OPTIMA 98

Mathematical Optimization Society Newsletter

Contents of Issue 98 / June 2015

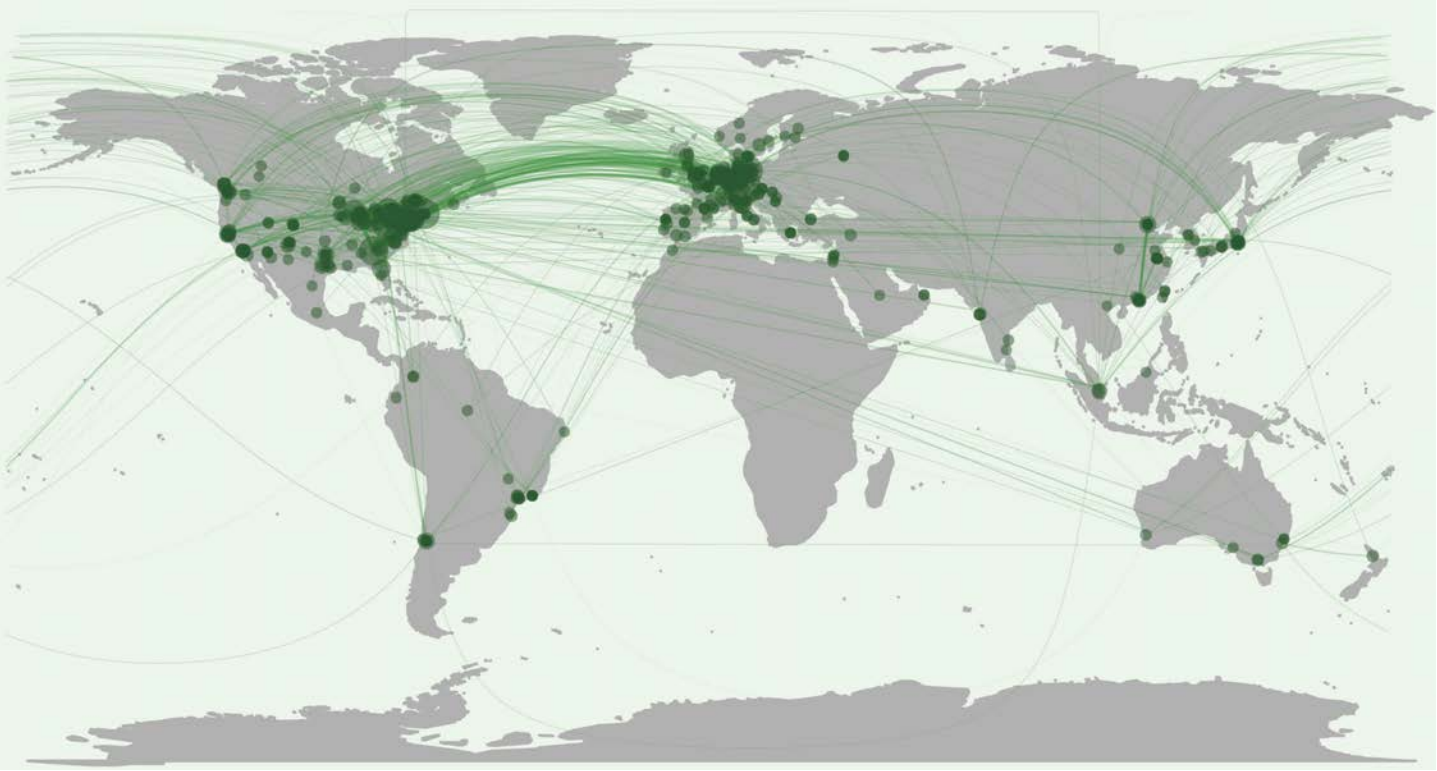
Note from the Editors

Dear MOS members,
A $15m computer that uses 'quantum physics' effects to boost its speed is to be installed at a NASA facility. It will be shared by Google, NASA, and other scientists, providing access to a machine said to be up to 3,600 times faster than conventional computers. (BBC Radio, May 16, 2013, see www.bbc.com/news/science-environment-22554494)

Such news certainly attracts the attention of many people working in Mathematical Optimization, especially when the benchmark problem refered to is an optimization problem and the comparisons are done against branch-and-bound based algorithms on a classical computer. In the main scientific article of this issue of *Optima*, Sanjeeb Dash and Jean Francois Puget (both from IBM) shed some light on the validity of conclusions like the one cited above by reporting about subsequent experiments on classical computers. In particular they analyze in detail the experimental setup of the CPLEX solver used as the classical algorithm in the comparison. Their results are quite interesting with respect to the question of whether to use linear instead of quadratic models for certain 0/1-optimization problems.

In case your expectations of quantum computing have been raised by news like above but then slightly lowered when reading Sanjeeb's and Jean Francois' article, we must warn you: the discussion column contributed by Matthias Troyer (ETH Zürich, Institute for Theoretical Physics) might be further disillusioning, but certainly illuminating.

This issue of our newsletter in front of you appears at the time of the *22nd International Symposium on Mathematical Programming (ISMP)* held July 12–17, 2015, in Pittsburgh – you might in fact have grabbed a copy at the conference desk. We take this opportunity to com-



ISMP 2015: Contributions and connections between groups in the optimization community, visualized by Sam Burer (The complete code for computing the figure can be downloaded from http://sburer.github.io/2015/06/17/ISMP-2015.html) memorate one of the most crucial figures in the development of Integer Programming: Manfred Padberg, who received his doctoral degree from Carnegie Mellon University in Pittsburgh in 1971, and who passed away in May 2014. You will find inside this issue a contribution by his PhD-advisor Egon Balas and a gallery of some pictures from the scientific life of one of the most enthusiastic, influential, and memorable characters in the history of Mathematical Optimization.

ISMP 2015 in Pittsburgh is going to feature around 1,400 presentations. The figure on the first page (produced by our co-editor Sam Burer) visualizes how the contributing groups are distributed worldwilde and tries to derive a picture of their interactions from the program information on all sessions, all session chairs, and all presenters. A *group* of participants is defined to be those from the same institution (e.g., university or company) as determined by the domain name in a participants's email address (generic domains such as gmail.com have been ignored). For each group, the number of sessions in which it participates, and for every pair of groups, the number of sessions in which that pair participates together were calculated. Finally, the groups have been plotted geographically as points with the point size indicating the group's participation frequency, and great-circle connections have been drawn to indicate the pairs that participated together. A connection's thickness and opacity indicates the pair's participation frequency.

Finally, this newsletter also provides an outlook into the future with information about the outcome of the recent MOS elections (congratulations to everybody elected, in particular to the next chair of our society: Karen Aardal) and announcements of some of the most important Mathematical Optimization conferences coming up in 2016: IPCO in Liège, Belgium, ICCOPT in Tokyo, Japan, and ICSP in Búzios, Brasil.

We hope you enjoy reading and wish you a pleasant and fruitful ISMP!

Sam Burer, Co-Editor
Volker Kaibel, Editor
Jeff Linderoth, Co-Editor

Sanjeeb Dash and Jean-François Puget

On quadratic unconstrained binary optimization problems defined on Chimera graphs

1 Introduction

An important question in physics and computer science is whether it is possible in practice to build a quantum computer that can solve interesting computational problems faster than a classical computer. The D-Wave Two system, stated to be the "first commercial quantum computer" by its manufacturer [19], is a special-purpose machine designed to solve the Ising Spin Model (IM) problem. Recently McGeoch and Wang [27] compared the time to solve certain combinatorial optimization problems on a D-Wave Two system (after transforming them to equivalent IM problem instances) to the running time of some software solvers on an Intel Xeon workstation. For the quadratic unconstrained binary optimization (QUBO) problem, the best software solver in their experiments (IBM ILOG CPLEX Optimizer 12.3 [23], CPLEX 12.3 for short) was about 3600 times slower than D-Wave Two in obtaining comparable results – in terms of objective function value. This conclusion was reported in some news articles, e.g., [22, 24], as implying that D-Wave Two was 3600 times faster than "conventional" or "traditional" computers for this problem. However, this inference is not well-founded as CPLEX 12.3 produces optimal solutions via a branch-and-bound algorithm whereas D-Wave Two is not guaranteed to produce optimal solutions.

A natural question is if heuristics running on classical computers can outperform D-Wave Two on the Ising Spin Model problem. This was shown to be the case in [10]; a simulated annealing based heuristic running on a regular workstation outperforms D-Wave Two (see Section 2.1). A second question is whether the running times to obtain optimal solutions for the QUBO instances in McGeoch and Wang [27] can be improved significantly, especially with LP-based branch-and-bound techniques. In this paper, we answer this question affirmatively and argue that there is a weakness in the CPLEX versus D-Wave comparisons of McGeoch and Wang. When we use a well-known alternative mathematical programming formulation of the QUBO problem to the one used by them, CPLEX 12.3 can solve their problem instances in a maximum of 94 seconds – and in 26 seconds on the average – on one core of a Windows 7 workstation (and within 56 seconds on four cores of the same machine) instead of the 1800 seconds reported in their paper.

