

A Bi-Objective Constrained Robust Gate Assignment Problem: Formulation, Instances and Algorithm

Xinye Cai^{1b}, Member, IEEE, Wenxue Sun, Mustafa Misir^{2b}, Kay Chen Tan^{3b}, Fellow, IEEE, Xiaoping Li^{4b}, Senior Member, IEEE, Tao Xu, and Zhun Fan^{5b}, Senior Member, IEEE

Abstract—The gate assignment problem (GAP) aims at assigning gates to aircraft considering operational efficiency of airport and satisfaction of passengers. Unlike the existing works, we model the GAP as a bi-objective constrained optimization problem. The total walking distance of passengers and the total robust cost of the gate assignment are the two objectives to be optimized, while satisfying the constraints regarding the limited number of flights assigned to apron, as well as three types of compatibility. A set of real instances is then constructed based on the data obtained from the Baiyun airport (CAN) in Guangzhou, China. A two-phase large neighborhood search (2PLNS) is proposed, which accommodates a greedy and stochastic strategy (GSS) for the large neighborhood search; both to

speed up its convergence and to avoid local optima. The empirical analysis and results on both the synthetic instances and the constructed real-world instances show a better performance for the proposed 2PLNS as compared to many state-of-the-art algorithms in literature. An efficient way of choosing the tradeoff from a large number of nondominated solutions is also discussed in this article.

Index Terms—Bi-objective optimization, large neighborhood search (LNS), Pareto local search (PLS), robust gate assignment, solutions of interest.

I. INTRODUCTION

AIRCRAFT have become one of the major transportation choices with the varying customer needs, travelling prices and the growing number of airports. The airport gates (i.e., aircraft stand positions) play a critical role in the efficient utilization of the aircraft. Thus, their assignment to the aircraft needs to be carefully scheduled, as it greatly affects the efficient operation of airports, as well as the passengers' satisfaction. Such assignment task is usually called the gate assignment problem (GAP) [24]. The GAP requires delivering gate schedules considering various factors, e.g., the aircraft types, the domestic/international attribute of the flight, the flights' arrival/departure time, the total passenger number of the flight, and the gate preferences [22].

From a practical point of view, the gate assignments should be determined under the consideration of the possible changes on the existing flight schedules. In other words, the gate assignments should be enough robust to the minor changes on the flight schedules [22]. For instance, a late arrival of one aircraft may result in a chain of delayed arrivals for other aircraft assigned to the same gate. This may make the current gate assignments inapplicable, leading to a complete rescheduling.

For GAP, maintaining the operational efficiency in airports and the passengers' satisfaction simultaneously is a very complicated task. Under this circumstance, the GAP is modelled as a combinatorial multiobjective optimization problem (CMOP) with different and also possibly conflicting objectives [17], [19], [20], [23]–[25], [31], [36], [38]. Unlike a single-objective optimization problem where only one optimal solution exists, the multiobjective optimization problems have a set of tradeoff solutions among different

Manuscript received September 7, 2019; accepted November 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61300159, Grant 61732006, Grant 61876162, Grant 61572127, and Grant 61872077, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20181288, in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180307123637294, in part by China Postdoctoral Science Foundation under Grant 2015M571751, in part by the Aeronautical Science Foundation of China under Grant 2017552042, and in part by the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China under Grant CAAC-ITRB-201703. This article was recommended by Associate Editor S. Yang. (Corresponding author: Xinye Cai.)

X. Cai and W. Sun are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin 300300, China (e-mail: xinye@nuaa.edu.cn; sunwenxue2013@163.com).

M. Misir is with the Department of Computer Engineering, Istinye University, 34010 Istanbul, Turkey, and also with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: mustafa.misir@istinye.edu.tr).

K. C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: kaytan@cityu.edu.hk).

X. Li is with the Department of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: xpli@seu.edu.cn).

T. Xu is with the Department of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China (e-mail: txu@cauc.edu.cn).

Z. Fan is with the Guangdong Provincial Key Laboratory of Digital Signal and Image Processing, Shantou University, Shantou 515063, China, and also with the Department of Electronic Engineering, School of Engineering, Shantou University 515063, Shantou, China (e-mail: zfan@stu.edu.cn).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. This PDF file contains supplementary experimental results. This material is 6350 KB in size. The code in this paper can be downloaded at <http://xinyecai.github.io/>.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCYB.2019.2956974

objectives. These solutions are called a Pareto set (PS), forming a Pareto front (PF) as their projection on the objective space.

Despite the multiobjective nature of the GAP, it has been widely handled as a single-objective optimization problem. For instance, in [23], the GAP contains three objectives, that is, 1) maximizing the total assignment preference score; 2) minimizing the number of ungated flights; and 3) minimizing the number of tows. These objectives were aggregated into a single objective via the weighted-sum approach. The weights are usually determined either by the airport or the airline manager.

In this article, the GAP is formulated using a constrained bi-objective robust gate assignment model (CBR-GAM). The walking distance of the passengers and the robust cost of the gate assignment are treated as the two objectives. The first objective is essentially used to evaluate the passenger satisfaction [22]. The latter objective relates to the delay cost on the idle time allocation [18].

For easy access, passenger bridges are usually equipped between terminal gates and aircraft. Yet, aprons need buses to transport passengers. For shortening walking distance of passengers, terminal gates are assigned to flights preferably over aprons [24]. The assignment of flights to aprons is likely to reduce the operational efficiency of the airport. Under this circumstance, the number of flights that can be assigned to the apron is limited as a constraint in the CBR-GAM.

To address the CBR-GAM efficiently, we propose a two-phase large neighborhood search (2PLNS). In the first phase, the bi-objective optimization problem is decomposed into a set of single-objective subproblems using the weighted-sum approach [40]. One solution is maintained for each subproblem. The collaborative local search (LS) is conducted among subproblems for the fast convergence toward the PF. After that, Pareto LS (PLS) is conducted on the resulting obtained solutions, for further approximating the whole PF.

Although large neighborhood search (LNS) is an effective solution strategy, it has poor scalability with the increasing neighborhood size [28]. To alleviate such an issue as well as avoid local optima, a greedy and stochastic strategy (GSS) is introduced for guiding the LS/PLS.

To validate the effectiveness and efficiency of 2PLNS on CBR-GAM, experimental studies have been conducted on both synthetic benchmarks of different scales and a set of real instances, which is constructed based on data obtained from the Baiyun airport (CAN) in Guangzhou, China. The same experiments have also been conducted by a group of state-of-the-art multiobjective optimization algorithms for comparison. The results show the effectiveness of 2PLNS in terms of both solution quality and scalability. In addition, solutions obtained by the proposed 2PLNS are analyzed through Gantt charts for the real instance.

In summary, the main contributions of this article can be listed as follows.

- 1) A constrained bi-objective model is proposed for the robust gate assignment problem. Differently from the existing works, more realistic aircraft-gate company,

flight attribute and flight apron compatibility have been modeled as constraints (see Section III-C).

- 2) An objective function for measuring the robustness of the gate assignment is proposed. As a new addition to the literature, a weighted sum of the robust costs regarding all the flights is utilized since the importance of the flights can significantly differ from each other (see Section III-B).
- 3) A 2PLNS is designed for solving the aforementioned bi-objective model. A GSS-based LNS is proposed for effectively balancing between the exploitive and explorative search.

The rest of this article is organized as follows. Section II discusses the existing studies on the GAP as well as the relevant optimization approaches. Section III introduces the CBR-GAM in details. 2PLNS is elaborated in Section IV. The construction of real instances based on data obtained from CAN in China, as well as the synthetic instances, is given in Section V. The experimental setups are presented in Section VI. The experimental results are discussed and analyzed in Section VII. Finally, Section VIII concludes this article with a summary and the future research directions.

II. BACKGROUND

A. Gate Assignment Problem

Based on the number of objectives targeted, studies on the GAP can be primarily categorized into: the GAP with one objective, that is, single-objective, and the GAP with more than one objective, that is, multiobjective, as follows.

1) *Single-Objective GAP*: The majority of the studies try to minimize the passengers' required walking distance as their sole objective [2], [5], [7], [14], [17], [20], [24], [29], [34], [36], [37]. This distance is usually calculated based on three types of walking area: 1) the distance from check-in to gates for embarking or originating passengers; 2) the distance from gates to the baggage claim areas (check-out) for disembarking or destination passengers; and 3) the distance from gate to gate for transfer or connecting passengers.

The robustness is another commonly used objective in GAP. The robust GAP has already been investigated in [6], [13], [18], [22], [23], [26], and [37]. In [6], the idle period of time between two successive utilizations of one gate has been used for measuring the robustness. The robust gate assignments were obtained by minimizing the variance of such idle time. In [18], a utility function for measuring the robustness of the gate assignment was proposed by penalizing the shorter idle-time with much higher cost. In [23], the robustness of the gate scheduling is measured by the expected number of constraint violations, that is, ignoring gate closures, the violation of the shadow restrictions, and the gate conflicts in which two aircraft are assigned to the same gate. Although this approach can ensure that the corresponding constraints are satisfied, it does not provide any effective guidance for the construction of the robust gate assignment. A more comprehensive survey on the robustness in flight gate scheduling can be found in [22].

