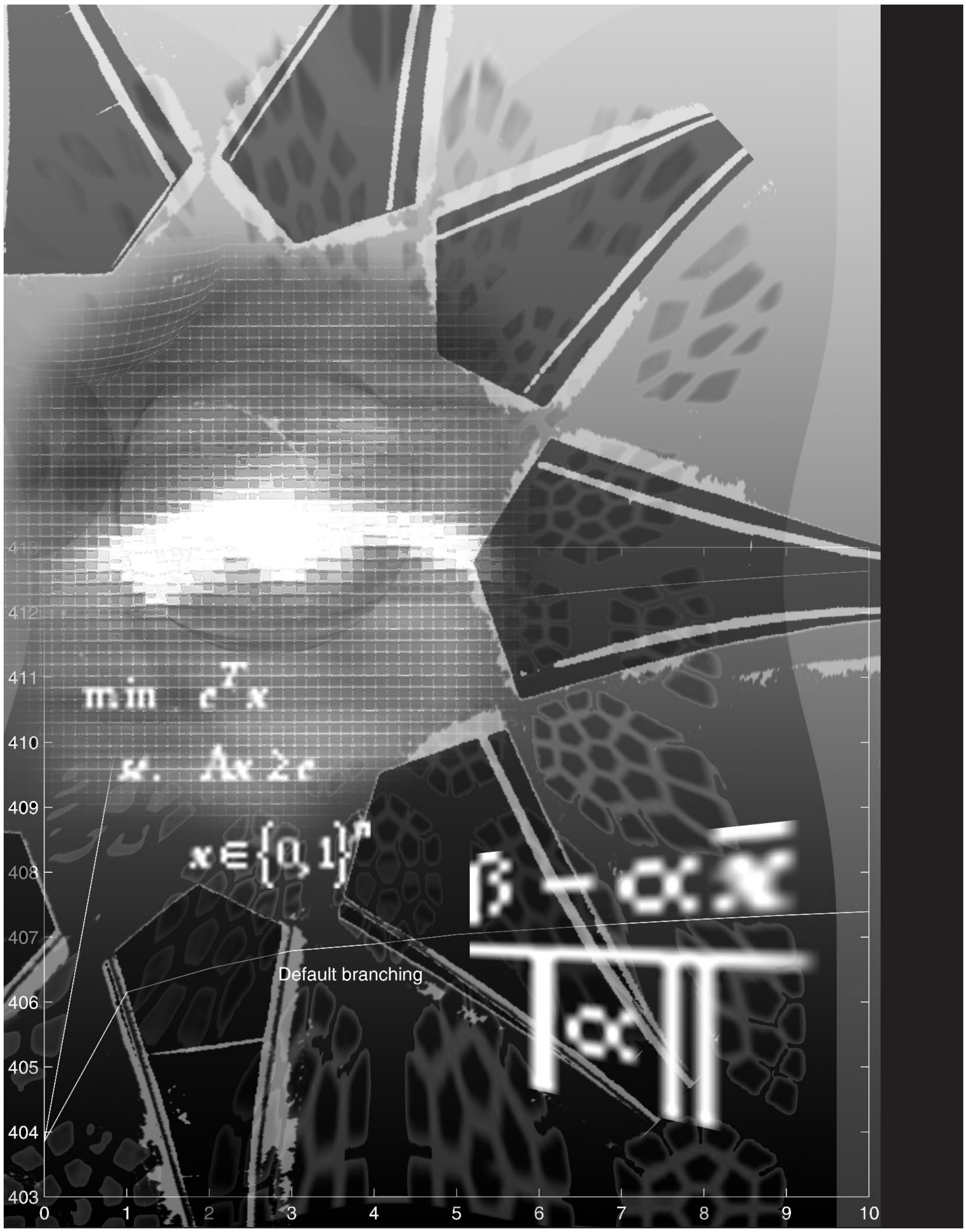
N°66

OPTIMA

Mathematical Programming Society Newsletter

OCTOBER 2001



Solving the *seymour* problem

Michael C. Ferris
Gábor Pataki
Stefan Schmieta

66

Solving the *seymour* problem[*]

Michael C. Ferris[†]
Gábor Pataki[‡]
Stefan Schmieta[§]

July 11, 2001

Optimization problems are at the heart of much of operations research and can vary substantially both in complexity and size. In many problems, the sheer size of the instance makes it very difficult to solve due to time or space limitations. In others, the complexity of the problem (nonlinearities, nonconvexities, or discreteness) can make it difficult or impossible to solve to optimality, even for reasonable sized instances. This note addresses an instance of the latter type of problem, arising as a mixed integer program (MIP) involving discrete variables and linear functions.

Hard problems in MIPLIB

The MIPLIB library of mixed integer programs was created in 1992 ([4]) and most recently updated in 1998 ([5]). Several problems in the library gained some notoriety, for being among the toughest. Some of these are:

- The *danoint* and *dano3mip* problems that arise from network design; the latter of these is unsolved to this date.
- The *markshare* problems, that were created with particular malice to challenge branch-and-bound, and cutting plane algorithms.
- The *seymour* problem: a relatively small setcovering problem with a fascinating origin, and of remarkable difficulty.

A group of researchers, consisting of the authors, of Sebastian Ceria at Columbia University, and Jeff Linderoth at Argonne National Laboratory has recently succeeded in solving the *seymour* problem. In this article, we describe why we found this problem so alluring, what experiments we have done, and eventually, what techniques led us to its solution.

Background on *seymour*

The *seymour* problem is a setcovering problem; i.e. a problem of the form

$$\begin{aligned} \min \quad & e^T x \\ st. \quad & \mathrm{A}x \geq e \\ & x \in \{0,1\}^n \end{aligned}$$

where e denotes a vector of all ones of appropriate dimension, and A is a matrix of zeros and ones. The number of rows in A is 4944 and the number of columns is 1372. The instance was posed by Paul Seymour, as a by-product of a new proof of the Four Color Theorem (FCT) by Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas [12, 13].

An interesting short history of this problem is given by these authors at [9] which we reproduce here verbatim.

The Four Color Problem dates back to 1852 when Francis Guthrie, while trying to color the map of counties of England noticed that four colors sufficed. He asked his brother Frederick if it was true that any map can be colored using four colors in such a way that adjacent regions (i.e. those sharing a common boundary segment, not just a point) receive different colors. Frederick Guthrie then communicated the conjecture to DeMorgan. The first printed reference is due to Cayley in 1878.

