- manuscript No.
(will be inserted by the editor)

The p-Lagrangian relaxation for nonconvex MIQCQP problems with complicating constraints

Tiago Andrade · Nikita Belyak · Andrew Eberhard · Silvio Hamacher · Fabricio Oliveira

Version: July 24, 2020

Abstract This paper presents a novel technique to solve nonconvex mixed-integer quadratically constrained quadratic programming (MIQCQP) with separable structures, such as those arising in deterministic equivalent representations of two-stage stochastic programming problems. In general, the nonconvex nature of these models still poses a challenge for available solvers, which do not consistently perform well for larger-scale instances. Therefore, we propose an appealing alternative algorithm that allows for overcoming computational performance issues. Our novel technique, named the p-Lagrangian relaxation, is a primal-dual decomposition method that combines a bundle-method inspired Lagrangian decomposition with mixed-integer programming-based relaxations. These relaxations are obtained using the reformulated normalised multiparametric disaggregation technique (RNMDT) and can be made arbitrarily precise by means of a precision parameter p. We provide a technical analysis showing the convergent behaviour of the approach as the approximation is made increasingly precise. We observe that, in addition to demonstrating superior performance in terms of convergence behaviour, the proposed method presents significant reductions in computational time.

1 Introduction

In this paper, we focus on nonconvex mixed-integer quadratically constrained quadratic programming (MIQCQP) models with bounded variables, arising, for example, from the deterministic equivalent model (DEM) of two-stage stochastic programming problems. These models can be generally represented as

$$\text{DEM}: \max. \quad \sum_{j\in\mathcal{VI}} I_j^0 x_j + \sum_{s\in\mathcal{S}} P^s \left(\sum_{i\in\mathcal{VC}} \sum_{j\in\mathcal{VC}} Q_{i,j}^{s,0} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_{i,s}^{s,0} y_i^s \right) \tag{1}$$

$$\text{s.t.:} \quad \sum_{i\in\mathcal{VC}} \sum_{j\in\mathcal{VC}} Q_{i,j}^{s,r} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_i^{s,r} y_i^s + \sum_{j\in\mathcal{VI}} I_j^{s,r} x_j + K^{s,r} \leq 0,$$

Address(es) of author(s) should be given

$$\forall s \in \mathcal{S}, \forall r \in \mathcal{R} \quad (2)$$

$$y_i^s \in [Y_i^{L,s}, Y_i^{U,s}], \forall s \in \mathcal{S}, \forall i \in \mathcal{VC} \quad (3)$$

$$x_j \in \{X_j^L, \ldots, X_j^U\}, \forall j \in \mathcal{VI}, \quad (4)$$

where $\mathcal{S}$ is the set of scenarios, P^s is the probability of the scenario $s \in S$, $\mathcal{R}$ is the set of the constraint indices for each scenario, $\mathcal{VI}$ ($\mathcal{VC}$) is the set of integer (continuous) variables indices, $Q_{i,j}^{s,r}$ are symmetric matrices for all $s \in \mathcal{S}$, $r \in \{0\} \cup \mathcal{R}$. $I_j^{s,r}$ ($C_i^{s,r}$) is a set of linear coefficients correspondent to the integer (continuous) variables and $K^{s,r}$ is a set of constants. Variables x_j, $\forall j \in \mathcal{VI}$, can assume any integer value between its bounds X_j^L and X_j^U and variables y_i^s, $i \in \mathcal{VC}$ and $s \in \mathcal{S}$, assumes any continuous value between $Y_i^{L,s}$ and $Y_i^{U,s}$.

The range of useful applications of MIQCQP models is noticeably broad, comprising of sectors such as finance, engineering, and the process industry, permeating several applications arising in management science and operations research [43]. Recent papers utilising this type of model formulation include problems such as planning optimal batteries management for the reduction of power losses [35], the enhanced index-tracking problem formulation [53], the operational planning of refineries [2], maximising the gross profit in a three-level supply chain [33], storage investment planning [57] and solution approaches for a gas flow problem with general setups [51], to name only a few of its potential applications.

A MIQCQP model is defined as convex if its continuous relaxation is convex, regardless of the nonconvexity introduced by the integrality requirements. Therefore, the DEM is convex if $Q_{i,j}^{s,r}$ is positive semi-definite for all $s \in \mathcal{S}$ and $r \in \{0\} \cup \mathcal{R}$, being nonconvex otherwise. The latter (nonconvex) is the case addressed in this paper.

A MIQCQP problem is an undecidable problem if the variables are unbounded [31] and a NP-hard problem otherwise. Thus, in this paper, we assume that all variables have known bounds. In principle, MIQCQP models can be solved by available open-source and commercial solvers such as recent releases of Gurobi [28], Couenne [40] or Baron [55]. However, these solvers still lack in performance robustness when facing larger-scale instances, such as those arising as DEM from two-stage stochastic programs.

Due to the challenging nature of MIQCQP problems, several solution approaches have been developed [12]. These can be generally divided into exact and heuristic methods. While the former can guarantee the convergence to the global optimum, the latter can only guarantee local optimality of the solutions obtained. Furthermore, almost all methods involve the employment of relaxation techniques as a subroutine. The approximation of the original MIQCQP problem with mixed-integer linear programs (MILP) [1, 41] is the most typical relaxation strategy employed in this setting. However, this type of relaxation can significantly increase the number of variables and constraints compared to the original (primal) model, thus deteriorating computational time.

One of the most widely used exact algorithms to solve a MIQCQP problem is the branch-and-bound (BnB) method and its variants. If the problem is convex, BnB is capable of providing globally optimal solutions by relaxing the integrality constraints to obtain bounds. In the case of nonconvex MIQCQP model, a spatial BnB is typically used, employing (convex) relaxations of the nonconvex terms that are the source of the nonlinearity. Castro [17] present a spatial BnB with the relax-

ations relying on normalised multiparametric disaggregation technique for solving nonconvex MIQCQP problems. Ding et al. [23] present an integration of spatial BnB and standard BnB method named bi-level branch-and-bound technique capable of solving MIQCQP problems with superior solution quality and convergence characteristics. Berthold et al. [5] develop MIQCQP problems solver based on the combination of constraint programming and branch-and-cut algorithm exploiting cutting planes to tighten the relaxations.

Another conceptually different approach to solve MIQCQP problems is the employment of decomposition, which is one of the main focuses of this paper. The key concept of this method is to split the problem into several smaller subproblems that are more tractable and can be solved independently, possibly in parallel. This separability feature, in turn, can improve computational performance when the relaxation is introduced.

By making explicit the non-anticipativity conditions (NAC) of the first-stage variables x in the DEM, the reformulated deterministic equivalent model (RDEM) can be represented as

$$\text{RDEM}: \max. \quad \sum_{s\in\mathcal{S}} P^s \left(\sum_{j\in\mathcal{VI}} I_j^0 x_j^s + \sum_{i\in\mathcal{VC}} \sum_{j\in\mathcal{VC}} Q_{i,j}^{s,0} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_i^{s,0} y_i^s \right) \tag{5}$$

$$\text{s.t.:} \quad \sum_{i\in\mathcal{VC}} \sum_{j\in\mathcal{VC}} Q_{i,j}^{s,r} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_i^{s,r} y_i^s + \sum_{j\in\mathcal{VI}} I_j^{s,r} x_j^s + K^{s,r} \leq 0, \quad \forall s \in \mathcal{S}, \forall r \in \mathcal{R} \tag{6}$$

$$x_j^s \in \{X_j^L, \ldots, X_j^U\}, \forall s \in \mathcal{S}, \forall j \in \mathcal{VI} \tag{7}$$

$$x_j^{s'} - x_j^s = 0, \forall s \in \mathcal{S} \setminus \{s'\}, \forall j \in \mathcal{VI} \tag{8}$$

and (3),

where $s' \in S$ is a reference scenario.

The introduction of the NAC in the above RDEM problem results in an explicit nearly-decomposable equivalent of the original problem DEM with an exposed block-angular structure [8,9]. Consequently, one can obtain $s \in \mathcal{S}$ essentially independent MIQCQP subproblems where (8) is the only set of linear constraints that relates variables from distinct subproblems. These constraints are referred to as linking or complicating constraints. Therefore, if one could remove these constraints, each of the subproblems could be solved independently. Hereinafter, we will use the term subproblems when referring to each element of the block-angular structure of the RDEM problem. It is important to highlight that there are multiple alternative ways in which the NAC could be represented. In what follows, we use the representation presented by Oliveira et al. [42], as the authors report better computational results in a similar context. Nonetheless, the developments presented hereinafter are not dependent on the specific choice for representing NACs.

In classical linear programming, the three most common decomposition framework are Dantzig-Wolfe decomposition (DWD), Benders decomposition (BD) and Lagrangian decomposition (LD), of which the last is explored in this paper. It should be noted that LD is not only limited to be considered in a decomposition framework, but it can also be used to devise easier-to-solve (i.e., typically

smaller in scale due to separability) equivalent problems, which is often referred to as *Lagrangian relaxation* (LR). On the other hand, LD involves creating copies of the complicating variables – i.e., the first stage variables x – for each scenario and introducing NAC to ensure primal feasibility (the reformulation used to obtain RDEM). LD would then be the employment of LR to remove the generated complicating constraints (NAC). There is a well-known connection between these decomposition approaches, in that if the LD is solved using the cutting-planes algorithm, it can be viewed as a BD. Furthermore, BD can be stated as the dual of DWD.

The decision on which decomposition methods is the most appropriate depends on the problem structure, such as the presence of complicating constraints. Furthermore, the BD (and DWD) in its classical form cannot be applied to general nonlinear programming problems. Addressing this shortcoming, Geoffrion [26] proposed the generalised Benders decomposition (GBD) based on BD to decompose convex nonlinear programming problems. Later, Li, Tomasgard, and Barton [37] improved the GBD through the nonconvex generalised Benders decomposition (NGBD) to decompose nonconvex nonlinear programming. In contrast, the advantage of the LD is that it can be directly applied to a nonconvex problem. However, it should be noted that the nonconvex subproblems in this context must be solved to global optimality, as it will be discussed in further details in Section 2. This can be challenging, since one would still have to solve nonconvex MIQCQP subproblems.

This paper presents a new class of relaxations called p-Lagrangian. The p-Lagrangian is a composition of mixed-integer programming (MIP)-based relaxation method named reformulated normalised multiparametric disaggregation technique (RNMDT) [1] and LR. The idea of employing RNMDT to devise a MIP approximation of resulting LR problem is inspired by the approach used to expand the GBD to the NGBD. Therefore, the p-Lagrangian relaxation allows one to use the same decomposition strategies as the classic LR, but with subproblems that can be solved to global optimality using more robust MILP technology.

This paper is structured as follows. Section 2 reviews the fundamental concepts of the LR. Section 3 describes the RNMDT relaxation. The p-Lagrangian relaxation of the MIQCQP problem is presented in Section 4 followed by technical results that describe its convergence behaviour. A iterative algorithm for solving p-Lagrangian relaxation problems is described in Section 5. Furthermore, numeric experiments are presented in Section 6, where the contributions of this paper are tested on randomly generated instances. Finally, in Section 7 we draw conclusions and provide directions for further research.

2 Theoretical background

Lagrangian relaxation is a bounding technique for solving a given arbitrary optimisation problem which has important applications for nonconvex problems such as MILP problems [14,27]. However, methods based on Lagrangian duality cannot be straightforwardly applied to nonconvex problems, as discussed in what follows.

First, let us formally define the concept of a relaxation, which is central for the developments hereinafter. Consider the following problem $P : \max\{f(x) : x \in S\}$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $S \subseteq \mathbb{R}^n$ and the problem $P_R : \max\{f_R(x) : x \in S_R\}$ where $f_R : \mathbb{R}^n \to \mathbb{R}$ and $S_R \subseteq \mathbb{R}^n$.

Definition 1 *P_R is a relaxation of P if and only if:*

1. $f_R(x) \geq f(x)$, $\forall x \in S$
2. $S \subseteq S_R$

Let $f^s : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|} \to \mathbb{R}$ and $g^{s,r} : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|} \to \mathbb{R}$ $\forall s \in S$ and $\forall r \in R$ be continuous twice differentiable functions. Suppose that we are interested in solving the (primal) problem (9), which is introduced to ease the notation in our developments.

$$\begin{aligned} f^* = &\max._{x,y} \sum_{s\in S} f^s(x^s, y^s) \\ \text{s.t.:}\ & g^{s,r}(x^s, y^s) \leq 0, \forall s \in S, \forall r \in R \\ & x^{s'} - x^s = 0, \forall s \in S \setminus \{s'\} \\ & x^s \in X, \forall s \in S, \\ & y^s \in Y^s, \forall s \in S. \end{aligned} \tag{9}$$

As can be noticed, (9) is equivalent to the RDEM problem, by making

$$f^s(x^s, y^s) = P^s(\sum_{j\in\mathcal{VI}} I_j^0 x_j^s + \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q_{i,j}^{s,0} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_i^{s,0} y_i^s),$$

$$g^{s,r}(x^s, y^s) = \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q_{i,j}^{s,r} y_i^s y_j^s + \sum_{i\in\mathcal{VC}} C_i^{s,r} y_i^s + \sum_{j\in\mathcal{VI}} I_j^{s,r} x_j^s + K^{s,r},$$

while X, Y^s in (9) represents the variable bounds ((7), (3)) for integer and continuous variables, respectively.

