a hierarchy of efficiency measures. Some preliminary experiments are done measuring the relative performance of 20 Danish research departments in economics.

Capacity Planning under Uncertain Demands in Telecommunications Networks*

Adam Ouorou

Dpt of Management Studies, University of Geneva
102 Bd Carl Vogt, CH-1211 Geneva 4, Switzerland
ouorou@hec.unige.ch

Jean-Philippe Vial

Dpt of Management Studies, University of Geneva
102 Bd Carl Vogt, CH-1211 Geneva 4, Switzerland
jean-philippe.vial@hec.unige.ch

A. Lisser

France Telecom, CNET
38-40 rue de Général Leclerc
F-92131 Issy-les-Moulineaux, France

This paper deals with the dimensioning of telecommunications networks offering private line services to a few clients. The clients ask for some transfer capacity between some pair of nodes, but their demand is uncertain. In case of high demand and insufficient capacity, some clients may be denied the transfer; the telecommunications operator pays a penalty cost for that.

The network has a fixed topology. In planning the network capacity, the operator wants to balance the investment cost with the total expected penalty cost. The planning situation is modeled as a stochastic programming problem. The scenarios are built under the assumption that the clients have independent demands. The solution method is based on Benders decompositon coupled with the analytic center cutting plane method. We solve some large size problem instances. For one problem instance, we perform sensitivity analysis and draw the trade-off cost curve vs the unitary penalty cost.

*Research funded by France Telecom/CNET

LAWRA – Linear Algebra With Recursive Algorithms*

Jerzy Waśniewski, Bjarne S. Andersen, Alexander Karaivanov

Danish Computing Centre for Research and Education, (UNI•C)
DTU, Building 304, DK-2800 Lyngby, Denmark
jerzy.wasniewski@uni-c.dk
bjarne.stig.andersen@uni-c.dk, alex@uni-c.dk

Fred Gustavson

IBM T.J. Watson Research Center
P.P. Box 218, Yorktown Heights, NY 10598. USA
gustav@watson.ibm.com

Plamen Y. Yalamov

Center of Applied Mathematics and Informatics
University of Rousse, 7017 Rousse, Bulgaria
yalamov@ami.ru.acad.bg

Abstract. Fortran 90 allows writing recursive procedures. Recursion leads to automatic variable blocking for dense linear-algebra algorithms. The recursive way of programming algorithms eliminates using BLAS level 2 during the factorization steps. For this and other reasons recursion usually speeds up the algorithms.
The Cholesky factorization algorithm for positive definite matrices, LU factorization for general matrices, and LDL^T factorization for symmetric indefinite matrices using recursion is formulated in this extended abstract.
The performance numbers of the recursive algorithms are measured on several computers, and will be explained on the lecture.

*This research was partially supported by the UNI•C collaboration with the IBM T.J. Watson Research Center at Yorktown Heights. The second and fourth author was also supported by the Danish Natural Science Research Council through a grant for the EPOS project (Efficient Parallel Algorithms for Optimization and Simulation). The last author was partially supported by the Grant I-702/97 from the Bulgarian Ministry of Education and Science.

1. Introduction

This work is a continuation of the work of Gustavson and Toledo described in [7],[11]. These papers describe the application of recursion to the numerical dense linear algebra algorithms. Recursion leads to automatic variable blocking for the dense linear-algebra algorithms. This is some modification of the LAPACK [1] algorithms. LAPACK's level-2 version routines are transformed into level-3 codes by using recursion.

Fortran 90 allows recursion (see [10]). The programs are simple and the recursion part is automatic as it is handled by the compiler. The intermediate subroutines have Fortran 90 standard too (see [5]).

Section 2 shows the recursive Cholesky factorization algorithm. Section 3 formulates the recursive algorithm of general Gauss factorization without pivoting and with partial pivoting. Section 4 demonstrates the factorization algorithm with the pivoting strategy introduced by Bunch-Kaufman for symmetric indefinite matrices (see [6, pages 161–170]).

2. Cholesky Factorization

We would like to compute the solution to a system of linear equations $AX = B$, where A is real symmetric or complex Hermitian and, in either case, positive definite matrix, X and B are rectangular matrices or vectors. The Cholesky decomposition should be used to factor A, $A = L\,L^T$ or $A = U^T U$, where U is an upper triangular matrix and L is a lower triangular ($L = U^T$). The factored form of A is then used to solve the system of equations $A\,X = B$.