2) *Multiobjective GAP*: A considerable number of studies have been conducted on multiobjective GAP, which have

TABLE I
LITERATURE REVIEW OF THE GAP

Year	Reference	Objectives	Constraints	Method	Multi/Single-objective
Proposed	CBR-GAM	1) Minimize the total walking distance 2) Maximize the total robust cost of gate assignment	Const-1, Const-2 Const-type, Const-company Const-attribute, Const-Apron	2PLNS	Multi-objective
2017	Sena DAŞ [14]	1) Minimize the total walking distance 2) Maximize the number of passengers at the gates close to shopping facilities	Const-1, Const-2	Two Phase Local Search and a Pareto Local Search Algorithm (TPLS+PLS)	Multi-objective
2017	Dorndorf <i>et al</i> [23]	1) Maximize total assignment preference score 2) Minimize the number of unassigned flights 3) Minimize the number of tows	Const-1, Const-2 Shadow restrictions Tow time restrictions	Ejection Chain Algorithm	Single-objective
2017	Aktek <i>et al</i> [2]	1) Minimize the number of unassigned flights 2) Minimize the total walking distance	Const-1, Const-2	Tabu search algorithm	Single-objective
2017	Dell <i>et al</i> [17]	1) Minimize the number of unassigned flights 2) Minimize the total walking distance	Const-1, Const-2 Const-type	Fuzzy Bee Colony Optimization (FBCO)	Single-objective
2017	Zhang <i>et al</i> [39]	1) Minimize flight conflict probability 2) Minimize the number of flights assigned to aprons	Const-1, Const-2 Const-type	Biogeography-based optimization algorithm	Single-objective
2016	Yu <i>et al</i> [38]	1) Maximize the robustness of gate scheduling 2) Minimize the tow cost 3) Minimize the transfer distance	Const-1, Const-2 Const-type	general Mixed Integer Programming (MIP) solving algorithms	Single-objective
2015	Gupet <i>et al</i> [25]	1) Maximize the number of aircraft at contact stands 2) Minimize the number of towing movements	Const-1 Shadow restrictions Tow time restrictions	MIP based approaches	Single-objective
2008	Drexel <i>et al</i> [24]	1) Minimize the number of un-gated flights 2) Minimize the total walking distance of the passengers 3) Maximize the total gate assignment preferences	Const-1, Const-2	Pareto simulated annealing (PSA)	Multi-objective

Const-1 corresponds to the constraint that no two aircraft may be assigned to the same gate simultaneously.

Const-2 corresponds to the constraint that every flight must be assigned to exactly one gate.

Shadow restrictions restrict assigning two aircraft to neighboring gates because of wing tip proximity.

Tow time restrictions ensure that the total time for towing does not exceed the available time for parking.

Const-type corresponds to the aircraft-gate type compatibility, **Const-company** corresponds to the aircraft-gate company compatibility, **Const-attribute** corresponds to the flight attribute compatibility, **Const-Apron** corresponds to the maximum allowed number of flights assigned to the apron. These four constraints will be discussed in Section III.

been summarized in Table I. The third column (objectives) in Table I lists the objectives to be optimized in GAP for the corresponding reference. The fourth column (constraints) displays the restrictions to be considered during the process of gate allocation. It is clear to see that most studies only consider two basic constraints: Const-1—only one aircraft can be assigned to a gate at a certain period of time and Const-2—the constraint that every flight must be assigned to exactly one gate.

Since the CBR-GAM is mainly modelled for a real-world scenario, an extensive set of constraints is taken into account. These constraints include the aircraft-gate type compatibility, the aircraft-gate company compatibility, the flight attribute compatibility and the maximum allowed number of flights assigned to the apron as well as two basic constraints.

Table I also reports the type of approaches (single-objective or multiobjective) used to solve GAP. The studies that convert a multiobjective GAP into a single-objective GAP by aggregating all the objectives, dominate the literature. Only a limited number of research [14], [24] has adopted multiobjective approaches to approximate the whole PF. The more recent reviews on GAP can be referred to in [4], and [15].

B. Multiobjective Optimization Problem

In this article, the GAP is formulated as a MOP, which can be defined as follows:

$$\begin{aligned} & \text{maximize } F(x) = (f_1(x), \dots, f_t(x)) \\ & \text{subject to } x \in \Omega \end{aligned} \quad (1)$$

where Ω is the *decision space*, $F : \Omega \rightarrow R^t$ is composed of t objective functions. The *attainable objective set* is $\{F(x)|x \in \Omega\}$. In the case when Ω becomes a finite set, (1) is usually called a combinatorial MOP (CMOPs).

Let $u, v \in R^t$, u is said to *dominate* v (i.e., $u < v$), if and only if $u_i \leq v_i$ for every $i \in \{1, \dots, t\}$ and $u_j < v_j$ for at least one index $j \in \{1, \dots, t\}$.¹ A solution $x^* \in \Omega$ is *Pareto-optimal* to (1) if no solution $x \in \Omega$ exists such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is then called a *Pareto-optimal (objective) vector*. Apparently, any improvement in one objective of a Pareto-optimal solution will lead to the deterioration of at least another objective.

C. Decomposition Approaches

An MOP can be decomposed into a number of single-objective optimization subproblems to be solved simultaneously in a collaborative way. The representative approaches, such as [8], [9], [12], have adopted this decomposition idea. One of the most commonly used decomposition methods [30] is the weighted-sum approach, which can be stated as follows.

Suppose $\lambda^i = (\lambda_1, \dots, \lambda_t)^T$ is a direction vector regarding the i th subproblem, where $\lambda_j \geq 0$, $j \in \{1, \dots, t\}$ and $\sum_{j=1}^t \lambda_j = 1$.

Weighted Sum (WS): The i th subproblem is

$$\begin{aligned} & \text{minimize } g^{\text{ws}}(x|\lambda^i) = \sum_{j=1}^t \lambda_j^i f_j(x) \\ & \text{subject to } x \in \Omega. \end{aligned} \quad (2)$$

¹In the case of maximization, the inequality signs should be reversed.

D. Pareto Local Search and Large Neighborhood Search

As the GAP is an NP-hard combinatorial optimization problem [20], [26], the heuristic methods, such as LS and meta-heuristics [11], [20], have been widely used for addressing it. On the other hand, PLS [10], [27], [32] is a natural extension of the single-objective local search, thus it can be used with the same motivation for the multiobjective GAP. PLS works by exploring the neighbors of a nondominated solution set and update the population by recently generated nondominated solutions in an iterative way.

A key issue in LS or PLS is the choice of the neighborhood structure. As a rule of thumb, the larger the neighborhood, the better quality of the local optima are. However, it should be noted that the larger neighborhood is tended to be much slower to traverse compared to its small-sized alternatives. There exist a number of studies utilizing LNS for the GAP [33]. In [21], LNS was applied for improving a layered branch-and-bound algorithm on the GAP. An adaptive LNS (ALNS) algorithm was introduced in [37] for the robust GAP.

LNS can be used for the GAP, usually requiring two operators: 1) destroy and 2) repair. The destroy operator modifies the assignments of the partial flights. Then, the repair operator reallocates flights to the new available gates. For example, Xu and Bailey [34] devised three different operators (moves) for the neighborhood search, that is, insert, exchange_1 and exchange_2. Nevertheless, those three operators are unable to obtain high-quality solutions when the number of flights for assignment in a period of time is very large [19]. To overcome this shortcoming, an interval exchange move (IEM) [19], which partially swaps two flight sequences, was introduced. An apron swap operator was also developed to switch a flight originally assigned to the apron with a flight assigned to a terminal gate. More recently, an operator for neighborhood search called exchange and greedy move (EGM) was proposed in [14] for the GAP. In the exchange move, the gates of two successive flights are exchanged. In the greedy move, the selected flights are first assigned to the apron, then these flights are reassigned to the gates with the shortest walking distance if possible.

Nevertheless, the aforementioned operators used for LNS are computationally expensive and also ignore the feasibility constraints (see Section III-C). In this article, we propose a GSS. The greedy move prefers the gates having small possibility of time conflict, out of all the available gates for each flight constructed in advance. This strategy can significantly decrease the number of times that the feasibility constraints are matched during the allocation process, which eventually reduce the time complexity.

III. FORMULATION FOR CONSTRAINED BIOBJECTIVE ROBUST GAP

In this section, the mathematical model is provided for the constrained bi-objective robust gate assignment problem.

