SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2024, Vol. 67, Iss. 12, 222205:1–222205:13
https://doi.org/10.1007/s11432-024-4210-2

A distributed decomposition algorithm for solving large-scale mixed integer programming problem

Fangzheng TIAN[1], Hongzhe LIU[2*] & Wenwu YU[2]

[1]*School of Cyber Science and Engineering, Southeast University, Nanjing* 210000*, China;*
[2]*School of Mathematics, Southeast University, Nanjing* 210000*, China*

Received 26 May 2024/Revised 22 September 2024/Accepted 31 October 2024/Published online 11 December 2024

Abstract Mixed integer programming is inherently involved in solving a significant number of practical problems. This paper focuses on mixed integer programming, where the objective function is the summation of N functions, and the constraints include both scalar coupling and set constraints. Given the potentially large scale of these problems, the goal of this work is to propose a distributed method to solve large-scale problems more efficiently. The right-hand side allocation decomposition approach is employed to address the large-scale mixed integer programming problem. Algorithms are then proposed for solving these problems, based on the analysis of the continuity, differentiability, and local convexity properties of the decomposed subproblems. Simulation experiments with randomly generated coefficients demonstrate the superior performance of the proposed algorithms compared to the Gurobi solver, offering higher solution accuracy and faster processing time for large-scale mixed integer programming problems with nonlinear objective and constraint functions.

Keywords mixed integer programming, decomposition methods, distributed optimization

1 Introduction

Mixed integer programming (MIP) problems constitute a class of optimization problems that involve integer variables. Addressing different forms of MIP is essential for solving various practical tasks, including resource allocation and path planning. Consequently, the study of MIP is of substantial importance. On the other hand, MIP problems are known to be NP-Hard [1], which renders the exact and efficient solution of large-scale MIP problems highly challenging.

Although MIP problems belong to the class of NP-Hard problems, exact algorithms exist for solving them. The branch and bound algorithm proposed in [2] is the basis for most of the algorithms and commercial solvers. Besides, cut methods and the combination of the branch and bound improve the efficiency of the solving process greatly [3]. The methods described above are the basic techniques for solving general integer programming problems. For convex MIP problems, which are convex programs when the integrality constraints are relaxed, some specialized algorithms are available. Ref. [4] studied the rule and efficiency of the branch and bound method in convex nonlinear integer programming. With integer variables fixed, if the convex programming problem is easy to solve, Ref. [5] proposed generalized benders decomposition to formulate an efficient algorithm. Later, Ref. [6] extended this method to non-convex MIP problems. Using the method of tangential approximation to delineate the feasible region at specific points, Ref. [7] proposed an iterative extended cutting plane algorithm that can handle constraint functions with high nonlinearity. Similarly, Ref. [8] employed the outer approximation technique to construct a mixed-integer linear programming problem equivalent to the original non-linear problem. Moreover, Ref. [9] examined a class of problems featuring separable nonlinear functions and introduced enhancements to the outer approximation technique. Specifically, the separable nonlinear function is expressed as the sum of compositions involving a univariate convex function and an affine function. Additionally, Ref. [10] achieved tighter linear approximations for a class of functions known as almost additively separable.

* Corresponding author (email: 101300130@seu.edu.cn)

© Science China Press 2024

The aforementioned method typically simplifies problem formulations using approximation and linearization techniques. Nevertheless, it tends to overlook large-scale MIP problems characterized by separable structural properties. Correspondingly, decomposition methods are appropriate for these problems. As a classical decomposition method for large-scale optimization problems, right-hand side allocation decomposes these problems into a main problem and subproblems that can be solved using parallel or distributed methods [11]. Thus this approach can improve the efficiency of solving large-scale problems and has recently attracted significant research attention. Initially, the right-hand side allocation method was primarily applied to decoupling convex programming problems [12]. Further, due to its effectiveness in solving large-scale optimization problems, it has been extended to the case of solving large-scale MIP. Specifically, applications of the right-hand side allocation method to mixed integer linear programming have been successfully demonstrated in [13–16]. On the other hand, it is worth noting that applying the right-hand side allocation decomposition method to mixed integer nonlinear programming still presents significant challenges.

In addition to the decomposition method mentioned above, Lagrangian dual decomposition is a conventional decoupling technique used to address large-scale problems. Based on this approach, Ref. [17] proposed a method to obtain an approximate solution. Furthermore, Ref. [18] designed an algorithm to compute a solution feasible for the original problem by performing distributed calculations for local MIP and centralized calculations for the multiplier. In contrast to decomposition methods, Ref. [19] solved the problem using a distributed cutting plane and constraint exchange algorithm within a peer-to-peer network. Additionally, heuristic principles and mathematical programming methodologies serve as fundamental strategies for addressing the challenges posed by integer programming [20–22]. However, all of these decomposition methods focus on mixed-integer linear programming and are not readily applicable to nonlinear problems.

This paper extends the right-hand side allocation decomposition method to mixed integer nonlinear programming problems, addressing a gap in existing decomposition methods that are generally limited to convex programming problems or linear integer programming problems. Our approach demonstrates that the decomposition results in subproblems with complex properties, requiring further analysis to develop effective algorithms. We begin by analyzing the continuity of the subproblems post-decomposition, establishing that each subproblem is continuous. Beyond continuity, we examine their differentiability, showing that the positive derivative corresponds to a specific optimal multiplier, with analogous findings for the negative derivative. Furthermore, we analyze the global structure of the subproblems and present a detailed formulation for the case where continuous and integer variables are separated. The resulting formulation represents the subproblems as interconnected convex functions, making each subproblem locally convex. Based on these insights, we propose two algorithms that employ the properties of the subproblems and analyze their convergence characteristics. Through extensive simulation experiments, we demonstrate the superior performance of our algorithms compared to the Gurobi solver, highlighting their practical effectiveness and contribution to the field.

2 Preliminaries

2.1 Problem formulation

The MIP problem studied in this paper is formulated as

$$\begin{aligned} \min_{\{(x_i,y_i)\}_{i=1}^N} \quad & \sum_{i=1}^N f_i(x_i,y_i) \\ \text{s.t.} \quad & \sum_{i=1}^N g_i(x_i,y_i) \leqslant 0, \\ & (x_i,y_i) \in X_i \times Y_i,\ i=1,\ldots,N, \end{aligned} \tag{1}$$

where each $X_i \subset \mathbb{R}^{p_i}$ is a compact convex set, $Y_i \subset \mathbb{Z}^{q_i}$ is a finite integer set, the objective function $f_i(\cdot)$ represents the function from $\mathbb{R}^{p_i+q_i}$ to $\mathbb{R}$ and the constraint function $g_i(\cdot)$ represents the function from $\mathbb{R}^{p_i+q_i}$ to $\mathbb{R}$. Without generality, let each Y_i have the same cardinality J. Problem (1) has been well studied recently since it is generally involved in resource allocation problems. For example, in the economic dispatch problem, the continuous variable x_i typically represents the standby power of a generator, while the discrete variable y_i represents the generator's switching status. Consequently, the variables x_i and y_i belong to distinct sets, denoted as $(x_i, y_i) \in X_i \times Y_i$. Furthermore, the coupling function $g_i(x_i, y_i)$ represents the total load of each generator, which can be expressed as the sum of functions of x_i and y_i. Additionally, the coupling constraint $\sum_{i=1}^{N} g_i(x_i, y_i) \leqslant 0$ reflects the total load limitation. Specifically, this problem has been extensively studied in [14,17,18]. As is well known, the MIP is an NP-hard problem, which means that the computational complexity of solving the MIP problems will grow exponentially as the number of variables increases. Therefore, it remains challenging to solve large-scale MIP problems exactly and efficiently. This paper aims to propose an approach that employs decomposition methods to solve large-scale MIPs more efficiently. Next, we make some assumptions first.

