A Polynomial-time Algorithm with Tight Error Bounds for Single-period Unit Commitment Problem

Ruotian Gao, Shu-Cherng Fang, Cheng Lu and Wenxun Xing

November 1, 2019

Abstract

This paper proposes a Lagrangian dual based polynomial-time approximation algorithm for solving the single-period unit commitment problem, which can be formulated as a mixed integer quadratic programming problem and proven to be NP-hard. Tight theoretical bounds for the absolute errors and relative errors of the approximate solutions generated by the proposed algorithm are provided. Computational results support the effectiveness and efficiency of the proposed algorithm for solving large-scale problems.

Keywords: nonlinear programming, Lagrangian dual, unit commitment problem, mixed integer quadratic programming, convex relaxation

1 Introduction

Unit commitment (UC) problems play an important role in the electricity market. The task of UC is to find an optimal schedule of available generators with limited capacities over a planning horizon to achieve a minimum cost while the demand and some operational constraints are satisfied. A general UC problem is usually a large-scale nonconvex optimization problem. The nonconvexity is mainly caused by the binary nature of UC decisions (on or off status of each generator). Effectively solving UC problems is deemed as a challenging task, see [37], [27] and [19].

In this paper, we focus on the single-period UC problem (SPUC) that tries to select a best plan for allocating the demand to a group of generators in hand in order to achieve a minimum cost in one particular planning period. The problem is also known as centralized pool scheduling problem [13], which can be mathematically formulated as the following mixed-integer quadratic program:

$$\begin{aligned} min \quad & \sum_{i=1}^{n}(a_i x_i^2 + b_i x_i + c_i y_i) \\ s.t. \quad & \sum_{i=1}^{n} x_i = d \\ & 0 \le x_i \le u_i y_i, y_i \in \{0,1\}, i = 1, \cdots, n, \end{aligned} \tag{1.1}$$

where $u_i > 0$ is the maximum capacity of generator i, $a_i x_i^2 + b_i x_i + c_i y_i$ is its cost function with $a_i \ge 0$, $b_i \ge 0$ and $c_i \ge 0$ to assure a monotonically increasing cost, and $y_i \in \{0,1\}$

represents the status of the i-th generator. Once the i-th generator is switched on, $y_i = 1$ and a set-up cost c_i arises. If $a_i = 0$, for $i = 1, \cdots, n$, then problem (1.1) becomes the single-node fixed-charge problem [25]. If $b_i = 0$ and $u_i = 1$, for $i = 1, \cdots, n$, and $d = 1$, then problem (1.1) becomes the sensor placement problem [11]. In this paper, $d \in [0, \sum_{i=1}^{n} u_i]$ is assumed to assure the feasibility of problem (1.1). Absorbing the integer variable y_i, for $i = 1, \cdots, n$, into the objective function, problem (1.1) becomes:

$$\begin{aligned} min \quad & \sum_{i=1}^{n} f_i(x_i) \\ s.t. \quad & \sum_{i=1}^{n} x_i = d, \end{aligned} \tag{1.2}$$

where

$$f_i(x_i) = \begin{cases} 0, & \text{if } x_i = 0 \\ q_i(x_i) = a_i x_i^2 + b_i x_i + c_i, & \text{if } 0 < x_i \le u_i \\ +\infty, & \text{if } x_i \notin [0, u_i]. \end{cases} \tag{1.3}$$

Notice that the difficulty now lies in the discontinuity and nonconvexity of $f_i(x_i)$ with $c_i > 0$ for each i.

Although the UC problem (1.2) considered in this paper is of the simplest form, it is still an NP-hard problem, e.g., see [19], and it is the core function of various unit commitment problems. In the literature, there are different types of UC problems in different views. Concerning the planning horizon, UC problems include the single-period UC problems [13,19,22] and multi-period UC problems [8, 17, 31]. The latter ones involve multiple decision stages and need to determine the start-up and shut-down sequences as well as the quantity of commitment for each generator in each period, and thus more complex. Regarding the certainty of demand and capacities, there are deterministic and nondeterministic UC problems. For nondeterministic problems, stochastic programming [7], robust optimization [39] and chance-constrained optimization [36] methods were studied to address the uncertainty issue. In view of security [27], there are traditional UC problems, security-constrained UC problems [39] and price-based UC problems [40]. From the market perspective, UC problems could be considered in an integrated environment [7] or in a deregulated environment [31].

Much effort has been devoted to developing efficient methods for solving UC problems. Commonly seen approaches in the literature include at least the following: (i) branch-and-bound, (ii) dynamic programming, (iii) decomposition, (iv) Lagrangian relaxation (LR), (v) mixed integer programming (MIP) and (vi) meta-heuristics.

Branch-and-bound is a commonly used method for locating a global optimal solution, see [5] and [16]. A critical issue to speed up the branch-and-bound process is to construct an effective relaxation for providing a tight lower bound. Perspective relaxation (PR) is a promising approach of constructing tight relaxation for mixed-integer nonlinear programs. In [11], Frangioni et.al showed that, under some assumptions, the PR of a mixed-integer quadratic program can be reformulated as a piecewise-quadratic program (QP), which could ultimately yield a QP relaxation of roughly the same size of the standard continuous relaxation by replacing $\{0, 1\}$ with [0,1]. Frangioni et.al [12] constructed an approximated projected PR (AP^{2}R), which in some cases produces a reformulation with the same size and structure as the standard continuous relaxation, but with substantially improved bounds provided. A conjugate function based convex relaxation developed in [19] provides a tight lower bound for branching-and-bounding.

The corresponding branch-and-bound algorithm is shown to be much more efficient than state-of-the-art solvers like Cplex and Gurobi.

Dynamic programming (DP) is the earliest optimization method used for solving UC problems. DP is able to maintain solution feasibility but suffers from the “curse of dimensionality", see [23] and [18]. The key issue is how to reduce the dimensionality of the problem for achieving an acceptable computational time. Rong et.al [31] introduced the DP-RSC1 algorithm, which is a variant of DP based on a linear relaxation of the on/off status of the units, to address the multi-period combined heat and power (CHP) production planning problem. Their algorithm is shown to have a good solution quality with a fast speed. They also proved the complexity is proportional to the number of units, the number of the time periods and the computation time of the single-period economic dispatch problem. This highlights as well the importance of studying the SPUC problem.

Decomposition such as Dantzig-Wolfe decomposition [35] and Benders decomposition [20] could be used to handle both of the deterministic problems and nondeterministic problems. For example, Ma and Shahidehpour [20] designed an efficient algorithm by using Benders decomposition to decompose a deterministic transmission-constrained unit commitment problem into a master problem and a subproblem, and Schulze et.al [35] used a Dantzig-Wolfe reformulation to decompose a stochastic UC problem by the scenarios generated by parameter distributions.

Lagrangian relaxation (LR) adjoints constraints to the objective function using Lagrangian multipliers. Then a relaxed dual problem is derived to provide a lower bound for the original problem. For the dual problem, subgradient method [28] or proximal bundle method [3] could be applied. However, the corresponding primal solution obtained by the dual information is in general infeasible. Hence some Lagrangian heuristics are developed in order to find a good primal feasible solution, see [29] and [15]. More recently, Ghaddar et.al [14] used the Lagrangian relaxation to decompose a hydro plant scheduling problem by time periods and then found a high-quality feasible solution in a relatively short period of time.

Mixed integer programming (MIP) including the mixed integer linear programming (MILP) and mixed integer quadratic programming (MIQP) approaches has been naturally employed to handle the on-and-off switching status of generators. For example, Chang et.al [4] proposed an MILP formulation to yield a near optimal solution by using multiple piece-wise linear segments to approximate the objective function and constraints. Sawa et al. [32] relaxed the binary variables in an MIQP model to solve a security constrained unit commitment problem using quadratic programming methods.

Meta-heuristics, such as simulated annealing [9], genetic algorithm [21], ant colony [33], artificial neural network [34], and particle swarm optimization [6], have all been applied to solve various UC problems in recent years. In general a local optimal solution or near-optimal solution can be identified in a timely manner, but there usually lacks theoretical analysis on complexity and quality of solutions obtained.

Notice that, in practice, mixed approaches are commonly employed to treat UC problems. For example, Benders decomposition could be combined with Dantzig-Wolfe or Lagrangian approaches to obtain a cross-decomposition method which may perform better, see [24]. Also, Bosch et al. [38] proposed a decomposition method for their UC problem and then used dynamic programming methods to solve the corresponding subproblems. As more industrial requirements arise for more complicated applications, hybrid models [26] have been popular in which more than one approaches are applied to gain computational efficiency.

From the literature, we understand that general UC problems are complex and hard to solve. Effectively and efficiently solving large-scale SPUC problems (1.2) remains to be a challenging first-step toward handling general UC problems. Unlike most known works in the area, this paper intends to design a polynomial-time solvable approximation algorithm with theoretic error bounds to generate an approximate solution in a fast manner with guaranteed quality. Remember that the difficulty of problem (1.2) lies in the discontinuity and nonconvexity of $f_i(x_i)$ with $c_i > 0$ for each i. A natural idea is to convexify functions $f_i(x_i)$, $i = 1, \cdots, n$. Instead of considering the commonly used linear convex relaxation or quadratic convex relaxation, we notice that the conjugate transform of the conjugate transform of $f_i(x_i)$ becomes its tightest closed convex envelop for relaxation, e.g., see [10]. More interestingly, since the functions involved are quadratic functions, this convex relaxation enjoys having explicit expressions, so is the Lagrangian dual problem of the convex relaxation problem. Corresponding optimality conditions in explicit expressions can then be fully exploited to design a polynomial-time solvable approximation algorithm, and to derive and prove tight error bounds for quality assurance. Computational experiments are also conducted to validate our findings. To the best of our knowledge, this work is the first of its kind for solving UC problems.

The rest of the paper is structured as follows. For completeness of this paper, in Section 2, we include some results of our work [19]. In particular, we introduce a conjugate function based relaxation for problem (1.2), its Lagrangian dual problem, and optimality properties of both problems. Section 3 shows the main work of presenting an approximation algorithm for solving the SPUC problem. The approximation algorithm is proven to be of $\mathcal{O}(n \log n)$ worst-case computational complexity. Moreover, tight theoretical error bounds on the approximate solution are provided. In Section 4, computational experiments are conducted to validate the efficiency of the proposed algorithm for solving-large scale SPUC problems and to show the effect of some problem parameters on the solution quality. Concluding remarks are provided in Section 5.

