Università di Pisa
Dipartimento di Informatica

Technical Report: TR-08-16

A Computational Comparison of Reformulations of the Perspective Relaxation: SOCP vs. Cutting Planes

A. Frangioni
Dipartimento di Informatica, Università di Pisa
Polo Universitario della Spezia, Via dei Colli 90, 19121 La Spezia, Italy
frangio@di.unipi.it

C. Gentile
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R.
Viale Manzoni 30, 00185 Roma, Italy
gentile@iasi.cnr.it

June 25, 2008

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

A Computational Comparison of Reformulations of the Perspective Relaxation: SOCP vs. Cutting Planes

A. Frangioni
Dipartimento di Informatica, Università di Pisa
Polo Universitario della Spezia, Via dei Colli 90, 19121 La Spezia, Italy
frangio@di.unipi.it

C. Gentile
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R.
Viale Manzoni 30, 00185 Roma, Italy
gentile@iasi.cnr.it

June 25, 2008

Abstract

The *Perspective Relaxation* is a general approach for constructing tight approximations to MINLP problems with semicontinuous variables. Two different reformulations have been proposed for solving it, one resulting in a Second-Order Cone Program, the other based on representing the perspective function by (an infinite number of) cutting planes. We compare the two reformulations on two sets of test problems to determine which one is most effective in the context of exact or approximate Branch-and-Cut algorithms.

Keywords: *Mixed-Integer Non Linear Programs, Reformulations, Second-Order Cone Programs, Valid Inequalities, Unit Commitment problem, Portfolio Optimization*

1 Introduction

Semi-continuous variables are very often found in models of real-world problems such as Distribution and Production Planning problems [12, 14, 6, 9, 8], Financial Trading and Planning problems [1, 7, 11], and many others [5, 2, 10]. These are continuous variables $p \in \mathbb{R}^m$ which are constrained either to assume the value 0, or to lie in some given compact set $\mathcal{P}$; in our applications it will always be a polyhedron. Often, $0 \notin \mathcal{P}$, i.e., the feasible set of p is disconnected; this is e.g. the case when p represents the output of a production process that can either be "inactive", and therefore nothing is produced, or "active", and therefore the output of the process must "lie between some minimum and maximum amount". Alternatively, 0 may belong to $\mathcal{P}$, but one may be forced to pay a fixed cost c (to "activate" the process) whenever $p \neq 0$.

We will consider mathematical programs where n semi-continuous variables are present; assuming that, for each $i \in N = \{1, \ldots, n\}$, the set $\mathcal{P}_i = \{p_i \in \mathbb{R}^{m_i} : A_i p_i \leq b_i\}$ has the property that $\{p_i \in \mathbb{R}^{m_i} : A_i p_i \leq 0\} = \{0\}$, each semi-continuous variable is typically modeled by using an associated binary variable u_i. In particular, we will consider Mixed-Integer NonLinear Programs (MINLP) of the form

$$\min \quad g(z) + \textstyle\sum_{i \in N} f_i(p_i) + c_i u_i \tag{1}$$

$$A_i p_i \leq b_i u_i \qquad i \in N \tag{2}$$

$$(p, u, z) \in \mathcal{O} \tag{3}$$

$$u \in \{0,1\}^n \ , \ p \in \mathbb{R}^m \ , \quad z \in \mathbb{R}^q \tag{4}$$

where all f_i and g are closed convex functions, z is the vector of all the "other" variables, and $\mathcal{O}$ is any subset of $\mathbb{R}^{m+n+q}$ (with $m = \sum_{i \in N} m_i$), representing all the "other" constraints of the problem.

It is known that the convex hull of the (disconnected) domain $\{0\} \cup \mathcal{P}_i$ of each p_i can be conveniently represented in a higher-dimensional space, which allows to derive *disjunctive cuts* for the problem [13]; this leads to defining the *Perspective Relaxation* of (1)—(4) [4, 6]

$$\min \quad g(z) + \textstyle\sum_{i \in N} u_i f_i(p_i/u_i) + c_i u_i \tag{5}$$

$$(2) \quad , \quad (3)$$

$$u \in [0,1]^n \quad , \quad p \in \mathbb{R}^m \quad , \quad z \in \mathbb{R}^q \tag{6}$$

which is significantly stronger than the ordinary continuous relaxation, and therefore a more convenient starting point to develop exact and approximate solution algorithms [6, 7, 2, 8, 10]. An interesting feature of (5)—(6) is that it is in fact a *reformulation* of (1)—(4), as the two coincide if $u \in \{0,1\}^n$ (while (5) is formally undefined when some $u_i = 0$, one can extend it by continuity to allow for null values).

