Multiobjective Optimization-Aided Decision-Making System for Large-Scale Manufacturing Planning

Zhenkun Wang, *Member, IEEE*, Hui-Ling Zhen, Jingda Deng, Qingfu Zhang, *Fellow, IEEE*, Xijun Li, Mingxuan Yuan, and Jia Zeng, *Senior Member, IEEE*

***Abstract*—This work is geared toward a real-world manufacturing planning (MP) task, whose two objectives are to maximize the order fulfillment rate and minimize the total cost. More important, the requirements and constraints in real manufacturing make the MP task very challenging in several aspects. For example, the MP needs to cover many production components of multiple plants over a 30-day horizon, which means that it involves a large number of decision variables. Furthermore, the MP task's two objectives have extremely different magnitudes, and some constraints are difficult to handle. Facing these uncompromising practical requirements, we introduce an interactive multiobjective optimization-based MP system in this article. It can help the decision maker reach a satisfactory tradeoff between the two objectives without consuming massive calculations. In the MP system, the submitted MP task is modeled as a multiobjective integer programming (MOIP) problem. Then, the MOIP problem is addressed via a two-stage multiobjective optimization algorithm (TSMOA). To alleviate the heavy calculation burden, TSMOA transforms the optimization of the MOIP problem into the optimization of a series of single-objective problems (SOPs). Meanwhile, a new SOP solving strategy is used in the MP system to further reduce the computational cost. It utilizes two sequential easier SOPs as the approximator of the original complex SOP for optimization. As part of the MP system, TSMOA and the SOP solving strategy are demonstrated to be efficient in real-world MP applications. In addition, the effectiveness of TSMOA is also validated on benchmark problems. The results indicate that TSMOA as well as the MP system are promising.**

***Index Terms*—Integer programming, manufacturing planning (MP), multiobjective optimization, real-world application, two-stage.**

I. Introduction

MANUFACTURING planning (MP) is critical to any industrial company, as it directly affects production efficiency and supply capacity. The MP aims to seek efficient coordination and scheduling of all production activities during the planning period to achieve the company's objectives [1], [2]. Real-world MP tasks usually consider more than one objective, and they are often regarded as multiobjective optimization problems [3], [4]. For example, [5] has two objectives, namely: 1)maximizing the customer satisfaction and 2)minimizing the total cost. In [5], the three objective functions related to economy, environment, and society are optimized simultaneously.

This article investigates a real-world MP task with the two objectives: 1) maximizing the order fulfillment rate and 2) minimizing the total cost. Nevertheless, the requirements and constraints in real manufacturing make it very challenging. First, the MP task involves many decision variables because it needs to cover multiple plants' numerous production items over a 30-day horizon. Besides, its two objectives have badly different magnitudes, and the value ranges have large variations with MP data. In addition, the definition of the order fulfillment rate contains some intractable constraints.

Many kinds of optimization algorithms have been developed, whether for specific problems [6], [7] or general frameworks [8], [9]. But to the best of our knowledge, none of them can handle such MP tasks well. For instance, some multiobjective optimization algorithms [10], [11] proposed in our previous work have rarely considered these real-world challenges (e.g., integer variables and complex constraints). The development of algorithms or systems for such MP tasks has economic value and research significance since many other

Manuscript received June 7, 2020; revised October 18, 2020; accepted January 3, 2021. This work was supported in part by the SUSTech start-up funding; in part by the National Science Foundation of China under Grant 61876163, Grant 11991023, Grant 62076197, Grant 62072364, and Grant 61721002; in part by the ANR/RGC Joint Research Scheme through the Research Grants Council of the Hong Kong Special Administrative Region, China, and France National Research Agency under Project A-CityU101/16; in part by the Major Project of National Science Foundation of China under Grant U1811461; and in part by the Key Project of National Science Foundation of China under Grant 11690011. This article was recommended by Associate Editor F. J. Cabrerizo. *(Corresponding author: Zhenkun Wang.)*

Zhenkun Wang is with the School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China, also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: wangzhenkun90@gmail.com).

Hui-Ling Zhen, Mingxuan Yuan, and Jia Zeng are with Huawei Noah's Ark Lab, Hong Kong (e-mail: zhen.huiling@huawei.com; yuan.mingxuan@huawei.com; zeng.jia@huawei.com).

Jingda Deng is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jddeng@xjtu.edu.cn).

Qingfu Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China (e-mail: qingfu.zhang@cityu.edu.hk).

Xijun Li is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230052, China, and also with the Huawei Noah's Ark Lab, Shenzhen 518173, China (e-mail: xijun.li@huawei.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3049712.

Digital Object Identifier 10.1109/TCYB.2021.3049712

real-world applications (e.g., portfolio optimization [12] and logistic scheduling [13]) also share similar properties.

This article presents an interactive multiobjective optimization-based MP system, to help the dispatcher or decision maker (DM) achieve a satisfactory compromise between the two objectives without causing severe computational burden. Compared with noninteractive algorithms [14]–[17], interactive algorithms [8], [18], [19] require less computation and are more suitable for these MP tasks [14]. Specifically, the submitted MP task is first modeled as a multiobjective integer programming (MOIP) problem [20], [21]. Thereafter, the MOIP problem is solved by a two-stage multiobjective optimization algorithm (TSMOA). Finally, the MP system outputs the solution of the submitted MP task.

In the design of TSMOA, the problem's main challenges and the algorithm's practicability are fully taken into account. TSMOA adopts a two-stage interaction. Its first stage aims to globally approximate the Pareto front of the MOIP problem with a small number of Pareto-optimal solutions. For this purpose, a series of well-designed single-objective problems (SOPs) is constructed for optimization. These SOPs' optimal solutions can give the DM a rough but comprehensive understanding of the entire MP, thus enabling the DM to determine a region of interest (RoI) in the objective space. Within the RoI, the second stage of TSMOA conducts a fine-grained approximation. Here, a modified ϵ-constraint method [22] is used to construct SOPs for optimization. From these SOPs' optimal solutions, the DM can choose the most suitable one as the final solution of the MP task. Compared with the noninteractive MOIP algorithm, TSMOA only needs to obtain fewer Pareto-optimal solutions and, thus, requires less computational effort.

Along with TSMOA, a new SOP solving strategy is introduced in the MP system to lessen the computational burden further. In this strategy, we use two sequential easier SOPs as the approximator of the original complex SOP for optimization. As part of the MP system, TSMOA and the new SOP solving strategy are demonstrated to be efficient in real-world MP applications. Beyond that, in order to illustrate TSMOA's applicability to other problems, we also validate its performance on the benchmark problems.

The remainder of this article proceeds as follows. Section II presents some concepts of the MOIP problem. The MP task is described mathematically in Section III. In Section IV, the proposed MP system is introduced in detail. Section V applies the MP system to some real-world MP tasks. In Section VI, TSMOA is performed and compared on benchmark problems. Some discussions are provided in Section VII. Finally, this article concludes in Section VIII.

II. Background

The MOIP problem can be written as

$$\begin{aligned} \text{minimize} \;\; & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\mathsf{T} \\ \text{subject to} \;\; & \mathbf{x} \in X \end{aligned} \tag{1}$$

where $X \subset \mathcal{Z}^n$ denotes the set of all feasible solutions. For a feasible solution $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T}$, all its decision variables must take non-negative integers. The m objective functions $f_1, \ldots, f_m$ together define the mapping from the decision space to the objective space, and $\mathbf{f}(\mathbf{x})$ is the objective vector corresponding to $\mathbf{x}$. Accordingly, $F = \{\mathbf{f}(\mathbf{x}) : \mathbf{x} \in X\} \subset \mathcal{Z}^m$ is called the feasible set in the objective space.

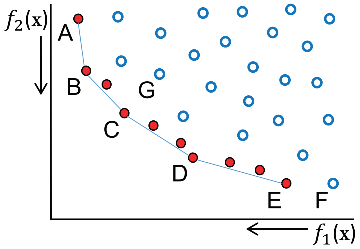


Fig. 1. Illustration of the feasible set of a MOIP in the objective space.

Definition 1: $\mathbf{f}(\mathbf{x}^1)$ is said to *strictly dominate* (weakly dominate) $\mathbf{f}(\mathbf{x}^2)$, if $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ $(f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2))$ for each $i \in \{1, \ldots, m\}$.

Definition 2: $\mathbf{f}(\mathbf{x}^1)$ is said to *dominate* $\mathbf{f}(\mathbf{x}^2)$, if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for each $i \in \{1, \ldots, m\}$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one $j \in \{1, \ldots, m\}$.

Definition 3: $\mathbf{f}(\mathbf{x}^*)$ is said to be *nondominated*, if there is no $\mathbf{f}(\mathbf{x}) \in F$ that can dominate it. Correspondingly, $\mathbf{x}^*$ is called an *efficient* or *Pareto-optimal* solution.

Definition 4: The set of all efficient solutions is represented by X_E, and its image in the objective space (denoted by F_N) is called the *nondominated front* or *Pareto front*.

Definition 5: For a given $\mathbf{f}(\mathbf{x}^*) \in F_N$, $\mathbf{f}(\mathbf{x}^*)$ is called an *unsupported nondominated* objective vector if and only if it can be strictly dominated by any convex combination of the objective vectors in F_N. Otherwise, $\mathbf{f}(\mathbf{x}^*)$ is a *supported nondominated* objective vector. Accordingly, $\mathbf{x}^*$ is called an *unsupported efficient* solution or a *supported efficient* solution.

Definition 6: For any $f_i \in \{f_1, \ldots, f_m\}$, we define its lower bound and upper bound concerning the Pareto front as $f_i^{\min} = \min_{\mathbf{x} \in X_E} f_i(\mathbf{x})$ and $f_i^{\max} = \max_{\mathbf{x} \in X_E} f_i(\mathbf{x})$. For a subregion $[a, b] \subseteq [f_i^{\min}, f_i^{\max}]$, we use $\{\mathbf{f}(\mathbf{x}^{B_1}), \ldots, \mathbf{f}(\mathbf{x}^{B_m})\}$ to denote its *boundary-objective vectors*, where

$$\mathbf{x}^{B_k} = \underset{\substack{a \leq f_i(\mathbf{x}) \leq b \\ \mathbf{x} \in X_E}}{\operatorname{argmin}} \, f_k(\mathbf{x}) \quad \forall k \in \{1, \ldots, m\}. \tag{2}$$

Definition 7: $\mathbf{f}(\mathbf{x}^*)$ is said to be *weakly nondominated*, if there is no $\mathbf{f}(\mathbf{x}) \in F$ that can strictly dominate it. $\mathbf{x}^*$ is called a *weakly efficient* solution.

