# Model and algorithm for resolving regional bus scheduling problems with fuzzy travel times

Ming Wei[a,b,*], Xuewu Chen[b], Bo Sun[a] and Yao-Yao Zhu[a]
[a]*School of Transportation, NANTONG University, Nantong, China*
[b]*Jiangsu Key Laboratory of Urban ITS, Southeast University, Nanjing, China*

**Abstract**. Regional bus scheduling is necessary to urban public transport that is complicated by the necessity of assigning trips that belong to several routes to buses located at different depots while reducing fleet size and operating costs. Considering the reality of emergencies that may interfere with the ability of vehicles to complete trips on time, it is reasonable to use fuzzy numbers to express uncertain delay times. Based on this idea, this paper proposes a chance-constrained programming model of regional bus scheduling that will reflect additional constraints such as the capacities of related depots and fueling needs. The objective of this paper is to maximize utilization of fleet vehicles. To overcome the defect of premature convergence in the particle swarm optimization algorithm (PSO), an improved PSO is proposed by using an organic fusion with group search optimization. Finally, an example demonstrates the correctness and effectiveness of the model and algorithm.

Keywords: Urban traffic, regional bus scheduling problem, fuzzy travel time, particle swarm optimization, group search optimization

## 1. Introduction

In contrast to traditional single-depot bus scheduling, the regional bus scheduling problem (RBSP) [5] aims to achieve a unified schedule by sharing vehicle resources across all routes. This is accomplished by inserting deadhead trips between depots to assign trips belonging to several routes to buses located at different depots. Due to the non-uniformity of departure frequency for all routes in a given period, RBSP can significantly reduce operating costs and improve the efficient utilization of vehicles. Bodin and Golden [1] have proven the RBSP is an NP-hard.

Most scholars study RBSP using the optimal scheduling theory, which is divided into network flow [3, 8, 9,

14, 19] and set partitioning [4, 11–13]. Most research into RBSP relies on established travel durations. However, conditions of real traffic environments, such as fluctuations in passenger loads at different time etc., cause deviations between the plan and actual operation, which make practical applications of RBSP difficult to analyze. In general, delays to trip completion are very difficult to predict in advance. Problems in which delay time is assumed to be an uncertain variable are RBSP with uncertain travel times (RBSPUTT). Research into RBSPUTT is rare [6, 7, 11–13]. Huisman and Albert studied RBSPUTT [6, 7] by assuming that future travel time is fixed or predicted in advance. Wei, et al. [11] studied a more realistic RBSPUTT with a stochastic travel time. Wei, et al. [12] proposed a multi-vehicle-type RBSPUTT with a grey travel time, as well as a bi-level programming model for uncertain regional bus scheduling that revealed the overall relationship between a scheduling plan and its procurement scheme

*Corresponding author. Ming Wei, School of Transportation, Nantong University, Nantong, 226019, China. Tel.: +86 1875 1304326; Fax: +86 0513 85012930; E-mail: oscar.weiming@gmail.com.

[13]. However, these models are typically difficult to solve in practice because of a lack of sufficient reliable historical data to determine the distribution function of the random variable or predict future growth trends.

This paper studies an RBSP with fuzzy travel times (RBSPFTT), which assumes that travel time is an uncertain variable. In combination with the rich experience and knowledge of scheduling staff, fuzzy numbers can represent this type of uncertain information when provided with capacity constraints and other practical factors. In this case, the triangular fuzzy number is used to express the actual travel time; it is then transformed to the corresponding chance-constrained programming model. A new hybrid algorithm based on the fuzzy simulation is proposed to solve the model. Finally, an actual example calculates a bus scheme to test the correctness and effectiveness of the model and algorithm.

## 2. Mathematical model

### 2.1. Mathematical model

- Each trip requires a bus to depart from its depot on time.
- When historical data and expertise are considered, it is possible to use triangular fuzzy numbers to represent certain time delays for each bus trip not completed on time.
- To ensure that a bus does not require refueling during daily operations, it should be fully fueled before or after the work day.

### 2.2. Decision variables

$T_d^k = \{x_1, x_2, \ldots, x_{|T_d^k|}\}$ denotes a sequence of trips accomplished in turn by each bus $\forall k \in V_d$, which begins and ends at same depot $\forall d \in D$.

### 2.3. Decision variables

$T = \{T_1, T_2, \ldots, T_{|T|}\}$ denotes a given set of trips belonging to several routes. Each trip $\forall T_i \in T$ requires uninterrupted driving from the starting point $sp_{T_i}$ at starting time $st_{T_i}$ to the ending point $dp_{T_i}$ at ending time $et_{T_i}$.

$D$ denotes a given set of depots. For each depot $\forall d \in D$, let $V_d$ denote a given set of buses initially located at the depot, and let $capacity_d$ denote the capacity of the depot.

Each vehicle $k$ ($\forall k \in V_d$, $\forall d \in D$) arrives at the trip $\forall x_i \in T_d^k$ starting point $sp_{x_i}$ at the time indicated by $t_{x_i}^k = et_{x_{i-1}} + \Delta T_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i})$. If $time(dp_{x_{i-1}}, sp_{x_i}) \neq 0$, this situation is called a Deadhead Trip (DH Trip). In this situation, $time(dp_{x_{i-1}}, sp_{x_i})$ denotes the time spent in traveling from the ending point of the trip $\forall x_{i-1} \in T_d^k$ to the starting point of the trip $\forall x_i \in T_d^k$. $\Delta T_{x_i}$ denotes the fuzzy delay time that prevents a vehicle from accomplishing a trip $x_i$ on time.