However, an issue with (5)—(6) is the high nonlinearity in the objective function due to the added fractional term. A specialized Interior Point (IP) method has been proposed [4], but no efficient implementation seem to exist; besides, within enumerative algorithms IP approaches are often bested by active-set ones due to the latters' better reoptimization capabilities. Two alternative reformulations of (5)—(6) have been proposed; one uses Second-Order Conic inequalities [2, 10], and thus allows to exploit the available efficient IP solvers for Second-Order Cone Programs (SOCP). Another reformulation is based on (an infinite family of) linear inequalities, called Perspective Cuts [6], which define the nonlinear feasible region. In this paper we compare the two reformulation, from a computational standpoint, in the context of exact or approximate Branch-and-Cut algorithms; in Section 2 we recall them, while in Section 3 we report the computational comparison on two different applications, the Unit Commitment problem (§ 3.1) and the Mean-Variance problem (§ 3.2), respectively.

2 The solution methods

2.1 Direct SOCP reformulation

It is well-known that many convex functions, and the perspective function amongst them, can be can be represented by means of *conic inequalities* of the form $\sqrt{\sum_{i=1}^{k-1} x_i^2} \leq x_k$, where $x \in \mathbb{R}^k_+$ [3]; it is therefore not surprising that (5)—(6) is *SOCP-representable*. In fact, one can prove that when $f(p) = p^{r/s}$, $r, s \in \mathbb{Z}_+$,

$$\begin{aligned} conv\big(\ \{\ (u,p,v) \in \mathbb{R}^3\ :\ v \geq f(p)\ ,\ u \in \{0,1\}\ \}\ \big) = \\ \{\ (u,p,v) \in \mathbb{R}^3\ :\ v \geq u f(p/u)\ ,\ u \in [0,1]\ \} \end{aligned}$$

and the convex hull is representable by conic quadratic constraints [2]. The result can be extended to sums of convex quadratic functions, showing that (5)—(6) defines the best convex relaxation of (1)—(4) in the quadratic unconstrained case ($\mathcal{O} = \mathbb{R}^{m+n+q}$) [10]. The reformulation of (5)—(6) as a SOCP is actually quite simple in the quadratic case, as when $u > 0$ a constraint $t \geq ap^2/u$ can be algebraically transformed into the equivalent $(t+u)^2/4 \geq ap^2 + (t-u)^2/4$, which can then be easily written in conic form. Thus, when $f_i(p_i) = a_i p_i^2 + b_i p_i$, (1)—(4) can be reformulated as the Mixed-Integer SOCP

$$\begin{aligned} \min \quad & g(z) + \textstyle\sum_{i \in N} t_i + b_i p_i + c_i u_i \\ & \sqrt{a_i p_i^2 + (t_i - u_i)^2/4} \leq (t_i + u_i)/2 \quad i \in N \\ & (2)\ ,\ (3)\ ,\ (4)\ ,\ t \in \mathbb{R}^n_+ \end{aligned}$$

which can be approached with solvers such as `Cplex`; this can be more efficient than attacking (1)—(4) directly [2, 10]. We call the above the Conic Program (CP) reformulation.

2.2 Perspective Cuts

An alternative reformulation has been proposed in [6], based on the fact that the *epigraph* of $uf(p/u)+cu$ on $conv(\ \{0\} \cup \mathcal{P}\ )$ can be represented by the following (infinite) family of linear inequalities, called *perspective cuts*,

$$v \geq sp + (c + f(\bar{p}) - s\bar{p})u \tag{7}$$

indexed over all the (uncountably many) $\bar{p} \in \mathcal{P}$ and $s \in \partial f(\bar{p})$, where $\partial f(\bar{p})$ denotes the subdifferential of f at $\bar{p}$. When f is quadratic, this leads to the following Semi-Infinite MINLP

