Global Solution of Quadratic Problems by Interval Methods and Convex Reformulation

Sourour Elloumi, Amélie Lambert, Bertrand Neveu, Gilles Trombettoni

▶ **To cite this version:**

Sourour Elloumi, Amélie Lambert, Bertrand Neveu, Gilles Trombettoni. Global Solution of Quadratic Problems by Interval Methods and Convex Reformulation. HUGO 2022 - 15th Workshop on Global Optimization, Sep 2022, Szeged, Hungary. hal-03699921

HAL Id: hal-03699921
https://hal.science/hal-03699921

Submitted on 12 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Global Solution of Quadratic Programs by Interval Methods and Convex Reformulations*

S. Elloumi,[1,2] A. Lambert,[2] B. Neveu,[3] and G. Trombettoni[4]

[1]*UMA, ENSTA Paris, IPP, 91120 Palaiseau, France,* sourour.elloumi@ensta-paris.fr

[2]*Cédric-Cnam, 292, rue saint martin 75003 Paris* amelie.lambert@cnam.fr

[3]*LIGM, Ecole des Ponts ParisTech, Univ. Gustave Eiffel, CNRS, France* Bertrand.Neveu@enpc.fr

[4]*LIRMM, University of Montpellier, CNRS* Gilles.Trombettoni@lirmm.fr

Abstract Interval B&B solvers handle non-convex optimization problems in a rigorous way and deal with a wide variety of operators. However, these solvers are not dedicated to QPs and do not exploit quadratic convex relaxations. We present an interval B&B code that can efficiently solve QPs. At each node, we use a quadratic convex relaxation as strong as a SDP relaxation, and a bisection heuristic dedicated to QPs. Experiments show significant speedups on integer instances.

Keywords:
MINLP, QP, Interval methods, SDP, Experiments

1. Introduction

We consider the optimization problem (QP) of a quadratic function of n variables under quadratic inequalities:

$$(QP)\begin{cases} \min f(x) \equiv \langle Q_0, xx^T\rangle + c_0^T x & \\ \quad g_r(x) \equiv \langle Q_r, xx^T\rangle + c_r^T x \leq e_r & r \in \mathcal{R} \\ \quad \ell_i \leq x_i \leq u_i & i \in \mathcal{I} \quad (1) \\ \quad x_i \in \mathbb{N} & i \in \mathcal{J} \subseteq \mathcal{I} \quad (2) \end{cases}$$

*This research benefited from the support of the FMJH'Program Gaspard Monge for optimization and operations research and their interactions with data science.

where $\mathcal{R}$ is the set of inequality indices, $\mathcal{I}$ is the set of variable indices, and $\mathcal{J} \subseteq \mathcal{I}$ is the subset of integer variables. The quadratic forms f and g_r use symmetric real matrices Q_0 and Q_r and real vectors c_0 and c_r. $\langle A_1, A_2 \rangle$ denotes a dot product between two matrices A_1, A_2 of same dimensions. Each variable x_i lies in the interval $[\ell_i, u_i]$ where ℓ_i and u_i are real scalars.

The objective is to make enumeration methods based on interval arithmetic and quadratic convex reformulation methods cooperate in order to better solve, globally and rigorously, Problem (QP). Our work leads to a new quadratic solver called `QIBEX` that is a quadratic variant of `IbexOpt`, a constrained nonlinear optimization tool using rigorous interval algorithmic operators [1, 8]. Interval methods provide `IbexOpt` with two main advantages: first, the guarantee of the solution obtained despite rounding problems on floating numbers; second, the possibility of defining the constraints and the objective function based on a wide variety of mathematical operators. To reduce the domain of variables and improve the lower bound, it uses several *contraction* operators: `HC4` [2] that is the state of the art constraint propagation algorithm, `ACID(HC4)` [7] that performs specific work on a few adaptively chosen variables, or X-Newton [1] that is an interval contractor based on a specific X-Taylor polyhedral relaxation.

Our main contribution is an interval B&B algorithm that can solve globally, efficiently and rigorously (QP). At each node, the hybrid solver `QIBEX` uses a quadratic convex relaxation that is calculated thanks to SDP (Section 2), together with a bisection heuristic dedicated to quadratic optimization. The interval features can then propagate efficiently this information for contracting all variable domains (Section 3). Our experiments show significant speedups on integer quadratic instances (Section 4).

