Journal of Global Optimization (2022) 84:43–76
https://doi.org/10.1007/s10898-022-01138-y

The *p*-Lagrangian relaxation for separable nonconvex MIQCQP problems

Tiago Andrade[1] · Nikita Belyak[2] · Andrew Eberhard[3] · Silvio Hamacher[4] · Fabricio Oliveira[2]

Received: 11 December 2020 / Accepted: 21 January 2022 / Published online: 22 February 2022

Abstract
This paper presents a novel technique to compute Lagrangian bounds for nonconvex mixed-integer quadratically constrained quadratic programming problems presenting a separable structure (i.e., a separable problems) 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 to the 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 decomposition, is a decomposition method that combines Lagrangian decomposition with mixed-integer programming-based relaxations. These relaxations are obtained using the *reformulated normalised multiparametric disaggregation technique* 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 the proposed method presents significant reductions in computational time when compared with a previously proposed techniques in the literature and the direct employment of a commercial solver. Moreover, our computational experiments show that the employment of a simple heuristic can recover solutions with small duality gaps.

Keywords Lagrangian decomposition · Reformulated normalised multiparametric disaggregation technique · Nonconvex mixed integer quadratically constrained quadratic programs · Decomposition methods

✉ Fabricio Oliveira
fabricio.oliveira@aalto.fi

Tiago Andrade
tiago.andrade@psr-inc.com

Nikita Belyak
nikita.belyak@aalto.fi

1 PSR, Rio de Janeiro, Brazil

2 Aalto University, Espoo, Finland

3 RMIT University, Melbourne, Australia

4 Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

1 Introduction

In this paper, we focus on nonconvex mixed-integer quadratically constrained quadratic programming (MIQCQP) models that can be made separable with the employment of a Lagrangian relaxation-based approach [16]. Specifically, we focus on problems that possess a block angular structure that allows for such a separation, as those presented by deterministic equivalent formulations of two-stage stochastic programming models (DEM) [10]. To introduce the notation used hereinafter, let us define the problem DEM as

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

$$\text{s.t.:} \quad \sum_{i\in VC} \sum_{j\in VC} Q_{i,j}^{s,r} y_i^s y_j^s + \sum_{i\in VC} C_i^{s,r} y_i^s + \sum_{j\in VI} I_j^{s,r} x_j + K^{s,r} \leq 0, \tag{1}$$

$$\forall s \in S, \forall r \in R \tag{2}$$

$$y_i^s \in [Y_i^{L,s}, Y_i^{U,s}], \forall s \in S, \forall i \in VC \tag{3}$$

$$x_j \in \{X_j^L, \ldots, X_j^U\}, \forall j \in VI, \tag{4}$$

where S is the set of scenarios, P^s is the probability of the scenario $s \in S$, R is the set of the constraint indices for each scenario, VI (VC) is the set of integer (continuous) variables indices, $Q^{s,r}$ are symmetric matrices for all $s \in S$, $r \in \{0\} \cup R$. The parameters $I_j^{s,r}$ ($C_i^{s,r}$) correspond to linear coefficients associated with the integer (continuous) variables and $K^{s,r}$ represent constants. Variables $x_j, \forall j \in VI$, can assume any integer value between its bounds X_j^L and X_j^U and variables y_i^s, $i \in VC$ and $s \in S$, assume any continuous value between $Y_i^{L,s}$ and $Y_i^{U,s}$. Note that this boundedness assumption, which is indeed satisfied in many applications, will become essential in our developments.

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 [21]. Recent papers utilising MIQCQP models include problems such as planning optimal battery management for the reduction of power losses [41], the enhanced index-tracking problem formulation [59], the operational planning of refineries [2], profit maximisation in a three-level supply chain [39], storage investment planning [63] and solution approaches for a gas flow problem with general setups [57], to mention only a few of the potential applications that could benefit from the developments presented in this paper.

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 $Q^{s,r}$ is positive semidefinite for all $s \in S$ and $r \in \{0\} \cup R$, being nonconvex otherwise. The latter (nonconvex) is the case addressed in this paper.

In principle, MIQCQP models can be solved by available open-source and commercial solvers such as recent releases of Gurobi [34], Couenne [26] or Baron [61]. However, these solvers still lack 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 [14]. These can be generally divided into exact and heuristic methods. While the former can guarantee the convergence to the global optimum, the latter can generally only offer local optimality guarantee 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, 46] 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 performance.

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 models, a spatial BnB is typically used, employing (convex) relaxations of the nonconvex terms that are the source of the nonlinearity. Pursuing this standpoint, Castro [19] presented a spatial BnB with relaxations relying on normalised multiparametric disaggregation technique for solving nonconvex MIQCQP problems. Ding et al. [29] presented 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. [6] developed MIQCQP problem solver based on the combination of constraint programming and branch-and-cut algorithm exploiting cutting planes to tighten the relaxations. Billionnet et al. [9] developed the mixed-integer quadratic convex reformulation (MIQCR) method, which consists of a (convex) reformulation approach for nonconvex mixed-integer quadratic programming problems with linear constraints, which is also embedded within a BnB framework.

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.

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

$$\begin{aligned}
\text{RDEM}: \max. \quad & \sum_{s\in S} P^s \left(\sum_{j\in VI} I_j^0 x_j^s + \sum_{i\in VC}\sum_{j\in VC} Q_{i,j}^{s,0} y_i^s y_j^s + \sum_{i\in VC} C_i^{s,0} y_i^s \right) \\
\text{s.t.:} \quad & \sum_{i\in VC}\sum_{j\in VC} Q_{i,j}^{s,r} y_i^s y_j^s + \sum_{i\in VC} C_i^{s,r} y_i^s + \sum_{j\in VI} I_j^{s,r} x_j^s + K^{s,r} \le 0, && (5)\\
& \forall s \in S, \forall r \in R && (6)\\
& x_j^s \in \{X_j^L, \ldots, X_j^U\}, \forall s \in S, \forall j \in VI && (7)\\
& x_j^{s'} - x_j^s = 0, \forall s \in S \setminus \{s'\}, \forall j \in VI && (8)\\
& \text{and } (3),
\end{aligned}$$

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 DEM problem with an exposed block-angular structure [10, 11]. Consequently, one can obtain $|S|$ 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. [48], as the authors report better computational results in a similar context. Nonetheless, the developments presented are not dependent on the specific choice for representing NACs.

In classical linear programming, the three most common decomposition frameworks 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 method 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 [32] proposed the generalised Benders decomposition (GBD) based on BD to decompose convex nonlinear programming problems. Later, Li, Tomasgard, and Barton [43] 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 Sect. 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 LD. The idea of employing RNMDT to devise a MIP approximation of the resulting LD 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 LD, 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 LD. Section 3 describes the RNMDT relaxation. The p-Lagrangian relaxation of the MIQCQP problem is presented in Sect. 4, followed by technical results that describe its convergence behaviour. An iterative algorithm for solving p-Lagrangian relaxation problems is described in Sect. 5. Furthermore, numerical experiments are presented in Sect. 6, where the contributions of this paper are tested on randomly generated instances. Finally, in Sect. 7 we draw conclusions and provide directions for further research.

2 Background

Lagrangian relaxation is a bounding technique for solving a given arbitrary optimisation problem that has important applications for nonconvex problems such as MILP problems [16, 33]. However, methods based on Lagrangian duality cannot be straightforwardly applied to nonconvex problems, as discussed in what follows.

First, let us formally provide a definition for a relaxation which, although rather restrictive, will suffice for the developments hereinafter. Consider the following problem $P : \max.\{f(x) : x \in C\}$ where $f : \mathbb{R}^n \to \mathbb{R}$ and $C \subseteq \mathbb{R}^n$ and the problem $P_R : \max.\{f_R(x) : x \in C_R\}$ where $f_R : \mathbb{R}^n \to \mathbb{R}$ and $C_R \subseteq \mathbb{R}^n$.

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

1. $f_R(x) \geq f(x), \forall x \in C$
2. $C \subseteq C_R$.