Let $D^s = \{x^s, y^s \mid x^s \in X,\ y^s \in Y^s,\ g^{s,r}(x^s, y^s) \leq 0,\ \forall r \in R\}$ be a feasibility set $\forall s \in S$ and let $\mathcal{D} = \bigcup_{s\in S} D^s$. With this definition in mind, the Lagrangian dual function $\phi : \mathbb{R}^{(|S|-1)\times|\mathcal{VI}|} \to \mathbb{R}$ can be defined as

$$\phi(\lambda) = \sup_{(x,y)\in\mathcal{D}} \sum_{s\in S} f^s(x^s, y^s) + \sum_{s\in S\setminus s'} (\lambda^s)^\top (x^{s'} - x^s), \tag{10}$$

where the components of $\lambda \in \mathbb{R}^{(|S|-1)\times|\mathcal{VI}|}$ are the Lagrangian multipliers and $x, y = \{x^s, y^s\}_{s\in S}$. In this setting, it is common to say that the NAC are being *relaxed* [10]. In what follows, we present important properties of the Lagrangian dual function ϕ for the developments presented.

Proposition 1 *The Lagrangian dual function ϕ is convex.*

Proof. A function is convex if and only if its epigraph is a convex set. The epigraph of $\phi(\lambda)$ is the intersection of the epigraphs of affine functions

$$\sum_{s\in S} f^s(x_f^s, y_f^s) + \sum_{s\in S\setminus s'} (\lambda^s)^\top (x_f^{s'} - x_f^s)$$

where x_f^s and y_f^s are given by

$$x_f^s, y_f^s = \arg \sup_{x^s, y^s \in \mathcal{D}} \sum_{s \in S} f^s(x^s, y^s) + \sum_{s \in S \setminus s'} (\lambda_f^s)^\top (x^{s'} - x^s)$$

for the fixed λ_f^s. Since any affine function is convex, so are their epigraphs. An arbitrary intersection of convex sets is convex as well, and thus, ϕ is convex. ■

For any value of the Lagrangian multiplier λ, the Lagrangian dual function ϕ provides an upper bound (UB) for the primal problem (9). Our objective is to obtain the tightest (i.e., the lowest) UB. Therefore, we are interested in solving the following Lagrangian dual problem (LDP)

$$\phi^* = \min_{\lambda} \phi(\lambda) \tag{11}$$

Even though the function ϕ is convex, evaluating $\phi(\lambda)$ for a given λ_f may require solving a nonconvex problem if either f^s or $g^{s,r}$ are nonconvex for some $s \in S$ and $r \in R$. Thus, evaluating ϕ may be as hard to solve to global optimality as the primal problem. We refer to this later on as Issue 1. Proposition 2 describes how one can use LDP to obtain bounds for (9), a property generally referred to as *weak duality*.

Proposition 2 *Let $f^s : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|} \to \mathbb{R}$ and $g^{s,r} : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|} \to \mathbb{R}$, $\forall s \in S$ and $\forall r \in R$, be continuous twice-differentiable functions and suppose we consider the primal problem* (9)*. Let $\phi : \mathbb{R}^{(|S|-1)\times|\mathcal{VI}|} \to \mathbb{R}$ be Lagrangian dual function forming the Lagrangian dual problem* (11)*. Let f^* and ϕ^* be the optimal values of the primal and dual problems accordingly. Then, the following statements are true:*

1. $\phi(\lambda) \geq \sum_{s \in S} f^s(x^s, y^s)$, $\forall \lambda \in \mathbb{R}^{(|S|-1)\times|\mathcal{VI}|}, x, y \in \mathcal{D} \mid x \in \mathcal{X}^{NAC}$, *where* $\mathcal{X}^{NAC} = \{x \mid (x^{s'} - x^s) = 0, \forall s \in S \setminus \{s'\}\}$.
2. $\phi^* \geq f^*$.

Proof. To prove the first part of the proposition, assuming that $x, y \in \mathcal{D}$ and recalling the formulation of the dual problem, we have

$$\begin{aligned}
\phi(\lambda) &= \sup_{x,y \in \mathcal{D}} \sum_{s \in S} f^s(x^s, y^s) + \sum_{s \in S \setminus s'} (\lambda^s)^\top (x^{s'} - x^s) \\
&\geq \sup_{x,y \in \mathcal{D} | x \in \mathcal{X}^{\mathrm{NAC}}} \sum_{s \in S} f^s(x^s, y^s) + \sum_{s \in S \setminus s'} (\lambda^s)^\top (x^{s'} - x^s) \\
&= \sup_{x,y \in \mathcal{D} | x \in \mathcal{X}^{\mathrm{NAC}}} \sum_{s \in S} f^s(x^s, y^s) \geq \sum_{s \in S} f^s(x^s, y^s).
\end{aligned}$$

The second item of the proposition is a direct consequence of the first. ■

It is important to highlight that, for problems including integer decision variables, a duality gap might exist (i.e., only weak duality is valid). This will hereinafter be called Issue 2. Nevertheless, the Lagrangian dual problem can still be used to obtain bounds for the primal problem (9), as stated in part 1 of Proposition 2.

Therefore, if the primal problem is nonconvex, there are two main issues to be addressed when using the Lagrangian duality to devise a solution method, i.e., we

must solve nonconvex subproblems (Issue 1), and only weak duality holds (Issue 2).

The second issue is traditionally addressed by modifying the Lagrangian function to include penalty terms. Linear penalty expressions were introduced by Pietrzykowski [45] and Zangwill [59] and quadratic penalty expressions by Courant [21]. Penalty expressions with both linear and quadratic pieces were proposed by Rockafellar [47]. A generalised approach called the augmented Lagrangian, also known as the methods of multipliers, was first studied by Hestenes [30] and Powell [46]. Two disadvantages of the augmented Lagrangian is that it usually ruins the problem separable structure when it exists, thus, compromising decomposition strategies [54], and it also adds nonlinearities to the problem. The progressive hedging (PH) method, proposed by Rockafellar and Wets [48] overcomes the former obstacle. Santos et al. [50] evaluated the application of PH for solving hydrothermal systems short-term operational planning problems comparing to the Nested Decomposition. Veliz et al. [56] demonstrated that PH is competitive with a direct solution when applied to solve the mixed-integer problem of medium-term forest planning. Lokketangen and Woodruff [38] combined the tabu search method with PH to solve multistage, stochastic mixed-integer (0, 1) programming problems. Boland et al. [10] enhanced the PH convergence characteristics by introducing a Frank–Wolfe-based method to compute primal updates in PH and Boland et al. [11] presented how it could be further considered under a bundle method setting. The p-Lagrangian decomposition method proposed in this paper can be potentially generalised to those settings as well. However, in this study, we only concentrate on Issue 1 for the nonconvex MIQCQP problems. As it will become clear, all of these developments could also be incorporated within the framework of the p-Lagrangian relaxation in an attempt to address Issue 2 and is left for future research.

A common approach to solve decomposable QCQP problems is to relax all quadratic constraints to obtain a subproblem that is equivalent to a semidefinite programming problem that can be solved to global optimality using appropriate interior-point methods [3]. On the other hand, the formulation of the LDP is not trivial to find, especially if not all constraints are relaxed. Another issue with this approach is that the quality of the dual bounds is dependent on the subset of constraints to be relaxed. Dentcheva and Römisch [22] proposed a framework to estimate how large the duality gap will be when a specific subset of constraints is relaxed. The framework was then applied to decide which constraints to relax to generate the smallest duality gap. In this paper, only complicating constraints (eqs. (8)) are considered for being relaxed, as we focus on problems with block-angular separable structure, such as two-stage stochastic programming problems. It is worth highlighting that the framework presented here is, however, general enough to be employed to any other MIQCQP problem with a similar separable structure.

In what follows, we concentrate on addressing Issue 1, by developing a new class of dual problems replacing the nonconvex MIQCQP subproblems with MIP-based relaxations obtained using the reformulated normalised multiparametric disaggregation technique (RNMDT). Even though MIP problems might also be computationally challenging, there are more reliable techniques available for solving MIP problems that are efficient in many cases (e.g., branch-and-cut) and widely avail-

able off-the-shelf commercial implementations in solvers such as Gurobi [28] and CPLEX [20].

2.1 Motivating example

The following motivating example illustrates that, even in simple cases, solving the LDP might not provide a valid dual bound if the Lagrangian dual function ϕ cannot be appropriately evaluated. That will be the case if $x^s, y^s, \forall s \in S$, is not a global minimiser for $\sup_{x,y\in\mathcal{D}} \sum_{s\in S} f^s(x^s, y^s) + \sum_{s\in S\setminus s'} (\lambda_f^s)^\top (x^{s'} - x^s)$ for a given λ_f. Consider the following problem.

$$\begin{aligned} \max._{x_1,x_2} \quad & x_1 x_2 \\ \text{s.t.:} \quad & x_1 + 2x_2 = 1 \\ & x_1, x_2 \in [0, 0.75]. \end{aligned} \tag{12}$$

For a fixed Lagrangian multiplier $\lambda \in \mathbb{R}$, the evaluation of the Lagrangian dual function ϕ corresponding to the Problem (12) is

$$\begin{aligned} \max._{x_1,x_2} \quad & x_1 x_2 + \lambda(x_1 + 2x_2 - 1) \\ \text{s.t.:} \quad & x_1, x_2 \in [0, 0.75]. \end{aligned} \tag{13}$$

This problem and the evaluation of ϕ were implemented in Julia (version 1.3.1) [7]. Solving the Problem (12) with the global solver Gurobi (version 9.0.0), we obtain the optimal value 0.125. Since the primal problem and its LDP have a (weak) dual relationship, Problem (13) should provide an UB as its optimal value for all fixed values of the Lagrangian multiplier λ_f, i.e., a value greater or equal than 0.125. However, if we solve for a fixed value of $\lambda_f = -0.100$ using the local solver Ipopt [58], it returns a solution which is zero for all variables, corresponding to the objective function value equal to 0.100. This is not a valid UB since the feasible solution $(x_1, x_2) = (0.502, 0.249)$ has a greater value than that.

We could improve the solution for this problem by using another local solver that utilises alternative methods providing a warm start or even by applying a multi-start strategy. However, if one cannot guarantee that the solutions obtained for (13) are global maxima, one cannot be sure about the validity of the resulting bounds which, in turn, compromises the validity of solutions methods that rely on Lagrangian relaxation. In Section 4.2, we present how to address the issue of generating valid Lagrangian bounds for nonconvex problems using RNMDT (Issue 1).

3 Reformulated normalised multiparametric disaggregation technique

The reformulated normalised multiparametric disaggregation technique (RNMDT) [1] is an enhanced version of the normalised multiparametric disaggregation technique (NMDT), originally proposed in [34] and further developed in [15, 16]. RNMDT allows for the development of mixed-integer based relaxations for MIQCQP models that can be set to be arbitrarily precise, at the expense of trading off precision and the total of additional auxiliary (binary and continuous) variables

required. The enhancement of the original framework is due to (i) a significant reduction in the number of auxiliary variables and constraints and (ii) a dynamic procedure for selecting variables to have their discretised representation made more precise.

Suppose that we seek to solve the following problem, which is the same as RDEM problem but without the separable structure, to ease the notation

$$\text{max.} \quad \sum_i \sum_j Q^0_{i,j} y_i y_j + \sum_j I^0_j x_j + \sum_i C^0_i y_i \tag{14}$$

$$\text{s.t.:} \quad \sum_i \sum_j Q^r_{i,j} y_i y_j + \sum_j I^r_j x_j + \sum_i C^r_i y_i + K^r \leq 0, \forall r \in R \tag{15}$$

$$y_i \in [Y^L_i, Y^U_i], \forall i \in \mathcal{VC} \tag{16}$$

$$x_j \in \{X^L_j, \ldots, X^U_j\}, \forall j \in \mathcal{VI}, \tag{17}$$

where the parameters $Q^r, I^r, C^r,$ and K^r are as in Section 1.

Let

$$QT = \{(i,j) \in \mathcal{VC} \times \mathcal{VC} \mid j \geq i, \exists r \in R, |Q^r_{i,j}| > 0\}$$

and

$$DS = \{j \in \mathcal{VC} \mid \exists i \in \mathcal{VC}, (i,j) \in QT\}.$$

The set QT comprises the indices of variables that appear in at least one quadratic term, while DS corresponds to the set of variables that will be discretised. In this context, the discretisation of the variables y_j, $\forall j \in DS$, can be achieved by introducing the following constraints:

$$y_j = (Y^U_j - Y^L_j)(\sum_{l \in Z_{p,-1}} 2^l z_{j,l} + \Delta y_j) + Y^L_j, \forall j \in DS \tag{18}$$

$$0 \leq \Delta y_j \leq 2^p, \forall j \in DS \tag{19}$$

$$z_{j,l} \in \{0,1\}, \forall j \in DS, l \in Z_{p,-1}, \tag{20}$$

where $Z_{a,b} = \{a, \ldots, b\}$ is the subset of integer numbers ranging from a to b (inclusive), (18) is used to normalise decision variables y_j and the term

$$\left\{ \sum_{l \in Z_{p,-1}} 2^l z_{j,l} + \Delta y_j \right\} \in [0,1]$$

is discretised in partitions of the size 2^p each, where the integer parameter $p < 0$ corresponds to a precision factor. The auxiliary variables Δy_j are added to allow the term to attain all values in the interval [0, 1]. Notice that, if the discretisation would have used the partitions of the size 10^p instead, the absolute value of the precision factor p would be equivalent to the number of digits used when normalising the integer variables. However, we use a basis 2 instead, since it allows for more compact relaxations in terms of the number of auxiliary variables, as demonstrated in [1].