A. Notations

All the notations used in CBR-GAM are given in Table II. Two dummy gates have been adopted in our model, based

TABLE II
NOTATIONS IN CBR-GAM

Parameter	Description
n	Total number of flights
m	Total number of gates
a_i	Arrival time of flight i
d_i	Departure time of flight i ($a_i < d_i$)
$w_{k,l}$	Walking distance from gate k to gate l
$p_{i,j}$	The number of passengers transferring from flight i to flight j
$S_{i,k}$	The latest idle time of gate k before the arrival of flight i
$c(S_{i,k})$	The robust cost of idle time $S_{i,k}$
Co_i	Weight coefficient for the robust cost of flight i
DC_i	The delay cost of the i -th flight
DC_{min}	The minimum delay cost of all the flights
DOC	Delayed operating cost
PEL	Passenger economic loss
γ_i	Maximum passengers of flight i
PLR	Passenger load factor
$APrice$	Average ticket price
$ANPM$	Average net profit margin
$AFly$	Average flying time
ϑ_i	The unit time profit loss for flight i
$MAX Apron$	The maximum number of allowed flights parked at the apron
η	The minimum allowed safety interval in minutes

101	107	203	106	205	211
-----	-----	-----	-----	-----	-----

Fig. 1. Solution coding example.

on [20]: Gate 0 represents the entrance or exit of the airport. Gate $(m+1)$ represents the apron when the terminal gates are all occupied. The binary variable $x_{i,k} = 1$ denotes that gate k is assigned to flight i , $1 \leq k \leq m+1$, and $x_{i,k} = 0$ otherwise.

A solution x is coded in real value, as shown in Fig. 1. Each value in x represents the assigned gate number of each flight in the order from left to right based on its arrival time.

B. Objective Functions in CBR-GAM

$$\begin{aligned} \min z_1 = & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} p_{i,j} w_{k,l} x_{i,k} x_{j,l} \\ & + \sum_{i=1}^n \sum_{k=1}^{m+1} p_{0,i} w_{0,k} x_{i,k} + \sum_{i=1}^n \sum_{k=1}^{m+1} p_{i,0} w_{k,0} x_{i,k}. \end{aligned} \quad (3)$$

The first objective z_1 is the total walking distance of the passengers, presented in (3)

$$\min z_2 = \sum_{k=1}^m \sum_{i=2}^n Co_i c(S_{i,k}) x_{i,k}. \quad (4)$$

In this article, we also propose an effective objective function in (4), in terms of the total robust cost for the gate assignment, as the second objective z_2 , where the robust cost $c(S_{i,k})$ can be evaluated as follows [18]:

$$c(S_{i,k}) = \begin{cases} 1000 \left(\arctan(0.21(5 - S_{i,k})) + \frac{\pi}{2} \right), & S_{i,k} \geq \eta \\ \infty, & \text{otherwise.} \end{cases} \quad (5)$$

Unlike [18], the objective of the robustness formulated in this article is the weighted sum of the robust costs of all the flights, due to the fact that the importance of different flights

TABLE III
DELAYED OPERATING COST OF AIRCRAFT WITH DIFFERENT TYPE

Aircraft Type	Max Take-off Weight/t	Wake Type	Delayed Operating Cost (RMB/hr)
Large aircraft	> 136	Heavy wake	4167
Medium aircraft	7 ~ 136	Medium wake	2916
Small aircraft	< 7	Light wake	208

is quite different from each other. The weight coefficient Co_i of the i th flight can be computed as follows:

$$Co_i = DC_i / \min(DC) \quad (6)$$

$$DC_i = DOC + \vartheta_i + PEL \quad (7)$$

where DC_i denotes the delay cost of the i th flight, $\min(DC)$ is the minimum delay cost of all the flights, DOC is the delayed operating cost, ϑ_i is the profit loss of delayed flights, and PEL is the passenger economic loss [35]. Each of them is further specified as follows.

1) *Delayed Operating Cost*: The delayed operating cost includes the cost of the parking fee, take-off and landing fees as well as the passenger service charges for the aircraft, varying with the aircraft type. The delayed operating cost for all types of aircraft is given in Table III [35].

As shown in Table III, we can get the corresponding delayed operating cost with different aircraft types.

2) *Profit Loss of Delayed Flights*: Profit loss of delayed flights is evaluated by the hourly profit earned by each aircraft type as follows [35]:

$$\vartheta_i = \gamma_i \times PLR \times APrice \times ANPM / AFLY \quad (8)$$

where PLR denotes the ratio of the number of carried passengers with the total number of available seats, reflecting the degree of aircraft utilization; γ_i indicates the maximum passengers of flight i ; $APrice$ is the average ticket price; $ANPM$ is the profit factor (less than 1); and $AFLY$ indicates the average flight time (in hours) from the departure to the destination.

3) *Passenger Economic Loss*: The average delay cost per passenger is 50 per hour for the domestic flights and 100 per hour for the international flights [35]. It is worth noting that the economic loss for passengers can be evaluated by computing the average maximum number of passengers for the specific type of aircraft.

C. Constraints

1) Basic Constraints:

$$\sum_{k=1}^{m+1} x_{i,k} = 1, \quad 1 \leq i \leq n \quad (9)$$

$$x_{i,k} x_{j,k} (d_j - a_i) (d_i - a_j) \leq 0, \quad k \neq m+1 \quad (10)$$

$$x_{i,k} \in \{0, 1\}, \quad 1 \leq i, j \leq n, \quad 1 \leq k \leq m+1. \quad (11)$$

Equation (9) ensures that one flight is only allowed to be assigned to one and only one gate. In the condition of none available terminal gate, the flight will be assigned to the apron. Equation (10) ensures there will not be any overlapped period of time for flights assigned to the same terminal gate. Equation (11) denotes if a flight has been assigned to a gate or not.

2) *Feasibility Constraints*: In contrast to the previous works, beside the aforementioned basic constrains, some other feasibility constraints are also taken into account in this article, for modeling more realistic scenarios.

Aircraft-Gate Type Compatibility: Suppose there are two types of gates: 1) large gate and 2) small gate. Generally speaking, a large gate can park large, medium and small aircraft, and small gate can only park medium and small aircraft. The type constraint should be satisfied when assigning gates to the aircraft.

Aircraft-Gate Company Compatibility: Gates are resources that owned or leased by a particular airline company for a period of time based on a medium to long term contract. Flights can only be assigned to the area of its agent.

Flight Attribute Compatibility: According to the attribute of flights, flights can be broadly divided into two categories: domestic/international flights. Taking customs or border control checks into account, international flights should be assigned to international gates while domestic flights should be allocated to the domestic gates

$$x_{i,k} \leq o_{i,k}, \quad 1 \leq i \leq n, \quad 1 \leq k \leq m \quad (12)$$

where $o_{i,k}$ is a binary variable, representing if a terminal gate k can be allocated to flight i or not. $o_{i,k} = 1$ indicates that a terminal gate k can be allocated to flight i when the following compatibility constraints are satisfied. A terminal gate is allowed to be assigned to a flight only if the airline of such a flight is permitted for the corresponding gate. Moreover, the flight attribute must be matched with the terminal gate attribute (domestic/international); and the aircraft type must be supported by the terminal gate type. In addition, a constraint has been added to limit the number of flights assigned to the apron, based on (13)

$$\sum_{i=1}^n x_{i,m+1} \leq \text{MAXApron}. \quad (13)$$

MAXApron is a parameter determined by the airport manager, which can be revised if required.

IV. TWO-PHASE LARGE NEIGHBORHOOD SEARCH

In this section, a 2PLNS is designed for CBR-GAM. The weighted-sum approach is utilized to speed up the convergence of the population toward PFs by the collaborative LS in Phase 1, where PLS is further conducted on the obtained solutions for extending PFs in Phase 2.

Algorithm 1 specifies 2PLNS for the CBR-GAM. A set of weight vectors $W = \{\lambda^1, \dots, \lambda^N\}$ is first generated uniformly, where N is the number of subproblems/solutions. In addition, 2PLNS maintains two populations:

- 1) the working population WP for PLS;
- 2) the external archive EA, which contains the obtained nondominated solutions.