Assumption 1 (Optimality)**.** There exists an optimal solution $(x_1^*, y_1^*, \ldots, x_N^*, y_N^*)$, with the optimal value $f^* = \sum_{i=1}^{N} f_i(x_i^*, y_i^*)$.

Assumption 2 (Convexity)**.** For a given y_i, the functions $f_i(x_i, y_i)$ and $g_i(x_i, y_i)$ are convex with respect to x_i.

The first assumption outlined herein serves to uphold the optimality of the problem, while the second assumption ensures that, given y_i in advance, the optimization problem is convex, thereby facilitating efficient resolution. Specifically, both assumptions are standard and commonly found in [6–9].

2.2 Decomposition of the MIP problem

In this subsection, we introduce the right-hand side allocation method to decompose the MIP problem. First, by introducing an auxiliary variable z_i, the problem can be equivalently rewritten as

$$\begin{aligned} \min_{\{(x_i,y_i,v_i,z_i)\}_{i=1}^N} \quad & \sum_{i=1}^{N} (f_i(x_i, y_i) + M v_i) \\ \text{s.t.} \quad & g_i(x_i, y_i) - v_i \leqslant z_i, \\ & \sum_{i=1}^{N} z_i = 0, \\ & (x_i, y_i) \in X_i \times Y_i, v_i \geqslant 0, \ i = 1, \ldots, N, \end{aligned} \tag{2}$$

where v_i is a relaxing variable which makes the inequality constraint $g_i(x_i, y_i) \leqslant z_i$ has a feasible solution in $X_i \times Y_i$, and M is sufficiently large. Then Eq. (2) can be further rewritten as

$$\begin{aligned} \min_{\{z_i\}_{i=1}^N} \quad & \sum_{i=1}^{N} p_i(z_i) \\ \text{s.t.} \quad & \sum_{i=1}^{N} z_i = 0, \end{aligned} \tag{3}$$

where $p_i(z_i) : \mathbb{R} \to \mathbb{R}$ is defined as the optimal value of the subproblem

$$\begin{aligned} \min_{(x_i,y_i,v_i)} \quad & f_i(x_i, y_i) + M v_i \\ \text{s.t.} \quad & g_i(x_i, y_i) - v_i \leqslant z_i, \\ & (x_i, y_i) \in X_i \times Y_i, \ v_i \geqslant 0, \end{aligned} \tag{4}$$

which is lower bounded since each set $X_i \times Y_i$ is compact. Then, we have $p_i(z_i) = \min_{j \in \{1,\ldots,J\}} \hat{p}_i(z_i, y_{ij})$, and $\hat{p}_i(z_i, y_{ij})$ is defined as

$$\begin{aligned} \min_{x_i} \quad & f_i(x_i, y_{ij}) + M v_i \\ \text{s.t.} \quad & g_i(x_i, y_{ij}) - v_i \leqslant z_i, \\ & x_i \in X_i, \ v_i \geqslant 0, \end{aligned} \tag{5}$$

where $y_{ij} \in Y_i = \{y_{i1}, \ldots, y_{iJ}\}$.

Note that problem (2) is a relaxed version of the original problem (1); however, the equivalence between the two problems can be proven. With y_i fixed to y_{ij}, and according to Proposition III.3 in [12], when M is sufficiently large, the optimal solutions to the relaxed problem

$$\begin{aligned} \min_{\{(x_i,v_i)\}_{i=1}^N} \ & \sum_{i=1}^N (f_i(x_i,y_{ij}) + Mv_i) \\ \text{s.t.} \ & \sum_{i=1}^N g_i(x_i,y_{ij}) \leqslant Nv_i, \\ & x_i \in X_i,\ v_i \geqslant 0,\ i=1,\ldots,N \end{aligned}$$

is an optimal solution to the problem

$$\begin{aligned} \min_{\{x_i\}_{i=1}^N} \ & \sum_{i=1}^N f_i(x_i,y_{ij}) \\ \text{s.t.} \ & \sum_{i=1}^N g_i(x_i,y_{ij}) \leqslant 0, \\ & x_i \in X_i,\ i=1,\ldots,N. \end{aligned}$$

Therefore, this proves the equivalence between the relaxed problem (2) and the original problem (1). Furthermore, since solving the main problem (3) and the subproblems (4) is equivalent to solving the relaxed problem (2), it is also equivalent to solving the original problem (1).

Remark 1. The right-hand side allocation method has been extensively studied since it can decouple large-scale coupled optimization problems into a main problem and subproblems, with smaller scales and thereby enhance the efficiency of the solution process. However, this method is primarily applied to constraint-coupled convex optimization problems and is seldom used for MIP. On the other hand, as discussed in [11], the essential point of employing this approach is that the decoupled main problem and subproblems can be efficiently solved. Therefore, in this paper, to apply the right-hand side allocation method to the large-scale MIP problem (1), the assumption that the continuous and integer variables can be separated is introduced, ensuring that the objective function $p_i(z_i)$ in the decoupled problem (3) exhibits locally convex properties. Furthermore, the locally convex nature of $p_i(z_i)$ enables problem (3) to be formulated as a locally convex optimization problem. Additionally, the calculation process for the subgradient of $p_i(z_i)$ is provided in detail, enabling the use of subgradient-based methods to solve the problem (3). Moreover, due to the relatively small scale, the decoupled subproblem (4) being the MIP, is also computationally tractable. Since both the decoupled problems (3) and (4) can be efficiently solved based on the provided analyses, the right-hand side allocation method can be successfully applied in addressing the mixed integer nonlinear programming problem (1) in this paper.

2.3 Semicontinuity of the function $p_i(z_i)$

It appears that the set $\Omega_i(z_i) = \{(x_i,y_i,v_i)|g_i(x_i,y_i) - v_i \leqslant z_i, (x_i,y_i) \in X_i \times Y_i, v_i \geqslant 0\}$ constitutes a sub-level set of g_i within a confined domain. This set could alternatively be interpreted as a point-to-set mapping: $\mathbb{R} \to \mathbb{R}^{p_i} \times \mathbb{Z}^{q_i} \times \mathbb{R}^+$. In order to finely characterize the point-to-set mapping, we introduce the semicontinuity from [23].

For a point-to-set mapping Ω from domain D to range G, we define the semicontinuity as follows.

Definition 1. Ω is upper semicontinuous at a point $s \in G$. If $s_i \to s$, $\{s_i\} \subset D$, and $t_i \to t$ with $t_i \in \Omega(s_i)$ for each i, then it implies that $t \in \Omega(s)$.

Definition 2. Ω is lower semicontinuous at a point $s \in G$. If $t \in \Omega(s)$, $s_i \to s$, $\{s_i\} \subset D$, then there exist an integer m and a sequence $\{t_m, t_{m+1}, \ldots\}$ with the properties that (a) $t_i \in \Omega(s_i)$ for $i \geqslant m$ and (b) $t_i \to t$.

Ω is continuous at a point $s \in G$ if and only if it is both upper and lower semicontinuous at s.

Next, focusing on the point-set mapping $\Omega_i(z)$, we can derive the following proposition.

Proposition 1. The point-to-set mapping $\Omega_i(z_i)$ is continuous at every point z_i, and $p_i(z_i)$ is continuous at the corresponding point z_i.