Furthermore, McGeoch and Wang report that CPLEX 12.3 was able to solve only 34 out of 120 weighted Max-2SAT instances within half an hour when these instances were given to CPLEX as QUBO instances. We observe that CPLEX can solve 117 out of these 120 instances within half an hour when the alternative formulation mentioned above is used.

2 Background

Quantum adiabatic computation [20] is a model of quantum computation equivalent to [1] the more well-known quantum circuit model of computation ([30] explains the latter model). The D-Wave Two system, in the words of the manufacturer, "consists of 512 tiny superconducting circuits, known as qubits, chilled close to absolute zero to get quantum effects"; see [14] for a description. D-Wave Two uses these quantum effects to implement "quantum annealing", a heuristic search process closely related to quantum adiabatic computation. (There is some debate on the quantum nature of D-Wave machines, see [10, 32, 35, 36].) The qubits and couplers (pairs of interacting qubits) are arranged as a subgraph of the Chimera graph C_8. A *Chimera graph* C_k [27] has $8k^2$ vertices arranged in a grid-like pattern: each node in a $k \times k$ grid graph is replaced by a complete bipartite graph $K_{4,4}$, and the nodes in the "right partition" are connected to the respective nodes in the right partitions of the $K_{4,4}$s on the left and right (if they exist), and the nodes in the "left partition" are connected to the respective nodes in the left partitions of the $K_{4,4}$s above and below. See [10, Supplementary Information, Figure 1] for a depiction of the qubits and couplers in a D-Wave One system which form a subgraph of a Chimera graph C_4; some of the nodes in the graph are active qubits and some are inactive, and there are no couplers incident with inactive nodes.

The D-Wave Two system uses quantum annealing to approximately solve Ising Spin Model (IM) problem instances with at most 512 variables (or spins) defined on a subgraph of the Chimera graph C_8. Given an $n \times n$ matrix J and n-vector h, the IM problem is:

$$\text{Min} \sum_{i \neq j} J_{ij} s_i s_j + \sum_i h_i s_i \quad \text{subject to} \quad s \in \{-1, +1\}^n. \qquad (1)$$

(The coefficient of $s_i s_j$ is $J_{ij} + J_{ji}$, and thus J can be assumed to be symmetric or upper triangular.) The IM problem with (nonzero) fields refers to the case when $h \neq 0$; otherwise the problem is one with zero fields. An IM instance is associated with an unweighted *connectivity graph* $G = (V, E)$ where $V = \{1, \dots, n\}$ and $E = \{(i, j) : i < j \text{ and } J_{ij} + J_{ji} \neq 0\}$; we refer to $J_{ij} + J_{ji}$ values as node weights and h_i values as edge weights. An instance is defined on a graph G' if its connectivity graph is a subgraph of G'. To the best of our knowledge, the quantum annealing algorithm in D-Wave Two cannot be changed by the user, and so we refer to D-Wave Two itself as a heuristic solver for IM instances. For an overview of the computational model used in D-Wave, see [27, 28].

Any other combinatorial optimization problem must be transformed into an IM instance defined on C_8 in order to be "natively" solved on D-Wave Two; see [15], for example, on solving a map-coloring problem on D-Wave. If such a transformation is not possible (see [9] for discussions on this issue), D-Wave provides the "QSage" hybrid solver (a previous version was called "Blackbox") which combines computations on a conventional computer along with queries to the D-Wave quantum hardware.

The QUBO problem is the problem of minimizing a quadratic function of n $\{0,1\}$ variables:

$$\text{Min} \sum_{i,j} Q_{ij}x_ix_j \quad \text{subject to} \quad x \in \{0,1\}^n, \tag{2}$$

where Q is an $n \times n$ matrix. See [11] for applications. The connectivity graph of an instance is defined as in the case of IM instances. We refer to $Q_{ij} + Q_{ji}$ values as edge-weights and Q_{ii} values as node weights. The decision version of the QUBO problem is NP-complete even when the connectivity graph is a planar cubic graph [4], or a Chimera graph. The latter result follows from [13], where it is shown that the complete graph on $4n$ nodes is a minor of C_n.

The transformation $s_i = 2x_i - 1$ maps the IM problem (1) to the QUBO problem:

$$\text{Min} \sum_{i\neq j} 4J_{ij}x_ix_j + \sum_i 2(h_i - \sum_{j\neq i} J_{ij})x_i + c \quad \text{subject to} \quad x \in \{0,1\}^n, \tag{3}$$

where c is the constant $\sum_{i\neq j} J_{ij} - \sum_i h_i$. Under this transformation, the connectivity graph is unchanged.

In this paper, a random IM instance (defined on a graph G) is one where all node and edge weights are chosen independently at random from a uniform distribution, and a random QUBO instance is defined similarly. An important point is that the transformation $s_i = 2x_i - 1$ does not map a random IM instance to a random QUBO instance (with the same uniform distribution). The diagonal entries of Q – which can be assumed to be equal to the coefficients of x_i as $x_i^2 = x_i$ when $x_i \in \{0,1\}$ – are correlated with the off-diagonal entries. Depending on whether $h = 0$ or $h \neq 0$ in the IM instance, we call the corresponding QUBO instances *strongly correlated* or *weakly correlated* random instances, respectively.

McGeoch and Wang [27] compared three software packages – CPLEX 12.3, the Max-SAT solver akmaxsat[25], and the Tabu Search module of METSlib[29] – against a D-Wave Two system on the QUBO problem, and against the hybrid Blackbox solver of D-Wave Two on the Max-2SAT and Quadratic Assignment (QAP) problems. The three solvers above were run on one core of a 4-core Intel Xeon (E5-2609) 2.4 GHz processor with a 64-bit Ubuntu 12.04 LTS operating system. The D-wave Two system had a Vesuvius 5 (V5) hardware chip with 439 active qubits. The time taken by D-Wave for an IM instance is given by $t_1 + kt_2$ where t_1 – the *overhead time* – is the time to initialize D-Wave, t_2 – the *sampling time* – is the time to perform one (quantum) "anneal" and get one solution, and k is the number of "anneals"; in their experiments t_1=201ms, t_2=0.29ms and k was set to 1000. In other words, D-Wave would run for a fixed time of 491ms.

For the QUBO/IM problem, McGeoch and Wang experimented with 600 random IM instances defined on the D-Wave hardware graph with between 32 and 439 variables. Out of the 1000 solutions returned by D-Wave within 491ms, the solution with the best objective value was recorded. After transforming these IM instances into QUBO instances, the authors ran the CPLEX 12.3 mixed-integer quadratic programming (MIQP) solver (and akmaxsat and METSLib) for up to 1800 seconds. For all 600 instances, the best solution returned by D-Wave in 491ms was the best among all solvers. Furthemore, for 585 (97%) instances, the best solution returned by D-Wave was optimal, as certified by running CPLEX for up to 1800 seconds. They conclude from this that CPLEX "returns comparable results running about 3600 times slower than the [D-Wave] hardware".

For the Max-2SAT problem, McGeoch and Wang used 120 weighted Max-2SAT instances (40 instances with $n = 100$ variables, and the same number with 120 and 140 variables, and with roughly 1200-1600 clauses). These instances are constructed by modifying the clause weights of the random, weighted Max-2SAT instances from Max-SAT 2012 [2] from random integers in the interval [1,10] to random values from the numbers $\{-7.5, -6.5, \dots, -.5, .5, \dots, 7.5\}$. It is easy to transform a Max-2SAT instance over variables $x_1, \dots, x_n$ into an equivalent QUBO instance over the same variables. For example, for a clause $x_1 \vee x_2$ in a Max-2SAT instance, the corresponding term in the QUBO objective function (assuming a maximization objective) is $1 - (1 - x_1)(1 - x_2)$. McGeoch and Wang report that the optimal solutions of only 34 out of 120 instances can be certified in half an hour with CPLEX.

We do not perform any comparisons with the QAP results in [27], and therefore do not discuss them in this paper.

2.1 Heuristics for IM instances

Optimal solutions (without proofs of optimality) of instances very similar to the IM instances of McGeoch and Wang seem to be easily obtainable with heuristics. A simulated annealing code by Boixo et al. [10] (also see [32]) finds optimum solutions with high probability to random IM instances with fields defined on 512 node C_8 Chimera graphs in roughly 0.02 seconds on an 8-core Intel Xeon E5-2670 CPU. For instances without fields, it takes roughly 0.12 seconds. A heuristic by Selby [34] designed for Chimera graphs returns optimal solutions to similar 439 node instances with fields with high probability in about 0.01 seconds on one core of a 3.2 GHz CPU.

Boros, Hammer and Tavares [11] show that local-search heuristics for QUBO can be competitive with other heuristics. We implemented a very simple, randomized, local-search heuristic that returns optimal solutions to at least 99 out of the 100 McGeoch-Wang 439 node instances within 0.25 seconds in 99 out of 100 invocations with different random seeds (we use all 4 cores of a 2.2 GHz Intel CPU specified in the next section). The heuristic is: start with a random ± 1 assignment to variables; then repeatedly flip the sign of the variable that most reduces the current objective function if one exists, else flip the signs of a random subset of variables of fixed size ($n/30$ in our tests). In other words, find local minima via a 1-opt algorithm and then perturb (randomly) to escape from local minima (while recording the best solution found at any step).