The recursive algorithm of the Cholesky factorization is in detail described in [13], [7], [2]. Here we give the final recursive algorithms for the lower triangular and upper triangular cases. We assume that A is square.

Recursive Algorithm 2.1. *Cholesky recursive algorithm if lower triangular part of A is given (rcholesky):*

- *if* $n > 1$ *then*
  - $L_{11} :=$ *rcholesky of* A_{11}
  - $L_{21} L_{11}^T = A_{21} \;\rightarrow$ **RTRSM**
  - $\hat{A}_{22} := A_{22} - L_{21} L_{21}^T \;\rightarrow$ **RSYRK**
  - $L_{22} :=$ *rcholesky of* $\hat{A}_{22}$

- *otherwise* • $L := \sqrt{A}$

Recursive Algorithm 2.2. *Cholesky recursive algorithm if upper triangular part of A is given (rcholesky):*

- *if* $n > 1$ *then*
  - U_{11} := *rcholesky of* A_{11}
  - $U_{11}^T U_{12} = A_{12}$ $\rightarrow$ **RTRSM**
  - $\hat{A}_{22} := A_{22} - U_{12}^T U_{12}$ $\rightarrow$ **RSYRK**
  - U_{22} := *rcholesky of* $\hat{A}_{22}$
- *otherwise* • $U := \sqrt{A}$

The matrices A_{11}, A_{12}, A_{21}, A_{22}, L_{11}, L_{21}, L_{22}, U_{11}, U_{12} and U_{22} are submatrices of A, L and U respectively.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad L = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \text{ and } U = \begin{pmatrix} U_{11} & U_{12} \\ & U_{22} \end{pmatrix}$$

The sizes of the submatrices are: for A_{11}, L_{11} and U_{11} is $h \times h$, for A_{21} and L_{21} is $(n-h) \times h$, for A_{12} and U_{12} is $h \times (n-h)$, and for A_{22}, L_{22} and U_{22} is $(n-h) \times (n-h)$, where $h = n/2$. Matrices L_{11}, L_{22}, U_{11} and U_{22} are lower and upper triangular respectively. Matrices A_{11}, A_{12}, A_{21}, $A22$, L_{21} and U_{12} are rectangular.

The RTRSM and RSYRK are recursive BLAS of _TRSM and _SYRK respectively. _TRSM solves a triangular system of equations. _SYRK performs the symmetric rank k operations.

3. *LU* factorization

We would like to compute the solution to a system of linear equations $AX = B$, where A is real or complex matrix, and X and B are rectangular matrices or vectors. Gaussian elimination with row interchanges is used to factor A as $A = P^T L U$, where P is a permutation matrix, L is a unit lower triangular, and U is an upper triangular. The factored form of A is then used to solve the system of equations $AX = B$.

The recursive algorithm of the Cholesky factorization is in detail described in [3], [7]. We are giving the final recursive algorithms here.

Recursive Algorithm 3.1. *Recursive LU factorization without pivoting (rgausslu):*

- *if* $n > 1$ *then*
  - (L_1, U_1) = *rgausslu of* A_1
  - $L_{11} U_{12} = A_{12}$ $\rightarrow$ **RTRSM**
  - $\hat{A}_{22} = A_{22} - L_{21} U_{12}$ $\rightarrow$ **RGEMM**
  - (L_{22}, U_{22}) = *rgausslu of* $\hat{A}_{22}$
  - *if* $n > m$ *then* • $L\,U_3 = A_3$ $\rightarrow$ **RTRSM**
- *otherwise* • $L_1 := A_1/a_{11}$

The matrices A_1, A_2, A_3, A_{12}, A_{22}, L_1, L_{11}, L_{21}, L_{22}, U_1, U_3, U_{12} and U_{22} are submatrices of A, L and U respectively, $a_{11} \in A_1$.

$A = \begin{pmatrix} A_1 & A_2 & A_3 \end{pmatrix}$, and $A_1 = \{a_{ij} \mid i \in \{1,2,\ldots,m\}, j \in \{1,2,\ldots,h\}\}$, $A_2 = \{a_{ij} \mid i \in \{1,2,\ldots,m\}, j \in \{h+1,\ldots,n_0\}\}$, $A_3 = \{a_{ij} \mid i \in \{1,2,\ldots,m\}, j \in \{n_0+1\ldots,n\}\}$, and $n_0 = \min(m,n)$ and $h = n_0/2$. $A_{12} = \{a_{ij} \mid i \in \{1,2,\ldots,h\}, j \in \{h+1,\ldots,n_0\}\}$ and $A_{22} = \{a_{ij} \mid i \in \{h+1,\ldots,m\}, j \in \{h+1,\ldots,n_0\}\}$.