For each bilinear term $y_i y_j$, only one of the variables need to be discretised, as the resulting product between continuous and integer variables can be linearised by

a standard equivalent reformulation. Multiplying both sides of (18) by $y_i, \forall i \in \mathcal{VC}$, gives the constraints

$$y_i y_j = (Y_j^U - Y_j^L)(\sum_{l \in Z_{p,-1}} 2^l y_i z_{j,l} + y_i \Delta y_j) + y_i Y_j^L, \forall i, j \in QT. \quad (21)$$

Next, we introduce auxiliary variables $w_{i,j}$, $\hat{y}_{i,j,l}$ and $\Delta w_{i,j}$ to represent the products $y_i y_j$, $y_i z_{j,l}$ and $y_i \Delta y_j$, respectively. The resulting set of constraints obtained is given by

$$w_{i,j} = (Y_j^U - Y_j^L) \left(\sum_{l \in Z_{p,-1}} 2^l \hat{y}_{i,j,l} + \Delta w_{i,j} \right) + y_i Y_j^L, \forall i, j \in QT \quad (22)$$

$$Y_i^L z_{j,l} \leq \hat{y}_{i,j,l} \leq Y_i^U z_{j,l}, \forall i, j \in QT, l \in Z_{p,-1} \quad (23)$$

$$Y_i^L (1 - z_{j,l}) \leq y_i - \hat{y}_{i,j,l} \leq Y_i^U (1 - z_{j,l}), \forall i, j \in QT, l \in Z_{p,-1} \quad (24)$$

$$2^p (y_i - Y_i^U) + Y_i^U \Delta y_j \leq \Delta w_{i,j} \leq 2^p (y_i - Y_i^L) + Y_i^L \Delta y_j, \forall i, j \mid (i,j) \in QT \quad (25)$$

$$Y_i^L \Delta y_j \leq \Delta w_{i,j} \leq Y_i^U \Delta y_j, \forall i, j \mid (i,j) \in QT, \quad (26)$$

where constraints (23) and (24) form an exact linearisation of the product between binary and a continuous variable. The constraints (25) and (26) provide a relaxation of the product of two continuous variables and are known as McCormick envelopes [15].

Furthermore, using the previously defined variable $w_{i,j}$, the objective function (14) and the original constraints (15) are replaced by (27) and (28), respectively.

$$\text{max.} \quad \sum_{i|(i,i) \in QT} Q_{i,i}^0 w_{i,i} + 2 \sum_{(i,j) \in QT | j > i} Q_{i,j}^0 w_{i,j} + \sum_j I_j^0 x_j + \sum_i C_i^0 y_i \quad (27)$$

$$\sum_{i|(i,i) \in QT} Q_{i,i}^r w_{i,i} + 2 \sum_{(i,j) \in QT | j > i} Q_{i,j}^r w_{i,j} + \sum_j I_j^r x_j + \sum_i C_i^r y_i + K^r \leq 0, \quad \forall r \in R \quad (28)$$

Summarising all the above, we can define the RNMDT relaxation as follows.

Definition 2 *For every integer $p < 0$, the RNMDT relaxation of the problem* (14)–(17) *is defined as the problem of maximising the objective function* (27)*, subject to the constraints* (16)–(17)*,* (18)–(20)*,* (22)–(26) *and* (28)*.*

Hereinafter, we will refer to the relaxation obtained using RNMDT with an arbitrary parameter p as to RNMDT$_p$. The following results allow for describing the relationship between the problem (14)–(17) and the RNMDT$_p$, which will be useful in the remaining of the paper.

With Definition 1 in mind, the following theorem defines the relationship between the original problem (14)–(17) and its RNMDT$_p$ for any $p < 0$.

Theorem 3 *Suppose we consider MIQCQP problem* (14)–(17) *and correspondent RNMDT$_p$ problems, $\forall p < 0$, introduced in Definition 2. Then the RNMDT$_p$ problem is a relaxation to the problem* (14)–(17) *for every $p < 0$. Moreover, for any pair of (p_1, p_2) with $p_1 < p_2 < 0$, the RNMDT$_{p_2}$ is a relaxation of the problem RNMDT$_{p_1}$. Consequently, for any pair of (p_1, p_2) with $p_1 < p_2 < 0$, the RNMDT$_{p_1}$ is a tighter (or equal) relaxation of the problem* (14)–(17) *than RNMDT$_{p2}$.*

Proof. For the proof of the first part, we refer to [1]. The main idea is to introduce an intermediate problem that contains exact linearisations of the terms x_j from the original MIQCQP and additional redundant constraints and therefore, is equivalent to it. Using this auxiliary problem one can prove that RNMDT_p is the relaxation of it and consequentially the relaxation of the problem (14)–(17).

To prove the second part, for every p, let FS-RNMDT_p be a feasible set of the problem RNMDT_p. That is, $(y, x, w, z, \hat{y}, \Delta y, \Delta w) \in \text{FS-RNMDT}_p$ if and only if it satisfies constraints (16)–(17), (18)–(20), (22)–(26) and (28). It should be noted that RNMDT_{p_1} has more variables than RNMDT_{p_2}, therefore, we can not directly compare FS-RNMDT_{p_1} and FS-RNMDT_{p_2} due to a different dimensions. To allow for such comparison, the mapping $M : \text{FS-RNMDT}_{p_1} \to \text{FS-RNMDT}_{p_2}$ is constructed so that every element $(y, x, \hat{y}, w, z, \Delta y, \Delta w) \in \text{FS-RNMDT}_{p_1}$ evaluated in the objective function of RNMDT_{p_1} is equal to

$$M(y, x, \hat{y}, w, z, \Delta y, \Delta w)$$

evaluated in the objective function of RNMDT_{p_2}. Let M be defined as

$$\begin{aligned}
y_i^{\text{RNMDT}_{p_2}} &= y_i^{\text{RNMDT}_{p_1}}, \forall i \in Z_{1,n_1} \\
x_j^{\text{RNMDT}_{p_2}} &= x_j^{\text{RNMDT}_{p_1}}, \forall j \in Z_{1,n_2} \\
w_{i,j}^{\text{RNMDT}_{p_2}} &= w_{i,j}^{\text{RNMDT}_{p_1}}, \forall i,j \in QT \\
z_{j,l}^{\text{RNMDT}_{p_2}} &= z_{j,l}^{\text{RNMDT}_{p_1}}, \forall j \in DS, l \in Z_{p_2,-1} \\
\hat{y}_{i,j,l}^{\text{RNMDT}_{p_2}} &= \hat{y}_{i,j,l}^{\text{RNMDT}_{p_1}}, \forall i,j \in QT, l \in Z_{p_2,-1} \\
\Delta y_j^{\text{RNMDT}_{p_2}} &= \Delta y_j^{\text{RNMDT}_{p_1}} + \sum_{l \in I_{p_1,p_2-1}} 2^l z_{j,l}^{\text{RNMDT}_{p_1}}, \forall j \in DS \\
\Delta w_{i,j}^{\text{RNMDT}_{p_2}} &= \Delta w_{i,j}^{\text{RNMDT}_{p_1}} + \sum_{l \in I_{p_1,p_2-1}} 2^l \hat{y}_{i,j,l}^{\text{RNMDT}_{p_1}}, \forall i,j \in QT
\end{aligned}$$

It is straightforward to verify that the image of this mapping is in the feasibility set FS-RNMDT_{p_2} completing the proof. To prove the final part of the theorem, we can refer to the first item of it indicating that both RNMDT_{p_1} and RNMDT_{p_2} are the relaxations of the original MIQCQP. By the second item of the theorem, RNMDT_{p_2} is a relaxation of RNMDT_{p_1}, it follows that RNMDT_{p_1} is a tighter (or equal) relaxation of the problem (14)–(17) than RNMDT_{p_2} ■

4 The p-Lagrangian relaxation

The combination of Lagrangian relaxation with the RNMDT makes it possible to construct separable mixed-integer problems that retain a weak dual relationship with the original MIQCQP problem. More specifically, it means that one can apply a Lagrangian relaxation to the MIQCQP and subsequently employ RNMDT to the relaxed MIQCQP to obtain a MIP-based relaxation for a given arbitrary value of the precision factor p. Hence, for each fixed p, we obtain a mixed-integer approximation of the LDP, which will be hereinafter called p-Lagrangian dual problem (p-LDP). The procedure to relax one or more constraints with this method is what we refer to as the p-Lagrangian relaxation (p-LR), and the framework

analogous to the Lagrangian decomposition is p-Lagrangian decomposition (p-LD). It is worth highlighting that one can alternatively employ the RNMDT to obtain the RNMDT$_p$ equivalent of the primal problem and then apply the LR to relax the set of constraints (8) and to obtain an identical formulation of the p-LDP.

The p-LDP of the primal RDEM problem can be constructed by employing RNMDT to discretise the variables y_j in the LDP of the primal model represented by the Problem (11). Therefore, this results in a p-LDP that is decomposable by $s \in \mathcal{S}$. Each scenario subproblem can, thus, be stated as

$$\begin{aligned}
\hat{\phi}_p^s(\lambda) = \quad & \text{max. } P^s\Big(\sum_{j\in\mathcal{VI}} I_j^0 x_j^s + \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q_{i,j}^{s,0}\, w_{i,j}^s + \sum_{i\in\mathcal{VC}} C_i^{s,0} y_i^s + \sum_{j\in\mathcal{VI}} I_j^{s,0} x_j^s\Big) \\
& + \sum_{j\in\mathcal{VI}} L^s(\lambda)_j x_j^s \\
\text{s.t.:} \quad & \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q_{i,j}^{s,r} w_{i,j}^s + \sum_{i\in\mathcal{VC}} C_i^{s,r} y_i^s + \sum_{j\in\mathcal{VI}} I_j^{s,r} x_j^s + K^{s,r} \leq 0, \forall r \in \mathcal{R} \\
& y_j^s = (Y_j^{U,s} - Y_j^{L,s})\Big(\sum_{l\in Z_{p,-1}} 2^l z_{j,l}^s + \Delta y_j^s\Big) + Y_j^{L,s}, \forall j \in DS \\
& \omega_{i,j}^s = (Y_j^{U,s} - Y_j^{L,s})\Big(\sum_{l\in Z_{p,-1}} 2^l \hat{y}_{i,j,l}^s + \Delta\omega_{i,j}^s\Big) + y_i^s Y_j^{L,s}, \forall i,j \mid (i,j) \in QT \\
& 0 \leq \Delta y_j^s \leq 2^p, \forall j \in DS \\
& 2^p(y_i^s - Y_i^{U,s}) + Y_i^{U,s}\Delta y_j^s \leq \Delta\omega_{i,j}^s, \forall i,j \mid (i,j) \in QT \\
& \Delta\omega_{i,j}^s \leq 2^p(y_i^s - Y_i^{L,s}) + Y_i^{L,s}\Delta y_j^s, \forall i,j \mid (i,j) \in QT \\
& Y_i^{L,s}\Delta y_j^s \leq \Delta w_{i,j}^s \leq Y_i^{U,s}\Delta y_j^s, \forall i,j \mid (i,j) \in QT \\
& Y_i^{L,s} z_{j,l}^s \leq \hat{y}_{i,j,l}^s \leq Y_i^{U,s} z_{j,l}^s, \forall i,j \mid (i,j) \in QT, l \in Z_{p,-1} \\
& Y_i^{L,s}(1 - z_{j,l}^s) \leq y_i^s - \hat{y}_{i,j,l}^s \leq Y_i^{U,s}(1 - y_{j,l}^s), \forall i,j \mid (i,j) \in QT, l \in Z_{p,-1} \\
& y_i^s \in [Y_i^{L,s}, Y_i^{U,s}], \forall i \in \mathcal{VC} \\
& x_j^s \in \{X_j^L, \ldots, X_j^U\}, \forall j \in \mathcal{VI} \\
& z_{j,l}^s \in \{0,1\}, \forall j \in \mathcal{DS}, l \in Z_{p,-1}, \qquad (29)
\end{aligned}$$

where $\hat{\phi}_p(\lambda) = \sum_s \hat{\phi}_p^s(\lambda)$ and $L^s(\lambda)$ is given by

$$L^s(\lambda) = \begin{cases} \text{vector with elements } [\sum_{s\in\mathcal{S}\setminus\{s'\}} \lambda_j^s], \forall j \in \mathcal{VI} & \text{, if } s = \{s'\} \\ \text{vector with elements } [-\lambda_j^s], \forall j \in \mathcal{VI} & \text{, otherwise.} \end{cases}$$

One appealing feature of the p-LR method is the possibility of regulating the precision of the dual bound by means of the parameter p that controls the precision of the p-LDP. Therefore, if we solve two p-LDPs setting two different values for p, the one with the smaller parameter p will provide a better or equal dual bound compared with the one with the larger p (cf. Theorem 3). The p-LDP is

equivalent to constructing RNMDT$_p$ of the LDP and consists of replacing the dual function ϕ with an over-estimator $\hat{\phi}_p$ with its respective RNMDT-associated auxiliary variables and constraints.

4.1 Convergence of the p-Lagrangian relaxation

The following technical results state the convergence of the sequence $\{\hat{\phi}_{-k}\}_{k=1}^{+\infty}$ of the values generated by the p-Lagrangian dual function $\hat{\phi}$ to the Lagrangian dual function ϕ. We start by referring to epi-convergence [49], which has been shown to be the ideal tool to study the convergence and approximation of optimisation problems, especially in settings considering duality as a coordination framework. We refer the reader to [49, Chapter 7] for a detailed study of the properties of epi-convergence, opting to list only the properties relevant to our purposes. In order to use this theory, we need the addition of an auxiliary set of variables to the DEM problem along with the framing of the problem as a minimisation. This can be achieved by reformulating the DEM as:

$$P \ : \ v\left(\{-f^s\}_{s\in S}, \{C^s\}_{s\in S}\right) = \min_{(x,y,w)} \left[\sum_{s\in S} \left(-f^s\right)(x,(y^s,w^s)) : (x,y^s,w^s) \in C^s\right]$$

where, for $s \in S$,

$$C^s = \left\{(x,(y^s,w^s)) \mid g^{s,r}\left(x,(y^s,w^s)\right) \leq 0,\, r \in R,\, w^s_{i,j} = y^s_i y^s_j,\, x \in X,\, y^s \in Y^s\right\},$$

and

$$f^s(x,(y^s,w^s)) = P^s\Big(\sum_{j\in\mathcal{VI}} I^0_j x_j + \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q^{s,0}_{i,j} w^s_{i,j} + \sum_{i\in\mathcal{VC}} C^{s,0}_i y^s_i\Big),$$
$$g^{s,r}(x,(y^s,w^s)) = \sum_{i\in\mathcal{VC}}\sum_{j\in\mathcal{VC}} Q^{s,r}_{i,j} w^s_{i,j} + \sum_{i\in\mathcal{VC}} C^{s,r}_i y^s_i + \sum_{j\in\mathcal{VI}} I^{s,r}_j x_j + K^{s,r},$$

while X and Y^s represents the box constraints (16) and (17) for integer and continuous variables, respectively.