A year later the first 'proof' by Kempe appeared; its incorrectness was pointed out by Heawood 11 years later. Another failed proof is due to Tait in 1880; a gap in the argument was pointed out by Petersen in 1891. Both failed proofs did have some value, though. Kempe discovered what became known as Kempe chains, and Tait found an

*This material is based on research supported by the National Science Foundation Grants CDA-9726385, CCR-9972372 and DMS 95-27-124 and the Air Force Office of Scientific Research Grant F49620-01-1-0040

†Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706 (ferris@cs.wisc.edu)

‡Department of Operations Research, Smith Building, University of North Carolina, Chapel Hill, NC 27599 (gabor@unc.edu). This research was done while the author was at the Department of IE/OR, at Columbia University

§Axioma Inc. (sschmieta@axiomainc.com). This research was done while the author was at the Department of IE/OR, at Columbia University

equivalent formulation of the Four Color Theorem in terms of 3-edge-coloring.

The next major contribution came from Birkhoff whose work allowed Franklin in 1922 to prove that the four color conjecture is true for maps with at most 25 regions. It was also used by other mathematicians to make various forms of progress on the four color problem. We should specifically mention Heesch who developed the two main ingredients needed for the ultimate proof - reducibility and discharging. While the concept of reducibility was studied by other researchers as well, it appears that the idea of discharging, crucial for the unavoidability part of the proof, is due to Heesch, and that it was he who conjectured that a suitable development of this method would solve the Four Color Problem. This was confirmed by Appel and Haken in 1976, when they published their proof of the Four Color Theorem.

The web page also gives an outline of the proof of the theorem, and a longer list of pertinent references.

The *seymour* IP formulates the problem of finding the smallest unavoidable set of configurations that must be "reduced" in order to prove the FCT. Here "reduced" is a technical term meaning that the configuration can be shown not to exist in a minimal counterexample. Seymour has found a solution of value 423, but until this work, we are aware of no one who has been able to reproduce such a solution. The problem actually has many solutions of value 423.

The value of its LP relaxation is 403.84. Therefore, all one must do is raise the lower bound to say 422.0001 (or to better safeguard against numerical errors, to say 422.1) to prove the optimality of Seymour's solution; in fact, for a long time, we were aiming for 423.0001, as the best solution we could find was of value 424.

One may question the value of spending months of research effort trying to solve such hard IP's, which have no particular "realistic" application. We can argue though, that it is the small, and hard problems from which one can learn the most - and the techniques one develops through their study are very much applicable to real-world, difficult problems.

First attempts: branch-and-bound

The first attempt to solve *seymour* was done in 1996 by Greg Astfalk running CPLEX 4.0 with default settings on an HP SPP2000 with 16 processors, each processor having 180 MHz frequency, and 720 Mflops peak performance, for the total of approximately 58 hours, enumerating about 1,275,000 nodes, and using approximately 1360 Mbytes of memory. In this run, CPLEX did not even close 9 units of the gap; remember that we must close a bit more than 18.16 units.

We can do quite a bit better, just by using CPLEX, with the variable selection rule of *strong branching*. Strong branching (SB) was developed by Applegate, Bixby, Chvatal and Cook in their work on the TSP, and it is an available setting in several commercial MIP solvers now. At every node of the branch-and-bound tree, SB tests several variables as a candidate to branch on (by partially reoptimizing on both branches with a limited number of dual simplex pivots), and picks the most promising one.

Figure 1 shows what lower bound the CPLEX 6.0 branch-and-bound code has achieved after enumerating a hundred thousand nodes by using default branching variable selection vs. SB variable selection (all other settings were default). On the horizontal axis one mark means 10 thousand branch-and-bound nodes. The run with SB closed nearly 9 units of the gap, and took about a week on a 337 MHz speed machine.

Cutting

Disjunctive cuts were introduced by Balas in the seventies [1], then rediscovered from a different viewpoint in the nineties [11, 14, 2, 3]. They were termed lift-and-project cuts and computationally tested in the nineties by Balas, Ceria and Cornuéjols in [2, 3]. For our experiments, we used the more recent implementation described in [6].

Here we only give a description of disjunctive cuts in a nutshell. Given P, the linear programming relaxation of a 0-1 program, and a variable x_i, the inequality $\alpha x \geq \beta$ is a 0-1

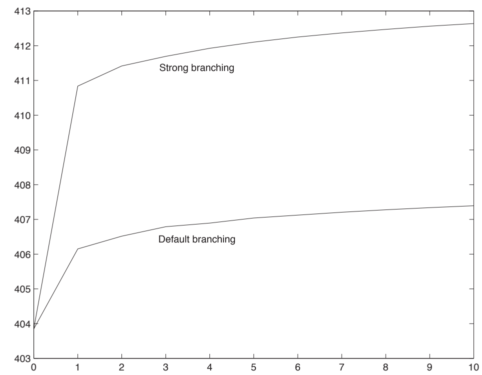


Figure 1: Strong branching vs. default branching on *seymour*

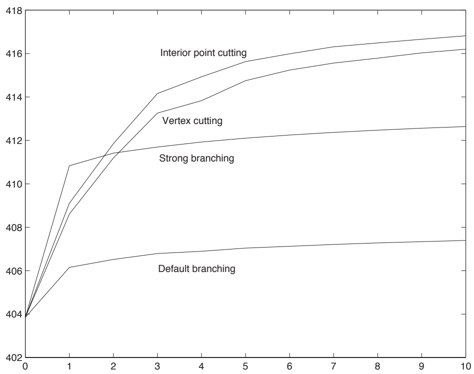


Figure 2: Cutting with 2 options vs. branch-and-bound with 2 options

disjunctive cut from the disjunction $x_i = 0 \vee x_i = 1$, if it is valid for both of the sets $P \cap \{x| \, x_i = 0\}$ and $P \cap \{x| \, x_i = 1\}$. From among all such inequalities, following the method developed in [2], one generates the cut which is violated by the current LP optimum $\bar{x}$ by the largest possible amount, i.e. $\beta - \alpha\bar{x}$ is maximized, with respect to some normalization constraint. Generating such a cut is done by solving an LP. Disjunctive cuts (as all cuts in MIP) are added to the LP formulation in rounds; i.e. one adds a batch of cuts, reoptimizes the LP, drops all nonbinding cuts, then repeats the procedure.

According to our experience, it is easy to tell, whether it is worth applying disjunctive cuts on a particular IP, just by looking at the result of two branch-and-bound runs. Disjunctive cuts work (i.e. adding them significantly raises the LP lower bound) if (and one can say, only if) strong branching works! The informal explanation is that both techniques attempt to enhance the effect of the branching operation. Strong branching does this by selecting the best variable to branch on. Generating disjunctive cuts from say 50 variables mimics the effect that can be gained from branching on those variables (of course, adding these 50 cuts will not result in a lower bound as good as the one from a 50 level deep branch-and-bound tree).

In our first experiment, we generated 10 rounds of 100 cuts, by selecting the 100 variables that were the most fractional in the current LP optimum. This run took about 6 hours, and closed about 9.45 units of the gap!