The effectiveness of this heuristic may be because of the many optimal solutions to the McGeoch-Wang instances. We obtain a lower bound of between 2^{21} and 2^{30} on the number of *distinct* optimal solutions to the 439 node instances by finding maximal independent sets of variables such that flipping their values in an optimal solution individually does not change the objective function value (such variables correspond to *free qubits*, see the discussion on degeneracies in [10]). We did not evaluate our heuristic on the zero field problems as it was difficult to solve them to optimality.

The D-Wave V6 chip is estimated [27] to be three to five times faster than the V5 chip, and D-Wave V7 is likely to be even faster, so the above comparisons with simple heuristics may not hold for newer D-Wave machines.

On the theoretical side, Saket [33] recently developed a PTAS for the IM problem on Chimera graphs.

3 Computational results

McGeoch and Wang solve QUBO instances using the CPLEX 12.3 MIQP solver; they say "Since QUBO has a quadratic objective function, the quadratic programming (QP) module was used throughout". We refer to the natural integer quadratic programming formulation of QUBO in (2) as QUBO-iqp; however, we do not know if McGeoch and Wang used this quadratic formulation. There is a well-known alternative mixed-integer linear programming (MILP) formulation of QUBO, which we call QUBO-milp:

$$\begin{aligned} \text{Min } \textstyle\sum_{i<j} Q_{ij}z_{ij} + \sum_{i=1}^{n} Q_{ii}x_i & \quad \text{subject to} & \\ z_{ij} \le x_i & \quad \forall i<j, & (4)\\ z_{ij} \le x_j & \quad \forall i<j, & (5)\\ x_i + x_j - z_{ij} \le 1 & \quad \forall i<j, & (6)\\ z_{ij} \ge 0 & \quad \forall i<j, & (7)\\ x \in \{0,1\}^n & & (8) \end{aligned}$$

(we assume, without loss of generality, that Q is upper triangular). For any i, j, the constraints (4)-(7) (called Fortet inequalities [21] or McCormick inequalities [26]) force z_{ij} to equal x_ix_j when $x_i, x_j \in \{0,1\}$, and define the convex hull of $\{(x_i, x_j, x_ix_j) : x_i, x_j \in \{0,1\}\}$. If $Q_{ij} > 0$, then (4)-(5) can be dropped and if $Q_{ij} < 0$ then the other two constraints can be dropped; we use the subset of constraints used in [8]. In our experiments, we solve QUBO-milp with the CPLEX 12.3 MILP solver.

In our first experiment, we solve random QUBO instances defined on Chimera graphs (all weights are chosen from a uniform distribution over $\{-1,+1\}$) similar to the ones described by McGeoch and Wang; they state that for their QUBO instances, "Weights are drawn uniformly from $\{-1,+1\}$". We were informed by Geordie Rose of D-Wave Systems Inc. and by McGeoch that this statement is incorrect; their results are instead based on random IM instances which map to weakly correlated QUBO instances, which they kindly provided to us. Therefore, in our second experiment, we solve these weakly correlated QUBO instances. In our third experiment, we work with the Max-2SAT instances of McGeoch and Wang. All our experiments were performed on a Windows 7 64-bit workstation with a 4-core 2.2 GHz Intel Core i7-2720QM CPU and 8 GB memory. Unless stated otherwise, only one CPU core was used.

We observe that random QUBO instances defined on Chimera graphs are easy to solve to optimality (up to the default optimality tolerance of CPLEX). On the average, the CPLEX MILP solver takes less than 0.2 seconds to solve instances on C_8 graphs with 512 nodes, and less than 23.1 seconds to solve instances based on C_{50} graphs with $20,000$ nodes. Furthermore, QUBO-iqp is harder to solve for these instances. For C_4 graphs, solving QUBO-iqp with CPLEX's MIQP solver is 5 times slower on the average than solving QUBO-milp, and over 8,000 times slower in the worst case (see Section 4).

For the 600 randomly generated IM instances with fields used by McGeoch and Wang, QUBO-milp can be solved to optimality with the CPLEX MILP solver in at most 93.80 seconds (and 25.58 seconds on the average) using one core of our CPU, though using some non-default CPLEX settings (which we feel is fair as McGeoch and Wang also perform a "pilot study aimed at finding good parameter settings"). This is in contrast to the running time of up to 1800 seconds reported by McGeoch and Wang to solve these instances with CPLEX's MIQP solver. If we use all 4 cores of our CPU, QUBO-milp can be solved in at most 56 seconds (and in 20 seconds on the average).

If the zero field version of the same instances are used (i.e., the terms h_i in (1) are set to zero), QUBO-milp is very hard to solve, which takes over 20,000 seconds for some 439 node instances, even with 4 CPU cores. In other words, the instances used by McGeoch and Wang do not seem difficult for classical computers, but other instances which can be defined on the hardware graph of the D-Wave Two computer seem hard, at least for CPLEX (but not necessarily so for heuristics). Furthermore, random QUBO instances seem much easier than weakly correlated instances, which in turn seem much easier than strongly correlated instances.

Finally, 117 out of 120 Max-2SAT instances are solved to optimality using QUBO-milp within 1800 seconds, and the average time to solve all instances is 280.53 seconds, whereas only 34 out of 120 instances are solved to optimality using CPLEX's MIQP solver in the McGeoch-Wang experiments.

We note that the optimal solutions of similar Chimera graph based instances have been obtained in [10] and [34] using other exact methods.

3.1 Random QUBO instances

We use random QUBO instances defined on Chimera graphs C_k with $k = 8, 20, 35$ and 50, where each node or edge weight of the Chimera graph is chosen uniformly at random from $\{-1,1\}$. Additional experiments with integer weights in the range $[-100,100]$ can be found in an earlier version of this paper (arXiv:1306.1202v2).

In Table 1, we give the time (in seconds) to solve QUBO-milp with CPLEX running with default settings. We give the graph, and its number of nodes and edges in columns one-three. In the remaining five columns, we give, respectively, the arithmetic mean, the geometric mean, and the minimum, maximum and standard deviation of the running times across 50 randomly generated instances.

Table 1. *Run times (in seconds) with CPLEX 12.3 for QUBO-milp on random QUBO instances*

Graph	nodes	edges	Mean	G. Mean	Min	Max	Std. Dev.
C_8	512	1472	0.12	0.11	0.08	0.19	0.02
C_{20}	3200	9440	1.31	1.29	0.97	2.42	0.27
C_{35}	9800	29120	7.32	7.26	6.10	11.09	0.99
C_{50}	20000	59600	23.12	22.92	10.44	29.50	2.71

The maximum time to solve any instance is 29.50 seconds. Out of the 200 instances in this table, 198 are solved without branching and the total number of branch-and-bound nodes for the remaining two is 96. Thus the LP relaxation of QUBO-milp augmented with cutting planes (or cuts for short) is enough to solve almost all instances; we dicuss cuts for QUBO-milp in the next section. CPLEX only generates two types of cuts in these experiments, namely zero-half cuts [12] and Gomory cuts, and these seem essential. When cuts are turned off, the C_8 instances are all solved within 1.81 seconds with pure branch-and-bound, but larger instances are hard to solve in this way; for C_{20} instances the worst-case solution time without cuts is over 1000 seconds.

3.2 Random Ising Model Instances

We solve QUBO-milp for the IM instances with fields used by McGeoch and Wang. There are 6 groups of instances, with 100 in each group. All instances in each group are based on the same graph but

Table 2. *Run times (in seconds) with CPLEX 12.3 for QUBO-milp on McGeoch-Wang instances*

cores	nodes	edges	Mean	G. Mean	Min	Max	Std. Dev.
1	32	80	0.05	0.04	0.01	0.39	0.05
	119	305	1.12	1.03	0.23	2.06	0.41
	184	471	2.38	2.18	0.53	5.77	0.96
	261	672	5.72	5.10	1.64	18.00	2.99
	349	899	11.81	10.80	3.04	30.36	5.07
	439	1119	25.58	22.18	7.71	93.80	15.29
4	439	1119	20.12	18.01	8.11	55.55	10.49

have different edge and node weights. To create an instance, edge and node weights are drawn uniformly at random from ±1. In Table 2, the first column gives the number of CPU cores used. The remaining columns have the same interpretation as the last seven columns in Table 1. We use non-default CPLEX settings as these instances are harder than those in Table 1. As Gomory cuts and zero-half cuts are the only cuts employed by CPLEX in solving the latter, we turn on "aggressive" generation of these two families of cuts for the current instances. It takes 10 times longer to solve these instances with default settings.

The hardest QUBO-milp instances (the 439 node ones) can be solved within 25.58 seconds on the average and within a maximum of 93.80 seconds. Furthermore, if 4 cores are used, then the maximum time is 55.55 seconds and the average time is 20.12 seconds. Additional computations with CPLEX 12.5.1 and 32 cores can be found at [31]; version 12.5.1 in fact automatically chooses between QUBO-iqp and QUBO-milp.

Interestingly, even if we provide the optimal solution at the outset, it still takes 22.8 seconds on the average to solve QUBO-milp for the 439 node McGeoch-Wang instances, and 88.6 seconds at most. In other words, most of the work done by CPLEX is for verifying optimality (which D-Wave does not do).

