Balancing Common Treatment and Epidemic Control in Medical Supplies Procurement: Transform-and-Divide Evolutionary Optimization

***Abstract*—Balancing common disease treatment and epidemic control is a key objective of medical supplies procurement in hospitals during a pandemic such as COVID-19. This problem can be formulated as a bi-objective optimization problem for simultaneously optimizing the effects of common disease treatment and epidemic control. However, due to the large number of supplies, difficulties in evaluating the effects, and the strict budget constraint, it is difficult for existing evolutionary multiobjective algorithms to efficiently approximate the Pareto front of the problem. In this paper, we present an approach that first transforms the original high-dimensional, constrained multiobjective optimization problem to a low-dimensional, unconstrained multiobjective optimization problem, and then evaluates each solution to the transformed problem by solving a set of simple single-objective optimization subproblem, such that the problem can be efficiently solved by existing evolutionary multiobjective algorithms. We applied the transform-and-divide evolutionary optimization approach to six hospitals in Zhejiang Province, China, during the peak of COVID-19. Results showed that the proposed approach exhibits significantly better performance than that of directly solving the original problem. Our study has also shown that transform-and-divide evolutionary optimization based on problem-specific knowledge can be an efficient solution approach to many other complex problems and, therefore, enlarge the application field of EAs.**

***Index Terms*—Medical supplies procurement, epidemic control, multiobjective optimization, evolutionary algorithms, transform-and-divide.**

I. Introduction

In COVID-19, hospitals must procure medical supplies for epidemic control. However, the total budget of any hospital are limited: if a hospital procures too many supplies for epidemic control, it has to reduce supplies for common disease treatment, which would damage its medical services. Consequently, it is important for hospitals to balance between common disease treatment and epidemic control in medical supplies procurement in a pandemic. The problem of determining the purchase quantity of each supply can be formulated as a *bi-objective optimization problem* for simultaneously optimizing the effect of epidemic control and effect of common disease treatment. There are three main challenges to solving this problem. The first is to evaluate the effects of epidemic control and common disease treatment in a relatively accurate manner. The second is to meet the budget constraint, which is often strict in the pandemic. The third is to approximate the Pareto front of the problem in an efficient manner. For the first challenge, we develop a procedure to simulate the arrival and treatment of cases of infection and cases of common diseases according to a general principle of disease treatment and medical supplies usage. However, this also makes evaluating the objective functions too expensive. Moreover, a major hospital often involves tens of thousands of medical supplies, which makes the dimension of the solution space too high. The combination of these reasons makes the problem very difficult to solve.

Decomposition is a general approach to solving large complex problems that are beyond the reach of standard techniques. Decomposition in optimization appears in early work on large-scale linear programming problems from the 1960s [1]. Many problems with separable objective functions are trivial to solve by mathematical methods. For additively decomposed functions that are not able to optimize by standard genetic algorithms, Mühlenbein and Mahnig [2] proposed the factorized distribution algorithm that factors the distribution into conditional and marginal distributions based on function structures. Many combinatorial optimization problems can be solved by using efficient methods to solve subproblems and combining the results to the obtain solutions to the original problems. Zheng and Xue [3] utilized this characteristics to automatically derive efficient problem-solving algorithms, including evolutionary algorithms (EAs) that are mainly used for NP-hard problems. Unfortunately, the medical supplies procurement problem considered in this paper does not satisfy the basic conditions of decomposition, because it is quite common that one supply can be used in multiple diseases. and the treatment of a disease can involve many supplies.

A multiobjective optimization problem is much more difficult than its single-objective counterpart, and decomposition is also a basic strategy in multiobjective optimization. Zhang and Li [4] proposed a multiobjective evolutionary algorithms based on decomposition (MOEA/D), which decomposes a multiobjective optimization problem into a set of single-objective optimization subproblems using decomposition approaches such as weighted sum, weighted Tchebycheff, and penalty-based boundary interaction. However, for some problems, these decomposition approaches may not be suitable for balancing the diversity and convergence. Wang et al. [5] revolved this difficulty by imposing a constraint to an unconstrained subproblem, where the improvement region of each subproblem is determined by an adaptive control parameter. MOEA/D makes an assumption that two neighboring subproblems should have similar optimal solutions, but some combinatorial optimization problems do not satisfy this assumption. Mei et al. [6] proposed a decomposition-based memetic algorithm with neighborhood search for multiobjective capacitated arc routing problem, which combines decomposition-based and domination-based techniques for solution selection. Cai et al. [7] also combined domination-based sorting and decomposition in a multiobjective EA, which works with an internal population evolved using a decomposition-based strategy and an external archive maintained using a domination-based sorting. Jan and Zhang [8] introduced a penalty function to MOEA/D to deal with multiobjective constrained optimization problems. Konstantinidis and Yang [9] adapted MOEA/D to solve a K-connected deployment and power assignment problem by introducing a problem-specific repair heuristic that transforms an infeasible solution into a feasible one. Zhang et al. [10] extended MOEA/D for big optimization problems by embedding a gradient-based local search. Chen et al. [11] extended MOEA/D for constrained problems by assigning each subproblem with an upper bound vector based on the ϵ-constraint method. There have been many other extensions and applications of MOEA/D in recent years [12]. Unfortunately, we found that, although using decomposition-based strategies in MOEA/D and other similar algorithms can reduce the complexity to a certain degree, the performance of those algorithms is still far from satisfactory in solving the medical supplies procurement problem in practice.