It is interesting to note that the terms *efficient* and *nondominated* are counterparts of each other. A solution $\mathbf{x}$ is said to be *efficient* if and only if the objective vector $\mathbf{f}(\mathbf{x})$ is *nondominated*. Likewise, *unsupported efficient/nondominated*, *supported efficient/nondominated*, and *weakly efficient/nondominated* are defined in the same manner. More detailed coverage of these definitions can be found in [23] and [24].

From the example in Fig. 1, we can observe that point C strictly dominates point G, and point F is a weakly nondominated objective vector. The Pareto front (F_N) is represented by the solid points. It can be divided into two subsets: points A–E are supported nondominated objective vectors, while the other solid points are unsupported nondominated ones. Points A and E are the boundary-objective vectors of the Pareto front.

As stated in [25], supported nondominated objective vectors have some useful properties. They are located at the "lower left boundary" of the convex hull of the feasible set [26]. Therefore, good global insights about the Pareto front can be gained by finding supported nondominated objective vectors. Furthermore, for any supported nondominated objective vector, its corresponding Pareto-optimal solution must be the optimal solution of (3) with a specific weight vector

$$\begin{aligned} &\text{minimize} \;\; g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}) \\ &\text{subject to} \;\; \mathbf{x} \in X \end{aligned} \tag{3}$$

where $\mathbf{w} = (w_1, \ldots, w_m)^\mathsf{T}$ is a weight vector that satisfies $w_i \geq 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$.

III. Problem Description

In this section, the real-world MP task is described mathematically. Here, we adopt "items" to generally refer to all kinds of production components (e.g., raw materials, semi finished products, and finished products). With the items, each plant has the following production activities.

1) Store the items transported from other plants or purchased from the market in the warehouse.
2) Transport the items from the warehouse to other plants.
3) Supply customers with the items in the warehouse to fulfill their orders.
4) Manufacture products using the items in the warehouse. Newly manufactured products are treated as inbound items to be stored in the warehouse first.

All these production activities serve two objectives: 1) maximizing the order fulfillment rate and 2) minimizing the total cost.

Beyond that, the constraints in real manufacturing are also taken into account. For example, the maximum production capacity of each plant cannot be exceeded. Meanwhile, each plant has its minimum production lot-size for each item. These constraints are all related to the MP data described in Table I. Each MP data is a constant provided by the company. The indices $i \in I$, $p \in P$ and $t = 1, 2, \ldots, T$ are used to denote item i, plant p, and the t-th day, where $T = 30$ and I and P are the sets of all the items and all the plants.

There are four types of decision variables in the MP task, which are described as follows.

$z_{i,p,t}$: The quantity of item i supplied to the customers by plant p on the t-th day.

$s_{i,p,t}^{p'}$: The quantity of item i transported from plant p' to plant p on the t-th day.

$x_{i,p,t}$: The number of batches of item i produced by plant p on the t-th day.

$r_{i,p,t}^{i'}$: The quantity of item i' used to replace item i for the production in plant p on the t-th day.

TABLE I
Descriptions of the MP Data

Notation	Description
$D_{i,t}$	Quantity of item i demanded on the t-th day according to the customer orders
$PC_{i,p}$	Unit cost of producing item i in plant p
$PT_{i,p}$	Unit time of producing item i in plant p
PM_i	Quantity of item i produced per batch
$MLS_{i,p,t}$	Minimum production quantity of item i in plant p on the t-th day
$U_{i,p}$	Unit capacity occupied to produce item i in plant p
$CAP_{p,t}$	Maximum production capacity of plant p on the t-th day
$B_{i,p,t}^{i'}$	Quantity of item i' consumed to produce item i in plant p on the t-th day
$PAIR_{i,p}^{i'}$	Production quantity of item i' paired in the production of item i in plant p
$BIB_{i,p,t}$	Quantity of item i inbound to plant p through procurement on the t-th day
$HC_{i,p}$	Unit cost of storing item i in plant p
$TC_{i,p}^{p'}$	Unit cost of transporting item i from plant p' to plant p
$LT_{p}^{p'}$	Transportation time from plant p' to plant p
$V_{i,p}$	Initial inventory quantity of item i in plant p
M_i	Initial backordered quantity of item i
$R_{i,p,t}$	Maximum replacement quantity of item i in plant p on the t-th day

A. Objective Functions

The first objective function of the MP task defines the order fulfilment rate (i.e., average satisfaction of the order demands), and it is expressed as

$$\text{maximize} f_{\text{sat}} = \frac{\sum_{i \in I} \sum_{t=1}^{T} u_{i,t}}{|I|T} \tag{4}$$

where $u_{i,t}$ is an intermediate variable that calculates the fill rate of the order demand concerning item i on the t-th day. It is defined as

$$\begin{aligned} u_{i,t} &= \max\left\{ \frac{\sum_{p \in P} z_{i,p,t} - m_{i,t-1} + 1}{D_{i,t} + 1}, 0 \right\} \\ &\forall i \in I, t = 1, 2, \ldots, T. \end{aligned} \tag{5}$$

In (5), adding 1 to both numerator and denominator is to avoid the denominator being 0. The adoption of the max function is to prevent $u_{i,t}$ from being negative. $m_{i,t-1}$ represents the backordered quantity of item i in the previous $t - 1$ days. Its definition is given as

$$\begin{cases} m_{i,0} = M_i \\ m_{i,t} = m_{i,t-1} + D_{i,t} - \sum_{p \in P} z_{i,p,t} \\ m_{i,t} \geq 0 \quad \forall i \in I, t = 1, 2, \ldots, T. \end{cases} \tag{6}$$

The second objective function of the MP task calculates the total cost, which includes the production cost (Pcost), transportation cost (Tcost) and holding cost (Hcost). It is written as

$$\text{minimize}\; f_{\text{eco}} = P\text{cost} + T\text{cost} + H\text{cost} \tag{7}$$

where Pcost and Tcost are defined in (8) and (9), respectively. While (10) defines Hcost as the sum of the holding costs incurred during inventory (IHcost), production (PHcost), and transportation (THcost). Their corresponding definitions are given in (11)–(13), respectively

$$P\text{cost} = \sum_{i\in I}\sum_{p\in P} PC_{i,p}\sum_{t=1}^{T}\hat{x}_{i,p,t}. \tag{8}$$

$$T\text{cost} = \sum_{i\in I}\sum_{p'\in P}\sum_{p\in\{P\backslash p'\}} TC_{i,p}^{p'}\sum_{t=1}^{T} s_{i,p,t}^{p'}. \tag{9}$$

$$H\text{cost} = IH\text{cost} + PH\text{cost} + TH\text{cost}, \tag{10}$$

$$IH\text{cost} = \sum_{i\in I}\sum_{p\in P} HC_{i,p}\sum_{t=1}^{T} v_{i,p,t}, \tag{11}$$

$$PH\text{cost} = \sum_{i\in I}\sum_{p\in P} HC_{i,p}PT_{i,p}\sum_{t=1}^{T}\hat{x}_{i,p,t}, \tag{12}$$

$$TH\text{cost} = \sum_{i\in I}\sum_{p\in P} HC_{i,p}\sum_{p'\in\{P\backslash p\}} LT_{p}^{p'}\sum_{t=1}^{T} s_{i,p,t}^{p'}. \tag{13}$$

$\hat{x}_{i,p,t}$ is an intermediate variable [defined in (14)] that represents the quantity of item i produced in plant p on the t-th day. The intermediate variable $v_{i,p,t}$ denotes the inventory quantity of item i in plant p on the t-th day, and its definition is given in (17).

B. Constraints

In the production of each item in each plant, the lot-size and minimum production constraints are considered, that is

$$\hat{x}_{i,p,t} = \begin{cases} 0, & \text{if } x_{i,p,t}\text{PM}_i < MLS_{i,p,t} \\ x_{i,p,t}\text{PM}_i, & \text{otherwise} \end{cases}$$
$$\forall i \in I \quad \forall p \in P, t = 1, 2, \ldots, T. \tag{14}$$

It indicates in (14) that the production is executed only when $\hat{x}_{i,p,t}$ is greater than the minimum production quantity $MLS_{i,p,t}$. Moreover, the capacity constraints of each plant are taken into account, as shown in the following equation:

$$\sum_{i\in I} U_{i,p}\hat{x}_{i,p,t} \le \text{CAP}_{p,t} \quad \forall p \in P, t = 1, 2, \ldots, T. \tag{15}$$

In this MP, the paired production is also considered. The production of one item needs to match a certain number of production of another item, such as one charger should match one phone. The corresponding constraints are given as

$$\hat{x}_{i',p,t} = \text{PAIR}_{i,p}^{i'}\hat{x}_{i,p,t} \quad \forall i', i \in I \quad \forall p \in P, t=1, 2, \ldots, T. \tag{16}$$

The inventory level of each item in each plant on each day is calculated as

$$\begin{cases} v_{i,p,0} = V_{i,p} \\ v_{i,p,t} = v_{i,p,t-1} + in_{i,p,t} - \text{out}_{i,p,t} \\ v_{i,p,t} \ge 0 \quad \forall i \in I \quad \forall p \in P, t = 1, 2, \ldots, T \end{cases} \tag{17}$$

where the intermediate variables $in_{i,p,t}$ and $\text{out}_{i,p,t}$ represent the inbound and outbound quantity of item i in plant p on the t-th day. As shown in (18), the inbound of each item consists of three parts: 1) the production inbound (i.e., $\text{PIB}_{i,p,t}$); 2) the transportation inbound (i.e., $\text{TIB}_{i,p,t}$); and 3) the purchase inbound (i.e., $\text{BIB}_{i,p,t}$). Note that Δ_1 and Δ_2 herein represent the time delays caused by public holidays

$$\begin{cases} in_{i,p,t} = \text{PIB}_{i,p,t} + \text{TIB}_{i,p,t} + \text{BIB}_{i,p,t} \\ \text{PIB}_{i,p,t} = \hat{x}_{i,p,t-PT_{i,p}-\Delta_1} \\ \text{TIB}_{i,p,t} = \sum_{p'\in P} s^{p'}_{i,p,t-LT_p^{p'}-\Delta_2} \\ \forall i \in \quad \forall p \in P, t = 1, 2, \ldots, T. \end{cases} \tag{18}$$

Equation (19) calculates the outbound quantity of each item in each plant on each day. It is the sum of the production outbound (i.e., $\text{POB}_{i,p,t}$), the transportation outbound (i.e., $\text{POB}_{i,p,t}$), the replacement outbound (i.e., $\text{ROB}_{i,p,t}$), and the supply quantity to customers (i.e., $z_{i,p,t}$)

$$\begin{cases} \text{out}_{i,p,t} = \text{POB}_{i,p,t}+\text{TOB}_{i,p,t} + \text{ROB}_{i,p,t} + z_{i,p,t} \\ \text{POB}_{i,p,t} = \sum_{i'\in I} B_{i,p,t}^{i'}\hat{x}_{i',p,t} - \sum_{i'\in I} r_{i,p,t}^{i'} \\ \text{TOB}_{i,p,t} = \sum_{p'\in P} s_{i,p',t}^{p} \\ \text{ROB}_{i,p,t} = \sum_{i'\in I} r_{i',p,t}^{i} \\ \forall i \in I \quad \forall p \in P, t = 1, 2, \ldots, T. \end{cases} \tag{19}$$

It reveals in the definition of $\text{POB}_{i,p,t}$ that the consumption of one item can be reduced if it is substituted with another item in the production. This kind of substitution often occurs when some items are insufficient.

