
Volume 37, Issue 3, May 2009
ISSN 0167-6377
ELSEVIER
Operations Research Letters
EDITOR-IN-CHIEF
Jan Karel Lenstra
AREA EDITORS
approximation and heuristics
Gerhard Woeginger
book reviews
Frits Spieksma
continuous optimization
Patrice Marcotte
financial engineering
Jussi Keppo
game theory
Nimrod Megiddo
graphs and networks
Gianpaolo Oriolo
inventory, reliability and control
Sridhar Seshadri
linear and stochastic optimization
Rüdiger Schultz
logistics and revenue management
Retsef Levi
mixed integer optimization
Christoph Helmberg and Alexander Martin
sequencing and scheduling
Steef van de Velde
stochastic networks and queues
Sem Borst and Tetsuya Takine
www.elsevier.com/locate/orl

Operations Research Letters 37 (2009) 187–191

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

A strong conic quadratic reformulation for machine-job assignment with controllable processing times

M. Selim Aktürk [a], Alper Atamtürk [b,*], Sinan Gürel [a]

[a] Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey

[b] Department of Industrial Engineering & Operations Research, University of California, Berkeley, CA 94720-1777, USA

ARTICLE INFO

Article history:
Received 24 January 2008
Accepted 3 December 2008
Available online 30 January 2009

Keywords:
Separable convex functions
Conic quadratic integer programming

ABSTRACT

We describe a polynomial-size conic quadratic reformulation for a machine-job assignment problem with separable convex cost. Because the conic strengthening is based only on the objective of the problem, it can also be applied to other problems with similar cost functions. Computational results demonstrate the effectiveness of the conic reformulation.

1. Introduction

We consider a machine-job assignment problem with controllable processing times arising in flexible manufacturing systems. Processing times on computer numerically controlled (CNC) machines can be compressed by increasing the cutting speed and the feed rate at a convex increasing cost for compression. Thus, when processing time becomes a decision variable, one is faced with a trade-off between increasing yield and cost of machining, which can be modeled as a nonlinear mixed 0–1 profit maximization problem.

If compression of processing times is not allowed, the machine-job assignment problem reduces to the classical generalized assignment problem, which is $\mathcal{NP}$-hard as it contains the 0–1 knapsack problem. In practice, the nonlinearity of the compression cost makes this assignment problem particularly difficult to solve. Even for the quadratic case, commercially available software packages that employ fast quadratic programming (QP) algorithms within a branch-and-bound framework are far from solving large instances of the problem.

In this paper we reformulate the problem using a polynomial number of conic quadratic constraints [1,2]. Our approach for developing conic reformulations is analogous to the polyhedral approach for linear integer programming with the goal of strengthening bounds from continuous relaxations of the problem. We construct strong conic reformulations based on the convex hull description of a simple mixed integer set defined by nonlinear inequalities. We refer the reader to [3,4] for convexification techniques for nonlinear integer programs.

Whereas semidefinite programming relaxations of max-cut and related combinatorial problems have been investigated extensively (e.g., [5–8]), research on conic mixed-integer programming is so far fairly limited. Çezik and Iyengar [9] describe Chvátal–Gomory and disjunctive cuts for conic integer programs. Atamtürk and Narayanan [10] give nonlinear conic mixed-integer rounding cuts for conic mixed-integer programming. Atamtürk and Narayanan [11] describe lifting techniques for conic integer programming. Whereas these papers develop cuts for general conic mixed-integer programs, in this study we exploit the structure of a certain objective function in order to derive strong conic formulations.

Two recent papers study a similar structure and propose alternative approaches to the one given here. Frangioni and Gentile [12] describe an interesting cutting plane procedure based on linear outer approximations of the perspective of convex functions and apply it to the unit commitment problem with a quadratic cost function. Günlük et al. [13] give problem-specific linear and nonlinear cuts for a quadratic cost facility location problem. Although in the current paper we apply the conic strengthening to the machine-job assignment problem with controllable processing times, because the conic reformulation technique is based on only the objective function of the problem, it can also be applied to other mixed 0-1 optimization problems with similar cost functions, including those studied in these two recent papers.