In this study, we present a transform-and-divide approach to efficiently solve the problem. First, we transform the original problem of determining the purchase quantity of each supply to a new problem of distributing the budget to epidemic control and all common diseases. In our case studies, the dimension of the transformed problem is only one to two percent of that of the original problem. However, evaluating each solution to the transformed problem is itself a nontrivial optimization problem. Second, we divide the evaluation problem into a set of low-dimensional, single-objective optimization subproblem. We propose a hybrid evolutionary optimization approach, which employs a multiobjective EA to evolve a population of main solution to the transformed problem and uses a tabu search algorithm to solve the divided subproblems. During the peak of COVID-19, we applied the proposed approach to six hospitals in Zhejiang Province, China. Results demonstrated that the transform-and-divide evolutionary optimization approach exhibits significantly better performance than that of directly using multiobjective EAs to solve the original problem. The main contributions of this paper are twofold:

- We propose a transform-and-divide evolutionary optimization approach to medical supplies procurement and demonstrate its practicability and efficiency during COVID-19.
- We show that using problem-specific knowledge to transform and divide a complex optimization problem can lead to competitive EAs for the problem. This approach can be extend to many other problems and enlarge the application field of EAs.

The remainder of this paper is organized as follows. Section II presents the medical supplies procurement problem. Section III simply describes how to directly use basic multiobjective EAs to solve the original problem. Section IV proposes the transform-and-divide evolutionary optimization approach. section V presents the computational results. Section VI concludes with a discussion.

II. Problem Description

A. Supplies for Epidemic Control and Common Treatment

We consider a medical supplies procurement problem formulated as follows. In a pandemic, a hospital plans to procure medical supplies, including a set $\mathbf{S} = \{S_1, S_2, \dots, S_n\}$ of n supplies for epidemic control, and a set $\mathbf{S}' = \{S_{n+1}, S_{n+2}, \dots, S_{n+n'}\}$ of n' supplies for normal disease treatment. For each supply S_k, the current inventory is a_k, the unit price is c_k, and the unit volume is v_i. The problem is to determine the purchase quantity x_k of each supply ($1 \le k \le n+n'$), such that the effects of epidemic control and normal disease treatment are simultaneously optimized.

The supplies for epidemic control can be divided into two classes. The first class consists of supplies such as latex gloves and normal saline that must be used in the treatment of a suspected case of infection; we use Ψ_0 to denote the set of these supplies, and use $q_{0,k}$ to denote the quantity of each $S_k \in \Psi_0$ required to treat a case. The second class consists of supplies that are alternative in some treatment items. Table I presents six treatment items and their alternative supplies used for COVID-19 control in this study. The seven sets of supplies are denoted by $\Psi_1, \Psi_2, \dots, \Psi_6$, respectively, and the quantity of each alternative $S_k \in \Psi_j$ required to treat a case is denoted by $q_{j,k}$. Different alternatives have different treatment effects, and the treatment effect of using each alternative $S_k \in \Psi_j$ is estimated as $e_{j,k}$. For example, the effects of peroxide, impermeable gown, and normal gown in "body protection" item are estimated as 1, 0.9 and 0.7, respectively. If we choose the $S_{k_j} \in \Psi_j$ for the j-th treatment item ($1 \le j \le 6$), the corresponding epidemic control effect on the case is empirically estimated as:

$$E(k_1, ...k_6) = (0.4e_{1,k_1} + 0.6e_{2,k_2})e_{3,k_3}(0.2e_{4,k_4} + 0.8e_{5,k_5})e_{6,k_6} \tag{1}$$

The hospital is capable of treating a set $\mathbf{D} = \{D_1, D_2, \dots, D_m\}$ of m diseases. Similarly, for each disease D_i, the set of supplies that must be used is denoted by $\Psi_{i,0}$, and the set of supplies that are alternative in J_i treatment items are denoted by $\Psi_{i,1}, \Psi_{i,2}, \dots, \Psi_{i,J_i}$, respectively. The quantity of each $S_k \in \Psi_{i,0}$ required to treat a case is $q_{i,0,k}$, and the quantity of each alternative $S_k \in \Psi_{i,j}$ required to treat a case is $q_{i,j,k}$. Different alternatives have different treatment effects, and the treatment effect of using $S_k \in \Psi_{i,j}$ is estimated as $e_{i,j,k}$. If we choose the $S_{k_j} \in \Psi_{i,j}$ for the j-th treatment item ($1 \le j \le J_i$), the corresponding treatment effect on the case is empirically estimated by a therapeutic effect function $E_i(k_1, k_2, ..., k_{J_i})$. Like Eq. (1), the typical expression of E_i is a weighted sum or product of $e_{i,j,k}$ [13].

B. Number of Cases

Let T be the procurement decision cycle. In our case study, the hospital procures medical supplies every 15 days. The supply quantities are determined based on the estimation of the number of hospital visits in the next decision cycle. For the number of cases of each disease D_i, we estimate three values: the expected value r_i, lower limit (optimistic value) $\underline{r_i}$, and upper limit (pessimistic value) $\overline{r_i}$. The values can be

TABLE I
ALTERNATIVE SUPPLIES FOR EPIDEMIC CONTROL IN THIS STUDY.

Items	Body protection	Face protection	Detection	Oxygen therapy	Antivirus	Disinfectant
Supplies	protective clothing	face shield	nucleic acid kit	high-flow nasal cannula	α-interferon	peroxide
	impermeable gown	N95 mask+goggle	antibody kit	nasal cannula	lopinavir	chlorine-containing
	normal gown	surgical mask+goggle		oxygen mask	chloroquine phosphate	alcohols
					arbidol	

obtained based on historical morbidity data and environmental influence factors [14]–[17].