The constraints defined in (21) and (21) are to prevent unnecessary and excessive substitution

$$\text{POB}_{i,p,t} + z_{i,p,t} \ge 0 \quad \forall i \in I \quad \forall p \in P, t = 1, 2, \ldots, T. \tag{20}$$

$$\sum_{i'\in I} r_{i,p,t}^{i'} \le R_{i,p,t} \quad \forall i \in I \quad \forall p \in P, t = 1, 2, \ldots, T. \tag{21}$$

Equation (21) indicates that the consumption of one item should be higher than the use of its substitutes. While (21) suggests that the replacement of one item cannot exceed a specific number.

C. Challenges

As described previously, the real-world MP task is modeled as a MOIP problem, where $f_1 = -f_{\text{sat}}$ and $f_2 = f_{\text{eco}}$. The problem's main challenges can be summarized as follows.

1) *Dissimilar and Changeable Magnitudes:* The two objective functions have extremely different magnitudes. The maximum value of f_{sat} is not greater than 1, while the maximum value of f_{eco} can be millions or billions. Moreover, the two objective functions' value ranges vary greatly depending on the MP data.
2) *Many Variables and Constraints:* The real-world MP needs to cover many items of multiple plants over 30 days. As a result, the MOIP problem has to include a large number of decision variables (up to millions) and constraints.
3) *Complex Constraints:* The max function is adopted in (5), which may cause the MOIP problem to be multimodal and nonsmooth. As we know, general MOIP problems are much easier to solve than multimodal or nonsmooth ones.

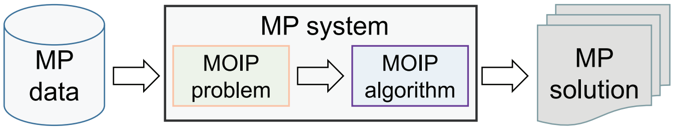


Fig. 2. Workflow of the proposed MP system.

IV. Proposed MP System

To deal with such an MP task, we propose a multiobjective optimization-based MP system. Fig. 2 illustrates the workflow of the MP system. First, the system converts the MP task into a MOIP problem based on the submitted MP data. Then, a MOIP algorithm is employed to solve the MOIP problem. Finally, the system outputs the solution of the MP task.

The details of converting the MP task into a MOIP problem has been presented in Section III. To solve the MOIP problem more efficiently, we adopt a newly designed interactive MOIP algorithm (i.e., TSMOA) in the MP system. The adoption of an interactive algorithm in the system is based on the following two considerations.

1) Compared with the noninteractive algorithm, the interactive algorithm generally only needs to obtain fewer Pareto-optimal solutions. Therefore, the interactive algorithm consumes less computational time, especially when addressing such large-scale and complex MOIP problems.
2) The dispatcher or DM needs to participate in the decision making process of the MP. The interactive algorithm communicates with the dispatcher or DM in a timely manner, thus better avoiding some unexpected situations or industrial disasters.

A. TSMOA

As shown in Fig. 3, TSMOA adopts a two-stage interaction. In the first stage, TSMOA aims to globally approximate the entire Pareto front with a small number of Pareto-optimal solutions. With the knowledge provided by these solutions, the DM can determine an RoI. In the second stage, TSMOA conducts a fine-grained approximation for the RoI. The obtained approximate solutions are provided to the DM, and the DM can choose the most suitable one among them.

In the design of TSMOA, the problem's main challenges and the algorithm's practicability are sufficiently considered.

1) TSMOA has only two loops of interaction. It is because the calculation time for such a large-scale problem in each loop is quite long, and too many interactive loops are a heavy burden for the DM. In this MP task as well as many other real-world scenarios, two interactive loops are sufficient for the DM to reach a final decision.
2) In terms of information interaction, TSMOA adopts a straightforward manner. The algorithm displays the distribution of all the obtained objective vectors to the DM. The DM determines an RoI by giving upper and lower bounds on one objective function. This intuitive interaction is to reduce the DM's cognitive burden and facilitate the DM to make decisions or express preferences.

Algorithm 1: TSMOA-stage1

```
Input: the MOIP problem and the maximal number of the optimized SOPs $SOP_{max}$;
Output: the approximate set $S_1$;
Achieve the boundary solutions $\mathbf{x}^{B_1}$ and $\mathbf{x}^{B_2}$ via lexicographic optimization (cf., Eq. (22));
Let $S_1 = \{(\mathbf{x}^{B_1}, \mathbf{f}(\mathbf{x}^{B_1})), (\mathbf{x}^{B_2}, \mathbf{f}(\mathbf{x}^{B_1}))\}$, $U = \{\overline{\mathbf{f}(\mathbf{x}^{B_1})\mathbf{f}(\mathbf{x}^{B_2})}\}$ and $SOPs = 4$;
while $(SOPs < SOP_{max})$ and $(U \neq \phi)$ do
    Utilize (23) to select the longest line segment from $U$ and denote it as $\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^b)}$;
    Construct a SOP using (25), and optimize it to obtain a solution $\mathbf{x}^*$;
    if $\mathbf{f}(\mathbf{x}^*)$ is different from any objective vector in $S_1$ then
        $S_1 \leftarrow S_1 \cup \{(\mathbf{x}^*, \mathbf{f}(\mathbf{x}^*))\}$,
        $U \leftarrow U \cup \{\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^*)}, \overline{\mathbf{f}(\mathbf{x}^*)\mathbf{f}(\mathbf{x}^b)}\}$;
    end
    $U \leftarrow U \setminus \{\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^b)}\}$, $SOPs \leftarrow SOPs + 1$;
end
```

3) TSMOA globally approximates the Pareto front in its first stage. This is to cope with the challenge caused by the two objective functions' dissimilar and changeable magnitudes. The yielded solutions can give the DM a holistic perception of the value ranges of two objective functions, thereby enabling the DM to determine a RoI.

B. TSMOA-Stage1

As mentioned in Section II, supported nondominated objective vectors have particular advantages in providing global insights about the Pareto front. Therefore, the objective vectors obtained by TSMOA-stage1 are best to be supported nondominated ones. Moreover, the more uniformly these objective vectors are distributed, the better global information they could offer. Last but not least, the Pareto front's boundary-objective vectors have special physical meanings in this MP task. They reveal the production and supply limits, which are indispensable for the DM to make overall considerations.

As shown in Algorithm 1, TSMOA-stage1 first uses the lexicographic optimization method [27] to achieve the boundary solutions/objective vectors. More specifically, each boundary solution $\mathbf{x}^{B_k}$ for $k \in \{1, 2\}$ is obtained by optimizing the following two SOPs:

$$\begin{cases} \mathbf{x}^{B_k} = \operatorname{argmin}_{f_k(\mathbf{x}) \leq f_k(\hat{\mathbf{x}})} f_l(\mathbf{x}) \\ \hat{\mathbf{x}} = \operatorname{argmin} f_k(\mathbf{x}) \\ k, l \in \{1, 2\} \text{ and } k \neq l. \end{cases} \tag{22}$$

Here, we adopt the lexicographic optimization method because it can not only avoid the weakly nondominated objective vectors but also overcome the difficulties caused the dissimilar objective function magnitudes. Note that the constraints introduced in Section III are also included in each SOP, but they are not mentioned here and below for brevity.

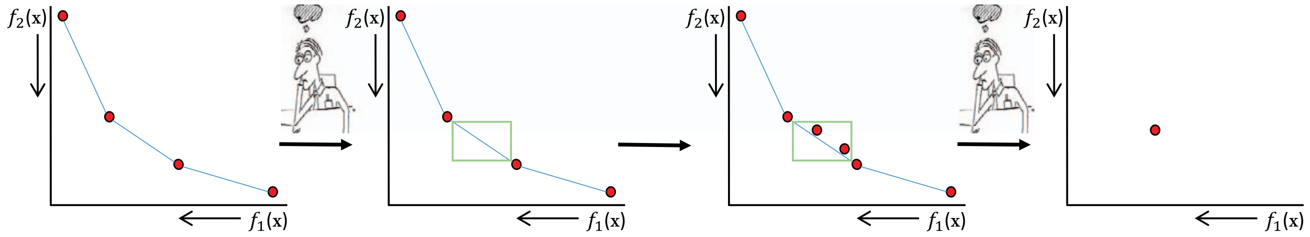


Fig. 3. Illustration of the working principle of TSMOA.

After $\mathbf{x}^{B_1}$ and $\mathbf{x}^{B_2}$ are obtained, the approximate set S_1 is initialized as in line 4 of Algorithm 1. The set U is initialized to $\{\overline{\mathbf{f}(\mathbf{x}^{B_1})\mathbf{f}(\mathbf{x}^{B_2})}\}$, where $\overline{\mathbf{f}(\mathbf{x}^{B_1})\mathbf{f}(\mathbf{x}^{B_2})}$ indicates the line segment between the two objective vectors $\mathbf{f}(\mathbf{x}^{B_1})$ and $\mathbf{f}(\mathbf{x}^{B_2})$. Since four SOPs are already optimized in obtaining the two boundary solutions, the number of optimized SOPs (i.e., SOPs) is set to 4. TSMOA-stage1 then iterates until SOPs reaches a given number $\text{SOP}_{\max}$.