Proof. For the given z_i^*, let $B(z_i^*, \varepsilon)$ denote the neighborhood of z_i^* for a positive ε. According to the definition of the problem (4), the optimal solution is bounded $\forall z_i \in B(z_i^*, \varepsilon)$. Therefore, we could only consider v_i in a compact set V_i for convenience. In the next discussion, let $\Omega_i(z_i) = \{(x_i, y_i, v_i) | g_i(x_i, y_i) - v_i \leqslant z_i, (x_i, y_i, v_i) \in X_i \times Y_i \times V_i, z_i \in B(z_i^*, \varepsilon)\}$.

Referring to Theorem 1.4 in [23], If $\Omega_i(z_i)$ is continuous at z_i and $X_i \times Y_i \times V_i$ is sequentially compact, then $p_i(z_i)$ exhibits continuity at z_i. Given the compactness of $X_i \times Y_i \times V_i$, the remaining task is to ascertain whether $\Omega_i(z_i)$ is continuous. We prove the upper continuity first.

Suppose there exists a sequence $\{z_i^n\}$, with $z_i^n \in B(z_i^*, \varepsilon)$, and z_i^n converging to z_i^*. Correspondingly, there is a sequence $\{(x_i^n, y_i^n, v_i^n)\}$ satisfying $(x_i^n, y_i^n, v_i^n) \in \Omega_i(z_i^n)$ and (x_i^n, y_i^n, v_i^n) converging to (x_i, y_i, v_i). Given the continuity of $g_i(x_i, y_i) - v_i$ and the compactness of set $X_i \times Y_i \times V_i$, $(x_i, y_i, v_i) \in \Omega(z_i^*)$ holds evidently. So, $\Omega(z_i)$ is upper semicontinuous at z_i^*, and $p_i(z_i)$ is lower semicontinuous at z_i^*.

Next, we establish the lower continuity of $\Omega(z_i)$. Let $(x_i, y_i, v_i) \in \Omega(z_i)$, $\{z_i^n\} \subset B(z_i^*, \varepsilon)$, and $z_i^n \to z_i^*$. By the continuity of $g_i(x_i, y_i) - v_i$, if z_i^n is sufficiently close to z_i^*, there exists a sequence $\{(x_i^n, y_i, v_i^n)\}$ such that $(x_i^n, y_i, v_i^n) \in \Omega_i(z_i^n)$ and $(x_i^n, y_i^n, v_i^n) \to (x_i, y_i, v_i)$, respectively. Hence, we have proved the lower continuity of $\Omega(z_i)$, thus completing the proof.

2.4 Differentiability of function $p_i(z_i)$

In addition to continuity, we can explore certain differential properties of the function $p_i(z_i)$. Initially, let $\Omega_i^I(z_i) = \{y_i | g_i(x_i, y_i) - v_i \leqslant z_i, (x_i, y_i) \in X_i \times Y_i, v_i \geqslant 0\}$ represent the integer part of $\Omega_i(z_i)$. Following this, we will introduce a proposition that clarifies how $\Omega_i^I(z_i)$ evolves as z_i changes.

Proposition 2. For each i, there exists $\varepsilon^* > 0$ such that $\forall \varepsilon \in [0, \varepsilon^*]$, $\Omega_i^I(z_i + \varepsilon)$ remains unchanged.

Proof. Suppose the proposition is false. Then, for any $\varepsilon > 0$, there exists an $(x_i, y_i, v_i) \in \Omega_i(z_i + \varepsilon)$ such that $y_i \notin \Omega_i^I(z_i)$. Consequently, we can construct a sequence ε_n satisfying this property with $\varepsilon_n \to 0$.

Since the sequence $\{(x_i^n, y_i^n, v_i^{n_k})\}$ is bounded, it possesses a convergent subsequence $\{(x_i^{n_k}, y_i^{n_k}, v_i^{n_k})\}$ that converges to $(\bar{x}_i, \bar{y}_i, \bar{v}_i)$. Given that $y_i^{n_k}$ takes integer values, it follows that $\bar{y}_i \notin \Omega_i^I(z_i)$.

By taking the limit of both sides of $g_i(x_i^{n_k}, y_i^{nk}) - v_i^{n_k} \leqslant z_i + \varepsilon_{n_k}$, we conclude that $g_i(\bar{x}_i, \bar{y}_i) - \bar{v}_i \leqslant z_i$, leading to a contradiction. This completes the proof of the proposition.

Drawing from Proposition 2, we can derive the following theorem concerning the right derivative of $p_i(z_i)$.

Theorem 1. For each i, assuming that Assumptions 1 and 2 hold, then the progressive derivative of $p_i(z_i)$ at z_i exists. It is equivalent to the negative value of Lagrange multiplier λ_i^* for the problem (5) with y_i fixed to a specific integer value y_{ik}.

Proof. For a sufficiently small ε, consider the interval $[\bar{z}_i, \bar{z}_i + \varepsilon]$ where $p_i(z_i)$ is defined. According to Proposition 2, the part of the feasible set that consists of integer values remains unchanged. Without loss of generality, let the integer variable y_i belong to the enumerated set $\{y_{i1}, \ldots, y_{iJ}\}$. Hence, we have $p_i(z_i) = \min_{j \in \{1,\ldots,J\}} \hat{p}_i(z_i, y_{ij})$, where $\hat{p}_i(z_i, y_{ij})$ is defined previously. It is evident that $\hat{p}_i(z_i, y_{ij})$ is continuous on the interval $[\bar{z}_i, \bar{z}_i + \varepsilon]$.

Next, let $I \subseteq \{1, \ldots, J\}$ denote an index set such that

$$I = \mathop{\mathrm{argmin}}_{j \in \{1,\ldots,J\}} \{\hat{p}_i(\bar{z}_i, y_{ij})\}.$$

Following this, according to the boundedness theorem of continuous functions, with a small increase in ε, we only need to consider the functions $\hat{p}_i(z_i, y_{ij})$ where $j \in I$. These functions dominate the value of $p_i(z_i)$ over the interval $[\bar{z}_i, \bar{z}_i + \varepsilon]$.

Considering the convexity of $\hat{p}_i(z_i, y_{ij})$ over the interval $[\bar{z}_i, \bar{z}_i + \varepsilon]$, it exhibits a right derivative at $\bar{z}_i$, denoted as $-\lambda_j(\bar{z}_i)$, where $\lambda_j(\bar{z}_i)$ is the Lagrange multiplier for the problem $\hat{p}_i(z_i, y_{ij})$, as detailed in [11, Subsection 4.2]. It is evident that $\hat{p}_i(\bar{z}_i + \varepsilon, y_{ij}) = \hat{p}_i(\bar{z}_i, y_{ij}) - \lambda_j(\bar{z}_i)\varepsilon + o(\varepsilon)$, where $o(\varepsilon)$ represents higher-order infinitesimals of ε. Therefore, if $j, k \in I$ and $\lambda_j(\bar{z}_i) > \lambda_k(\bar{z}_i)$, then $\hat{p}_i(\bar{z}_i + \varepsilon, y_{ij}) < \hat{p}_i(\bar{z}_i + \varepsilon, y_{ik})$ for sufficiently small ε, respectively.