The number of suspected cases of epidemic infection is estimated based on the number of hospital visits of different diseases. For each disease D_i, we estimate a probability p_i that a patient of D_i is a suspected case of COVID-19. In general, a disease having more similar symptoms with the epidemic has a higher p_i. For example, an acute respiratory infectious disease has a high p_i. For a disease (such as fracture) that is unrelated with the epidemic, we set p_i to the current incidence p_e of infection (including suspected infection) in the local region. We also estimate an average number r'_i of accompanying persons of a patient of D_i; in general, a critical disease has a large r'_i. The probability that an accompanying person of a patient of D_i is a suspected case of COVID-19 is p'_i, which is set to $0.5p_i$ if D_i has similar symptoms with the epidemic and p_e otherwise. The total number of suspected cases of infection in the next decision cycle is estimated as follows (we use $\overline{r_i}$ as we take a serious or pessimistic view of epidemic control):

$$r=\sum_{i=1}^{m}(p_i+p'_i r'_i)\overline{r_i} \tag{2}$$

C. Objective Function Evaluation

A solution to the medical supplies procurement problem can be represented by a $(n+n')$-dimensional vector $\mathbf{x}=\{x_1,\dots,x_n,x_{n+1},\dots,x_{n+n'}\}$. The fitness of $\mathbf{x}$ is evaluated by two objective functions: (1) the epidemic control effect $\Upsilon(\mathbf{x})$, which is the sum of treatment effects of all suspected cases of infection; (2) the common disease treatment effect $\Upsilon'(\mathbf{x})$, which is the weighted sum of treatment effects of all common cases, where the weight of each disease D_i is w_i. It is assumed that the arrival time of cases follows a uniform distribution. That is, for each disease D_i, as the expected number of cases in a decision cycle of 15 days is r_i, then there is a case arriving every $(15\times h_w)/r_i$ hours, while h_w is the daily working hours (24 for emergency diseases and 8 for emergency diseases in our study). Moreover, there is one suspected case of infection in every $1/(p_i+p'_i r'_i)$ cases of D_i.

A general principle of disease treatment is "focusing on the current patient", i.e., whenever a new case arrives, the physician always chooses the most effective supply from the available alternatives, as he does not know how many cases would come later. Based on this first-come-first-served discipline, we sort supplies in Ψ_j $(1\le j\le 6)$ or $\Psi_{i,j}$ $(1\le i\le m; 1\le j\le J_i)$ in nonincreasing order of $e_{j,k}$ or $e_{i,j,k}$, and simulate the arrival and treatment of all cases according to the procedure shown in Algorithm 1 to calculate the values of $\Upsilon(\mathbf{x})$ and $\Upsilon'(\mathbf{x})$. In Algorithm 1, the boolean variable *tr* denotes whether the remaining supplies are capable of treating a suspected case of epidemic infection, and tr_i denotes whether the remaining supplies are capable of treating a case of D_i.

Algorithm 1: Procedure for evaluating the effects of epidemic control and disease treatment for the original problem.

Initialize $\Upsilon=0$, $tr=true$;
for $i=1$ *to* m **do** initialize $\Upsilon_i=0$, $tr_i=true$;
for $k=1$ *to* $n+n'$ **do** $a_k\leftarrow a_k+x_k$;
Start timing simulation;
while *the decision cycle is not complete* **do**
  if *a new case of D_i arrives* and $tr_i=true$ **then**
    foreach $S_k\in\Psi_{i,0}$ **do**
      $a_k\leftarrow a_k-q_{i,0,k}$;
      if $a_k<q_{i,0,k}$ **then** $tr_i\leftarrow false$;
    for $j=1$ *to* J_i **do**
      let S_k be the first supply in $\Psi_{i,j}$;
      while $a_k<q_{i,j,k}$ **do**
        Remove S_k from $\Psi_{i,j}$;
        if $\Psi_{i,j}=\emptyset$ **then** $tr_i\leftarrow false$;
      let $k_j=k$;
      $a_k\leftarrow a_k-q_{i,j,k}$;
    $\Upsilon_i\leftarrow\Upsilon_i+E_i(k_1,\dots,k_{J_i})$;
  if *the case is a suspected infected case* and $tr=true$ **then**
    foreach $S_k\in\Psi_0$ **do**
      $a_k\leftarrow a_k-q_{0,k}$;
      if $a_k<q_{0,k}$ **then** $tr\leftarrow false$;
    for $j=1$ *to* 6 **do**
      let S_k be the first supply in Ψ_j;
      while $a_k<q_{j,k}$ **do**
        Remove S_k from Ψ_j;
        if $\Psi_j=\emptyset$ **then** $tr\leftarrow false$;
      $a_k\leftarrow a_k-q_{j,k}$;
      let $k_j=k$;
    $\Upsilon\leftarrow\Upsilon+E(k_1,\dots,k_6)$;
return $\Upsilon(\mathbf{x})=\Upsilon$ and $\Upsilon'(\mathbf{x})=\sum_{i=1}^{m}w_i\Upsilon_i$.

D. Constraints

A procurement solution $\mathbf{x}$ must satisfy problem constraints. First, the total procurement cost cannot exceed the budget C:

$$\sum_{k=1}^{n+n'}c_k x_k\le C \tag{3}$$

The hospital should perform its normal functions. In this study, it is required that the hospital is able to treat $\underline{r_i}$ (the lower limit of the number) cases of each disease D_i. These constraints can be tested by simulating the arrival and treatment of the lower numbers of cases in Algorithm 1: if any new case cannot be treated, i.e., whenever the condition tr_i=*false* (Line 6 of Algorithm 1) is triggered, the constraint is violated.

It is also required that the hospital is able to treat r suspected cases of epidemic infection. Whenever the condition *tr*=*false* (Line 18) in Algorithm 1 is triggered, the constraint is violated.