Recall that the constraints (25) and (26) in Definition 2 provide a relaxation of the product of two continuous variables and are known as McCormick envelopes [15]. Denote these by S_{R_p}, which comprises the variables $(y, x, \hat{y}, z, \Delta y, \Delta w)$ satisfying the constraints (23) to (26).

For the purpose of convergence analysis, we would like the relaxation to reside in the same variable space of $(x,(y,w)) \in X \times Y^s \times \mathbb{R}^{|\mathcal{VC}|^2} \subseteq \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|+|\mathcal{VC}|^2}$ for $s \in S$. The current space of variables consists of $(y, x, \hat{y}, z, \Delta y, \Delta w)$. Analogously to the derivations in the proof of Theorem 3, let us define the mapping $M : (y, x, \hat{y}, z, \Delta y, \Delta w) \to \left(x', \left(y', w'\right)\right)$ to be

$$x' = x \quad \text{and} \quad y'_j = (Y^U_j - Y^L_j)\Big(\sum_{l\in Z_{p,-1}} 2^l z_{j,l} + \Delta y_j\Big) + Y^L_j \text{ with}$$
$$w'_{i,j} = (Y^U_j - Y^L_j)\left(\sum_{l\in Z_{p,-1}} 2^l \hat{y}_{i,j,l} + \Delta w_{i,j}\right) + y_i Y^L_j.$$

Then, we constrain $(x,(y,w))$ to the set $M\left(S_{R_p}\right)$. The RNMDT$_p$ relaxation of the problem P for the fixed value of $p<0$ can be re-formulated as

$$P_{R_p} \;:\; v\left(\left\{-f^s\right\}_{s\in S},\left\{C^s_{R_p}\right\}_{s\in S}\right)=\min_{(x,y,w)}\left[\sum_{s\in S}\left(-f^s\right)\left(x,\left(y^s,w^s\right)\right):\left(x,y^s,w^s\right)\in C^s_{R_p}\right]$$

where, for $s\in S$,

$$C^s_{R_p}=\left\{(x,(y^s,w^s))\in M\left(S_{R_p}\right)\mid g^{s,r}\left(x,\left(y^s,w^s\right)\right)\leq 0,\, r\in R,\, x\in X,\, y^s\in Y^s\right\}.$$

The advantage of framing the approximation in this form is that the additional integer variables z required to describe the McCormick envelopes are embedded in the constraint set $M(S_{R_p})$, as they simply constitute additional variables required for the description of the McCormick envelopes. Notice that the objective is not perturbed at all in this formulation. We have the following due to the properties of the McCormick envelopes [15].

Proposition 4 *We have P_{R_p} a relaxation of $P_{R_{p-1}}$ and of P as*

$$M\left(S_{R_{p-1}}\right)\subseteq M\left(S_{R_p}\right)$$

where

$$\left\{(x,(y^s,w^s))\mid w^s_{i,j}=y^s_i y^s_j\right\}=\bigcap_{p<0}M\left(S_{R_p}\right).$$

Proof. The proof of the first part is equivalent to that of Theorem 3 in Section 3. To prove the second part, first, notice that P_{R_p}, $\forall p<0$, is a reformulated equivalent of NMDT relaxation presented in [16]. As demonstrated in [16, Property 3], the NMDT relaxation converges to the primal problem P as $p\to-\infty$, which in turn implies that $\lim_{p\to-\infty} w^s_{i,j}=y^s_i y^s_j$. ■

We now relate the convergence of the RNMDT$_p$ relaxations to the notion of epi-convergence.

Definition 3 *Let $f_k:\mathbb{R}^n\to\mathbb{R}\cup\{+\infty\}$ be an extended sequence of real-valued functions. We say that $\{f_k\}$ epi-converges to f and write e-$\lim_k f_k=f$ if and only if $\lim_k \operatorname{epi} f_k=\operatorname{epi} f$, where $\operatorname{epi} f_k=\{(x,\alpha)\mid f_k(x)\leq\alpha\}$ and $\operatorname{epi} f=\{(x,\alpha)\mid f(x)\leq\alpha\}$.*

The limit of sets in Definition 3 is understood to be in the Kuratowski-Painleve sense. That is, (1) all accumulation points of the subsequence $(x_{k_l},\alpha_{k_l})\in\operatorname{epi} f_{k_l}$ are in $\operatorname{epi} f$ (i.e., $(x_{k_l},\alpha_{k_l})\to(x,\alpha)\in\operatorname{epi} f$) and (2) for all $(x,\alpha)\in\operatorname{epi} f$ there exists a sequence $(x_k,\alpha_k)\in\operatorname{epi} f_k$ with $(x_k,\alpha_k)\to(x,\alpha)$. Recall that a function f is lower semi-continuous if and only if $\operatorname{epi} f$ is closed.

One of the reasons that epi-convergence lends itself to the analysis of approximations is that any optimisation problem of the form $v(g_k,C_k)=\min.\{g_k(x)\mid x\in C_k\}$ can be represented as an infimum of an extended real-valued function $f_k=g_k+\delta_{C_k}$ where $\delta_{C_k}(x)=0$ if and only if $x\in C_k$, and $+\infty$, otherwise. Thus, the general properties of approximating optimisation problems can be framed in terms of the behavior of infima of extended real valued functions that are fully capable of modelling constraints sets containing a fixed number of integer variables.

Consider the optimisation problem $P:\max.\{f(x)\mid x\in C\}$. Recall that $P_{R_2}:\max.\{f_{R_2}(x)\mid x\in C_{R_2}\}$ is a tighter relaxations of P than $P_{R_1}:\max.\{f_{R_1}(x)\mid x\in C_{R_1}\}$ if and only if

1. $f_{R_1}(x) \geq f_{R_2}(x) \geq f(x)$ for all $x \in C$ and
2. $C \subseteq C_{R_2} \subseteq C_{R_1}$.

Note that $v(f_{R_2}, C_{R_2}) = \max_x \left(f_{R_2}(x) + \delta_{C_{R_2}}(x)\right) \leq v(f_{R_1}, C_{R_1})$. To place the study of such relaxations into the framework of epi-convergence, we must consider the equivalent minimization problem, so

$$v(-f_{R_2}, C_{R_2}) = \min\left(-f_{R_2}(x) + \delta_{C_{R_2}}(x)\right) \geq v(-f_{R_1}, C_{R_1}),$$

because $-f_{R_2} + \delta_{C_{R_2}} \geq -f_{R_1} + \delta_{C_{R_1}}$. Clearly, a sequence of tighter convex relaxations P_{R_p} with $p = -k$ leads to a non-decreasing sequence of objective and value functions that are associated with a sequences $g_k = -f_{R_k} + \delta_{C_{R_k}}$ of optimisation problems that are monotonically non-decreasing, i.e., $g_{k+1} \geq g_k$, for all k. From [49, Proposition 7.4 and 7.15], we know that such sequences epi-converge to the closure of their pointwise supremum. Moreover uniformly convergent sequences also epi-converge. This enables epi-convergence to be invoked to study the convergence of our relaxations. Later, we will show that the associated sequence of convex dual problems also epi-converges in the strong sense of uniform convergence on bounded sets.

In what follows, we do not assume that the set C is convex nor connected. So it could contain integer constraints on variables. In most integer programming problems it is best to posit the integer constraints in the constraint set C and hence also in the relaxations C_{R_k}, in that they do not pose a barrier to convergence, for the theory of epi-convergence applies to extended real valued, lower semi-continuous or closed functions, such as (for instance) $\left\{\delta_{C_{R_k}}\right\}_k$. We know that the relaxations epi-converge (as they are monotonically improve as we tighten) but we need to establish what problem it converges to. Notice the role of Proposition 4 in ensuring that the RNMDT$_p$ relaxation satisfies the condition 2 in Proposition 5.

Proposition 5 *Consider the optimisation problem $P : \min\{-f(x) \mid x \in C\}$ involving a proper, closed, lower semi-continuous function f and a closed set S. Suppose we have a sequence of tighter relaxations of P, given by $P_{R_k} : min.\{-f_{R_k}(x) \mid x \in C_{R_k}\}$ where*

1. *$-f_{R_k}(x) \leq -f_{R_{k+1}}(x) \leq -f(x)$ for all $x \in C$ and $\operatorname{epi}(-f) = \overline{\bigcap_k \operatorname{epi}(-f_k)}$ with*
2. *$C \subseteq C_{R_{k+1}} \subseteq C_{R_k}$ and $\overline{\bigcap_k C_{R_k}} = C$ (which is equivalent to $\bigcap_k C_{R_k} = C$ when each C_{R_k} is closed) .*

Then $\left\{g_k = -f_{R_k} + \delta_{C_{R_k}}\right\}_{k=1}^{\infty}$ epi-converges to $g = -f + \delta_C$.

Proof. Let $g = -f + \delta_C$. Note that we have for any x that

$$g_k(x) = -f_{R_k}(x) + \delta_{C_{R_k}}(x) \leq -f(x) + \delta_C(x) = g(x)$$

Let $B_\delta(\bar{x}) = \{x \mid ||x - \bar{x}|| \leq \delta\}$ where $||\cdot||$ is any norm. Therefore, for any $\delta > 0$ and $\bar{x} \in \operatorname{dom} f \cap C$ we have (cf. [49, Exercise 7.3])

$$\begin{aligned} e\text{-}\limsup_{k\to\infty} g_k(\bar{x}) &= \sup_{\delta>0} \limsup_{k\to\infty} \inf_{x\in B_\delta(\bar{x})} g_k(x) \\ &\leq \sup_{\delta>0} \inf_{x\in B_\delta(\bar{x})} g(x) \leq g(\bar{x}) = -f(\bar{x}) + \delta_C(\bar{x}), \end{aligned} \tag{30}$$

with the last inequality following from $\inf_{x\in B_\delta(\bar{x})} g(x) \leq g(\bar{x})$ for all $\delta > 0$. By [49, Theorem 7.46] we always have (for any extended real-valued sequence)

$$\begin{aligned} e\text{-}\liminf_{k\to\infty} (-f_{R_k})(\bar{x}) + e\text{-}\liminf_{k\to\infty} \delta_{C_{R_k}}(\bar{x}) &\leq e\text{-}\liminf_{k\to\infty} \left(-f_{R_k} + \delta_{C_{R_k}}\right)(\bar{x}) \\ &= e\text{-}\liminf_{k\to\infty} g_k(\bar{x}). \end{aligned} \tag{31}$$

Under our assumption,

$$e - \liminf_{k\to\infty}(-f_{R_k})(\bar{x}) = -f(\bar{x})$$

and

$$e - \liminf_{k\to\infty} \delta_{C_{R_k}}(\bar{x}) = \delta_C(\bar{x}),$$

because $C \subseteq C_{R_{k+1}} \subseteq C_{R_k}$ and $\overline{\bigcap_k C_{R_k}} = C$ implies $\{\delta_{C_{R_k}}\}_{k=1}^{\infty}$ is a monotone non-decreasing sequence of functions with

$$\operatorname{epi}\left[\overline{\sup_k \delta_{C_{R_k}}}\right] = \overline{\operatorname{epi}\delta_{\cap_k C_{R_k}}} = \operatorname{epi}\delta_{\overline{\cap_k C_{R_k}}} = \operatorname{epi}\delta_C.$$

Hence (30) and (31) implies

$$\begin{aligned} -f(\bar{x}) + \delta_C(\bar{x}) \leq e\text{-}\liminf_{k\to\infty} \left(-f_{R_k} + \delta_{C_{R_k}}\right)(\bar{x}) &= e\text{-}\liminf_{k\to\infty} g_k(\bar{x}) \\ \leq e\text{-}\limsup_{k\to\infty} g_k(\bar{x}) &\leq -f(\bar{x}) + \delta_C(\bar{x}) \end{aligned}$$

and, thus, equality ensues. ■

The following result shows that the study of perturbations of convex optimisations problems and how this affects their Lagrangian relaxations is essentially the analysis of the interaction of epi-convergence with conjugation. Note that this result does not assume $\{f_k\}_{k=1}^{\infty}$ is a sequence of convex functions.