Therefore, to compute $p_i(z_i)$ over $[\bar{z}_i, \bar{z}_i + \varepsilon]$, we need only consider the index set $\bar{I} = \mathrm{argmax}_{j \in I} \lambda_j(\bar{z}_i)$. For simplicity, define $\lambda^*(\bar{z}_i) = \max_{j \in I}\{\lambda_j(\bar{z}_i)\}$. The following equation is then obtained:

$$\lim_{z_i \to \bar{z}_i^+} \frac{p_i(z_i) - p_i(\bar{z}_i)}{z_i - \bar{z}_i} = \lim_{z_i \to \bar{z}_i^+} \frac{\min_{j \in \bar{I}}\{\hat{p}_i(z_i, y_{ij})\} - p_i(\bar{z}_i)}{z_i - \bar{z}_i} = -\lambda^*(\bar{z}_i),$$

which completes the proof.

As demonstrated in the proof of Theorem 1, we also get the method for computing the right derivative.

• Get the optimal value set for y_i, which means the problem (5) has the smallest value at z_i. Let I denote the index set.

• From the set I, calculate the right derivative $-\lambda_j(z_i)$ of $\hat{p}_i(z_i, y_{ij})$. Then find the smallest $-\lambda_j(z_i)$.

Given that $p_i(z_i)$ may not be upper semicontinuous, the left derivative might not exist. However, in cases where $p_i(z_i)$ exhibits upper semicontinuity, we can similarly compute the left derivative. The calculation method for the left derivative parallels that of the right derivative, except that we replace the smallest $\lambda_j(z_i)$ with the largest $\lambda_j(z_i)$. Further, as outlined in [11, Subsection 4.2], the right or left derivative of $\hat{p}_i(z_i, y_{ij})$ corresponds to the optimal Lagrange multiplier minus of the problem (5).

2.5 Calculation of function $p_i(z_i)$

The right derivative and semicontinuity of $p_i(z_i)$ are localized properties. If the variables x_i and y_i in functions $f_i(x_i, y_i)$ and $g_i(x_i, y_i)$ are separable, we can gain a more comprehensive understanding of $p_i(z_i)$ from a global perspective. Therefore, we introduce the following assumption regarding $f_i(x_i, y_i)$ and $g_i(x_i, y_i)$.

Assumption 3 (Separation). The functions $f_i(x_i, y_i)$ and $g_i(x_i, y_i)$ have the following form:

$$\begin{aligned} f_i(x_i, y_i) &= f_{ic}(x_i) + f_{ib}(y_i), \\ g_i(x_i, y_i) &= g_{ic}(x_i) + g_{ib}(y_i). \end{aligned}$$

This assumption serves as a generalization of the linear function. Furthermore, according to Assumption 3, $\hat{p}_i(z_i, y_{ij})$ takes on a special form as defined below:

$$\begin{aligned} \min_{(x_i, v_i)} \quad & f_{ic}(x_i) + f_{ib}(y_{ij}) + Mv_i \\ \text{s.t.} \quad & g_{ic}(x_i) + g_{ib}(y_{ij}) - v_i \leqslant z_i, \\ & x_i \in X_i,\ v_i \geqslant 0, \end{aligned} \tag{6}$$

where $y_{ij} \in Y_i = \{y_{i1}, \ldots, y_{iJ}\}$.

Without loss of generality, consider the case where only two points, y_{i1} and y_{i2}, exist in Y_i. Consequently, we can derive $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$. It is evident that if the functions $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$ are disjoint, then $p_i(z_i)$ is readily obtained. Hence, the remaining case involves the intersection of two curves. In the case where the two curves partially overlap, their overlapping sections are considered to intersect once. Next, we express $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$ as

$$\begin{aligned} \min_{(x_i, v_i)} \quad & f_{ic}(x_i) + Mv_i + \gamma_j \\ \text{s.t.} \quad & g_{ic}(x_i) - v_i + \delta_j \leqslant z_i, \\ & x_i \in X_i,\ v_i \geqslant 0, \end{aligned} \tag{7}$$

where $\gamma_j = f_{ib}(y_{ij})$, $\delta_j = g_{ib}(y_{ij})$, $j = 1, 2$. If $\gamma_1 = \gamma_2$, $p_i(z_i)$ is equivalent to the $\hat{p}_i(z_i, y_{ij})$ with the larger δ_j. Then, in the subsequent discussion, we presume that $\gamma_1 > \gamma_2$.

To derive the expression for $p_i(z_i)$, we initially introduce the characteristic of a general convex function.

Lemma 1. Let $h(t)$ be a convex function defined on $\mathbb{R}$. Consider $t_1, t_2 \in \mathbb{R}$ and $t_1 < t_2$. Let $l(t)$ represent the line passing through the points $(t_1, h(t_1))$ and $(t_2, h(t_2))$. Then $\forall t \in [t_1, t_2]$, $h(t) \leqslant l(t)$; and $\forall t \in [t_2, \infty]$, it holds that $h(t) \geqslant l(t)$. Furthermore, if $l(t)$ does not act as the separating hyperplane, $\forall t \in (t_2, \infty]$, we have $h(t) > l(t)$.

Proof. We only need to establish the second part of the lemma. Hence, assume without loss of generality that $l(t)$ is not the supporting hyperplane of $h(t)$. Consequently, $\exists t^* \in [t_1, t_2]$ such that $l(t^*) > h(t^*)$. Moreover, there exists a separating hyperplane $\hat{l}(t)$ that passes through the point $(t_2, h(t_2))$ and satisfies $\hat{l}(t) \leqslant h(t)$. The case is shown in Figure 1.

Therefore, we deduce that $l(t^*) > \hat{l}(t^*)$. Furthermore, since $l(t_2) = \hat{l}(t_2)$, we conclude that for all $t \in (t_2, \infty]$, it holds that $l(t) < \hat{l}(t)$, thereby proving $l(t) < h(t)$.

Using Lemma 1, we can present an intuitive description of $p_i(z_i)$ as follows.

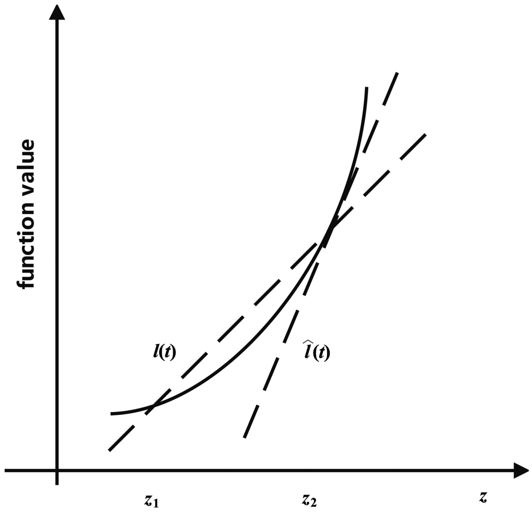

Figure 1 Illustration of Lemma 1.

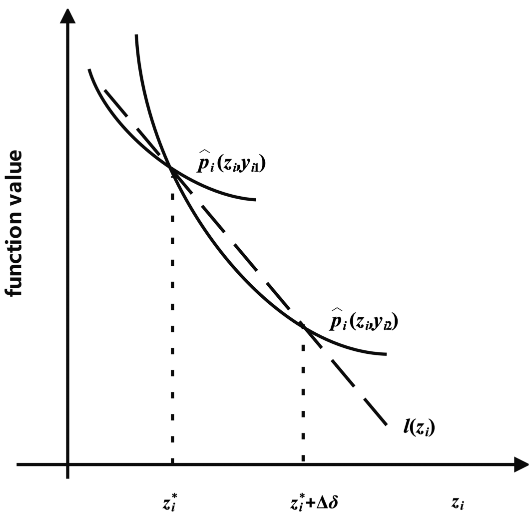

Figure 2 Illustration of the case where two curves intersect and diverge at z_i^*, with solid lines representing $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$, and the dotted line representing $l(z_i)$.

