

Quadratic Convex Reformulations for MultiObjective Binary Quadratic Programming

Marianna De Santis¹, Lucas Létocart², Yue Zhang²

¹ Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze
Via di Santa Marta 3, 50139 Firenze, Italia
marianna.desantis@unifi.it

² LIPN, CNRS UMR 7030, Université Sorbonne Paris Nord
99 Av. Jean Baptiste Clément, 93430 Villetaneuse, France
{lucas.letocart,yue.zhang}@lipn.univ-paris13.fr

Keywords : *Multiobjective Optimization, Binary Quadratic Problems, Quadratic Convex Reformulations, Branch-and-Bound algorithm.*

1 Introduction

In the context of multiobjective integer programming $\{\min_{x \in \mathcal{X}} (f_1(x), \dots, f_p(x))^\top\}$, the *nondominated set* \mathcal{N} can be enumerated via branch-and-bound frameworks [3, 4, 6]. Inspired by Quadratic Convex Reformulation (QCR) [1, 2] methods, this work focuses on multiobjective binary (nonconvex) quadratic problems (MO-BQPs) and devise a branch-and-bound framework, named MObbBQ.

$$\begin{aligned} \min_x \quad & (f_1(x), \dots, f_p(x))^\top \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0, 1\} \quad i \in \{1, \dots, n\}. \end{aligned} \tag{MO-BQP}$$

$f_j(x) = x^\top Q_j x + (c^j)^\top x$, for all $j \in [p]$ with symmetric matrix $Q_j \in \mathcal{S}^n$ and vector $c^j \in \mathbb{R}^n$.

2 MObbBQ: a branch-and-bound framework for MultiObjective Binary Quadratic programs

Due to nonconvex functions, the difficulty of building a valid lower bound set (LBS) arises from the nonconvex nondominated boundaries of the continuous relaxation of MO-BQP. As QCR methods provide an equivalent convex reformulation and a tight continuous relaxation bound, we propose to employ the intersection of hyperplanes that outer approximates the nondominated boundaries of the continuous relaxation of MO-BQP convexified by QCR (see Figure 1). We theoretically prove the validity of the proposed LBS as well.

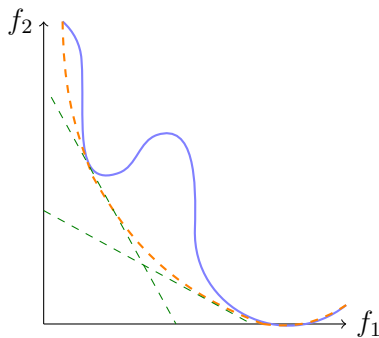


FIG. 1: Lower bound set.

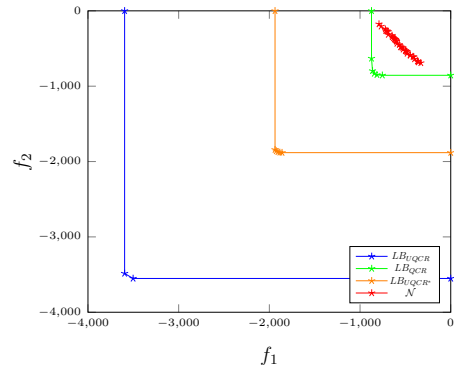


FIG. 2: Comparison on different LBS strategies.

The optimal QCR parameters can be obtained by solving a semidefinite problem (SDP). However computing p SDPs at every node is highly time-consuming. Assuming the MOBBQ tree is branched in a static order, for example from x_1 to x_n , then all nodes at the same level d share the same p matrices in \mathcal{S}^{n-d} . Consequently we propose to calculate and store QCR parameters for the $n \times p$ quadratic terms in a preprocessing phase.

On the other hand for a better trade-off between LBS quality and computing time, we propose different strategies for LBS construction with different tightness, mainly based on the three following approaches:

- MOBBQ_{QCR} applies QCR on each objective function i.e. $\forall j \in [p], f_j(x) + \sum_{i=1}^n \delta_i(x_i^2 - x_i) + \beta(A=x - b_{=})^2$
- MOBBQ_{UQCR} applies UQCR on each objective function i.e. $\forall j \in [p], f_j(x) + \sum_{i=1}^n \delta_i(x_i^2 - x_i)$
- MOBBQ_{UQCR*} applies UQCR but considering the relaxed constraint $A^d x \leq b$ in the SDPs for calculating δ , where A^d denotes the residual constraint matrix at level d

Figure 2 illustrates different LBS at the root node of a bi-objective k-item quadratic knapsack instance of size 20.

Our MOBBQ branches in a static variable order and in depth-first search order. The upper bound set is initialized to empty set and is updated whenever a new nondominated point is found [5]. Additionally we provide theoretical proofs on the correctness of MOBBQ. Our implementations are available at <https://github.com/Yue0925/MultiObjectiveAlgorithms.jl>.

3 Experimental results

We evaluate the performance of MOBBQ on the following bi- and tri-objective instances:

- Multiobjective max-cut: Within the time limit of 1800 CPU seconds, our MOBBQ_{UQCR} algorithm beats the ϵ -constraint method on bi-objective instances with matrices density $\geq 75\%$. For tri-objective instances, MOBBQ_{UQCR} significantly outperforms other strategies.
- Multiobjective k-item quadratic knapsack: We notice that there is no huge difference in tree size among all different versions of algorithms. For the majority of instances, MOBBQ_{UQCR} is slightly faster (especially on bi-objective instances), which explains the better strategy of MOBBQ is to fast enumerate integer points rather than pruning dominated nodes. Compared to the ϵ -constraint method, our algorithms are still more competitive on dense instances.

References

- [1] Alain Billionnet and Sourour Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Math. Program.*, 109:55–68, 2007.
- [2] Alain Billionnet, Sourour Elloumi, and Marie-Christine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discret. Appl. Math.*, 157(6):1185–1197, 2009.
- [3] Marianna De Santis, Giorgio Grani, and Laura Palagi. Branching with hyperplanes in the criterion space: The frontier partitioner algorithm for biobjective integer programming. *European J. Oper. Res.*, 283(1):57–69, 2020.
- [4] Nicolas Forget, Sune Lauth Gadegaard, Kathrin Klamroth, Lars Relund Nielsen, and Anthony Przybylski. Branch-and-bound and objective branching with three or more objectives. *Comput. Oper. Res.*, 148:106012, 2022.
- [5] Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. On the representation of the search region in multi-objective optimization. *European J. Oper. Res.*, 245(3):767–778, 2015.
- [6] Sophie N. Parragh and Fabien Tricoire. Branch-and-bound for bi-objective integer programming. *INFORMS J. Comput.*, 31(4):805–822, 2019.