2 Conjugate Function and Lagrangian Dual

The conjugate function of $f_i(x_i)$ is defined as $f_i^*(\lambda) = \sup_{x_i \in \mathbb{R}}(\lambda x_i - f_i(x_i))$, which is a convex function. Then the conjugate function of $f_i^*(\lambda)$, i.e., $f_i^{**}(x_i) = \sup_{\lambda \in \mathbb{R}}(x_i\lambda - f_i^*(\lambda))$, is the tightest convex relaxation of $f_i(x_i)$, see [10]. Notice that $f_i^{**}(x_i) \le f_i(x_i)$ is true in general. Moreover, $(f_i^{**})^*(\lambda) = \sup_{x_i \in \mathbb{R}}(\lambda x_i - f_i^{**}(x_i)) = f_i^*(\lambda)$ holds, see Chapter 11 of [30].

A convex relaxation of problem (1.2) becomes

$$\begin{aligned} min \quad & \sum_{i=1}^{n} f_i^{**}(x_i) \\ s.t. \quad & \sum_{i=1}^{n} x_i = d. \end{aligned} \tag{2.1}$$

The explicit form of $f_i^{**}(x_i), i = 1, \cdots, n$, can be easily computed as below.
If "$c_i = 0$", then

$$f_i^{**}(x_i) = \begin{cases} a_i x_i^2 + b_i x_i, & \text{for } x_i \in [0, u_i] \\ +\infty, & \text{for } x_i \notin [0, u_i]. \end{cases} \tag{2.2}$$

If "$c_i > 0, a_i = 0$" or "$c_i > 0, a_i > 0$ and $\sqrt{c_i/a_i} \ge u_i$", then

$$f_i^{**}(x_i) = \begin{cases} (a_i u_i + b_i + c_i/u_i)x_i, & \text{for } x_i \in [0, u_i] \\ +\infty, & \text{for } x_i \notin [0, u_i]. \end{cases} \tag{2.3}$$

If "$c_i > 0, a_i > 0$ and $\sqrt{c_i/a_i} < u_i$", then

$$f_i^{**}(x_i) = \begin{cases} (2\sqrt{a_i c_i} + b_i)x_i, & \text{for } x_i \in [0, \sqrt{c_i/a_i}) \\ a_i x_i^2 + b_i x_i + c_i, & \text{for } x_i \in [\sqrt{c_i/a_i}, u_i] \\ +\infty, & \text{for } x_i \notin [0, u_i]. \end{cases} \tag{2.4}$$

Accordingly, we divide $\mathcal{I} = \{1, \cdots, n\}$ into the following four index sets:

$$\begin{aligned} \mathcal{I}_1 &= \{i \in \mathcal{I}|\ c_i > 0, a_i > 0, \sqrt{c_i/a_i} < u_i\}, \\ \mathcal{I}_2 &= \{i \in \mathcal{I}|\ c_i > 0, a_i = 0\} \cup \{i \in \mathcal{I}|\ c_i > 0, a_i > 0, \sqrt{c_i/a_i} \geq u_i\}, \\ \mathcal{I}_3 &= \{i \in \mathcal{I}|\ c_i = 0, a_i > 0\}, \\ \mathcal{I}_4 &= \{i \in \mathcal{I}|\ c_i = 0, a_i = 0\}. \end{aligned} \tag{2.5}$$

In this paper, we may assume that $\mathcal{I}_1 \cup \mathcal{I}_2 \neq \emptyset$. Otherwise, problem (1.2) is a convex quadratic problem with linear constraints which can be solved easily. We may also assume that $b_i > 0, \forall i \in \mathcal{I}_4$. Otherwise, this generator involves no cost and can be omitted without affecting our analysis.

Lemma 1. *If $i \in \mathcal{I}_1$, then*

$$\begin{aligned} &f_i(x_i) > f_i^{**}(x_i), \text{ for } x_i \in (0, \sqrt{c_i/a_i}), \\ &f_i(x_i) = f_i^{**}(x_i), \text{ for } x_i = 0 \text{ or } x_i \in [\sqrt{c_i/a_i}, u_i]. \end{aligned} \tag{2.6}$$

If $i \in \mathcal{I}_2$, then

$$\begin{aligned} &f_i(x_i) > f_i^{**}(x_i), \text{ for } x_i \in (0, u_i), \\ &f_i(x_i) = f_i^{**}(x_i), \text{ for } x_i = 0 \text{ or } x_i = u_i. \end{aligned} \tag{2.7}$$

If $i \in \mathcal{I}_3 \cup \mathcal{I}_4$, then

$$f_i(x_i) = f_i^{**}(x_i), \text{ for } x_i \in [0, u_i]. \tag{2.8}$$

The above results can be shown easily and will be used for the gap analysis in Section 3. In addition, we can easily observe that for any given $d \in [0, \sum_{i=1}^n u_i]$, the optimal value of problem (1.1) and the optimal value of problem (2.1) are both finite and attainable.

Since $f_i^{**}(x_i) \leq f_i(x_i)$, for $i = 1, \cdots, n$, problem (2.1) provides a lower bound for problem (1.2), or problem (1.1) equivalently. To solve problem (2.1), we consider its Lagrangian dual problem. The Lagrangian function is

$$L(x, \lambda) = \sum_{i=1}^n f_i^{**}(x_i) + \lambda(d - \sum_{i=1}^n x_i), \tag{2.9}$$

then the Lagrangian dual function becomes

$$p(\lambda) = \min_{x \in \mathbb{R}^n} L(x, \lambda) = \lambda d - \sum_{i=1}^n \max_{x_i \in \mathbb{R}} (\lambda x_i - f_i^{**}(x_i)) = \lambda d - \sum_{i=1}^n (f_i^{**})^*(\lambda) = \lambda d - \sum_{i=1}^n f_i^*(\lambda).$$

Thus problem (2.1) has the following Lagrangian dual problem:

$$max_{\lambda\in\mathbb{R}} \quad p(\lambda) = \lambda d - \sum_{i=1}^{n} f_i^*(\lambda). \tag{2.10}$$

The strong duality theorem holds for the convex pair of problems (2.1) and (2.10), see [1]. Actually, problem (2.10) is also the Lagrangian dual of problem (1.2), but there may exist a gap due to the nonconvexity of $f_i(x_i), i = 1, \cdots, n$.

Notice that problem (2.10) is a concave maximization problem, whose necessary and sufficient optimality condition is

$$d \in \sum_{i=1}^{n} \partial f_i^*(\lambda), \tag{2.11}$$

where $\partial f_i^*(\lambda)$ is the subdifferential of f_i^* at λ. Since $\partial f_i^*(\lambda) = argmax_{x_i}(\lambda x_i - f_i^{**}(x_i))$, see Theorem 11.3 of [30], by (2.2), (2.3) and (2.4), we have the following results:

If $i \in \mathcal{I}_1$, then

$$\partial f_i^*(\lambda) = \begin{cases} \{0\}, & \text{for } \lambda < \lambda_i^1 \\ [0, \sqrt{c_i/a_i}], & \text{for } \lambda = \lambda_i^1 \\ \{(\lambda - b_i)/(2a_i)\}, & \text{for } \lambda_i^1 < \lambda \le \lambda_i^2 \\ \{u_i\}, & \text{for } \lambda > \lambda_i^2, \end{cases} \tag{2.12}$$

where $\lambda_i^1 = 2\sqrt{a_i c_i} + b_i$ and $\lambda_i^2 = 2a_i u_i + b_i$.

If $i \in \mathcal{I}_2 \cup \mathcal{I}_4$, then

$$\partial f_i^*(\lambda) = \begin{cases} \{0\}, & \text{for } \lambda < \lambda_i^1 \\ [0, u_i], & \text{for } \lambda = \lambda_i^1 \\ \{u_i\}, & \text{for } \lambda > \lambda_i^1, \end{cases} \tag{2.13}$$

where $\lambda_i^1 = a_i u_i + b_i + c_i/u_i$.

If $i \in \mathcal{I}_3$, then

$$\partial f_i^*(\lambda) = \begin{cases} \{0\}, & \text{for } \lambda < \lambda_i^1 \\ \{(\lambda - b_i)/(2a_i)\}, & \text{for } \lambda_i^1 \le \lambda \le \lambda_i^2 \\ \{u_i\}, & \text{for } \lambda > \lambda_i^2, \end{cases} \tag{2.14}$$

where $\lambda_i^1 = b_i$ and $\lambda_i^2 = 2a_i u_i + b_i$.

Consequently, $\partial f_i^*(\lambda)$ is an interval for "$i \in \mathcal{I}_4$ and $\lambda = \lambda_i^1$" and it is a singleton set for all $\lambda \in \mathbb{R}$ if $i \in \mathcal{I}_3$. This is the reason for us to separately define $\mathcal{I}_3$ and $\mathcal{I}_4$ before. Define $\sum_{i=1}^n \partial f_i^*(\lambda) := \{\sum_{i=1}^n z_i | z_i \in \partial f_i^*(\lambda)\}$. It is not difficult to check that $\sum_{i=1}^n \partial f_i^*(\lambda)$ is a singleton set if $\lambda \notin \{\lambda_i^1 | \ i \in \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_4\}$. Otherwise, it is a closed interval. Notice that $\sum_{i=1}^n \partial f_i^*(\lambda) \subset [0, \sum_{i=1}^n u_i]$ is a bounded closed set, we may define $l(\lambda) := \min \sum_{i=1}^n \partial f_i^*(\lambda)$ and $u(\lambda) := \max \sum_{i=1}^n \partial f_i^*(\lambda)$, where "min" and "max" refer to taking the minimum and maximum of the set $\sum_{i=1}^n \partial f_i^*(\lambda)$, respectively. Obviously, if $\sum_{i=1}^n \partial f_i^*(\lambda)$ is a singleton set, then $l(\lambda) = u(\lambda)$. Moreover, $l(\lambda') \ge u(\lambda)$ for any $\lambda' > \lambda$.