In each iteration, the algorithm selects the longest line segment from U (line 4 of Algorithm 1). Here, we use $\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^b)}$ to represent it, where $\mathbf{f}(\mathbf{x}^a)$ and $\mathbf{f}(\mathbf{x}^b)$ are its two endpoints. The process can be expressed as

$$\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}\left(\mathbf{x}^b\right)} = \underset{\overline{\mathbf{f}(\mathbf{x}')\mathbf{f}(\mathbf{x}'')}\in U}{\operatorname{argmax}} \left\|\bar{\mathbf{f}}\left(\mathbf{x}'\right) - \bar{\mathbf{f}}\left(\mathbf{x}''\right)\right\|_2 \tag{23}$$

where $\bar{\mathbf{f}}(\mathbf{x}')$ and $\bar{\mathbf{f}}(\mathbf{x}'')$ are two normalized objective vectors of $\mathbf{x}'$ and $\mathbf{x}''$, respectively. For a solution $\mathbf{x}$, its normalized objective vector $\bar{\mathbf{f}}(\mathbf{x})$ is defined as

$$\bar{\mathbf{f}}(\mathbf{x}) = \left(\frac{f_1(\mathbf{x}) - f_1\left(\mathbf{x}^{B_1}\right)}{f_1\left(\mathbf{x}^{B_2}\right) - f_1\left(\mathbf{x}^{B_1}\right)}, \frac{f_2(\mathbf{x}) - f_2\left(\mathbf{x}^{B_2}\right)}{f_2\left(\mathbf{x}^{B_1}\right) - f_2\left(\mathbf{x}^{B_2}\right)}\right)^{\mathsf{T}}. \tag{24}$$

With the selected line segment $\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^b)}$, we employ the Aneja and Nair method [28], [29] to construct an SOP, that is

$$\mathbf{x}^* = \operatorname{argmin}(w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x})) \tag{25}$$

where $w_1 = f_2(\mathbf{x}^a) - f_2(\mathbf{x}^b)$ and $w_2 = f_1(\mathbf{x}^b) - f_1(\mathbf{x}^a)$. After the SOP's optimal solution $\mathbf{x}^*$ is obtained, it is compared to the previously gained solutions. If its objective vector $\mathbf{f}(\mathbf{x}^*)$ is different from any objective vector in S_1, $(\mathbf{x}^*, \mathbf{f}(\mathbf{x}^*))$ is added to S_1. Accordingly, the two new line segments $\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^*)}$ and $\overline{\mathbf{f}(\mathbf{x}^*)\mathbf{f}(\mathbf{x}^b)}$ are added to U, $\overline{\mathbf{f}(\mathbf{x}^a)\mathbf{f}(\mathbf{x}^b)}$ is removed from U, and SOPs is added one. If $\mathbf{f}(\mathbf{x}^*)$ is the same as any existing objective vector, U and SOPs are updated as shown in line 11 of Algorithm 1. After all iterations are conducted, the algorithm outputs the approximate set S_1.

In TSMOA-stage1, the Aneja and Nair method is adopted to ensure that the corresponding SOP's optimal solution is a supported efficient solution. The selection of the longest line segment in each iteration is to make the obtained objective vectors distributed as uniformly as possible.

C. TSMOA-Stage2

In our MP system, the DM determines an RoI by specifying the upper and lower bounds on f_2 (denoted by f_2^{up} and f_2^{low}). Then, TSMOA-stage2 conducts a fine-grained approximation for this RoI. Thus, the obtained objective vectors are required to be distributed as uniformly as possible. To achieve this, as well as cope with the problem's particular challenges, we adopt a modified ϵ-constraint method [22] in TSMOA-stage2.

Algorithm 2: TSMOA-stage2

```
Input: the MOIP problem, the upper bound $f_2^{up}$ and the lower bound $f_2^{low}$, the number of intervals $I_{max}$ and the approximate set obtained by TSMOA-stage1 $S_1$;
Output: the approximate set $S_2$ and the number of the optimized SOPs $SOPs$;
Estimate $f_1^{up}$ and $f_1^{low}$ with (27), achieve $\mathbf{x}^{up}$ and $\hat{\mathbf{x}}$ by optimizing the SOPs in (26) and (28), and set $SOPs = 2$;
if $f_2(\hat{\mathbf{x}}) \geq f_2^{low}$ then
    Set $\mathbf{x}^{low}$ to $\hat{\mathbf{x}}$;
else
    Obtain $\mathbf{x}^{low}$ with (29), and $SOPs \leftarrow SOPs + 1$;
end
Let $S_2 = \{(\mathbf{x}^{up}, \mathbf{f}(\mathbf{x}^{up})), (\mathbf{x}^{low}, \mathbf{f}(\mathbf{x}^{low}))\}$, $\epsilon = \frac{f_2(\mathbf{x}^{up}) - f_2(\mathbf{x}^{low})}{I_{max}}$ and $i = 2$;
while $i \leq I_{max}$ do
    $e_i = f_2(\mathbf{x}^{up}) - (i-1) \cdot \epsilon$;
    Construct a SOP using (30), and optimize it to obtain a solution $\mathbf{x}^*$;
    $S_2 \leftarrow S_2 \cup \{(\mathbf{x}^*, \mathbf{f}(\mathbf{x}^*))\}$, $SOPs \leftarrow SOPs + 1$;
    Update $i$ with (31);
end
```

As shown in Algorithm 2, the first step of TSMOA-stage2 is to obtain the boundary solutions/objective vectors of the RoI. The upper boundary solution $\mathbf{x}^{\text{up}}$ is achieved by optimizing the following SOP:

$$\mathbf{x}^{\text{up}} = \underset{f_2(\mathbf{x}) \leq f_2^{\text{up}}}{\operatorname{argmin}} \left(f_1(\mathbf{x}) + \gamma_1 f_2(\mathbf{x})\right) \tag{26}$$

where $\gamma_1 = ([0.1(f_1^{\text{up}} - f_1^{\text{low}})]/[f_2^{\text{up}} - f_2^{\text{low}}])$. f_1^{up} and f_1^{low} are two values that estimated based on the objective vectors achieved by TSMOA-stage1. Assuming that $\mathbf{f}'$ and $\mathbf{f}''$ are the two closest objective vectors above and below $f_2^{\text{up/low}}$, we can estimate $f_1^{\text{low/up}}$ as

$$f_1^{\text{low/up}} = f_1'' - \frac{\left(f_2^{\text{up/low}} - f_2''\right)\left(f_1'' - f_1'\right)}{f_2' - f_2''}. \tag{27}$$

To obtain the lower boundary solution $\mathbf{x}^{\text{low}}$, we first attain an intermediate solution $\hat{\mathbf{x}}$ with the following equation:

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{f_2(\mathbf{x})\le\hat{f}_2} (f_1(\mathbf{x}) + \gamma_1 f_2(\mathbf{x})) \tag{28}$$

where $\hat{f}_2 = f_2^{\text{low}} + ([f_2^{\text{up}} - f_2^{\text{low}}]/2I_{\max})$. If $f_2(\hat{\mathbf{x}}) \ge f_2^{\text{low}}$, $\mathbf{x}^{\text{low}}$ is set to $\hat{\mathbf{x}}$; otherwise, $\mathbf{x}^{\text{low}}$ is gained by optimizing another SOP, that is

$$\mathbf{x}^{\text{low}} = \operatorname*{argmin}_{f_1(\mathbf{x})<f_1(\hat{\mathbf{x}})} (f_2(\mathbf{x}) + \gamma_2 f_1(\mathbf{x})) \tag{29}$$

where $\gamma_2 = (0.01/\gamma_1)$.

After $\mathbf{x}^{\text{up}}$ and $\mathbf{x}^{\text{low}}$ are obtained, the approximate set S_2 and the interval size ϵ are set as in line 9 of Algorithm 2. With ϵ, the RoI is divided into $I_{\max}$ intervals between $f_2(\mathbf{x}^{\text{up}})$ and $f_2(\mathbf{x}^{\text{low}})$. For each interval, TSMOA-stage2 aims to obtain a nondominated objective vector that is closest to its upper bound. Since $\mathbf{f}(\mathbf{x}^{\text{up}})$ is already the best choice for the first interval, we omit it and set the interval index $i = 2$.

The ith interval's upper bound e_i can be calculated as in line 11 of Algorithm 2. Based on e_i, the SOP is defined as

$$\left(\mathbf{x}^*, y^*\right) = \operatorname*{argmin}_{f_2(\mathbf{x})+y=e_i} (f_1(\mathbf{x}) + \gamma_3 \cdot y) \tag{30}$$

where $\gamma_3 = ([0.1(f_2(\mathbf{x}^{\text{up}}) - f_2(\mathbf{x}^{\text{low}}))]/[f_1(\mathbf{x}^{\text{low}}) - f_1(\mathbf{x}^{\text{up}})])$, and $y \ge 0$ is a slack variable. The SOP's optimal solution $\mathbf{x}^*$ is added to S_2 and the interval index i is updated as

$$i = i + 1 + \left\lfloor \frac{y^*}{\epsilon} \right\rfloor. \tag{31}$$

Equation (31) indicates that based on the value of y^*, the intervals containing no nondominated objective vector can be skipped.

TSMOA-stage2 adopts a similar framework to AUGMECON2 [15], a well-known algorithm based on the ϵ-constraint method. In terms of approximating a RoI of the Pareto front, TSMOA-stage2 has two advantages compared to AUGMECON2. TSMOA-stage2 elaborately achieves the boundary solutions and interval division of the RoI, thereby can obtain objective vectors with better distribution. Moreover, TSMOA-stage2 carefully estimates the parameters (i.e., γ_1, γ_2 and γ_3) in the SOPs. Thereby, it can be less troubled by weakly nondominated objective vectors, especially when the two objective functions have badly different magnitudes.

D. SOP Solving Strategy

TMSOA converts the optimization of the MOIP problem into the optimization of several SOPs. Ideally, each SOP solver can serve the purpose. But for such a complex problem in our MP task, the SOP solving strategy needs to be elaborately designed. As stated in [30], the SOP solving strategies can be roughly divided into two categories: 1) mathematical programming solving (MPS) strategies and 2) nonmathematical programming solving (non-MPS) strategies. MPS strategies typically reform a complex SOP into multiple standard or quasi-standard mathematical programming SOPs, and then solve them using well-developed mathematical programming methods (e.g., simplex method [31] and sequential quadratic programming method [32]). Non-MPS strategies mainly include evolutionary algorithms [33], tabu search [34], memetic algorithms [35], and so on. They are usually time consuming and cannot guarantee to find the optimal solution.

We adopt the MPS strategy in our MP system for two reasons. On one hand, we fail to find a non-MPS algorithm or solver that is efficient enough to handle such complex SOPs. On the other hand, the system is ultimately to serve real-world MP applications. The MPS strategy is more stable and robust, which is very important to the company. When solving each SOP, we use two modified easier SOPs as its approximator, and utilize the commercial optimizer Gurobi [36] to solve them sequentially. Gurobi has good performance on this kind of regular SOPs, as it not only integrates plentiful heuristic and mathematical programming methods but also can adaptively match the most suitable solving method for each problem.