The machine-job assignment problem with controllable processing times arises in flexible manufacturing systems, where the processing times of machines are numerically controlled. In such

* Corresponding author.
E-mail addresses: akturk@bilkent.edu.tr (M.S. Aktürk), atamturk@berkeley.edu (A. Atamtürk), sgurel@bilkent.edu.tr (S. Gürel).

0167-6377/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2008.12.009

systems one employs a host of non-identical machines each having different applicable machining power levels. The high cost of investment in a flexible manufacturing system necessitates careful planning and scheduling of jobs on the machines as discussed in Gürel and Aktürk [14].

In the scheduling literature Vickson [15] was the first to consider controllable processing times. In recent years there has been a growing interest in controllable processing times. For a similar problem to ours with linear processing cost functions, Trick [16] provides certain optimality properties and heuristic algorithms based on these properties. We refer the reader to Shabtay and Steiner [17] for a recent survey of related studies.

This paper is organized as follows. In Section 2 we give the definition of the machine-job assignment problem with controllable processing times and a nonlinear mixed 0–1 programming formulation for it. In Section 3 we describe the conic strengthening in general and apply it to the machine-job assignment problem. In Section 4 we present extensive computational results on the introduced formulations.

2. Problem definition

Given n jobs and m non-identical parallel machines with finite capacity, the machine-job assignment problem is to choose a subset of the jobs and assign them to the machines so that the total profit from the assignment is maximized. Letting c_i denote the available machining time for machine $i = 1, \ldots, m$, and p_{ij} and h_{ij}, the regular processing time and profit corresponding to job j if it is assigned to machine i, the problem can be modeled as a linear 0–1 program. This problem is also referred to as the generalized assignment problem [18].

In a flexible manufacturing system, where jobs are processed on computer numerically controlled (CNC) machines, processing times can be reduced by appropriately setting the machining parameters such as cutting speed and feed rate. However, compressing processing time naturally leads to reduced tool life, and, consequently, increased machining cost. We model the change in the machining cost due to processing time compression $y \geq 0$ as $f(y) = ky^{a/b}$, where a and b are integers satisfying $a \geq b > 0$ and $k > 0$, so that f is an increasing and convex function of compression. The function f reflects the relationship between compression and cost in that as one decreases the processing time of a job, it becomes more expensive to compress it further. Technical specifications of a job such as its length, diameter, required surface quality, as well as machine and tool type determine the cost coefficients k, a, and b. Defining a binary assignment variable x_{ij} equal to 1 if job j is assigned to machine i and 0 otherwise, and compression variable y_{ij} for each machine-job pair, the *machine-job assignment problem with controllable processing times* can be formulated as the following nonlinear mixed 0-1 program:

$$\max \sum_{i=1}^{m}\sum_{j=1}^{n}(h_{ij}x_{ij} - f_{ij}(y_{ij}))$$

$$\text{s.t.} \sum_{j=1}^{n}(p_{ij}x_{ij} - y_{ij}) \leq c_i, \qquad i = 1, \ldots, m, \tag{1}$$

$$\text{(MJ0)} \qquad \ell_{ij}x_{ij} \leq y_{ij} \leq u_{ij}x_{ij}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n, \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, \qquad j = 1, \ldots, n, \tag{3}$$

$$x_{ij} \in \{0, 1\},\ y_{ij} \in \mathbb{R}_+, \qquad i = 1, \ldots, m,\ j = 1, \ldots, n. \tag{4}$$

Constraint (1) ensures that the jobs assigned to machine i take no more than the machine capacity c_i. Constraint (2) ensures that compression is allowed on the processing time of job j on machine i only if job j is assigned to machine i and that compression is within specified limits $0 \leq \ell_{ij} \leq u_{ij} < p_{ij}$. Finally, constraint (3) guarantees that each job is assigned to at most one machine.