On the other hand, if we simply remove the random field (h_i terms) from the IM instances of McGeoch and Wang before transforming to strongly correlated QUBO instances, QUBO-milp seems very hard to solve. The worst case solution time with four cores is over 20,000 seconds compared to the maximum of 93.80 seconds on one core for the instances with fields. We believe that specialized LP based codes as in [7] are likely to be faster than CPLEX.

3.3 Max-2SAT instances

The Max-2SAT instances used by McGeoch and Wang were provided to us by D-Wave as QUBO instances, and we simply solve QUBO-milp for these instances. McGeoch and Wang report that the optimal solutions of only 34 out of 120 instances can be obtained in half an hour with CPLEX's MIQP solver. We observe that the optimal solutions of 117 instances can be obtained via QUBO-milp in half an hour. In Table 3, we give the run-times for QUBO-milp for the problems grouped by number of variables. The first column of the table gives the number of variables n, the last column gives the number of instances for which QUBO-milp was solved in over half an hour, and the remaining columns have the same meaning as in the previous tables. We note that increasing n from 100 to 140 does not change running times as much as increasing the number of clauses from 1200 to 1600. In fact all three instances which take more than half an hour to solve (via QUBO-milp) have about 1500 clauses.

Table 3. *Run times in seconds for random Max-2SAT instances with different number of variables*

n	Mean	G. Mean	Min	Max	num > 1800
100	211.31	113.21	32.18	1595.81	0
120	323.21	195.09	39.72	2046.14	2
140	306.97	199.52	57.08	2257.30	1

4 QUBO formulations

The two approaches to solving QUBO instances we discussed are based on either modeling QUBO instances as integer quadratic programs or as mixed-integer linear programs (MILP). For the first approach, letting $\bar{Q} = (Q + Q^T)/2$, QUBO-iqp in (2) can be rewritten as

$$\text{Min } \; x^T \bar{Q} x \; \text{ subject to } \; x \in \{0,1\}^n. \tag{9}$$

An MIQP solver, such as the one in CPLEX, typically uses a branch-and-bound algorithm, where lower bounds on QUBO-iqp are obtained by solving a QP relaxation of the form

$$\text{Min } \; x^T \bar{Q} x \; \text{ subject to } \; x \in [0,1]^n. \tag{10}$$

If $\bar{Q}$ is not positive semidefinite, then the QP problem above is non-convex and NP-hard. For QUBO instances, a simple transformation makes $\bar{Q}$ positive semidefinite without changing the optimal solution. Let D be a $n \times n$ diagonal matrix. Then

$$\text{Min } \; x^T(\bar{Q} + D)x - \sum_{i=1}^{n} D_{ii} x_i \; \text{ subject to } \; x \in \{0,1\}^n, \tag{11}$$

is equivalent to QUBO-iqp: when $x_i \in \{0,1\}$, $x_i^2 = x_i$ and for every $x \in \{0,1\}^n$, the objective function in (11) has the same value as the objective function in (9). By choosing D such that $\bar{Q} + D$ is positive semidefinite, e.g., if $D = -\lambda_{min}(\bar{Q})I$ where $\lambda_{min}(\bar{Q})$ is the minimum eigenvalue of $\bar{Q}$ and I is the identity matrix, and letting $x \in [0,1]^n$, one gets a convex QP relaxation of (11) which can be solved efficiently and used in a branch-and-bound algorithm. The choice of D influences the quality of the lower bound on the optimal solution value of (11) from this relaxation, see [8].

As for the second approach, QUBO-milp can often be solved quickly with linear programming (LP) based branch-and-cut algorithms when the connectivity graph is sparse. The QUBO (or IM) problem maps, via a one-to-one linear transformation, to the Max Cut problem [16] on a graph with one extra node connected to all existing nodes. Some IM instances on 3D grid graphs were shown to be easy to solve [5] using LP relaxations combined with *cycle* inequalities [4] for the Max Cut problem (mapped back to the IM variable space). The same techniques combined with branch-and-bound were used in [6] to solve randomly generated QUBO instances on sparse graphs with ($n =$) 100 nodes (and average node degree $\leq .0625n$), and – with additional cutting planes – in [17] to solve IM instances on 100×100 2D grid graphs. In [6], 160 out of 162 instances were solved by strengthening the LP relaxation with cycle inequalities and without branching; almost all instances in [17] were solved in this manner. It was shown in [8] that QUBO-milp instances based on sparse, randomly generated graphs G with $n \leq 80$ (average degree of a node is $.2n$ or less) can be solved very quickly with a general MILP solver, specifically CPLEX 8.1. They use edge weights in the range $[-50, 50]$ and node weights in the range $[-100, 100]$. The average node degree of a Chimera graph is between 5 and 6, and much less than the average degrees considered in [8]. On the other hand, it is observed in [18] and [7] that random IM instances with zero fields (often called "+-J Ising spin glasses") and their associated QUBO istances are quite difficult to solve with LP based methods (Ojas Parekh and Matthias Troyer told us about the difficulty of these instances).

For the random QUBO instances in Section 3.1, we compare the time to solve QUBO-iqp and QUBO-milp with the respective solvers of CPLEX (with default settings). Even for relatively small C_4-based instances, the difference in time is significant, in the worst case. The geometric mean of solution times for QUBO-milp over 50 random C_4-based instances is 0.02 seconds and 0.10 seconds for QUBO-iqp, whereas the maximum solution time for QUBO-milp is 0.16 seconds, and the maximum solution time for QUBO-iqp is 1355.85 seconds. Therefore, solving QUBO-iqp can take much longer than solving QUBO-milp on these instances (the gap is even larger for the larger instances).

We believe the running-time disparity is mainly due to formulation differences rather than differences in solver quality. Firstly, even though the QP relaxation of QUBO-iqp is stronger than the LP relaxation of QUBO-milp in general, when $\bar{Q}$ is modified as in (11) to make it positive semidefinite ("repairing indefiniteness" in CPLEX language), the associated convex QP relaxation yields *significantly worse* lower bounds for many of the above instances. Secondly, QUBO-milp is an extended formulation with extra variables (representing $x_i x_j$). One can derive new linear constraints (e.g., cycle inequalities) to get a better approximation of $\text{conv}(\{xx^T : x \in \{0,1\}^n\})$ than $\text{conv}(\{xx^T : x \in [0,1]^n\})$. This is not possible in QUBO-iqp. CPLEX does not generate cycle inequalities specifically when solving QUBO-milp; however it generates zero-half cuts and Gomory cuts, and cycle inequalities form a subclass of Gomory cuts.

We also observe high computation times for QUBO-iqp for the McGeoch-Wang IM instances (even though the initial "repaired" QP bound is stronger than the LP bound from QUBO-milp). For the 439 node instances, QUBO-milp can be solved in at most 93.80 seconds, whereas even after 600 seconds and over 900,000 nodes, CPLEX is unable to solve QUBO-iqp and there is a large optimality gap (22% on the average). However, in this case, the best solution found is optimal for about a third of the instances, and within 0.3% of the optimal one for all instances. That said, QUBO-iqp may be much better for dense graphs [8].

Acknowledgements

We would like to thank Geordie Rose of D-Wave Systems, and Catherine McGeoch and Carrie Wang for providing us the precise instances used in the QUBO tests and Max-2SAT tests in [27]. We would like to thank Francisco Barahona for helpful discussions on the QUBO problem.

Sanjeeb Dash, Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA. sanjeebd@us.ibm.com

Jean-François Puget, IBM Software Group, 1681 Route des Dolines, Valbonne 06560, France. j-f.puget@fr.ibm.com

Bibliography

[1] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, O. Regev, Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation, *SIAM Journal on Computing* **37**(1), 166–194 (2007).

[2] J. Argelich, C. M. Li, F. Manyà, J. Planes (organizers), Max-SAT 2012: Seventh Max-SAT Evaluation, www.maxsat.udl.cat.

[3] F. Barahona, On the computational complexity of Ising spin glass models, *Journal of Physics A: Mathematical and General* **15**(10), 3241–3253 (1982).

[4] F. Barahona, A.R. Mahjoub, On the cut polytope, *Math. Programming* **36**, 157—173 (1986).

[5] F. Barahona, E. Maccioni, On the exact ground states of three-dimensional Ising spin glasses, *Journal of Physics A: Mathematical and General* **15**, 611–615 (1982).

[6] F. Barahona, M. Jünger, G. Reinelt, Experiments in quadratic 0–1 programming, *Math. Programming* **44**, 127–137 (1989).

[7] F. Barahona, L. Ladanyi, Branch and cut based on the volume algorithm: Steiner trees in graphs and max-cut, *RAIRO - Operations Research* **40**(1), 53–73 (2006).

[8] A. Billionnet and S.Elloumi, Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem, *Math. Programming* **109**, 55–68 (2007).

[9] Z. Bian, F. Chudak , R. Israel, B. Lackey, W. G. Macready, A. Roy, Discrete optimization using Quantum Annealing on sparse Ising models, *Front. Phy.* **2**(56), (2014), doi 10.3389/fphy.2014.00056.

[10] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, M. Troyer, Evidence for quantum annealing with more than one hundred qubits, *Nature Physics* **10**, 218–224 (2014).

[11] E. Boros, P. Hammer and G. Tavares, Local search heuristics for quadratic unconstrained binary optimization (QUBO), *Journal of Heuristics* **13**, 99–132 (2007).