Let $f^s : \mathbb{R}^{|VI|+|VC|} \to \mathbb{R}$ and $g^{s,r} : \mathbb{R}^{|VI|+|VC|} \to \mathbb{R}, \forall s \in S$ 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 later developments.

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

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

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

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

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 $D = \times_{s \in S} D^s$, that is D is the Cartesian products of the sets $D_s, \forall s \in S$. With this definition in mind, the Lagrangian dual function $\phi : \mathbb{R}^{(|S|-1)\times|VI|} \to \mathbb{R}$ can be defined as

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

where the components of $\lambda \in \mathbb{R}^{(|S|-1)\times|VI|}$ 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* [12]. In what follows, we present important and widely known properties of the Lagrangian dual function ϕ, which we formally state as Propositions 2 and 3 for reference later on.

Proposition 2 [5, Theorem 6.3.1] *The Lagrangian dual function 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)

$$\overline{\phi} = \inf_{\lambda} \phi(\lambda) \tag{11}$$

Even though the function ϕ is convex, evaluating $\phi(\lambda)$ for a given λ 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 3 describes how one can use the LDP (11) to obtain bounds for (9), a property generally referred to as *weak duality*.

Proposition 3 [5, Theorem 6.2.1] *Let* $f^s : \mathbb{R}^{|VI|+|VC|} \to \mathbb{R}$ *and* $g^{s,r} : \mathbb{R}^{|VI|+|VC|} \to \mathbb{R}$, $\forall s \in S$ *and* $\forall r \in R$, *be continuous twice-differentiable functions and suppose we consider the primal problem* (9). *Let* $\phi : \mathbb{R}^{(|S|-1)\times|VI|} \to \mathbb{R}$ *be Lagrangian dual function forming the Lagrangian dual problem* (11). *Let* $\overline{f}$ *and* $\overline{\phi}$ *be the optimal values of the primal and dual problems accordingly. Then, the following statements are true:*

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

It is important to highlight that, for problems including integer decision variables or nonconvex functions, a duality gap might exist (i.e., only weak duality is valid). This will hereinafter be called Issue 2. Nevertheless, the LDP can still be used to obtain bounds for the primal problem (9), as stated in Proposition 3.

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 terms were introduced by Pietrzykowski [50] and Zangwill [65] and quadratic penalty expressions by Courant [24]. Penalty expressions with both linear and quadratic pieces were proposed by Rockafellar [53]. A generalised approach called the augmented Lagrangian, also known as the methods of multipliers, was first studied by Hestenes [37] and Powell [51]. Two disadvantages of the augmented Lagrangian is that it usually ruins the problem's separable structure when it exists, thus, compromising decomposition strategies [60], and it also adds nonlinearities to the problem. The progressive hedging (PH) method, proposed by Rockafellar and Wets [54] overcomes the former obstacle. Santos et al. [56] evaluated the application of PH for solving hydrothermal systems short-term operational planning problems comparing to the Nested Decomposition. Veliz et al. [62] 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 [44] combined the tabu search method with PH to solve multistage, stochastic mixed-integer (0, 1) programming problems. Boland et al. [12] enhanced the PH convergence characteristics by introducing a Frank-Wolfe-based method to compute primal updates in PH and Boland et al. [13] presented how it could be further improved under a bundle method setting. In a similar vein, although not directly related to PH, de Oliveira et al. [28] proposed an inexact proximal bundle method decomposition for solving two-stage stochastic programming problems, which allows for the consideration of subsets of scenarios, ultimately yielding performance improvements. 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 [25] 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 (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 available off-the-shelf commercial implementations in solvers such as Gurobi [34] and CPLEX [23].

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 maximiser for $\phi(\lambda)$ for a given λ. Consider the following problem.

$$\begin{aligned} \max_{x_1,x_2} & x_1 x_2 \\ \text{s.t.:} & x_1 + 2x_2 = 1 \\ & x_1, x_2 \in [0, 1.75]. \end{aligned} \tag{12}$$

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

$$\begin{aligned} \max_{x_1,x_2} & x_1 x_2 + \lambda(x_1 + 2x_2 - 1) \\ \text{s.t.:} & x_1, x_2 \in [0, 1.75]. \end{aligned} \tag{13}$$

Solving (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 [64], 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 $(x_1, x_2) = (0.500, 0.250)$ has a greater value than that.

We could improve the solution for this problem by using another local solver that utilises alternative methods, by 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 Sect. 4.2, we revisit this example to demonstrate how to address the issue of generating valid Lagrangian bounds for nonconvex problems using RNMDT, thereby addressing Issue 1.

3 Reformulated normalised multiparametric disaggregation

The reformulated normalised multiparametric disaggregation technique (RNMDT) [1] is an enhanced version of the normalised multiparametric disaggregation technique (NMDT), originally proposed in [40] and further developed in [17, 18]. 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 (i.e., scenarios), to ease the notation.

$$\max. \quad \sum_{i \in VC} \sum_{j \in VC} Q^0_{i,j} y_i y_j + \sum_{j \in VI} I^0_j x_j + \sum_{i \in VC} C^0_i y_i \tag{14}$$

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

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

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

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

Let

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

and

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

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:

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

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

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

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 needs to be discretised, as the resulting product between continuous and binary variables can be linearised by a standard equivalent reformulation. Multiplying both sides of (18) by $y_i, \forall i \in VC$, gives the constraints

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

Next, we introduce the 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 then

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

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

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

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

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

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 [17].

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.

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

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

Note that we use the assumption that the matrices Q^r are symmetric for all $r \in R \cup \{0\}$ to replace the terms

$$\sum_{i \in VC} \sum_{j \in VC} Q_{i,j}^r y_i y_j$$

with

$$\sum_{i \mid (i,i) \in QT} Q_{i,i}^r w_{i,i} + 2 \sum_{(i,j) \in QT \mid j > i} Q_{i,j}^r w_{i,j}, \ \forall r \in R \cup \{0\}.$$

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

Definition 4 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 $p < 0$. An equivalent version of Theorem 5 has been proved in [1], but is formally stated for reference in the developments to follow.

Theorem 5 *Suppose we consider MIQCQP problem (14)–(17) and correspondent RNMDT$_p$ problems, $\forall p < 0$, introduced in Definition 4. Then, RNMDT$_p$ 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$_{p_2}$.*

Proof The proof readily follows from [1, Proposition 6 and Theorem 6]. □

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 formulate the Lagrangian relaxation to the MIQCQP and subsequently employ RNMDT to the relaxed MIQCQP probelm 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, 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$_p$ reformulation of the primal problem and then employ Lagrangian relaxation to relax the set of constraints (8), which would lead to 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 represented by the Problem (11). Therefore, this results in a p-LDP that is decomposable by $s \in S$. Each scenario subproblem can, thus, be stated as

$$\begin{aligned}
&\hat{\phi}_p^s(\lambda) = \\
&\quad \max.\ P^s \left(\sum_{j \in VI} I_j^0 x_j^s + \sum_{i|(i,i) \in QT} Q_{i,i}^r w_{i,i} + 2 \sum_{(i,j) \in QT | j>i} Q_{i,j}^r w_{i,j} \right. \\
&\quad \left. + \sum_{i \in VC} C_i^{s,0} y_i^s + \sum_{j \in VI} L^s(\lambda)_j x_j^s \right) \qquad (29) \\
&\text{s.t.:} \quad \sum_{i|(i,i) \in QT} Q_{i,i}^r w_{i,i} + 2 \sum_{(i,j) \in QT | j>i} Q_{i,j}^r w_{i,j} \\
&\qquad + \sum_{i \in VC} C_i^{s,r} y_i^s + \sum_{j \in VI} I_j^{s,r} x_j^s + K^{s,r} \leq 0, \forall r \in R
\end{aligned}$$