MJ0 is $\mathcal{NP}$-hard as it reduces to the generalized assignment problem when all u_{ij}'s are zero. The nonlinearity introduced with the option of compression of processing times makes the problem much harder to solve, in practice, compared to the generalized assignment problem. Note that MJ0 is a maximization problem with a concave objective and the feasible set of its continuous relaxation

$$P = \left\{ (x, y) \in \mathbb{R}^{2mn}_+ : (1), (2), (3) \right\}$$

is a polyhedron. In contrast to the case of generalized assignment problem, optimal solutions to its continuous relaxation are found typically in the interior of this polyhedron or almost all x_{ij} are fractional. Consequently, branch-and-bound algorithms based on such relaxations require excessive branching to find feasible integer solutions. Even when f is quadratic, i.e., $a/b = 2$, it is a challenge to solve practical-size instances of MJ0 with quadratic MIP solvers of commercial software packages. We will elaborate on the computational difficulty of solving MJ0 in Section 4.

Rather than developing a special purpose algorithm for MJ0, our goal is to reformulate the problem so that its continuous relaxation is stronger and the formulation may be solved by readily available solvers of optimization software packages. In particular, we describe a conic quadratic relaxation for MJ0. The conic strengthening presented here is also applicable to other mixed 0-1 minimization problems with a similar objective function.

3. Conic reformulations

In this section we describe the conic strengthening and show how to express it using a polynomial number of conic quadratic constraints. For strengthening the formulation it is convenient to work with the epigraph of f. So, by introducing auxiliary variables $t_{ij} \in \mathbb{R}_+$ we bring the nonlinear objective into the constraints and linearize the objective of the formulation as

$$\max \qquad \sum_{i=1}^{m}\sum_{j=1}^{n}(h_{ij}x_{ij} - k_{ij}t_{ij})$$

$$\text{(MJ1)} \ \text{s.t.} \ y_{ij}^{a_{ij}/b_{ij}} \leq t_{ij}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n, \tag{5}$$

$$(1), (2), (3), (4).$$

MJ1 is not necessarily easier to solve than MJ0. On the contrary, solvers can usually deal with nonlinearity in the objective easier than nonlinearity in the constraints. MJ1 is an intermediate formulation that will enable us to derive a strong conic formulation.

For our purpose it suffices to concentrate on the mixed 0–1 set

$$C = \left\{ (x, y, t) \in \{0, 1\} \times \mathbb{R}_+ \times \mathbb{R}_+ : \ y^{a/b} \leq t,\ \ell x \leq y \leq ux \right\}$$

with $a \geq b > 0$ and $u \geq \ell \geq 0$. Observe that constraints of C are of the form (2), (4) and (5). The proposed strengthening is applicable to any optimization problem that contains C as a substructure. Consider solutions of C satisfying $y = ux$. It is easy to see that for $a > b$ and $u > 0$ each point on the curve defined as

$$L = \left\{(x, y, t) \in \mathbb{R}^3 : 0 < x < 1,\ y = ux,\ y^{a/b} = t\right\}$$

is an extreme point of the continuous relaxation of C. The set of points L is illustrated with the dashed curve in Fig. 1(a). Next we will reformulate C so that L is eliminated from its continuous relaxation.

3.1. Strengthening the continuous relaxation

First, observe that for C, as $y, t \geq 0$ and $b > 0$, inequality $y^{a/b} \leq t$ is equivalent to

$$y^a \leq t^b. \tag{6}$$

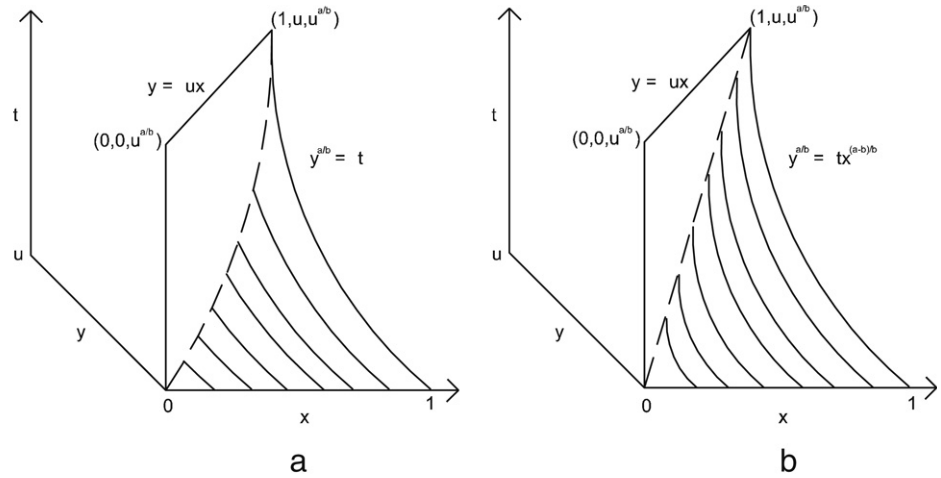

Fig. 1. Surfaces defined by inequalities (6) and (7).