To give a more clear picture of $\sum_{i=1}^n \partial f_i^*(\lambda)$, we introduce the set:

$$\Lambda := \{\lambda_i^1, \lambda_i^2 | i \in \mathcal{I}_1 \cup \mathcal{I}_3\} \cup \{\lambda_i^1 | i \in \mathcal{I}_2 \cup \mathcal{I}_4\}. \tag{2.15}$$

Without loss of generality, we may write Λ as $\{\lambda_1, \cdots, \lambda_r\}$, where $r \le 2n$ and $\lambda_1 < \cdots < \lambda_r$. Then we have $l(\lambda_1) = 0$ and $u(\lambda_r) = \sum_{i=1}^n u_i$, since $0 \in \partial f_i^*(\lambda_1)$ and $u_i \in \partial f_i^*(\lambda_r)$ for any

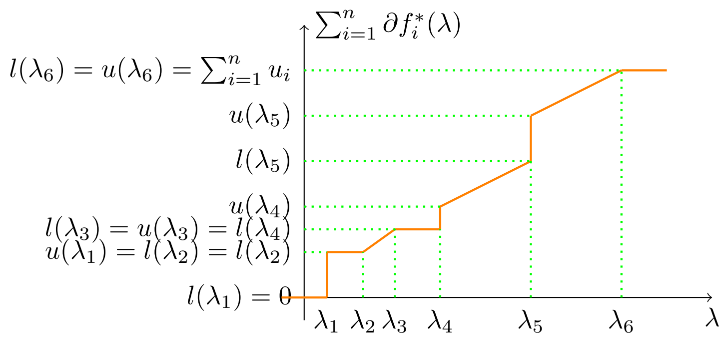

Figure 1: A typical figure of $\sum_{i=1}^{n} \partial f_i^*(\lambda)$

$i = 1, \cdots, n$. Consequently, we have $0 = l(\lambda_1) \le u(\lambda_1) \le \cdots \le l(\lambda_r) \le u(\lambda_r) = \sum_{i=1}^{n} u_i$. A typical figure of $\sum_{i=1}^{n} \partial f_i^*(\lambda)$ looks like Figure 1. A similar figure could be found in [13].

In the figure, $r = 6$ and we can see the points of $\lambda \in \Lambda$ are where $\sum_{i=1}^{n} \partial f_i^*(\lambda)$ is an interval, such as $\lambda_1, \lambda_4, \lambda_5$, or a turning point of $\sum_{i=1}^{n} \partial f_i^*(\lambda)$ when it is a singleton set, such as $\lambda_2, \lambda_3, \lambda_6$.

Lemma 2. *For any given $d \in [0, \sum_{i=1}^{n} u_i]$, the optimal value of problem (2.10) is finite and attainable.*

Proof: Since the strong duality theorem holds for problems (2.1) and (2.10) and the optimal value of problem (2.1) is finite, the optimal value of problem (2.10) is finite. Notice that $\bigcup_{\lambda \in \mathbb{R}} \sum_{i=1}^{n} \partial f_i^*(\lambda) = [0, \sum_{i=1}^{n} u_i]$. For any $d \in [0, \sum_{i=1}^{n} u_i]$, there must exist a $\bar{\lambda}$ such that $d \in \sum_{i=1}^{n} \partial f_i^*(\bar{\lambda})$. Then $\bar{\lambda}$ satisfies the necessary and sufficient optimality condition (2.11) of problem (2.10), thus the minimum of problem (2.10) can be attained, no matter the optimal solution is unique or not. □

With the optimality condition $d \in \sum_{i=1}^{n} \partial f_i^*(\lambda)$ in mind, we can identify optimal solutions of problems (2.10) and (2.1). The corresponding results of [19] can be rearranged into the next two theorems. Since more details can be found therein, the proofs are omitted here.

Theorem 1. *For any given $d \in [0, \sum_{i=1}^{n} u_i]$, if there exists a $\lambda_q \in \Lambda$ such that $d \in [l(\lambda_q), u(\lambda_q)]$, then $\lambda^* = \lambda_q$ is an optimal solution of problem (2.10). Otherwise, there must exist a $q \in \{1, \cdots, r-1\}$ such that $u(\lambda_q) < d < l(\lambda_{q+1})$, and the optimal solution of (2.10) is uniquely determined by*

$$\lambda^* = \frac{d - \sum_{j \in \bar{S}} \hat{x}_j + \sum_{j \in S} b_j/(2a_j)}{\sum_{j \in S} 1/(2a_j)}, \tag{2.16}$$

where $S = \{j \in \mathcal{I}_1 \cup \mathcal{I}_3 | \lambda_j^1 \le \lambda_q$ and $\lambda_j^2 \ge \lambda_{q+1}\}$, $\bar{S} = \{1, \cdots, n\} \backslash S$, and $\hat{x}_j = u_j$, if "$j \in \mathcal{I}_1 \cup \mathcal{I}_3$ and $\lambda_j^2 \le \lambda_q$" or "$j \in \mathcal{I}_2 \cup \mathcal{I}_4$ and $\lambda_j^1 \le \lambda_q$"; $\hat{x}_j = 0$, otherwise. Overall, an optimal solution of (2.10) can be obtained in $\mathcal{O}(n \log n)$ time.

Theorem 2. *For any given $d \in [0, \sum_{i=1}^{n} u_i]$ and an optimal solution λ^* of problem (2.10), x^* is optimal to problem (2.1) if and only if x^* solves the system*

$$\begin{cases} \sum_{i=1}^{n} x_i = d \\ x_i \in \partial f_i^*(\lambda^*), i = 1, \cdots, n. \end{cases} \tag{2.17}$$

For Theorem 2, if $\lambda^* \notin \{\lambda_i^1 | \, i \in \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_4\}$, then $\partial f_i^*(\lambda^*)$ is a singleton set and x_i can be uniquely determined for $i = 1, \cdots, n$, i.e., there is a unique feasible solution of system (2.17).

3 Approximation Algorithm

To provide further analysis of system (2.17), we partition $\mathcal{I} = \{1, \cdots, n\}$ by a dual optimal solution $\lambda^* \in \mathbb{R}$ such that

$$\mathcal{I}^a = \{i \in \mathcal{I}_1 \cup \mathcal{I}_2 | \ \lambda_i^1 = \lambda^*\}, \ \mathcal{I}^b = \{i \in \mathcal{I}_4 | \ \lambda_i^1 = \lambda^*\} \text{ and } \mathcal{I}^c = \mathcal{I} \backslash (\mathcal{I}^a \cup \mathcal{I}^b). \tag{3.1}$$

For convenience, for any given $\lambda \in \mathbb{R}$, if $\partial f_i^*(\lambda)$ is a singleton set, then we denote its unique element by $X_i(\lambda)$. Then system (2.17) with λ^* being given is equivalent to

$$\begin{cases} \sum_{i=1}^n x_i = d \\ x_i \in [0, t_i], i \in \mathcal{I}^a \\ x_i \in [0, u_i], i \in \mathcal{I}^b \\ x_i = X_i(\lambda^*), i \in \mathcal{I}^c, \end{cases} \tag{3.2}$$

where $t_i = \sqrt{c_i/a_i}$ for $i \in \mathcal{I}^a \cap \mathcal{I}_1$ and $t_i = u_i$ for $i \in \mathcal{I}^a \cap \mathcal{I}_2$. For any solution $\bar{x}$ of system (3.2), it is optimal to problem (2.1) and feasible for problem (1.2), thus $\sum_{i=1}^n f_i^{**}(\bar{x}_i)$ and $\sum_{i=1}^n f_i(\bar{x}_i)$ provide a lower bound and an upper bound for problem (1.2), respectively. And we have the following result by Lemma 1.

Remark 1. *For (3.1), if $i \in \mathcal{I}^a$, then $f_i(x_i) = f_i^{**}(x_i)$, for $x_i = 0$ or $x_i = t_i$, and $f_i(x_i) > f_i^{**}(x_i)$, for $x_i \in (0, t_i)$. If $i \in \mathcal{I}^b$, then $f_i(x_i) = f_i^{**}(x_i)$, for $x_i \in [0, u_i]$. If $i \in \mathcal{I}^c$, then $f_i(x_i) = f_i^{**}(x_i)$, for $x_i = X_i(\lambda^*)$.*

It is easy to check that if $\mathcal{I}^a = \mathcal{I}^b = \emptyset$, i.e., $\sum_{i=1}^n \partial f_i^*(\lambda^*)$ is a singleton set, then the unique solution $\bar{x}$ of system (3.2) is optimal to problem (1.2) since $\sum_{i=1}^n f_i(\bar{x}_i) = \sum_{i=1}^n f_i^{**}(\bar{x}_i)$. More generally, if $\bar{x}$ solves system (3.2) with $\bar{x}_i = 0$ or $\bar{x}_i = t_i$, $\forall i \in \mathcal{I}^a$, then $\sum_{i=1}^n f_i(\bar{x}_i) = \sum_{i=1}^n f_i^{**}(\bar{x}_i)$, thus $\bar{x}$ is optimal to problem (1.2). However, it is difficult to determine whether such a solution exits when $|\mathcal{I}^a|$, the cardinality of $\mathcal{I}^a$, is large.

Our idea is to find a solution $\bar{x}$ of system (3.2) with $\sum_{i=1}^n f_i(\bar{x}_i) - \sum_{i=1}^n f_i^{**}(\bar{x}_i)$ being small and then construct a better solution of problem (1.2) using $\bar{x}$. For convenience, we denote

$$\Delta = \sum_{i \in \mathcal{I}^b} u_i, \ d^* = d - \sum_{i \in \mathcal{I}^c} X_i(\lambda^*) \text{ and } g(x) = \sum_{i=1}^n f_i(x_i) - \sum_{i=1}^n f_i^{**}(x_i). \tag{3.3}$$

If $\mathcal{I}^a \neq \emptyset$, then, without loss of generality, we may write $\mathcal{I}^a = \{i_1, \cdots, i_{|\mathcal{I}^a|}\}$ satisfying $t_{i_1} \geq \cdots \geq t_{i_{|\mathcal{I}^a|}}$. Furthermore, if $d^* \in (0, \sum_{i \in \mathcal{I}^a} t_i]$, then we can identify a unique $\bar{p} \in \{1, \cdots, |\mathcal{I}^a|\}$ such that $\sum_{j=1}^{\bar{p}-1} t_{i_j} < d^*$ and $\sum_{j=1}^{\bar{p}} t_{i_j} \geq d^*$, where $\sum_{j=1}^{0} t_{i_j}$ is defined to be 0. If $\bar{p} > 1$, then define $I(\bar{p}) = \{i_1, \cdots, i_{\bar{p}-1}\}$. Otherwise, $I(\bar{p}) = \emptyset$. In addition, define

$$\Lambda^1 := \Lambda \cap \{\lambda_i^1 | \ i \in \mathcal{I}_1 \cup \mathcal{I}_2\} \tag{3.4}$$

to be a subset of Λ.