[12] A. Caprara and M. Fischetti, $\{0, 1/2\}$-Chvátal-Gomory cuts, *Math. Programming* **74**, 221–235 (1996).

[13] V. Choi, Minor embedding in adiabatic quantum computation: I. The parameter setting problem, *Quantum Inf. Process.* **7**, 193–2009 (2008).

[14] E. D. Dahl, Quantum computing 101, *Machine Design*, Jan 2015.

[15] E. D. Dahl, Programming a quantum computer, *Machine Design*, Jan 2015.

[16] C. De Simone, The cut polytope and the Boolean quadric polytope, *Discrete Math.* **79**(1), 71–75 (1990).

[17] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm *J. Stat. Phy.* **80**(1-2), 487–496 (1995).

[18] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, Exact ground states of two-dimensional +-J Ising spin glasses, *J. Stat. Phy.* **84**(5-6), 1363–1371 (1996).

[19] D-Wave Systems Inc., 4401 Still Creek Drive, Burnaby, British Columbia, Canada V5C 6G9.

[20] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem, *Science* **292**(5516), 472–475 (2001).

[21] R. Fortet, Applications de l'algebre de Boole en recherche opérationelle. *Revue Française de Recherche Opérationelle* **4**(14), 17–26 (1960).

[22] Q. Hardy, Google buys a quantum computer, *New York Times*, Bits (The Business of Technology Blog), 16 May, 2013.

[23] IBM ILOG CPLEX Optimizer, www-01.ibm.com/software/commerce/optimization/cplex-optimizer.

[24] N. Jones, Google and NASA snap up quantum computer, *Nature* News, 16 May, 2013, doi:10.1038/nature.2013.12999.

[25] A. Kuegel, Improved Exact Solver for the Weighted MAX-SAT Problem, *POS-10*: Pragmatics of SAT (D. Le Berre, Ed.), EPiC Series volume **8**, 15–27, EasyChair, 2012.

[26] G. P. McCormick, Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems, *Math. Programming* **10**, 147–175 (1976).

[27] C. C. McGeoch and C. Wang, Experimental Evaluation of an adiabatic quantum system for combinatorial optimization, *Proceedings of the 2013 ACM Conference on Computing Frontiers* 2013, Ischia, Italy.

[28] C. C. McGeoch, Adiabatic quantum computation and quantum annealing: Theory and Practice, *Synthesis Lectures on Quantum Computing* **8**, Morgan and Claypool Publishers, 2014.

[29] `METSlib`, COIN-OR METSlib, An open source Tabu Search metaheuristic framework in modern C++, https://projects.coin-or.org/metslib.

[30] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, UK, 2000.

[31] J. F. Puget, D-Wave vs CPLEX Comparison. Part 2: QUBO, 2013, https://www.ibm.com/developerworks/community/blogs/jfp/?lang=en.

[32] T. F. Ronnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, M. Troyer, Defining and detecting quantum speedup, *Science* **345**(6195), 420–424 (2014).

[33] R. Saket, A PTAS for the Classical Ising Spin Glass Problem on the Chimera Graph Structure, arXiv:1306.6943 (2013).

[34] A. Selby, D-Wave: comment on comparison with classical computers, 2013, http://www.archduke.org/stuff/d-wave-comment-on-comparison-with-classical-computers.

[35] J. A. Smolin and G. Smith, Classical signature of quantum annealing, *Front. Phy.*, **2**(52), 2014, doi: 10.3389/fphy.2014.00052.

[36] S. W. Shin, G. Smith, J. A. Smolin, and U. Vazirani, How "Quantum" is the D-Wave Machine? arXiv:1401.7087 [quant-ph] (2014).

Matthias Troyer

Quantum annealing and the D-Wave devices

Faced by the computational challenge of simulating interacting quantum systems, which grows exponentially with the number of particles, Richard Feynman in 1982 proposed to use quantum mechanics itself in a so-called quantum computer to simulate problems in quantum physics [1]. Quantum computation remained a theoretical curiosity until Peter Shor showed in 1994 that a quantum computer could factor integers in polynomial time [2]. This raised interest in quantum computing since the RSA public key encryption scheme is based on the presumed hardness of factoring, and with Shor's algorithm a quantum computer would be able to break RSA encryption. While Shor's algorithm has led to intensive efforts of building a quantum computer, factoring is not a long-term 'killer app' for quantum computers. Before we get close to building large enough quantum computers that can run Shor's algorithm on currently used key sizes, one can change to so-called 'post-quantum' public key encryption schemes that are not vulnerable to any known quantum attack [3].

Faced with the challenge of finding broad applications of quantum computers, and the difficulties of building large-scale quantum computers, the Canadian company D-Wave systems went a different path from academic research. Instead of trying to produce better and better quantum bits (qubits), which could ultimately be used to run quantum algorithms, they looked for application areas where a quantum device built from many, but imperfect, qubits could potentially be useful.

The D-Wave devices are designed to implement an approach called 'quantum annealing' to find the ground states of Ising spin glass problems. Ising spin glasses are models in statistical physics, consisting of N spins s_i ($i = 1, \dots, N$) that can take the values $+1$ or -1. The energy of a configuration is given by

$$H = \sum_{i<j} J_{ij} s_i s_j - \sum_i h_i s_i, \tag{1}$$

with arbitrary coupling constants J_{ij} and h_i. Finding the ground state of an Ising spin glass, i.e. the assignments of s_i that minimize the energy, is an NP-complete problem [4], which makes any new approach to solving these Ising spin glasses interesting to the optimization community. In fact, the widely used algorithm of simulated annealing was initially proposed for Ising spin glasses [5]. In simulated annealing, a Monte Carlo simulation of the Ising spin glass is performed, slowly cooling the system. Using thermal fluctuations the configuration of the system can escape a local minimum, by climbing over the barrier separating it from neighbouring local (or global) minima.

A quantum system can escape a local energy also by 'tunneling' through barriers, that is without needing the energy to climb over the barrier. Especially for narrow and tall barriers, quantum tunneling is more efficient than thermal excitation to the top of the barrier. It has thus been proposed that a quantum-mechanical version of simulated annealing, called 'quantum annealing' [6, 7, 8] may be more efficient than classical annealing in finding the ground states of hard problems [9, 10]. Quantum annealing may not require perfect qubits, since even an imperfect quantum annealer may profit from quantum effects in finding ground states more efficiently than a classical algorithm.

D-Wave's approach in being the first company to bring quantum computers to the market has been to aim to implement a quantum annealer with more than one thousand imperfect qubits, which should be compared to academic efforts that develop substantially better qubits, but are limited to about a dozen (and which can still be easily simulated classically). The company and its product have received a very skeptical reception, with doubts being raised if the qubits are even good enough to be called quantum and not just noisy classical bits. With the sale of two devices to Lockheed Martin corporation and Google, researchers have now been able to investigate these devices and shed some light on their potential.

While there is evidence for quantum effects being present in the D-Wave devices [11] and their performance on random problems has been consistent with that of a quantum annealer [12], there also exists an effective classical description for its performance on spin glass instances [13], which raises doubts about how much quantum effects can help the device solve problems more efficiently than a classical one. More importantly, no evidence for so-called 'quantum speedup' was seen [14], which would refer to a more benign scaling of the time to solution as a function of problem size N on a quantum device compared to the best classical algorithm. At this time we can only speculate about the reasons for not seeing quantum speedup. Potential explanations might be that the qubits are not 'coherent' enough, i.e. they do not stay quantum long enough to give an advantage. Alternatively it has also been suggested that the benchmark problems studied so far may not profit from quantum tunneling since the barriers may be very shallow and wide (recall that quantum tunneling is most efficient for tall and narrow barriers) [15]. The controversy about the D-Wave devices is ultimately rooted in the absence of any mathematical proof of advantages of quantum annealing in application problems. In the absence of rigorous results, the only way to ultimately answer this question requires the development of better quantum annealers and a search for problem classes with the potential for quantum speedup.

Even in the absence of any scaling advantage, a quantum annealer may still be useful if it solves a problem more efficiently (in terms of time to solution or power consumption) than a classical algorithm. While optimized special-purpose codes on a single Intel Xeon CPU perform slightly better than the 512-qubit D-Wave Two device [12, 14] there have been claims that D-Wave outperforms the best classical general purpose optimizers, and in particular CPLEX, by a factor of 3600 and more [16]. This claim is questioned in the article by Dash and Puget in this issue, who present the performance of CPLEX on these problems, if CPLEX is used in an optimal way.

This study is important in several ways. While for the scientist the right question to ask may be the comparison of a special-purpose device such as D-Wave to a special-purpose code such as those used in Ref. [12, 14], for the practitioner the comparison of a commercial device from D-Wave to the best general-purpose commercial code may make sense. More broadly, the discussion about the performance of the D-Wave devices, shows the challenge faced by new technology, such as quantum annealers or quantum computers, to compete with well established and highly optimized classical hardware and software. Fair benchmarks such as the ones by Dash and Puget, comparing new devices to existing algorithms run in the optimal way, are required to assess the potential for new technologies.