$$
\begin{aligned}
& y_j^s = (Y_j^{U,s} - Y_j^{L,s})\Bigg(\sum_{l \in Z_{p,-1}} 2^l z_{j,l}^s + \Delta y_j^s\Bigg) + Y_j^{L,s}, \forall j \in DS \\
& \omega_{i,j}^s = (Y_j^{U,s} - Y_j^{L,s})\Bigg(\sum_{l \in Z_{p,-1}} 2^l \hat{y}_{i,j,l}^s + \Delta \omega_{i,j}^s\Bigg) + y_i^s Y_j^{L,s}, \forall i, j \mid (i,j) \in QT \\
& 0 \leq \Delta y_j^s \leq 2^p, \forall j \in DS \\
& 2^p (y_i^s - Y_i^{U,s}) + Y_i^{U,s} \Delta y_j^s \leq \Delta \omega_{i,j}^s, \forall i, j \mid (i,j) \in QT \\
& \Delta \omega_{i,j}^s \leq 2^p (y_i^s - Y_i^{L,s}) + Y_i^{L,s} \Delta y_j^s, \forall i, j \mid (i,j) \in QT \\
& Y_i^{L,s} \Delta y_j^s \leq \Delta w_{i,j}^s \leq Y_i^{U,s} \Delta y_j^s, \forall i, j \mid (i,j) \in QT \\
& Y_i^{L,s} z_{j,l}^s \leq \hat{y}_{i,j,l}^s \leq Y_i^{U,s} z_{j,l}^s, \forall i, j \mid (i,j) \in QT, l \in Z_{p,-1} \\
& Y_i^{L,s}(1 - z_{j,l}^s) \leq y_i^s - \hat{y}_{i,j,l}^s \leq Y_i^{U,s}(1 - z_{j,l}^s), \forall i, j \mid (i,j) \in QT, l \in Z_{p,-1} \\
& y_i^s \in [Y_i^{L,s}, Y_i^{U,s}], \forall i \in VC \\
& x_j^s \in \{X_j^L, \ldots, X_j^U\}, \forall j \in VI \\
& z_{j,l}^s \in \{0,1\}, \forall j \in DS, l \in Z_{p,-1},
\end{aligned} \tag{30}
$$

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

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

One appealing feature of the p-LD 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 5). 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 [55], 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 [55, Chapter 7] for a detailed study of the properties of epi-convergence, opting to list only the properties relevant to our purposes.

4.1.1 Reformulation of DEM

To use this theorical framework, it will be convenient to include 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:

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

where, for $s \in S$,

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

and

$$f^s\left(x,\left(y^s,w^s\right)\right) = P^s\left(\sum_{j\in VI} I^0_j x_j + \sum_{i\in VC}\sum_{j\in VC} Q^{s,0}_{i,j} w^s_{i,j} + \sum_{i\in VC} C^{s,0}_i y^s_i\right), \tag{31}$$

$$g^{s,r}\left(x,\left(y^s,w^s\right)\right) = \sum_{i\in VC}\sum_{j\in VC} Q^{s,r}_{i,j} w^s_{i,j} + \sum_{i\in VC} C^{s,r}_i y^s_i + \sum_{j\in VI} I^{s,r}_j x_j + K^{s,r}, \tag{32}$$

while X and Y^s represent the box constraints (16) and (17) for integer and continuous variables, respectively. Denote the convex hull of a set C by conv C. In our final result, we will assume that $0 \in \text{int conv } C^s$ for all $s \in S$. This, without loss of generality, can in turn be ensured via a translation (and a subsequent change of variable) whenever int conv $C^s \neq \emptyset$ for all $s \in S$.

Recall that the constraints (25) and (26) in Definition 4 provide a relaxation of the product of two continuous variables and are known as McCormick envelopes [17]. Denote these by S_{R_p}, which comprises the variables $\left(y, x, \hat{y}, z, \Delta y, \Delta w\right)$ 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 $(x,(y,w)) \in X \times Y^s \times \mathbb{R}^{|VC|^2} \subseteq \mathbb{R}^{|VI|+|VC|+|VC|^2}$ for $s \in S$. The current space of variables consists of $\left(y, x, \hat{y}, z, \Delta y, \Delta w\right)$. Analogously to the derivations in the proof of Theorem 5, let us define the mapping $M : \left(y, x, \hat{y}, z, \Delta y, \Delta w\right) \to \left(x', \left(y', w'\right)\right)$ to be

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

$$w'_{i,j} = (Y^U_j - Y^L_j)\left(\sum_{l\in Z_{p,-1}} 2^l \hat{y}_{i,j,l} + \Delta w_{i,j}\right) + y_i Y^L_j.$$

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

$$P_{R_p} \;:\; v\left(\left\{-f^s\right\}_{s\in S}, \left\{C^s_{R_p}\right\}_{s\in S}\right) = \min.{}_{(x,y,w)} \left[\sum_{s\in S}\left(-f^s\right)\left(x,\left(y^s,w^s\right)\right) : \left(x,y^s,w^s\right) \in C^s_{R_p}\right] \tag{33}$$

where, for $s \in S$,

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

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 [17].

Proposition 6 *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

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

Proof The proof of the first part is equivalent to that of Theorem 5 in Sect. 3. To prove the second part, first, notice that $P_{R_p}, \forall p < 0$, is a reformulated equivalent of NMDT relaxation presented in [18]. As demonstrated in [18, Property 3], the NMDT relaxation converges to the primal problem P as $p \to -\infty$, which in turn implies that $\lim_{p\to-\infty} w^s_{i,j} = y^s_i y^s_j$. □

4.1.2 Convergence of relaxations of nonconvex optimisation problems

We will be relating the convergence of the RNMDT$_p$ relaxations to the notion of epi-convergence. In the subsection we develop a general theoretical framework applicable to this context.

Definition 7 Let $f_k : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a sequence of extended real-valued functions. We say that $\{f_k\}$ epi-converges to f and write $e\text{-}\lim_k f_k = f$ if and only if $\lim_k \text{epi } f_k = \text{epi } f$, where $\text{epi } f_k = \{(x, \alpha) \mid f_k(x) \leq \alpha\}$ and $\text{epi } f = \{(x, \alpha) \mid f(x) \leq \alpha\}$.

The limit of sets in Definition 7 is understood to be in the Kuratowski-Painleve sense. That is, (i) all accumulation points of the subsequence $\left(x_{k_l}, \alpha_{k_l}\right) \in \text{epi } f_{k_l}$ are in epi f (i.e., $\left(x_{k_l}, \alpha_{k_l}\right) \to (x, \alpha) \in \text{epi } f$) and (ii) for all $(x, \alpha) \in \text{epi } f$ there exists a sequence $(x_k, \alpha_k) \in \text{epi } f_k$ with $(x_k, \alpha_k) \to (x, \alpha)$. Recall that a function f is lower semi-continuous if and only if 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 relaxation (cf. Definition 1) 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 $v\left(f_{R_2}, C_{R_2}\right) = \max.\left(f_{R_2}(x) + \delta_{C_{R_2}}(x)\right) \leq v\left(f_{R_1}, C_{R_1}\right)$. To place the study of such relaxations into the framework of epi-convergence, we must consider the equivalent minimization problem, so

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

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 extended real-valued 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 [55, Propositions 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 and, thus, it could contain integer constraints on variables. In the analysis of 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 $\left\{\delta_{C_{R_k}}\right\}_k$. Although it is natural that the relaxations C_{R_k} epi-converge (as they are monotonically become "tighter" as p decreases, as stated in Proposition 6) but we need to establish to what problem it converges to. This is stated in Proposition 8. Moreover, notice the role of Proposition 6 in ensuring that the RNMDT$_p$ relaxation satisfies the condition 2 in Proposition 8.

