

SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP

Antonio Frangioni*

Dipartimento di Informatica, Università di Pisa

Claudio Gentile

Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti”, C.N.R.

March 28, 2019

Abstract

We present a new approach, requiring the solution of a SemiDefinite Program, for decomposing the Hessian of a nonseparable Mixed-Integer Quadratic problem to permit using perspective cuts to improve its continuous relaxation bound. The new method favorably compares with a previously proposed one requiring a minimum eigenvalue computation.

Keywords: *Mixed-Integer Quadratic Programs, Valid Inequalities, SemiDefinite Programming, Portfolio Optimization*

*Corresponding author: Largo B. Pontecorvo 3, 56127 Pisa – Italy, e-mail frangio@di.unipi.it

1 Introduction

We consider Mixed-Integer Quadratic Programs (MIQP) of the form

$$\begin{aligned} \min \quad & x^T Q x + q x + c y \\ & A x + B y \geq b \\ & l_i y_i \leq x_i \leq u_i y_i \quad , \quad y_i \in \{0, 1\} \quad i = 1, \dots, n \end{aligned} \tag{1}$$

where Q is positive semidefinite (denoted by $Q \succeq 0$); that is, each x_i is a *semi-continuous* variable whose domain is the (disconnected) set $0 \cup [l_i, u_i]$, and/or one pays a *fixed-charge* cost c_i whenever $x_i \neq 0$. Actually, these ideas can be extended to more general cases where several x variables depend on the same y and the linking constraints are different, but for the sake of clarity of the present discussion the above simple form is more appropriate.

When Q is diagonal, the continuous relaxation of (1) can be strengthened by using *perspective cuts* [3], a simple family of disjunctive cuts which operate on the objective function. These are faces of the epigraph of the *convex envelope* of the objective function on the set $\{l_i y_i \leq x_i \leq u_i y_i \quad , \quad y_i \in \{0, 1\}\}$, which can be separated at the feasible (fractional) point (x_i^*, y_i^*) by the simple closed formula

$$v_i \geq (2Q_{ii}\bar{x}_i + q_i)x_i + (c_i - Q_{ii}\bar{x}_i^2)y_i \tag{2}$$

where $\bar{x}_i = x_i^*/y_i^*$ and v_i is the variable representing the objective function value. Despite the low dimensionality of the faces that they represent, perspective cuts (2) significantly improve the efficiency of enumerative approaches to (MIQP)s with structure (1).

Using perspective cuts crucially requires the objective function to be separable among the blocks; however, a simple reformulation technique was proposed in [3] for nonseparable problems which moves “the nonseparable part” of the objective function to newly introduced variables, leaving a separable objective function to which (2) can then be applied. In this paper, we compare two different ways for decomposing the objective function: computing the minimum eigenvalue of Q , as done in [3], and a more costly procedure which requires the solution of a SemiDefinite Program. We show that, at least on instances of the Mean-Variance problem in portfolio optimization, the new procedure significantly improves the quality of the obtained lower bounds and the overall solution time.

2 The reformulation technique

In order to apply perspective cuts to (1) with a non-diagonal Q , one can select any non-negative diagonal $D \in \mathbb{R}^{n \times n}$ such that $Q - D \succeq 0$ and consider the following equivalent formulation:

$$\begin{aligned} \min \quad & x^T D x + z^T (Q - D) z + q x + c y \\ & A x + B y \geq b, \quad z = x \\ & l_i y_i \leq x_i \leq u_i y_i, \quad y_i \in \{0, 1\} \quad i = 1, \dots, n \end{aligned} \quad (3)$$

Model (3) allows direct application of perspective cuts, and retains most of the structure of the original problem by introducing a copy of the x variables and assigning it all the non-separability in the objective function. Intuitively, the “larger” D —the “fraction” of the objective function that is reflected on the separable costs—is, the more perspective cuts could be expected to improve the lower bound; thus, procedures have to be devised for (efficiently) finding a “large” D .

In [3] the *Minimum Eigenvalue (ME) approach* was proposed: $D = \lambda_{\min} I$, λ_{\min} being the minimum eigenvalue of Q . This requires Q to be *strictly* positive definite, for otherwise $\lambda_{\min} = 0 \Rightarrow D = 0$.