We propose to strengthen (6) as

$$y^a \leq t^b x^{a-b}. \tag{7}$$

Because $a \geq b$, for $0 \leq x \leq 1$ inequality (7) implies (6). It is also clear that (7) is valid for C as for $x \in \{0, 1\}$ it reduces to (6). Thus, we may replace (6) with (7). Consider, then, the strengthened continuous relaxation of C:

$$C_S = \left\{(x, y, t) \in \mathbb{R}^3 : y^a \leq t^b x^{a-b}, \right.$$
$$\left. \ell x \leq y \leq ux,\ 0 \leq x \leq 1,\ 0 \leq t \right\}.$$

Although (7) is highly nonlinear, C_S is a convex set. Indeed, it is easy to show that C_S is the smallest convex relaxation of C.

Proposition 1. *The convex hull of C, $\text{conv}(C)$, equals C_S.*

Proof. Consider the disjunction $C_0 \cup C_1$, where $C_0 := \{(x, y, t) \in C : x = 0\}$ and $C_1 := \{(x, y, t) \in C : x = 1\}$; thus, $C = C_0 \cup C_1$. To see that $\text{conv}(C) \subseteq C_S$ consider points $(0, 0, t_0) \in C_0$, $(1, y_1, t_1) \in C_1$, and a convex combination

$$(x, y, t) = (1 - \lambda)(0, 0, t_0) + \lambda(1, y_1, t_1)$$
$$= (\lambda, \lambda y_1, (1 - \lambda)t_0 + \lambda t_1)$$

for some $0 \leq \lambda \leq 1$. Clearly, $t \geq 0$, $0 \leq x = \lambda \leq 1$, and $\ell x = \lambda \ell \leq \lambda y_1 = y \leq \lambda u = ux$. To see that (7) holds as well for (x, y, t), observe that

$$(\lambda y_1)^a = (\lambda^b y_1^a)(\lambda^{a-b}) = \left((1 - \lambda)0 + \lambda y_1^{a/b}\right)^b \lambda^{a-b}$$
$$\leq ((1 - \lambda)t_0 + \lambda t_1)^b \lambda^{a-b},$$

where the last inequality holds from $0 \leq t_0$ and $y_1^{a/b} \leq t_1$.

For $C_S \subseteq \text{conv}(C)$, consider an arbitrary point $(x, y, t) \in C_S$. If $x = 0$ or $x = 1$, then $(x, y, t) \in C \subseteq \text{conv}(C)$ trivially. On the other hand, if $0 < x < 1$, then (x, y, t) is a convex combination of $(0, 0, 0) \in C_0$ and $(1, y/\lambda, t/\lambda)$ with $\lambda = x$. To see that the latter point is in C_1, observe that $y/\lambda \leq u$ and $(y/\lambda)^a \leq (t/\lambda)^b$, or $y^a \leq t^b \lambda^{a-b}$ as $(x, y, t) \in C_S$. $\square$

Inequality (7) is illustrated in Fig. 1(b). This figure shows that (7) defines the curved boundary of $\text{conv}(C)$ and that any point (x, y, t) on it with $0 \leq x \leq 1$ is a convex combination of C_0 and C_1.

It should be clear from the proof of Proposition 1 that inequality (7) indeed defines the epigraph of the perspective of f. Recently, Frangioni and Gentile [12] proposed an interesting cut generation method based on perspective functions. Their approach is to generate supporting hyperplanes of the perspective of a convex function as cuts to improve relaxations with convex objective on a polyhedral set. Although this is a more general approach as it applies to any separable convex function, as also stated by Frangioni and Gentile in their computational study, there are practical difficulties with approximating the perspective with a large number of linear inequalities and solving a relaxed problem as this leads to finding infeasible integer solutions that need to be avoided. Here we use the nonlinear constraint (7) explicitly by reformulating it via conic quadratic constraints as discussed in Section 3.2.