Prof. Dr. Matthias Troyer, Institut für Theoretische Physik, ETH Zurich,
Wolfgang-Pauli-Strasse 27, 8093 Zurich, Switzerland
matthias.troyer@itp.phys.ethz.ch

Bibliography

[1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467–488, 1982.

[2] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pp. 124–134, 1994.

[3] D. J. Bernstein, J. Buchmann, and E. Dahmen, eds., *Post-quantum cryptography*. Springer-Verlag, 2009.

[4] F. Barahona, “On the computational complexity of Ising spin glass models,” *Journal of Physics A: Mathematical and General*, vol. 15, no. 10, p. 3241, 1982.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, May 1983.

[6] P. Ray, B. K. Chakrabarti, and A. Chakrabarti, “Sherrington-Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations,” *Physical Review B*, vol. 39, pp. 11828–11832, June 1989.

[7] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, “Quantum Annealing - a New Method for Minimizing Multidimensional Functions,” *Chemical Physics Letters*, vol. 219, no. 5-6, pp. 343–348, 1994.

[8] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” *Physical Review E*, vol. 58, pp. 5355–5363, Nov. 1998.

[9] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, “A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem,” *Science*, vol. 292, pp. 472–476, 2001.

[10] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, “Theory of quantum annealing of an ising spin glass,” *Science*, vol. 295, no. 5564, pp. 2427–2430, 2002.

[11] T. Lanting, A. J. Przybysz, A. Y. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, and G. Rose, “Entanglement in a quantum annealing processor,” *Phys. Rev. X*, vol. 4, p. 021041, May 2014.

[12] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, “Evidence for quantum annealing with more than one hundred qubits,” *Nature Physics*, vol. 10, pp. 218–224, Mar. 2014.

[13] S. W. Shin, G. Smith, J. A. Smolin, and U. Vazirani, “How “quantum" is the D-Wave machine?,” *arXiv:1401.7087*, 2014.

[14] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, “Defining and detecting quantum speedup,” *Science*, vol. 345, no. 6195, pp. 420–424, 2014.

[15] H. G. Katzgraber, F. Hamze, and R. S. Andrist, “Glassy Chimeras Could Be Blind to Quantum Speedup: Designing Better Benchmarks for Quantum Annealing Machines,” *Physical Review X*, vol. 4, p. 021008, Apr. 2014.

[16] C. C. McGeoch and C. Wang, “Experimental evaluation of an adiabatic quantum system for combinatorial optimization,” in *Proceedings of the 2013 ACM Conference on Computing Frontiers*, 2013.

In memoriam Manfred Padberg

On May 12, 2014, Manfred Padberg, one of the pioneers of integer programming, passed away. In January 2015, the organizers of the Workshop on Combinatorial Optimization *in Aussois, France, Michael Jünger, Giovanni Rinaldi, and Gerhard Reinelt, dedicated one morning to his memory. Some of his closest scientific friends and collaborators honored Manfred Padberg by recalling his ground breaking scientific contributions and sharing memories and pictures with the audience. We are very grateful to them for allowing us to use some of the material in this issue of our newsletter. In particular, we thank Egon Balas (Manfred Padberg's doctoral advisor) for allowing us to reprint his speech, which, unfortunateley, he could not give himself during the meeting, and to Martin Grötschel for providing the scientific curriculum vitae you find below.*

Scientific curriculum vitae

1961–1967	Studies of mathematics at U Münster (Diploma)
1967–1968	Research assistant at U Mannheim
1968–1971	Masters' degree and doctorate (1971) at Carnegie Mellon University, Pittsburgh
1971–1974	Research fellow at International Institute of Management, Berlin, Germany
1973–1974	Guest professor at U Bonn
1974–2002	Associate/full professor at Stern School of Business, NYU
1983	Lanchester Prize of ORSA
1985	George B. Dantzig Prize of MPS/SIAM
1989	Alexander von Humboldt Senior US Scientist Research Award
2000	John von Neumann Theory Prize (INFORMS)
2002	INFORMS Fellow

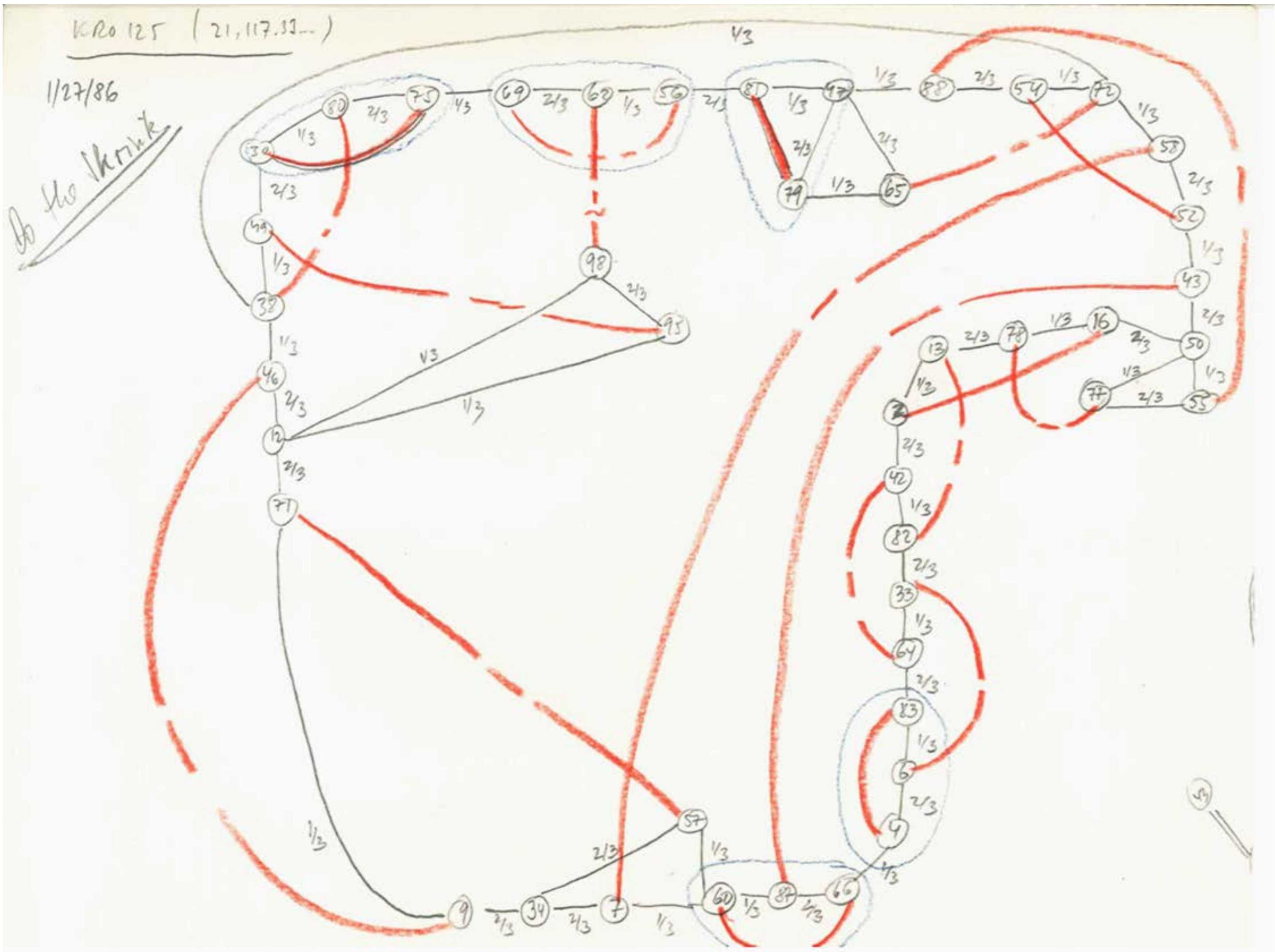
Manfred Padberg
(10 October 1941 – 12 May 2014)

Egon Balas

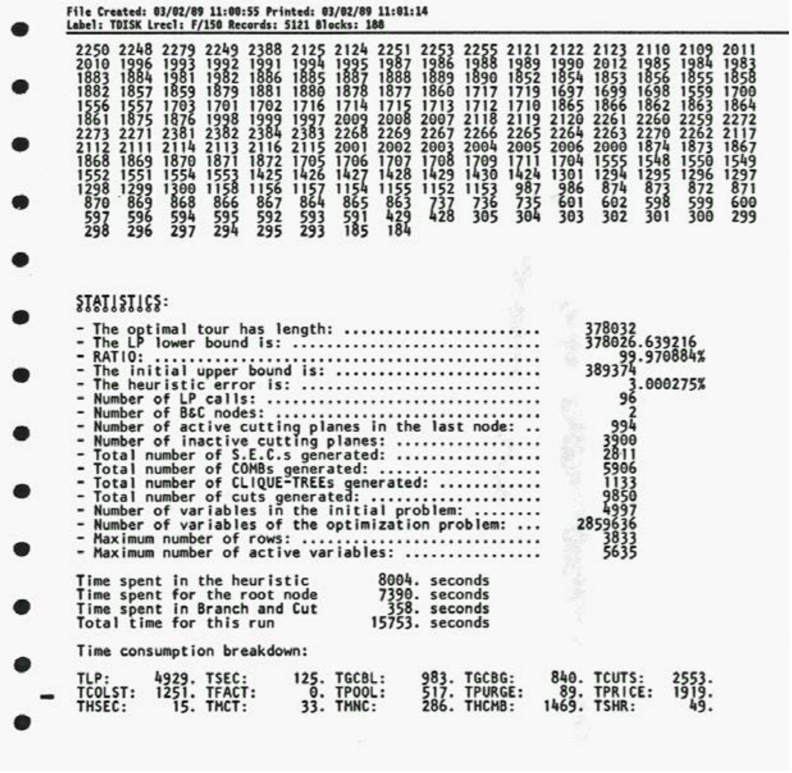
Manfred Padberg was my first and best doctoral student.