Proposition 8 *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. *Let us denote* $\overline{C}$ *as the closure of the set* C. *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. [55, Exercise 7.3])

$$\begin{aligned} e\text{-}\limsup_{k\to\infty} g_k(\bar{x}) &= \sup_{\delta>0} \limsup_{k\to\infty} \inf_{x\in B_\delta(\bar{x})} g_k(x) \\ &\leq \sup_{\delta>0} \inf_{x\in B_\delta(\bar{x})} g(x) \leq g(\bar{x}) = -f(\bar{x}) + \delta_C(\bar{x}), \end{aligned} \tag{34}$$

with the last inequality following from $\inf_{x\in B_\delta(\bar{x})} g(x) \leq g(\bar{x})$ for all $\delta > 0$. By [55, Theorem 7.46], we always have (for any extended real-valued sequence)

$$e\text{-}\liminf_{k\to\infty}\left(-f_{R_k}\right)(\bar{x})+e\text{-}\liminf_{k\to\infty}\delta_{C_{R_k}}(\bar{x})\leq e\text{-}\liminf_{k\to\infty}\left(-f_{R_k}+\delta_{C_{R_k}}\right)(\bar{x})$$
$$=e\text{-}\liminf_{k\to\infty}g_k(\bar{x})\,. \tag{35}$$

Under our assumptions, $e\text{-}\liminf_{k\to\infty}(-f_{R_k})(\bar{x})=-f(\bar{x})$ and $e\text{-}\liminf_{k\to\infty}\delta_{C_{R_k}}(\bar{x})=\delta_C(\bar{x})$. This is because $C\subseteq C_{R_{k+1}}\subseteq C_{R_k}$ and $\overline{\bigcap_k C_{R_k}}=C$ implies that $\{\delta_{C_{R_k}}\}_{k=1}^{\infty}$ 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, (34) and (35) implies

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

and, thus, equality ensues. □

4.1.3 Application to the convergence of the Lagrangian bounds

In this subsection, we will demonstrate the connection between the p-Lagrangian dual function $\hat{\phi}_p$ and the Lagrangian dual function ϕ of the primal RDEM problem, analogously to the developments presented in Sect. 2. We begin with some elementary observations that will be useful in the developments hereinafter.

Proposition 9 *Let f be the objective function of the primal MIQCQP problem and $\bar{f}$ be its optimal value. Let ϕ be the Lagrangian dual function of the correspondent LDP and the functions $\hat{\phi}_p:\mathbb{R}^{|S|-1\times|VI|}\to\mathbb{R}$ be the p-Lagrangian dual function of the correspondent p-LDP, where $p\in\mathbb{Z}^-$. Then the following statements are true.*

(i) $\hat{\phi}_p$ *is convex,* $\forall p<0$.
(ii) $\hat{\phi}_{p_2}\geq\hat{\phi}_{p_1}\ \forall p_1<p_2<0$.
(iii) $\hat{\phi}_p\geq\phi,\ \forall\ p<0$.
(iv) $\hat{\phi}_p\geq\overline{f},\ \forall\ p<0$.
(v) $\inf_\lambda\hat{\phi}_{p_2}(\lambda)\geq\inf_\lambda\hat{\phi}_{p_1}(\lambda)\geq\inf_\lambda\phi(\lambda),\forall p_1<p_2<0$.

Proof Statement *(i)* is analogous to Proposition 2. On the other hand, the statements *(ii)* and *(iii)* are a direct consequence of Theorem 5. In particular, *(iii)* is also a consequence of the fact that $\phi_p\to\phi$ as $p\to-\infty$.

From Proposition 3, we know that $\phi\geq\overline{f}$. From statement *(iii)*, we have that $\hat{\phi}_p\geq\phi$. It follows by transitivity that $\hat{\phi}_p\geq\overline{f}$, which proves *(iv)*. Lastly, combining statements *(ii)* and *(iii)* and taking the infimum leads us to the conclusion in statement *(v)*. □

The following results show 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.

To this end, let $f^s:\mathbb{R}^{|VI|+|VC|+|VC|^2}\to\mathbb{R}$ and $g^{s,r}:\mathbb{R}^{|VI|+|VC|+|VC|^2}\to\mathbb{R}$, $\forall s\in S$ 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 are given by:

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

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

We will need to utilise a result that examines epi-convergence considered through the conjugation operator to obtain a result regarding the convergence of the optimal Lagrangian bound. The reason for this is that epi-convergence is widely recognised as the primary form of convergence under which we have convergence of the associated marginal mapping.

From Proposition 9, we have that $-\hat{\phi}_p(\lambda)\le v\left(\{-f^s\}_{s\in S},\left\{C^s_{R_p}\right\}_{s\in S}\right)$, which implies that $\inf_\lambda\hat{\phi}_p(\lambda)\ge v\left(\{f^s\}_{s\in S},\left\{C^s_{R_p}\right\}_{s\in S}\right)$, 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 consider this dual form of 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 $f_k:=-\sum_{s\in S}f^s_{R_k}+\delta_{C^s_{R_k}}$ are equi-hypercoercive due to the boundedness assumption on $C^s_{R_k}$ and, thus, we can use the following result to state the epi-convergence of $\hat{\phi}_p$ as $p\to-\infty$. Note that this result does not assume $\{f_k\}_{k=1}^{\infty}$ is a sequence of convex functions.

Theorem 10 (adapted from [49, 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*

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

This enables us to state the following general result.

Corollary 11 *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 increasingly tighter relaxations of the optimisation problem given by*

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

We assume

(i) $f^s_{R_k}(x,(y^s,w^s)) \geq f^s_{R_{k+1}}(x,(y^s,w^s)) \geq f^s(x,(y^s,w^s))$ *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*

(ii) for some $K > 0$ *we have* $C^s \subseteq C^s_{R_{k+1}} \subseteq C^s_{R_k} \subseteq B_K(0)$, $s \in S$ *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 6 and 8 , 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 10, 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 9), we can finally arrive to the convergence result we were aiming at.

Proposition 12 *Consider the optimisation problem P 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 increasingly tighter approximations of P given by*

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

posit the assumption as in Corollary 11. Then $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ *converges uniformly on bounded sets.*

Proof We utilise [55, Theorem 7.17], that says that when ϕ is a convex, lower semi-continuous function on $\mathbb{R}^n$, and $\operatorname{dom}\phi$ has nonempty interior, the epi-convergence of $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ to uniform convergence of $\phi(\cdot)$ 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 [55, Theorem 7.17] and Corollary 11 we immediately obtain uniformly convergence on bounded sets. □

We now finish this analysis by verifying that our problem satisfies the assumptions of Proposition 12 and that this ensure convergence of the optimal Lagrangian bound. We note that the latter could not be proven without the utilisation of epi-convergence. Denote by $\operatorname{int} C$ the interior of set C and by $\operatorname{conv} f$ the convex function whose epigraph is formed by taking the convex hull of the set $\operatorname{epi} f$. We say that a sequence $\{f_k\}$ of functions is eventually level-bounded if for each $\alpha \in \mathbb{R}$ the sequence of sets $\operatorname{lev}_\alpha f_k := \{x \mid f_k(x) \leq \alpha\}$ is eventually contained in a bounded set.

Theorem 13 *Consider the problem DEM as formulated as problem (P) in Sect. 4.1.1 along with its approximation (P_{R_p}). We employ Lagrangian duality to the problem P_{R_k} to obtain the p-LDP with $p = -k$ and denote the corresponding Lagrangian dual function by $\hat{\phi}_{-k}$. Then, $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ converges uniformly on bounded sets and epi-converges. If, in addition, $0 \in \operatorname{int}\operatorname{conv} C^s$, $\forall s \in S$, then we also have the optimal dual values:*

$$\inf_{\lambda} \hat{\phi}_{-k}(\lambda) \xrightarrow[k\to\infty]{} \inf_{\lambda} \phi(\lambda)$$

Furthermore if $\varepsilon_k \downarrow 0$ and $\lambda_k \in \varepsilon_k\text{-}\arg\min \hat{\phi}_{-k} := \left\{\lambda \mid \hat{\phi}_k(\lambda) - \varepsilon_k \leq \inf \hat{\phi}_k\right\}$, the sequence $\{\lambda_k\}$ is bounded with all its accumulation points in $\arg\min \phi$. If $\arg\min \phi$ consists of a unique point $\bar{\lambda}$ then $\lambda_k \to \bar{\lambda}$.