After some experimentation, we produced a formulation called *Formulation 1*, that we thought was worth trying to finish off with branch-and-bound. The setup was as follows:

- In each cutting iteration we generated cuts from *all* the fractional variables; there were typically about 600 of these.
- We sorted the cuts, by putting the one first from which the euclidean distance of the LP optimal solution is the largest, and so on. The distance of the hyperplane $\{x \mid \alpha x = \beta\}$ from the point $\bar{x}$ is

$$\frac{\beta - \alpha\bar{x}}{\|\alpha\|}.$$

Then, assuming that we have c cuts, we perform the following step for $i = 1, \ldots, c$:

  – If the cosine of the angle of i^{th} cut hyperplane with any one of the first $i - 1$ cut hyperplanes is greater than 0.999, we discarded the i^{th} cut.

From the remaining cuts we picked the first 250, and added them to the LP formulation.

We call the above method *cut selection by distance*. Another method that is quite natural is called *cut selection by usage*; we describe this method next. Suppose again, that we would like to select the "best" 250 cuts to be added to the LP formulation. We tentatively add all of them, then track the course of the reoptimization by the dual simplex algorithm using the steepest edge pivot rule. Whenever a cut is pivoted on, we mark it. We let dual simplex run, until 250 cuts get marked; these will be the selected ones. We never unmark a cut, and whether a cut is pivoted on once, or more than once does not matter. Perhaps surprisingly, we found that out of more than 600 cuts, each of which is violated by the current solution, we could never choose more than about 350 with this method – at most this many are *ever* pivoted on in the course of the reoptimization!

Cut selection by distance and by usage performed quite similarly on the *seymour* problem; it would be interesting to see how they work on other hard IP's, especially, when more than one type of cut (e.g. knapsack, flow-cover, Gomory-cuts) is used.

Figure 2 depicts the progress of the cutting plane algorithm with two different settings versus branch-and-bound with default branching and strong branching. The cutting strategy that worked best was cutting off an LP solution in the interior of the optimal face, as opposed to the usual vertex cutting. On the horizontal axis one mark means 10 thousand enumerated nodes for branch-and-bound, and one round of cutting for lift-and-project cuts. The progress made by branch-and-bound and cutting planes is quite well comparable this way, even though the time taken by the four different algorithms between two tickmarks on the horizontal axis can be different of course. For example branch-and-bound with SB took about a week to enumerate 100 thousand nodes, while with default branching it took only 3 days. The most time-consuming was cutting with the "interior point cutting" option; this took about 2 weeks. Still, in the case of *seymour* the question is simply solving it, or not; hence the few days difference in the running times is irrelevant in this case. From Figure 2 it is clear that after 100 thousand nodes, or 10 rounds, both algorithms completely "ran out of steam"; even after several more months, or years they would not solve the problem.

Formulation 1 was fed to the CPLEX branch-and-bound solver, again with the well-tested SB setting. After about 100 thousand nodes, the lower bound was further pushed up by about 3 units, for the total of about 15 units; at that point it was clear, that this way we will never solve the problem. At the same time, it also became clear that producing a limited number of nodes in a *branch-and-cut tree*, each with at least 15-16 units of the gap closed, would do the trick; we would simply need to process those nodes by branch-and-bound afterwards. Hence we set up a run to generate the required nodes, in which a certain number of cutting rounds was followed by branching for a number of levels in the branch-and-bound tree, then the process repeated. Precisely, we

- Generated 10 rounds of cuts at the root node.
- Ran B&B for 4 levels.
- At each of the $2^4 = 16$ nodes, we generated 2 more rounds of cuts.
- Ran B&B for 4 levels.
- At each of the $2^8 = 256$ nodes, we generated 1 more round.

That is, at the end we had $2^8 = 256$ nodes in the tree, and on the way from the root to any one of them 13 rounds of cuts were generated. We used SB for the variable selection; interior-point cutting, and selecting 250 cuts by usage for cut generation. We remark that all cuts generated within the tree were globally valid, i.e. they were used at the other nodes as well.

In the end, the gap closed at

- the best node was: 16.77
- the worst: 15.17
- the median: 16.29

We remark that the problem was preprocessed at the root node by deleting all dominated rows and columns as usual in setcovering problems. The reduced problem has 4323 rows, and 882 columns; the IP value of 423 in the original problem corresponds to the value of 238 in the preprocessed problem. Although in the parallel processing of the nodes we had to set the cutoff values to take into account the preprocessing, we translated these values back to correspond to the original instance.

The Condor system

Heterogeneous clusters of workstations are becoming an important source of computing resources. One approach to use these clusters of machines more effectively allows users to run their (computing intensive) jobs on idle machines that belong to somebody else. The Condor system [10, 8] that has been developed at University of Wisconsin-Madison is one scheme that manages such resources in a local intranet setting. It monitors the activity on all participating machines, placing idle machines in the Condor pool, that are allocated to service job requests from users. Users' programs are allowed to run on any machine in the pool, regardless of whether the user submitting the job has an account there or not. The system guarantees that heavily loaded machines will not be selected for an application.

Machines enter the pool when they become idle, and leave when they get busy, e.g. the machine owner returns. To protect ownership rights, whenever a machine's owner returns, Condor immediately interrupts any job running on that machine, migrating the job to another idle machine. In fact, the running job is initially suspended in case the executing machine becomes idle again within a short timeout period. If the executing machine remains busy, then the job is migrated to another idle workstation in the pool or returned to the job queue. For a job to be restarted after migration to another machine a checkpoint file is generated that allows the exact state of the process to be re-created. This design feature ensures the eventual completion of a job. In order to use the checkpoint feature, the job to be executed must just be relinked before being submitted to the Condor manager. An additional benefit of this relinking is that remote I/O can be performed on the submitting machine, therefore limiting the footprint of the job on the executing machine.

There are various priority orderings used by Condor for determining which jobs and machines are matched at any given instance. A job advertises its requirements via the simple mechanism of a "job description file". This file informs Condor of the location of the executable and the input and output files, along with the required architecture, operating system and memory needs of the job. A machine similarly advertises its properties and a matching scheme (implemented within the resource manager) pairs jobs to machines. Based on the priority orderings, running jobs may sometimes be preempted to allow higher priority jobs to run instead. Condor is freely available and has been used in a wide range of production environments for more than ten years.

Since the CPLEX suite of optimization procedures comes in library form, it is very easy to carry out the relinking of a simple driver program to run the *seymour* problem. On June 23, 1999, we submitted two separate CPLEX 6.0 jobs in an attempt to solve Formulation 1 described above. Both were set to run in depth first search mode to ensure the size of the stored branch and bound tree did not exceed the memory of the machines on which it ran. A cutoff value was set that excludes the presumed optimal solution by 1. In one job, the remaining parameters to CPLEX were set to default values, while in the other job, strong branching was carried out. At the time of writing this article, both jobs are still running. Condor has provided both jobs with over 600 days of CPU time in the ensuing two years.