Almost half a century ago I became Manfred's professor and then his doctoral thesis advisor. This happened at GSIA, the Graduate School of Industrial Administration of Carnegie Mellon University, where Manfred studied for his PhD between 1968 and 1971. Those were heady days for me. I had joined the place just one year earlier, as a fresh immigrant from behind the Iron Curtain and was thoroughly enjoying my new freedom and the wonderful research environment. One day a young German showed up at my office, with a Diplom in Mathematics from Münster and with a Ford Foundation fellowship to study in the US for a doctorate, for which he chose the area of Operations Research.

The two of us hit it off well from the beginning. As I later found out others did not do so well in getting along with Manfred, but the fact that he was my first doctoral student may have facilitated the process. It seems that we have entered each other's lives at the appropriate moment. We worked closely together for four years and developed an interaction from which I certainly benefitted a lot, given Manfred's talent and enthusiasm for our subject, which of course was integer programming. I am talking about enthusiasm, but it was more than that: Manfred did have fire in his belly, as they say. If the proof of a theorem or a step in the proof was open or in doubt at midnight, he would not go to sleep until the question was settled. As to Manfred's perception of our relationship, in the preface of his wonderful book on Linear Optimization and Extensions, Manfred talks about “the ‘invisible hand' of Egon … whose enthusiasm and superb teaching of the subject literally got me hooked on linear and combinatorial optimization”. If anybody else had called my teaching superb, I would have taken it as a routine compliment, but Manfred can certainly not be accused of throwing around praise and kind words too easily. So when I read this I thought to myself that, maybe, I should reconsider my bias towards research in the “teaching-versus-research” equation: if indeed I got Manfred Padberg



"TSP-art" – this is how Giovanni Rinaldi calls Manfred Padberg's drawings like this one visualizing fractional TSP-solutions and created in order to solve separation problems. Probably nobody would complain if these pieces were hanging in The Museum of Modern Art.

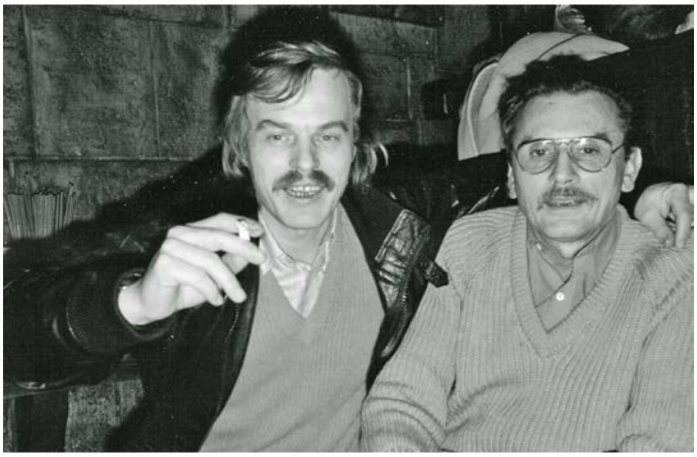


```
File Created: 03/02/89 11:00:55 Printed: 03/02/89 11:01:14
Label: TDISK Lrecl: F/150 Records: 5121 Blocks: 188

 2250 2248 2279 2249 2388 2125 2124 2251 2253 2255 2121 2122 2123 2110 2109 2011
 2010 1996 1993 1992 1991 1994 1995 1987 1986 1988 1989 1990 2012 1985 1984 1983
 1883 1884 1981 1982 1886 1885 1887 1888 1889 1890 1852 1854 1853 1856 1855 1858
 1882 1857 1859 1879 1881 1880 1878 1877 1860 1717 1719 1697 1699 1698 1559 1700
 1556 1557 1703 1701 1702 1716 1714 1715 1713 1712 1710 1865 1866 1862 1863 1864
 1861 1875 1876 1998 1999 1997 2009 2008 2007 2118 2119 2120 2261 2260 2259 2272
 2273 2271 2381 2382 2384 2383 2268 2269 2267 2266 2265 2264 2263 2270 2262 2117
 2112 2111 2114 2113 2116 2115 2001 2002 2003 2004 2005 2006 2000 1874 1873 1867
 1868 1869 1870 1871 1872 1705 1706 1707 1708 1709 1711 1704 1555 1548 1550 1549
 1552 1551 1554 1553 1425 1426 1427 1428 1429 1430 1424 1301 1294 1295 1296 1297
 1298 1299 1300 1158 1156 1157 1154 1155 1152 1153  987  986  874  873  872  871
  870  869  868  866  867  864  865  863  737  736  735  601  602  598  599  600
  597  596  594  595  592  593  591  429  428  305  304  303  302  301  300  299
  298  296  297  294  295  293  185  184


 STATISTICS:

 - The optimal tour has length: .......................    378032
 - The LP lower bound is: .............................    378026.639216
 - RATIO: .............................................        99.970884%
 - The initial upper bound is: ........................    389374
 - The heuristic error is: ............................         3.000275%
 - Number of LP calls: ................................        96
 - Number of B&C nodes: ...............................         2
 - Number of active cutting planes in the last node: ..       994
 - Number of inactive cutting planes: .................      3900
 - Total number of S.E.C.s generated: .................      2811
 - Total number of COMBs generated: ...................      5906
 - Total number of CLIQUE-TREEs generated: ............      1133
 - Total number of cuts generated: ....................      9850
 - Number of variables in the initial problem: ........      4997
 - Number of variables of the optimization problem: ...   2859636
 - Maximum number of rows: ............................      3833
 - Maximum number of active variables: ................      5635

 Time spent in the heuristic          8004. seconds
 Time spent for the root node         7390. seconds
 Time spent in Branch and Cut          358. seconds
 Total time for this run             15753. seconds

 Time consumption breakdown:

 TLP:      4929. TSEC:      125. TGCBL:      983. TGCBG:     840. TCUTS:   2553.
 TCOLST:   1251. TFACT:       0. TPOOL:      517. TPURGE:     89. TPRICE:  1919.
 THSEC:      15. TMCT:       33. TMNC:       286. THCMB:    1469. TSHR:      49.
```

An incredible world record! In 1986 Manfred Padberg and Giovanni Rinaldi solved to optimality a TSP-instance on 2392 nodes, thus pushing the size of relevant integer programs which might be conceivable to be solved to optimality beyond everything people could imagine before. The 1986 computations were performed at the National Bureau of Standards in Washington DC; the printout is from a repeated computation done in 1989 at IBM Thomas J. Watson Research Center, New York.

During the 80's large parts of the most important and influential work in integer programming were done in the group around Manfred Padberg at NYU. The picture shows (from left to right) Giovanni Rinaldi, Michele Conforti, Monique Laurent, Ram Rao, and Manfred Padberg.

In 1989, a few months before the Berlin wall came down, Manfred Padberg together with his first doctoral student Martin Grötschel, his first two doctoral grand children Michael Jünger and Gerhard Reinelt, and his (non-doctoral) son Marc Oliver visited East Germany in order to give lectures on Combinatorial Optimization. The picture was taken at „Auerbach's Keller" in Leipzig, a pub and restaurant that had been famous since 200 years before due to one scene of the probably most famous German play „Faust" by Johann Wolfgang von Goethe.



Claude Berge was one of Manfred Padberg's closest friends. The investigation of polyhedral aspects of perfect graphs (which were invented and pioneered by Berge) became a driving momentum in the development of integer programming and polyhedral combinatorics.

For Manfred Padberg's 60th birthday in 2001, Martin Grötschel and his group at Zuse-Institute Berlin organized a wonderful day of lectures and celebration that ended in a big party on a boat at Spree river. On this occasion, Martin Grötschel reported about his historical research revealing that Manfred Padberg descends from an old family of robber barons of the Sauerland region in Westphalia, his family name still being present in the name of a part of the town of Marsberg.

hooked on Combinatorial Optimization, and Manfred got Martin Grötschel hooked, and Martin got Michael Jünger and Gerd Reinelt hooked, and so on, then who knows what is more important for the future of the profession: the research, the facts and properties that it establishes, or the teaching, the talent that it attracts and sometimes "gets hooked"?

This is not the place to enumerate Manfred's contributions to our field, but let me mention a few crucial facts. In his early work, Manfred discovered some basic properties of set partitioning polytopes, like the fact that any vertex can be reached from any other vertex by fewer integer pivots than the number of equations. His characterization of perfect zero-one matrices reinforced the already existing ties between graph theory and 0–1 programming. Padberg's focus on characterizing the facets of various combinatorial polyhedra set a trend. His work with Groetschel laid the foundations of the polyhedral study of the traveling salesman polytope. These and other discoveries of Padberg and his coauthors paved the way towards the larger use of polyhedral methods in integer programming.