$$\begin{array}{ll} \min & g(z) + \sum_{i \in N} v_i \\ & v_i \geq (2a_i\bar{p}_i + b_i)p_i + (c_i - a_i\bar{p}_i^2)u_i \qquad \bar{p}_i \in \mathcal{P}_i\ ,\ i \in N \\ & (2)\ \ ,\ \ (3)\ \ ,\ \ (4)\ \ ,\ \ v \in \mathbb{R}^n \end{array}$$

which we call the Perspective Cut (P/C) reformulation of (1)—(4). While the above formulation cannot be solved directly, it lends itself nicely to approximation techniques whereby a (small) finite subset of the inequalities (7) are kept, and new ones are generated as needed when they are violated by the current solution (p^*, u^*, v^*) of the relaxation. This can easily be implemented by using the standard tools made available by off-the-shelf solvers such as `Cplex`; again, this is usually more efficient than approaching (1)—(4) directly [6, 7, 8].

2.3 Implementation details

A nice feature of both reformulations is that, when g and $\mathcal{O}$ are "simple" (e.g., linear), they can be implemented within widely available solution codes. In particular, for our experiments we have used `Cplex` version 11. It allows to directly input the CP reformulation as a Mixed-Integer Quadratically Constrained Quadratic Program, and solve it with the whole sophisticated machinery (valid inequalities, branching rules, ...) implemented in that state-of-the-art commercial solver. As for the P/C reformulation, the dynamic generation of (7) can be easily implemented by means of the `cutcallback` procedure, retaining all the other (sophisticated) aspects of the system. Thus, apart from the basic formulation, the same (sophisticated) tools are used for both approaches. Some differences, of course, remain; while the CP formulation will require the use of the IP SOCP solver, the P/C one will allow the use of active-set solvers for Linear or Quadratic Programs. On the other hand, the need for invoking the callback functions disables the—allegedly—more efficient *dynamic search* of `Cplex 11` for P/C, whereas it is used when the CP formulation is employed. Apart from that, the very same machinery is used with both formulations, allowing a fair comparison.

The tests have been performed on an Opteron 246 (2 GHz) computer with 2 GigaBytes of RAM, running Linux Fedora Core 3. Unless otherwise stated, the default required gap for Mixed-Integer programs (0.01%) has been set; a maximum time limit of 24 hours (86400 seconds) of CPU time has been set.

3 Computational results

3.1 The Unit Commitment problem

The Unit Commitment (UC) problem in electrical power production requires to optimally operate a set I of thermal generating units and a set H of hydro generating units to satisfy a given total power demand on each of a set T of discretized time instants, covering some time horizon (e.g., hours or half-hours in a day or a week). Each unit $i \in I$ is characterized by a minimum and maximum power output $0 < p_i^{min} < p_i^{max}$, when the unit is operational, and by a convex quadratic power (fuel) cost function $f_i(p) = a_i p^2 + b_i p + c_i$. Thus, introducing binary variables u_{it}, indicating the commitment of unit i at time instant t, and continuous variables p_{it} indicating the corresponding power output, the problem contains $|T| \times |I|$ blocks with the required structure

$$\min\ \{\ a_i p_{it}^2 + b_i p_{it} + c_i u_{it}\ :\ p_i^{min} u_{it} \leq p_{it} \leq p_i^{max} u_{it}\ ,\ u_{it} \in \{0,1\}\ \}\ \ .$$

These blocks are then linked by a variety of other constraints; demand ones, relative to power demand satisfaction at each $t \in T$ by active units, and technical ones imposing operational requirements on the

units. These usually comprise *minimum up- and down-time* constraints (whenever a unit is turned on it must remain committed at least a given number of consecutive time instants, and, analogously, whenever a unit is turned off it must remain decommitted for at least a given number of consecutive time instants), *ramp rate* constraints (the production level at each time instant cannot be too far from that of the preceding and following one), *spinning reserve* constraints, *network* constraints, constraints describing time-dependent *start-up costs*, and others. Also, hydro generating units produce electricity at basically no cost, but only have a limited allowance of water which has to be wisely allocated taking into account constraints on basin level, unit maximum production, and effect on downstream unit in the same hydro cascade. The exact details of these constraints are largely immaterial for the current discussion; the interested reader is referred e.g. to [9, 8] and the references therein. What is relevant here is that UC problems have a "rich" structure, besides that of nonlinear-cost semicontinuous variables.