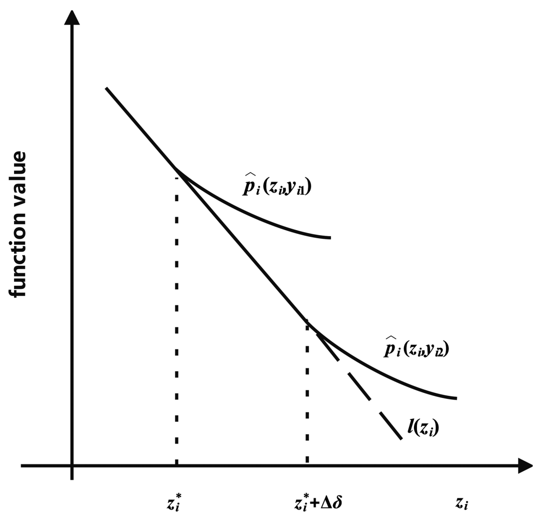

Figure 3 Illustration of the case where two curves intersect and diverge at z_i^*, with solid lines representing $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$, and the dotted line representing the supporting hyperplane $l(z_i)$.

Theorem 2. Suppose $\gamma_1 > \gamma_2$, then there exists a point $z_i^* \in \mathbb{R}$, such that $\forall z_i \in (-\infty, z_i^*]$, $p_i(z_i) = \hat{p}_i(z_i, y_{i1})$; and $\forall z_i \in (z_i^*, \infty]$, $p_i(z_i) = \hat{p}_i(z_i, y_{i2})$.

Proof. In the case where $\delta_1 \geqslant \delta_2$, it is evident that $p_i(z_i) = \hat{p}_i(z_i, y_{i2})$. Conversely, in the case where $\delta_1 < \delta_2$, let $\Delta\delta = \delta_2 - \delta_1$. Without loss of generality, assume that the two curves intersect. If the two curves continue to overlap, then the theorem's conclusion is self-evident. Hence, we focus exclusively on the case where the two curves intersect, without necessarily coinciding at all points. Let z_i^* denote a point such that $p_{i1}(z_i^*) = p_{i2}(z_i^*)$.

First, we consider the case where $z_i \in (z_i^*, +\infty]$. We can infer the existence of a line $l(z_i)$ intersecting the points $(z_i^*, p_{i2}(z_i^*))$ and $(z_i^* + \Delta\delta, p_{i2}(z_i^* + \Delta\delta))$. There are two possible cases: first, when $l(z_i)$ is not a supporting hyperplane of $\hat{p}_i(z_i, y_{i2})$, as shown in Figure 2; and second, when $l(z_i)$ is a supporting hyperplane of $\hat{p}_i(z_i, y_{i2})$, as shown in Figure 3.

Next, we concentrate on the interval $(z_i^*, z_i^* + \Delta\delta]$. Initially, According to Lemma 1, we have $l(z_i) \geqslant \hat{p}_i(z_i, y_{i2})$ on $(z_i^*, z_i^* + \Delta\delta]$ and $\hat{p}_i(z_i, y_{i2}) \geqslant l(z_i)$ on $(z_i^* + \Delta\delta, \infty]$. Based on the fact that $\hat{p}_i(z_i, y_{i1})$ results from translating $\hat{p}_i(z_i, y_{i2})$, where the translation vector is $(-\Delta\delta, p_{i2}(z_i^*) - p_{i2}(z_i^* + \Delta\delta))$, we conclude that $\hat{p}_i(z_i, y_{i1}) \geqslant l(z_i) \geqslant \hat{p}_i(z_i, y_{i2})$ on $(z_i^*, z_i^* + \Delta\delta]$.

The case within the interval $(z_i^* + \Delta\delta, \infty)$ remains to be examined. Given that $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$

are monotonically decreasing and convex over $(z_i^* + \Delta\delta, \infty)$, it follows that $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$ are finite and convex functions on this open interval. Based on Theorem 25.3 in [24], it can be concluded that the derivatives of $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$ exist almost everywhere and are non-decreasing. Therefore, $\hat{p}_i(z_i, y_{i1})$ and $\hat{p}_i(z_i, y_{i2})$ are Lipschitz continuous, which implies they are also absolutely continuous. According to the generalized Newton-Leibniz formula, we can get

$$\hat{p}_i(z_i, y_{ij}) - \hat{p}_i(z_i^* + \Delta\delta, y_{ij}) = \int_{z_i^*+\Delta\delta}^{z_i} \hat{p}_i'(z_i, y_{ij})\mathrm{d}z_i,\ j = 1, 2, \tag{8}$$

where $\hat{p}_i'(z_i, y_{ij})$ represents derivative of $\hat{p}_i(z_i, y_{ij})$ and $z_i \in [z_i^* + \Delta\delta, \infty)$. Given that $\hat{p}_i'(z_i, y_{ij})$ is non-decreasing and that $\hat{p}_i(z_i, y_{i1})$ is obtained by translating $\hat{p}_i(z_i, y_{i2})$, it follows that $\hat{p}_i'(z_i, y_{i2}) \leqslant \hat{p}_i'(z_i, y_{i1})$ for $z_i \in [z_i^* + \Delta\delta, \infty)$, which holds almost everywhere. Consequently, $\int_{z_i^*+\Delta\delta}^{z_i} \hat{p}_i'(z_i, y_{i2})\mathrm{d}z_i \leqslant \int_{z_i^*+\Delta\delta}^{z_i} \hat{p}_i'(z_i, y_{i1})\mathrm{d}z_i$, $z_i \in [z_i^* + \Delta\delta, \infty)$. From the above discussion, it can be concluded that $\hat{p}_i(z_i, y_{i2}) \leqslant \hat{p}_i(z_i, y_{i1})$, $\forall z_i \in (z_i^* + \Delta\delta, \infty)$.

For the case where $z_i \in (-\infty, z_i^*]$, it can be proved, similar to the proof for the case $(z_i + \Delta\delta, \infty)$, that $\hat{p}_i(z_i, y_{i2}) \geqslant \hat{p}_i(z_i, y_{i1})$. Thus the aforementioned discussion concludes the proof.

Since we can compare $\hat{p}_i(z_i, y_{ij})$ and $\hat{p}_i(z_i, y_{ik})$ pairwise, the proof could be extended to the case where $y_{ij} \in Y_i = \{y_{i1}, \ldots, y_{iJ}\}$. According to Theorem 2, $p_i(z_i)$ comprises different functions $\hat{p}_i(z_i, y_{ij})$ defined over distinct intervals. This aligns with the derivative calculation of $p_i(z_i)$ described in Theorem 1.

3 Algorithm and convergence analysis

During the process of computing the derivative of $p_i(z_i)$, we held the integer variable y_i constant, thereby transforming $p_i(z_i)$ into a convex function. Moreover, under the separation assumption, $p_i(z_i)$ can be construed as a composition of different convex functions. Hence, at any given z_i, we can devise a point-to-set mapping $Y_i(z_i) : \mathbb{R} \to Y_i$, ensuring that

$$p_i(z_i) = \hat{p}_i(z_i, y_i),\ \forall y_i \in Y_i(z_i).$$

Correspondingly, for a given y_i, $\hat{p}_i(z_i, y_i)$ serves as an overestimate of $p_i(z_i)$. To streamline the discussion, let $z(k) = (z_1(k), \ldots, z_N(k))$, $y(k) = (y_1(k), \ldots, y_N(k))$, $p(z(k)) = \sum_{i=1}^N p_i(z_i(k))$, and $\hat{p}(z(k), y(k)) = \sum_{i=1}^N \hat{p}_i(z_i(k), y_i(k))$. Then, based on the preceding discussion, an overestimation-based algorithm is formulated as Algorithm 1. Under certain assumptions, the following convergence theorem for the algorithm can be established.