2. Quadratic Convex Reformulation

Quadratic convex reformulation first introduces a new variable X_{ij} that models the product of variables x_i and x_j, for each pair $(i,j) \in \mathcal{I}^2$. Then, it builds the following program, parameterized by the positive semidefinite matrices S_0 and $S_r \; \forall \; r \in \mathcal{R}$:

$$(PC) \begin{cases} \min F(x,X) \equiv \langle S_0, xx^T \rangle + \langle Q_0 - S_0, X \rangle + c_0^T x & \\ \quad (2) & \\ \quad G_r(x) \equiv \langle S_r, xx^T \rangle + \langle Q_r - S_r, X \rangle + c_r^T x \leq e_r & r \in \mathcal{R} \\ \quad X_{ij} \geq u_j x_i + u_i x_j - u_j u_i, \;\; X_{ij} \geq \ell_j x_i + \ell_i x_j - \ell_j \ell_i & i,j \in \mathcal{I}^2 \quad (3) \\ \quad X_{ij} \leq u_j x_i + \ell_i x_j - u_j \ell_i, \;\; X_{ij} \leq \ell_j x_i + u_i x_j - \ell_j u_i & i,j \in \mathcal{I}^2 \quad (4) \\ \quad X_{ij} = x_i x_j & i,j \in \mathcal{I}^2 \quad (5) \end{cases}$$

Thanks to equalities (5), problems (QP) and (PC) are equivalent for any parameters S_0, S_r ($r \in \mathcal{R}$). Moreover, since the latter matrices are positive semidefinite, the quadratic forms F and G_r are convex, and the only non-convexities in (PC) come from Constraints (2) and (5). By dropping these constraints, we obtain Problem $(\overline{PC})$ a convex QCQP relaxation to (PC) and thus to (QP). Now, an important issue is the choice of matrices S_0 and S_r. The criterion adopted in [3] is to choose the matrices such that relaxation $(\overline{PC})$ is as tight as possible. It was proved in [5] that the best choice is to set S_r equal to the null matrix and to deduce S_0 from a dual optimal solution of (SDP) the "Shor plus RLT" relaxation of (QP).

$$(SDP)\begin{cases} \min\langle Q_0, X\rangle + c_0^T x \\ \quad \langle Q_r, X\rangle + c_r^T x \le e_r \quad r \in \mathcal{R} \\ \quad (3)-(4) \\ \quad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \end{cases}$$

To sum up, we solve (SDP) and deduce a positive semidefinite matrix S_0^* in order to build the following quadratic convex relaxation of (QP):

$$(PC^*)\begin{cases} \min F(x,X) \equiv \langle S_0^*, xx^T\rangle + \langle Q_0 - S_0^*, X\rangle + c_0^T x \\ \quad G_r(x) \equiv \langle Q_r, X\rangle + c_r^T x \le e_r \qquad r \in \mathcal{R} \\ \quad (3)-(4) \end{cases}$$

Problem (PC^*) is a quadratic convex problem that has the same optimal value as (SDP). It can be used to compute a tight lower bound to the optimal value of (QP) and it can also be used within a spatial B&B to globally solve (QP). This is implemented in the software `SMIQP` [6].

3. Improving an interval B&B using QCR

Our hybrid algorithm `QIBEX` is built upon solvers `IbexOpt` [1, 8] and `SMIQP` [6]. The steps of `IbexOpt` algorithm that were modified for designing `QIBEX` are surrounded in the pseudo-code of Algorithms 1 and 2.

Algorithm 1 describes the main procedure of our interval B&B. It starts from an initial node with domain $[\ell, u]$. The auxiliary variable $x_{obj} = f(x)$ represents the objective function value. First, the initial box is *contracted*. Then, `QIBEX` calls a procedure `QCR` that computes the matrix S_0^* and produces (PC^*). Finally, the B&B is described in the while loop, and works in best-first order. Once a node is selected, its domain is split into two parts by the `Bisect` separation procedure, and both sub-nodes (v_L, v_R) are handled by the `Contract&Bound` procedure before being added into the list of nodes by `UpdateNodes`. If one or both sub-nodes reach a sufficiently small

size ϵ_{sol}, they are rather pushed into the $smallV$ list and cannot be selected anymore. `QIBEX` uses a bisection heuristic adapted from a separation strategy dedicated to QPs [5]. If the strategy selects no variable, we resort to the `SmearSumRel` heuristic available in `IbexOpt` [8].