Algorithm 1: 2PLNS

Input: a stopping criterion;
max_num: the maximum number of LNS.
Output: The external archive EA .

```

1  $EA = \text{Initialization}(WP, W);$ 
  /* Phase one */
2  $EA = \text{Collaborative\_LS}(WP, W, \text{max\_num}, EA);$ 
  /* Phase two */
3  $EA = \text{PLS}(WP, \text{max\_num}, EA);$ 
4 If the stopping criteria has been met, terminate and output  $EA$ .
  Otherwise, go to Step 3.
```

Algorithm 2: Initialization

Input: $WP = \{x^1, \dots, x^N\}$
Output: WP, W

```

1 A CBR-GAM is decomposed into  $N$  subproblems by the weighted sum
  approach with  $W = \{\lambda^1, \dots, \lambda^N\}$ . Each solution  $x^i$  of the  $i$ -th
  subproblem regarding  $\lambda^i$  is initialized by a heuristic.
2 The Euclidean distance between any two weight vectors are computed
  for obtaining the  $T$  closest weight vectors for every subproblem. For
  each  $i \in \{1, \dots, N\}$ , a set  $B(i) = \{i_1, \dots, i_T\}$ , where  $\{\lambda^{i_1}, \dots, \lambda^{i_T}\}$ ,
  stores  $T$  closest weight vectors of  $\lambda^i$ .
```

Algorithm 3: Collaborative_LS

Input : $WP = \{x^1, \dots, x^N\}, EA, W, \text{max_num}$
Output: WP, EA .

```

1  $count = 0;$ 
2 while  $count < \text{max\_num}/2$  do
3   foreach  $i \in \{1, \dots, N\}$  do
4     /*  $LNS(x)$  denotes  $x$ 's neighbors */
5     foreach  $y \in LNS(x^i)$  do
6       if  $j \in B(i) \&\& g^{ws}(y|\lambda^j) \leq g^{ws}(x^i|\lambda^j)$  then
7          $x^j = y;$ 
8       end
9     end
10     $count ++;$ 
11 end
```

A. Initialization

Algorithm 2 presents the initialization procedure. First, a CBR-GAM is decomposed into N subproblems by the weighted-sum approach, with a set of uniform weight vectors W . WP is initialized by a heuristic as follows. For each solution $x^k \in WP$, the flights are sorted with respect to the arrival time. For each flight, its available gates are composed of all the gates which satisfy all the constraints (see Section III-C). A gate is randomly chosen for each flight from its available gates in the case of ensuring that there is no time conflict between the current flight and other flights on the gate. If all terminal gates have been occupied, the flights are assigned to the apron. Each solution x^i of the i th subproblem is associated with the weight vector λ^i . The T closest weight vectors of each weight vector λ^i regarding the i th subproblem are considered as its neighbors.

B. Phase 1

In Algorithm 3, the population WP seeks the fast convergence toward PF by the collaborative local search. For each solution $x \in WP$, we use LNS (Algorithm 5) to generate its

Algorithm 4: PLS

Input : $WP, \text{max_num}, EA$
Output: EA .

```

1 Set  $TS = EA$ .
2  $count = 0;$ 
3 while  $count < \text{max\_num}/2$  do
4   foreach  $x \in WP$  do
5     /*  $LNS(x)$  denotes  $x$ 's neighbors. */
6     foreach  $y \in LNS(x)$  do
7       if  $y$  is not dominated by any solution in  $EA$ , add  $y$  to  $EA$ ;
8       and remove those solutions dominated by  $y$  from  $EA$ .
9     end
10     $count ++;$ 
11 end
12  $WP = EA \setminus TS$ .
13 if  $|WP| == 0$  then
14    $WP = EA;$ 
15 end
```

Algorithm 5: LNS

Input : x
Output: A neighborhood of x .

/* LNS based on greedy and stochastic strategy (GSS) */

```

1 Randomly remove  $\Upsilon$  flights from  $x$  to form a set  $\Phi$ .
  /*  $\Upsilon$  is the number of flights to be removed (the
  neighborhood step size); */
2  $G(h) = \{\text{The feasible gates for the flight } h \in \Phi\}$ .
3 foreach  $h \in \Phi$  do
4   Randomly generate a number  $r \in [0, 1]$ .
5   /*  $P$  is the probability of applying greedy
6   strategy; */
7   if  $r \leq P$  then
8     /* greedy strategy */
9     Sort all the gates in  $G(h)$  based on the number of flights that
10    can be assigned to them in an ascending order.
11   end
12   /* To determine whether the flight  $h$  can be
13   assigned to a new gate. */
14    $isAssigned = \text{false};$ 
15   foreach  $g \in G(h)$  do
16     if the flight  $h$  is not time conflicting with the assigned flights
17     of gate  $g$  then
18       Assign  $g$  to  $h$ ;
19        $isAssigned = \text{true};$ 
20       break;
21     end
22   end
23   if  $isAssigned == \text{false}$  then
24     The apron is assigned to  $h$ .
25   end
26 end
```

neighboring solutions. Then, each solution $y \in LNS(x)$ is utilized for updating x 's neighborhood using the weighted-sum approach, as presented in (2).

C. Phase 2

PLS in Phase 2 is presented in Algorithm 4. PLS is conducted by searching the neighborhood $LNS(x)$ for each solution x in WP . Then, $LNS(x)$ is utilized for updating EA . For each newly generated solution $y \in LNS(x)$, if it is not dominated by any solutions in EA , then y will be added to EA while all the solutions dominated by it will be removed from EA . After that, the newly added solutions in EA are stored in WP for the next round of PLS. If WP is empty,

TABLE IV
FLIGHT DATA AND GATE DATA EXAMPLE

Flight Data							Gate Data			
Flight ID	Entry Time	Departure Time	District	Company	Type	Passengers	Gate ID	District	Company	Type
CZ1056	2016-07-01 00:05:00	2016-07-01 06:40:00	Domestic	CZ	738	172	101	International	2P,3K,3U,5J,7P,8M,AE	312,313,319,320,321,32B,32V,732,733
CA1072	2016-07-01 01:05:00	2016-07-01 07:05:00	Domestic	CZ	73G	138	124	Domestic	9C,BK,CA,GJ,HO,KY,SC,ZH	732,733,734,735,736,738,73G,73H,73W
9C1015	2016-07-01 01:20:00	2016-07-01 02:25:00	International	9C	320	128	201	Domestic	CZ,MF,NS,OQ	312,313,319,320,321,32B,32V,332,342,732

Flight Data has seven attributes, those are Flight ID, Entry Time, Departure Time, District, Company and Type.
Gate Data has four attributes, those are Gate ID, District, Company and Type.

EA is assigned to WP. The variable count records the total number of neighborhood search [i.e., the times of calling LNS(x)]. When it reaches a predefined value, the algorithm is terminated.

D. Large Neighborhood Search

Algorithm 5 describes the process of LNS. The neighborhood of a solution x can be generated as follows. First, Υ flights are removed from x and stored as a flight set Φ . After that, for each flight $h \in \Phi$, its feasible gates that satisfied the feasibility constraints (see Section III-D) are stored as a gate set $G(h)$. For each flight $h \in \Phi$, if a random number r is not larger than a predefined probability P , a greedy strategy is applied by sorting all the gates in $G(h)$ based on the number of flights that can be assigned to them in an ascending order. In this way, the gates with the less number of flights are preferred for reducing the risk of possible conflicts between different flights. A flag variable *isAssigned*, which denotes whether the flight h can be assigned to a new gate, is set to False. For each gate $g \in G(h)$, if h is not time conflicting with the assigned flights on gate g , then g is assigned to h and the flag *isAssigned* is set to true. After traversing all the gates of $G(h)$, if the flight h is still unassigned (i.e., flag *isAssigned* is false), then the apron is assigned to the flight h .

An example of LNS referring to Algorithm 5 is illustrated in Fig. 2, where the top subfigure shows the original gate schedule and the bottom one displays its neighbor. Assume that the neighborhood step size Φ is 2; all the gates are feasible for Flight 5 and only Gate1 and Gate2 are feasible for Flight 6. A neighbor of a solution x is generated by LNS as follows. First, Flight 5 and Flight 6 are removed from x to form the flight set Φ (line 1). This indicates that Gate1 and Gate2 are no longer assigned to Flight 5 and Flight 6. They are to be reassigned to the new gates. We further assume that the randomly generated number r is not larger than P for Flight 5 and it is larger than P for Flight 6. This means that the feasible gates of Flight 5 are to be sorted by a greedy strategy and that of Flight 6 keeps in a random order (lines 4–7). For Flight 5, its feasible gates are tracked one by one to find a feasible gate g that has no time conflict with the existing assigned flights on g . In the example, Gate3 is assigned to Flight 5 and the flag variable *isAssigned* is set to true (lines 9–15). If no available gate can be found for Flight 5 (i.e., flag *isAssigned* is false), then the apron is assigned to Flight 5 (lines 16–18). Similarly, Gate1 is assigned to Flight 6.

It is worth noting that this large neighborhood structure has been applied in both first and second phases for LS/PLS.

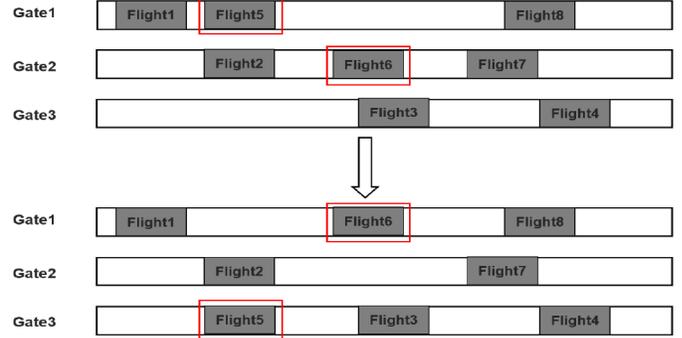


Fig. 2. LNS example.