Theorem 3. *For an optimal solution λ^* of problem (2.10), if $\lambda^* \notin \Lambda^1$, then any solution $\bar{x}$ of system (3.2) is optimal to problems (1.2) and (2.1) with $g(\bar{x}) = 0$.*

Proof: Since $\lambda^* \notin \Lambda^1$, $\mathcal{I}^a = \emptyset$. For any solution $\bar{x}$ of system (3.2), since $f_i(\bar{x}_i) = f_i^{**}(\bar{x}_i)$, for $i \in \mathcal{I}^b \cup \mathcal{I}^c$, we have $g(\bar{x}) = 0$. Thus $\bar{x}$ is optimal to problems (1.2) and (2.1) with the equal objective value. □

Remark 2. *For an optimal solution* λ^* *of problem (2.10), if* $\lambda^* \notin \Lambda^1$ *and* $\lambda^* \notin \{\lambda_i^1 | \ i \in \mathcal{I}_4\}$*, then Theorem 3 reduces to Theorem 3 of [19], where a zero gap between problems (1.2) and (2.1) is claimed.*

Theorem 4. *For an optimal solution* λ^* *of problem (2.10), when* $\lambda^* \in \Lambda^1$*, if one of the following cases holds: (i)* $\{j \in \mathcal{I}^a | t_j > \Delta\} = \emptyset$*, (ii)* $\{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ *and* $0 \le d^* \le \Delta$*, and (iii)* $\{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ *and* $d^* \ge \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$*, then there is no gap between problems (1.2) and (2.1).*

Proof: $\mathcal{I}^a \neq \emptyset$ is obvious.

Case(i): If $\{j \in \mathcal{I}^a | t_j > \Delta\} = \emptyset$, then $\Delta \ge t_i > 0$, $\forall i \in \mathcal{I}^a$. If $d^* > \sum_{i \in \mathcal{I}^a} t_i$, then let $\bar{x}_i = t_i$ for $i \in \mathcal{I}^a$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$. Since system (3.2) is feasible, $0 < d^* - \sum_{i \in \mathcal{I}^a} t_i = d - \sum_{i \in \mathcal{I}^c} X_i(\lambda^*) - \sum_{i \in \mathcal{I}^a} t_i \le \Delta$. Let $\bar{x}_i = \frac{d^* - \sum_{i \in \mathcal{I}^a} t_i}{\Delta} u_i$ for $i \in \mathcal{I}^b$, then $\bar{x}$ solves system (3.2) with $g(\bar{x}) = 0$ by Remark 1, thus there is no gap. If $d^* = 0$, then let $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \cup \mathcal{I}^b$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, which is a solution of system (3.2) with a zero gap between problems (1.2) and (2.1). Otherwise, $0 < d^* \le \sum_{i \in \mathcal{I}^a} t_i$ and the corresponding $I(\bar{p})$ and $i_{\bar{p}}$ can be determined uniquely. If $\sum_{i \in I(\bar{p}) \cup \{i_{\bar{p}}\}} t_i = d^*$, then let $\bar{x}_i = t_i$ for $i \in I(\bar{p}) \cup \{i_{\bar{p}}\}$, $\bar{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, thus there is no gap. Otherwise, $0 < d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}} \le \Delta$, and we let $\bar{x}_i = t_i$ for $i \in I(\bar{p})$, $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \backslash I(\bar{p})$, $\bar{x}_i = \frac{d^* - \sum_{i \in I(\bar{p})} t_i}{\Delta} u_i$ for $i \in \mathcal{I}^b$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, then there is no gap.

For case (ii) and case (iii), since $\{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$, we have $\Delta < \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$.

Case(ii): If $0 \le d^* \le \Delta$ and $\Delta = 0$ (i.e., $\mathcal{I}^b = \emptyset$), then $d^* = 0$ and we let $\bar{x}_i = 0$ for $i \in \mathcal{I}^a$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, thus there is no gap. If $0 \le d^* \le \Delta$ and $\Delta > 0$, then let $\bar{x}_i = 0$ for $i \in \mathcal{I}^a$, $\bar{x}_i = \frac{d^*}{\Delta} u_i$ for $i \in \mathcal{I}^b$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, thus there is no gap.

Case(iii): If $d^* = \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$, then let $\bar{x}_i = t_i$ for $i \in \{j \in \mathcal{I}^a | t_j > \Delta\}$, $\bar{x}_i = 0$ for $i \in \{j \in \mathcal{I}^a | t_j \le \Delta\} \cup \mathcal{I}^b$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, thus there is no gap. If $d^* > \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$ and furthermore $d^* > \sum_{i \in \mathcal{I}^a} t_i$, then there is no gap, see the proof of case (i). If $\sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i < d^* \le \sum_{i \in \mathcal{I}^a} t_i$, then we have $t_{i_{\bar{p}}} \le \Delta$. As shown in the proof of case (i), there is no gap. □

Theorem 5. *For an optimal solution* λ^* *of problem (2.10), when* $\lambda^* \in \Lambda^1, \{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ *and* $\Delta < d^* < \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$*, if (i)* $\sum_{i \in I(\bar{p}) \cup \{i_{\bar{p}}\}} t_i = d^*$ *or (ii)* $d^* - \sum_{i \in I(\bar{p})} t_i \le \Delta$*, then there is no gap between problems (1.2) and (2.1).*

Proof: When $\lambda^* \in \Lambda^1, \{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ and $\Delta < d^* < \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$, $t_{i_{\bar{p}}} > \Delta$ must hold. Otherwise, $\sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i \le \sum_{i \in I(\bar{p})} t_i < d^*$, which contradicts to $d^* < \sum_{i \in \{j \in \mathcal{I}^a | t_j > \Delta\}} t_i$.

In case (i), let $\bar{x}_i = t_i$ for $i \in I(\bar{p}) \cup \{i_{\bar{p}}\}$, $\bar{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$. Then we know there is no gap. In case (ii), since $0 < d^* - \sum_{i \in I(\bar{p})} t_i \le \Delta$, we may let $\bar{x}_i = t_i$ for $i \in I(\bar{p})$, $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \backslash I(\bar{p})$, $\bar{x}_i = \frac{d^* - \sum_{i \in I(\bar{p})} t_i}{\Delta} u_i$ for $i \in \mathcal{I}^b$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$. Then we know there is no gap. □

Putting Theorems 3, 4 and 5 together, we need to focus on the only remaining case of "$\lambda^* \in \Lambda^1, \{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ and $\Delta < d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}}$", where a positive gap between problems (1.2) and (2.1) may arise. In this case, for any solution $\bar{x}$ of system (3.2) with $\bar{x}_i = t_i$ for $i \in I(\bar{p})$ and $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$, we would like to determine $\bar{x}_i, i \in \mathcal{I}^b \cup \{i_{\bar{p}}\}$, to minimize the gap $g(\bar{x})$.

Theorem 6. *For an optimal solution λ^* of problem (2.10), if $\lambda^* \in \Lambda^1, \{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ and $\Delta < d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}}$, then for any solution $\bar{x}$ of system (3.2) with $\bar{x}_i = t_i$ for $i \in I(\bar{p})$ and $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$, we have $g(\bar{x}) = a_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}}^2 + b_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}} + c_{i_{\bar{p}}} - \lambda^* \bar{x}_{i_{\bar{p}}}$, which achieves its minimum when $\bar{x}_{i_{\bar{p}}} = d^* - \sum_{i \in I(\bar{p})} t_i$ and $\bar{x}_i = 0$ for $i \in \mathcal{I}^b$.*

Proof: Since $\bar{x}_i = t_i$ for $i \in I(\bar{p})$ and $\bar{x}_i = 0$ for $i \in \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$, we have $f_i(\bar{x}_i) = f_i^{**}(\bar{x}_i)$ for $i \in \mathcal{I}^a \backslash \{i_{\bar{p}}\}$ by Remark 1. Moreover, $f_i(\bar{x}_i) = f_i^{**}(\bar{x}_i)$ for $i \in \mathcal{I}^b$ and $f_i(\bar{x}_i) = f_i^{**}(\bar{x}_i)$ for $i \in \mathcal{I}^c$. Therefore, $g(\bar{x}) = f_{i_{\bar{p}}}(\bar{x}_{i_{\bar{p}}}) - f_{i_{\bar{p}}}^{**}(\bar{x}_{i_{\bar{p}}})$. Since $\sum_{i \in \mathcal{I}^b} u_i = \Delta < d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}}$, we know $\bar{x}_{i_{\bar{p}}} \in (0, t_{i_{\bar{p}}})$ and $f_{i_{\bar{p}}}(\bar{x}_{i_{\bar{p}}}) = a_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}}^2 + b_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}} + c_{i_{\bar{p}}}$. Since $i_{\bar{p}} \in \mathcal{I}^a$, we have $\lambda_{i_{\bar{p}}}^1 = \lambda^*$. If $i_{\bar{p}} \in \mathcal{I}_1$, from (2.4), we have $f_{i_{\bar{p}}}^{**}(\bar{x}_{i_{\bar{p}}}) = (2\sqrt{a_{i_{\bar{p}}} c_{i_{\bar{p}}}} + b_{i_{\bar{p}}}) \bar{x}_{i_{\bar{p}}} = \lambda_{i_{\bar{p}}}^1 \bar{x}_{i_{\bar{p}}} = \lambda^* \bar{x}_{i_{\bar{p}}}$. If $i_{\bar{p}} \in \mathcal{I}_2$, from (2.3), we have $f_{i_{\bar{p}}}^{**}(\bar{x}_{i_{\bar{p}}}) = (a_{i_{\bar{p}}} u_{i_{\bar{p}}} + b_{i_{\bar{p}}} + c_{i_{\bar{p}}}/u_{i_{\bar{p}}}) \bar{x}_{i_{\bar{p}}} = \lambda_{i_{\bar{p}}}^1 \bar{x}_{i_{\bar{p}}} = \lambda^* \bar{x}_{i_{\bar{p}}}$. Hence $g(\bar{x}) = f_{i_{\bar{p}}}(\bar{x}_{i_{\bar{p}}}) - f_{i_{\bar{p}}}^{**}(\bar{x}_{i_{\bar{p}}}) = a_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}}^2 + b_{i_{\bar{p}}} \bar{x}_{i_{\bar{p}}} + c_{i_{\bar{p}}} - \lambda^* \bar{x}_{i_{\bar{p}}}$.