Proof To apply Proposition 12, we note that ($P_{R_{-k}}$), as stated in Sect. 4.1.1, has an objective function that is continuous and that is not dependent on k, hence assumption *(i)* of Corollary 11 is satisfied. Assumption *(ii)* of Corollary 11 follows immediately from Proposition 6 and the fact that $C^s_{R_p}$ are closed and uniformly bounded sets due to the box constraints X and Y^s being bounded. It follows from Proposition 12 that $\left\{\hat{\phi}_{-k}(\cdot)\right\}_{k=1}^{\infty}$ epi-converges (and uniformly on bounded sets). All other claims follow from applying [55, Theorem 7.33] combined with [55, Exercise 7.32(c)], once we have established that $\hat{\phi}_{-k}$ are eventually level set bounded. This follows from applying [52, Theorem 4C], once we establish that $0 \in \operatorname{int}\operatorname{dom}\operatorname{conv} \sum_{s\in S\setminus\{s'\}} \left(-f^s + \delta_{C^s}\right)$ and $0 \in \operatorname{int}\operatorname{dom}\operatorname{conv}\left(-f^{s'} + \delta_{C^{s'}}\right)$, noting that convex conjugate satisfies $(\operatorname{conv} g)^* = (g)^*$. This, in turn, is true whenever $0 \in \operatorname{int}\operatorname{conv} C^s$, $\forall s \in S$. □

4.2 Motivating example: part 2

We revisit the motivating example in Sect. 2.1. By employing the $p-$Lagrangian relaxation as described before, we obtain the p-LDP formulation

$$\begin{aligned}
\max._{w,\Delta w,z,\Delta x,\hat{x},x}\ & w_{1,2} + \lambda(x_1 + 2x_2 - 1) \\
\text{s.t.:}\ & w_{1,2} = 2^{-1}\hat{x}_{1,2,-1} + v_{1,2} \\
& x_2 = 2^{-1}z_{2,-1} + \Delta x_2 \\
& 0 \leq \hat{x}_{1,2,-1} \leq z_{2,-1} \\
& 0 \leq x_1 - \hat{x}_{1,2,-1} \leq 1 - z_{2,-1} \\
& 2^{-1}(x_1 - 1) + \Delta x_2 \leq \Delta w_{1,2} \leq 2^{-1}x_1 \\
& 0 \leq \Delta w_{1,2} \leq \Delta x_2 \\
& z_{2,-1} \in \{0, 1\} \\
& x_1, x_2 \in [0, 1.75] \\
& \Delta x_2 \in [0, 2^{-1}].
\end{aligned} \tag{36}$$

In this example, the variable x_2 is discretised with a precision $p = -1$ and the Lagrangian multiplier value is fixed to -0.100. Solving the resulting problem with Gurobi 9.0 [34], we obtain the p-Lagrangian dual bound $\hat{\phi}_{-1}(-0.100) = 0.800$, 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 by solving the p-Lagrangian dual problem employing a nonsmooth optimisation method. In this example, employing a subgradient method [36], the optimal Lagrangian multiplier $\lambda = -0.333$ and correspondent value of the dual objective function $\phi(-0.333) = 0.334$ 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 -0.333 provides with a bound $\hat{\phi}_{-1}(-0.333) = 0.334$.

If one was to obtain a tighter bound $\hat{\phi}_p$, it would be necessary to choose more carefully the parameter p. In Sect. 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 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) tightening 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 employs 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 have 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 (30). 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.

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

$$f_{rank}(s, j) = \sum_{r \in R} \left\{ |Q_{j,j}^{s,r}(w_{j,j}^s - y_j^{s2})| + 2 \sum_{(r,i)|i>j|(i,j)\in QT} |Q_{i,j}^{s,r}(w_{i,j}^s - y_i^s y_j^s)| \right\}. \quad (37)$$

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 (measured in number of iterations) of the periodic increase in precision of the variables that remained unchanged (in the last N_2 iterations).

Algorithm 1 p-LD 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 Lagrangian-based heuristic mentioned in Step 2 of Algorithm 1 can be any method capable of generating primal feasible solutions. In this study, the heuristic employed at each iteration of the p-LD algorithm consists 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.

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

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 of $x_j^{avg}, \forall j \in VI, s \in S$. 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 higher 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 to update the Lagrangian multipliers values. In this paper, we used the bundle method. For a discussion on methods for performing the Lagrangian multiplier updates, please refer to the "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-LD 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-LD algorithm, with 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 "A").

For the sake of completeness, the following theorem formally states the convergence of Algorithm 1. Notice that optimality gap obtained from Algorithm 1 is not guaranteed to converge to zero, as, while we can provide convergence guarantees for the dual bound (UB), no such guarantee can be provided for the Lagrangian-heuristic generating primal bounds (LB). Nevertheless, as illustrated in Sect. 6, often Algorithm 1 is capable of obtaining reasonably good solutions with reasonably small optimality gaps.

Theorem 14 *Consider the RDEM problem defined in Sect. 1 and corresponding p-Lagrangian dual problem p-LDP with the objective function $\hat{\phi}_p$ for a fixed value $p < 0$ as defined in Sect. 4. Let f^* be the optimal objective value of the RDEM problem and UB^* (LB^*) be the corresponding dual bound (primal bound) value resulting from applying Algorithm 1 to p-LDP. Then, provided that Step 3 of Algorithm 1 utilises a method guaranteed to converge to the optimal Lagrangian multipliers $\overline{\lambda}_p := \arg\inf \hat{\phi}_p$, Algorithm 1 converges to the point UB^* (LB^*) that provides an upper (lower) estimate of the optimal objective value, i.e. $UB^* \geq f^* \geq LB^*$.*

Proof Note that the employment of the bundle method in Step 4 of Algorithm 1, guarantees finite convergence when solving p-LDP for a given value of p (see for example [7, Proposition 5.3.2]). Moreover, note that Step 4 guarantees that $p \to -\infty$ since, at every N_2 iterations, all of the variables that have not had their precision p updated will have their value p_j^s decremented in one unit. Therefore, we have that

$$UB^* = \lim_{p \to -\infty} \left\{ \inf_{\lambda} \hat{\phi}_p \right\} \geq f^* \geq LB^*.$$

by Proposition 3 and Theorem 13. □

6 Computational experiments

In this section, we present numerical results for randomly generated nonconvex RDEM problems solved with Algorithm 1 (p-LD).We have generated random 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) [34] and the direct employment of dp-RNMDT. All the experiments were designed using the Julia (version 1.3.1) language [8] and the commercial solver Gurobi (version 9.0.0). All code and instances generated are freely available on 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 collection of randomly generated instances. Each set of instances contained the RDEM problems with 5, 10, 15, 20, and 25 scenarios,

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

represented in three sizes (small, medium, and large) as described in Table 1. Thus, the smallest instance has a total of 5 scenarios and therefore 100 continuous variables, 25 integer variables and 225 constraints, while the largest instance has 1000 continuous variables, 375 integer variables, and 1875 constraints. The quadratic matrices $Q_{i,j}^{s,r}$ for all $s \in S, r \in \{0\} \cup R$ were randomly generated with approximately 80% density, a number that was arbitrarily chosen to match that observed in instances from [63]. 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. More details regarding the instance generation procedure are given in "B".

The p-LD Algorithm 1 was implemented in two versions utilising sequential and parallel computing (using multiple threads). 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 2 presents the parameter values for p-LD, including the parameter values for the dynamic precision-based algorithm and for bundle method used for updating the Lagrangian multipliers within p-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 the bundle method was based on the pairwise differences between the Lagrangian dual function values considering the last three iterations (see Sect. 5 and "A", respectively, for more details on the termination criteria; a discussion on setting different values for N_1 and N_2 and their impact on the algorithm's performance is given in [1]). It is worth mentioning that both methods can be sensitive to the initial parameter values, which were set 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 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 serially (i.e., with no parallelisation) 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. We highlight (with bold font) the cells corresponding to the serial method having superior

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, $\forall j \in VI$ and $\forall s \in S$	0

performance in terms of optimality gaps, and running time at convergence (indicated by the solution time is seconds) or termination due to the time limit of 3600s (indicated by the letter "T" in the column "Time (s)").