E. 2PLNS Versus TPLS+PLS

TPLS+PLS [14] is also a two-phase algorithm for the multiobjective gate assignment problem. However, 2PLNS and TPLS+PLS are fundamentally different in the following aspects.

- 1) In Phase one, both TPLS + PLS and 2PLNS adopt the weighted-sum approach to transform a MOP into a set of a single-objective subproblems. However, TPLS + PLS conduct LS on each of these subproblems at a time without any collaboration among subproblems, while 2PLNS conducts the collaborative LS among these subproblems simultaneously. This can greatly help speed up the convergence in Phase 1 due to the collaboration among subproblems. In this sense, Phase 1 of 2PLNS is similar to MOEA/D [40].
- 2) More importantly, a LNS is tailored for the GAP in both phases of 2PLNS, which makes it fundamentally different from TPLS + PLS. More specifically, a GSS is designed for effectively balancing between the exploitive and explorative search in 2PLNS, leading to the satisfactory performance, as presented in Section III of the supplementary material.

V. INSTANCES

A. Construction of Real Instances

A set of real instances is constructed based on the data from the CAN, China. CAN is one of the three major international hub airports in China with 68 fixed terminal gates. As shown in Table IV, each flight has seven attributes: flight ID, entry time, departure time, the district, company, type, and passengers. The flight ID is an identity number of a flight. The entry time and departure time are the arrival time and leaving time of the flight. District means whether the flight is domestic or



Fig. 3. Terminal map of Guangzhou Baiyun international airport [1].

international. Company represents the owner of the flight. Type represents the aircraft type. Passengers indicates the number of passengers on the aircraft. It is worth noting that the number of passengers is estimated based on the aircraft type and the seat occupancy rate randomly generated within a reasonable range [3].

It can be observed from Fig. 3 that the facility has 68 fixed terminal gates with boarding bridges, 33 boarding gates in area A and 35 boarding gates in area B. The boarding gates in area A start from A101 to A133 and the boarding gates in area B start from B201 to B235. Among them, the gates from A101 to A112 can only provide service for the international flights. In addition, as presented in Table IV, each gate has four attributes: gate ID, international or domestic gate, companies that the gate provides service for and the aircraft types. The distances are measured based on navigation map provided by the airport. Due to the page limit, more details of the real instance construction can be found in Section I of the supplementary material.

B. Test Instances

Eight days' real instances and nine synthetic instances of different scales are used. As shown in Table V, the real instances are named from Day-1 to Day-8. They contain 402, 427, 400, 426, 409, 438, 425, and 432 flights, respectively. The number of gates in the real instances are fixed to 68 and the value of MaxApron is set to 45 except Day-8 instance. The synthetic instances of different sizes are called S120-G16, S150-G18, S180-G18, M200-G20, M220-G20, M280-G24, L350-G28, and L400-G30, respectively. More details on instances can be found in Table V.

The synthetic instances are generated in a similar way to [24]. The feasibility constraint in the CBR-GAM is considered in our instance generation procedure unlike [24]. This means that the information on the aircraft-gate type compatibility, aircraft-gate company compatibility, aircraft-gate mission compatibility and flight attribute compatibility should be present. To simplify the instance generation process, it

TABLE V
TEST INSTANCE

Instance Name	Number of Flights	Number of Gates	MaxApron
Day-1	402	68	45
Day-2	427	68	45
Day-3	400	68	45
Day-4	426	68	45
Day-5	409	68	45
Day-6	438	68	45
Day-7	425	68	45
Day-8	432	68	50
S80-G10	80	10	20
S120-G16	120	16	20
S150-G18	150	18	20
S180-G18	180	18	20
M200-G20	200	20	35
M220-G20	220	20	35
M280-G24	280	24	35
L350-G28	350	28	40
L400-G30	400	30	40

is assumed that the constraints of flight attribute compatibility and aircraft-gate mission compatibility are satisfied for all pairs of aircraft and gates. The number of passengers on the aircraft should be determined by the type of aircraft, so its capacity. Correspondingly, there are two types of gates: large gates and small gates. For aircraft-gate compatibility, large flights can only be allocated to large gate. The medium and small flights can be parked to any type of gates. For aircraft-gate company compatibility, we randomly generate the companies which the flights and gates affiliated to. The constraint is satisfied only if the flight airline is in the supported list regarding to the airport gate.

VI. EXPERIMENTAL SETUPS

A. Parameter Settings

In the experimental studies, 2PLNS is compared with a Pareto dominance-based approach (NSGA-II [16]), four decomposition-based approaches (MOEA/D-LS (WS, TCH, PBI) [40] and Pareto simulated annealing (PSA) [24]) and a state-of-the-art TPLS+PLS [14], which is a hybridization

TABLE VI
VALUES OF C-METRIC (%) BETWEEN 2PLNS AND TPLS+PLS, NSGA-II-PLS, PSA
AND MOEA/D-LS (WS, TCH, PBI) OVER 20 RUNS ON 17 TEST INSTANCES

instance	TPLS+PLS		NSGA-II-PLS		PSA		MOEA/D-LS(WS)		MOEA/D-LS(TCH)		MOEA/D-LS(PBI)		
	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	
Real	Day-1	70.23	17.93	100	0.00	100	0.00	48.72	25.74	98.52	0.36	59.46	18.16
	Day-2	91.10	4.70	100	0.00	100	0.00	43.02	34.97	59.57	24.27	81.03	5.25
	Day-3	46.48	38.45	82.56	2.83	100	0.00	39.73	36.70	67.07	22.87	32.45	24.74
	Day-4	95.19	2.73	88.19	1.89	100	0.00	57.05	28.13	66.03	13.52	71.30	4.01
	Day-5	44.02	37.61	89.02	1.78	100	0.00	74.46	16.52	81.62	6.36	67.14	10.42
	Day-6	68.58	23.66	95.89	1.17	81.90	13.49	58.87	28.40	89.33	3.89	72.52	11.43
	Day-7	53.20	35.71	100	0.00	100	0.00	35.58	32.39	84.80	10.51	82.46	3.91
	Day-8	87.34	8.09	83.21	4.84	100	0.00	61.73	18.02	100	0.00	36.18	15.81
Synthetic	S80-G10	59.24	40.69	48.78	11.63	98.64	0.09	95.08	0.12	96.90	1.06	95.69	0.60
	S120-G16	70.68	25.75	80.97	8.38	98.15	1.04	78.97	6.44	90.15	3.92	93.94	2.08
	S150-G18	60.23	33.24	52.70	42.58	100	0.00	41.88	46.15	58.64	29.12	46.01	30.21
	S180-G18	75.00	24.49	83.06	17.14	100	0.00	31.66	46.54	72.67	10.69	40.17	18.70
	M200-G20	27.79	59.90	95.49	0.79	100	0.00	0.00	100	71.01	16.36	69.63	11.49
	M220-G20	92.69	6.72	90.37	2.01	100	0.00	52.04	13.43	61.76	17.11	89.68	2.73
	M280-G24	45.62	46.79	83.42	6.62	100	0.00	18.40	62.11	45.56	29.60	70.07	23.18
	L350-G28	79.15	15.66	78.76	18.29	100	0.00	4.62	86.21	97.43	0.75	98.33	0.62
L400-G30	75.39	19.82	100	0.00	100	0.00	64.71	24.49	69.51	16.07	99.28	0.06	

A corresponds to 2PLNS.

B corresponds to the compared algorithms.

of TPLS [32] and PLS, on both synthetic and real instances. The parameter settings of all the compared algorithms are summarized as follows.

- 1) *The Number of Subproblems N for 2PLNS and TPLS+PLS is Set to 100*: The population size for MOEA/D-LS (WS, TCH, PBI), NSGA-II-PLS and PSA is set to 100 for all the real and synthetic instances.
- 2) *Neighborhood LNS(x) for Local Search*: For a fair comparison, GSS is used for generating the neighboring solutions for all the compared algorithms. In LNS, we set $\Upsilon = 10$ and $P = 0.8$.
- 3) *Stopping Criterion*: For all the compared algorithms, it is terminated when the number of calling LNS(x) is up to $1E + 06$ for all the test instances.
- 4) *Miscellaneous*: For the model, safety interval in minutes η is set to 10. All the other parameters in the compared algorithms are set based on their original references [14], [16], [24], [40] as follows. The neighborhood size for MOEA/D-LS (WS, TCH, PBI) is set to 20. The penalty parameter θ for PBI is set to 5. For PSA, the initial annealing temperature $T_0 = 2*$ (the number of flights), cooling schedule $T_q = 0.998 \times T_0$, the acceptance probability a and b are set to 1. Each algorithm was run 20 times on each test instance. The solution coding of all the other compared algorithms is the same as that in 2PLNS, as shown in Fig. 1.

B. Performance Metrics

- 1) *Set Coverage (c-Metric) [41]*: Let A and B be two PF approximations for an MOP. $C(A, B)$ indicates the percentage of the solutions in B dominated by at least one solution in A , which can be stated as follows:

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|} \times 100\%. \quad (14)$$

Note that $C(A, B)$ is not necessarily equal to $1 - C(A, B)$. However, $C(A, B) = 0$ means that no solution in B is dominated by a solution in A while $C(A, B) = 1$ indicates that all solutions in B are dominated by solutions in A .