We have compared the two reformulations on a test bed of randomly generated realistic instances already employed in several occasions [6, 9, 8], and freely available at

`http://www.di.unipi.it/optimize/Data` .

In the following tables, column "p" reports the number of thermal generating units (hence $n = 24p$, as the time discretization is hourly on daily instances), while column "h" reports the number of hydro units; the first half of the table, with $h = 0$, is therefore composed by "pure thermal" instances, and each row reports averaged results of 5 instances of the same size. Then, for each formulation columns "nds" and "time" report the number of nodes in the B&C tree and the total running time (in seconds) required by each approach, while column "gap" reports, only for those cases where not all the instances could be solved to optimality within the allotted time limit, the attained gap (in percentage) at termination. For P/C, column "LPs" and "t/LP" report respectively the total number of Linear Programs solved, and the average time required for solving one of them; analogously, for CP column "CPs" and "t/CP" report respectively the total number of SOCPs solved, and the average time required for solving one of them.

Table 1 reports the results with standard optimality tolerance 0.01%; these are limited to smaller-size instances, as *none* the approaches could solve *any* of the largest-size ones within the 24 hours time limit.

Table 1: Results for UC with optimality tolerance 0.01%

		P/C					CP				
p	h	gap	nds	LPs	time	t/LP	gap	nds	CPs	time	t/CP
10	0		4.3e2	7.8e2	14	0.018		5.8e2	1.0e3	20	0.021
20	0		5.0e4	5.8e4	6805	0.094		6.6e4	7.5e4	13392	0.145
50	0	0.08	1.7e5	2.1e5	86400	0.421	0.08	9.1e4	1.1e5	86400	0.781
20	10		1.1e4	1.3e4	161	0.014		1.4e4	1.8e4	626	1.937
50	20		5.5e5	6.6e5	29874	0.037	0.00	5.0e5	6.1e5	86400	0.460
75	35	0.01	8.5e5	1.0e6	73076	0.073	0.01	1.8e5	2.2e5	86400	0.314

This table shows a distinct advantage of P/C over CP. When both approaches solve the instances, P/C requires less nodes; since the lower bound is the same (actually, being an approximated approach P/C can only be slightly worse in this respect), this is likely to be mostly due to the fact that the more "central" solutions generated by IP methods are less effective to drive primal rounding heuristics than the "extreme" solutions generated by the simplex method. Coupled with the fact that LPs require less time than CPs to be solved (cf. columns t/LP and t/CP), P/C ends up requiring less time to solve the instances. When the time limit kicks in, P/C is thus able to enumerate more nodes in the same time, thereby often being able to solve more instances. This is especially true for hydro-thermal instances, where LPs can be up to two orders of magnitude faster to solve than CPs; this is not surprising, as these instances have a much larger number of constraints and continuous variables (those devoted to modeling the hydro subsystem) and therefore the continuous relaxations are much larger. It has to be noted, however, that the difference lessens as the size of the instances grow, perhaps reflecting the better asymptotic complexity of IP methods.

In order to be able to test the approach on larger instances we also experimented with the much coarser optimality tolerance of 0.5%; this is the advised value for quickly obtaining approximated solutions when the operational environment requires fast response times [9, 8]. The corresponding results are reported in Table 2.

Table 2: Results for UC with optimality tolerance 0.5%

		P/C					CP				
p	h	gap	nds	LPs	time	t/LP	gap	nds	CPs	time	t/CP
10	0	0.09	0	30	0.67	0.023	0.06	0	70	1.91	0.028
20	0	0.06	0	34	2.81	0.085	0.09	0	65	6.78	0.106
50	0	0.18	0	39	15.45	0.411	0.19	0	91	37.91	0.421
75	0	0.22	0	30	23.28	0.785	0.23	0	71	63.27	0.933
100	0	0.15	0	29	34.16	1.182	0.19	0	64	100.28	1.578
150	0	0.10	0	75	90.13	1.410	0.11	0	106	233.46	2.256
200	0	0.09	0	57	126.28	2.313	0.11	0	104	386.36	3.860
20	10	0.11	0	83	2.77	0.034	0.24	20	194	10.45	2.372
50	20	0.04	0	79	6.53	0.102	0.35	1	115	20.55	0.575
75	35	0.09	0	61	10.60	0.182	0.08	15	202	64.50	0.319
100	50	0.04	0	81	20.17	0.267	0.08	10	193	97.03	0.421
150	75	0.06	100	417	247.73	0.596	0.04	15	331	368.92	0.778
200	100	0.04	30	222	247.22	1.111	0.03	5	165	385.03	1.563