3.2. Conic quadratic representation

Now we give an efficient representation of the set C_S using a polynomial number of conic quadratic constraints. As explained in [2], for a positive integer l, an inequality of the form

$$r^{2^l} \leq s_1 s_2 \cdots s_{2^l}, \tag{8}$$

for $r, s_1, \ldots, s_{2^l} \geq 0$ can be expressed equivalently using $O(2^l)$ variables and $O(2^l)$ hyperbolic inequalities of the form

$$u^2 \leq v_1 v_2, \quad u, v_1, v_2 \geq 0. \tag{9}$$

Furthermore, each constraint $u^2 \leq v_1 v_2$ can be written as a conic quadratic (second-order cone) constraint

$$\|(2u, v_1 - v_2)\| \leq v_1 + v_2. \tag{10}$$

Proposition 2. *For integral $a \geq b \geq 0$ inequalities*

$$y^a \leq t^b x^{a-b}, \quad x, y, t \geq 0$$

can be expressed equivalently using $O(\log_2 a)$ variables and $O(\log_2 a)$ conic quadratic constraints of the form (10) and $x, y, t \geq 0$.

Proof. For $l = \lceil \log_2 a \rceil$, using $y \geq 0$, we may rewrite constraint (7) as

$$y^{2^l} \leq t^b x^{a-b} y^{2^l - a}. \tag{11}$$

Now it is clear that (11) is a special case of (8) with $s_1 = \cdots = s_b = t$, $s_{b+1} = \cdots = s_a = x$, $s_{a+1} = \cdots = s_{2^l} = y$. Following the construction in [1], inequalities (9) can be built using a binary tree with leaf nodes for $1, t, t^2, \ldots, t^{2^{\lfloor \log_2(b) \rfloor}}, x, x^2, \ldots, x^{2^{\lfloor \log_2(a-b) \rfloor}}$, and $y, y^2, \ldots, y^{2^{l-1}}$. Each non-leaf node of the binary tree represents a new hyperbolic inequality (9) and variable introduced. Because the number of nodes in a binary tree is at most twice the number of its leaves, the number of inequalities and variables in the conic quadratic representation is at most $O(\log_2 a)$. $\square$

Observe that conic reformulations based on (6) can be obtained by simply fixing $x = 1$ in this derivation. We refer to the conic reformulation of MJ1 as CMJ1 and to the conic reformulation of

Table 1
Computational results for the quadratic case: $f(y) = ky^2$.

κ	n	m	MJ0					MJ1/CMJ1					CMJ2				
			rgap	egap	opt	nodes	cpu	rgap	egap	opt	nodes	cpu	rgap	egap	opt	nodes	cpu
		1	7.52	–	5	189	0	7.52	–	5	1,218	1	0.10	–	5	12	0
	50	5	12.51	–	5	128,323	132	12.51	0.43	3	323,893	464	3.65	–	5	457	1
		10	22.47	4.41	0	624,766	1043	22.47	3.79	0	392,314	1008	8.39	–	5	4,510	12
		1	6.09	–	5	3,712	1	6.09	–	5	16,783	20	0.05	–	5	10	0
0.1	100	5	9.37	2.58	0	672,453	1024	9.37	4.36	0	423,227	1008	1.17	–	5	22,821	43
		10	12.83	7.39	0	329,917	1030	12.83	9.24	0	151,869	1009	1.95	–	5	1,700	16
		1	6.02	1.08	3	989,865	613	6.02	1.74	0	484,487	1006	0.01	–	5	7	0
	200	5	8.98	5.61	0	361,756	1020	8.98	6.62	0	213,382	1011	0.30	–	5	33,813	139
		10	11.04	8.77	0	175,442	1015	11.04	10.26	0	106,441	1010	0.93	0.34	1	81,283	960
		1	4.78	–	5	479	0	4.78	–	5	3,142	3	0.05	–	5	6	0
	50	5	9.38	0.00	4	454,232	567	9.38	2.64	0	419,522	1006	1.01	–	5	1,385	3
		10	13.29	5.30	0	424,676	1032	13.29	7.48	0	163,283	1005	2.72	–	5	110	3
		1	4.25	–	5	57,907	20	4.25	–	5	256,231	448	0.01	–	5	4	0
0.2	100	5	8.34	3.68	0	705,626	1014	8.34	5.13	0	239,838	1006	0.17	–	5	401	4
		10	10.67	7.24	0	302,122	1018	10.67	8.83	0	111,759	1011	0.83	–	5	5,321	60
		1	4.05	1.36	0	2,013,387	1015	4.05	2.38	0	174,711	1006	0.00	–	5	0	0
	200	5	8.58	6.43	0	356,130	1016	8.58	7.28	0	122,386	1012	0.04	–	5	903	12
		10	9.77	8.27	0	140,988	1012	9.77	10.11	0	45,361	1008	0.29	0.06	1	41,833	891
Optimal					35.56%					25.56%					91.11%		