III. Basic Evolutionary Optimization Methods

For the above $(n+n')$-dimensional, constrained bi-objective optimization problem, we can use evolutionary constrained multiobjective algorithms to search for the Pareto optimal solutions. The search range of each dimension k is $[\underline{x_k}, \overline{x_k}]$. The lower limit $\underline{x_k}$ is set to the total quantity of S_k required in non-alternative treatment items for all cases:

$$\underline{x_k} = \begin{cases} \max(0, rq_{0,k} - a_k), & 1 \le k \le n \\ \max(0, \sum_{i=1}^{m} r_i q_{i,0,k} - a_k), & n+1 \le k \le n+n' \end{cases} \tag{4}$$

The upper limit $\overline{x_k}$ can be set to the total required quantity of S_k under the assumption that S_k is always chosen whenever S_k is an alternative. That is, if $1 \le k \le n$, we set

$$\overline{x_k} = \Big(\sum\nolimits_{j' \in \{j | 0 \le j \le 6 \wedge S_k \in \Phi_j\}} rq_{j',k}\Big) - a_k \tag{5}$$

Otherwise, we set

$$\overline{x_k} = \Big(\sum\nolimits_{(i',j') \in \{(i,j) | 1 \le i \le m \wedge 0 \le j \le J_i \wedge S_k \in \Phi_{i,j}\}} r_{i'} q_{i',j',k}\Big) - a_k \tag{6}$$

We adopt the following four well-known evolutionary constrained multiobjective algorithms to solve the considered problem:

- The nondominated sorting genetic algorithm II (NSGA-II) with the constrained-domination principle [18].
- The multiobjective evolutionary algorithm based on decomposition (MOEA/D) [4] with a penalty function for constrain handling [8].
- The differential evolution with self-adaptation and local search for constrained multiobjective optimization (DECMOSA) [19], which combines the constrained-domination principle and penalty function for constrain handling.
- The constrained multiobjective evolutionary algorithm (CMOEA) based on an adaptive penalty function and a distance measure [20].

The last three algorithms employ penalty functions for constrain handling. Violation of constraint (3) is calculated as $\max(0, \sum_{k=1}^{n+n'} c_k x_k - C)$. For constraints that all suspected cases of infection and the lower number of cases of each common disease must be treated, we set the violation of each constraint equal to the budget C, i.e., the violation is C times the number of false tr_i and *tr* in Algorithm 1.

Nevertheless, the performance of all the above algorithms is not satisfying, mainly because the dimension $(n+n')$ is very high (approximately 20,000∼50,000 in a major hospital in our case study) and the evaluation of a solution using Algorithm 1 is computationally expensive.

Fig. 1. The existence probability.

IV. A New Transform-and-Divide Evolutionary Optimization Method

In this section, we propose a new transform-and-divide approach to efficiently solve the problem. First, we transform the original high-dimensional, constrained bi-objective optimization problem to a low-dimensional, unconstrained bi-objective optimization problem, which can be solved using evolutionary (unconstrained) multiobjective algorithms. The evaluation of each solution to the transformed problem can be divided into a set of low-dimensional, single-objective optimization subproblems, which can be solved using a tabu search algorithm.

A. Problem Transformation

We transform the original problem of determining the purchase quantity of each supply to a problem of determining the purchase budget for epidemic control and the purchase budget for each disease. First of all, we calculate the cost for purchasing the supplies that must be used in the non-alternative treatment items and, therefore, obtain the remaining budget as:

$$C' = C - \sum_{k=1}^{n+n'} c_k \underline{x_k} \tag{7}$$

Consequently, the transformed problem is to distribute C' to $m+1$ components, denoted by $\{y_0, y_1, \ldots, y_m\}$, where y_0 is the budget for purchasing alternative supplies for epidemic control, and y_i is the budget for purchasing alternative supplies for treating disease D_i ($1 \le i \le m$). The dimension of the transformed problem is m (approximately 300∼600 in a major hospital in our case study), which is significantly smaller than the dimension $n+n'$ of the original problem.

Moreover, the search range of each dimension of the transformed problem is also much smaller than that of the original problem. For epidemic control, the lower limit $\underline{y_0}$ of budget y_0 can be obtained using the following steps:

1) Use supplies in storage to treat as many suspected cases of infection as possible;
2) If there is no remaining case, set $\underline{y_0}=0$;
3) Else, for each remaining case, always purchase the cheapest supply among the alternatives, and set $\underline{y_0}$ to the total purchase cost.

And the upper limit $\overline{y_0}$ of budget y_0 can be obtained using the following steps:

1) Treat each suspected case in the most effective way, i.e, always select the supply with the maximum treatment effect $e_{j,k}$ among the alternatives, and calculate the total required quantity of each supply;
2) Calculate the purchase quantity of each supply, and set $\overline{y_0}$ to the total purchase cost.

Therefore, the search range of y_0 is limited to $[\underline{y_0}, \overline{y_0}]$. We can obtain the search range $[\underline{y_i}, \overline{y_i}]$ for each y_i in a similar manner. In our case study, the average value of $(\overline{y_i} - \underline{y_i})$ is approximately 75 (in unit of 100 RMB), while the average value of $(\overline{x_k}-\underline{x_k})$ is approximately 1100 (in minimum order quantity); consequently, by transformation, the number of all possible solutions is reduced from approximately 1100^{35000} to 75^{450}.

B. Problem Division

The solution space of the transformed problem is significantly smaller than that of the original problem. But how to evaluate a solution $\mathbf{y} = \{y_0, y_1, \dots, y_m\}$ to the transformed problem? The task can be divided into $m+1$ optimization subproblems. The first subproblem is to determine the purchase quantities under the budget y_0 so as to maximize the epidemic control effect. Each of the remaining m subproblems is to determine the purchase quantities under the budget y_i so as to maximize the overall treatment effect of disease D_i $(1 \le i \le m)$.

However, the division raises a difficulty in allocating supplies in storage to different diseases. We overcome this difficulty by employing a procedure similar to Algorithm 1 to simulate the arrival and treatment of all cases. But the procedure has two differences from Algorithm 1:

- Initially, we only consider supplies in storage, i.e., Line 3 of Algorithm 1 is not executed.
- If there is no supply in storage that can be used for a treatment item (i.e., the condition in Line 14 or Line 26 is satisfied), we temporarily purchase "in advance" the cheapest alternative supply for the item.

The procedure also produces the "cheapest" solution to each subproblem, which can be evolved to an optimal or near-optimal solution, as described in the next subsection.

C. Hybrid Evolutionary Optimization

The proposed method employs an evolutionary multiobjective algorithm to evolve a population of main solutions to the transformed problem, and employs a tabu search algorithm to solve the subproblems for evaluating each main solution.