Theorem 6 ([44, Corollary 20]) *If the functions $\{f_k\}$ and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are proper, lower semi-continuous with f bounded below on bounded sets and $\{f_k\}$ equi-hypercoercive (in the sense that $\lim_{\|x\|\to\infty} \frac{f_k(x)}{\|x\|} = +\infty$ uniformly in k) then*

$$e\text{-}\lim_k f_k = f \quad \textit{implies} \quad e\text{-}\lim_k f_k^* = f^*.$$

Now, let us show the connection between the p-Lagrangian dual function $\hat{\phi}_p$ and the conjugate of the primal problem represented by RNMDT_p relaxation P_{R_p} in the same way of the developments presented in Section 2. For that, let $f^s : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|+|\mathcal{VC}|^2} \to \mathbb{R}$ and $g^{s,r} : \mathbb{R}^{|\mathcal{VI}|+|\mathcal{VC}|+|\mathcal{VC}|^2} \to \mathbb{R}$, $\forall s \in S$ and $\forall r \in R$, be continuous twice-differentiable functions. By $x^{-s'}$ we denote the vector reduced by one dimension when removing the s' component, which, in turn, represents the reference scenario in the formulation of the RDEM. The associated Lagrangian relaxations is given by:

$$
\begin{aligned}
&-\hat{\phi}_p(\lambda)\\
&=\inf_{(x,(y,w))}\left[\sum_{s\in S}(-f^s)(x^s,(y^s,w^s))+\sum_{s\in S\setminus\{s'\}}(-\lambda^s)^\top\left(x^{s'}-x^s\right):(x^s,(y^s,w^s))\in C^s_{R_p}\right]\\
&=-\sup_{(x,(y,w))}\left[\sum_{s\in S\setminus\{s'\}}(\lambda^s)^\top\left(x^{s'}-x^s\right)-\sum_{s\in S}\left((-f^s)(x^s,(y^s,w^s))+\delta_{C^s_{R_p}}(x^s,(y^s,w^s))\right)\right]\\
&=-\sup_{(x^{-s'},(y^{-s'},w^{-s'}))}\sum_{s\in S\setminus\{s'\}}\left[(-\lambda^s)^\top\ x^s-\left(-f^s(x^s,(y^s,w^s))+\delta_{C^s_{R_p}}(x^s,(y^s,w^s))\right)\right]\\
&\qquad-\sup_{(x^{s'},(y^{s'},w^{s'}))}\left[\left(\lambda^{s'}\right)x^{s'}-\left(-f^{s'}\left(x^{s'},\left(y^{s'},w^{s'}\right)\right)+\delta_{C^{s'}_{R_p}}\left(x^{s'},\left(y^{s'},w^{s'}\right)\right)\right)\right]\\
&=-\left(\sum_{s\in S\setminus\{s'\}}\left(-f^s+\delta_{C^s_{R_p}}\right)\right)^*\left(-\lambda^{-s'},(0,0)\right)-\left(-f^{s'}+\delta_{C^{s'}_{R_p}}\right)^*\left(\lambda^{s'},(0,0)\right),
\end{aligned}
$$

where $\lambda^{s'}=\sum_{s\in S\setminus\{s'\}}\lambda^s$.

Proposition 7 *Let f be the objective function of the primal MIQCQP problem and f^* be its optimal value. Let ϕ be the Lagrangian dual function of the correspondent LDP and the functions $\hat{\phi}_p:\mathbb{R}^{|S|-1\times|\mathcal{VI}|}\to\mathbb{R}$ be the p-Lagrangian dual function of the correspondent p-LDP, where $p\in\mathbb{Z}^-$. Then the following statements are true.*

1. *$\hat{\phi}_p$ is convex, $\forall p<0$.*
2. *$\hat{\phi}_p\geq\phi,\ \forall\ p<0$.*
3. *$\hat{\phi}_p\geq f^*,\ \forall\ p<0$.*
4. *$\hat{\phi}_{p_2}\geq\hat{\phi}_{p_1}\ \forall p_1<p_2<0$.*
5. *$\inf_\lambda\hat{\phi}_{p_2}(\lambda)\geq\inf_\lambda\hat{\phi}_{p_1}(\lambda)\geq\inf_\lambda\phi(\lambda),\forall p_1<p_2<0$.*

Proof. The proof of the first item of the proposition is analogous to Proposition 1. The second item of the proposition is a direct consequence of Theorem 3. To prove 3, we use Proposition 2 which states that $\phi\geq f^*$. The second item of the proposition gives that $\hat{\phi}_p\geq\phi$. It follows by transitivity that $\hat{\phi}_p\geq f^*$. Analogous to 2, the fourth part is a direct consequence of the Theorem 3. And the final part of the proposition is a consequence of the second and fourth items combined. ■

From Proposition 7, we have that $-\hat{\phi}_p(\lambda)\leq v\left(\{-f^s\}_{s\in S},\left\{C^s_{R_p}\right\}_{s\in S}\right)$, which implies that $\inf_\lambda\hat{\phi}_p(\lambda)\geq v\left(\{f^s\}_{s\in S},\left\{C^s_{R_p}\right\}_{s\in S}\right)$, where

$$
\inf_\lambda\hat{\phi}_p(\lambda)=\inf_{\lambda^{-s'}}\left(\sum_{s\in S\setminus\{s'\}}\left(-f^s+\delta_{C^s_{R_P}}\right)\right)^*\left(-\lambda^{-s'},0\right)+\inf_{\lambda^{s'}}\left(-f^{s'}+\delta_{C^{s'}_{R_p}}\right)^*\left(\lambda^{s'},0\right).
$$

When we apply this duality to the problem P_{R_k} to obtain the p-LDP with $p=-k$ we will denote the corresponding dual function by $\hat{\phi}_{-k}$. In what follows, the functions $-\sum_{s\in S}f^s_{R_k}+\delta_{C^s_{R_k}}$ are equi-hypercoercive due to the boundedness assumption on $C^s_{R_k}$. Thus, we can state the epi-convergence of $\hat{\phi}_p$ as $p\to-\infty$.

Corollary 8 *Consider the optimisation problem P_S involving a proper-closed, lower semi-continuous functions $\{f^s\}_{s\in S}$ and a closed sets $\{C^s\}_{s\in S}$. Suppose we have a sequence of tighter relaxations of the optimisation problem given by*

$$P_{R_k} : \min\left\{ -\sum_{s\in S} f^s_{R_k}(x,(y^s,w^s)) \mid (x,(y^s,w^s)) \in C^s_{R_k},\ s\in S \right\}.$$

We assume

1. *$f^s_{R_k}(x,(y^s,w^s)) \geq f^s_{R_{k+1}}(x,(y^s,w^s)) \geq f^s(x,(y^s,w^s))$ for all $(x,(y^s,w^s)) \in C^s_{R_k}$, $s\in S$ and $\operatorname{epi}(-f^s) = \overline{\bigcap_k \operatorname{epi}\left(-f^s_{R_k}\right)}$ with*
2. *for some $M>0$ we have $C^s \subseteq C^s_{R_{k+1}} \subseteq C^s_{R_k} \subseteq B_M(0)$, $s\in S$ and $\overline{\bigcap_k C^s_{R_k}} = C^s$ (which is equivalent to $\bigcap_k C^s_{R_k} = C^s$ when each $C^s_{R_k}$ is closed).*

Let $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ be the associated sequence of scenario wise relaxation of the problems P_{R_k}. Then $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ epi-converges to $\phi(\cdot)$ where $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ and $\phi(\cdot)$ are all convex, proper and closed.

Proof. As the functions $-\sum_{s\in S} f^s_{R_k} + \delta_{C^s_{R_k}}$ are equi-hypercoercive in the following (due to the boundedness assumption on $C^s_{R_k}$) we have, from combining Proposition 5 and 4, that $\left\{-\sum_{s\in S} f^s_{R_k} + \delta_{C^s_{R_k}}\right\}$ epi-converges to $-\sum_{s\in S} f^s + \delta_{C^s}$ and so by Theorem 6, and the fact that $\hat{\phi}_{-k}$ (and ϕ) are obtained via conjugation, we have $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ epi-converges to $\phi(\cdot)$. ■

Furthermore, as the Lagrangian dual functions are convex (cf. Proposition 7), we can finally arrive to the convergence result we were aiming at.

Proposition 9 *Consider the optimisation problem P_S involving a proper-closed, lower semi-continuous functions $\{f^s\}_{s\in S}$ and a closed sets $\{C^s\}_{s\in S}$. Suppose we have a sequence of tighter approximations of the optimisation problem given by $P_{R_k} : \min\left\{-\sum_{s\in S} f^s_{R_k}(x,(y^s,w^s)) \mid (x,(y^s,w^s)) \in C^s_{R_k},\ s\in S\right\}$. Posit the assumption as in Corollary 8. Then $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ converges uniformly on bounded sets.*

Proof. We utilise [49, Theorem 7.17], that says that when ϕ is a convex, lower semi-continuous function on $\mathbb{R}^n$, when $\operatorname{dom}\phi$ has nonempty interior, epi-convergence of $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ to $\phi(\cdot)$ is equivalent to $\left\{\hat{\phi}_{-k}\right\}$ converges uniformly on all compact subsets of $\operatorname{dom}\phi$ that does not contain boundary points of $\operatorname{dom}\phi$. As $\left\{C^s_{R_k}\right\}_{k=1}^{\infty}$ are contained in a bounded set then $\operatorname{dom}\phi$ is the whole space so combining [49, Theorem 7.17] and Corollary 8 we immediately obtain uniformly convergence on bounded sets. ■

4.2 Motivating example: part 2

We revisit the motivating example in Section 2.1. By applying the $p-$LR, we obtain p-LDP formulation

$$
\begin{aligned}
\max._{w,\Delta w,z,\Delta x,\hat{x},x} \; & w_{1,2} + \lambda(x_1 + 2x_2 - 1) \\
\text{s.t.:} \; & w_{1,2} = 2^{-1}\hat{x}_{1,2,-1} + v_{1,2} \\
& x_2 = 2^{-1}z_{2,-1} + \Delta x_2 \\
& 0 \leq \hat{x}_{1,2,-1} \leq z_{2,-1} \\
& 0 \leq x_1 - \hat{x}_{1,2,-1} \leq 1 - z_{2,-1} \\
& 2^{-1}(x_1 - 1) + \Delta x_2 \leq \Delta w_{1,2} \leq 2^{-1}x_1 \\
& 0 \leq \Delta w_{1,2} \leq \Delta x_2 \\
& z_{2,-1} \in \{0, 1\} \\
& x_1, x_2 \in [0, 0.75] \\
& \Delta x_2 \in [0, 2^{-1}].
\end{aligned}
\tag{32}
$$

In this example, the variable x_2 is discretised with a precision $p = -1$ and the Lagrangian multiplier value is fixed to -0.l00. Solving the resulting problem with Gurobi 9.0 [28], we obtain the p-Lagrangian dual bound $\hat{\phi}_{-1}(-0.100) = 0.500$, which is valid since it is greater than 0.125. Notice, however, that this is not a tight bound since it was obtained considering an arbitrary Lagrangian multiplier. This bound value could be strengthened solving the p-Lagrangian dual problem employing a nonsmooth optimisation method. In this example, employing a subgradient method [29], the optimal Lagrangian multiplier $\lambda = -0.250$ and correspondent value of the dual objective function $\phi(-0.250) = 0.250$ would be obtained. In the next section, we discuss the solution methods for LDPs. Solving the p-LDP with a precision $p = -1$ and Lagrangian multiplier value fixed to 0.250 provides with a bound $\hat{\phi}_{-1}(-0.250) = 0.313$.

To obtain a tighter bound $\hat{\phi}_p$, it is necessary to choose more carefully the parameter p. If p and λ are set respectively to -10 and -0.250, the p-LDP bound obtained is $\hat{\phi}_{-10}(-0.250) = 0.250$, the same as the optimal LDP solution value. In Section 5 we present a solution method for p-LDPs that simultaneously finds appropriate values for the Lagrangian multipliers and precision factor p.

5 A p-Lagrangian decomposition algorithm

Algorithms to solve a nonconvex problem, such as the MIQCQP being considered, usually rely on computing (and successively improving) primal and dual bounds. In the proposed setting, dual bounds - upper (lower) bounds, in a maximisation (minimisation) problem - are computed by finding the optimal value for the p-LDP for a given set of Lagrangian multipliers, which in turn have their accuracy regulated by the value of the precision factor p. The primal bound - a lower (upper) bound for a maximisation (minimisation) problem - can be obtained as the value of the objective function for a primal feasible solution.

The algorithm for solving the p-LDP combines two strategies: i) search for optimal Lagrangian multipliers and ii) tighten the RNMDT$_p$ as the iterations progress by decreasing the parameter p, thus gradually decreasing the UB. However, this method might have a significant disadvantage associated with a rapid increase in the number of binary variables that are added to the LDP, since all discretised variables are expanded using the same number of partitions. This makes it harder

to compute the solution for the p-LDP due to the accelerated increase in the number of variables and constraints. To mitigate this adverse effect, the algorithm proposed in this section is inspired by the dynamic precision algorithm in [1]. The major benefit of this algorithm is that it only increases the precision (and thus the number of auxiliary variables and constraints) of the variables that will potentially improve (i.e., tighten) the relaxation and which are dynamically chosen at each iteration. Concurrently, the method convergence is ensured by periodically increasing the precision of the variables that remained unchanged (that is, that have not been selected in a predefined number of past iterations).

Therefore, the single precision parameter p is replaced with a vector $p_j^s, \forall s \in S$ and $j \in DS$, for each scenario-based p-LDP subproblem (29). Each entry is associated with the number of partitions that is used to increase the precision of the variable y_j^s for all $s \in S$ and $j \in DS$. The procedure for solving p-LDP is summarised in Algorithm 1.