Algorithm `Qibex` $(f, g, x, [\ell, u]\ , \epsilon_{obj}, \epsilon_{sol})$
- $v \leftarrow$ `createNode`$([\ell, u])$
- $v \leftarrow$ `Contraction`(v, g)
- $\boxed{PC^* \leftarrow \texttt{QCR}\ (f, g, x, v.box)}$
- $vs \leftarrow \{v\}\ ; \quad smallV \leftarrow \emptyset$
- **while** $vs \neq \emptyset$ **and** $\widetilde{f} - f_{min} > \epsilon_{obj}$ **and** $\frac{\widetilde{f} - f_{min}}{|\widetilde{f}|} > \epsilon_{obj}$ **do**
  - $v \leftarrow$ `SelectNode`$(vs)\ ; \quad vs \leftarrow vs \setminus \{v\}$ /* node selection */
  - $(v_L, v_R) \leftarrow$ `Bisect` $(v, \boxed{PC^*})$ /* separation/bisection step */
  - $(v_L, vs, x_{\widetilde{f}}, \widetilde{f}) \leftarrow$ `Contract&Bound` $(v_L, vs, f, g, x, \epsilon_{obj}, x_{\widetilde{f}}, \widetilde{f}, \boxed{PC^*})$
  - $(v_R, vs, x_{\widetilde{f}}, \widetilde{f}) \leftarrow$ `Contract&Bound` $(v_R, vs, f, g, x, \epsilon_{obj}, x_{\widetilde{f}}, \widetilde{f}, \boxed{PC^*})$
  - $(vs, smallV) \leftarrow$ `UpdateNodes` $(v_L, v_R, \epsilon_{sol}, vs, smallV)$
  - $f_{min} \leftarrow \min_{vs\ \cup\ smallV} l_{obj}$
  - **return** $(f_{min}, x_{\widetilde{f}}, \widetilde{f})$

Algorithm 1: The `QIBEX` interval-based B&B dedicated to QPs

Algorithm `Contract&Bound` $(v, vs, f, g, x, \epsilon_{obj}, x_{\widetilde{f}}, \widetilde{f}, \boxed{PC^*})$
- $v \leftarrow$ `Contraction` $(v, g \cup \{f(x) = x_{obj}\} \cup \{x_{obj} \leq \widetilde{f} - \epsilon_{obj}\})$
- **if** $v.box \neq \emptyset$ **then**
  - $\boxed{(x_{pc*}, cost_{pc*}) \leftarrow \texttt{ConvexOptimize}(PC^*, v.box)}$
  - $(x_{\widetilde{f}}, cost) \leftarrow$ `FeasibleSearch` $(v, f, g, \epsilon_{obj}, \boxed{x_{pc*}})$ // Upperbounding
  - **if** $cost < \widetilde{f}$ **then**
    - $\widetilde{f} \leftarrow$ cost
    - $vs \leftarrow$ `FilterOpenNodes`$(vs, \widetilde{f} - \epsilon_{obj})$
  - $\boxed{v \leftarrow \texttt{Contraction}\ (v, g \cup \{cost_{pc*} \leq x_{obj} \leq \widetilde{f} - \epsilon_{obj}\} \cup \{f(x) = x_{obj}\}\ )}$
- **return** $(v, vs, x_{\widetilde{f}}, \widetilde{f})$

Algorithm 2: The `Contract&Bound` procedure run at each node

The other main improvement relates to the `Contract&Bound` procedure of Algorithm 2. First, the standard `Contraction` procedure implemented in `IbexOpt` is called. If it leads to an empty box, it proves the absence of solution in this domain, and we are done. Otherwise, the `ConvexOptimize` procedure evaluates PC^*, whose optimal solution is called x_{pc*}. The four sub-

sequent instructions carry out the upper bounding phase: `FeasibleSearch` tries to find a feasible point using several techniques of `IbexOpt`. In addition, it tests whether x_{pc*} is feasible. If such a point $x_{\widetilde{f}}$ is found, then $\widetilde{f}$ is updated and the open nodes are filtered by `FilterOpenNodes` to remove those with a lower bound l_{obj} greater than $\widetilde{f} - \epsilon_{obj}$. A last call to the `contraction` procedure is useful either if $cost_{pc*}$ improves l_{obj}, or a better upper bound $\widetilde{f}$ has been found. A last contribution is the handling of integer variables in `QIBEX` which was not provided in `IbexOpt`. For this, rounding to integer operations enforcing the integrality constraints are launched after the contractions and during feasible search computations.