Padberg is the originator and main architect of the approach known as branch-and-cut. Concentrating on the traveling salesman problem as their main testbed, Padberg and Rinaldi have shown that if cutting planes generated at various nodes of a search tree can be lifted so as to be valid everywhere, then interspersing them with branch-and-bound yields procedures that vastly amplify the power of either enumeration or cuts by themselves. One of the basic discoveries of the 1980's in the realm of combinatorial optimization, arrived at by three different groups of researchers in the wake of the advent of the ellipsoid method, was the equivalence of optimization and separation: Padberg and Rao were one of these groups.

Padberg's work combines theory with algorithm development and computational testing in the best tradition of Operations Research. In his joint work with several researchers he set an example of how to formulate and handle efficiently very large scale 0–1 programs with practical applications in industry and transportation. The fields of Combinatorial Optimization, Integer Programming and, more generally, Operations Research, will badly miss him.

2015 MOS Elections

We are pleased to announce the results of the 2015 MOS Elections. Thanks to everyone who voted, and special thanks to the great slate of nominated candidates who agreed to stand for election. Here are the results:
MOS Chair: Karen Aardal
MOS Treasurer: Marina A. Epelman
MOS Council Members-at-Large: Shabbir Ahmed, Roberto Cominetti, Jacek Gondzio, Adrian Lewis

Karen Aardal will be the Vice Chair starting after the ISMP, and she will be the Chair starting July 2016.
Marina Epleman will be the Treasurer starting July 2016.
The four elected Council Members will begin their terms after the ISMP.

William Cook, MOS Chair

ICSP 2016 – Búzios, Brazil

The XIV International Conference on Stochastic Programming will take place in Búzios, a charming town 160 km from Rio de Janeiro, Brazil, from June 25 to July 1, 2016.

A two-day introductory series of tutorials held during the weekend precedes the main conference to provide sufficient background in Stochastic Programming. The conference is developed over the next five week days with parallel sections, plenary talks from recognized leading researchers in stochastic optimization and a set of mini symposia, featuring recent contributions on state-of-the-art topics.

To keep registration fees low, we have negotiated preferential rates with hotel Atlântico for the lecture rooms, in exchange of certain minimal rooms occupancy. Hotel Atlântico Búzios is a deluxe resort by the sea, with an excellent restaurant. To encourage ICSP 2016 participants to choose hotel Atlântico for their venue, we offer two perks: R$300 (ca. US$100) discount in the registration fee for attendants staying in the hotel; the negotiated hotel rates include also lunch, a very good deal! Please consider staying in hotel Atlântico to help our organization!

Beware of the early submission deadlines; this is done to allow conference participants to make travel arrangements with anticipation, since prices may rise in 2016 due to the realization of the Olympic Games in Rio in August. The deadline for registration payment at early-bird rate is December 15th, 2015.

The ICSP is the premier event of the Stochastic Programming Society (SPS), a technical section of the Mathematical Optimization Society, that brings together researchers who work on decisions under uncertainty. The 2016 conference will be the first one held in South America and will be jointly organized by Brazil and Chile.

ICSP 2016 is the closing event of SVAN 2016, a full trimester organized at IMPA on Stochastic Variational Analysis, click here for details on this thematic program.

Plenary Speakers

- Shabbir Ahmed (Georgia Institute of Technology, USA)
  Mixed-Integer Models in Stochastic Programming
- Johannes Royset (Naval Postgraduate School, USA)
  Designing Uncertainty Models in Stochastic Programming
- Andrzej Ruszczyński (Rutgers University, USA)
  Risk Models for Stochastic Programming
- Mario Pereira (PSR, Brazil)
  Stochastic Programming Models for Energy Planning

Minisymposia Organizers

- Güzin Bayraksan (Ohio State University, USA)
  Data-driven Methods for Stochastic Programming
- David Brown (Duke University, USA)
  Advances in Stochastic Dynamic Programming
- Miloš Kopa (Charles University in Prague, Czech Republic)
  Applications of Stochastic Programming in Finance
- Warren Powell (Princeton University, USA)
  Applications of Machine Learning in Stochastic Optimization
- David Woodruff (University of California, Davis, USA)
  Applications of Stochastic Programming in Natural Resources
- Huifu Xu (University of Southampton, UK)
  Stochastic Equilibrium and Stochastic Variational Inequalities

Tutorials

- Roger J-B Wets (University of California, Davis, USA)
  Introduction to Stochastic Programming (Saturday afternoon)
- Simge Küçükyavuz (Ohio State University, USA)
  Stochastic Integer Programming (Sunday morning)
- Alexander Shapiro (Georgia Institute of Technology, USA)
  Risk measures (Sunday afternoon)

http://icsp2016.sciencesconf.org

ICCOPT 2016 – Tokyo, Japan

The Fifth International Conference on Continuous Optimization (ICCOPT 2016) will take place in Tokyo, Japan, from August 6 to August 11, 2016. ICCOPT is a flagship conference of the Mathematical Optimization Society, organized every three years. It is designed to provide ample opportunities in which researchers and practitioners in continuous optimization can exchange ideas, techniques and applications.

ICCOPT 2016 consists of Conference and Summer School. The Conference (August 8–11) comprehends a series of plenary and semi-plenary talks, organized and contributed sessions, and poster sessions. It will be held on the campus of the National Graduate Institute for Policy Studies (GRIPS) located at Roppongi, the most advanced fashionable central area in Tokyo. The Summer School (August 6–7) is directed to students and researchers in continuous optimization and related fields. It will be held on the National Olympics Memorial Youth Center (NYC) located at Yoyogi.

The meeting is chaired by Shinji Mizuno (Organizing Committee) and Yinyu Ye (Program Committee) and locally coordinated by Takashi Tsuchiya (Local Organizing Committee).

Tokyo is one of the largest cities in the world with full of exciting events, attractions and entertainments! At the same time, the city is quite safe and well-organized. The venue (GRIPS) is in an unbeatable location with an exciting neighborhood. The area has a bunch of nice restaurants which are familiar with foreign guests. We are sure that participants will enjoy both academic events in the daytime and social events in the evening. We hope that researchers and practitioners from all over the world get together in Tokyo on this occasion.

Important Deadlines
March 15: Summer School Accommodation Reservation for Students
April 15: Abstract Submission for Parallel Session, Summer School Accommodation Reservation for Non-students
May 16: Abstract Submission for Poster Session
May 31: Registration for Presenting Authors and Early Registration

Organizing Committee: support@iccopt2016.tokyo
Further information: www.iccopt2016.tokyo

IPCO 2016 – Liège, Belgium

The 18th conference on Integer Programming and Combinatorial Optimization (IPCO) will be held in Liège, Belgium. Authors are invited to submit extended abstracts of their recent work by November 20, 2016. Submission details and other information can be found at www.ipco2016.be.

The aim of the conference is to present original unpublished work in various aspects of integer programming and combinatorial optimization: theory, computation and applications of discrete optimization are most welcome.

Program committee: ○ Karen Aardal (Delft) ○ Daniel Bienstock (Columbia) ○ José Correa (Chile) ○ Oktay Günlük (IBM) ○ Satoru Iwata (Tokyo) ○ Volker Kaibel (Magdeburg) ○ Jochen Könemann (Waterloo) ○ Andrea Lodi (Bologna/Montréal) ○ Quentin Louveaux (Liège) ○ Gianpaolo Oriolo (Tor Vergata) ○ András Sebő (Grenoble) ○ Bruce Shepherd (McGill) ○ Martin Skutella (TU Berlin, chair) ○ Leen Stougie (VU Amsterdam) ○ Gerhard Woeginger (Eindhoven)

Organizing committee: ○ Yves Crama (Liège) ○ Quentin Louveaux (Liège) ○ Mathieu Van Vyve (Louvain) ○ Laurence Wolsey (Louvain)

Important dates

Submission:	November 20, 2015
Summer School:	May 30–31, 2016
Conference:	June 1–2–3, 2016

Website: www.ipco2016.be

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.*

E-mail

Signature

Mail to:
Mathematical Optimization Society
3600 Market St, 6th Floor
Philadelphia, PA 19104-2688
USA

Cheques or money orders should be made payable to The Mathematical Optimization Society, Inc. Dues for 2015, including subscription to the journal *Mathematical Programming*, are US $ 90. Retired are $ 45. Student applications: Dues are $ 22.50. Have a faculty member verify your student status and send application with dues to above address.

Faculty verifying status

Institution

IMPRINT
Editor: Volker Kaibel, Institut für Mathematische Optimierung, Otto-von-Guericke Universität Magdeburg, Universitätsplatz 2, 39108 Magdeburg, Germany. kaibel@ovgu.de ■ **Co-Editors:** Samuel Burer, Department of Management Sciences, The University of Iowa, Iowa City, IA 52242-1994, USA. samuel-burer@uiowa.edu ■ Jeff Linderoth, Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706-1572, USA. linderoth@wisc.edu ■ **Founding Editor:** Donald W. Hearn ■ Published by the Mathematical Optimization Society ■ Design and typesetting by Christoph Eyrich, Berlin, Germany. optima@0x45.de ■ Printed by Oktoberdruck AG, Berlin, Germany.