However, assuming $\text{tr}(D)$ —the sum of the diagonal elements of D —to be a relevant metric, finding the “largest” possible D can be directly cast as the following dual pair of SemiDefinite Programs (SDP)

$$\begin{aligned} \max \quad & \left\{ \sum_{i=1}^n d_i : Q - \sum_{i=1}^n d_i (e_i e_i^T) \succeq 0, \quad d \geq 0 \right\} \\ \min \quad & \left\{ \text{tr}(QX) : \text{diag}(X) \geq e, \quad X \succeq 0 \right\} \end{aligned} \quad (4)$$

where e is the vector of all ones and e_i is the i -th vector of the canonical base of \mathbb{R}^n ; clearly, this *SDP approach* can produce a nonzero D even if $\lambda_{\min} = 0$. One might have chosen different cost coefficients for the d_i variables in order to reflect different relevance of having a large quadratic coefficient for each x_i in (3); in want of sensible rules for computing those weights, for our computational results we have used unitary costs. Problems (4) can be easily solved by any of the several available SDP codes. Furthermore, the constraint $d \geq 0$ may be redundant (it was such on all the instances of our test bed); eliminating it from the primal turns the dual problem into

$$\min \left\{ \text{tr}(QX) : \text{diag}(X) = e, \quad X \succeq 0 \right\} \quad (5)$$

which is usually faster to solve and can be tackled by some approaches, such as the spectral bundle method, which require constant trace of the variable.

For both SDP and ME, in order to avoid that $Q - D$ turns out not to be positive semidefinite due to numerical errors, we subtracted from D (hence, added to $Q - D$) a matrix εI for a suitably chosen “small” $\varepsilon > 0$.

3 Computational results

We have tested the influence of ME and SDP on the overall efficiency of a B&C approach using perspective cuts to nonseparable (MIQP); as in [3], we have applied it to instances of the Mean-Variance (MV) problem with minimum and maximum buy-in thresholds, which is

$$\min \left\{ x^T Q x \mid \begin{array}{l} ex = 1 \text{ , } \mu x \geq \rho \text{ ,} \\ l_i y_i \leq x_i \leq u_i y_i \text{ , } y_i \in \{0, 1\} \text{ } i = 1, \dots, n \end{array} \right\} \quad (6)$$

where μ_i , l_i and u_i are respectively the expected unitary return and the minimum and maximum buy-in thresholds for asset i , Q is the variance-covariance matrix and ρ is the desired level of return. Several alternative forms of portfolio selection problems have been proposed in the literature (e.g., [1] and many others), many of which are difficult due to some form of discrete decisions [5]. Our choice of model (6) is motivated by the fact that it is quite simple but rather demanding for general-purpose (MIQP) solvers; this is due to the fact that the root node gaps of the continuous relaxation are huge (cf. Table 2, column “Cplex/r.gap”) and that it has very few constraints with no structure, so that the classical polyhedral approaches to improve the lower bounds [2] are ineffective. However, its simplicity also means that simple MIP-rounding heuristics usually find good solutions if the lower bounds are reasonably tight. Thus, (6) is an ideal candidate for testing the effect of different ways for selecting D , as these only influence the lower bounds produced by perspective cuts.

For our tests, we have generated 30 (MV) instances for each value of $n \in \{200, 300, 400\}$: Q has been generated using the well-known random generator of [6], while ρ , l_i and u_i has been uniformly drawn at random from intervals $[0.002, 0.01]$, $[0.075, 0.125]$ and $[0.375, 0.425]$, respectively. For each value of n , we have generated three different set of matrices Q . The “+” set has been generated with the same parameters as in [3], which turn out to produce matrices that are strongly diagonally dominant: indeed, their average *dominance index* $S_i = (Q_{ii} - \sum_{j \neq i} |Q_{ij}|) / Q_{ii}$ is around 0.6. In order to evaluate the effect of diagonal dominance on the proposed procedure, we have also modified the parameters in order to produce a “0” set with average $S_i \approx 0$, and a “−” set with average $S_i \approx -0.5$. The data required for reproducing the instances is available upon request by the authors.