For the first subproblem, each solution $\mathbf{z}$ can be represented by six vectors as follows (the vector lengths do not need to be the same):

$$\{z_{1,1}, z_{1,2}, \dots, z_{1,|\Psi_1|}\}$$
$$\{z_{2,1}, z_{2,2}, \dots, z_{2,|\Psi_2|}\}$$
$$\vdots$$
$$\{z_{6,1}, z_{6,2}, \dots, z_{6,|\Psi_6|}\}$$

where $z_{j,k}$ denotes the number of cases that use the k-th alternative supply for the j-th treatment item, and each vector satisfies $\left(\sum_{k=1}^{|\Psi_j|} z_{j,k}\right) = r$.

The procedure described in Sec. IV-B produces the cheapest solution to the subproblem, denoted by $\mathbf{z}^\dagger$. First, we continually use the following steps to improve $\mathbf{z}^\dagger$ by replacing an alternative supply to a more effective alternative for a randomly selected case until $\mathbf{z}^\dagger$ cannot be further improved:

1) Randomly selecting two components $z_{j,k}$ and $z_{j,k'}$ in a vector satisfying $z_{j,k'} > 0$;
2) Set $z_{j,k'} = z_{j,k'} - 1$ and $z_{j,k} = z_{j,k} + 1$ if doing so would not violate the budget constraint.

Starting from the improved $\mathbf{z}^\dagger$, the tabu search algorithm continually uses the following steps to search around and improve $\mathbf{z}^\dagger$ until the stopping condition is satisfied:

1) Generate k_N neighboring solutions of the current $\mathbf{z}^\dagger$ as the current solution, each being obtained by randomly selecting two components $z_{j,k}$ and $z_{j',k'}$ satisfying $k < |\Psi_j|$, $k' < |\Psi_{j'}|$, $z_{j,k} > 0$, and $z_{j',k'+1} > 0$, and setting $z_{j,k} = z_{j,k} - 1$, $z_{j,k+1} = z_{j,k+1} + 1$, $z_{j',k'} = z_{j',k'} + 1$, and $z_{j',k'+1} = z_{j',k'+1} - 1$, if doing so would not violate the budget constraint;
2) Select the best neighbor that is not tabued or is better than the current $\mathbf{z}^\dagger$, make $\mathbf{z}^\dagger$ move to this neighbor, and add this move to the tabu list.

The remaining m subproblems can be solved by tabu search in a similar way. As demonstrated by the experiments, the tabu search algorithm can quickly obtain optimal solutions for most subproblem instances, given that the dimensions of the subproblems are relatively small. For example, as we can observe from Table I, the dimension of the first subproblem is 18 (note that the last dimension of each vector can be determined by other dimensions of the vector, and the actual dimension in the solution space is only 12). Therefore, the tabu search algorithm is very suitable for the subproblems, as it will be invoked many times to evaluate main solutions.

For the main transformed problem, we adopt the following evolutionary multiobjective algorithms to evolve main solutions and invoke the tabu search algorithm:

- NSGA-II [18].
- MOEA/D [4].
- An extension of differential evolution for multiobjective optimization (GDE3) [21].
- A multiobjective particle swarm optimization (MOPSO) algorithm [22] which extends comprehensive learning [23] for multiobjective optimization.

V. Computational Results

A. Problem Instances

We use the proposed method for medical supplies procurement in Zhejiang Hospital of Traditional Chinese Medicine (ZJHTCM) from 15 Feb to 15 Apr, 2020, the peak of COVID-19 in Zhejiang Province, China. Since 15 Mar, we also extend the application to other five hospitals (denoted by H1–H5). Therefore, there are total 14 real-world instances of the medical supplies procurement problem. Table II summarizes the main characteristics of the instances, where $\sum_i r_i$ denotes the total expected number of cases of all common diseases, $\overline{J}$ denotes the average treatment items per disease, $\overline{|\Upsilon|}$ denotes the average number of alternatives per treatment item, and the budget C is in RMB. The instances are solved on a workstation with an i7-6500 2.5GH CPU, 8GB DDR4 RAM, and an NVIDIA Quadro M500M card.

B. Performance for Solving the Subproblem

Before testing the algorithms for solving the main problem, we first test the performance of the tabu search algorithm

TABLE II
SUMMARY OF THE REAL-WORLD INSTANCES OF THE MEDICAL SUPPLIES PROCUREMENT PROBLEM.

Hospital	Period	m	$n+n'$	$\sum_i r_i$	r	$\overline{J}$	$\overline{\lvert\Upsilon\rvert}$	C
ZJHTCM	2nd half Feb	476	32,535	71,196	64	5.84	7.27	3,516,000
	1st half Mar	476	32,416	76,580	38	5.84	7.27	3,378,000
	2nd half Mar	479	32,628	78,331	34	5.86	7.29	3,022,000
	1st half Apr	479	32,628	90,459	36	5.86	7.32	3,698,000
H1	2nd half Mar	162	17,522	8,208	4	7.46	5.41	521,000
	1st half Apr	162	17,510	13,640	3	7.46	5.41	830,000
H2	2nd half Mar	193	15,666	17,353	24	8.06	5.25	785,000
	1st half Apr	193	15,681	19,309	14	8.06	5.25	902,500
H3	2nd half Mar	328	24,469	32,052	14	7.84	5.87	1,682,000
	1st half Apr	328	24,469	42,667	17	7.84	5.97	2,127,000
H4	2nd half Mar	393	27,600	35,733	50	6.90	6.13	2,415,500
	1st half Apr	399	27,215	38,452	28	6.87	6.14	2,607,200
H5	2nd half Mar	573	35,906	60,900	27	6.66	5.36	3,920,000
	1st half Apr	573	34,902	75,393	30	6.66	5.48	4,818,000

for subproblems. From the above real-world main problem instances, we select 16 subproblem instances, the dimensions D of which range from 12 to 72. For the algorithm, we set the neighborhood size k_N to $2D$, tabu length to 12, and the maximum number of iterations to $50D$. On each instance, we run the algorithm 50 times to test whether and how long can obtain the exact optimal solution (validated by an exact branch-and-bound algorithm [24]).