Theorem 3. Let the sequence $\{z(k)\}$ be generated by Algorithm 1. Then every limit point of $\{z(k)\}$ is a locally optimal point. And $p(z(k))$ converges in finite time.

Proof. First, since $z(k+1)$ is the optimal solution of problem (9), we get $\hat{p}(z(k+1), y(k)) \leqslant \hat{p}(z(k), y(k))$. Based on the definition of $y(k)$, we get a series of inequalities as follows:

$$p(z(k+1)) \leqslant \hat{p}(z(k+1), y(k)) \leqslant \hat{p}(z(k), y(k)) = p(z(k)).$$

Then $\{p(z(k))\}$ is a monotonically decrease sequence, and also bounded. So, $p(z(k))$ converges to some value p^*. Next, we will prove the local optimality of p^*.

Clearly, according to Theorem 2, for each $p_i(z_i)$, we could split $\mathbb{R}$ into finite mutually disjoint intervals $\{[a_{il}, b_{il})\}$, $l = 1, \ldots, L$. For convenience, one of the a_{il} is set to $+\infty$. And on each interval $[a_{il}, b_{il})$, $p_i(z_i)$ is convex. So, as for each $z_i(k)$, without generality, we adopt $[a_{il}, b_{il})$ as the interval, and $\hat{p}(z(k), y(k))$ is the overestimate of $p(z(k))$ correspondingly. Since the number of all the intervals is finite, the $\hat{p}(z(k), y(k))$ has a limited number of possibilities and optimal values.

First, we consider the case that the equality $\hat{p}(z(k+1), y(k)) = \hat{p}(z(k), y(k))$ holds. Then we have already achieved local optimality, and $z(k)$ is a locally optimal point indeed. Next, suppose $\hat{p}(z(k+1), y(k)) < \hat{p}(z(k), y(k))$. If the $z(k+1)$ and $z(k)$ are located within the Cartesian product of the same intervals, we could get $\hat{p}(z(k+1), y(k)) = \hat{p}(z(k+1), y(k+1))$. Since $z(k+1)$ is an optimal point of problem (9), we have $\hat{p}(z(k+1), y(k+1)) = \hat{p}(z(k+2), y(k+1))$. This situation is equivalent to the first case.

What remains is the case where $\hat{p}(z(k+1), y(k)) < \hat{p}(z(k), y(k))$, with $z(k+1)$ and $z(k)$ situated within distinct intervals. As the combinations of intervals are finite, there exists a K such that the intervals

Algorithm 1 Iterative solving method

Require: Initial $k = 0$.

1: **while** $p(z(k))$ does not converge **do**

2: For each $z_i(k)$, if $Y_i(z_i(k))$ contains more than one element, select $y_i \in Y_i(z_i(k))$ such that $y_i \neq y_i(k)$, and set $y_i(k) = y_i$. If $Y_i(z_i(k))$ has only one element, choose $y_i(k) \in Y_i(z_i(k))$. This defines the function $\hat{p}_i(z_i, y_i(k))$.

3: Solve the convex optimization

$$\begin{aligned} \min_{\{z_i\}_{i=1}^N} \quad & \sum_{i=1}^N \hat{p}_i(z_i, y_i(k)) \\ \text{s.t.} \quad & \sum_{i=1}^N z_i = 0, \end{aligned} \tag{9}$$

and get the optimal solution $z(k+1)$. Let $k = k+1$.

4: **end while**

where $z(K)$ is located have previously occurred. Suppose $z(K)$ and $z(L)$ satisfy the aforementioned condition. Then, based on the inequality

$$\hat{p}(z(K+1), K) \leqslant \hat{p}(z(K), K) \leqslant \cdots \leqslant \hat{p}(z(L), L),$$

we get equality $\hat{p}(z(K+1), K) = \cdots = \hat{p}(z(L+1), L)$, which turns to be the first case. Thus, we complete the proof.

In each iteration of Algorithm 1, problem (9) must be solved first, making it crucial to address this problem efficiently. Specifically, the subgradient projection method can be employed, where Theorem 1 indicates that the subgradient of $\hat{p}_i(z_i, y_i(k))$ corresponds to the negative value of the Lagrange multiplier for the problem

$$\begin{aligned} \min_{(x_i, v_i)} \quad & f_i(x_i, y_i(k)) + M v_i \\ \text{s.t.} \quad & g_i(x_i, y_i(k)) - v_i \leqslant z_i, \\ & x_i \in X_i, \ v_i \geqslant 0. \end{aligned} \tag{10}$$

Solving the problem is much more efficient due to its convex nature and smaller scale. Additionally, Theorem 2 shows that the function $Y_i(z_i)$ can be computed independently of the iterative process, which allows for its precomputation prior to the iterative steps of Algorithm 1.

Notably, the computation of $p_i(z_i)$ can be executed independently, and the main problem (3) is coupled solely by the constraint $\sum_{i=1}^N z_i = 0$. Consequently, this problem can be effectively addressed through the collaborative processing capabilities of multi-agents. Distributed optimization is precisely a method for solving large-scale and complex problems through cooperation and coordination among multi-agents. With distributed methods, each agent only needs to know its own subproblem (4) and communicate with neighboring agents to collaboratively solve the main problem (3). The communication network is modeled as a connected, undirected graph $\mathcal{G} = (\{1, \ldots, N\}, \mathcal{E})$, where $\mathcal{E} \subseteq \{(i,j) \mid i, j \in \{1, \ldots, N\}\}$ denotes the set of edges. The set of neighbors of agent i is given by $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$. Next, the running process of the distributed algorithm proposed as [12,14] for solving problem (1) will be described in detail. First, each agent i computes $p_i(z_i)$ independently and subsequently calculates $\lambda_i(k)$ as the Lagrange multiplier for the problem $\hat{p}_i(z_i, y_i(k))$. Then, each agent i receives $\lambda_j(k)$ from its neighbors $j \in \mathcal{N}_i$ and updates $z_i(k)$ as

$$z_i(k+1) = z_i(k) - \alpha(k) \sum_{j \in \mathcal{N}_i} \left(\lambda_i(k) - \lambda_j(k)\right), \tag{11}$$

where $\alpha(k)$ is the step-size, satisfying $\alpha(k) \geqslant 0$, satisfies $\sum_{t=0}^{\infty} \alpha(k) = \infty, \sum_{t=0}^{\infty} \alpha^2(k) < \infty$. Overall, each agent in the network operates independently, exchanging only essential information with its neighboring agents. The distributed algorithm is presented as Algorithm 2.

Under some assumptions, we can get that $z(k)$ converge to the optimal point of some problem (5).

Theorem 4. Let Assumptions 1–3 hold. Let the sequence $\{z(k)\}$ be generated by Algorithm 2. $\{z(k)\}$ converges to an optimal point of some problem (9).