This table confirms the previous analysis. All the pure thermal instances are solved at the root node by both reformulations; despite the fact that P/C inherently requires repeated LP solutions due to the iterative nature of the approach, CP ends up actually requiring more relaxation solutions that P/C. Coupled with the fact that LPs require less time than CPs (not dramatically, but still visibly so), P/C is about a factor of three faster than CP; besides, the quality of the obtained solution is most often (slightly) better. Results for hydro-thermal instances are somewhat different because the two reformulations require a different amount of enumeration; however, LPs are significantly less expensive than CPs to solve (although less and less so as the size grows), so that P/C ends up being more efficient than CP, sometimes quite significantly so in both running time and solution quality.

3.2 Markowitz Mean-Variance model

A set of n risky assets are available; for each asset i, (an estimate of) the expected unitary return μ_i for the considered time horizon is known, and *minimum and maximum buy-in thresholds* $0 < p_i^{min} < p_i^{max}$ are set on the purchasable quantity (if the asset is purchased at all). The Mean-Variance (MV) model with minimum buy-in thresholds in portfolio optimization

$$\begin{array}{lll} \min & p^T Q p & \\ & ep = 1 \quad , \quad \mu p \geq \rho \quad , & u \in \{0,1\}^n \\ & u_i p_i^{min} \leq p_i \leq u_i p_i^{max} & i \in N \end{array} \tag{8}$$

where $Q \succeq 0$ is (an estimate of) the $n \times n$ variance-covariance matrix and e is the all-ones vector, requires to select a minimum-risk (as measured by variance) portfolio which produces a desired level of return ρ. Unlike UC, this Mixed-Integer Quadratic Program has a very "simple" structure; it does not even directly qualify for Perspective Relaxations, as the cost function is nonseparable in the semi-continuous variables. This can be dodged with a reformulation trick first proposed in [6]; compute a diagonal matrix $D \succeq 0$ such that $R = Q - D \succeq 0$, change the objective function to $p^T D p + z^T R z$, and add the additional constraint $z = p$. An effective way for computing a "large" D is by solving a single SemiDefinite Program; this allows to apply the Perspective Relaxation to (8), which turns out to be more efficient than working with the original quadratic formulation [7].

We have compared P/C and CP on 90 randomly generated MV instances, described in [7] and freely available at the previously mentioned web address. The instances are characterized by the value of $n \in \{200, 300, 400\}$, and by the *dominance index* of Q, measuring how much the matrix is diagonally dominant (as this turned out to have a significant impact on the effectiveness of the Perspective Relaxation [7]); the "$^+$" instances have strongly diagonally dominant Q, the "^{0}" instances have weakly diagonally dominant Q, and the "$^-$" instances have strongly *not* diagonally dominant Q. For each combination 10 instances are

generated. The results are shown in Table 3; the meaning of the data is the same as for the UC (with the obvious change due to the fact that QPs are solved within P/C).

Table 3: Results for MV

	P/C				CP				
n	nds	QPs	time	t/QP	nds	CPs	time	t/CP	gap
200^+	1.9e4	1.9e4	194	0.0008	9.2e3	1.1e3	17961	1.578	0.15(1)
200^0	1.7e4	1.8e4	90	0.0007	2.7e4	3.2e4	30785	1.648	0.32(2)
200^-	1.2e5	1.3e5	835	0.0006	1.6e4	1.9e5	55144	1.719	1.02(5)
300^+	3.4e4	3.5e4	433	0.0014	1.1e4	1.4e4	72075	8.334	0.58(7)
300^0	3.1e5	3.3e4	378	0.0019	1.0e4	1.3e4	59591	4.464	0.53(6)
300^-	5.5e5	5.8e4	654	0.0014	1.1e4	1.3e4	66863	5.272	0.81(7)
400^+	7.9e4	8.2e4	2066	0.0032	4.7e3	5.9e3	61810	10.397	1.01(6)
400^0	2.3e5	2.4e5	3974	0.0020	6.1e3	7.6e3	83782	10.588	1.79(9)
400^-	3.3e5	3.4e5	8092	0.0026	6.3e3	7.9e3	80382	10.764	2.71(8)