- 2) *Hypervolume Indicator (I_H) [41]*: Suppose $z^r = (z_1^r, \dots, z_m^r)^T$ is a reference point that is dominated by all the Pareto-optimal objective vectors. Assume P is the obtained PF approximation. Then, the I_H value of P (regarding z^r) is the volume of the region dominated by P and constrained by z^r , as follows:

$$I_H(P) = \text{volume} \left(\bigcup_{f \in P} [f_1, z_1^r] \times \dots \times [f_m, z_m^r] \right). \quad (15)$$

The higher the Hypervolume value, the better the approximation is. In the experiments, z^r is set to the 1.1 times of the maximum objective values of all the nondominated solutions delivered by all the algorithms.

VII. EXPERIMENTAL ANALYSIS

The following experimental studies have been conducted:

- 1) comparisons of 2PLNS with TPLS+PLS [14], NSGA-II-PLS [16], PSA [24] and MOEA/D-LS (WS, TCH, PBI) [40] for CBR-GAM;
- 2) investigation of the effect of GSS (see Section III of the supplementary material);
- 3) bi-objective optimization versus single-objective optimization;
- 4) sensitivity test of the parameters in 2PLNS;
- 5) selection for the solutions of interest (SOI) from a Pareto approximated set as well as their Gantt charts (see Section IV of the supplementary material).

TABLE VII
AVERAGE CPU TIME (IN SECONDS) OF 2PLNS, TPLS+PLS, NSGA-II-PLS, PSA
AND MOEA/D-LS (WS, TCH, PBI) ON 11 TEST INSTANCES OVER 20 RUNS

instance	2PLNS	TPLS+PLS	NSGA-II-PLS	PSA	MOEA/D-LS(WS)	MOEA/D-LS(TCH)	MOEA/D-LS(PBI)
Day-1	3540	6282	4560	4920	4343	4660	4423
Day-2	4066	6026	5529	5157	4650	4838	4347
Day-3	4434	6637	5417	5350	4613	4860	4726
Day-4	3420	6067	4446	4789	4100	4341	4104
Day-5	3622	6410	4680	4541	4394	4485	4361
Day-6	4108	6883	5229	4868	4392	4524	4285
Day-7	4210	6190	4526	4365	4474	4459	4453
Day-8	5150	6901	5376	5057	4515	4620	4514
S120-G16	246	483	487	100	310	379	350
M220-G20	1020	1168	1050	120	840	910	915
L400-G30	2663	4860	4200	260	3480	3260	3750

TABLE VIII
MEAN AND STANDARD DEVIATIONS OF HYPERVOLUME VALUES OBTAINED BY 2PLNS, TPLS+PLS, NSGA-II-PLS, PSA
AND MOEA/D-LS (WS, TCH, PBI) OVER 20 RUNS ON NINE SYNTHETIC INSTANCES OF DIFFERENT SIZES

instance	2PLNS	TPLS+PLS	NSGA-II-PLS	PSA	MOEA/D-LS(WS)	MOEA/D-LS(TCH)	MOEA/D-LS(PBI)
S80-G10	2.784e-01 (1.4e-03)	2.549e-01 (5.5e-03)−	2.756e-01 (4.2e-03)−	2.747e-01 (1.3e-03)−	2.758e-01 (1.7e-03)−	2.765e-01 (2.8e-03)−	2.684e-01 (3.9e-03)−
S120-G16	1.806e-01 (2.4e-03)	1.787e-01 (2.1e-03)≈	1.790e-01 (1.7e-03)−	1.778e-01 (1.7e-03)−	1.767e-01 (1.2e-03)−	1.786e-01 (9.8e-04)−	1.709e-01 (2.1e-03)−
S150-G18	4.901e-01 (2.0e-03)	4.812e-01 (3.3e-03)−	4.858e-01 (2.9e-03)−	4.329e-01 (4.6e-03)−	4.849e-01 (2.7e-03)−	4.816e-01 (2.9e-03)−	4.455e-01 (4.2e-03)−
S180-G18	4.682e-01 (3.0e-03)	4.568e-01 (2.8e-03)−	4.632e-01 (2.4e-03)−	4.118e-01 (2.7e-03)−	4.635e-01 (1.7e-03)−	4.594e-01 (2.9e-03)−	4.182e-01 (5.0e-03)−
M200-G20	2.142e-01 (1.0e-03)	2.120e-01 (8.6e-04)−	2.130e-01 (2.3e-03)−	1.977e-01 (1.0e-03)−	2.277e-01 (6.8e-02)−	2.101e-01 (3.2e-03)−	1.908e-01 (4.0e-03)−
M220-G20	1.981e-01 (1.6e-03)	1.915e-01 (2.3e-03)−	1.963e-01 (2.7e-03)−	1.818e-01 (1.7e-03)−	1.967e-01 (2.3e-03)−	1.953e-01 (3.2e-03)−	1.845e-01 (1.7e-03)−
M280-G24	4.203e-01 (2.8e-03)	4.071e-01 (2.1e-03)−	4.133e-01 (2.7e-03)−	3.427e-01 (3.5e-03)−	4.140e-01 (2.4e-03)−	4.094e-01 (2.7e-03)−	3.818e-01 (3.9e-03)−
L350-G28	4.097e-01 (2.6e-03)	3.945e-01 (4.1e-01)−	3.988e-01 (2.6e-03)−	3.142e-01 (3.5e-03)−	4.065e-01 (3.1e-03)−	3.984e-01 (2.4e-03)−	3.984e-01 (2.7e-03)−
L400-G30	2.461e-01 (1.7e-03)	2.422e-01 (1.7e-03)−	2.416e-01 (1.1e-03)−	1.988e-01 (2.4e-03)−	2.454e-01 (1.5e-03)≈	2.401e-01 (2.1e-03)−	2.314e-01 (1.4e-03)−

‘+’, ‘−’ and ‘≈’ denotes that the performance of the corresponding algorithm is significantly better, worse or similar to that of 2PLNS on the test instance, respectively. Wilcoxon rank sum test at a 0.05 significance level is performed.

A. Comparison of 2PLNS With Other Algorithms

To understand the convergence behavior of seven compared algorithms, all the nondominated solutions of 2PLNS over 20 runs are compared with those of TPLS+PLS, NSGA-II-PLS, PSA, and MOEA/D-LS (WS, TCH and PBI), in terms of c-metric, on eight real instances and nine synthetic instances with different sizes, as shown in Table VI. It is clear to see that 2PLNS significantly outperforms all the other algorithms on all the real instances and most of the synthetic instances, in terms of convergence. The final average CPU time (in seconds) used by all the algorithms over 20 runs are also presented in Table VII. Obviously, 2PLNS is the fastest among all the compared algorithms on most instances, which indicates that 2PLNS is an efficient algorithm. It is worth noting that, although PSA runs much faster than other algorithms on the synthetic instances, its performance is much worse. This can be explained by the fact that the PSA may select dominated solutions for local search, leading to less efficient search with much less solutions entering the external population that costs much less CPU time. The final nondominated solutions and boxplots in terms of hypervolume obtained by the seven compared algorithms on both synthetic and real instances can be found in Section II of the supplementary material.

Furthermore, the performance of all the compared algorithms, in terms of hypervolume, on nine synthetic instances of different sizes, is presented in Table VIII. It can be observed that the performance of 2PLNS is significantly better than that of other compared algorithms in almost all the instances. Its performance is better than that of TPLS+PLS on S120-G16

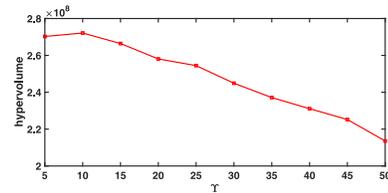


Fig. 4. Hypervolume values obtained by 2PLNS with different Γ values on Day-1 instance.

and MOEA/D-LS (WS) on L400-G30 without any statistical significance.

B. Bi-Objective Optimization Versus Single-Objective Optimization

Bi-objective optimization was used to solve the GAP model. Compared with single-objective optimization, bi-objective optimization is able to obtain a PF approximation, which can be of great help for the decision maker to understand the tradeoff between two conflicting objectives and select their preferred solutions.

In addition, the single-objective optimization for two different objectives (total robust cost and walking distance) can be considered as one subproblems with the weight vector (1, 0) and one with the weight vector (0, 1) in the bi-objective optimization respectively. Experimental studies on single-objective optimization have been conducted in this section. All the experimental setups are the same to the bi-objective optimization.