Proof. We first prove that the subgradient of each function $\hat{p}_i(z_{i1}, y_{ij})$ is greater than $-M$. Assume, for the sake of contradiction, that the subgradient $\hat{p}'_i(z_i, y_{ij})$ of $\hat{p}_i(z_i, y_{ij})$ is less than $-M$ at z_{i1}. Let $z_{i2} < z_{i1}$, then we have

$$\hat{p}_i(z_{i2}, y_{ij}) - \hat{p}_i(z_{i1}, y_{ij}) \geqslant \hat{p}'_i(z_{i1}, y_{ij})(z_{i2} - z_{i1}) > M(z_{i1} - z_{i2}).$$

Algorithm 2 Distributed decomposition method

Require: Initial $k = 0$, $\sum_{i=1}^{N} z_i(k) = 0$.
while $z(k)$ does not converge **do**
  For each agent i, given $z_i(k)$, choose $y_i(k) \in Y_i(z_i(k))$, and get $\hat{p}_i(z_i, y_i(k))$ and multiplier $\lambda_i(k)$;
  Receive $\lambda_i(k)$ from $j \in \mathcal{N}_i$ and update $z_i(k+1)$ with (11);
  Let $k = k + 1$;
end while
With $z_i(k)$, calculate $p_i(z_i(k))$.

However, according to the definition of $\hat{p}_i(z_i, y_{ij})$, we have $\hat{p}_i(z_{i2}, y_{ij}) \leqslant \hat{p}_i(z_{i1}, y_{ij}) + M(z_{i2} - z_{i1})$, leading to a contradiction. Therefore, we have proven that the subgradient of each function $\hat{p}_i(z_{i1}, y_{ij})$ is greater than $-M$. Furthermore, since each function $\hat{p}_i(z_i, y_{ij})$ is non-increasing, it follows that the subgradient of each function $\hat{p}_i(z_{i1}, y_{ij})$ is bounded. By Theorem 2, for each i, $\mathbb{R}$ can be partitioned into a finite number of mutually disjoint intervals, on each of which $p_i(z_i)$ is convex. Therefore, by applying the reasoning from the proof of Theorem II.6 in [12], the sequence z_k is shown to converge to the optimal solution of problem (9), thus completing the proof.

Both Algorithms 1 and 2 employ the classical projected subgradient method, for which the exact subgradient of the function $p_i(zi)$ is necessary. On the other hand, it is rather difficult to directly obtain the subgradient of $p_i(z_i)$ since the exact information of $p_i(z_i)$ is unknown. Therefore, to successfully apply the projected subgradient method in Algorithms 1 and 2, we provide the calculation method for the subgradient as outlined in Theorems 1 and 2. The above analysis of the function $p_i(z_i)$ confirms the effectiveness of the algorithms and the suitability of the subgradient projection method for this problem.

Remark 2. It is also worth mentioning that the decoupled main problem (3) is a continuous programming problem being proved locally convex in Theorem 2, and the decoupled subproblem (4) is a small-scale MIP problem. Consequently, addressing problems (3) and (4) alternately is more efficient than solving the large-scale MIP problem (1) directly. Since the complexity of solving MIP problems grows exponentially with the increasing problem scale.

4 Simulation

In this section, we provide a numerical simulation that illustrates the effectiveness of the algorithm we have proposed. Consider the 0-1 MIP problem which is shown as

$$\begin{aligned} \min_{(x_i, y_i)} \quad & \sum_{i=1}^{N} (a_i x_i^{b_i} + c_i y_i) \\ \text{s.t.} \quad & \sum_{i=1}^{N} (d_i x_i + e_i y_i) \leqslant \sigma, \\ & (x_i, y_i) \in X_i \times Y_i, \ i \in \{1, \ldots, N\}, \end{aligned} \tag{12}$$

where a_i, b_i, c_i, d_i, e_i are randomly selected from the sets $\{1, 4, 5\}$, $\{4, 2\}$, $\{1, 0.5\}$, $\{1, 3, 2\}$, $\{-1, -0.5\}$ respectively, and σ is set as -1.5. Additionally, for all $i = 1, 2, \ldots, N$, $X_i = [-3, 0]$, and $Y_i = \{0, 1\}$. Then, the decomposition subproblem $p_i(z_i)$ is written as

$$\begin{aligned} \min_{(x_i, y_i)} \quad & a_i x_i^{b_i} + c_i y_i + M v_i \\ \text{s.t.} \quad & d_i x_i + e_i y_i - v_i \leqslant z_i, \\ & (x_i, y_i) \in X_i \times Y_i, \ v_i \geqslant 0, \end{aligned}$$

where each $z_i \in [-4, 1]$. To further illustrate Theorem 2, we draw $p_1(z_1)$ when y_1 takes the value 1 or 0 respectively, which is shown in Figure 4.

According to Figure 4, two curves only have one intersection $(0, 1)$. When $z \leqslant 0$, $p_1(z) = \hat{p}_1(z, 1)$, and when $z > 0$, $p_1(z) = \hat{p}_1(z, 0)$, where $p_1(z, 0)$ and $p_1(z, 1)$ are defined as (5). All these are consistent with what is described in Theorem 2. Next, to illustrate the validity of the two algorithms, we compare the solving time and solving accuracy with Gurobi for different N. The distributed algorithm is based on the communication topology illustrated in Figure 5, where the number of agents, denoted by N, corresponds

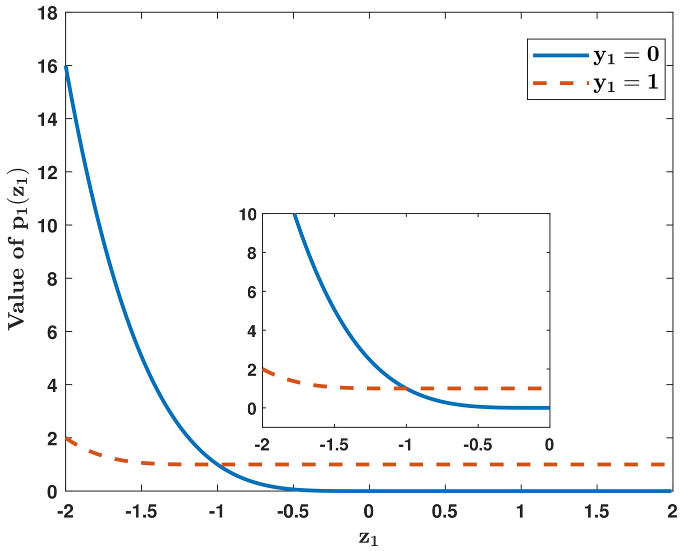

Figure 4 (Color online) Illustration of $p_1(z_1)$ with fixed $y_1 = 0$ or $y_1 = 1$.

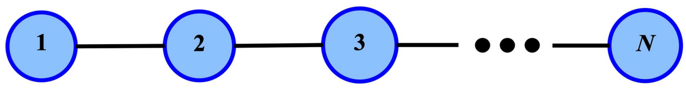

Figure 5 (Color online) Graph of communication topology.

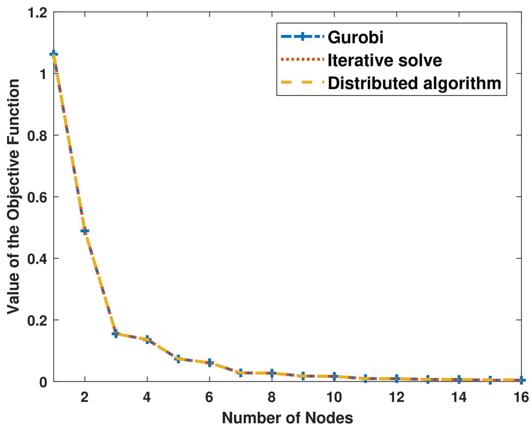

Figure 6 (Color online) Comparison of optimal values obtained from three methods.