$L_1 = \{l_{ij} \mid i \in \{2,3,\ldots,m\}, j \in \{1,2,\ldots,\min(i-1,h)\}\}$, and $L_{11} = \{l_{ij} \mid i \in \{2,3,\ldots,h\}, j \in \{1,2,\ldots,i-1\}\}$, $L_{21} = \{l_{ij} \mid i \in \{h+1,\ldots,m\}, j \in \{1,2,\ldots,h\}\}$, $L_{22} = \{l_{ij} \mid i \in \{h+2,\ldots,m\}, j \in \{h+1,\ldots,h+i\}\}$.

$U_1 = \{u_{i,j} \mid i \in \{1,2,\ldots,h\}, j \in \{i,\ldots,h\}\}$, $U_3 = \{a_{ij} \mid i \in \{1,2,\ldots,m\}, j \in \{n_0+1\ldots,n\}\}$, $U_2 = \{a_{ij} \mid i \in \{1,2,\ldots,m\}, j \in \{h+1,\ldots,n_0\}\}$, and $U_{22} = \{u_{i,j} \mid i \in \{h+1,\ldots,h\}, j \in \{h+1+i,\ldots,m\}\}$.

Recursive Algorithm 3.2. *Recursive LU factorization with partial pivoting (rgausslu):*

- *if* $n > 1$ *then*
  - (P, L_1, U_1) = *rgausslu of* A_1
  - *Forward pivot* A_2 *by* P $\rightarrow$ **RLASWP**
  - $L_{11} U_{12} = A_{12}$ $\rightarrow$ **RTRSM**
  - $\hat{A}_{22} := A_{22} - L_{21} U_{12}$ $\rightarrow$ **RGEMM**
  - $(\hat{P}, L_{22}, U_{22})$ = *rgausslu of* $\hat{A}_{22}$
  - *Back pivot* A_1 *by* $\hat{P}$ $\rightarrow$ **RLASWP**
  - *updating of* P *by* $\hat{P}$
  - *if* $n > m$ *then*

- *Forward pivot* A_3 *by* P $\rightarrow$ **RLASWP**
- $A_3 = LU_3$ $\rightarrow$ **RTRSM**

- *otherwise*
  - *pivot* A_1
  - $L_1 := A_1/a_{11}$

where P *and* $\hat{P}$ *are permutation matrices.*

The RGEMM, RTRSM and RLASWP are recursive BLAS of _GEMM, _TRSM and _LASWP respectively. _GEMM multiplies two matrices adding the third matrix. _TRSM solves a triangular system of equations (see page 51). _LASWP performs a series of row interchanges on the matrix.

4. LDL^T Factorization for Symmetric Indefinite Matrices

It is well-known that the Cholesky factorization can fail for symmetric indefinite matrices. In this case some pivoting strategy can be applied (e. g. the Bunch-Kaufman pivoting [6, ,]Section 4.4). The algorithm can be given as follows.

Algorithm 4.1.

$L = I$ *(identity matrix);* $k = 1$;
while $(k < n)$
  Apply pivoting: choose a $1 \times 1 (s = 0)$, *or* $2 \times 2 (s = 1)$ *pivot block, and exchange the corresponding rows and columns;*
  $E = A_{k:k+s,k:k+s}$
  $C = A_{k+s+1:n,k:k+s}$
  $B = A_{k+s+1:n,k+s+1:n}$
  $L_{k+s+1:n,k:k+s} = CE^{-1}$
  $D_{k:k+s,k:k+s} = E$
  $A_{k+s+1:n,k+s+1:n} = B - CE^{-1}C^T$
  $k = k+s+1$
end

As a result we get

$$PAP^T = LDL^T,$$

where L is unit lower triangular, D is block diagonal with 1×1, or 2×2 blocks, and P is a permutation matrix.

Now let us look at the recursive formulation of this algorithm. This is given below. The recursion is done on the second dimension of matrix A, i. e. the algorithm works on full columns like in the LU factorization.