To minimize the gap $g(\bar{x})$, we consider the following optimization problem:

$$\begin{aligned} min \quad & a_{i_{\bar{p}}} z^2 + b_{i_{\bar{p}}} z + c_{i_{\bar{p}}} - \lambda^* z \\ s.t. \quad & z + \sum_{i \in I(\bar{p})} t_i \leq d^* \\ & z + \sum_{i \in I(\bar{p})} t_i + \Delta \geq d^*, \end{aligned} \tag{3.5}$$

where the decision variable $z \in \mathbb{R}$ determines $\bar{x}_{i_{\bar{p}}}$. If $a_{i_{\bar{p}}} = 0$, then $i_{\bar{p}} \in \mathcal{I}_2$ and the objective is a linear function. Since $b_{i_{\bar{p}}} - \lambda^* = b_{i_{\bar{p}}} - (b_{i_{\bar{p}}} + c_{i_{\bar{p}}}/u_{i_{\bar{p}}}) = -c_{i_{\bar{p}}}/u_{i_{\bar{p}}} < 0$, the optimal value is obtained at $z = d^* - \sum_{i \in I(\bar{p})} t_i$. If $a_{i_{\bar{p}}} > 0$, then the objective is a convex quadratic function with the axis of symmetry being

$$z = \frac{\lambda^* - b_{i_{\bar{p}}}}{2a_{i_{\bar{p}}}} = \begin{cases} \sqrt{\frac{c_{i_{\bar{p}}}}{a_{i_{\bar{p}}}}}, & \text{for } i_{\bar{p}} \in \mathcal{I}_1 \\ \frac{u_{i_{\bar{p}}}}{2} + \frac{c_{i_{\bar{p}}}}{2a_{i_{\bar{p}}} u_{i_{\bar{p}}}}, & \text{for } i_{\bar{p}} \in \mathcal{I}_2. \end{cases}$$

Notice that $t_{i_{\bar{p}}} = \sqrt{\frac{c_{i_{\bar{p}}}}{a_{i_{\bar{p}}}}}$ for $i_{\bar{p}} \in \mathcal{I}_1$ and $t_{i_{\bar{p}}} = u_{i_{\bar{p}}} \leq \frac{u_{i_{\bar{p}}}}{2} + \frac{c_{i_{\bar{p}}}}{2a_{i_{\bar{p}}} u_{i_{\bar{p}}}}$ for $i_{\bar{p}} \in \mathcal{I}_2$. Since $d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}}$, we have $d^* - \sum_{i \in I(\bar{p})} t_i < \frac{\lambda^* - b_{i_{\bar{p}}}}{2a_{i_{\bar{p}}}}$ no matter $i_{\bar{p}} \in \mathcal{I}_1$ or $i_{\bar{p}} \in \mathcal{I}_2$. Thus the feasible region of problem (3.5) lies on the left-hand side of the axis of symmetry and the minimum is attained at $z = d^* - \sum_{i \in I(\bar{p})} t_i$. □

Therefore, in the case of "$\lambda^* \in \Lambda^1, \{j \in \mathcal{I}^a | t_j > \Delta\} \neq \emptyset$ and $\Delta < d^* - \sum_{i \in I(\bar{p})} t_i < t_{i_{\bar{p}}}$", to achieve a small gap $g(\bar{x})$, we may let $\bar{x}_i = t_i$ for $i \in I(\bar{p})$, $\bar{x}_{i_{\bar{p}}} = d^* - \sum_{i \in I(\bar{p})} t_i$, $\bar{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$ and $\bar{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$. Figure 2 summarizes our logic of gap analysis for us to propose an approximation algorithm.

Approximation Algorithm (AA):
Step 1: Input n, d and a_i, b_i, c_i, u_i, for $i = 1, \cdots, n$. If $d > \sum_{i=1}^n u_i$, then terminate the algorithm with an output "infeasibility."
Step 2: Obtain $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \mathcal{I}_4$ as (2.5) and Λ as (2.15) and compute $l(\lambda_i), u(\lambda_i)$, $i = 1, \cdots, r$. Find a q such that $d \in [l(\lambda_q), u(\lambda_q)]$ or $d \in (u(\lambda_q), u(\lambda_{q+1}))$. If $d \in [l(\lambda_q), u(\lambda_q)]$, then let $\lambda^* = \lambda_q$. If $d \in (u(\lambda_q), u(\lambda_{q+1}))$, then compute λ^* by (2.16).
Step 3: Construct Λ^1 as (3.4). If $\lambda^* \notin \Lambda^1$ and $\lambda^* \notin \{\lambda_i^1 | \, i \in \mathcal{I}_4\}$, then let $\tilde{x}_i = X(\lambda^*)$ for $i = 1, \cdots, n$, and go to Step 8. If $\lambda^* \notin \Lambda^1$ and $\lambda^* \in \{\lambda_i^1 | \, i \in \mathcal{I}_4\}$, then let $\mathcal{J} = \{j | \, \lambda^* = \lambda_i^1, i \in \mathcal{I}_4\}$

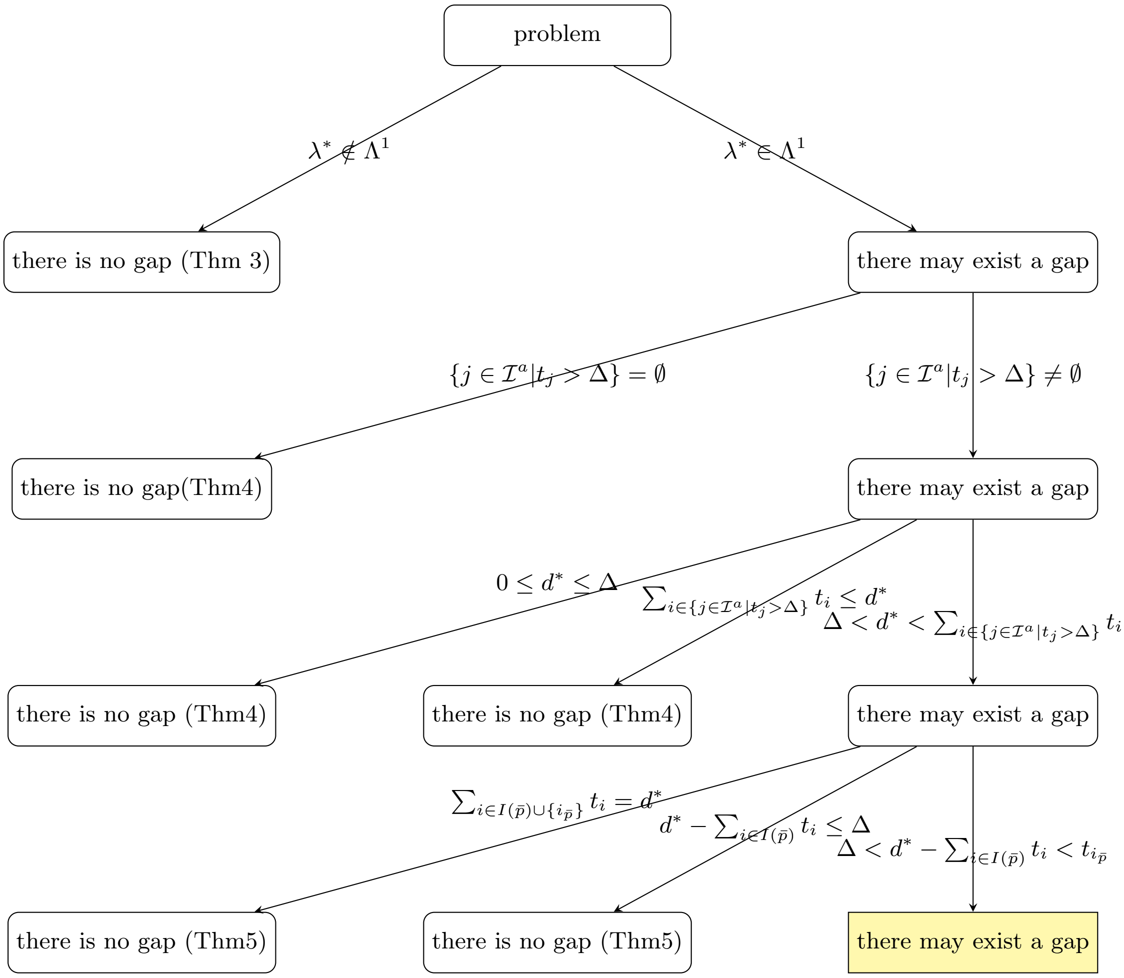

Figure 2: Logic diagram of gap analysis

and let $\tilde{x}_i = X_i(\lambda^*)$ for $i \notin \mathcal{J}$, $\tilde{x}_i = \frac{d-\sum_{i\notin\mathcal{J}} X_i(\lambda^*)}{\sum_{i\in\mathcal{J}} u_i} u_i$ for $i \in \mathcal{J}$, and go to Step 8. If $\lambda^* \in \Lambda^1$, then construct $\mathcal{I}^a$,$\mathcal{I}^b$ and $\mathcal{I}^c$ as (3.1), and compute Δ and d^* as (3.3).

Step 4: If $d^* > \sum_{i\in\mathcal{I}^a} t_i$, then let $\tilde{x}_i = t_i$ for $i \in \mathcal{I}^a$, $\tilde{x}_i = \frac{d^*-\sum_{i\in\mathcal{I}^a} t_i}{\Delta} u_i$ for $i \in \mathcal{I}^b$, $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, and go to Step 8. If $d^* = 0$, then let $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, $\tilde{x}_i = 0$ for $i \notin \mathcal{I}^c$, and go to Step 8.

Step 5: Determine $\bar{p}, i_{\bar{p}}$ and $I(\bar{p})$. If $\sum_{i\in I(\bar{p})\cup\{i_{\bar{p}}\}} t_i = d^*$, then let $\tilde{x}_i = t_i$ for $i \in I(\bar{p})\cup\{i_{\bar{p}}\}$, $\tilde{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$, $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, and go to Step 8. If $d^* - \sum_{i\in I(\bar{p})} t_i \le \Delta$, then let $\tilde{x}_i = t_i$ for $i \in I(\bar{p})$, $\tilde{x}_i = 0$ for $i \in \mathcal{I}^a \backslash I(\bar{p})$, $\tilde{x}_i = \frac{d^*-\sum_{i\in I(\bar{p})} t_i}{\Delta} u_i$ for $i \in \mathcal{I}^b$, $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, and go to Step 8.