One of the jobs has explored over 13.4 million nodes of the tree, while the other has processed close to 2.5 million nodes. As expected, neither has found a solution that exceeds the cutoff value, and furthermore, the lower bound has remained essentially stagnant for the large part of this time. Clearly, just applying brute force execution time to this problem is not going to solve it. However, it is interesting to note the reliability of both the Condor and CPLEX systems to be able to continue executing on a variety of different machines during this two year period.

One issue about the above computation is that each execution is limited to one processor. While it would be possible to use the parallel version of CPLEX to increase resources applied to the solution, it is not at all clear whether the parallel code would be able to run in the Condor environment.

Processing the nodes on Condor

Instead, we submitted the 256 IP subproblems described above, as 256 separate tasks. While the efficiencies generated by intercommunication between these tasks would be lost, the extra processing available at the root nodes of all the tasks that was described above was thought to more than compensate for this loss. Furthermore, any collection of resources could be used to solve these 256 instances, involving state-of-the-art commercial packages.

MPS input files for all 256 subproblems can be found at [7]. Each problem is listed with the ID of the corresponding node in the branch-and-bound tree. The file you get by clicking on the link will be called “node.mps”. The nodes are sorted by lower bound, which is computed as the value of the LP relaxation.

These problems were processed using CPLEX 6.6 and XPRESS 11.50. 219 of the problems were solved using CPLEX via Condor at Wisconsin. The remainder were processed using XPRESS 11.50 and CPLEX 6.6 at Columbia.

In general, we used 423.01 as the upper cutoff for the solvers, since at the outset of this work we were somewhat skeptical regarding the existence of a solution of value 423; we did have one with value 424 though. On July 4, 2000, we did find a solution with 423, after this the remaining subproblems were set up with an upper cutoff of 422.01. In the end, we were able to generate several solutions of value 423. The electronic citation [7] gives the binary variables that take on value 1 in two distinct optimal solutions.

For the 219 jobs that were run under Condor, the total CPU time used to process them all was 443.6 days, with 41.7 days idle time for jobs waiting in the Condor queue. During this time 10,244,500 nodes were explored using 3,261,696,402 pivots running on a total of 883 different machines. The longest single node took 36 days to complete, and the shortest completed in just under 53 minutes. At Columbia, a further 48.9 CPU days were used to explore 934,868 nodes. The actual elapsed time between starting the process and ending the process was 37 days, starting in June 2000 and ending on July 26, 2000.

Acknowledgement

Thanks are due to Oktay Günlük for helpful comments on a draft of this paper.

References

[1] E. Balas, Disjunctive programming, *Annals of Discrete Mathematics 5* (1979) 3-51.

[2] E. Balas, S. Ceria and G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0-1 programs, *Mathematical Programming 58* (1993) 295-324.

[3] E. Balas, S. Ceria and G. Cornuéjols, Mixed 0-1 Programming by lift-ad-project in a branch-and-cut framework, *Management Science 42* (1996) 1229-1246.

[4] R. E. Bixby, E. A. Boyd, and R. R. Indovina, MIPLIB: A Test Set of Mixed Integer Programming Problems, *SIAM News* 25 (1992).

[5] R. E. Bixby, S. Ceria, C. M. McZeal, M. W. P. Savelsbergh, An Updated Mixed Integer Programming Library: MIPLIB 3.0, *Optima 54* (1998), 12-15.

[6] S. Ceria and G. Pataki, Solving Integer and Disjunctive Programs by Lift-and-Project, *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization* (1998) 271-283.

[7] http://firulete.ieor.columbia.edu/~schmieta/seymour.html, List of 256 nodal subproblems.

[8] http://www.cs.wisc.edu/condor, The Condor Project, High Throughput Computing.

[9] http://www.math.gatech.edu/~thomas/FC/fourcolor.html, The four color theorem.

[10] M. J. Litzkow, M. Livny, and M. W. Mutka, Condor: A hunter of idle workstations, In *Proceedings of the 8th International Conference on Distributed Computing Systems*, pages 104-111, June 1988.

[11] L. Lovász and A. Schrijver, Cones of matrices and set-functions and 0-1 optimization, *SIAM Journal on Optimization 1* (1991), 166-190.

[12] N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, The four colour theorem, *J. Combin. Theory* Ser. B. 70 (1997), 2-44.

[13] N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, A new proof of the four colour theorem, *Electron. Res. Announc. Amer. Math. Soc. 2* (1996), 17-25 (electronic).

[14] H. Sherali and W. Adams, A hierarchy of relaxations between the continuous and convex hull representation for zero-one programming problems, *SIAM Journal on Discrete Mathematics 3* (1990) 411-430.

ISMP 2003: Bridging theory and practice

ISMP 2003 – the 18th International Symposium on Mathematical Programming – will take place in Copenhagen, Denmark on August 18-22, 2003. Being the largest meeting on Mathematical Programming, the symposium will cover numerous research areas and applications, illustrating the cross-disciplinarity and creativity which has characterized this exciting field for more than half a century.

The conference is organized by the Technical University of Denmark (DTU), in cooperation with the University of Copenhagen. Both universities have a long tradition in convex and combinatorial optimization. The Operations Research Section at the Department of Informatics and Mathematical Modelling (DTU) conducts research in areas as logistics, transport optimization, vehicle routing, facility location, production and inventory planning, timetabling and crew scheduling. The Institute for Mathematical Sciences at the University of Copenhagen has research activities in econometrics, mathematical finance, numerical analysis and operations research. Finally, at the Department of Computer Science, University of Copenhagen, algorithms for solving combinatorial optimization problems are studied, including cutting/packing problems, network/VLSI design problems and problems in computational geometry and biology.

With the bridge to Malmo, opened in the spring 2000, the universities in Copenhagen have become closely connected to the Swedish counterparts in Lund and Malmo. This means that the region – officially called the Sound region – has one of the worlds largest concentrations of universities in driving distance from the center of Copenhagen. As expected, this has caused a considerable synergetic effect to the research activities, and several international companies have opened research and development centers in the region to benefit from this concentration of competence.

The main elements of the ISMP 2003 logo is the Copenhagen-Malmo bridge and the Petersen Graph. The bridge has been chosen to symbolise the cross-disciplinarity of ISMP 2003, and also to illustrate the constructive mixture between theory and practice which is characteristic of the region. Julius Petersen (1839-1910) was a teacher at the Polytechnical School in Copenhagen (now DTU) and became later professor at the University of Copenhagen. He is considered to be one of the founders of graph theory which plays an important role in modeling and solving mathematical programming problems.

Although the symposium should be a sufficient motivation for visiting Copenhagen, the city has indeed numerous new attractions to offer. In the harbour area several monumental sites have recently been constructed or are on the drawing board: The extension of the National Library (also called the "black diamond"), the forthcoming Opera, the new city area at Holmen – located at an historic navy base – and the largest off-shore wind turbine park. You may explore Copenhagen by free bicycles, which are available all around the city center. Tivoli, the Little Mermaid and Nyhavn are some of the well-known sights you will meet on your ride.