As summarized in Section III, the main difficulty in optimizing the SOP stems from the definition of fillrate [i.e., (5)]. In (5), the max function is used to prevent the variables from being negative. But the max function can cause the SOP to be extremely multimodal and nonsmooth, especially it acts on so many variables. Our strategy is to replace the max function by identifying potential variables and adding linear constraints to them. To achieve this, we first modify the SOP by relaxing its definition in (5) as

$$u_{i,t} = \frac{\sum_{p\in P} z_{i,p,t} - m_{i,t-1} + 1}{D_{i,t} + 1} \quad \forall i \in I, t = 1, 2, \ldots, T. \tag{32}$$

Then, we utilize Gurobi to optimize the new SOP to obtain the value of f_{eco} (denoted as f'_{eco}) and the value of $u_{i,t}$ (denoted as $u'_{i,t}$). If $u'_{i,t} > 0$, we believe the variable $u_{i,t}$ is very likely to satisfy the non-negative constraint. Therefore, we can use (33) as an approximation of (5) to define the fillrate

$$\begin{cases} u_{i,t} = \frac{\sum_{p\in P} z_{i,p,t} - m_{i,t-1} + 1}{D_{i,t}+1} \\ u_{i,t} \ge 0 \\ \quad \forall (i,t) \in \left\{(i,t) | u'_{i,t} > 0 \text{ for } i \in I, t = 1, 2, \ldots, T\right\}. \end{cases} \tag{33}$$

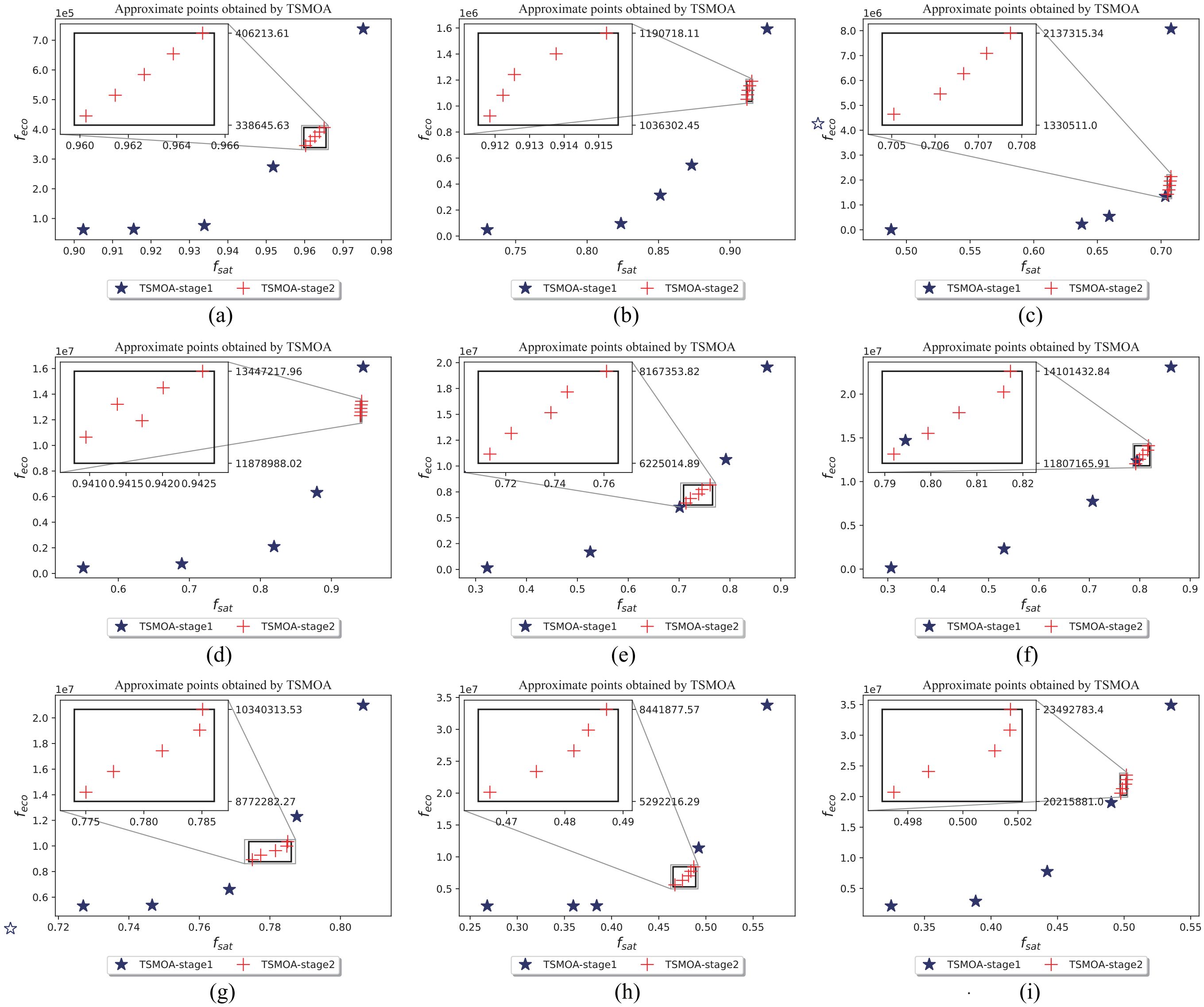
Also with the constraint $f_{\text{eco}} \le f'_{\text{eco}}$, we define the other new SOP to be optimized by Gurobi. The obtained value of $u_{i,t}$ (denoted as $\hat{u}_{i,t}$) may still not satisfy the non-negative constraint. So, we further repair it as

$$u^*_{i,t} = \begin{cases} 0, & \text{if } \hat{u}_{i,t} < 0 \\ \hat{u}_{i,t}, & \text{otherwise.} \end{cases} \tag{34}$$

Finally, the repaired value $u^*_{i,t}$ is used to calculate the value of the objective function f_{sat}.

V. Real-World MP Application

In this section, the proposed MP system is applied to real-world MP tasks. Table II describes the MP data adopted. Columns Item and Plant reveal the number of plants and the number of items involved in the MP task. Columns Planning horizon shows the planning period. Column Scale indicates



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4. Plots of the nondominated points obtained by TSMOA on each MP data. (a) Data-I-1. (b) Data-I-2. (c) Data-I-3. (d) Data-II-1. (e) Data-II-2. (f) Data-II-3. (g) Data-III-1. (h) Data-III-2. (i) Data-III-3.

TABLE II
DESCRIPTION OF THE NINE REAL-WORLD MP DATA

Data name	Item	Plant	Planning horizon	Scale
Data-I-1	35	6	02/11-01/12	10 thousand
Data-I-2	35	6	17/11-16/12	10 thousand
Data-I-3	35	6	02/12-31/12	10 thousand
Data-II-1	996	11	15/12-13/01	1 million
Data-II-2	996	11	14/01-12/02	1 million
Data-II-3	996	11	13/02-15/03	1 million
Data-III-1	4409	20	03/12-01/01	10 million
Data-III-2	4409	20	02/01-31/01	10 million
Data-III-3	4409	20	01/02-02/03	10 million

the scale of the decision variables of the corresponding MOIP problem. In the experiment, we assume that the RoI has a size of $0.1(f_2^{\max}-f_2^{\min})$ and is randomly selected within the range $[f_2^{\min}, f_2^{\max}]$. The parameters $\mathrm{SOP}_{\max}$ in TSMOA-stage1 and $I_{\max}$ in TSMOA-stage2 are set to 7 and 3, respectively.

A. Experimental Results

The experimental results concerning the nine real-world MP data are exhibited in Fig. 4. The approximate objective vectors obtained at the first stage are indicated by pentagrams, while those yielded at the second stage are shown with plus symbols.

From these figures, we can observe that the objective vectors yielded by the first stage are distributed widely and uniformly. These objective vectors not only reveal the maximum production capacity and the maximum order supply capacity of the plants but also roughly show the tradeoff between the total cost and the average order fillrate. With this comprehensive knowledge, the DM is capable of making the most appropriate preference for the RoI.

For the selected RoI, a fine-grained approximation is conducted at the second stage. As shown in Fig. 4, the obtained objective vectors have a uniform spread in the RoI. The DM

TABLE III
STATISTICS OF DOMINATION RELATIONSHIP BETWEEN THE TWO SETS OF APPROXIMATE POINTS ON EACH MP DATA

Problem	Data-I-1	Data-I-2	Data-I-3	Data-II-1	Data-II-2	Data-II-3	Data-III-1	Data-III-2	Data-III-3
N^+	3	10	7	9	9	9	10	9	9
$N^\approx$	7	0	3	1	1	1	0	1	1
N^-	0	0	0	0	0	0	0	0	0

N^+, $N^\approx$, and N^- denote the numbers of approximate points obtained by TSMOA-RXL, which are dominated by, are the same with, and dominate the ones yielded by TSMOA.

can choose the most suitable one from them as the final solution of the MP task. It is worth mentioning that one of the objective vectors achieved in the RoI related to Data-II-1 is not a nondominated one. This is caused by the insufficient accuracy of the commercial solver. Nevertheless, our experiments indicate that this phenomenon rarely happens.

B. Effectiveness of the SOP Solving Strategy

To demonstrate the effectiveness of the new SOP solving strategy, we compare two algorithms: 1) TSMOA and 2) TSMOA-RLX on the MOIP problems corresponding to the nine real-world MP data. In TSMOA, the proposed SOP solving strategy is employed. Considering that the commercial solver cannot solve the original problem, we use the SOP solving strategy of straightforwardly replacing the complex constraints with the relaxed ones [i.e., (32)] in TSMOA-RLX. The other parameter settings of the two algorithms are the same as introduced earlier.

Fig. 8 in the supplementary material plots the objective vectors achieved by the two algorithms on the nine MOIP problems. Besides, Table III reveals the domination relationships between the approximate objective vectors obtained by TSMOA-RXL and those yielded by TSMOA. In addition, Table VI in the supplementary material lists the objective function values of these obtained objective vectors. These results indicate that TSMOA significantly outperforms TSMOA-RXL on all data except the problem related to Data-I-1. The objective vectors achieved by TSMOA are able to dominate most of those attained by TSMOA-RXL. On Data-I-1, TSMOA gains two better objective vectors and eight identical objective vectors compared with TSMOA-RXL. The reason behind is that these solutions' f_{sat} values are very close to the upper bound 1, and the enhancements raised from our SOP solving strategy are not obvious.

All in all, the experimental results indicate that the proposed SOP solving strategy is effective on the MOIP problems related to real-world MP tasks.

C. Computational Time

The experiments in terms of the MP task are conducted on a standalone development server with dual Intel Xeon Gold 6254 3.10-GHz 18-core CPU and 192-GB RAM. The algorithm is implemented in Python. Table IV shows the average computational time of TSMOA-stage1 and TSMOA-stage2 on the three MP data of different sizes. It can be observed that each stage of TSMOA still needs a certain amount of computational time, especially when the scale of the MP data is very large. Nevertheless, considering the complexity and scale of the MP task, such a calculation time is acceptable to the dispatcher or DM.

TABLE IV
AVERAGE COMPUTATIONAL TIME OF TSMOA-STAGE1 AND TSMOA-STAGE2 ON THREE MP DATA OF DIFFERENT SIZES

Data name	TSMOA-stage1	TSMOA-stage2
Data-I	0.13 hours	1.25 hours
Data-II	4.57 hours	11.56 hours
Data-III	12.02 hours	30.93 hours

VI. VALIDATION ON BENCHMARK PROBLEMS

As a newly proposed interactive multiobjective optimization algorithm, TSMOA is by no means only applicable to our MP tasks. It can also work for other MOIP problems with similar properties. To illustrate this, we validate the effectiveness of TSMOA on the benchmark problems.