Step 6: Let $\tilde{x}_i = t_i$ for $i \in I(\bar{p})$, $\tilde{x}_{i_{\bar{p}}} = d^* - \sum_{i\in I(\bar{p})} t_i$, $\tilde{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash (I(\bar{p}) \cup \{i_{\bar{p}}\})$ and $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$. If $I_3 \cup I_4 = \emptyset$ or $\sum_{i\in I_3\cup I_4} u_i < d$, then go to Step 8.

Step 7: Solve problem (1.2) with $x_i = 0$ for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$ to yield an optimal solution x' and the optimal value v'. If $\sum_{i=1}^n f_i(\tilde{x}_i) > v'$, then let $\tilde{x} = x'$.

Step 8: Terminate the algorithm and output the approximate solution $\tilde{x}$.

Remark 3. *In Step 2, q may be obtained in $\mathcal{O}(\log n)$* time by using the bisection method. In Step 7, since $x_i = 0$ for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$, we could omit the information of generator i, for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$, and obtain a new problem only involving the remaining $n - |\mathcal{I}_1 \cup \mathcal{I}_2|$ generators. To solve this new problem, we could carry out the algorithm from Step 1 and the corresponding optimal solution will be obtained before Step 4.

Remark 4. *The approximate solution generated before Step 7 is optimal to the relaxation problem (2.1). The purpose of Step 7 is to provide a possibly better approximate solution which is not necessarily subject to system (3.2).*

Theorem 7. *The worst-case computational complexity of the proposed AA is $\mathcal{O}(n \log n)$.*

Proof: It requires $\mathcal{O}(n)$ computational work to implement Step 1. In Step 2, a dual optimal solution λ^* of problem (2.10) can be derived in $\mathcal{O}(n \log n)$ time, see Theorem 1. In Step 3, if $\lambda^* \notin \Lambda^1$, then it takes $\mathcal{O}(n)$ work to construct a solution $\tilde{x}$. If $\lambda^* \in \Lambda^1$, then it takes $\mathcal{O}(n \log n)$ work to obtain $\mathcal{I}^a$,$\mathcal{I}^b$ and $\mathcal{I}^c$, since sorting $\mathcal{I}^a$ can be done in $\mathcal{O}(n \log n)$ time. Thus it takes $\mathcal{O}(n \log n)$ work to implement Step 3. In addition, it requires $\mathcal{O}(n)$ computational work to carry out Step 4 and Step 6. In Step 5, $\bar{p}, i_{\bar{p}}$ and $I(\bar{p})$ can be determined in $\mathcal{O}(n)$ time and other operations require $\mathcal{O}(n)$ work. In Step 7, by Remark 3, x' and v' can be obtained in $\mathcal{O}(n \log n)$ time. In conclusion, the worst-case computational complexity of the algorithm is $\mathcal{O}(n \log n)$. □

Notice that the approximate solution generated before Step 6 is indeed optimal. In Step 6, we have $g(\tilde{x}) = a_{i_{\bar{p}}}(d^* - \sum_{i\in I(\bar{p})} t_i)^2 + (b_{i_{\bar{p}}} - \lambda^*)(d^* - \sum_{i\in I(\bar{p})} t_i) + c_{i_{\bar{p}}}$, see Theorem 6. If we refine $\tilde{x}$ in Step 7, then the gap becomes smaller. Particularly, if v' equals to the optimal value of problem (2.1), then x' is optimal to problem (1.2), thus the output approximate solution $\tilde{x}$ is optimal to problem (1.2).

Denote the optimal value of problem (1.2) by f^* and the objective of problem (1.2) at the output approximate solution $\tilde{x}$ by $\tilde{f} = \sum_{i=1}^n f_i(\tilde{x}_i)$, then we have

$$p(\lambda^*) \le f^* \le \tilde{f} \le p(\lambda^*) + a_{i_{\bar{p}}}(d^* - \sum_{i\in I(\bar{p})} t_i)^2 + (b_{i_{\bar{p}}} - \lambda^*)(d^* - \sum_{i\in I(\bar{p})} t_i) + c_{i_{\bar{p}}}.$$

Consequently,

$$\begin{aligned}\tilde f - f^* &\le a_{i_{\bar p}}(d^* - \sum_{i\in I(\bar p)} t_i)^2 + (b_{i_{\bar p}} - \lambda^*)(d^* - \sum_{i\in I(\bar p)} t_i) + c_{i_{\bar p}} \\ &= \begin{cases} a_{i_{\bar p}}(d^* - \sum_{i\in I(\bar p)} t_i)^2 - 2\sqrt{a_{i_{\bar p}} c_{i_{\bar p}}}(d^* - \sum_{i\in I(\bar p)} t_i) + c_{i_{\bar p}}, & \text{for } i_{\bar p} \in \mathcal{I}_1 \\ a_{i_{\bar p}}(d^* - \sum_{i\in I(\bar p)} t_i)^2 - (a_{i_{\bar p}} u_{i_{\bar p}} + c_{i_{\bar p}}/u_{i_{\bar p}})(d^* - \sum_{i\in I(\bar p)} t_i) + c_{i_{\bar p}}, & \text{for } i_{\bar p} \in \mathcal{I}_2 \end{cases} \\ &\le c_{i_{\bar p}}.\end{aligned}$$

The last inequality comes from that $0 \le \Delta < d^* - \sum_{i\in I(\bar p)} t_i < t_{i_{\bar p}}$ and $a_{i_{\bar p}} z^2 + (b_{i_{\bar p}} - \lambda^*)z + c_{i_{\bar p}}$ is a decreasing function of z over $[0, t_{i_{\bar p}}]$ for both $i_{\bar p} \in \mathcal{I}_1$ and $i_{\bar p} \in \mathcal{I}_2$. It is easy to check that $\tilde f - f^*$ approaches 0 when $d^* - \sum_{i\in I(\bar p)} t_i$ approaches $t_{i_{\bar p}}$ and $\tilde f - f^*$ approaches $c_{i_{\bar p}}$ when "$d^* - \sum_{i\in I(\bar p)} t_i$ approaches 0 and $\Delta = 0$".

As analyzed before, if the approximate solution $\tilde x$ is generated by Steps 3, 4 or 5, then it is optimal with $\tilde f = f^*$. If Step 6 is activated, then $\tilde f - f^*$ may be positive but less than $c_{i_{\bar p}}$, depending on $i_{\bar p}$. When d changes, $i_{\bar p}$ may change even if other parameters remain the same. In addition, we are unlikely to know $i_{\bar p}$ beforehand. Thus we are interested in finding a bound depending only on the original information of the problem.

Define the absolute error as $A_E = |\tilde f - f^*| = \tilde f - f^*$ and relative error as $R_E = \frac{\tilde f - f^*}{f^*}$, for $f^* > 0$, then we have the following result:

Theorem 8. *Let $\tilde x$ be the approximate solution provided by AA, then*

$$A_E \le \max_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}. \tag{3.6}$$

If $f^* = 0$, then $A_E = 0$. Otherwise, R_E is well defined and

$$R_E \le \frac{\max_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}}{\min_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}}. \tag{3.7}$$

Moreover, the bounds for A_E and R_E are both tight.

Proof: Notice that $c_{i_{\bar p}} \le \max_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}$ since $i_{\bar p} \in \mathcal{I}^a \subset \mathcal{I}_1 \cup \mathcal{I}_2$, thus (3.6) is obvious. If $f^* = 0$, then $d = 0$. Thus $\tilde x = 0$ is the unique solution of problem (1.2). Then we have $\tilde f = 0$ and $A_E = 0$. If $f^* > 0$, then R_E is well defined. Since $\mathcal{I}_1 \cup \mathcal{I}_2 \ne \emptyset$, we have $\min_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\} > 0$ for (3.7). If there exits an optimal solution x^* of problem (1.2) with $x_i^* = 0$ for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$, then $v' = f^*$ and $\tilde x$ must be optimal with $\tilde f = v' = f^*$. Thus (3.7) holds. Otherwise, for x^* being optimal to problem (1.2), at least one $x_i^* > 0$ for $i \in \mathcal{I}_1\cup\mathcal{I}_2$, and hence $f^* = \sum_{i=1}^n f_i(x_i^*) \ge \min_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}$. Combining with $\tilde f - f^* \le \max_{i\in\mathcal{I}_1\cup\mathcal{I}_2}\{c_i\}$, we obtain (3.7).

To prove the tightness of the two bounds, consider problem (1.2) with $d = 1+\epsilon$ where $\epsilon > 0$ is sufficiently small and $f_1(0) = 0$, $f_1(x_1) = \frac{1}{2}$, for $0 < x_1 \le 1$ and $f_1(x_1) = +\infty$, otherwise; $f_2(x_2) = 10x_2$, for $x_2 \in [0,1]$ and $f_2(x_2) = +\infty$, otherwise; $f_3(0) = 0$, $f_3(x_3) = x_3^2 + x_3 + 1$, for $0 < x_3 \le 3$, $f_3(x_3) = +\infty$, otherwise. Then $\mathcal{I}_1 = \{3\}, \mathcal{I}_2 = \{1\}, \mathcal{I}_3 = \emptyset$ and $\mathcal{I}_4 = \{2\}$. Moreover, $\sum_{i=1}^3 \partial f_i^*(\lambda)$ is shown below.