TABLE IX
THE MEAN OBJECTIVE VALUES (TOTAL WALKING DISTANCE OR TOTAL ROBUST COST) OBTAINED BY THE SINGLE-OBJECTIVE OPTIMIZATION VERSUS THAT AT THE BOTH ENDS OF THE PF APPROXIMATION OBTAINED BY THE BI-OBJECTIVE OPTIMIZATION, OVER 20 RUNS ON 11 TEST INSTANCES

instance	Total Walking Distance		Total Robust Cost	
	2PLNS(Bi-objective)	LNS(Single-objective)	2PLNS(Bi-objective)	LNS(Single-objective)
Day-1	1.082e+04 (9.9e+01)	1.055e+04 (5.6e+01) ⁺	2.197e+04 (2.4e+02)	2.225e+04 (1.8e+02) ⁻
Day-2	1.115e+04 (1.2e+02)	1.220e+04 (1.4e+01) ⁻	2.845e+04 (1.2e+02)	2.925e+04 (3.9e+02) ⁻
Day-3	1.085e+04 (5.5e+01)	1.066e+04 (2.6e+01) ⁺	2.204e+04 (1.7e+02)	2.255e+04 (3.1e+02) ⁻
Day-4	1.193e+04 (4.8e+01)	1.258e+04 (2.1e+01) ⁻	2.971e+04 (5.3e+02)	3.033e+04 (3.2e+02) ⁻
Day-5	9.910e+03 (9.4e+01)	1.163e+04 (2.2e+01) ⁻	2.454e+04 (3.2e+02)	2.455e+04 (1.9e+02) [≈]
Day-6	1.199e+04 (6.5e+01)	1.284e+04 (2.0e+01) ⁻	3.347e+04 (5.1e+02)	3.349e+04 (3.9e+02) [≈]
Day-7	1.151e+04 (9.8e+01)	1.134e+04 (9.0e+01) ⁺	2.882e+04 (1.9e+02)	2.971e+04 (3.0e+02) ⁻
Day-8	1.238e+04 (2.7e+01)	1.335e+04 (1.3e+01) ⁻	3.018e+04 (2.8e+02)	3.035e+04 (2.9e+02) ⁻
S120-G16	3.207e+04 (1.2e+01)	3.206e+04 (1.2e-11) [≈]	1.966e+04 (4.1e+02)	2.010e+04 (4.5e+02) ⁻
M220-G20	6.449e+04 (5.7e+01)	6.443e+04 (1.9e+01) [≈]	3.706e+04 (4.6e+02)	3.864e+04 (8.7e+02) ⁻
L400-G30	1.422e+05 (6.7e+01)	1.426e+05 (2.2e+01) ⁻	9.885e+04 (1.2e+03)	1.022e+05 (9.0e+02) ⁻

⁺, ⁻ and [≈] denotes that the performance of the corresponding algorithm is significantly better, worse or similar to that of 2PLNS on the test instance, respectively. Wilcoxon's rank sum test at a 0.05 significance level is performed.

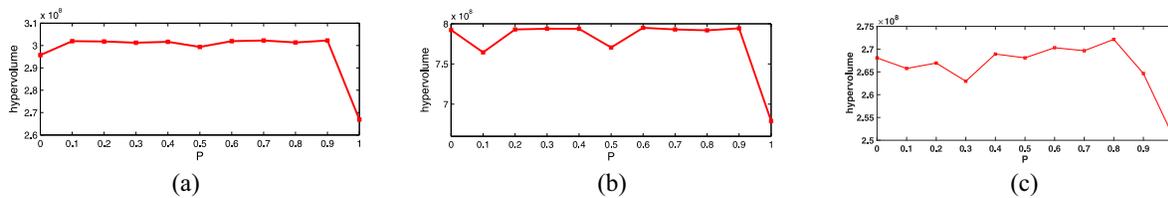


Fig. 5. Hypervolume values obtained by 2PLNS with different P values on instances of different sizes. (a) Hypervolume values obtained by 2PLNS with different P values on S120-G16 instance. (b) Hypervolume values obtained by 2PLNS with different P values on M220-G20 instance. (c) Hypervolume values obtained by 2PLNS with different P values on Day-1 (L402-G68) instance.

Table IX shows the mean objective values obtained by the single-objective optimization, compared with that at both ends of the PF approximation [i.e., the solution of the subproblem with the weight vector $(1, 0)$ or $(0, 1)$] obtained by the bi-objective optimization, over 20 runs on eleven test instances.

It can be clearly observed that the mean objective values obtained by the bi-objective approach are significantly lower (better) than those obtained by the single-objective optimization on most instances, which can be explained as follows. In the decomposition-based bi-objective optimization, the optimization for the subproblem with the weight vector $(1, 0)$ or $(0, 1)$ is conducted in a collaborative manner with their neighboring subproblems, which can be helpful for them to get out of the local optimal.

C. Parameter Sensitivity Test

As the neighborhood step size Υ and the probability of applying greedy strategy P are two important parameters in 2PLNS, their sensitivity test is conducted in this section. Fig. 4 shows the performance of 2PLNS in terms of hypervolume with different Υ on Day-1 instance. It can be observed in Fig. 4 that $\Upsilon = 10$ when 2PLNS has the best performance. In addition, Fig. 5 shows the performance of 2PLNS in terms of hypervolume with different P values on instances of different sizes. It can be observed in Fig. 5 that $P = 0.8$ or 0.9 when 2PLNS has the best performance. Another interesting observation is that optimal probability of applying greedy strategy

(i.e., P value) in the LNS increases with the increase of the problem size.

VIII. CONCLUSION

The present study proposed a CBR-GAM. Walking distance of passengers and robust cost of the gate assignment solution were treated as the two objectives. To address CBR-GAM efficiently, a 2PLNS was devised. A LNS called GSS was developed to accelerate convergence speed and avoid local optima. Nine synthetic instances of different scales, as well as eight real instances obtained from the CAN in Guangzhou, China, were constructed. The performance of 2PLNS has been compared against five classical multiobjective optimization algorithms, MOEA/D-LS (WS,TCH,PBI), NSGA-II-PLS, PSA; and a state-of-the-art algorithm (TPLS + PLS) on both real and synthetic instances. The experimental results indicate that 2PLNS has the best overall performance in terms of both convergence and diversity. In addition, the recursive computation of the expected marginal utility (EMU) has been adopted to identify the SOI out of a large number of Pareto approximated solutions. Gantt charts based on the selected SOI verify the effectiveness of the gate scheduling.

The uncertainty of the arrival and departure time has not yet been considered in the airport gate schedule. One of our future research direction includes formulating and addressing the GAP with stochastic arrival and departure time.