The bi objective assignment problem (BOAP) [37] is adopted in our investigation. The BOAP is also a MOIP problem, and its scale (dimension of decision variables) can be easily controlled via a parameter. Furthermore, all the supported nondominated objective vectors (denoted as the set F_{SN}) of a BOAP can be enumerated using the Hungarian method [38] and the Fukuda algorithm [39]. They can be used as a reference when comparing algorithms, which is of great importance. The BOAP can be written as

$$\begin{aligned}
\text{minimize}\ \ & \mathbf{f}(\mathbf{x}) = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}^{1} x_{i,j}, \sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}^{2} x_{i,j}\right)^{\mathsf{T}} \\
\text{subject to}\ \ & \sum_{i=1}^{n} x_{i,j} = 1, \quad i = 1, 2, \ldots, n \\
& \sum_{j=1}^{n} x_{i,j} = 1, \quad j = 1, 2, \ldots, n \\
& x_{i,j} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, n \qquad (35)
\end{aligned}$$

where $x_{i,j}$ is the decision variable, and $c_{i,j}^1$ and $c_{i,j}^2$ are the objective function coefficients. As in [26], the coefficients of each objective function are integers generated from the range [0, 20]. In this study, three BOAPs with $n = 100$, 150 and 200 are employed.

A. Overall Comparison

It is known that the comparison of interactive algorithms is troublesome, especially when they are designed for specific applications. Nevertheless, as stated in [14], the effectiveness of interactive algorithms can be demonstrated by showing

TABLE V
NUMBER OF SOPS OPTIMIZED BY EACH ALGORITHM AND THE NUMBER OF NONDOMINATED OBJECTIVE VECTORS IT OBTAINS IN THE ROI OF EACH BOAP

Problem	$n = 100$		$n = 150$		$n = 200$	
Indicator	$SOPs$	NDs	$SOPs$	NDs	$SOPs$	NDs
TSMOA	**12**	**5**	**12**	**5**	**12**	**5**
MOEA/D-WS	504	4	55	**5**	24	**5**
AUGMECON2	39	**5**	50	**5**	44	**5**

advantages over noninteractive algorithms in reducing computational load and cognitive burden. Toward this direction, we adopt two classic noninteractive algorithms: 1) MOEA/D-WS [40] 2) and AUGMECON2 [15] for comparison.

1) In MOEA/D-WS, the SOPs are generated by the weighted sum approach [i.e., (3)] using a set of uniformly distributed weight vectors [41].
2) In AUGMECON2, each SOP is constructed by adding the constraint on each interval's upper boundary [i.e., (30)]. But the parameter γ_3 in the AUGMECON2 is set manually, and we set it to $(10^{-6}/[f_2(\mathbf{x}^{B_1}) - f_2(\mathbf{x}^{B_2})])$ as recommended by [15].

For fair comparisons, we employ Gurobi as the SOP solver in all the three algorithms.

In this experiment, the parameters of TSMOA and the RoIs are set to be the same as in Section V and each BOAP's experimental data are randomly generated. The results of TSMOA on the three BOAPs are illustrated in Fig. 9(a) in the supplementary material. It can be seen that TSMOA-stage1 yields five supported nondominated objective vectors on each BOAP. They are widely and uniformly distributed along the Pareto front, thus can give the DM a rough but comprehensive holistic perception. Thereafter, the DM selects the RoIs as marked by the black boxes in Fig. 9 in the supplementary material. Within each RoI, TSMOA-stage2 achieves five uniformly distributed nondominated objective vectors. From them, the DM can choose the most suitable one as the final solution of the corresponding problem.

Since the computational effort is mainly consumed in the optimization of SOPs, we want to know how many SOPs the two noninteractive algorithms have to optimize to achieve such an approximation for the RoI of each BOAP. For MOEA/D-WS and AUGMECON2, we gradually increase the number of SOPs in each algorithm and repeatedly execute it on the BOAP, until it can obtain five objective vectors in the RoI or SOPs reaches a maximum of 500. Fig. 9(b) and (c) in the supplementary material shows the approximated objective vectors gained by MOEA/D-WS and AUGMECON2, respectively. It is worth noting that MOEA/D-WS fails to achieve enough nondominated objective vectors in the RoI of the BOAP with $n = 100$. It is because MOEA/D-WS can only find supported nondominated objective vectors, while that RoI contains only four supported nondominated objective vectors. In addition, Table V presents the number of SOPs optimized (denoted as SOPs) and the number of the nondominated objective vectors (represented by *NDs*) by each algorithm in the RoI of each BOAP. From these results, we can find that TSMOA needs to optimize fewer SOPs than the other two noninteractive algorithms. So, we can conclude that TSMOA can indeed save a certain amount of computational effort.

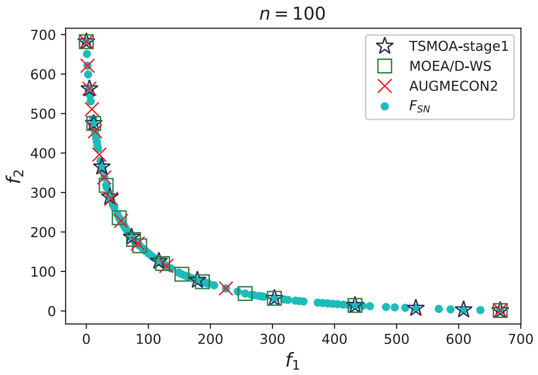


Fig. 5. Plot of the approximate objective vectors obtained by the three algorithms with SOPs = 15 on the BOAP with $n = 100$.

Since TSMOA consists of two parts: 1) TSMOA-stage1 and 2) TSMOA-stage2, we also want to validate their effectiveness separetly.

B. Comparison on the First Stage

As mentioned, the first stage of the algorithm aims at giving the DM a holistic perception of the Pareto front using fewer nondominated objective vectors. The objective vectors are preferably supported nondominated ones and distributed as uniformly as possible. In view of this, we adopt the following two indicators to evaluate the performance of the algorithms.

1) *SN$_{rate}$:* Let S indicate the approximate set obtained by an algorithm, SN$_{\text{rate}}$ is defined as

$$\text{SN}_{\text{rate}}(S) = \frac{\text{SNs}}{\text{SOPs}} \tag{36}$$

where SNs represents the number of supported efficient objective vectors in S, while SOPs indicates the number of SOPs optimized in the algorithm. Since the calculation is mainly spent on optimizing each SOP, SOPs can reflect the computational cost consumed by each algorithm. A larger SN$_{\text{rate}}$ value means the algorithm achieves more supported nondominated objective vectors with the same computational cost.

2) *SP [42]:* Let S indicate the approximate set, the spacing metric SP is calculated as

$$\text{SP}(S) = \sqrt{\frac{1}{|S|-1}\sum_{i=1}^{|S|}\left(\widehat{d} - d_i\right)^2} \tag{37}$$

where $d_i = \min_{\mathbf{x}^i, \mathbf{x}^j \in S,\ \mathbf{x}^i \neq \mathbf{x}^j} \|\mathbf{f}(\mathbf{x}^i) - \mathbf{f}(\mathbf{x}^j)\|_1$, and $\widehat{d}$ represents the mean of d_i for each $\mathbf{x}^i \in S$. The SP metric is widely used to assess the uniformity of the approximate set [43]. The smaller the SP value, the more uniform the distribution of the approximate objective vectors.

To be fair, we let each algorithm utilize the same procedure [see (22)] to obtain the boundary solutions of the Pareto front (i.e., $\mathbf{x}^{B_1}$ and $\mathbf{x}^{B_2}$), and let each algorithm generate the same number of SOPs for optimization. We perform the three

TABLE VI

MEAN SN_{rate} AND SP VALUES OBTAINED BY TSMOA-STAGE1, MOEA/D-WS, AND AUGMECON2 ON EACH BOAP WITH 20 INDEPENDENT DATA

Problem		SN_{rate}			SP		
n	SOP_s	TSMOA-stage1	MOEA/D-WS	AUGMECON2	TSMOA-stage1	MOEA/D-WS	AUGMECON2
100	10	**0.8000**	**0.8000** $\approx$	0.4750 $-$	**48.8070**	114.6764 $-$	176.7575 $-$
	15	**0.8667**	**0.8667** $\approx$	0.5167 $-$	**22.9688**	64.2385 $-$	115.1700 $-$
	20	**0.9000**	0.8800 $-$	0.5300 $-$	**19.7526**	40.1469 $-$	85.6478 $-$
150	10	**0.8000**	**0.8000** $\approx$	0.4950 $-$	**34.0720**	75.3797 $-$	161.5694 $-$
	15	**0.8667**	0.8600 $\approx$	0.5167 $-$	**24.2317**	37.9032 $-$	101.1427 $-$
	20	**0.9000**	0.8850 $-$	0.5250 $-$	**16.6149**	32.4018 $-$	73.8853 $-$
200	10	**0.8000**	**0.8000** $\approx$	0.5350 $-$	**28.8861**	49.2810 $-$	142.1319 $-$
	15	**0.8667**	**0.8667** $\approx$	0.5133 $-$	**24.8751**	33.2086 $-$	88.4139 $-$
	20	**0.9000**	0.8525 $-$	0.5300 $-$	**15.5667**	37.1102 $-$	64.1947 $-$

$+$,$-$ and $\approx$ denote that the performance of the corresponding algorithm is significantly better than, worse than, and similar to TSMOA-stage1 according to the Wilcoxon's rank sum test with $\alpha = 0.05$.

algorithms until SOPs reaches 10, 15, and 20 on each BOAP with 20 independent random experimental data.

Table VI shows the mean SN_{rate} and SP values obtained by the three algorithms along with the statistical tests. It clearly indicates that TSMOA-stage1 significantly outperforms MOEA/D-WS and AUGMECON2 in terms of the SP metric. As for the SN_{rate} metric, TSMOA-stage1 and MOEA/D-WS are remarkably better than AUGMECON2. TSMOA-stage1 achieves better mean SN_{rate} values than MOEA/D-WS when SOPs is large. But when SOPs is small, TSMOA-stage1 and MOEA/D-WS have similar performance concerning the SN_{rate} metric. When SOPs is small, the SOPs of MOEA/D-WS are widely dispersed, and the optimization of each SOP can lead to a different supported nondominated objective vector. This means that MOEA/D-WS can attain the best SN_{rate} value on each instance, just like TSMOA-stage1.

Fig. 5 shows the objective vectors attained by the three algorithms on a BOAP with $n = 100$. It can be seen that the approximated objective vectors gained by TSMOA-stage1 have a more uniform distribution than those generated by MOEA/D-WS and AUGMECON2.