Fig. 2 presents the convergency curves (averaged over the 50 runs) of the tabu search algorithm on the subproblem instances. The algorithm reaches the optima within 100 iterations (10 ms in our computing environment) when the problem dimension is smaller than 24, within 200 iterations (30 ms) when the problem dimension is smaller than 40, and within 400 iterations (120 ms) on all instances. In our case study, the average dimension of subproblem instances is approximately 37, which can be solved using approximately 160 iterations (25 ms); the dimension of the largest instance is 72, which can be solved using 369 iterations (116 ms). Using multithreading and GPU acceleration, the average CPU time for evaluating a main solution to a problem of 400 diseases is approximately 600 ms.

C. Performance for Solving the Original and Transformed Problems

For each main problem instance, we use four evolutionary constrained multiobjective algorithms, including NSGA-II with constraint handling (denoted by NSGA-II-C) [18], MOEA/D with constraint handling (denoted by MOEA/D-C) [8], DECMOSA [19], and CMOEA [20], to solve the original problem, and four evolutionary multiobjective algorithms, including NSGA-II [18], MOEA/D [4], GDE3 [21], and MOPSO [22], all combined with tabu search, to solve the transformed problem. The control parameters of all algorithms are tuned on the whole set of instances. For a fair comparison, all the algorithms use the same stopping criterion that the CPU time does not exceed three hours, which is also applied in our practices. On each instance, each algorithm is run for 30 times.

Fig. 3 compares the hyperarea (the area under the Pareto-approximated front in objective space, also known as the hypervolume) [25], [26] obtained by each algorithm on each main problem instances. It is clear that the last four algorithms using the transform-and-divide method exhibit significant performance advantages over the first four algorithms. On most instances, the median hyperareas of the four transform-and-divide EAs are approximately two to three times of the basic EAs. In general, performance of the transform-and-divide EAs is mainly affected by m (the number of diseases) and C (the total budget), while performance of the basic EAs is mainly affected by $n+n'$ (the number of supplies). This is why all algorithms obtain relatively high hyperareas on the two instances of H1, whose m, $n+n'$, and C are the smallest among the instances. Nevertheless, the performance advantages of the transform-and-divide EAs over the basic EAs are very significant on the two instance. On the last two instances of H5, the values of these parameters are the largest, and the performance advantages of the transform-and-divide EAs over the basic EAs are not so significant. On most other instances, the maximum hyperareas of the basic EAs are often smaller than the minimum hyperareas of the transform-and-divide EAs. This demonstrates that the proposed transform-and-divide method can significantly reduce the difficulty of solving the complex ordinal problem.

We also make pairwise comparison between NSGA-II-C and NSGA-II as well as between MOEA/D-C and MOEA/D in terms of the coverage (*Cov*) metric [25] of the resulting solution set obtained by the algorithm, i.e., $Cov(X, X')$ is fraction of solution set X that are strictly dominated by at least one solution of X'. As shown in Table III, on the first 11 instances, none of solutions obtained by the transform-and-divide EA is dominated by at least a solution obtained by the corresponding basic EA; on the last three instances, only a very small fraction (1%∼2%) of solutions obtained by the transform-and-divide EA are dominated by the best solutions obtained by the corresponding basic EA. On the contrary, in most cases, 100% solutions obtained by the basic EA are dominated by at least a solution obtained by its transform-and-divide counterpart; in the remaining cases, almost all (over 90%) solutions obtained by the basic EA are dominated by at least a solution obtained by its transform-and-divide counterpart. Consequently, decision-makers always prefer to adopt solutions produced by transform-and-divide EAs, while solutions obtained by the basic EAs can hardly provide reference.