Again, the results clearly favor P/C over CP. While the latter never solves to optimality all instances of any single group (the number in parenthesis in the "gap" column reports the number of unsolved instances), the former solves them all, requiring *at least* one, and up to more than two, orders of magnitude less time for doing so. This is clearly due to the fact that the very simple QPs solved by P/C typically require *three to four* order of magnitude less time to be solved than the CPs employed by CP; this allows P/C to enumerate up to two orders of magnitude more nodes within the allotted time limit, and therefore to solve even the largest instances, where CP cannot do better than an average gap larger than 1%.

3.3 Conclusion

The Perspective Relaxation is a useful reformulation tool for nonlinear programs with semicontinuous variables, but the high nonlinearity of its objective function does not allow to approach it directly with available efficient solution technology. Both the Conic Program reformulation and the Perspective Cut reformulation overcome this issue, allowing to exploit state-of-the-art off-the-shelf solvers in conjunction with the Perspective Relaxation. Currently, the P/C relaxation seems to be favored, at least in the two applications that we tested; this is mostly due to the much more efficient reoptimization capabilities of active-set solution algorithms with respect to Interior Point ones, with possibly a small contribution from the different continuous solutions generated ("central" vs. "extreme"). It should be noted that the CP reformulation may ultimately prove attractive, especially for large-scale programs with "rich" structure; besides, it may be inherently be better suited for applications where the "other" structures of the problem (g, $\mathcal{O}$) are nonlinear, say conic, since then an IP method might have to be used anyway, and in that case a compact formulation would likely be favored. However, for the current state of solution technology, and on original formulations with linear constraints, the apparently more awkward P/C reformulation seems to have an edge over the more compact an elegant CP one.

References

[1] S. Ahn, L.F. Escudero, and M. Guignard-Spielberg. On modeling robust policies for financial trading. In T.A. Ciriani and R.L. Leachman, editors, *Optimization in Industry 2*, pages 163–184. Wiley, Chichester, 1994.

[2] S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. Technical Report BCOL 07.01, Industrial Engineering and Operations Research, University of California, Berkeley, 2007.

[3] A. Ben-Tal. Conic and robust optimization. Lecture Notes.

[4] S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86:595–614, 1999.

[5] M.A. Duran and I.E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

[6] A. Frangioni and C. Gentile. Perspective Cuts for 0-1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225–236, 2006.

[7] A. Frangioni and C. Gentile. SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP. *Operations Research Letters*, 35(2):181 – 185, 2007.

[8] A. Frangioni, C. Gentile, and F. Lacalandra. Tighter Approximated MILP Formulations for Unit Commitment Problems. Research Report R. 660, Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti" del C.N.R., Roma, 2007.

[9] A. Frangioni, C. Gentile, and F. Lacalandra. Solving Unit Commitment Problems with General Ramp Contraints. *International Journal of Electrical Power and Energy Systems*, 30:316 – 326, 2008.

[10] O. Günlük and J. Linderoth. Perspective relaxation of minlps with indicator variables. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *Integer Programming and Combinatorial Optimization, 13th International Conference, IPCO 2008 Bertinoro, Italy, May 26-28, 2008 Proceedings*, volume 5035 of *Lecture Notes in Computer Science*, pages 1–16, 2008.

[11] N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. In *Quantitative Finance*, volume 1, pages 1–13. Wiley, Chichester, 2001.

[12] J. Kallrath and J.M. Wilson. *Business Optimization*. Macmillan Press Ltd., Houndmills, 1997.

[13] R.A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

[14] J.M. Zamora and I.E. Grossmann. A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. *Comput & Chem. Engin.*, 22:367–384, 1998.