MJ2, where

$$
\begin{aligned}
&\max && \sum_{i=1}^{m}\sum_{j=1}^{n}(h_{ij}x_{ij} - k_{ij}t_{ij})\\
&(\mathrm{MJ2})\ \text{s.t.} && y_{ij}^{a_{ij}} \le t_{ij}^{b_{ij}} x_{ij}^{a_{ij}-b_{ij}}, \quad i = 1,\ldots,m,\ j = 1,\ldots,n,\\
& && (1), (2), (3), (4),
\end{aligned}
$$

as CMJ2. In the next section we compare these alternative conic reformulations computationally.

4. Computational experiments

In order to test the computational impact of the conic strengthening we have performed experiments with different formulations of the problem using quadratic and cubic objective functions: $f(y) = ky^2$ and $f(y) = ky^3$. All experiments are performed using ILOG CPLEX Version 11.0 with default settings on a 3.12 GHz Linux workstation with 1 GB memory with a 1000 CPU seconds time limit.

We performed experiments on data sets with varying number of jobs ($n = 50, 100, 200$), machines ($m = 1, 5, 10$), and capacity factors ($\kappa = 0.1, 0.2$). For each experimental configuration of n, m, κ, we generated five instances with h_{ij} from Uniform [2.0, 6.0], k_{ij} from Uniform [1.0, 3.0], p_{ij} from Uniform [1.0, 3.0], $\ell_{ij} = 0$, and u_{ij} from $p_{ij}\times$ Uniform [0.2, 0.8]. All machines have capacity equal to

$$c = \kappa/m \times \sum_{j=1}^{n}\sum_{i=1}^{m} p_{ij}/m,$$

so that the capacity factor κ controls the total machining capacity mc, independent of the number of machines.

We compare three formulations for the quadratic case $f(y) = ky^2$. The first formulation is MJ0, which is a mixed 0–1 program with quadratic objective, solved by CPLEX MIQP solver. The second one is CMJ1, which is a quadratically constrained quadratic MIP (it is equal to MJ1 for the quadratic case) with constraints $y^2 \le t$ for each machine-job pair. Finally, the third one is CMJ2, the conic reformulation based on the strengthened inequality (7) $y^2 \le tx$, which is already hyperbolic for the quadratic case.

We summarize the results of this experiment in Table 1. For each formulation we report the averages for the percentage gap between the continuous relaxation at the root node and best feasible solution known (rgap), the number of branch-and-bound nodes explored (nodes), and the total cpu seconds (cpu). We also report the number of instances that could be solved to optimality within the time limit (opt) and if there are instances that could not be solved, for them, we report the average percentage gap between the best known upper bound and lower bound at termination (egap). The continuous relaxations at the root node were solved within a fraction of second for all instances; therefore, we do not report them in the tables.

Table 2
Alternative conic formulations for the cubic case: $f(y) = ky^3$.