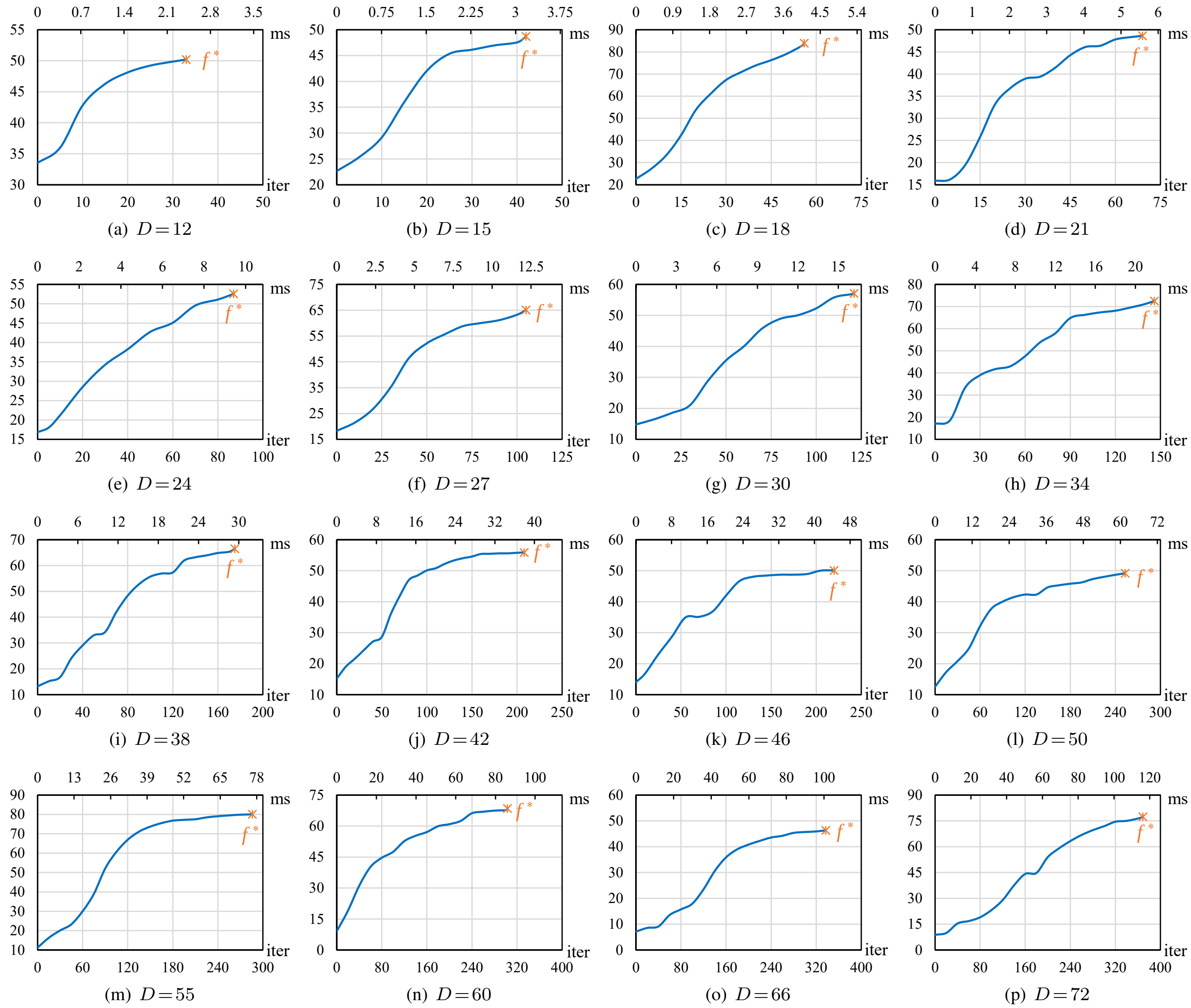


Fig. 2. Convergency curves of the tabu search algorithm on subproblem instances. The bottom horizontal axis is the number of iterations, the top horizontal axis is the CPU time (in milliseconds), the vertical axis is the objective function value, and f^* is the exact optimal objective function value.

VI. Conclusion

This paper presents a transform-and-divide evolutionary optimization approach to medical supplies procurement under the background of COVID-19. Our approach first transforms the original high-dimensional, constrained multiobjective optimization problem to a low-dimensional, unconstrained multiobjective optimization problem, and then evaluates each solution to the transformed problem by solving a set of simple single-objective optimization subproblem, such that the problem can be efficiently solved by existing evolutionary multiobjective algorithms. We applied the transform-and-divide evolutionary optimization approach to six hospitals in Zhejiang Province, China, during the peak of COVID-19. Results showed that our approach exhibits significantly better performance than that of directly solving the original problem. The proposed transform-and-divide evolutionary optimization based on problem-specific knowledge can be an efficient solution approach to many other complex problems and, therefore, enlarge the application field of EAs.

References

[1] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Oper. Res.*, vol. 8, pp. 101–111, 1960.

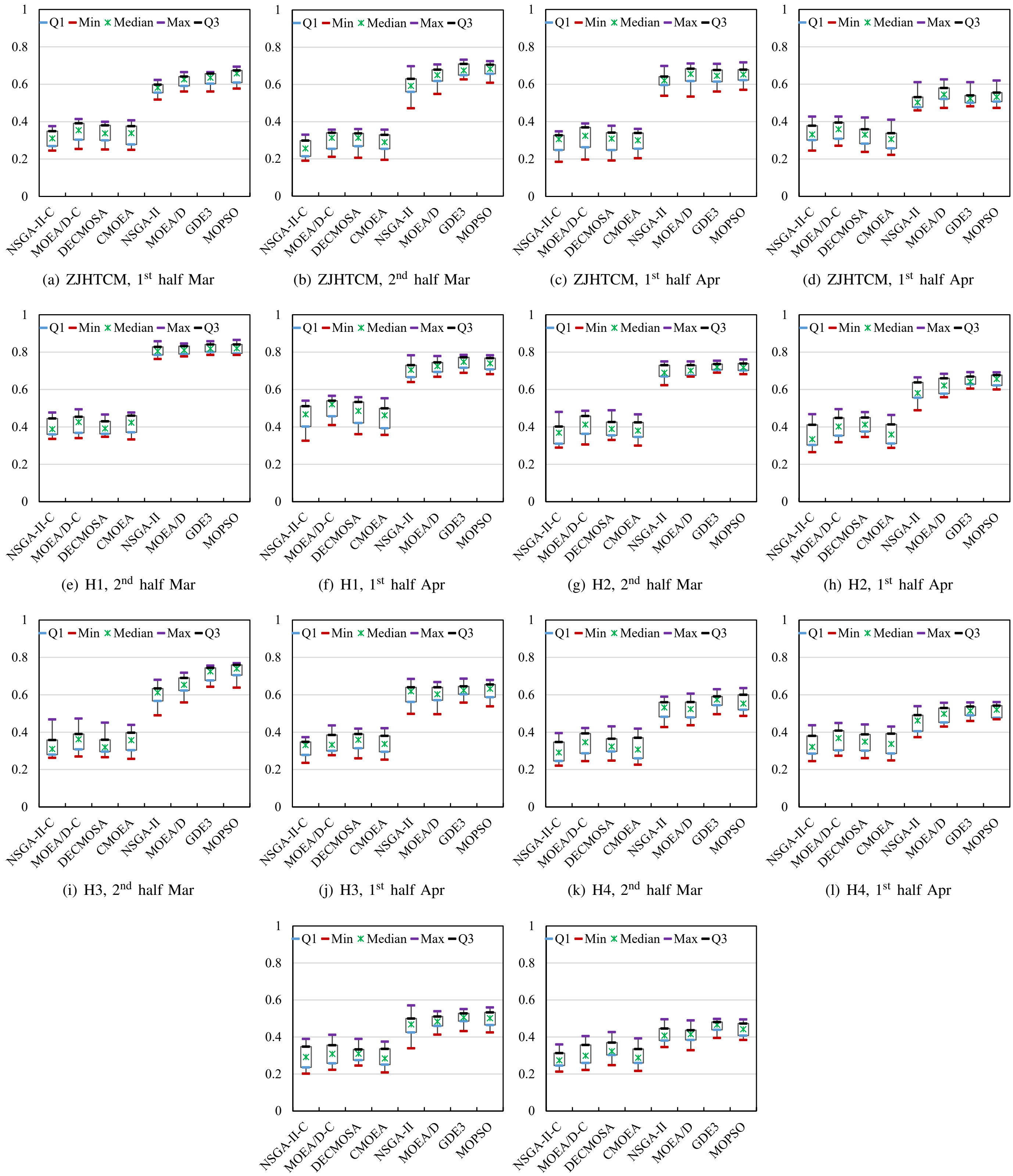
[2] H. Mühlenbein and T. Mahnig, "Fda - a scalable evolutionary algorithm for the optimization of additively decomposed functions," *Evol. Comput.*, vol. 7, no. 4, pp. 353–376, 1999.