For our experiments we have used the B&C approach using perspective cuts described in [3]. Some aspects of the approach are nonstandard, due to the cuts being applied to the objective function, but this is immaterial for the issue at stake here, so we need not discuss them; the interested reader is referred to the original paper for further details. We also compared the approach with the general-purpose B&C algorithm of **Cplex 9.1**; for both codes, a global time limit of 10000 seconds was set. The experiments were performed on a PC with an Opteron 246 processor and 2Gb RAM, running Linux. All the codes were compiled with **gcc 4.0** and **-O3** optimizations. Our B&C code uses **Cplex 9.1** to solve the continuous relaxations at each node of the enumeration tree.

In Table 1 we report some data that describes the average results of the initialization phase alone. Columns “ d_{max} ”, “ d_{min} ” and “ d_{avg} ” report the ratio between, respectively, the maximum, minimum and average element of the diagonal of D obtained by SDP and λ_{min} used by ME. All other columns report the time (in seconds) required for computing D . For ME, we have used the **eig()** function of the open-source package **octave 2.1**; more efficient methods may exist, but, as it will be clear shortly, there is little point in improving upon this. For SDP, we have tested all open-source solvers listed at C. Helmberg’s SDP page [4] which could be compiled as stand-alone applications and linked to our B&C code; this restricted the choice to **CSDP 4.9**, **DSDP 5.6**, **SBmethod 1.1.3**, **SDPA 6.0** and **SDPLR 1.02**. For each approach, columns “ \geq ” and “ $=$ ” report the time for solving (4) and (5), respectively; **SBmethod** can only solve the latter problem.

	d_{max}	d_{min}	d_{avg}	ME	CSDP		DSDP		SDPA		SDPLR		SB
					\geq	$=$	\geq	$=$	\geq	$=$	\geq	$=$	$=$
200 ⁺	1.96	0.97	1.47	0.13	3.12	2.98	1.86	0.10	1.81	0.29	3.71	2.23	23.77
200 ⁰	1.93	0.90	1.41	0.13	3.03	2.99	1.87	0.10	1.68	0.29	3.72	2.79	16.39
200 ⁻	1.86	0.87	1.37	0.13	3.00	2.95	1.86	0.10	1.62	0.40	2.30	2.19	16.58
300 ⁺	1.97	0.97	1.47	0.23	10.54	9.84	4.92	0.26	5.33	0.73	13.20	5.02	69.13
300 ⁰	1.93	0.91	1.42	0.23	10.91	9.55	4.99	0.26	4.97	0.71	8.58	9.08	46.01
300 ⁻	1.69	0.89	1.29	0.23	10.91	9.62	5.10	0.26	5.11	0.72	5.67	5.53	41.82
400 ⁺	1.98	0.97	1.47	0.39	31.03	29.28	10.56	0.52	5.02	1.40	17.48	21.60	146.07
400 ⁰	1.93	0.93	1.43	0.39	37.24	31.27	10.86	0.52	11.46	1.37	21.80	11.93	94.62
400 ⁻	1.87	0.89	1.38	0.39	36.77	31.61	10.75	0.52	11.10	1.38	15.10	21.11	90.07

Table 1: Comparison of SDP and ME initializations

On these instances, SDP finds a D whose diagonal elements are on average significantly larger than λ_{min} , although the ratios uniformly decrease as Q becomes less and less diagonally dominant. For doing so, it takes signif-

icantly longer than ME if the standard problem (4) is solved, but only very marginally longer—at least if DSDP 5.6 is used—if (5) is solved instead; a reasonable approach would then be to attempt solving (5) first, and revert to (4) if the optimal solution turns out not to be nonnegative.

Besides, the running times of the initialization are hardly significant when one considers the overall B&C approach, as shown in Table 2. Columns “nodes” and “time” report respectively the total number of explored nodes and the total time (in seconds) for the B&C approach, while columns “r.gap”, “p.gap” and “d.gap” report respectively the root node gap, the gap of the best primal solution and that of the best lower bound attained at the end of the enumerative process (in percentage); a blank entry corresponds to a gap less than 0.01%—the optimality tolerance of the B&C. As in the previous table, results are averaged on all 10 instances of each class. We have avoided to report column “time” for **Cplex** since it never terminated before the time limit, as well as columns “p.gap” for our B&C approach, using either SDP or ME, because it almost never attained a primal solution farthest from 0.01% from optimality; the only exception are “400[−]” instances, upon which ME attained an average primal gap slightly larger than 0.1% due to a *single* instance terminating with a gap of 0.12%. Note that, to be on the safe side, the SDP time comprises that for solving (4) with SDPA 6.0—so the actual time could be (slightly) reduced by using (5) instead—while the ME time does not include the (anyway, negligible) initialization phase.