Algorithm 1 p-LR algorithm

Step 0. Set $p_j^s = -1\ \forall s \in S$ and $j \in DS$. Let $UB = +\infty$, $iteration = 0$, and assume given starting Lagrangian multipliers, N_1, and N_2.
Step 1. Update iteration $k = k + 1$
Step 2. Compute a dual bound (UB) by finding an optimal value of the p-LDP for the fixed values of the Lagrangian multipliers and obtain a primal feasible solution (LB) using the Lagrangian-based heuristic.
Step 3. Update the Lagrangian multipliers values using a nonsmooth optimization algorithm.
Step 4.
if a stop condition of type 1 is met **then**
  if $iteration + 1$ is not a multiple of N_2 **then**
    Step 5. Rank the combinations of indexes $s, j\ \forall s \in S$ and $j \in DS$ using f_{rank} and for the
    first N_1 combinations s, j ranked by f_{rank} set $p_j^s = p_j^s - 1$.
  else
    Step 5. Let $\overline{p}_j^s = \max_{s,j}\{p_j^s\}$. Set $p_j^s = p_j^s - 1\ \forall j, s \mid p_j^s = \overline{p}_j^s$.
  end if
else return to Step 2.
end if
Step 6.
if a stop condition of type 2 is met **then** stop.
else return to Step 1.
end if

The variables for which the precision will be increased (i.e., p will be decreased) are chosen by ranking them using the function

$$f_{rank}(s,j) = \sum_{r \in R} \left\{ |Q_{j,j}^{s,r}(w_{j,j}^s - {y_j^s}^2)| + 2 \sum_{(r,i)|i>j|(i,j)\in QT} |Q_{i,j}^{s,r}(w_{i,j}^s - y_i^s y_j^s)| \right\}. \tag{33}$$

The term N_1 is an auxiliary parameter corresponding to the number of ranked variables that are selected to have their precision increased, while N_2 is the period (in the number of iterations) of the periodic increase in precision of the variables that remained unchanged (in the last N_2 iterations).

The Lagrangian-based heuristics mentioned in Step 2 of Algorithm 1 can be any method capable of generating primal feasible solutions. In this study, the heuristics employed at each iteration of the p-Lagrangian decomposition algorithm consist of two core elements. First, using the optimal values of the integer decision variables $\bar{x}_j^s$ for the p-LDP calculated at the iteration k, the averaged values are computed as follows.

$$x_j^{avg} = \frac{\sum\limits_{s \in S} \bar{x}_j^s}{|S|}, \forall j \in \mathcal{VI}.$$

Then, the optimal value of the variables $\bar{y}_i^s$, calculated at the iteration k, are used to find a feasible solution of the primal RDEM problem when integer variables x_j^s are fixed to the nearest integer (rounded) values x_j^{avg} for all $j \in \mathcal{VI}, s \in S$. This can be achieved by solving to optimality the primal problem with fixed integer variables and using the values $\bar{y}_{j,s}$ as a warm start for continuous variables. The objective function value of the primal problem is further used as a lower bound (LB) if the value obtained is smaller than the existing LB. Hereinafter, we refer to this heuristic to obtain primal feasible solutions as the Lagrangian-based heuristic. It is worth mentioning that the same strategy is also employed in the dynamic-precision algorithm for solving MIQCQP problems using RNDMT (dp-RNMDT) proposed in [1]. Yet, this strategy is likely to be inefficient in more general settings. Thus, when possible, appropriate knowledge about the problem structure should be exploited for generating primal feasible solutions.

The nonsmooth optimisation algorithm mentioned in Step 3 is employed in the Lagrangian multipliers update. In this paper, we used the bundle method. For more details on the multipliers update algorithms please refer to the Appendix A. Notice that there are two types of stopping criteria in the Algorithm 1. A stop condition of type 1 in Step 4 is a stop criterion for the p-LDP multipliers update algorithm (i.e., bundle method). A stop condition of type 2 is a condition to stop the whole algorithm, e.g., time limit, iteration limit, or a threshold on the relative or absolute gap computed using incumbent primal and dual bounds.

Aiming to improve the convergence behaviour of the p-LR algorithm when using the bundle method (in Step 3), we modified the initialisation of the bundle method parameters. As a starting value for the centre of mass λ_0^{centre}, we considered the final value of the Lagrangian multipliers from the previous iteration of the p-LR algorithm, with the the expectation that it would reduce the number of *serious steps* required until convergence (for details on the bundle method implementation, please refer to the Appendix A).

6 Computational experiments

In this section, we present numerical results for randomly generated nonconvex RDEM problems solved with Algorithm 1 (p-LD).We relied on randomly generated instances that replicate the separable structure of two-stage stochastic programming instances. The computational efficiency of the p-LD was compared with Gurobi's (version 9.0.0) spatial branching algorithm (Full-space) [28] and the direct employment of dp-RNMDT. All the experiments were designed using the Julia (version 1.3.1) language [7] and the commercial solver Gurobi (version 9.0.0). The code is located at the GitHub repository `github.com/gamma-opt/p-Lagrangian_`

`relaxation.jl`. The code was run on Triton, Aalto University's high-performance computing cluster, on a Dell PowerEdge C6420. The node has two Intel Xeon Gold 6148 20-core processors and 192GB of DDR4-2667 memory.

6.1 Design of the experiments

The three methods were applied to solve the collections of randomly generated instances. Each set of instances contained the RDEM problems with 5, 10, 15, 20 and 25 scenarios, represented in three sizes (small, medium and large) as described in Table 1. The quadratic matrices $Q_{i,j}^{s,r}$ for all $s \in \mathcal{S}$, $r \in \{0\} \cup \mathcal{R}$ were randomly generated with 80% density, a number that was arbitrarily chosen to match that observed in instances from [57]. The generation process of each group was replicated five times using different random seeds forming a total of 75 instances. In all experiments, we considered the first scenario as the reference index for formulating the NAC.

The p-LD Algorithm 1 was implemented in two versions utilising sequential and parallel computing. The parallelisation was based on the scenario subproblems of the p-LDP and for each instance, the number of processes utilised for parallel computing was equal to the number of scenarios in the instance.

Table 1: Instance problems dimensions (per scenario)

Instance size	**Number of continuous variables**	**Number of integer variables**	**Number of constraints**
Small (S)	20	5	45
Medium (M)	30	10	60
Large (L)	40	15	75

Table 2 presents the parameter values for the p-LD including the parameter values for the dynamic precision-based algorithm and for LD. The termination tolerance for the dynamic precision-based algorithm was used to control the gap between primal and dual bounds. In turn, the termination tolerance for LD was based on the pair-wise differences between the Lagrangian dual function values considering the last three iterations (see Section 5 and Appendix A respectively for more details on the termination criteria). It is worth mentioning that both methods are somewhat sensitive to the initial parameter values in terms of computational performance and that their initial values were determined based on preliminary experiments on the smaller instances.

6.2 Numerical results

Table 3 presents the average values of the optimality gaps achieved within the predefined time limit and average running times (i.e., the time required by the algorithms to converge or if it was terminated). The optimality gaps were calculated as the relative difference between the UB attained using one of the methods to solve the dual problem and LB obtained by generating the primal feasible solution

Table 2: Algorithm parameters

dp-RNMDT parameters	
Iteration limit	100
Time limit (s)	3600
Termination tolerance (relative)	0.1%
N_1	10% of the total number of the continuous variables
N_2	5
p-LD parameters	
Iteration limit	100
Termination tolerance (absolute)	100
β	0.7
c_k for all k	0.05
Initial values of λ_j^s, $\forall j \in \mathcal{VI}$ and $\forall s \in \mathcal{S}$	0

using the Lagrangian-based heuristic. In the rows highlighted with the symbol "*" we discarded the random seeds for which the heuristic method found a LB providing a relative gap higher than 500% for the p-LD. The values are shown for all three methods, i.e., Solver (employing Gurobi to solve the RDEM problem), dp-RNMDT and p-LD. The p-LD algorithm was executed sequentially and also parallelising the solution of the p-LDP. To highlight the potential improvement that the parallelisation would provide, we present in the last column the time taken for the parallel p-LD to perform the same number of iterations observed in the sequential execution of p-LD.

Table 3: Average results of the experiments

	Instance		**Solver**		**dp-RNMDT**		**p-LD**		**Parallel p-LD**
	Size	Scenarios	% gap	Time (s)	% gap	Time (s)	% gap	Time (s)	Time (s)
	S	5	128.645	3600.0	0.176	3600.0	0.228	3600.0	2497.8
	S	10	182.345	3600.0	0.183	2819.3	0.171	1285.2	662.2
	S	15	175.251	3600.0	0.303	3005.3	0.181	2806.9	1310.0
	S	20	204.715	3600.0	0.387	3600.0	0.078	1814.2	491.2
*	S	25	669.646	3600.0	0.460	3600.0	0.083	2569.0	661.1
	M	5	261.103	3600.0	0.408	3009.3	0.418	1637.2	820.1
	M	10	292.999	3600.0	0.767	3600.0	0.453	1820.0	467.6
	M	15	448.488	3600.0	1.175	3600.0	0.264	3292.2	763.6
	M	20	298.273	3600.0	1.422	3600.0	0.380	3600.0	730.2
*	M	25	355.598	3600.0	1.486	3600.0	0.234	3062.0	734.5
	L	5	364.983	3600.0	0.628	3330.2	1.013	2563.8	1073.3
	L	10	395.671	3600.0	1.826	3600.0	0.629	3600.0	1196.7
*	L	15	540.458	3600.0	2.205	3600.0	0.520	3124.3	654.1
*	L	20	1047.548	3600.0	3.220	3600.0	0.442	3600.0	710.6
	L	25	4779.444	3600.0	4.329	3600.0	1.785	3600.0	781.6

From the numerical results, one can conclude that the proposed method performed better than the commercial solver and dp-RNMDT. Solving the full-space problem never converged within the time limit set and never attained gap better than roughly 130%, highlighting how challenging these problems are under a computational standpoint even for relatively small instances. Comparing the dp-RNMDT and sequential p-LD, one can notice that for most instances, with exception of those with 5 scenarios, utilising the p-LD method allows for obtaining relative gaps up to six-times smaller, as is the case for the medium instance

with 25 scenarios. Moreover, the advantage of the dp-RNMDT considering the gap attained for the instances with 5 scenarios is partially due to the heuristic method employed for the LB generation, which happened to be able in some of the experiments to generate better primal bounds from the solution obtained for dp-RNMDT. Nevertheless, another takeaway is the superior performance of p-LD in terms of solution time for two-thirds of the instances. Furthermore, employing the parallelised version of p-LD yields improvements in solution time of up to approximately 5 times, as is the case for large instances with 25 scenarios.

Table 4 provides more information on the performance of dp-RNMDT and sequential p-LD. The numerical results for all of the instances were organised in groups by random seeds and in each group, two columns are associated with each of the techniques. The first column "UB" reports the relative difference between the UB generated by the two methods which was calculated as

$$\frac{\mathrm{UB} - min(\mathrm{UB}_{\text{dp-RNMDT}}, \mathrm{UB}_{p\text{-LD}})}{min(\mathrm{UB}_{\text{dp-RNMDT}}, \mathrm{UB}_{p\text{-LD}})} \times 100, \tag{34}$$

where "$\mathrm{UB}_{\text{dp-RNMDT}}$" and "$\mathrm{UB}_{p\text{-LD}}$" are the UB generated by dp-RNMDT and p-LD respectively and "UB" is the the UB generated by the method considered in the column. Notice that, for the same random seed, the entry with value 0.00% indicates that the method found the best of the UB found for that instance, while the other entries show how much worse (relatively larger) the other UB obtained are.

We highlight (with the bold font) the cells corresponding to the method having superior (or equal) performance. The second column "Time (s)" represents the time required by the method to converge, if the cell contains a number, or if it was terminated due to the time limit of 3600s, which is indicated with the letter "T". Analogously, with the bold font, we emphasise the cases when the correspondent method converged faster. The last row in both columns summarizes the information, showing the number of the cases when the method generated equal or better bound for the column "UB" and the number of the instances when the method converged for the column "Time". The cells in bold font indicate which method performed better in terms of generating bounds. Similarly, we highlight with the bold font which method converged more frequently. As one can observe from the table in each random seed-based group sequential p-LD presented a superior performance for both criteria.

Table 4: Comparison analysis of the dp-RNMDT and p-LD convergence

Instance		SEED 1				SEED 2				SEED 3				SEED 4				SEED 5			
		dp-RNMDT		p-LD		dp-RNMDT		p-LD		dp-RNMDT		p-LD		dp-RNMDT		p-LD		dp-RNMDT		p-LD	
Size	Scenarios	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)	UB	Time (s)
S	5	**0.00%**	T	0.08%	T	**0.00%**	T	**0.00%**	T	**0.00%**	T	**0.00%**	T	**0.00%**	T	**0.00%**	T	**0.00%**	T	0.02%	T
S	10	**0.00%**	252.7	0.19%	**197**	0.02%	T	**0.00%**	**1238.4**	0.09%	T	**0.00%**	**839.1**	**0.00%**	3020	**0.00%**	**444.9**	**0.00%**	T	0.04%	T
S	15	**0.00%**	1313.7	0.08%	**437.7**	0.02%	2874	**0.00%**	**1825.5**	0.59%	T	**0.00%**	T	0.12%	T	**0.00%**	T	0.08%	T	**0.00%**	T
S	20	**0.00%**	T	0.07%	**439.2**	0.11%	T	**0.00%**	**454.8**	0.75%	T	**0.00%**	T	0.15%	T	**0.00%**	**733.6**	0.27%	T	**0.00%**	T
S	25	0.39%	T	**0.00%**	**1213.9**	**0.00%**	T	**0.00%**	T	0.57%	T	**0.00%**	**3208.9**	0.27%	T	**0.00%**	**2049.3**	0.26%	T	**0.00%**	T
M	5	**0.00%**	**618.4**	**0.00%**	**150.9**	**0.00%**	T	0.10%	T	**0.00%**	3597	0.12%	**473.6**	**0.00%**	T	0.21%	**125.7**	**0.00%**	T	0.08%	T
M	10	**0.00%**	T	0.14%	**564.7**	**0.00%**	T	0.01%	T	**0.00%**	T	0.16%	**553.2**	**0.00%**	T	0.46%	**405.3**	0.19%	T	**0.00%**	T
M	15	0.16%	T	**0.00%**	T	0.58%	T	**0.00%**	T	0.07%	T	**0.00%**	T	0.70%	T	**0.00%**	T	**0.00%**	T	0.08%	**941**
M	20	1.35%	T	**0.00%**	T	0.36%	T	**0.00%**	T	0.51%	T	**0.00%**	T	0.86%	T	**0.00%**	T	**0.00%**	T	0.19%	**1422.1**
M	25	1.15%	T	**0.00%**	**2344.3**	1.61%	T	**0.00%**	T	0.78%	T	**0.00%**	T	0.79%	T	**0.00%**	T	0.33%	T	**0.00%**	**2013.4**
L	5	**0.00%**	T	3.71%	T	**0.00%**	2186.7	**0.00%**	**505.8**	**0.00%**	T	1.31%	T	**0.00%**	T	0.78%	T	**0.00%**	T	0.45%	**330.7**
L	10	1.11%	T	**0.00%**	T	1.49%	T	**0.00%**	T	1.06%	T	**0.00%**	T	0.94%	T	**0.00%**	T	0.11%	T	**0.00%**	**3339.6**
L	15	1.90%	T	**0.00%**	T	1.37%	T	**0.00%**	T	0.90%	T	**0.00%**	**1582.9**	0.55%	T	**0.00%**	T	1.00%	T	**0.00%**	**2132.3**
L	20	2.01%	T	**0.00%**	T	1.98%	T	**0.00%**	**2288.7**	1.17%	T	**0.00%**	T	0.75%	T	**0.00%**	T	0.68%	T	**0.00%**	**2985.6**
L	25	2.81%	T	**0.00%**	T	2.37%	T	**0.00%**	T	2.61%	T	**0.00%**	T	2.96%	T	**0.00%**	T	0.15%	T	**0.00%**	T
		7	3	**9**	**7**	5	2	**13**	**5**	4	1	**12**	**5**	5	1	**12**	**5**	6	0	**9**	**7**