REFERENCES

- [1] *Terminal Map of Guangzhou Baiyun International Airport, China*. Accessed: Dec. 1, 2017. [Online]. Available: <https://www.gbiac.net/byairport-web/amenity/amenity?urlKey=plan>
- [2] A. Aktel, B. Yagmahan, T. Özcan, M. M. Yenisey, and E. Sansarçi, "The comparison of the metaheuristic algorithms performances on airport gate assignment problem," *Transport. Res. Procedia*, vol. 22, pp. 469–478, May 2017.
- [3] Alibaba. (2016). *Details of the Allocation of Resources for Gate Scheduling*. [Online]. Available: <https://tianchi.aliyun.com/outsourcing/offer/projectdetails.htm?id=3092>
- [4] H. An, N. King, and S. O. Hwang, "Issues and solutions in air-traffic infrastructure and flow management for sustainable aviation growth: A literature review," *World Rev. Intermodal Transport. Res.*, vol. 8, no. 4, pp. 293–319, 2019.
- [5] O. Babić, D. Teodorović, and V. Tošić, "Aircraft stand assignment to minimize walking," *J. Transport. Eng.*, vol. 110, no. 1, pp. 55–66, 1984.
- [6] A. Bolat, "Procedures for providing robust gate assignments for arriving aircrafts," *Eur. J. Oper. Res.*, vol. 120, no. 1, pp. 63–80, 2000.
- [7] J. P. Braaksma and J. H. Shortreed, "Improving airport gate usage with critical path," *Transport. Eng. J.*, vol. 97, no. 2, pp. 187–203, 1971.
- [8] 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, Aug. 2015.
- [9] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 564–577, Aug. 2018.
- [10] X. Cai, H. Sun, Q. Zhang, and Y. Huang, "A grid weighted sum Pareto local search for combinatorial multi and many-objective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3586–3598, Sep. 2019.
- [11] X. Cai, C. Xia, Q. Zhang, Z. Mei, H. Hu, and L. Wang, "The collaborative local search based on dynamic-constrained decomposition with grids for combinatorial multiobjective optimization," *IEEE Trans. Cybern.*, to be published, doi: [10.1109/TCYB.2019.2931434](https://doi.org/10.1109/TCYB.2019.2931434).
- [12] X. Cai, Z. Yang, Z. Fan, and Q. Zhang, "Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2824–2837, Sep. 2017.
- [13] J. Castaing, I. Mukherjee, A. Cohn, L. Hurwitz, A. Nguyen, and J. J. Müller, "Reducing airport gate blockage in passenger aviation: Models and analysis," *Comput. Oper. Res.*, vol. 65, pp. 189–199, Jan. 2014.
- [14] G. S. Daş, "New multiobjective models for the gate assignment problem," *Comput. Ind. Eng.*, vol. 109, pp. 347–356, Jul. 2017.
- [15] G. S. Daş, F. Gzara, and T. Stützle, "A review on airport gate assignment problems: Single versus multi objective approaches," *Omega*, to be published. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305048319306000>
- [16] 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, Apr. 2002.
- [17] M. Dell'Orco, M. Marinelli, and M. G. Altieri, "Solving the gate assignment problem through the fuzzy bee colony optimization," *Transport. Res. C Emerg. Technol.*, vol. 80, pp. 424–438, Jul. 2017.
- [18] G. Diepen, J. M. V. D. Akker, J. A. Hoogeveen, and J. W. Smeltink, "Finding a robust assignment of flights to gates at Amsterdam airport Schiphol," *J. Scheduling*, vol. 15, no. 6, pp. 703–715, 2012.
- [19] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, "New heuristics for over-constrained flight to gate assignments," *J. Oper. Res. Soc.*, vol. 55, no. 7, pp. 760–768, 2004.
- [20] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, "The over-constrained airport gate assignment problem," *Comput. Oper. Res.*, vol. 32, no. 7, pp. 1867–1880, 2005.
- [21] U. Dorndorf, "Project scheduling with time windows: From theory to applications," vol. 83, Darmstadt Tech. Univ. Inst. Bus. Stud., Darmstadt, Germany, 2002.
- [22] U. Dorndorf, A. Drexler, Y. Nikulin, and E. Pesch, "Flight gate scheduling: State-of-the-art and recent developments," *Omega*, vol. 35, no. 3, pp. 326–334, 2007.
- [23] U. Dorndorf, F. Jaehn, and E. Pesch, "Flight gate assignment and recovery strategies with stochastic arrival and departure times," *OR Spectr.*, vol. 39, no. 1, pp. 65–93, 2017.
- [24] A. Drexler and Y. Nikulin, "Multicriteria airport gate assignment and Pareto simulated annealing," *IIE Trans.*, vol. 40, no. 4, pp. 385–397, 2007.
- [25] J. Guépet, R. Acuna-Agost, O. Briant, and J. P. Gayon, "Exact and heuristic approaches to the airport stand allocation problem," *Eur. J. Oper. Res.*, vol. 246, no. 2, pp. 597–608, 2015.
- [26] A. Lim and F. Wang, "Robust airport gate assignment," in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, 2005, pp. 74–81.
- [27] T. Lust and J. Teghem, "Two-phase Pareto local search for the biobjective traveling salesman problem," *J. Heuristics*, vol. 16, no. 3, pp. 475–510, 2010.
- [28] T. Lust, J. Teghem, and D. Tuytens, "Very large-scale neighborhood search for solving multiobjective combinatorial optimization problems," in *Proc. Int. Conf. Evol. Multi Criterion Optim. (EMO 2011)*. Berlin, Germany: Springer, 2011, pp. 254–268.
- [29] R. S. Mangoubi and D. F. X. Mathaisel, "Optimizing gate assignments at airport terminals," *Transport. Sci.*, vol. 19, no. 2, pp. 173–188, 1985.
- [30] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.
- [31] Y. Nikulin and A. Drexler, "Theoretical aspects of multicriteria flight gate scheduling: Deterministic and fuzzy models," *J. Scheduling*, vol. 13, no. 3, pp. 261–280, 2010.
- [32] L. Paquete and T. Stützle, "A two-phase local search for the biobjective traveling salesman problem," in *Proc. Int. Conf. Evol. Multi Criterion Optim. (EMO 2003)*. Berlin, Germany: Springer, 2003, pp. 479–493.
- [33] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proc. Int. Conf. Principles Pract. Constraint Program.*, 1998, pp. 417–431.
- [34] J. Xu and T. G. Bailey, "The airport gate assignment problem: Mathematical model and a tabu search algorithm," in *Proc. 34th Annu. Hawaii Int. Conf. Syst. Sci.*, 2001, p. 10.
- [35] X. Xu and L. Xiong, "Cost analysis of flight delays and simulation in ground-holding model," *J. Nanjing Univ. Aeronaut. Astronaut.*, vol. 38, no. 1, pp. 115–120, 2006.
- [36] S. Yan and C. M. Huo, "Optimization of multiple objective gate assignments," *Transport. Res. A Policy Pract.*, vol. 35, no. 5, pp. 413–432, 2001.
- [37] C. Yu, D. Zhang, and H. Y. K. Lau, "An adaptive large neighborhood search heuristic for solving a robust gate assignment problem," *Expert Syst. Appl.*, vol. 84, pp. 143–154, Oct. 2017.
- [38] C. Yu, D. Zhang, and H. Y. K. Lau, "MIP-based heuristics for solving robust gate assignment problems," *Comput. Ind. Eng.*, vol. 93, pp. 171–191, Mar. 2016.
- [39] H.-H. Zhang, Q.-W. Xue, and Y. Jiang, "Multi-objective gate assignment based on robustness in hub airports," *Adv. Mech. Eng.*, vol. 9, no. 2, pp. 1–13, 2017.
- [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] 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, Nov. 1999.



Xinye Cai (M'10) received the B.Sc. degree in information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004, the M.Sc. degree in electronic engineering from the University of York, York, U.K., in 2006, and the Ph.D. degree in electrical engineering from Kansas State University, Manhattan, KS, USA, in 2009.

He is an Associate Professor with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NCAA), Nanjing, China. His current research interests include optimization, machine learning, and their applications. He is currently leading an Intelligent Optimization Research Group, NCAA.

Dr. Cai has won the Evolutionary Many-Objective Optimization Competition at the Congress of Evolutionary Computation 2017. He is an Associate Editor of *Swarm and Evolutionary Computation*.



Wenxue Sun received the master's degree in computer science from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018.

He is currently working as an Engineer with NetEase, Guangzhou, China. His research interests include multiobjective optimization and data mining.



Xiaoping Li (M'09–SM'12) received the B.Sc. and M.Sc. degrees in applied computer science from the Harbin University of Science and Technology, Harbin, China, in 1993 and 1999, respectively, and the Ph.D. degree in applied computer science from the Harbin Institute of Technology, Harbin, in 2002.

He is a Full Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. He has authored or coauthored over more than 100 academic papers, some of which have been published in international journals, such as the *IEEE TRANSACTIONS ON COMPUTERS*, *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, *IEEE TRANSACTIONS ON SERVICES COMPUTING*, *IEEE TRANSACTIONS ON CYBERNETICS*, *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING*, *IEEE TRANSACTIONS ON CLOUD COMPUTING*, *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, INFORMATION SCIENCES*, *Omega*, the *European Journal of Operational Research*, the *International Journal of Production Research*, *Expert Systems With Applications*, and the *Journal of Network and Computer Applications*. His research interests include scheduling in cloud computing, scheduling in cloud manufacturing, service computing, big data, and machine learning.



Mustafa Misir received the Ph.D. degree in computer science from Katholieke Universiteit Leuven, Leuven, Belgium, in 2012.

He worked as a Postdoctoral Researcher with INRIA Saclay, Université Paris Sud—XI, Palaiseau, France, Singapore Management University, Singapore, and the University of Freiburg, Freiburg im Breisgau, Germany. He was also a Visiting Researcher with Zurich University, Zürich, Switzerland, and the Universitat Politècnica de Catalunya, Barcelona, Spain. He was a Faculty

Member with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is a Faculty Member of computer engineering with Istinye University, Istanbul, Turkey. He published over 40 papers in various international conferences/journals. His main research interests include automated algorithm design (machine learning + algorithm design), data science, and operations research.

Dr. Misir is a recipient of several prestigious academic awards.



Tao Xu received the B.Sc. degree in computer engineering and the M.Sc. degree in computer science and application from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1982 and 1989, respectively.

He is a Professor with the College of Computer Science and Technology, Civil Aviation University of China, Tianjin, China. His current research interests include intelligence information processing, information system analysis, and their applications.



Kay Chen Tan (SM'08–F'14) received the B.Eng. (First Class Hons.) degree in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Dr. Tan is the Editor-in-Chief of the *IEEE TRANSACTIONS ON EVOLUTIONARY*

COMPUTATION, was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013, and currently serves as an Editorial Board Member of over ten journals. He is an Elected Member of the IEEE CIS AdCom from 2017 to 2019.



Zhun Fan (M'04–SM'10) received the B.S. and M.S. degrees in control engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2004.

He was an Assistant Professor, and then an Associate Professor with the Technical University of Denmark, Lyngby, Denmark. He is the Head of the Department of Electronic Engineering and a Full Professor with Shantou University, Shantou, China. He has also been working with the BEACON Center for Study of Evolution in Action, Michigan State University. His major research interests include intelligent control and robotic systems, robot vision and cognition, MEMS, computational intelligence, design automation, and optimization of mechatronic systems.