From the figure, we can see $\lambda^* = 3$ since $d = 1+\epsilon$ with ϵ being sufficiently small. Thus we have $\mathcal{I}^a = \{3\}, \mathcal{I}^b = \emptyset, \mathcal{I}^c = \{1,2\}$ and $X_1(\lambda^*) = X_1(3) = 1, X_2(\lambda^*) = X_2(3) = 0$. Then the approximate solution generated by the proposed algorithm is $\tilde x = (\tilde x_1, \tilde x_2, \tilde x_3) = (1, 0, \epsilon)$ with $\tilde f = \epsilon^2+\epsilon+1.5$ and the optimal solution is $x^* = (x_1^*, x_2^*, x_3^*) = (1, \epsilon, 0)$ with $f^* = 10\epsilon+0.5$. Hence $\tilde f - f^* = \epsilon^2 - 9\epsilon + 1 \le \max\{c_1, c_3\} = \max\{0.5, 1\} = 1$ and $\frac{\tilde f - f^*}{f^*} = \frac{\epsilon^2 - 9\epsilon+1}{10\epsilon+0.5} \le \frac{\max\{0.5,1\}}{\min\{0.5,1\}} = 2$. As $\epsilon \to 0$, $\tilde f - f^* \to 1$ and $\frac{\tilde f - f^*}{f^*} \to 2$. Thus both bounds of (3.6) and (3.7) are tight. □

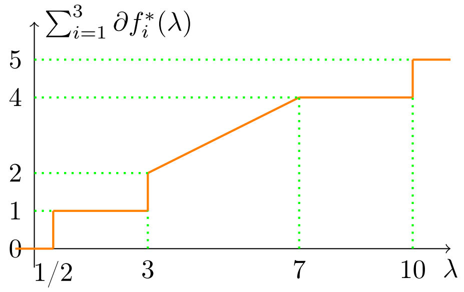

Figure 3: $\sum_{i=1}^{3} \partial f_i^*(\lambda)$

Remark 5. *As $d \to +\infty$, $f^* \to +\infty$ and $R_E \to 0$.*

Since the error bounds could be tight, we may want to relax the feasibility of system (3.2) for a better approximate solution. This idea leads to the following modification of Step 6:

"Let $\tilde{x}_i = t_i$ for $i \in I(\bar{p})$, $\tilde{x}_i = 0$ for $i \in \mathcal{I}^b \cup \mathcal{I}^a \backslash I(\bar{p})$ and $\tilde{x}_i = X_i(\lambda^*)$ for $i \in \mathcal{I}^c$, and construct $\mathcal{P} = \{j \mid u_j \geq \tilde{x}_j + d^* - \sum_{i \in I(\bar{p})} t_i\}$. Select any k from $argmin_{j \in \mathcal{P}}(a_j(\tilde{x}_j + d^* - \sum_{i \in I(\bar{p})} t_i)^2 + b_j(\tilde{x}_j + d^* - \sum_{i \in I(\bar{p})} t_i) + c_j - (a_j \tilde{x}_j^2 + b_j \tilde{x}_j + c_j \tilde{y}_j))$ where $\tilde{y}_j = 0$ if $\tilde{x}_j = 0$ and $\tilde{y}_j = 1$ if $\tilde{x}_j > 0$, and then add $d^* - \sum_{i \in I(\bar{p})} t_i$ to $\tilde{x}_k$. If $I_3 \cup I_4 = \emptyset$ or $\sum_{i \in I_3 \cup I_4} u_i < d$, then go to Step 8."

In other words, instead of allocating $d^* - \sum_{i \in I(\bar{p})} t_i$ to $i_{\bar{p}}$, we may check all generators which still have enough capacity to accommodate $d^* - \sum_{i \in I(\bar{p})} t_i$ and choose the one with the minimum increased cost.

Remark 6. *With the modified Step 6, the worst-case computational complexity and error bounds (3.6) and (3.7) of the proposed AA remain valid.*

Notice that the original problem (1.1) is a mixed-integer quadratic program, which is NP-hard and is unlikely to be solved explicitly in polynomial time unless P=NP. The proposed AA can be implemented in $\mathcal{O}(n \log n)$ time with guaranteed quality bounds. In the next section, we conduct computational experiments to show its efficiency.

4 Computational Experiments

In this section, we conduct computational experiments to show the efficiency of the proposed AA for solving large-scale problems.

In our experiments, we set n =100, 200, 300, 400, 500 or 1,000. To be consistent with the parameter setting of [19], a_i, b_i, c_i and u_i, for $i = 1, \cdots, n$, are randomly sampled from [0, 0.05], [20, 80], [200, 1000] and [100, 300], respectively, and d is randomly sampled from $[0, \sum_{i=1}^{n} u_i]$ to assure feasibility. For each n, we generate 1,000 instances. In each instance, the relative error is computed using the solution provided by the proposed AA and the optimal solution provided by the Branch-and-Bound Algorithm (BBA) in [19]. When the relative error R_E is less than 10^{-6}, we consider the approximation solution to be "optimal".

Table 1 shows the number of instances with an optimal solution being found (**optimal**), the distribution of R_E in the intervals of [0, 0.001), [0.001, 0.002), [0.002, 0.003), [0.003, 0.004), [0.004, 0.005), [0.005, 0.006), [0.006, 0.007), [0.007, 0.008), [0.008, 0.009), [0.009, 0.01) and

[0.01,+∞), the maximum (**max**), i.e., the worst-case R_E, the mean (**mean**) and the variance (**var**) of the 1,000 relative errors provided by the proposed algorithm for each n. Moreover, the average running time (CPU second) of the proposed Approximation Algorithm (**AA**) and the Branch-and-Bound Algorithm (**BBA**) are provided. Since the Cplex solver requires much longer time than the former two algorithms, we only conduct 10 random instances to obtain the average running time (**Cplex**) for each n. Notice that the Cplex solver is unable to solve problems when $n = 1000$.

value \ distribution / n	optimal	[0, 0.001)	[0.001, 0.002)	[0.002, 0.003)	[0.003, 0.004)	[0.004, 0.005)	[0.005, 0.006)	[0.006, 0.007)	[0.007, 0.008)	[0.008, 0.009)	[0.009, 0.01)	≥ 0.01	max	mean	var	AA(s)	BBA(s)	Cplex(s)
100	330	964	26	4	0	0	3	0	2	0	0	1	1.40×10^{-2}	1.78×10^{-4}	4.73×10^{-7}	4.06×10^{-4}	2.70×10^{-3}	5.79×10^{-2}
200	319	982	10	5	1	0	1	0	0	0	0	1	1.14×10^{-2}	1.03×10^{-4}	2.34×10^{-7}	4.59×10^{-4}	5.14×10^{-3}	4.83×10^{-1}
300	375	992	4	2	0	1	0	0	1	0	0	0	7.03×10^{-3}	5.96×10^{-5}	9.46×10^{-8}	6.06×10^{-4}	7.82×10^{-3}	2.43×10^{1}
400	355	995	2	1	0	1	1	0	0	0	0	0	5.30×10^{-3}	4.50×10^{-5}	5.71×10^{-8}	7.75×10^{-4}	1.04×10^{-2}	1.81×10^{3}
500	341	993	3	1	2	0	0	1	0	0	0	0	6.75×10^{-3}	4.74×10^{-5}	8.97×10^{-8}	9.40×10^{-4}	1.32×10^{-2}	1.13×10^{4}
1000	414	998	2	0	0	0	0	0	0	0	0	0	1.47×10^{-3}	1.52×10^{-5}	5.71×10^{-9}	1.65×10^{-3}	2.41×10^{-2}	—

Table 1: Relative error R_E of random instances

From Table 1, we can observe that:
(i) The computational time of the proposed AA is pretty flat as the complexity analysis indicates.
(ii) Since $c_i, i = 1, \cdots, n$, is randomly sampled from [200,1000], the bound for R_E is 5. But our experiments show that R_E is in general very low and the distribution of R_E is very light-tailed. This indicates the quality of the approximate solution obtained by the proposes AA.
(iii) As n goes large, R_E of more instances tend to be 0. Similarly, the worst-case R_E approaches 0. This observation is in line with Remark 5.
(iv) The proposed AA runs about 10 times faster than BBA and 10^2 to 10^8 times faster than using the Cplex solver. This indicates the computational efficiency of the proposed AA for solving large-scale UC problems.

To further study the effect of n on the worst-case R_E, we conduct ten groups of tests with n =100, 200, 300, 400, 500, 600, 700, 800, 900 or 1,000. For each n, we do ten experiments by carrying out 2,000 random instances in each experiment. We obtain the worst-case R_E of each experiment and then compute the mean of the 10 worst-case R_E for each n, which is plotted in Figure 4.

Notice that d tends to increase as n grows since d is randomly sampled from $[0, \sum_{i=1}^{n} u_i]$. Figure 4 shows that the worst-case R_E tends to decrease as n grows, which validates Remark 5.

We also study the effect of possible degeneracy of the cost function (from quadratic to linear) and the involvement of no set-up cost on R_E. We set the first half of the generators with linear costs, i.e., $a_i = 0$, for $i = 1, 2, \cdots, n/2$, and half of the generators without set-up cost, i.e., $c_i = 0$, for $i = n/4 + 1, n/4 + 2, \cdots, 3n/4$. Noting the setting of n, $n/2$ and $n/4$ are both integers for n =100, 200, 300, 400, 500 and 1000. Then we have four groups of generators. The first quarter of the generators have $a_i = 0$ and $c_i > 0$, the second quarter of the generators have $a_i = 0$ and $c_i = 0$, the third quarter of the generators have $a_i > 0$ and $c_i = 0$ and the last quarter of the generators have $a_i > 0$ and $c_i > 0$. The setting of other parameters remains as before. Table 2 shows the numerical results.