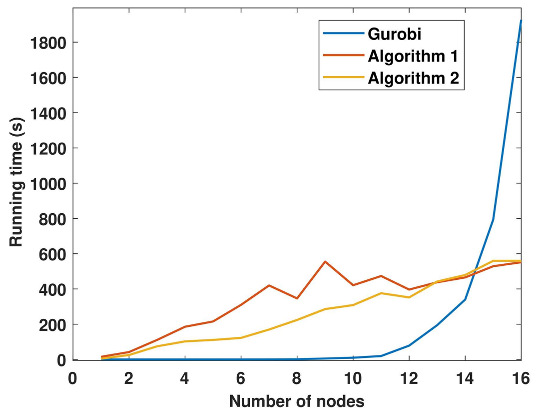

Figure 7 (Color online) Comparison of running time obtained from three methods.

to the number of subproblems as defined in (4). The update rule governing the behavior of each agent is specified in (11). For convenience, we execute the distributed algorithm agent by agent, resulting in a total running time that is the sum of the running time of all individual agents. For a single agent, the running time can be estimated by averaging. Since the proposed algorithm yields a local optimal solution, we execute the algorithm with different initial points and select the smallest value from the results of multiple runs. Each initial point is obtained through uniform sampling over the domain. During the process of varying N from 4 to 16, we select 100 distinct initial values by sampling. As for Algorithms 1 and 2, we both use the gradient projection method, where the gradient(multiplier) is calculated by the optimization toolbox offered by Matlab. Using the optimal value obtained by Gurobi as a benchmark, we list the results in Figures 6 and 7.

The results obtained by the three methods are basically consistent. It can be seen from Figures 6 and 7 that when N is small, the proposed algorithms require more running time to obtain the optimal solution compared to Gurobi. Although their solution accuracy is comparable to that of Gurobi. On the other hand, when $N \geqslant 15$, the running time of Gurobi grows more rapidly with increasing N compared to

Table 1 Simulation results with the same running time

	$N = 400$		$N = 500$		$N = 600$		$N = 700$	
	Optimal value	Gap (%)	Optimal value	Gap (%)	Optimal value	Gap (%)	Optimal value	Gap (%)
Gurobi	519.1352	0.0000	461.0271	0.0000	406.6998	0.0000	354.7986	0.0000
Algorithm 1	519.1509	0.0030	457.9358	0.6705	402.2658	1.0902	357.7518	0.8324
Algorithm 2	519.1384	0.0006	457.9346	0.6708	402.2596	1.0918	353.6415	0.3261

Algorithms 1 and 2. Further, for large N, with a fixed running time of 200 s and across different nodes, we present the optimal values obtained by each of the three methods in Table 1. The gap is calculated as the difference between the optimal values of the tested algorithm and Gurobi. It can be seen from Table 1 that, under the same time limit, the convergence accuracy of the proposed algorithms is better than that of Gurobi in most cases, which implies the good performance of the proposed algorithms for large-scale problems.

5 Conclusion

This paper addresses a specific class of mixed-integer programming problems relevant to power systems, supply chain management, and energy system optimization. These problems feature summation-based objective functions and constraints. Due to their large scale, direct solving is inefficient, prompting the use of the right-hand side allocation method for decomposition. However, the properties of the decomposed subproblems and effective algorithm design remain underexplored. We first investigate the continuity and differentiability of the subproblems post-decomposition and develop methods for computing derivatives. Next, we refine the subproblems for cases where continuous and integer variables are separated, proving their local convexity. We then design algorithms using these properties and employ distributed methods. Finally, we demonstrate that our proposed algorithm, which outperforms the Gurobi solver in handling large-scale mixed-integer programming problems with nonlinear objectives and constraints, achieves superior accuracy and faster execution, as shown by simulation results.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2022ZD0120001), Jiangsu Provincial Scientific Research Center of Applied Mathematics (Grant No. BK20233002), National Natural Science Foundation of China (Grant Nos. 62233004, 62073076, 62203110), and Natural Science Foundation of Jiangsu Province (Grant No. BK20242027).

References

1 Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman, 1979
2 Land A H, Doig A G. An Automatic Method for Solving Discrete Programming Problems. Berlin: Springer, 2010
3 Balas E, Ceria S, Cornuéjols G, et al. Gomory cuts revisited. Oper Res Lett, 1996, 19: 1–9
4 Gupta O K, Ravindran A. Branch and bound experiments in convex nonlinear integer programming. Manage Sci, 1985, 31: 1533–1546
5 Geoffrion A M. Generalized benders decomposition. J Optim Theor Appl, 1972, 10: 237–260
6 Li X, Tomasgard A, Barton P I. Nonconvex generalized benders decomposition for stochastic separable mixed-integer nonlinear programs. J Optim Theor Appl, 2011, 151: 425–454
7 Westerlund T, Pettersson F. An extended cutting plane method for solving convex MINLP problems. Comput Chem Eng, 1995, 19: 131–136
8 Duran M A, Grossmann I E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math Programm, 1986, 36: 307–339
9 Hijazi H, Bonami P, Ouorou A. An outer-inner approximation for separable mixed-integer nonlinear programs. INFORMS J Comput, 2014, 26: 31–44
10 Kronqvist J, Lundell A, Westerlund T. Reformulations for utilizing separability when solving convex MINLP problems. J Glob Optim, 2018, 71: 571–592
11 Bertsekas D P. Nonlinear programming. J Oper Res Soc, 1997, 48: 334
12 Notarnicola I, Notarstefano G. Constraint-coupled distributed optimization: a relaxation and duality approach. IEEE Trans Control Netw Syst, 2019, 7: 483–492
13 Karabulut E, Ahmed S, Nemhauser G. Decentralized algorithms for distributed integer programming problems with a coupling cardinality constraint. Discrete Optim, 2020, 38: 100595
14 Camisa A, Notarnicola I, Notarstefano G. Distributed primal decomposition for large-scale MILPs. IEEE Trans Autom Control, 2022, 67: 413–420
15 Yıldız B, Boland N, Savelsbergh M. Decomposition branching for mixed integer programming. Oper Res, 2022, 70: 1854–1872
16 Bodur M, Ahmed S, Boland N, et al. Decomposition of loosely coupled integer programs: a multiobjective perspective. Math Program, 2022, 196: 427–477
17 Vujanic R, Esfahani P M, Goulart P J, et al. A decomposition method for large scale MILPs, with performance guarantees and a power system application. Automatica, 2016, 67: 144–156
18 Falsone A, Margellos K, Prandini M. A decentralized approach to multi-agent MILPs: finite-time feasibility and performance guarantees. Automatica, 2019, 103: 141–150

19 Testa A, Rucco A, Notarstefano G. Distributed mixed-integer linear programming via cut generation and constraint exchange. IEEE Trans Autom Control, 2019, 65: 1456–1467
20 Xin B, Chen J, Peng Z H, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization. Sci China Inf Sci, 2010, 53: 980–989
21 Zhou Y, He F Z, Qiu Y M. Dynamic strategy based parallel ant colony optimization on GPUs for TSPs. Sci China Inf Sci, 2017, 60: 068102
22 Lv X L, Zhai Q Z, Hu J C, et al. An efficient binary programming method for black-box optimization and its application in processor design. Sci China Inf Sci, 2025, 68: 112101
23 Meyer R. The validity of a family of optimization methods. SIAM J Control, 1970, 8: 41–54
24 Borwein J, Lewis A. Convex Analysis. New York: Springer, 2006