	CMJ1	CMJ2′	CMJ2
Hyperbolic inequalities	$y^2 \le v_1$ $v_1^2 \le ty$	$y^2 \le v_1v_2$ $v_1^2 \le ty$ $v_2^2 \le x$	$y^2 \le v_1x$ $v_1^2 \le ty$

Note that the integrality gap at the root node is the same for MJ0 and CMJ1 and it takes longer time to solve CMJ1 than MJ0. Whereas, most of the instances could not be solved to optimality with either MJ0 or CMJ1 within the time limit, all but eight of the 90 instances could be solved to optimality using the strong conic formulation CMJ2. For those eight unsolved instances with CMJ2 the average optimality gap at termination is only 0.2%.

Because the continuous relaxation of MJ0 is a QP, it is solved faster than the quadratically constrained QP relaxation of CMJ1. Thus, a conic reformulation is not helpful when its relaxation has the same bound as for the QP. On the other hand, with conic formulation CMJ2, the integrality gap at the root node is reduced to only 1.20%, which in turn leads to a much smaller branch-and-bound tree. Even though the continuous conic relaxations take longer to solve than QPs, it pays off when solving the integer problem due to the bound strengthening.

In Table 1 we also observe that with tighter machining capacity, the integrality gap is higher for all problems sizes. Moreover, the integrality gap increases with the number of machines, but decreases with the number of jobs. Nevertheless, instances become harder to solve for all formulations as the size increases. However, whereas only the smallest instances can be solved with MJ0, conic reformulation CMJ2 scales well with size.

The next experiment is on the cubic case $f(y) = ky^3$. Inequalities (6) and (7), used in CMJ1 and CMJ2 for this case, are

Table 3
Computational results for the cubic case: $f(y) = ky^3$.

κ	n	m	CMJ1					CMJ2′					CMJ2				
			rgap	egap	opt	nodes	cpu	rgap	egap	opt	nodes	cpu	rgap	egap	opt	nodes	cpu
0.1	50	1	11.36	–	5	1,760	3	4.38	–	5	518	2	0.10	–	5	8	0
		5	17.10	–	5	119,490	457	10.13	0.57	4	47,076	324	3.74	–	5	904	3
		10	26.28	5.04	0	213,562	1021	17.76	4.28	0	86,957	1015	7.14	–	5	1,978	18
	100	1	8.77	–	5	35,302	114	3.39	–	5	5,556	33	0.06	–	5	13	1
		5	13.05	6.64	0	156,366	1015	6.41	2.82	0	98,823	1012	0.70	–	5	3,510	19
		10	17.24	13.26	0	68,649	1016	9.78	7.24	0	63,352	1014	1.71	0.94	3	52,681	431
	200	1	8.89	4.12	0	181,625	1010	3.31	1.04	0	88,516	1012	0.00	–	5	0	0
		5	12.81	10.34	0	77,939	1018	6.14	5.52	0	18,542	1057	0.18	–	5	12,612	121
		10	15.17	15.12	0	31,901	1021	8.12	8.66	0	15,506	1014	0.70	0.43	1	34,551	1031
0.2	50	1	7.42	–	5	9,567	26	2.47	–	5	915	3	0.07	–	5	11	0
		5	12.97	3.93	0	199,224	1006	6.38	1.02	1	91,371	971	0.82	–	5	437	4
		10	17.25	11.42	0	82,549	1013	10.18	6.58	0	30,532	1008	2.72	–	5	998	15
	100	1	6.49	2.30	0	117,997	1005	2.12	0.07	4	50,304	351	0.01	–	5	2	0
		5	11.64	8.24	0	69,291	1011	5.08	3.20	0	33,804	1007	0.13	–	5	1,337	11
		10	14.41	13.39	0	41,056	1011	7.44	7.31	0	13,800	1010	0.65	–	5	22,352	287
	200	1	6.17	4.68	0	30,758	1005	2.01	1.45	0	22,191	1041	0.00	–	5	0	0
		5	12.10	11.45	0	20,881	1011	5.34	4.92	0	11,905	1008	0.04	–	5	1,434	43
		10	13.51	14.99	0	8,910	1012	6.70	7.76	0	3,566	1011	0.17	0.06	0	25,953	1027
	Optimal				22.22%					26.67%					87.78%		

$y^3 \le t$ and $y^3 \le tx^2$. In addition, in order to see whether only a partial strengthening would be effective, we also compared CMJ1 and CMJ2 with a conic formulation with the simpler inequality $y^3 \le tx$. We refer to this partially strengthened formulation as CMJ2′. In Table 2 we present the corresponding hyperbolic constraints for these three formulations.