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 a 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, 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 the generated 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

$$\frac{\text{UB} - \min(\text{UB}_{\text{dp-RNMDT}}, \text{UB}_{p\text{-LD}})}{\min(\text{UB}_{\text{dp-RNMDT}}, \text{UB}_{p\text{-LD}})} \times 100, \tag{38}$$

where "$\text{UB}_{\text{dp-RNMDT}}$" and "$\text{UB}_{p\text{-LD}}$" 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 (larger) the other UB obtained are.

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

Bold font indicates the best performing method in terms of the corresponding criterion

As before, we highlight (with 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 cases in which the method generated an equal or better bound for the column "UB" and the number of instances for which 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 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.

Figure 1 illustrates the progress of the parallel p-LD with respect to the duality gap and the elapsed time for three arbitrarily selected instances of different sizes, each with 15 scenarios. As can be seen, the plots indicate the existence of a threshold value in the number of iterations performed by p-LD after which the reduction rate in the duality gap decreases considerably, while the computational time required per iteration continues to steadily increase (noticeable by the upward curvature of the time curve). The latter is a consequence of the increase in the number of binary variables in the p-LDP as the precision of the relaxation increases.

In the experiments conducted, we noticed that the time taken in each iteration is mostly (roughly 90%) dedicated to solving the (MIP scenario subproblems) p-LDP, while the remainder is majorly taken performing the heuristic to obtain primal feasible solutions. Only a negligible percentage of the iteration time (considerably less than 1%) is taken by the bundle method to calculate the Lagrangian multipliers. This pattern was observed for all instances, regardless of the number of scenarios or the size of the scenario subproblems, which motivates the observed benefits arising from parallelisation.

6.2.1 Performance profiles

To provide a structured comparison between the p-LD and dp-RNMDT, we present performance profiles based on [30]. 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

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

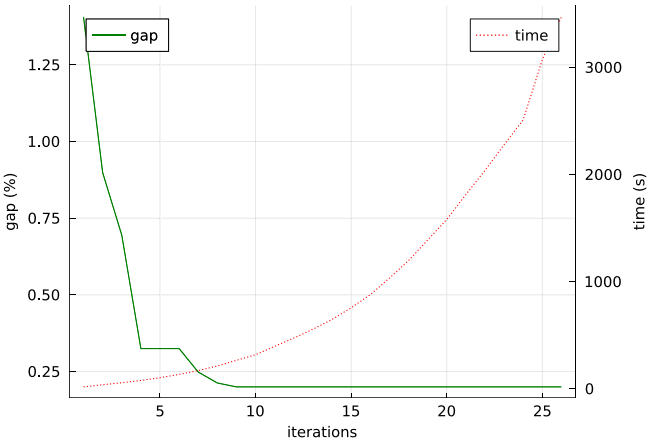
Analogously, let $ub_{p,a}$ be the UB achieved by the algorithm $a \in A$ for the problem $p \in P$. Let the UB performance ratio be defined as

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

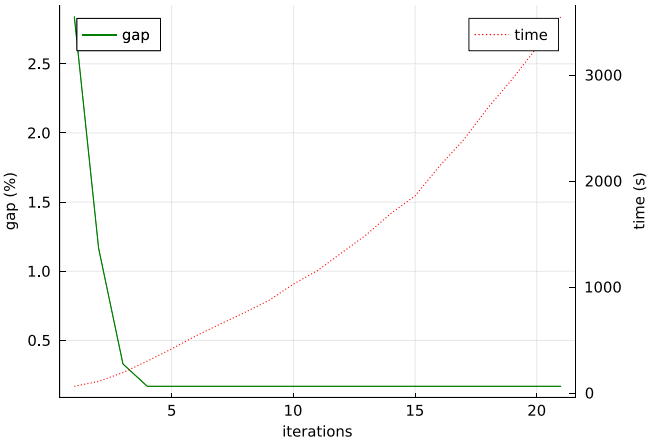
Table 4 Comparison analysis of the dp-RNMDT and p-LD convergence

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**

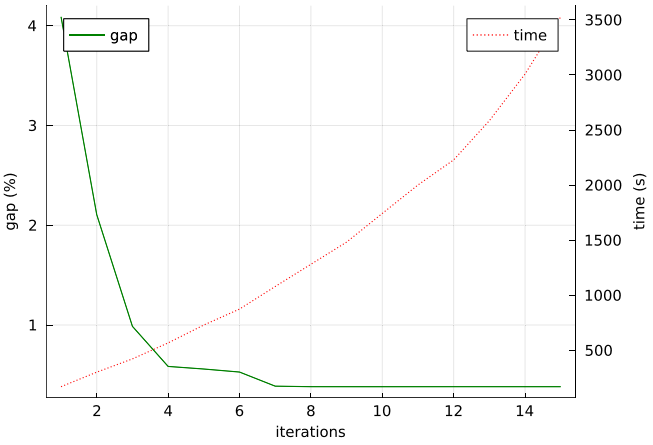
Bold font indicates the best performing method in terms of the corresponding criterion



(a) small



(b) medium



(c) large

Fig. 1 p-LD gap and time plots for instances with 15 scenarios

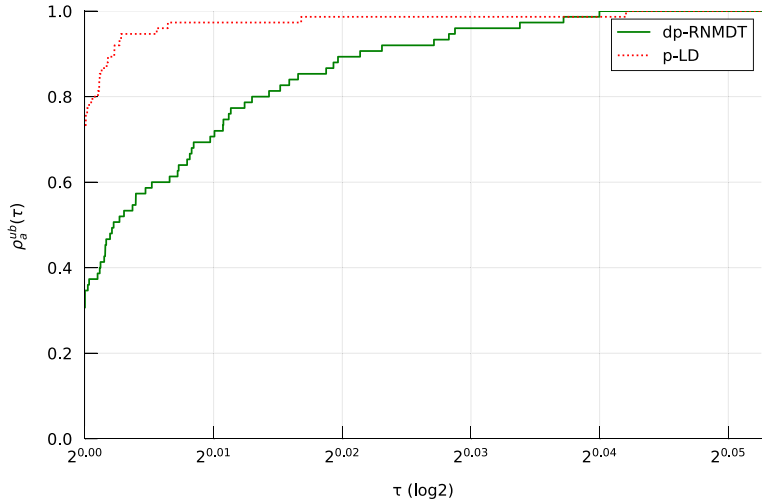


Fig. 2 Performance profile based on the upper bounds

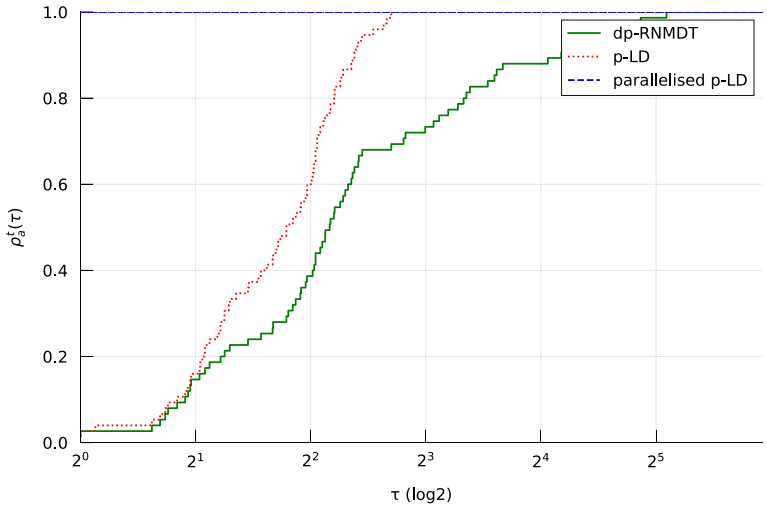


Fig. 3 Performance profiles based on the convergence time

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

$$\rho_a^{ub}(\tau) = \frac{|P_{\tau,a}^{ub}|}{|A|}.$$

Figures 2 and 3 present the dual (upper) bound and time performance profiles, respectively. In Fig. 3, 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 Figs. 2 and 3 , p-LD demonstrated superior performance when compared to dp-RNMDT in both bound generated and time performance criteria. In

addition, Fig. 3 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 decomposition, which consists of an alternative framework to achieve decomposition for nonconvex MIQCQP problems. The core idea of the p-Lagrangian decomposition is to combine two techniques: Lagrangian decomposition and RNMDT. Lagrangian decomposition 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 the precision parameter p of the RNMDT allow for controlling the quality of the relaxation. Therefore, we proposed a new algorithm named p-LD 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 the computational performance.