4. First experiments

We compare two variants of our new algorithm: `QIBEX-B` and `QIBEX-BH`, with the solvers `IbexOpt`, `Baron` [9] and `Gurobi` [10]. `QIBEX-B` uses the `IbexOpt SmearSumRel` bisection strategy, while `QIBEX-BH` uses our tailored bisection heuristic. We run our experiences on continuous, mixed and integer intances and the results reveal the same trends. We present here the results of 50 instances of the class $IQCP_5$ [3], where each instance consists in minimizing a quadratic function of n general integer variables subject to 5 quadratic inequality constraints. In our tests n varies from 10 to 50, and each variable belongs to the interval $[0, 20]$. We set the time limit to 2 hours. We solve the semi-definite programs, as described in [4].

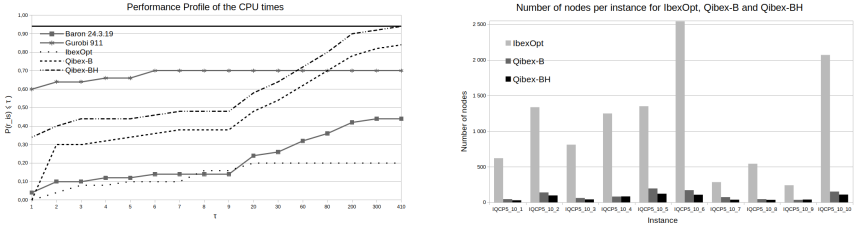

Figure 1. Performance profile of the CPU times (left) and number of nodes (right)

We present in Figure 1-left the performance profile [11] of the CPU times for methods `QIBEX-B`, `QIBEX-BH`, `IbexOpt`, `Baron 19.3.24`, and `Gurobi 9.1.1` for the 50 considered instances. We observe that `QIBEX-B` and `QIBEX- BH` perform best both in terms of CPU time and number of instances solved. In Figure 1-right, we compare the number of nodes required by `IbexOpt`,

QIBEX-B and QIBEX-BH on instances of size 10. The number of nodes is significantly reduced by using the PC^* bound (factor 11), and is further reduced by using the dedicated separation heuristic (additional factor 1.5). For larger instances, QIBEX-BH requires on average half number of nodes as QIBEX-B. The fact that QIBEX is even slower than Gurobi on the "easiest" problems comes from the pre-processing phase computing PC^*.

5. Conclusion and future research

Two main research directions come out of our work. The first one concerns the rigorous aspect of the solver which can be further improved. In particular, the solvers that compute the optimal solution to PC^* are not rigorous because they are subject to round-off errors. The other one concerns the extension of our approach to the case where the functions are non linear and non polynomial. A first idea would be to use a Taylor approximation of order 2 to approximate the problem by quadratic functions.

References

[1] I. Araya, G. Trombettoni, B. Neveu, G. Chabert. Upper Bounding in Inner Regions for Global Optimization under Inequality Constraints. *Journal of Global Optimization*, 60(2):145–164, 2014.

[2] F. Benhamou, F. Goualard, L. Granvilliers, J.-F. Puget. Revising Hull Box Consistency. In *Proc. of ICLP*, pages 230–244, 1999.

[3] A. Billionnet, S. Elloumi, A. Lambert. Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. *Mathematical Programming* 158(1):235–266, 2016.

[4] A. Billionnet, S. Elloumi, A. Lambert, A. Wiegele. Using a Conic Bundle method to accelerate both phases of a QCR. *IJOC*, 29(2):318–331, 2017.

[5] S. Elloumi A. Lambert. Global solution of non-convex quadratically constrained quadratic programs. *Opt. Meth. Soft.*, 34(1):98–114, 2019.

[6] A. Lambert. Solution of Mixed Integer Quadratic Programs (SMIQP), 2020.

[7] B. Neveu, G. Trombettoni, I. Araya. Adaptive Constructive Interval Disjunction: Algorithms Experiments. *Constraints Journal*, 20(4):452–467, 2015.

[8] G. Trombettoni, I. Araya, B. Neveu, G. Chabert. Inner regions interval linearizations for global optimization. In *Proceedings of AAAI*, volume 25, 2011.

[9] N.V. Sahinidis M. Tawarmalani. BARON 19.3.24: Global Optimization of Mixed-Integer Nonlinear Programs, 2019.

[10] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual 911, 2021 .

[11] D. Dolan, J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming* 91:201–213, 1986.