We can observe that, facing degeneracy, the proposed algorithm is still effective to yield a good-quality approximate solution in a short period of time. This indicates the robustness of

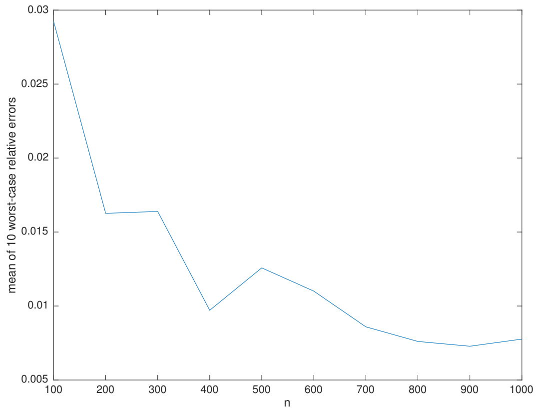

Figure 4: The effect of n on relative errors

value \ distribution / n	optimal	[0, 0.001)	[0.001, 0.002)	[0.002, 0.003)	[0.003, 0.004)	[0.004, 0.005)	[0.005, 0.006)	[0.006, 0.007)	[0.007, 0.008)	[0.008, 0.009)	[0.009, 0.01)	≥ 0.01	max	mean	var	AA(s)	BBA(s)	Cplex(s)
100	602	972	24	0	1	0	0	0	3	0	0	0	7.91×10^{-3}	1.14×10^{-4}	2.46×10^{-7}	2.97×10^{-4}	3.28×10^{-3}	5.62×10^{-3}
200	608	991	4	1	3	1	0	0	0	0	0	0	4.27×10^{-3}	6.15×10^{-5}	8.96×10^{-8}	4.23×10^{-4}	5.15×10^{-3}	1.09×10^{-2}
300	628	996	2	1	1	0	0	0	0	0	0	0	3.15×10^{-3}	3.17×10^{-5}	2.36×10^{-8}	5.74×10^{-4}	7.85×10^{-3}	2.61×10^{-2}
400	652	995	4	1	0	0	0	0	0	0	0	0	2.99×10^{-3}	2.78×10^{-5}	2.26×10^{-8}	7.24×10^{-4}	1.04×10^{-2}	4.84×10^{-2}
500	646	998	2	0	0	0	0	0	0	0	0	0	1.63×10^{-3}	1.74×10^{-5}	7.91×10^{-9}	8.47×10^{-4}	1.29×10^{-2}	1.99×10^{-1}
1000	724	1000	0	0	0	0	0	0	0	0	0	0	7.65×10^{-4}	4.95×10^{-6}	8.66×10^{-10}	1.53×10^{-3}	2.36×10^{-2}	2.71×10^{0}

Table 2: Relative error R_E of degenerate instances

the proposed algorithm. The results shown in Table 2 are similar to those in Table 1 except that more optimal solutions are obtained by the proposed AA and the running time of the Cplex solver becomes shorter. The reason is that in this experiment more linear objective functions (more $a_i = 0$) and less integer variables (more $c_i = 0$) are involved. Hence the problems become easier to solve, in particular for the Cplex solver.

5 Conclusions

In this paper, we have proposed a Lagrangian dual based $\mathcal{O}(n \log n)$ approximation algorithm for solving large-scale single-period unit commitment problems. Moreover, we have found tight theoretical bounds for the absolute errors and relative errors of the approximate solutions generated by the proposed algorithm. To the best of our knowledge, we are the first to propose a polynomial time solvable approximation algorithm with tight error bounds for this NP-hard problem. Computational experiments have shown that the proposed algorithm is very effective to yield a good-quality approximate solution in a very short period of time. Hence the proposed algorithm is promising for solving large-scale single-period UC problems online.

Although problem (1.1) studied in this paper is of the simplest form of unit commitment problems, it plays a fundamental role for us to handle UC problems of more complex structure. Possible extensions under consideration are (i) single-period unit commitment problems with other types of cost functions and side constraints; (ii) multi-period unit commitment problems; and (iii) single-period unit commitment problems with uncertain/stochastic demands and/or capacities. For (i), finding a tight convex relaxation with explicit expression for detailed analysis could be the key of designing a theoretically sound approximation algorithm. For (ii), the proposed AA may serve as a good estimate, under certain conditions, to decompose a multi-period allocation policy (of generators) to a sequence of single-period allocation decisions. For (iii), when the demand of capacities becomes uncertain, concepts of robust optimization [2] may be incorporated to refine the proposed AA.

Acknowledgments

Supports of this work come from the National Natural Science Foundation of China Grant #11571029, #11771243, and US Army Research Office Grant #W911NF-15-1-0223.

Reference

[1] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, Nonlinear Programming : Theory and Algorithms, 3rd edition *John Wiley & Sons*, New Jersey, 2013.

[2] A. Ben-Tal, L. El Ghaoui and A. S. Nemirovski, Robust Optimization, Princeton Series in Applied Mathematics, *Princeton University Press*, 2009.

[3] C. C. Carøe and R. Schultz, A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system, *Technical Report SC 98-11*, Konrad-Zuse-Zentrum für Informationstechnik.

[4] G. W. Chang, Y. D. Tsai, C. Y. Lai and J. S. Chung, A practical mixed integer linear programming based approach for unit commitment, *IEEE Power Engineering Society General Meeting*, 2004, 221–225.

[5] C. L. Chen and S. C. Wang, Branch-and-bound scheduling for thermal generating units, *IEEE Transactions on Energy Conversion*, **8** (1993), 184–189.

[6] M. Clerc and J. Kennedy, The particle swarm — explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, **6** (2002), 58–73.

[7] H. Dai, N. Zhang and W. Su, A literature review of stochastic programming and unit commitment, *Journal of Power and Energy Engineering*, **3** (2015), 206–214.

[8] C. Dang and M. Li, A floating-point genetic algorithm for solving the unit commitment problem, *European Journal of Operational Research*, **181** (2007), 1370–1395.

[9] G. Dudek, Adaptive simulated annealing schedule to the unit commitment problem, *Electric Power Systems Research*, **80** (2010), 465–472.

[10] S. -C. Fang and W. Xing, Linear Conic Optimization, *Science Press*, Beijing, 2013.

[11] A. Frangioni, C. Gentile, E. Grande and A. Pacifici, Projected perspective reformulations with applications in design problems, *Operations Research*, **59** (2011), 1225–1232.

[12] A. Frangioni, F. Furini and C. Gentile, Approximated perspective relaxations: a project and lift approach, *Computational Optimization and Applications*, **63** (2016), 705–735.

[13] F. D. Galiana, A. L. Motto and F. Bouffard, Reconciling social welfare, agent profits, and consumer payments in electricity pools, *IEEE Transactions on Power Systems*, **18** (2003), 452–459.

[14] B. Ghaddar, J. Naoum-Sawaya, A. Kishimoto, N. Taheri and B. Eck, A Lagrangian decomposition approach for the pump scheduling problem in water networks, *European Journal of Operational Research*, **241** (2015), 490–501.

[15] N. Gröwe-Kuska and W. Römisch, Stochastic unit commitment in hydrothermal power production planning, *Application of Stochastic Programming*, edited by S. W. Wallace and W. T. Ziemba, MOS-SIAM Series on Optimization, 2005, Chapter 30, 633–653.

[16] K.-Y. Huang, H.-T. Yang and C.-L. Huang, A new thermal unit commitment approach using constraint logic programming, *Proceedings of the 20th International Conference on Power Industry Computer Applications*, 1997, 176–185.

[17] R. Jiang, M. Zhang, G. Li and Y. Guan, Two-stage network constrained robust unit commitment problem, *European Journal of Operational Research*, **234** (2014), 751–762.

[18] P. G. Lowery, Generating unit commitment by dynamic programming, *IEEE Transactions on Power Apparatus and Systems*, **PAS-85** (1966), 422–426.

[19] C. Lu, Z. Deng, S. -C. Fang, Q. Jin and W. Xing, Fast computation of global solutions to the single-period unit commitment problem for electricity market applications, working paper, 2019.

[20] H. Ma and S. M. Shahidehpour, Transmission-constrained unit commitment based on Benders decomposition, *International Journal of Electrical Power and Energy Systems*, **20** (1998), 287–294.

[21] A. H. Mantawy, Y. L. Abdel-Magid and S. Z. Selim, Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem, *IEEE Transactions on Power Systems*, **14** (1999), 829–836.

[22] A. L. Motto and F. D. Galiana Unit commitment with dual variable constraints, *IEEE Transactions on Power Systems*, **19** (2004), 330–338.

[23] R. Nieva, A. Inda and J. Frausto, CHT: a digital computer package for solving short term hydro-thermal coordination and unit commitment problems, *IEEE Transactions on Power Systems*, **1** (1986), 168–174.

[24] E. Ogbe and X. Li, A new cross decomposition method for stochastic mixed-integer linear programming, *European Journal of Operational Research*, **256** (2017), 487–499.

[25] M. W. Padberg, T. J. VanRoy and L. A. Wolsey, Valid linear inequalities for fixed charge problems, *Operations Research*, **33** (1985), 842–861.

[26] N. P. Padhy, Hybrid Models for Unit Commitment Problems, Ph.D. Dissertation, Anna University, Chennai, India, 1997.

[27] N. P. Padhy, Unit commitmentâĂŤa bibliographical survey, *IEEE Transactions on Power Systems*, **19** (2004), 1196–1205.

[28] A. Papavasiliou and S. S. Oren, Multiarea stochastic unit commitment for high wind penetration in a tramsimission constrained network, *Operations Research*, **61** (2013), 578–592.

[29] M. P. Nowak and W. Römisch, Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty, *Annals of Operations Research*, **100** (2000), 251–272.

[30] R. T. Rockafellar and R. J. -B. Wets, Variational Analysis, A Series of Comprehensive Studies in Mathematics Vol. 317, Berlin: Springer, 2004.

[31] A. Rong, H. Hakonen and R. Lahdelma, A variant of the dynamic programming algorithm for unit commitment of combined heat and power systems, *European Journal of Operational Research*, **190** (2008), 741–755.

[32] T. Sawa, Y. Sato, M. Tsurugai and T. Onishi, Security constrained integrated unit commitment using quadratic programming, *IEEE Lausanne Power Tech*, 2007, 1858–1863.

[33] V. Selvi and Dr. R. Umarani, Comparative analysis of ant colony and particle swarm optimization techniques, *International Journal of Computer Applications*, **5** (2010), 1–6.

[34] M. S. Nagaraja, Optimum generation scheduling for thermal power plants using artificial neural network, *International Journal of Electrical and Computer Engineering*, **1** (2011), 135–139.

[35] T. Schulze, A. Grothey and K. McKinnon, A stabilised scenario decomposition algorithm applied to stochastic unit commitment problems, *European Journal of Operational Research*, **261** (2017), 247–259.

[36] M. Tahanan, W. van Ackooij, A. Frangioni and F. Lacalandra, Large-scale unit commitment under uncertainty: a literature survey, *4OR Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, **13** (2015), 115–171.

[37] C. -L. Tseng, On Power System Generation Unit Commitment Problems, Ph.D. Dissertation, University of California, Berkeley, 1996.

[38] P. P. J. van den Bosch and G. Honderd, A solution of the unit commitment problem via decomposition and dynamic programming, *IEEE Transactions on Power Apparatus and Systems*, **PAS-104** (1985), 1684–1690.

[39] L. Wu and M. Shahidehpour, Security-constrained unit commitment with uncertainties, in *Power Grid Operation in a Market Environment: Economic Efficiency and Risk Mitigation*, edited by Hong Chen, Chapter 5, 115–168, John Wiley & Sons, New Jersey, 2016.

[40] H. Y. Yamin, Review on methods of generation scheduling in electric power systems, *Electric Power Systems Research*, **69** (2004), 227–248.