The numerical experiments suggest that the p-LD algorithm 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 performance suggested by the numerical results, the p-LD algorithm has two important shortcomings. The first is the dependence on a Lagrangian-based heuristic for generating primal feasible solutions which are likely to not be able to attain a desired optimality tolerance. 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 framework 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 [22]. Furthermore, the employment of the p-Lagrangian relaxation for bound 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 [58], thus enhancing the computational efficiency of the method. Additionally, the p-Lagrangian relaxation could be employed for models arising from equilibrium problems [63], which naturally yield large-scale MIQCQPs with separable structure.

Moreover, synergies with the alternating direction method of multipliers (or Gauss-Siedel approaches, such as those explored in [47]) and the employment of inexact variants of bundle methods (which would allow for controlled imprecision in the solution of the p-Lagrangian dual problems; see [27, 28]) are interesting avenues to improve the computational perfor-

mance of the method. Finally, under a theoretical standpoint, it would be interesting to investigate the convergence rate of the proposed method and whether it could be improved considering alternative nonsmooth optimisation methods for solving the p-LDPs.

A Solution methods for Lagrangian dual problems

This section describes the approaches for solving the LDP in Sect. 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 [35] and Held et al. [36] proposed the classical approach, which became known as the subgradient method. Improvements of this method were proposed by Camerini et al. [15] and Fisher [31]. An alternative method that presents better convergence properties is the cutting-plane method proposed by Cheney and Goldstein [20] and Kelley [38]. An improvement of this method is presented by Marsten et al. [45]. Other methods include the Volume algorithm [4] and the bundle method [42, 66] 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. We highlight the interesting connection this creates with the works related to inexact bundle methods, such as [27, 28], which could be an exciting future direction for research. There are multiple variations of the bundle method. Our implementation followed its classical variant, as presented in [7]. The idea of the bundle method lies in iterating the argument λ_{k+1} as follows

$$\lambda_{k+1} \in \arg\min_{\lambda}\{F_k(\lambda) + p_k(\lambda)\}. \tag{39}$$

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

$$F_k(\lambda) = \max\{\phi(\lambda_0) + (\lambda - \lambda_0)'\phi'(\lambda_0), ..., \phi(\lambda_k) + (\lambda - \lambda_k)'\phi'(\lambda_k)\}, \tag{40}$$

while $p_k(\lambda)$ is given by

$$p_k(\lambda) = \frac{1}{2c_k}||\lambda - \lambda_k^{centre}||^2, \tag{41}$$

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

$$\lambda_{k+1}^{centre} = \begin{cases} \lambda_{k+1} & \text{if } \phi(\lambda_k^{centre}) - \phi(\lambda_{k+1}) \geq \beta\delta_k \ (serious\ step) \\ \lambda_k^{centre} & \text{if } \phi(\lambda_k^{centre}) - \phi(\lambda_{k+1}) < \beta\delta_k \ (null\ step), \end{cases} \tag{42}$$

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 Instance generation

All parameter values were generated using pseudo-random number generator function with predefined minimum and maximum values for each parameter. The pseudo-random number generator was initialised with the random seeds 0,1,2,3, and 4 to ensure the generation of the same random numbers in multiple executions of the algorithms.

The lower-bound values for the continuous and integer variables were set to 0 while the upper-bound values were randomly selected as integer numbers between 50 and 100.

All matrices Q^r, for all $r \in R \cup \{0\}$ were generated using Julia's (linear algebra) embedded functions for generating symmetric matrices with predefined densities of approximately 80 %. The lower and upper limits for each of the randomly generated matrix elements were fixed to 0 and 100, respectively. These bounds were also used to generate the coefficients of the linear functions and linear parts of the affine functions. The constants appearing in affine functions were generated using a pseudo-random number generator with lower and upper bounds set to 0 and 10000000 accordingly. The code used to generate the instances, as well as the instances in independent files, are available at the GitHub repository github.com/gamma-opt/p-Lagrangian_relaxation.jl.

C 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-LDP	p-Lagrangian dual problem
QCQP	Quadratically constrained quadratic programming
RDEM	Reformulated deterministic equivalent model
RNMDT	Reformulated normalised multiparametric disaggregation technique
UB	Upper bound

References

1. Andrade, T., Oliveira, F., Hamacher, S., Eberhard, A.: Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. J. Global Optim. **73**(4), 701–722 (2019). 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. Ind. Eng. Chem. Res. **55**(1), 144–155 (2016). https://doi.org/10.1021/acs.iecr.5b01132
3. Bao, X., Sahinidis, N.V., Tawarmalani, M.: Semidefinite relaxations for quadratically constrained quadratic programming: a review and comparisons. Math. Program. **129**(1), 129–157 (2011). https://doi.org/10.1007/s10107-011-0462-2
4. Barahona, F., Anbil, R.: The volume algorithm: producing primal solutions with a subgradient method. Math. Program. **87**(3), 385–399 (2000). https://doi.org/10.1007/s101070050002
5. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 3rd edn. Wiley, New York (2006)
6. Berthold, T., Heinz, S., Vigerske, S.: Extending a CIP Framework to Solve MIQCPs. In: Lee, J., Leyffer, S. (eds.) Mixed Integer Nonlinear Programming, The IMA Volumes in Mathematics and its Applications, pp. 427–444. Springer, New York (2012). https://doi.org/10.1007/978-1-4614-1927-3_15
7. Bertsekas, D.P.: Convex Optimization Algorithms. No. 4 in Optimization and computation series. Athena Scientific, Belmont (2015)
8. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. SIAM Rev. **59**(1), 65–98 (2017). https://doi.org/10.1137/141000671
9. Billionnet, A., Elloumi, S., Lambert, A.: Extending the qcr method to general mixed-integer programs. Math. Program. **131**(1), 381–401 (2012)
10. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering. Springer, New York (2011). https://doi.org/10.1007/978-1-4614-0237-4
11. Birge, J.R., Qi, L.: Computing Block–Angular Karmarkar projections with applications to stochastic programming. Manag. Sci. **34**(12), 1472–1479 (1988). https://doi.org/10.1287/mnsc.34.12.1472
12. 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 J. Optim. **28**(2), 1312–1336 (2018). https://doi.org/10.1137/16M1076290
13. Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Oliveira, F.: A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems. Math. Program. **175**(1–2), 503–536 (2019). https://doi.org/10.1007/s10107-018-1253-9
14. Burer, S., Letchford, A.N.: Non-convex mixed-integer nonlinear programming: a survey. Surv. Oper. Res. Manag. Sci. **17**(2), 97–106 (2012). https://doi.org/10.1016/j.sorms.2012.08.001
15. Camerini, P.M., Fratta, L., Maffioli, F.: On improving relaxation methods by modified gradient techniques. In: Balinski, M.L., Wolfe, P. (eds.) Nondifferentiable Optimization, Mathematical Programming Studies, pp. 26–34. Springer, Berlin, Heidelberg (1975). https://doi.org/10.1007/BFb0120697
16. Carøe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. Oper. Res. Lett. **24**(1–2), 37–45 (1999). https://doi.org/10.1016/S0167-6377(98)00050-9
17. Castro, P.M.: Tightening piecewise McCormick relaxations for bilinear problems. Comput. Chem. Eng. **72**, 300–311 (2015). https://doi.org/10.1016/j.compchemeng.2014.03.025
18. Castro, P.M.: Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems. J. Global Optim. **64**(4), 765–784 (2016). https://doi.org/10.1007/s10898-015-0342-z
19. 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). https://doi.org/10.1016/B978-0-444-63428-3.50092-8
20. Cheney, E.W., Goldstein, A.A.: Newton's method for convex programming and Tchebycheff approximation. Numer. Math. **1**(1), 253–268 (1959). https://doi.org/10.1007/BF01386389
21. Choi, T.M., Gao, J., Lambert, J.H., Ng, C.K., Wang, J.: Optimization and Control for Systems in the Big-Data Era: Theory and Applications. International Series in Operations Research and Management Science. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-53518-0
22. Cordova, M., de Oliveira, W., Sagastizabal, C.: Revisiting Augmented Lagrangian Duals. Mathematical Programming (2020). http://www.optimization-online.org/DB_HTML/2020/03/7709.html
23. Corporation, I.: Ibm ilog cplex optimization studio v 12.8.0 (2018)
24. Courant, R.: Variational methods for the solution of problems of equilibrium and vibrations. Bull. Am. Math. Soc. **49**(1), 1–23 (1943). https://doi.org/10.1090/S0002-9904-1943-07818-4