[3] Y. Zheng and J. Xue, "A problem reduction based approach to discrete optimization algorithm design," *Computing*, vol. 88, no. 1-2, pp. 31–54, 2010.

[4] 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, 2007.

[5] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, 2016.

[6] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151–165, 2011.



(a) ZJHTCM, 1st half Mar (b) ZJHTCM, 2nd half Mar (c) ZJHTCM, 1st half Apr (d) ZJHTCM, 1st half Apr

(e) H1, 2nd half Mar (f) H1, 1st half Apr (g) H2, 2nd half Mar (h) H2, 1st half Apr

(i) H3, 2nd half Mar (j) H3, 1st half Apr (k) H4, 2nd half Mar (l) H4, 1st half Apr

(m) H5, 2nd half Mar (n) H5, 1st half Apr

Fig. 3. Comparison of hyperareas obtained by the algorithms on main problem instances. Each box plot shows the maximum, minimum, median, first quartile (Q1), and third quartile (Q3) of hyperareas over the 30 runs of an algorithm.

[7] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, 2015.

[8] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *UK Workshop*

TABLE III
PAIRWISE COMPARISON OF COVERAGE BETWEEN NSGA-II-C AND NSGA-II AS WELL AS BETWEEN MOEA/D-C AND MOEA/D.

		NSGA-II-C (X) vs NSGA-II (X')		MOEA/D-C (X) vs MOEA/D (X')	
Hospital	Period	$Cov(X, X')$	$Cov(X', X)$	$Cov(X, X')$	$Cov(X', X)$
ZJHTCM	2nd half Feb	100%	0%	100%	0%
	1st half Mar	100%	0%	100%	0%
	2nd half Mar	100%	0%	100%	0%
	1st half Apr	98.29%	0%	100%	0%
H1	2nd half Mar	100%	0%	100%	0%
	1st half Apr	100%	0%	100%	0%
H2	2nd half Mar	100%	0%	100%	0%
	1st half Apr	98.88%	0%	100%	0%
H3	2nd half Mar	100%	0%	100%	0%
	1st half Apr	100%	0%	100%	0%
H4	2nd half Mar	100%	0%	99.53%	0%
	1st half Apr	93.42%	2.21%	95.75%	1.67%
H5	2nd half Mar	95.60%	1.31%	97.63%	0%
	1st half Apr	98.48%	0%	92.77%	1.59%

on Computational Intelligence, Colchester, UK, 2010, pp. 1–6.

[9] A. Konstantinidis and K. Yang, "Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition," *Comput Commun.*, vol. 34, no. 1, pp. 83–98, 2011.

[10] L. J. Z. M. e. a. Zhang, Y., "A multi-objective memetic algorithm based on decomposition for big optimization problems," *Memetic Comp.*, vol. 8, pp. 45–61, 2016.

[11] J. Chen, J. Li, and B. Xin, "DMOEA- εC : Decomposition-based multiobjective evolutionary algorithm with the ε-constraint framework," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 714–730, 2017.

[12] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, 2017.

[13] Q. Song, Y. Zheng, Y. Huang, Z. Xu, W. Sheng, and J. Yang, "Emergency drug procurement planning based on big-data driven morbidity prediction," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6379–6388, 2019.

[14] A. Dmitriev and V. Kotin, "Time series prediction of morbidity using artificial neural networks," *Biomed. Eng.*, vol. 47, no. 1, pp. 43–45, 2013.

[15] P. J. Landrigan, C. B. Schechter, J. M. Lipton, M. C. Fahs, and J. Schwartz, "Environmental pollutants and disease in american children: estimates of morbidity, mortality, and costs for lead poisoning, asthma, cancer, and developmental disabilities." *Environ. Health Perspect.*, vol. 110, no. 7, pp. 721–728, 2002.

[16] Q. Song, Y.-J. Zheng, Y. Xue, W.-G. Sheng, and M.-R. Zhao, "An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination," *Neurocomputing*, vol. 226, pp. 16–22, 2017.

[17] Q. Song, Y.-J. Zheng, and J. Yang, "Effects of food contamination on gastrointestinal morbidity: Comparison of different machine-learning methods," *Int. J. Env. Res. Public Health*, vol. 16, no. 5, 2019.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[19] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Differential evolution with self-adaptation and local search for constrained multiobjective optimization," in *IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 2009, pp. 195–202.

[20] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, 2009.

[21] S. Kukkonen and J. Lampinen, "GDE3: the third evolution step of generalized differential evolution," in *IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 443–450.

[22] Y. Zheng, H. Ling, J. Xue, and S. Chen, "Population classification in fire evacuation: A multiobjective particle swarm optimization approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 70–81, 2014.

[23] J. J. Liang, A. K. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, 2006.

[24] I. Derpich and J. R. Vera, "Improving the efficiency of the branch and bound algorithm for integer programming based on "flatness" information," *Eur. J. Oper. Res.*, vol. 174, no. 1, pp. 92–101, 2006.

[25] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[26] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Boston, MA: Springer, 2002.