C. Comparison on the Second Stage

In the second stage, the algorithm needs to approximate the RoI at a fine-grained level. However, the RoI is determined by the DM, and the performance of the algorithms may depend on its location. Taking these into consideration, we generate ten equal-sized successive intervals (denoted as $\{RGN_1, \ldots, RGN_{10}\}$) from $f_2^{\min}$ to $f_2^{\max}$, and the ith interval RGN_i for $i = 1, \ldots, 10$ can be expressed as

$$\left[f_2^{\min} + \frac{i-1}{10}\left(f_2^{\max} - f_2^{\min}\right),\ f_2^{\min} + \frac{i}{10}\left(f_2^{\max} - f_2^{\min}\right)\right]. \tag{38}$$

We take each interval as an RoI and compare algorithms on.

MOEA/D-WS is not compared here, as it cannot search in a specific subregion of the Pareto front. We include another preference-based MOIP algorithm, MOEA/D-NUMS [44], for comparison. MOEA/D-NUMS maps the uniformly distributed weight vectors to the RoI determined by a local reference vector $\mathbf{z}^r = (f_1(\mathbf{x}^{\text{up}}), f_2(\mathbf{x}^{\text{low}}))^\intercal$. The SOP in MOEA/D-NUMS is constructed with the augmented weighted Tchebycheff method, that is

$$g(\mathbf{x}|\mathbf{w}) = \max_{1\le i\le m} w_i\left(f_i(\mathbf{x}) - f_i^{\min}\right) + \rho\sum_{i=1}^{m}\left(f_i(\mathbf{x}) - f_i^{\min}\right) \tag{39}$$

where ρ is set to 10^{-6}, as suggested in [45].

For each interval of the three BOAPs with 20 independent data, the three algorithms are performed until SOPs reaches 5, 7, and 9, respectively. Their performance is evaluated with the hypervolume (HV) [46] metric.

1) *HV:* Let S be the approximate set obtained by an algorithm, and is a reference objective vector. The HV value calculates the volume between the approximate objective vectors and $\mathbf{z}^* = (1.2, \ldots, 1.2)^\intercal$, that is

$$\text{HV}\left(S, \mathbf{z}^*\right) = \text{vol}\left(\bigcup_{\mathbf{x}\in S}\left[\bar{f}_1(\mathbf{x}), z_1^*\right] \times \ldots \times \left[\bar{f}_m(\mathbf{x}), z_m^*\right]\right) \tag{40}$$

where $\bar{\mathbf{f}}(\mathbf{x}) = (\bar{f}_1(\mathbf{x}), \ldots, \bar{f}_m(\mathbf{x}))^\intercal$ is the normalized objective vector of $\mathbf{x}$ [see (24)]. vol($\cdot$) is the Lebesgue measure. In general, the higher the HV value, the better the performance of the algorithm.

Table VIII in the supplementary material shows the mean HV values achieved by the three algorithms with different SOPs settings in approximating each interval of the three BOAPs with 20 independent data. It can be observed that TSMOA-stage2 outperforms AUGMECON2 and MOEA/D-NUMS on almost all the test instances. Only on the intervals RNG_8 and RNG_{10}, TSMOA-stage2 is slightly worse than AUGMECON2, but the statistical significance tests indicate that the performance differences are not remarkable.

Fig. 6 plots the objective vectors obtained by the three algorithms with SOPs = 5 on the interval RNG_8 of a BOAP with $n = 150$. It can be found that the objective vectors obtained by TSMOA-stage2 have a more uniform distribution than those gained by AUGMECON2. MOEA/D-NUMS only attains one objective vector inside the RoI. Besides, AUGMECON2 yields two weakly nondominated objective vectors; whereas, the objective vectors achieved by TSMOA-stage2 are all nondominated ones.

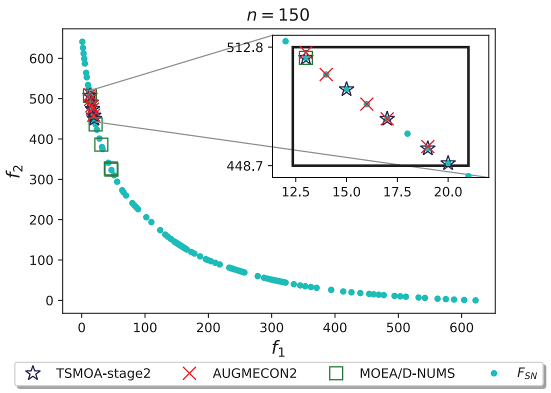


Fig. 6. Plot of the objective vectors obtained by the three algorithms with SOPs = 5 on the interval RNG_8 of a BOAP with $n = 150$.

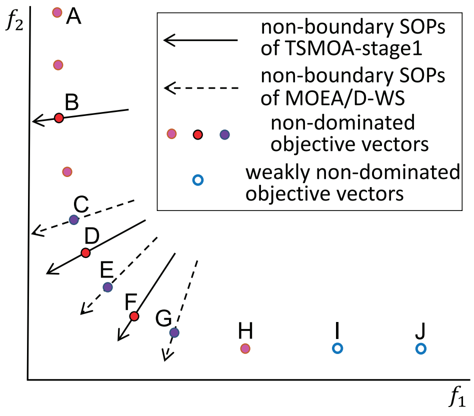

f2
f1
A
B
C
D
E
F
G
H
I
J
non-boundary SOPs of TSMOA-stage1
non-boundary SOPs of MOEA/D-WS
non-dominated objective vectors
weakly non-dominated objective vectors

Fig. 7. Example of SOPs constructed by TSMOA-stage1 and MOEA/D-WS.

VII. Discussion

A. TSMOA-Stage1 Versus MOEA/D-WS

The SOPs of TSMOA-stage1 are constructed differently from those of MOEA/D-WS. This is mainly due to the following two concerns.

1) As shown in Fig. 7, the optimization of the boundary SOPs of MOEA/D-WS may lead to weakly nondominated objective vectors (e.g., I and J) rather than the real nondominated ones (e.g., A and H). TSMOA-stage1 can overcome such a drawback, as it uses the lexicographic optimization method is employed to deal with the boundary SOPs. As stated in [44] and [45], the achievement of boundary nondominated objective vectors is very important for interactive or preference-based MOIP algorithms.

2) For the nonboundary SOPs, MOEA/D-WS constructs them using the uniformly distributed weight vectors. However, these SOPs' optimal objective vectors (e.g., C, E, and G) have a poor distribution when dealing with the problem whose Pareto front is extremely convex or of dissimilar objective function magnitudes. In contrast, TSMOA-stage1 uses the Aneja and Nair method to generate SOPs for optimization, thereby can yield nondominated objective vectors with better distribution (e.g., B, D, and F). The experimental study in Section VI-B has also demonstrated this viewpoint.

B. TSMOA-Stage2 Versus MOEA/D-NUMS

The differences between TSMOA-stage2 and MOEA/D-NUMS can be summarized as follows.

1) TSMOA-stage2 uses the upper and lower bounds of one objective to integrate the DM's preference for the RoI. In comparison, MOEA/D-NUMS incorporates the DM's preference through a local reference objective vector.

2) TSMOA-stage2 carefully achieves the boundary-objective vectors of the RoI, while MOEA/D-NUMS may obtain objective vectors outside of the RoI.

3) TSMOA-stage2 employs the ϵ-constraint method to construct SOPs. In contrast, MOEA/D-NUMS builds SOPs using the augmented weighted Tchebycheff method. The latter integrates the max function in the SOPs, thus making them difficult to solve via MPS strategies.

4) The control parameters in the SOPs of MOEA/D-NUMS are fixed values given by experience. Whereas, those parameters of TSMOA-stage2 are estimated from objective vectors achieved previously, making it less troubled by weakly nondominated objective vectors.

C. Parameter Sensitivity Analysis

$\text{SOP}_{\max}$ and $I_{\max}$ are the control parameters of TSMOA-stage1 and TSMOA-stage2, respectively. To investigate how they affect the performance of the two algorithms, we have tried to cover a wide range of values for each parameter. 10, 15, 20, 25, and 30 are considered for $\text{SOP}_{\max}$, while 3, 5, 7, 9, and 11 are considered for $I_{\max}$.

We have performed TSMOA-stage1 with the five $\text{SOP}_{\max}$ settings on the three BOAPs of Section VI. MOEA/D-WS and AUGMECON2 have also been performed on the three BOAPs for comparison. Fig. 10 in the supplementary material plots the mean SN_{rate} and SP values achieved by the three algorithms on each BOAP with 20 independent data. We can observe that TSMOA-stage1 obtains better SP results than the other two algorithms with any parameter setting. As for the SN_{rate} metric, TSMOA-stage1 always outperforms MOEA/D-WS and AUGMECON2 except when $\text{SOP}_{\max}$ is set to a very small value. As explained earlier, when $\text{SOP}_{\max}$ is very small, both TSMOA-stage1 and MOEA/D-WS can achieve the best mean SN_{rate} values.

As for the investigation of $I_{\max}$, we have taken the three BOAPs with RNG_2, RNG_5, and RNG_8 as the test instances. The three algorithms: 1) TSMOA-stage2; 2) AUGMECON2; and 3) MOEA/D-NUMS are tested with five different $I_{\max}$ settings (i.e., 3, 5, 7, 9, 11) on each of them. Their mean HV values over 20 independent data have been shown in Fig. 11 in the supplementary material. It clearly indicates that TSMOA-stage2 works better than the other two algorithms on almost all the cases. Only on RNG_8 of the BOAP with $n = 100$, TSMOA-stage2 is slightly worse than AUGMECON2 for $I_{\max} = 3$.

VIII. Conclusion

A real-world MP task has been investigated in this article. The MP task aims to maximize the order fulfillment rate and minimize the total cost by making each plant's production plan, deploying the items (raw materials or semi manufactured products) between the plants, and supplying the items (manufactured products) to customers over a 30-day horizon. The MP task involves a large number of decision variables, as it covers many items of multiple plants over 30 days. Moreover, its objectives have extremely different magnitudes, and the value ranges change drastically depending on the MP data. In addition, some particular constraints in the order fulfillment rate definition make the MP task hard to solve.

To address the MP task, we have proposed an interactive multiobjective optimization-based MP system. The system first formulates the submitted MP task as a MOIP problem whose two minimization objective functions are negative order fulfillment rate and total cost. Then, the system solves it using a TSMOA.