Due to the well-established infrastructure, travelling to Copenhagen is an easy task. From Copenhagen Airport, which is the largest airport in Scandinavia with several transcontinental departures daily, the city center is reached by train in just 10 minutes. The Copenhagen Metro will open by 2002, providing a fast means of transportation in central Copenhagen.

For further information on ISMP 2003 please see the conference website www.ismp2003.dk, or contact the organizers:

ISMP 2003, c/o Prof. Jens Clausen
Informatics and Mathematical Modelling
Technical University of Denmark, Building 321
DK-2800 Lyngby, Denmark Fax: +45 4588 1397 e-mail: jc@imm.dtu.dk

David Pisinger, Professor, Ph.D.
DIKU, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen O
DENMARK
phone:+45 35 32 13 54
fax:+45 35 32 14 01
e-mail: pisinger@diku.dk
http://www.diku.dk/~pisinger

IPCO 2002
Ninth Conference on Integer Programming and Combinatorial Optimization

May 27-29, 2002
MIT, Cambridge, MA

Announcement and Call for Papers This meeting, the ninth in the series of IPCO conferences held every year in which no International Symposium on Mathematical Programming takes place, is a forum for researchers and practitioners working on various aspects of integer programming and combinatorial optimization. The aim is to present recent developments in theory, computation, and applications of integer programming and combinatorial optimization. Topics include, but are not limited to: approximation algorithms, branch and bound algorithms, computational biology, computational complexity, computational geometry, cutting plane algorithms, Diophantine equations, geometry of numbers, graph and network algorithms, integer programming, matroids and submodular functions, on-line algorithms, polyhedral combinatorics, scheduling theory and algorithms, semidefinite programming. In all these areas, IPCO welcomes structural and algorithmic results, revealing computational studies, and novel applications of these techniques to practical problems. The algorithms studied may be sequential or parallel, deterministic or randomized. During the three days, approximately thirty-six papers will be presented, in a series of sequential (non-parallel) sessions. Each lecture will be thirty minutes long. The conference proceedings will contain full texts of all presented papers. Copies will be provided to all participants at registration time.

Summer School A Summer School on Integer Programming and Combinatorial Optimization shall precede IPCO 2002. Three leading researchers will present three lectures each, complemented by exercises. The summer school will take place on May 25-26, 2002 at MIT. PhD students, postdocs and others interested in participating are encouraged to pre-register informally by sending e-mail to ipco2002@mit.edu.

Important Dates Deadline for submission of extended abstract: October 29, 2001; notification of acceptance: January 21, 2002; final manuscript due: February 22, 2002.

Information Further information about the conference and the summer school is available at http://mit.edu/ipco2002/.

Organizers William J. Cook, Princeton University (program committee chair), Andreas S. Schulz, Massachusetts Institute of Technology (organizing committee chair).

Contact Address
Andreas S. Schulz
IPCO 2002
Massachusetts Institute of Technology
Building E53-361
77 Massachusetts Avenue
Cambridge MA 02139-4307 USA
E-mail: ipco2002@mit.edu

7th International Symposium on
Generalized Convexity/Monotonicity

Hanoi/Vietnam, August 27-31, 2002

Scope Various generalizations of convex functions have been introduced in areas such as mathematical programming, economics, management science, engineering and applied sciences. In addition, different kinds of generalized monotonicity have been proposed, for instance for variational inequalities and equilibrium problems. Such models are considerably more adaptable to real-world situations than their convex/monotone counterpart. A growing literature in this interdisciplinary field has appeared, including the proceedings of the preceding six international symposia since the NATO Advanced Study Institute in 1980 in Vancouver, Canada. The symposium is organized by the international Working Group on Generalized Convexity (http://genconv.ec.unipi.it). It is the first symposium in the series that will take place in the Asia-Pacific region. The conference is sponsored by the Pacific Optimization Research Activity Group (POP).

Program Committee J.E. Martinez-Legaz (Spain) (co-chair), P.H. Sach (Vietnam) (co-chair), R.Cambini (Italy), J.-P. Crouzeix (France), A. Eberhard (Australia), N. Hadjisavvas (Greece), S. Komlosi (Hungary), D.T. Luc (Vietnam, France), S. Schaible (USA).

Organizing Committee N.D. Cong (Vietnam) (co-chair), D.T. Luc (Vietnam, France) (co-chair), V.N. Chau (Vietnam), P.H. Dien (Vietnam), N. Hadjisavvas (Greece), S. Komlosi (Hungary), L.D. Muu (Vietnam), H.X. Phu (Vietnam), N.X. Tan (Vietnam), N.D. Yen (Vietnam).

International Advisory Committee The two committees are supported by an international group of researchers in the field of study, representing in particular many countries of the Asian-Pacific region.

Invited Speakers J.M. Borwein (Burnaby), R.E. Burkard (Graz), B. Mordukhovich (Detroit), H. Tuy (Hanoi).

Location. Hanoi Institute of Mathematics, Hanoi, Vietnam.

Language English.

Fee 100 US $

Proceedings Edited proceedings including the invited lectures and a selection of contributed talks will be published. Co-editors: A. Eberhard, N. Hadjisavvas, D.T. Luc.

Preliminary Registration Form Please provide your name, title, postal and e-mail address as well as a tentative title of your proposed contributed talk (if applicable).

Contact Information
E-mail: gcm7@thevinh.ncst.ac.vn,
dtluc@univ-avignon.fr, jemartinez@selene.uab.es;
Web site: http://genconv.ec.unipi.it;

Mailing address:

Professor Nguyen Dinh Cong,
Institute of Mathematics,
P.O.Box 631,
Boho, 10000 Hanoi, Vietnam;

Professor Dinh The Luc,
Departement de Mathematiques,
Universite d'Avignon,
33 rue Louis Pasteur,
84000 Avignon, France.

FIRST ANNOUNCEMENT

ICOTA'2001
The 5th International Conference on Optimization: Techniques and Applications

December 15-17, 2001, Hong Kong

THEME AND SCOPE: The 5th International Conference on Optimization: Techniques and Applications (ICOTA), jointly organized by The Chinese University of Hong Kong, The City University of Hong Kong, and The Hong Kong Polytechnic University, will be held in Hong Kong in 2001. It is a continuation of the ICOTA series, which has had its first four conferences held in Singapore (1989 and 1992), Chendu, China (1995), and Perth, Australia (1998).

The 5th ICOTA represents the first of the ICOTA series in the new millennium, and has been given the theme "Optimization for the New Millennium". The goal of the 5th ICOTA is to provide an international forum for scientists, researchers, software developers, and practitioners to exchange ideas and approaches, to present research findings and state-of-the-art solutions, to share experiences on potentials and limits, and to open new avenues of research and developments, on all issues and topics related to optimization.