We summarize the results with the cubic objective in Table 3. The first observation is that the integrality gaps are larger for the cubic case than for the quadratic case and, consequently, the cubic problems are overall more difficult to solve than the quadratic problems. Out of 90 instances only 20 could be solved to optimality with formulation CMJ1. Even though the partially strengthened formulation CMJ2′ resulted some improvement due to smaller integrality gap, most of the instances still could not be solved with it. On the other hand, all but 11 instances were solved to optimality with the strong formulation CMJ2. For those unsolved instances with CMJ2 the average optimality gap at termination is only 0.35%.

These experiments demonstrate clearly the effectiveness of the conic strengthening introduced here for solving the machine-job assignment problem with controllable times.

Acknowledgements

This research was conducted while S. Gürel was a visiting scholar at the University of California-Berkeley with support from the Scientific and Technological Research Council of Turkey under Grant 2214. A. Atamtürk was supported, in part, by the US National Science Foundation Grant 0700203.

References

[1] F. Alizadeh, D. Goldfarb, Second-order cone programming, Mathematical Programming 95 (2003) 3–51.
[2] A. Ben-Tal, A. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, in: MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.
[3] H.D. Sherali, W.P. Adams, A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Kluwer Academic Publishers, Dordrecht, Boston, London, 1999.
[4] M. Tawarmalani, N.V. Sahinidis, Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Springer, New York, 2003.
[5] M.F. Anjos, H. Wolkowicz, Strengthened semidefinite relaxations via a second lifting for the max-cut problem, Discrete Applied Mathematics 119 (1–2) (2002) 79–106.
[6] M.X. Goemans, Semidefinite programming in combinatorial optimization, Mathematical Programming 79 (1997) 143–161.
[7] C. Helmberg, F. Rendl, Solving quadratic (0, 1)-problems by semidefinite programs and cutting planes, Mathematical Programming 82 (3) (1998) 291–315.
[8] S. Poljak, F. Rendl, H. Wolkowicz, A recipe for semidefinite relaxation for (0, 1)-quadratic programming, Journal of Global Optimization 7 (1) (1995) 51–73.
[9] M.T. Çezik, G. Iyengar, Cuts for mixed 0-1 conic programming, Mathematical Programming 104 (2005) 179–202.
[10] A. Atamtürk, V. Narayanan, Conic mixed-integer rounding cuts, Research Report BCOL.06.03, University of California-Berkeley, December 2006, Mathematical Programming, forthcoming (doi:10.1007/s10107-008-0239-4).
[11] A. Atamtürk, V. Narayanan, Lifting for conic mixed-integer programming, Research Report BCOL.07.04, IEOR, University of California-Berkeley, October 2007, Mathematical Programming, forthcoming (doi:10.1007/s10107-008-0239-4).
[12] A. Frangioni, F. Gentile, Perspective cuts for a class of convex 0-1 mixed integer programs, Mathematical Programming 106 (2006) 225–236.
[13] O. Günlük, J. Lee, R. Weismantel, MINLP strengthening for separable convex qaudratic transportation-cost UFL, IBM Research Report RC24213, IBM, TJ Watson Research Center, York Town, NY, 2007.
[14] S. Gürel, M.S. Aktürk, Considering manufacturing cost and scheduling performance on a CNC turning machine, European Journal of Operational Research 177 (2007) 325–343.
[15] R.G. Vickson, Two single-machine sequencing problems involving controllable job processing times, AIEE Transactions 12 (1980) 258–262.
[16] M.A. Trick, Scheduling multiple variable-speed machines, Operations Research 42 (1994) 234–248.
[17] D. Shabtay, G. Steiner, A survey of scheduling with controllable processing times, Discrete Applied Mathematics 155 (2007) 1643–1666.
[18] M. Savelsbergh, A branch-and-price algorithm for the generalized assignment problem, Operations Research 45 (1997) 831–841.