To alleviate the computational cost and the DM's cognitive burden as much as possible without compromising problem-solving, TSMOA adopts a two-stage interaction. TSMOA-stage1 uses the lexicographic optimization method and the Aneja and Nair method to construct the boundary SOPs and a few widely distributed nonboundary SOPs, respectively. These SOPs' optimal solutions can give the DM a holistic perception of the entire MP, thereby enabling the DM to determine a RoI in the objective space. Within the RoI, TSMOA-stage2 adopts a modified ϵ-constraint method to generate a uniformly distributed SOPs. These SOPs' optimal solutions can approximate the RoI at a fine-grained level. From them, the DM chooses the most suitable one as the final solution of the MP task.

Along with TSMOA, a specifically designed SOP solver has been adopted in the MP system to reduce the calculation workload significantly. In the strategy, we use two modified easier SOPs as the approximator of the original complex SOP and utilize Gurobi to solve them sequentially. We have demonstrated this strategy is efficient on real-world MP applications.

Although TSMOA is oriented to real-world applications, its development is based on the theoretical analysis. We first carefully summarized the main challenges (e.g., many decision variables and dissimilar and changeable magnitudes) of the MP task and reviewed the existing technologies accordingly. Based on these technologies, the algorithm with general applicability for similar problems was then proposed. The experimental studies on benchmark problems and the application of portfolio optimization (see Section V in the supplementary material) have demonstrated the effectiveness of TSMOA.

References

[1] F. R. Jacobs, W. L. Berry, D. C. Whybark, and T. E. Vollmann, *Manufacturing Planning and Control for Supply Chain Management*. New York, NY, USA: McGraw-Hill, 2011.

[2] D. Gong, Y. Han, and J. Sun, "A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems," *Knowl. Based Syst.*, vol. 148, pp. 115–130, May 2018.

[3] A. Cheraghalikhani, F. Khoshalhan, and H. Mokhtari, "Aggregate production planning: A literature review and future research directions," *Int. J. Ind. Eng. Comput.*, vol. 10, no. 2, pp. 309–330, 2019.

[4] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.

[5] C. P. Tautenhain, A. P. Barbosa-Povoa, and M. C. Nascimento, "A multi-objective matheuristic for designing and planning sustainable supply chains," *Comput. Ind. Eng.*, vol. 135, pp. 1203–1223, Sep. 2019.

[6] C. Purcaru, R.-E. Precup, D. Iercan, L.-O. Fedorovici, R.-C. David, and F. Dragan, "Optimal robot path planning using gravitational search algorithm," *Int. J. Artif. Intell.*, vol. 10, no. S13, pp. 1–20, 2013.

[7] G. Wen, S. S. Ge, C. P. Chen, F. Tu, and S. Wang, "Adaptive tracking control of surface vessel using optimized backstepping technique," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3420–3431, Sep. 2019.

[8] W. Liu, Y. Dong, F. Chiclana, F. J. Cabrerizo, and E. Herrera-Viedma, "Group decision-making based on heterogeneous preference relations with self-confidence," *Fuzzy Optim. Decis. Making*, vol. 16, no. 4, pp. 429–447, 2017.

[9] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, "Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1877–1890, Nov./Dec. 2017.

[10] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 474–486, Feb. 2015.

[11] Z. Wang, Y.-S. Ong, J. Sun, A. Gupta, and Q. Zhang, "A generator for multiobjective test problems with difficult-to-approximate Pareto front boundaries," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 556–571, Aug. 2019.

[12] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-style Tchebycheff approach for portfolio management," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2010, pp. 1–8.

[13] X. Li, M. Yuan, D. Chen, J. Yao, and J. Zeng, "A data-driven three-layer algorithm for split delivery vehicle routing problem with 3D container loading constraint," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2018, pp. 528–536.

[14] M. J. Alves and J. Clímaco, "A review of interactive methods for multiobjective integer and mixed-integer programming," *Eur. J. Oper. Res.*, vol. 180, no. 1, pp. 99–115, 2007.

[15] G. Mavrotas and K. Florios, "An improved version of the augmented ϵ-constraint method (AugmeCon2) for finding the exact Pareto set in multi-objective integer programming problems," *Appl. Math. Comput.*, vol. 219, no. 18, pp. 9652–9669, 2013.

[16] M. Visée, J. Teghem, M. Pirlot, and E. L. Ulungu, "Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem," *J. Global Optim.*, vol. 12, no. 2, pp. 139–155, 1998.

[17] A. Przybylski, X. Gandibleux, and M. Ehrgott, "A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives," *Discr. Optim.*, vol. 7, no. 3, pp. 149–165, 2010.

[18] S. C. Narula and V. Vassilev, "An interactive algorithm for solving multiple objective integer linear programming problems," *Eur. J. Oper. Res.*, vol. 79, no. 3, pp. 443–450, 1994.

[19] D. Gong, J. Sun, and X. Ji, "Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems," *Inf. Sci.*, vol. 233, pp. 141–161, Jun. 2013.

[20] M. Ehrgott and M. M. Wiecek, "Multiobjective programming," in *Multiple Criteria Decision Analysis: State of the Art Surveys*. Cham, Switzerland: Springer, 2005, pp. 667–708.

[21] J. Shi, Q. Zhang, and J. Sun, "PPLS/D: Parallel Pareto local search based on decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1060–1071, Mar. 2020.

[22] M. Laumanns, L. Thiele, and E. Zitzler, "An efficient, adaptive parameter variation scheme for metaheuristics based on the ϵ-constraint method," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 932–942, 2006.

[23] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, vol. 233. New York, NY, USA: Wiley, 1986.

[24] M. Ehrgott, *Multicriteria Optimization*, vol. 491. New York, NY, USA: Springer, 2005.

[25] Ö. Özpeynirci and M. Köksalan, "An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs," *Manag. Sci.*, vol. 56, no. 12, pp. 2302–2315, 2010.

[26] A. Przybylski, X. Gandibleux, and M. Ehrgott, "Two phase algorithms for the bi-objective assignment problem," *Eur. J. Oper. Res.*, vol. 185, no. 2, pp. 509–533, 2008.

[27] K.-H. Chang, "Multiobjective optimization and advanced topics," Dec. 2015, pp. 325–406.

[28] Y. P. Aneja and K. P. Nair, "Bicriteria transportation problem," *Manag. Sci.*, vol. 25, no. 1, pp. 73–78, 1979.

[29] J. L. Cohon, *Multiobjective Programming and Planning*. New York, NY, USA: Academic, 1978.

[30] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.

[31] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[32] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” *Acta Numerica*, vol. 4, pp. 1–51, 1995.

[33] L. Costa and P. Oliveira, “Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems,” *Comput. Chem. Eng.*, vol. 25, nos. 2–3, pp. 257–266, 2001.

[34] F. Glover and M. Laguna, “Tabu search,” in *Handbook of Combinatorial Optimization*. New York, NY, USA: Springer, 1998, pp. 2093–2229.

[35] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, vol. 379. Berlin, Germany: Springer, 2011.

[36] *Gurobi Optimizer Reference Manual 2015*, Gurobi Optim., Inc., Houston, TX, USA, 2014.

[37] E. L. Ulungu and J. Teghem, “Multi-objective combinatorial optimization problems: A survey,” *J. Multi Criteria Decis. Anal.*, vol. 3, no. 2, pp. 83–104, 1994.

[38] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Res. Logist. Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[39] K. Fukuda and T. Matsui, “Finding all the perfect matchings in bipartite graphs,” *Appl. Math. Lett.*, vol. 7, no. 1, pp. 15–18, 1994.

[40] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[41] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.

[42] J. R. Schott, “Fault tolerant design using single and multicriteria genetic algorithm optimization,” *Cell. Immunol.*, vol. 37, no. 1, pp. 1–13, 1995.

[43] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, “Consistencies and contradictions of performance metrics in multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.

[44] K. Li, M. Liao, K. Deb, G. Min, and X. Yao, “Does preference always help? A holistic study on preference-based evolutionary multi-objective optimisation using reference points,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 6, pp. 1078–1096, Dec. 2020.

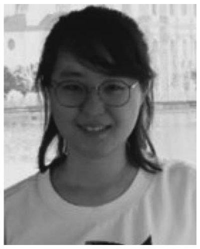
[45] K. Li, R. Chen, G. Min, and X. Yao, “Integration of preferences in decomposition multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3359–3370, Dec. 2018.

[46] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms—A comparative case study,” in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, 1998, pp. 292–301.

Zhenkun Wang (Member, IEEE) received the Ph.D. degree in circuits and systems from Xidian University, Xi’an, China, in 2016.

From 2017 to 2020, he was a Postdoctoral Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, and the Department of Computer Science, City University of Hong Kong, Hong Kong. He is currently an Assistant Professor with the School of System Design and Intelligent Manufacturing and the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include evolutionary computation, multiobjective optimization, machine learning, and their applications.

Dr. Wang is an Associate Editor of the *Swarm and Evolutionary Computation*.



Hui-Ling Zhen received the B.S. degree in numerical mathematics and the Ph.D. degree in applied mathematics from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011 and 2016, respectively.

She is currently working with Huawei Noah’s Ark Lab, Hong Kong. Her current research interests include large-scale optimization, constraint programming, as well as their applications in supply chain management and chip design.

Jingda Deng received the B.Sc. and M.Sc. degrees in mathematics from Xi’an Jiaotong University, Xi’an, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2020.

He is currently an Assistant Professor with the School of Mathematics and Statistics, Xi’an Jiaotong University. His main research interests include evolutionary computation, multiobjective optimization, machine learning, and their applications.

Qingfu Zhang (Fellow, IEEE) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi’an, China, in 1991 and 1994, respectively.

He is a Chair Professor of Computational Intelligence with the Department of Computer Science, City University of Hong Kong, Hong Kong. His main research interests include evolutionary computation, multiobjective optimization, neural networks, data analysis, and their applications.

Dr. Zhang has been a Web of Science Highly Cited Researcher in Computer Science for five consecutive years since 2016. He is an Associate Editor of the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Cybernetics.

Xijun Li received the M.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (HUAWEI-USTC Joint Ph.D. Program), Hefei, China.

He is currently a Senior Research Engineer with Huawei Noah’s Ark Lab, Hong Kong. He has published several papers on top peer-reviewed conferences (KDD, CIKM, and ICDCS). His research interests focus on learning to optimize combinatorial optimization problem and machine learning for computer systems.

Mingxuan Yuan received the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong.

He is a Principal Researcher with Huawei Noah’s Ark Lab, Hong Kong. He has led several projects in telecommunication data mining and supply chain optimization. His research interests include spatiotemporal data analytics and enterprise operation optimization models.

Jia Zeng (Senior Member, IEEE) received the B.Eng. degree from the Wuhan University of Technology, Wuhan, China, in 2002, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2007.

He is currently the AI Chief Scientist of Enterprise Intelligence (e.g., supply chain management) with Huawei Noah’s Ark Lab, Hong Kong, where he is also the Director of the AI Planning and Cooperation Department. His research interests are machine learning and big data applications.

Dr. Zeng is a member of CCF and ACM.