Plenary Speakers: Shu-Cherng Fang of North Carolina State University; Masao Fukushima of Kyoto University; Toshihide Ibaraki of Kyoto University; David G. Luenberger of Stanford University; Angelo Miele of Rice University; Panos M. Pardalos of University of Florida; and Yinyu Ye of University of Iowa.

Papers on issues related to optimization are welcome. Topics include (but not limited to) those in the following tracks:
· Optimization theory
· Algorithms analysis and design
· Applications in industry, service, finance, business and military.

MPORTANT DATES:

August 1, 2001	Deadline for paper submissions
September 1, 2001	Notification of acceptance
October 1, 2001	Camera-ready due and late registration fee applies
December 15-17, 2001	ICOTA'01, Hong Kong

CONTRIBUTED PAPER SUBMISSION GUIDELINES: Please submit your complete paper (or an extended abstract) to any of the three Program Committee Co-Chairs:
Program Co-Chair (North America)
Professor John R. Birge
E-mail: jrbirge@nwu.edu
Program Co-Chair (Asia and Rest of the World)
Professor Duan Li
Email: dli@se.cuhk.edu.hk
Program Co-Chair (Europe)
Professor Henry Wu
E-mail: q.h.wu@liv.ac.uk

GENERAL CONFERENCE CO-CHAIRS:
X.Q. Cai
Chinese University of Hong Kong,
Email: xqcai@se.cuhk.edu.hk

L. Qi
The Hong Kong Polytechnic University,
Email: maqilq@polyu.edu.hk

CONFERENCE WEB SITES:
www.se.cuhk.edu.hk/~icota
www.polyu.edu.hk/~ama/events/conference/icota

mind sharpener

We invite OPTIMA readers to submit solutions to the problems to Robert Bosch (bobb@cs.oberlin.edu). The most attractive solutions will be presented in a forthcoming issue.

Two Domino Problems

Robert A. Bosch
August 27, 2001

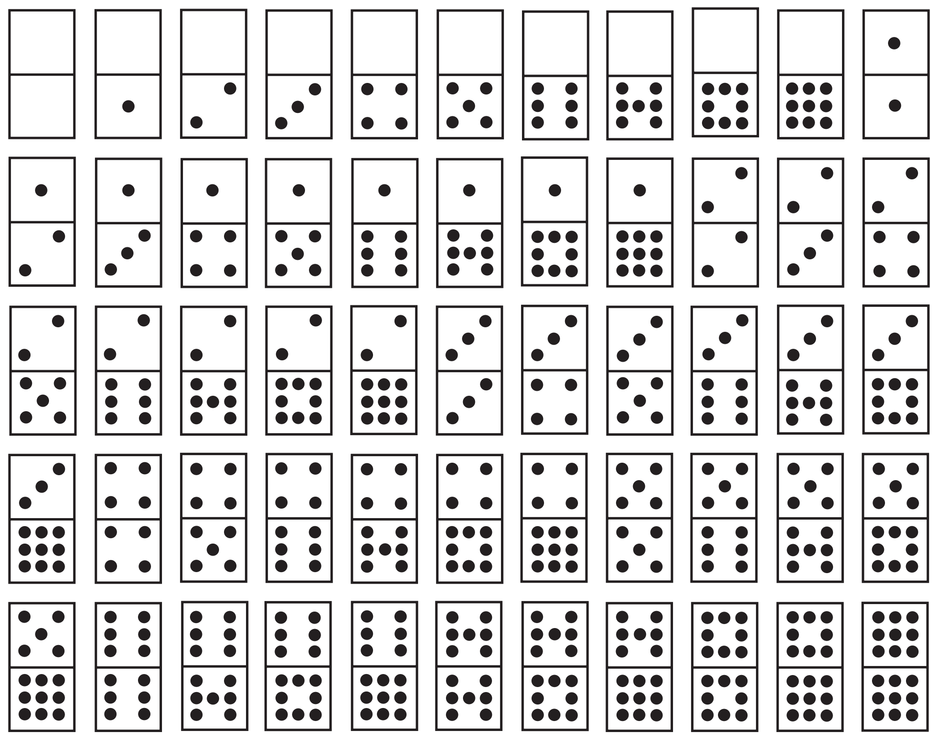


Figure 1 displays a complete set of "double nine" dominoes.
Each domino is one sided and can be used horizontally or vertically.

Problems

Interested readers may enjoy trying to solve the following problems:

1.Use a complete set of double nine dominoes to construct a replica of the abstract picture displayed in Figure 2. Note that the 5-7 domino can be placed (horizontally) in the upper left corner, but it cannot be made to fit in the lower left corner.

2.Using three complete sets of double nine dominoes, construct the "best possible approximation" of Leonardo DaVinci's Mona Lisa, as displayed in Figure 3. Incidentally, in 1993 the artist Ken Knowlton created a portrait of Scientific American's "Mathematical Games" columnist Martin Gardner using nine complete sets of double nine dominoes.
To see the portrait, point your browser to www.artists-nh.com/knowlton.htm

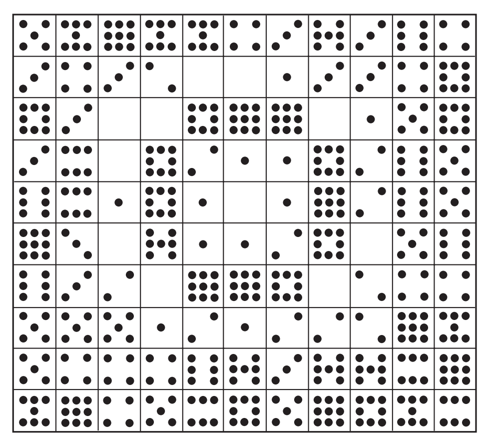


Figure 2

Painting by numbers revisited

Benoit Rottembourg reports that Constraint Programming works well on paint-by-numbers puzzles:

I'm in charge of the corporate OR department of Bouygues, a French holding company. We studied (enjoyably) with two students (F. Pernias and F. Buscaylet) your funny little mindsharpener published in *Optima* 65.

Our team has a tradition in Constraint Programming (we have our own CP solver, Choco, which is like ILOG's Solver) and naturally we formulated your problem in a very straightforward manner in CP, yielding fast solutions on the dragonfly example. We also tested some of the most difficult instances on the web site you mentioned, and in each case we obtained the solution in a few seconds.