6.2.1 Performance profiles

To provide a structured comparison between the p-LD and dp-RNMDT, we present performance profiles based on [24]. Let P be the set of all the problem instances and A be the set of the algorithms used to solve the problem instances $p \in P$, i.e. dp-RNMDT, p-LD and parallelised p-LD. Let $t_{p,a}$ be computing time required by the algorithm $a \in A$ to solve the problem $p \in P$. For all $p \in P$ and $a \in A$ let the time performance ratio be defined as

$$r^t_{p,a} = \frac{t_{p,a}}{min_{a \in A}\{t_{p,a}\}}.$$

Let $P^t_{\tau,a} = \{p \in P : r^t_{p,a} \leq \tau\}$ and, for every $a \in A$, let the overall assessment of the time performance be defined as

$$\rho^t_a(\tau) = \frac{|P^t_{\tau,a}|}{|A|}.$$

Analogously, let $ub_{p,a}$ be the UB achieved by the algorithm $a \in A$ for the problem $p \in P$. Let the UB performance ratio be defined as

$$r^{ub}_{p,a} = \frac{ub_{p,a}}{min_{a \in A}\{ub_{p,a}\}}.$$

Let $P^{ub}_{\tau,a} = \{p \in P : r^{ub}_{p,a} \leq \tau\}$ and let the UB performance assessment be defined as

$$\rho^{ub}_a(\tau) = \frac{|P^{ub}_{\tau,a}|}{|A|}.$$

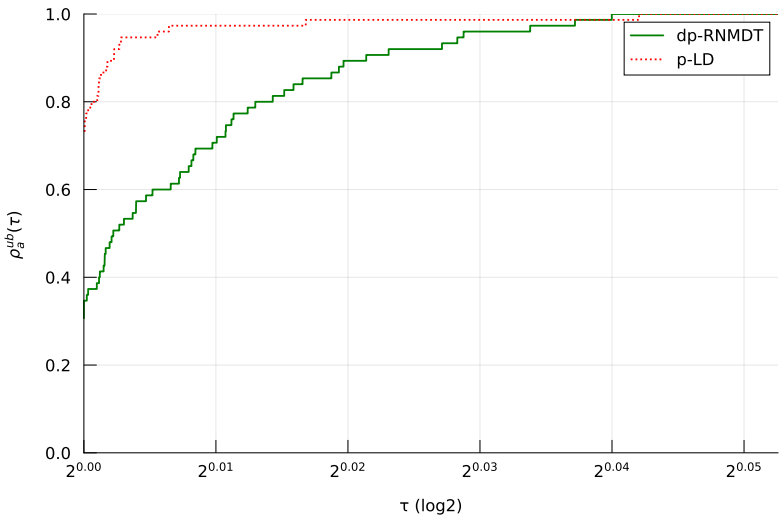

Fig. 1: Performance profile based on the upper bounds

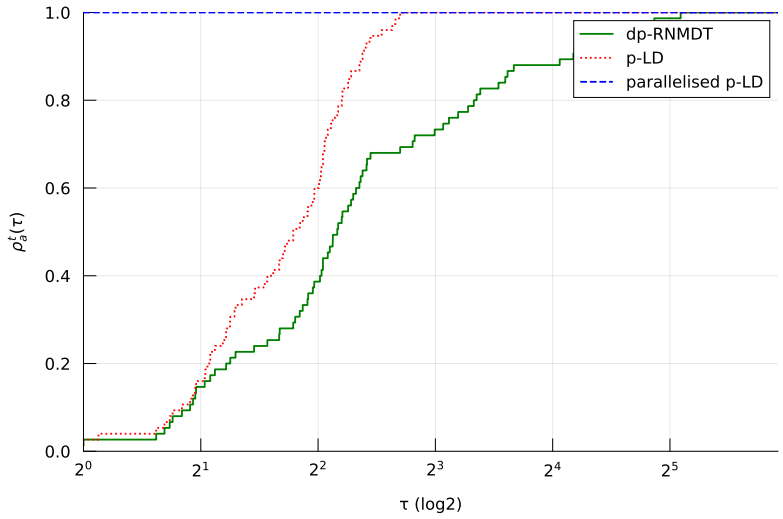

Fig. 2: Performance profiles based on the convergence time

Figures 1 and 2 present the dual (upper) bound and time performance profiles, respectively. In Figure 2, the parallelised version of the p-LD is excluded because the parallelisation did not considerably affect the quality of the bound generated. In both figures, the horizontal axes are plotted on a logarithmic scale.

As one can observe from the Figures 1 and 2, p-LD demonstrated superior performance when compared to dp-RNMDT in both bound generated and time performance criteria. In addition, Figure 2 indicates that the computational performance superiority of p-LD can be reinforced by means of parallelisation techniques.

7 Conclusions

In this paper, we proposed a novel decomposition method named p-Lagrangian relaxation, which consists of an alternative framework to achieve decomposition for nonconvex MIQCQP problems. The core idea of the p-Lagrangian relaxation is to combine two techniques: Lagrangian relaxation and RNMDT. Lagrangian relaxation is a broadly known decomposition framework commonly applied to solving large-scale constrained optimisation problems with exploitable structure, which is the case whenever the relaxation of the complicating constraints results in a decomposable version of the original problem. Nevertheless, the nonconvex nature of the primal problem may lead to substantial issues, in specific, the necessity of solving nonconvex problems when evaluating the Lagrangian dual function. Therefore, we address this issue by applying a mixed-integer based relaxation technique, named RNMDT. Consequently, the primal problem is converted into a decomposable mixed-integer problem with significantly easier tractable Lagrangian dual function.

The values of the Lagrangian multipliers along with the value of precision parameter p of the RNMDT allow for controlling the quality of the relaxation.

Therefore, we proposed a new algorithm named p-Lagrangian decomposition inspired by the dynamic-precision based method developed in [1] combined with the bundle method for updating the Lagrangian multipliers. Additionally, the decomposable structure of the Lagrangian dual problem is amenable to parallelisation, which can significantly enhance computational performance.

The numerical experiments suggest that the p-Lagrangian decomposition has considerable advantages over the commercial solver Gurobi in obtaining dual bounds within the predefined time limit. The experiments also indicate that significant savings in computational time may be achieved when introducing parallel computing.

Despite the promising numerical results concerning its performance, the p-Lagrangian decomposition algorithm has two important shortcomings. The first is the dependence on a Lagrangian-based heuristic for generating primal feasible solutions which is likely to be inefficient. The second issue relates to the duality gap arising from the mixed-integer nature of the primal problem combined with the imprecision of the RNMDT relaxation.

Therefore, future research should consider efficient ways to incorporate the p-Lagrangian decomposition within a branch-and-bound setting, which could potentially mitigate both issues. In particular, we believe that further advancement of the proposed method could be achieved by considering augmented Lagrangian instead [19]. Furthermore, the employment of the p-Lagrangian relaxation for bounds generation in a branch-and-bound framework could bring new light to the classic approach for solving nonconvex mixed-integer non-linear problems such as a combination of the symbolic reformulation and spatial branch-and-bound algorithm [52], thus enhancing the computational efficiency of the method. Additionally, the p-Lagrangian relaxation could be employed for models arising from equilibrium problems [57], which naturally yield large-scale MIQCQPs with separable structure.

References

1. Andrade, T., Oliveira, F., Hamacher, S., Eberhard, A.: Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. Journal of Global Optimization **73**(4), 701–722 (2019). DOI 10.1007/s10898-018-0728-9. URL `https://doi.org/10.1007/s10898-018-0728-9`
2. Andrade, T., Ribas, G., Oliveira, F.: A strategy based on convex relaxation for solving the oil refinery operations planning problem. Industrial & Engineering Chemistry Research **55**(1), 144–155 (2016). DOI 10.1021/acs.iecr.5b01132. URL `https://pubs.acs.org/doi/10.1021/acs.iecr.5b01132`
3. Bao, X., Sahinidis, N.V., Tawarmalani, M.: Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. Mathematical Programming **129**(1), 129–157 (2011). DOI 10.1007/s10107-011-0462-2. URL `http://link.springer.com/10.1007/s10107-011-0462-2`
4. Barahona, F., Anbil, R.: The volume algorithm: producing primal solutions with a subgradient method. Mathematical Programming **87**(3), 385–399 (2000). DOI 10.1007/s101070050002. URL `http://link.springer.com/10.1007/s101070050002`
5. Berthold, T., Heinz, S., Vigerske, S.: Extending a CIP Framework to Solve MIQCPs. In: J. Lee, S. Leyffer (eds.) Mixed Integer Nonlinear Programming, The IMA Volumes in Mathematics and its Applications, pp. 427–444. Springer, New York, NY (2012). DOI 10.1007/978-1-4614-1927-3_15
6. Bertsekas, D.P.: Convex Optimization Algorithms. No. 4 in Optimization and computation series. Athena Scientific, Belmont, Massachusetts (2015)

7. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A Fresh Approach to Numerical Computing. SIAM Review **59**(1), 65–98 (2017). DOI 10.1137/141000671. URL https://epubs.siam.org/doi/10.1137/141000671
8. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering. Springer New York, New York, NY (2011). DOI 10.1007/978-1-4614-0237-4. URL http://link.springer.com/10.1007/978-1-4614-0237-4
9. Birge, J.R., Qi, L.: Computing Block-Angular Karmarkar Projections with Applications to Stochastic Programming. Management Science **34**(12), 1472–1479 (1988). DOI 10.1287/mnsc.34.12.1472. URL http://pubsonline.informs.org/doi/abs/10.1287/mnsc.34.12.1472
10. Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F.: Combining Progressive Hedging with a Frank–Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer Programming. SIAM Journal on Optimization **28**(2), 1312–1336 (2018). DOI 10.1137/16M1076290. URL https://epubs.siam.org/doi/10.1137/16M1076290
11. Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Oliveira, F.: A parallelizable augmented lagrangian method applied to large-scale non-convex-constrained optimization problems. Mathematical Programming **175**(1-2), 503–536 (2019). DOI 10.1007/s10107-018-1253-9. URL https://doi.org/10.1007/s10107-018-1253-9
12. Burer, S., Letchford, A.N.: Non-convex mixed-integer nonlinear programming: A survey. Surveys in Operations Research and Management Science **17**(2), 97 – 106 (2012). DOI https://doi.org/10.1016/j.sorms.2012.08.001. URL http://www.sciencedirect.com/science/article/pii/S1876735412000037
13. Camerini, P.M., Fratta, L., Maffioli, F.: On improving relaxation methods by modified gradient techniques. In: M.L. Balinski, P. Wolfe (eds.) Nondifferentiable Optimization, Mathematical Programming Studies, pp. 26–34. Springer, Berlin, Heidelberg (1975). DOI 10.1007/BFb0120697. URL https://doi.org/10.1007/BFb0120697
14. Carøe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. Operations Research Letters **24**(1-2), 37–45 (1999). DOI 10.1016/S0167-6377(98)00050-9. URL https://linkinghub.elsevier.com/retrieve/pii/S0167637798000509
15. Castro, P.M.: Tightening piecewise McCormick relaxations for bilinear problems. Computers & Chemical Engineering **72**, 300–311 (2015). DOI 10.1016/j.compchemeng.2014.03.025. URL http://www.sciencedirect.com/science/article/pii/S0098135414001069
16. Castro, P.M.: Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems. Journal of Global Optimization **64**(4), 765–784 (2016). DOI 10.1007/s10898-015-0342-z. URL https://doi.org/10.1007/s10898-015-0342-z
17. Castro, P.M.: Spatial branch and bound algorithm for the global optimization of MIQCPs. In: Z. Kravanja, M. Bogataj (eds.) Computer Aided Chemical Engineering, *26th European Symposium on Computer Aided Process Engineering*, vol. 38, pp. 523–528. Elsevier (2016). DOI 10.1016/B978-0-444-63428-3.50092-8. URL http://www.sciencedirect.com/science/article/pii/B9780444634283500928
18. Cheney, E.W., Goldstein, A.A.: Newton's method for convex programming and Tchebycheff approximation. Numerische Mathematik **1**(1), 253–268 (1959). DOI 10.1007/BF01386389. URL http://link.springer.com/10.1007/BF01386389
19. Cordova, M., de Oliveira, W., Sagastizabal, C.: Revisiting Augmented Lagrangian Duals. Mathematical Programming (2020). URL http://www.optimization-online.org/DB_HTML/2020/03/7709.html
20. Corporation, I.: Ibm ilog cplex optimization studio v 12.8.0 (2018)
21. Courant, R.: Variational methods for the solution of problems of equilibrium and vibrations. Bulletin of the American Mathematical Society **49**(1), 1–23 (1943). DOI 10.1090/S0002-9904-1943-07818-4. URL http://www.ams.org/journal-getitem?pii=S0002-9904-1943-07818-4
22. Dentcheva, D., Römisch, W.: Duality gaps in nonconvex stochastic optimization. Mathematical Programming **101**(3), 515–535 (2004). DOI 10.1007/s10107-003-0496-1. URL http://link.springer.com/10.1007/s10107-003-0496-1
23. Ding, T., Bo, R., Li, F., Sun, H.: A Bi-Level Branch and Bound Method for Economic Dispatch With Disjoint Prohibited Zones Considering Network Losses. IEEE Transactions on Power Systems **30**(6), 2841–2855 (2015). DOI 10.1109/TPWRS.2014.2375322
24. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**(2), 201–213 (2002). DOI 10.1007/s101070100263. URL http://link.springer.com/10.1007/s101070100263