25. Dentcheva, D., Römisch, W.: Duality gaps in nonconvex stochastic optimization. Math. Program. **101**(3), 515–535 (2004). https://doi.org/10.1007/s10107-003-0496-1
26. Dept. of Mathematical Sciences, C.U.: Couenne: a user's manual. https://www.coin-or.org
27. de Oliveira, W., Sagastizábal, C.: Level bundle methods for oracles with on-demand accuracy. Optim. Methods Softw. **29**(6), 1180–1209 (2014)
28. de Oliveira, W., Sagastizábal, C., Scheimberg, S.: Inexact bundle methods for two-stage stochastic programming. SIAM J. Optim. **21**(2), 517–544 (2011)
29. 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 Trans. Power Syst. **30**(6), 2841–2855 (2015). https://doi.org/10.1109/TPWRS.2014.2375322
30. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263
31. Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. Manag. Sci. **27**(1), 1–18 (1981). https://doi.org/10.1287/mnsc.27.1.1
32. Geoffrion, A.M.: Generalized Benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972). https://doi.org/10.1007/BF00934810
33. Guignard, M.: Lagrangian Relaxation, pp. 845–860. Springer, Boston (2013). https://doi.org/10.1007/978-1-4419-1153-7_1168
34. Gurobi Optimization, L.: Gurobi Optimizer Reference Manual (2020). http://www.gurobi.com
35. Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees: part ii. Math. Program. **1**(1), 6–25 (1971)
36. Held, M., Wolfe, P., Crowder, H.P.: Validation of subgradient optimization. Math. Program. **6**(1), 62–88 (1974). https://doi.org/10.1007/BF01580223
37. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**(5), 303–320 (1969). https://doi.org/10.1007/BF00927673
38. Kelley, J.E., Jr.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**(4), 703–712 (1960). https://doi.org/10.1137/0108053
39. Kianfar, K.: Maximizing profit in a supply chain by considering advertising and price elasticity of demand. Comput. Ind. Eng. **135**, 265–274 (2019). https://doi.org/10.1016/j.cie.2019.06.007
40. Kolodziej, S., Castro, P.M., Grossmann, I.E.: Global optimization of bilinear programs with a multiparametric disaggregation technique. J. Global Optim. **57**(4), 1039–1063 (2013). https://doi.org/10.1007/s10898-012-0022-1
41. Lazzeroni, P., Repetto, M.: Optimal planning of battery systems for power losses reduction in distribution grids. Electr. Power Syst. Res. **167**, 94–112 (2019). https://doi.org/10.1016/j.epsr.2018.10.027
42. Lemaréchal, C.: An algorithm for minimizing convex functions. In: IFIP Congress, pp. 552–556 (1974)
43. Li, X., Tomasgard, A., Barton, P.I.: Nonconvex generalized benders decomposition for stochastic separable mixed-integer nonlinear programs. J. Optim. Theory Appl. **151**(3), 425–454 (2011). https://doi.org/10.1007/s10957-011-9888-1
44. Løkketangen, A., Woodruff, D.L.: Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. J. Heuristics **2**(2), 111–128 (1996). https://doi.org/10.1007/BF00247208
45. Marsten, R.E., Hogan, W.W., Blankenship, J.W.: The Boxstep method for large-scale optimization. Oper. Res. **23**(3), 389–405 (1975). https://doi.org/10.1287/opre.23.3.389
46. Misener, R., Floudas, C.A.: Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. Math. Program. **136**(1), 155–182 (2012). https://doi.org/10.1007/s10107-012-0555-6
47. Oliveira, F., Christiansen, J., Dandurand, B., Eberhard, A.: Combining penalty-based and Gauss–Seidel methods for solving stochastic mixed-integer problems. Int. Trans. Oper. Res. **27**(1), 494–524 (2020). https://doi.org/10.1111/itor.12525
48. Oliveira, F., Gupta, V., Hamacher, S., Grossmann, I.E.: A Lagrangean decomposition approach for oil supply chain investment planning under uncertainty with risk considerations. Comput. Chem. Eng. **50**, 184–195 (2013). https://doi.org/10.1016/j.compchemeng.2012.10.012
49. Penot, J.P., Zalinescu, C.: Continuity of the Legendre–Fenchel transform for some variational convergences. Optimization **53**, 549–562 (2004)
50. Pietrzykowski, T.: An exact potential method for constrained maxima. SIAM J. Numer. Anal. **6**(2), 299–304 (1969). https://doi.org/10.1137/0706028
51. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. Optimization, pp. 283–298 (1969). https://ci.nii.ac.jp/naid/20000922074/en/
52. Rockafellar, R.T.: Level sets and continuity of conjugate convex functions. Trans. Am. Math. Soc. **123**, 46–63 (1966). https://doi.org/10.2307/1994612

53. Rockafellar, R.T.: Lagrange multipliers and optimality. SIAM Rev. **35**(2), 183–238 (1993). https://doi.org/10.1137/1035044
54. Rockafellar, R.T., Wets, R.J.B.: Scenarios and policy aggregation in optimization under uncertainty. Math. Oper. Res. **16**(1), 119–147 (1991). https://doi.org/10.1287/moor.16.1.119
55. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis, vol. 317. Springer, Berlin (2009). Google-Books-ID: JSREAAAAQBAJ
56. Santos, M., Silva, E., Finardi, E., Gonçalves, R.: Solving the short term operating planning problem of hydrothermal systems by using the progressive hedging method. In: 16th Power Systems Computation Conference, PSCC 2008 (2008)
57. Singh, M.K., Kekatos, V.: Natural gas flow solvers using convex relaxation. IEEE Trans. Control Netw. Syst. (2020). https://doi.org/10.1109/TCNS.2020.2972593
58. Smith, E.M.B., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Comput. Chem. Eng. **23**(4–5), 457–478 (1999). https://doi.org/10.1016/S0098-1354(98)00286-5
59. Strub, O., Brandinu, S., Lerch, D., Schaller, J., Trautmann, N.: A three-phase approach to an enhanced index-tracking problem with real-life constraints. Eng. Econ. **64**(3), 227–253 (2019). https://doi.org/10.1080/0013791X.2019.1619887
60. Tappenden, R., Richtárik, P., Büke, B.: Separable approximations and decomposition methods for the augmented Lagrangian. Optim. Methods Softw. **30**(3), 643–668 (2015). https://doi.org/10.1080/10556788.2014.966824
61. The Optimization Firm, L.: Baron user manual v. 2019.12.7 (2019). https://www.minlp.com
62. 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. Ann. Oper. Res. **232**, 259–274 (2015). https://doi.org/10.1007/s10479-014-1608-4
63. Virasjoki, V., Siddiqui, A., Oliveira, F., Salo, A.: Utility-scale energy storage in an imperfectly competitive power sector. Energy Econ. (2020). https://doi.org/10.1016/j.eneco.2020.104716
64. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006). https://doi.org/10.1007/s10107-004-0559-y
65. Zangwill, W.I.: Non-linear programming via penalty functions. Manag. Sci. **13**(5), 344–358 (1967). https://doi.org/10.1287/mnsc.13.5.344
66. Zhao, X., Luh, P.: New bundle methods for solving Lagrangian relaxation dual problems. J. Optim. Theory Appl. **113**(2), 373–397 (2002). https://doi.org/10.1023/A:1014839227049