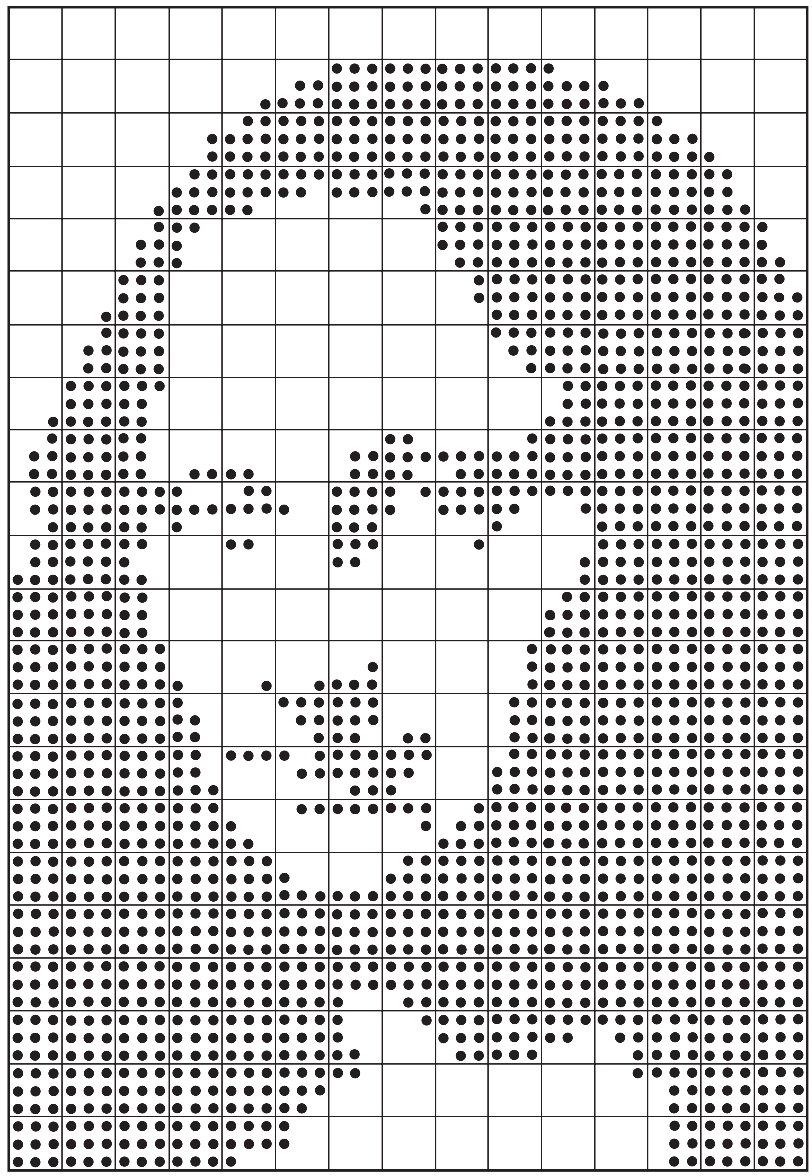


Figure 3

Professor in Applied Mathematics at the Swiss Federal Institute of TechnologyLausanne (EPFL)

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

The EPFL plans a substantial expansion in the basic sciences, including a significant reinforcement in mathematics, physics, and chemistry, and a major new effort in the life sciences.

As part of this broad program, the Mathematics Department has an opening at the full professor level. Applications for appointments at the Associate and Assistant Professor (tenure-track) levels will also be considered. We seek outstanding individuals in all areas of applied mathematics.

Applications in discrete mathematics and statistics are particularly encouraged. Successful candidates must develop an independent, internationally recognized program of scholarly research and must be willing to teach at both the undergraduate and graduate level. Substantial start-up resources will be provided. Women candidates are strongly encouraged to apply.

More information about EPFL and its Department of Mathematics at http://www.epfl.ch and http://dmawww.epfl.ch.

Applications, including CV, publication list, concise statement of research interests (3 pages or less) and three letters of reference, should be sent to:

Professor Gerard Ben Arous
Chairman of the Search Committee
Department of Mathematics
Ecole polytechnique fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Professor in Applied Mathematics
at the Swiss Federal Institute of Technology Lausanne (EPFL)

CALL FOR PAPERS

MATHEMATICAL PROGRAMMING Series B

Mathematical Programming in Biology and Medicine

We invite research articles for a forthcoming issue of Mathematical Programming, Series B, on the applications of mathematical programming methodology and techniques to the field of biology and medicine. For example, optimization problems and solutions in computational biology would be an area of interest among many others. Also possible are survey papers that give in-depth introduction to areas of biology and medicine where the use of mathematical programming is novel and promising. Our hope is that this special issue would bring the two fields a little closer.

Deadline for submission of full papers: February 28, 2002. We aim at completing a first review of all papers by August 31, 2002.

Electronic submissions in the form of postscript files are encouraged. All submissions will be refereed according to the usual standards of Mathematical Programming. Information about this issue can be obtained from the guest editors for this volume or at www.caam.rice.edu/~yzhang/mpb/

Guest Editors:

Michael C. Ferris
Department of Computer Sciences
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706, USA
Email: ferris@cs.wisc.edu

Yin Zhang
Department of Computational and Applied Mathematics
Rice University, MS-134
Houston, Texas 77005, USA
Email: yzhang@caam.rice.edu

CALL FOR PAPERS

4th Workshop on Algorithm Engineering and Experiments

ALENEX 02

January 4-5, 2002, San Francisco, California

Radisson Miyako Hotel

GENERAL INFORMATION:

The aim of the annual ALENEX workshops is to provide a forum for the presentation of original research in the implementation and experimental evaluation of algorithms and data structures. We invite submissions that present significant case studies in experimental analysis (such studies may tighten, extend, or otherwise improve current theoretical results) or in the implementation, testing, and evaluation of algorithms for realistic environments and scenarios, including specific applied areas (including databases, networks, operations research, computational biology and physics, computational geometry, and the world wide web) that present unique challenges in their underlying algorithmic problems. We also invite submissions that address methodological issues and standards in the context of empirical research on algorithms and data structures.

The scientific program will include invited talks, contributed research papers, and ample time for discussion and debate of topics in this rapidly evolving research area. A proceedings will be published, and a special issue of the ACM Journal of Experimental Algorithmics will feature invited contributions from the workshop.

This workshop is colocated with the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA02), and will take place in the two days preceding that conference. A paper that has been reviewed and accepted for presentation at SODA is not eligible for submission to ALENEX. We recognize, however that some research projects spawn multiple papers that elaborate on different aspects of the work and are willing to respond to inquiries about overlapping papers.

The workshop is supported by SIAM, the Society for Industrial and Applied Mathematics, and SIGACT, the ACM Special Interest Group on Algorithms and Computation Theory.

SUBMISSIONS:

Authors are invited to submit 10-page extended abstracts by 5:00 PM EDT, October 8, 2001 and must use the SIGACT electronic submissions server. Detailed instructions for submitting to the workshop can be found at the workshop's website.
http://cs.umd.edu/~mount/ALENEX02

Notification of acceptance or rejection will be sent by November 5, 2001. The deadline for receipt of papers in final version is December 10, 2001. Presenters must have submitted the final versions of their papers in order to be able to present them at the workshop.