	SDP				ME				Cplex			
	time	nodes	d.gap	r.gap	time	nodes	d.gap	r.gap	nodes	p.gap	d.gap	r.gap
200 ⁺	164	1.2e+4		1.14	904	7.7e+4		6.48	1.9e+7	0.14	45.33	85.63
200 ⁰	161	1.1e+4		2.14	320	2.8e+4		6.10	8.5e+6	0.38	51.27	84.47
200 [−]	1902	1.3e+5		3.65	3306	2.6e+5	0.02	6.69	8.9e+6	0.24	42.09	78.88
300 ⁺	818	2.9e+4		4.54	2061	9.3e+4		5.62	4.0e+6	0.41	64.68	92.01
300 ⁰	856	2.7e+4		1.97	1715	7.1e+4		6.28	3.6e+6	0.43	59.91	87.87
300 [−]	1709	5.2e+4		2.68	2797	9.4e+4	0.05	7.04	3.0e+6	0.53	45.11	78.77
400 ⁺	2264	7.0e+4		4.79	4756	1.1e+5	0.10	6.15	1.9e+6	1.03	61.47	89.06
400 ⁰	4378	7.2e+4	0.10	2.29	7421	1.6e+5	0.16	6.53	1.5e+6	1.18	68.68	90.03
400 [−]	6311	1.0e+5	0.23	3.06	6901	1.4e+5	0.36	6.49	1.5e+6	1.60	65.88	88.47

Table 2: Results of the three B&C approaches

As already seen in [3], the standard continuous relaxation has huge root node gaps, usually in the 80-90% range, that are only reduced to the 40-70% range within 10000 seconds of the standard B&C; while the MIP rounding heuristics of **Cplex** obtain relatively good primal solutions, these problems are unsolvable to optimality with standard means. With ME diagonaliza-

tion, perspective cuts close the root node gap to a more manageable 5-8%; this allows to solve most of the smallest instances, but it is not enough for the largest ones. There also is a clear trend between dominance of Q and effectiveness of the perspective cuts approach; indeed, ME could not solve two “-” instances for both $n = 200$ and $n = 300$, while for $n = 400$ it could not solve 3, 5 and 6 instances of class “+”, “0” and “-”, respectively.

The trend is clearly present for SDP, too: while the “better” D allows to solve all smaller instances and all “400+” ones, 3 and 6 instances of class “400⁰” and “400-”, respectively, remained unsolved. However, the bound improvements due to the SDP diagonalization—testified by the further reduction of the root node gaps—significantly improve the overall performances of the B&C approach: SDP solves more problems, it is faster on those that are solved, and obtains better dual gaps on those that are not, than ME.

Our results clearly show that, despite the cost of solving (4), the SDP approach is significantly more efficient than the ME one; for all our instances, by solving (5) instead, the extra effectiveness of perspective cuts could also be obtained at basically no extra cost. An interesting issue that still remains open is the development of different weighting schemes for the d_i variables in (4) which further improve the quality of the obtained bounds.

Acknowledgements

We are grateful to an anonymous referee for several comments which have helped us to significantly improve the paper. The second author has been partly supported by the EU “Marie Curie” Research Training Network no. 504438 ADONET.

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] D. Bienstock. Computational Study of a Family of Mixed-Integer Quadratic Programming Problems. *Mathematical Programming*, 74(2):121–141, 1996.
- [3] A. Frangioni and C. Gentile. Perspective Cuts for 0-1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225–236, 2006.

- [4] C. Helmberg. <http://www-user.tu-chemnitz.de/~helmberg/semidef.html>.
- [5] 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.
- [6] P.M. Pardalos and G.P. Rodgers. Computing aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45:131–144, 1990.