Recursive Algorithm 4.1. *Recursive Symmetric Indefinite Factorization (RSIF) of* $A_{1:m,1:n}$, $m = n$*:*

$k = 1$
if $(n = 1)$
  Define the pivot: 1×1, or 2×2.
  Apply interchanges if necessary
  $k = k + 1$, *or* $k = k + 2$
  If the pivot is 2×2: FLAG=1
else
  $n1 = n/2$
  $n2 = n - n1$
  RSIF *of* $A_{:,k:k+n1-1}$
  if *(FLAG = 1)*
    $n1 = n1 + 1$
    $n2 = n - n1$
  end
  update $A_{:,k:k+n2-1}$
  RSIF *of* $A_{:,k:k+n1-1}$
end

The LAWRA work on LDL^T is being going on now. More references will be given at the conference.

References

[1] E. Anderson, Z. Bai, C. H. Bischof, J. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. C. Sorensen. *LAPACK Users' Guide Release 2.0.* SIAM, Philadelphia, 1995.

[2] Andersen, B., Gustavson, F., Waśniewski, J.: A recursive formulation of the Cholesky factorization operating on a matrix in packed storage form, in: Parallel Processing for Scientific Computing, Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX , USA, March 24-27, 1999.

[3] Andersen, B., Gustavson, F., Waśniewski, J., Yalamov, P.: Recursive formulation of some dense linear algebra algorithms, in: Parallel Processing for Scientific Computing, Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX , USA, March 24-27, 1999.

[4] J.W. Demmel. *Applied Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

[5] J. Dongarra, and J. Waśniewski. *High Performance Linear Algebra Package – LAPACK90.* Lawn number 134: http://www.netlib.org/lapack/lawns/lawn134.ps Report UNIC-98-01, UNI•C, Lyngby, Denmark, 1998. Report ut-cs-98-384, University of Tennessee, Computer Science Department, Knoxville, April, 1998.

[6] G.H. Golub and C.F. Van Loan. *Matrix Computations (third edition).* Johns Hopkins University Press, Baltimore, MD, 1996.

[7] F.G. Gustavson. *Recursive Leads to Automatic Variable Blocking for Dense Linear-Algebra Algorithms.* IBM Journal of Research and Development, Volume 41, Number 6, November 1997.

[8] F.G. Gustavson, A. Henriksson, I. Jonsson, B. Kågström and P. Ling. *New Recursive Blocked Data Formats and New Recursive Blocked BLAS' for Dense Linear Algebra Algorithms.* This proceedings, 1998

[9] N.J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 1996.

[10] S. Metcalf and J. Reid. *Fortran 90/95 Explained.* Oxford, New York, Tokyo, Oxford University Press, 1996.

[11] S. Toledo. *Locality of Reference in LU Decomposition with Partial Pivoting.* SIAM Journal on Matrix Analysis and Applications, vol. 18, No. 4, 1997.

[12] L.N. Trefethen and D. Bau, III. *Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

[13] J. Wassśniewski, B.S. Andersen, and F. Gustavson. *Recursive Formulation of Cholesky Algorithm in FORTRAN 90,* in: Bo Kågström, E. Elmroth, J. Dongarra, and J. Waśniewski (Eds.), Applied Parallel Computing, Large Scale Scientific and Industrial Problems, Fourth International Workshop, PARA'98, Umeå, Sweden, July 1998, Proceedings, Lecture Notes in Computer Science, Springer-Verlag number 1541, 1998, pp. 574 – 578.

Logical Linear Programming

H. Paul Williams

University of Southampton, UK

H.P.Williams@soton.ac.uk

A Logical Linear Programme (LLP) is a model which involves maximising or minimising a linear expression over a LOGICAL combination of linear (in)equalities. Linear Programming (LP) is a special case where only a CONJUNCTION of constraints is allowed. For LLP we have a complete set of Boolean connectives (eg .and. , .or. , .not.).

This representationis an alternative to the usual approach of formulating logical problems as Mixed Integer Programmes (MIPs). The MIP representation can obscure some of the structure of the problem. There are few analogues of the mathematical properties of LP (eg the Duality Theorem,the Basis Theorem, Reflexive Duality etc).

LLPs preserve some of the structural properties of LP eg the Duality Theorem holds. Minimax algebra is a useful technique for representing the dual.It also leads to a generalised form of LLPs.