25. Fisher, M.L.: The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science **27**(1), 1–18 (1981). DOI 10.1287/mnsc.27.1.1. URL http://pubsonline.informs.org/doi/abs/10.1287/mnsc.27.1.1
26. Geoffrion, A.M.: Generalized Benders decomposition. Journal of Optimization Theory and Applications **10**(4), 237–260 (1972). DOI 10.1007/BF00934810. URL http://link.springer.com/10.1007/BF00934810
27. Guignard, M.: Lagrangian Relaxation, pp. 845–860. Springer US, Boston, MA (2013). DOI 10.1007/978-1-4419-1153-7_1168. URL https://doi.org/10.1007/978-1-4419-1153-7_1168
28. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020). URL http://www.gurobi.com
29. Held, M., Wolfe, P., Crowder, H.P.: Validation of subgradient optimization. Mathematical Programming **6**(1), 62–88 (1974). DOI 10.1007/BF01580223. URL http://link.springer.com/10.1007/BF01580223
30. Hestenes, M.R.: Multiplier and gradient methods. Journal of Optimization Theory and Applications **4**(5), 303–320 (1969). DOI 10.1007/BF00927673. URL http://link.springer.com/10.1007/BF00927673
31. Jeroslow, R.C.: There cannot be any algorithm for integer programming with quadratic constraints. Operations Research **21**(1), 221–224 (1973). DOI 10.1287/opre.21.1.221. URL https://doi.org/10.1287/opre.21.1.221
32. Kelley Jr., J.E.: The Cutting-Plane Method for Solving Convex Programs. Journal of the Society for Industrial and Applied Mathematics **8**(4), 703–712 (1960). DOI 10.1137/0108053. URL http://epubs.siam.org/doi/10.1137/0108053
33. Kianfar, K.: Maximizing profit in a supply chain by considering advertising and price elasticity of demand. Computers & Industrial Engineering **135**, 265–274 (2019). DOI 10.1016/j.cie.2019.06.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0360835219303389
34. Kolodziej, S., Castro, P.M., Grossmann, I.E.: Global optimization of bilinear programs with a multiparametric disaggregation technique. Journal of Global Optimization **57**(4), 1039–1063 (2013). DOI 10.1007/s10898-012-0022-1. URL https://doi.org/10.1007/s10898-012-0022-1
35. Lazzeroni, P., Repetto, M.: Optimal planning of battery systems for power losses reduction in distribution grids. Electric Power Systems Research **167**, 94–112 (2019). DOI 10.1016/j.epsr.2018.10.027. URL https://linkinghub.elsevier.com/retrieve/pii/S0378779618303432
36. Lemaréchal, C.: An algorithm for minimizing convex functions. In: IFIP Congress, pp. 552–556 (1974)
37. Li, X., Tomasgard, A., Barton, P.I.: Nonconvex Generalized Benders Decomposition for Stochastic Separable Mixed-Integer Nonlinear Programs. Journal of Optimization Theory and Applications **151**(3), 425–454 (2011). DOI 10.1007/s10957-011-9888-1. URL http://link.springer.com/10.1007/s10957-011-9888-1
38. Løkketangen, A., Woodruff, D.L.: Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. Journal of Heuristics **2**(2), 111–128 (1996). DOI 10.1007/BF00247208. URL https://doi.org/10.1007/BF00247208
39. Marsten, R.E., Hogan, W.W., Blankenship, J.W.: The Boxstep Method for Large-Scale Optimization. Operations Research **23**(3), 389–405 (1975). DOI 10.1287/opre.23.3.389. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.23.3.389
40. Dept. of Mathematical Sciences, C.U.: Couenne: a user's manual. URL https://www.coin-or.org
41. Misener, R., Floudas, C.A.: Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. Mathematical Programming **136**(1), 155–182 (2012). DOI 10.1007/s10107-012-0555-6. URL https://doi.org/10.1007/s10107-012-0555-6
42. Oliveira, F., Gupta, V., Hamacher, S., Grossmann, I.E.: A Lagrangean decomposition approach for oil supply chain investment planning under uncertainty with risk considerations. Computers & Chemical Engineering **50**, 184–195 (2013). DOI 10.1016/j.compchemeng.2012.10.012. URL http://www.sciencedirect.com/science/article/pii/S0098135412003249
43. optimization, for Systems in the Big-Data Era: Theory, C., Applications: Optimization and Control for Systems in the Big-Data Era: Theory and Applications, vol. 252. Springer International Publishing (2017)

44. Penot, J.P., Zalinescu, C.: Continuity of the legendre–fenchel transform for some variational convergences. Optimization **53**, 549–562 (2004). DOI 10.1080/02331930412331329533
45. Pietrzykowski, T.: An Exact Potential Method for Constrained Maxima. SIAM Journal on Numerical Analysis **6**(2), 299–304 (1969). DOI 10.1137/0706028. URL http://epubs.siam.org/doi/10.1137/0706028
46. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. Optimization pp. 283–298 (1969). URL https://ci.nii.ac.jp/naid/20000922074/en/
47. Rockafellar, R.T.: Lagrange multipliers and optimality. SIAM Review **35**(2), 183–238 (1993). DOI 10.1137/1035044. URL https://doi.org/10.1137/1035044
48. Rockafellar, R.T., Wets, R.J.B.: Scenarios and Policy Aggregation in Optimization Under Uncertainty. Mathematics of Operations Research **16**(1), 119–147 (1991). DOI 10.1287/moor.16.1.119. URL http://pubsonline.informs.org/doi/abs/10.1287/moor.16.1.119
49. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis, vol. 317. Springer Science & Business Media (2009). Google-Books-ID: JSREAAAAQBAJ
50. Santos, M., Silva, E., Finardi, E., Gonçalves, R.: Solving the short term operating planning problem of hydrothermal systems by using the progressive hedging method. 16th Power Systems Computation Conference, PSCC 2008 (2008)
51. Singh, M.K., Kekatos, V.: Natural Gas Flow Solvers using Convex Relaxation. IEEE Transactions on Control of Network Systems pp. 1–1 (2020). DOI 10.1109/TCNS.2020.2972593. URL https://ieeexplore.ieee.org/document/8988210/
52. Smith, E.M.B., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Computers & Chemical Engineering **23**(4-5), 457–478 (1999). DOI 10.1016/S0098-1354(98)00286-5. URL http://www.sciencedirect.com/science/article/pii/S0098135498002865
53. Strub, O., Brandinu, S., Lerch, D., Schaller, J., Trautmann, N.: A Three-phase Approach to an Enhanced Index-tracking Problem with Real-life Constraints. The Engineering Economist **64**(3), 227–253 (2019). DOI 10.1080/0013791X.2019.1619887. URL https://doi.org/10.1080/0013791X.2019.1619887
54. Tappenden, R., Richtárik, P., Büke, B.: Separable approximations and decomposition methods for the augmented Lagrangian. Optimization Methods and Software **30**(3), 643–668 (2015). DOI 10.1080/10556788.2014.966824. URL http://www.tandfonline.com/doi/full/10.1080/10556788.2014.966824
55. The Optimization Firm, L.: Baron user manual v. 2019.12.7 (2019). URL https://http://www.minlp.com
56. Veliz, F.B., Watson, J.P., Weintraub, A., Wets, R.J.B., Woodruff, D.L.: Stochastic optimization models in forest planning: a progressive hedging solution approach. Annals of Operations Research **232**, 259–274 (2015). DOI 10.1007/s10479-014-1608-4
57. Virasjoki, V., Siddiqui, A., Oliveira, F., Salo, A.: Utility-scale energy storage in an imperfectly competitive power sector. Energy Economics p. 104716 (2020). DOI 10.1016/j.eneco.2020.104716. URL http://www.sciencedirect.com/science/article/pii/S0140988320300554
58. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006). DOI 10.1007/s10107-004-0559-y. URL https://doi.org/10.1007/s10107-004-0559-y
59. Zangwill, W.I.: Non-Linear Programming Via Penalty Functions. Management Science **13**(5), 344–358 (1967). DOI 10.1287/mnsc.13.5.344. URL http://pubsonline.informs.org/doi/abs/10.1287/mnsc.13.5.344
60. Zhao, X., Luh, P.: New Bundle Methods for Solving Lagrangian Relaxation Dual Problems. Journal of Optimization Theory and Applications **113**(2), 373–397 (2002). DOI 10.1023/A:1014839227049. URL http://link.springer.com/10.1023/A:1014839227049

A Solution methods for Lagrangian dual problems

This section describes the approaches for solving the LDP in Section 2. In general, the dual problem is nonsmooth. Nevertheless, it is convex, and there is extensive literature on how to solve such problems by updating the Lagrangian multipliers.

Held and Karp [**?**] and Held et al. [29] proposed the classical approach, which became known as the subgradient method. Improvements of this method were proposed by Camerini et al. [13] and Fisher [25]. An alternative method that presents better convergence properties is the cutting-plane method proposed by Cheney and Goldstein [18] and Kelley [32]. An improvement of this method is presented by Marsten et al. [39]. Other methods include the Volume algorithm [4] and the bundle method [36,60] that typically present more stable convergence than cutting-planes methods. In this study, we employed the bundle method, since preliminary experiments showed that it provided a good trade-off between convergence behaviour and ease of implementation.

There are multiple variations of the bundle method. Our implementation followed its classical variant, as presented in [6]. The idea of the bundle method lies in iterating the argument λ_{k+1} as follows

$$\lambda_{k+1} \in \arg\min_{\lambda}\{F_k(\lambda) + p_k(\lambda)\}. \tag{35}$$

The F_k is a cutting-plane approximation to f and is defined as

$$F_k(\lambda) = \max\ \{\phi(\lambda_0) + (\lambda - \lambda_0)'\phi'(\lambda_0), ..., \phi(\lambda_k) + (\lambda - \lambda_k)'\phi'(\lambda_k)\}, \tag{36}$$

while $p_k(\lambda)$ is given by

$$p_k(\lambda) = \frac{1}{2c_k}||\lambda - \lambda_k^{centre}||^2, \tag{37}$$

where $\lambda_k^{centre} \in \{\lambda_i, i \leq k\}$ is a proximal centre. The computation of the new proximal centre λ_{k+1}^{centre} depends on the results of a specified test indicating whether "sufficient progress" has been made or not. This *serious step condition* can be stated as

$$\lambda_{k+1}^{centre} = \begin{cases} \lambda_{k+1} & \text{if } \phi(\lambda_k^{centre}) - \phi(\lambda_{k+1}) \geq \beta\delta_k \ (\textit{serious step}) \\ \lambda_k^{centre} & \text{if } \phi(\lambda_k^{centre}) - \phi(\lambda_{k+1}) < \beta\delta_k \ (\textit{null step}), \end{cases} \tag{38}$$

where $\beta \in (0, 1)$, and $\delta_k = \phi(\lambda_k^{centre}) - (F_k(\lambda_{k+1}) + p_k(\lambda_{k+1}))$.

As a termination criterion, $\lambda_{k+1} = \lambda_k^{centre}$ is used. However, unless ϕ is polyhedral, the finite termination is unlikely. Therefore, it is common to stop the algorithm when the difference between $\phi(\lambda_{k+1})$ and $\phi(\lambda_k)$ remains within a certain tolerance.

B List of acronyms and abbreviations

BD	Benders decomposition
BnB	Branch and bound
DEM	Deterministic equivalent model
DWD	Dantzing-Wolfe decomposition
GBD	Generalised Benders decomposition
LB	Lower bound
LD	Lagrangian decomposition
LDP	Lagrangian dual problem
LR	Lagrangian relaxation
MILP	Mixed-integer linear program
MIP	Mixed-integer programming
MIQCQP	Mixed-integer quadratically constrained quadratic programming
NAC	Non-anticipativity conditions
NGBD	Nonconvex generalised Benders decomposition
NMDT	Normalised multiparametric disaggregation technique
PH	Progressive hedging
p-LD	p-Lagrangian decomposition
p-LR	p-Lagrangian relaxation
QCQP	Quadratically constrained quadratic programming
RDEM	Reformulated deterministic equivalent model
RNMDT	Reformulated normalised multiparametric disaggregation technique
UB	Upper bound