PROGRAM COMMITTEE:

Nancy Amato, Texas A&M University
Marshall Bern, Xerox PARC
Michael Goodrich, Johns Hopkins University
Tom McCormick,
University of British Columbia
Michael Mitzenmacher, Harvard University
David Mount, (Co-chair),
University of Maryland
Giri Narasimhan, University of Memphis
Rajeev Raman, University of Leicester
Clifford Stein, (Co-chair), Columbia University

The Computational Optimization Research Center at Columbia University announces the 1st Columbia Optimization Day:

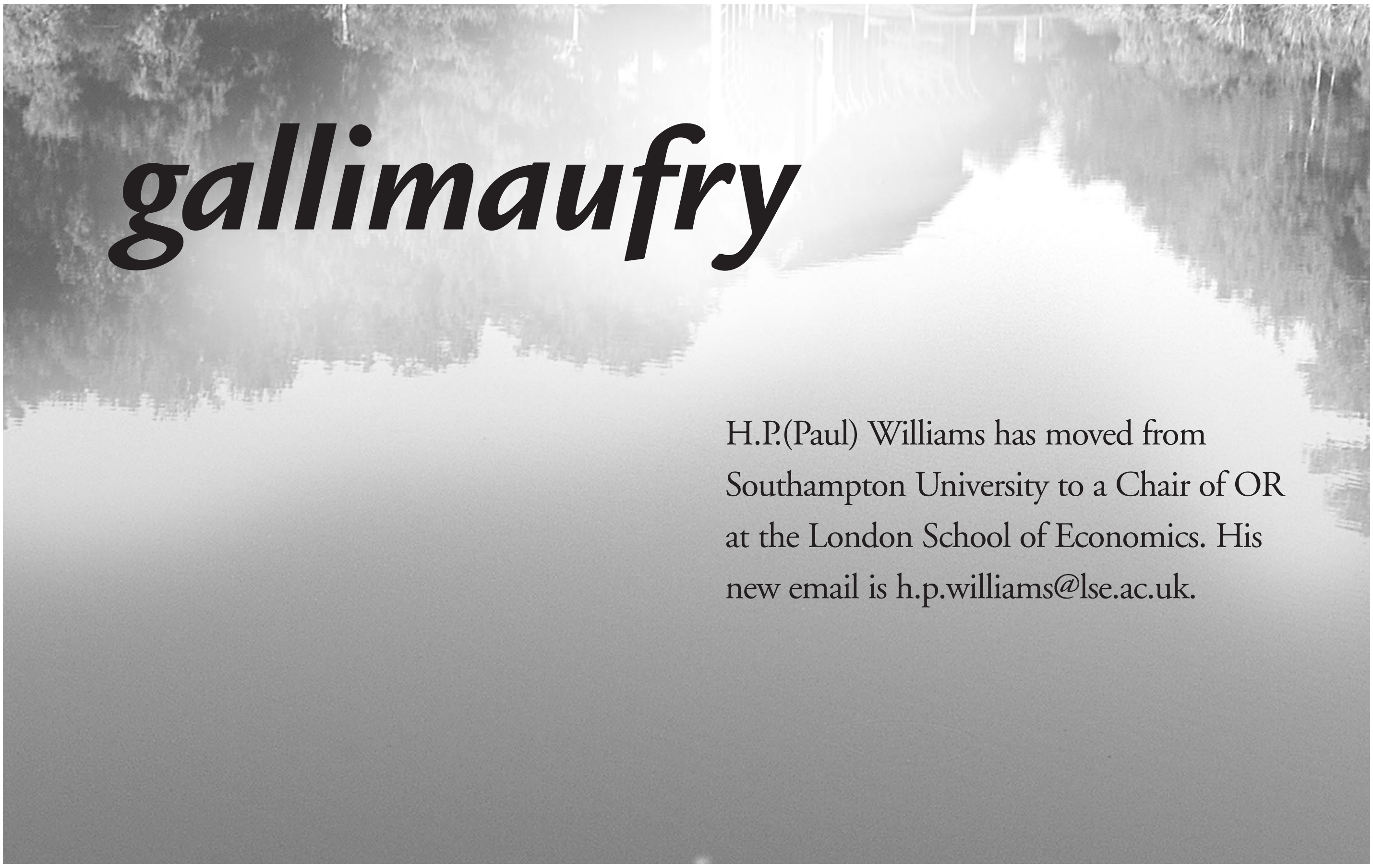
"Combinatorial Optimization and Integer Programming, The State of the Art"

Wednesday, November 28, 2001

Speakers:

Francisco Barahona, IBM Research
Robert Bixby, Rice University and ILOG Cplex
Sebastian Ceria, Axioma
William Cook, Princeton University
Vasek Chvatal, Rutgers University
David Johnson, AT&T Research
George Nemhauser, Georgia Tech
Bruce Shepherd, Lucent

This meeting will take place at Columbia University, New York
For further information, please visit
www.corc.ieor.columbia.edu/meetings/c1/c1.html



gallimaufry

H.P.(Paul) Williams has moved from Southampton University to a Chair of OR at the London School of Economics. His new email is h.p.williams@lse.ac.uk.

Application for Membership

I wish to enroll as a member of the Society.

My subscription is for my personal use and not for the benefit of any library or institution.

☐ I will pay my membership dues on receipt of your invoice.

☐ I wish to pay by credit card (Master/Euro or Visa).

CREDIT CARD NO. __________ EXPIRATION DATE __________

FAMILY NAME __________

MAILING ADDRESS __________

____

TELEPHONE NO. __________ TELEFAX NO. __________

EMAIL __________

SIGNATURE ▶ __________

Mail to:

Mathematical Programming Society
3600 University City Sciences Center
Philadelphia PA 19104-2688 USA

Cheques or money orders should be made payable to The Mathematical Programming Society, Inc. Dues for 2001, including subscription to the journal ***Mathematical Programming***, are US $75.
Student applications: Dues are one-half the above rate. Have a faculty member verify your student status and send application with dues to above address.

Faculty verifying status

____

Institution

____

Springer ad (insert film)

O P T I M A

MATHEMATICAL PROGRAMMING SOCIETY

UNIVERSITY OF FLORIDA

Center for Applied Optimization
371 Weil
PO Box 116595
Gainesville FL 32611-6595 USA

FIRST CLASS MAIL

EDITOR:
Jens Clausen
Informatics and Mathematical Modelling,
Technical University of Denmark
Building 305 room 218
DTU, 2800 Lyngby
Tlf: +45 45 25 33 87 (direct)
Fax: +45 45 88 26 73
e-mail: jc@imm.dtu.dk

FOUNDING EDITOR:
Donald W. Hearn
DESIGNER
Christina Loosli
PUBLISHED BY THE
MATHEMATICAL PROGRAMMING SOCIETY &
GATOREngineering® PUBLICATION SERVICES
University of Florida

Journal contents are subject to change by the publisher.