$a_0$ denotes the weight of number of the buses.

$a_1$ denotes the weight of waiting time.

$a_2$ denotes the weight of the deadhead time.

$Pos\{\bullet\}$ denotes the possibility of the case $\{\bullet\}$.

$|x|$ denotes the number of elements in the set $x$.

### 2.4. Objective function equation

$$\min \bar{f} \tag{1}$$

### 2.5. Constraint equations

$$\text{st. } V_p \cap V_q = \emptyset \ \forall p, q \in D \tag{2}$$

$$\underset{\forall d \in D, \forall k \in V_d}{\cup} T_d^k = T \tag{3}$$

$$T_p^u \cap T_q^v = \emptyset \ \forall u \in V_p, \forall v \in V_q \tag{4}$$

$$|V_d| \leq capacity_d \ \forall d \in D \tag{5}$$

$$Pos\{et_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i}) + \Delta T_{x_{i-1}}$$
$$\leq st_{x_i}\} \geq \alpha \ \forall d \in D, \forall k \in V_d, \forall x_i \in T_d^k \tag{6}$$

$$Pos\{f = a_0 \sum_{k \in D} |V_k| + \sum_{k \in D} \sum_{u \in V_k} \sum_{x_i \in T_k^u} [a_1.(st_{x_i} - t_{x_i}^u)$$
$$+ a_2.time(dp_{x_{i-1}}, sp_{x_i})] \geq \bar{f}\} \geq \beta \tag{7}$$

The objective function in Equation (1) aims to minimize the number of buses, the total waiting time, and the deadhead time. Equations (2–7) represent the constraints, as follows: Equation (2) ensures that each bus belongs to only one depot; Equation (3) ensures that all trips are successfully completed by all vehicles; Equation (4) expresses that each trip is assigned to exactly one vehicle, and a vehicle cannot simultaneously run two trips; Equation (5) ensures that no more than a given number of backup vehicles are dispatched from

each depot; Equation (6) ensures that the possibility of two trips normally accomplished in turn by the same bus in the random environment is at least $\alpha$; Equation (7) ensures that the confidence level of $\bar{f}$ representing the maximum value of $f$ is at least $\beta$.

## 3. Group particle swarm optimization algorithm for RBSPFTT solution

Particle Swarm Optimization (PSO) [17] is an evolutionary bionic algorithm that simulates flying birds, with advantages including smaller numbers of individuals and simple calculation. PSO performs well in multi-dimensional continuous space optimizations, and provides very rapid local convergence to a stable state. The group search optimization (GSO) [16], recently proposed by S. He and Q.H. Wu, is a novel algorithm based on the social behavior of hunting animals, such as lions and wolves. The GSO performs well in local searches, and can maintain the diversity of the solution to promote the evolution of the population. Each algorithm has advantages and disadvantages. Their homologous natures determine that they should easily integrate with one another; however, current studies involving both algorithms are rare [15].

This paper proposes a Group Particle Swarm Optimization (GPSO) based on a fuzzy simulation to solve RBSPFTT. According to the problem characteristics, the definite scheme solution process is described, and the following terms are defined: particle encoding scheme, constraints handling, and position and velocity correction equation.

### 3.1. Particle code scheme

In GPSO, each individual or potential solution in the swarm is called a particle. Every particle has its own position and velocity, where the position vector can be decoded as a solution to the problem, and the velocity vector is in the direction of particle movement.

The position vector $X = (x_1, x_2, \ldots, x_{|T|})$ is used to represent a potential solution of RBSPFTT [2, 10], wherein the $i$-th element is a trip $x_i$, which differs from other trips in the range $1 - |T|$. Furthermore, $V = (v_1, v_2, \ldots, v_{|T|})$ denotes the velocity vector, where the element $v_i$ is a real number ranging from $-|T| - |T|$. If $\forall x_{i-1}, x_i \in X$ meets constraint Equation (6), the bus will immediately execute trip $x_i$ after it has accomplished trip $x_{i-1}$; otherwise, the two trips would be executed by different buses. Hence, each individual can

be decoded into a sequence of trips accomplished by different buses according to Equation (6).

As mentioned above, any individual $X$ satisfies constraints Equations (2–5). Therefore, if the individual $X$ complies with constraints Equations (5–7), it is a feasible solution.

### 3.2. Fitness function

The fitness value, determined by the fitness function, is the standard according to which the merit of each particle position is evaluated. Each particle can be decoded into a set of trips finished by different vehicles. In this paper, Equation (1) is used as the fitness function to assess the quality of any individual. Because all individuals meet constraints Equations (2–4), constraint Equation (5) can be regarded as a penalty function to generate a new fitness function.

$$\min \ \bar{f} + A \sum_{\forall d \in D} Max(|V_d| - capacity_d, 0) \qquad (8)$$

where $A$, which is set of a sufficiently large number, denotes the penalty factor. The fitness function, expressed by Equation (8), ensures that an infeasible solution confers great fitness value and will be eliminated in the iteration.

### 3.3. Fuzzy objective function and handling constraints

Constraints Equations (6) and (7) with fuzzy parameters also determine the objective function Equation (8). Because the computer does not deal directly with fuzzy parameters, the fuzzy simulation technology proposed by Wu, et al. [18] can be used to examine its objective function and constraint with respective confidence levels of $\alpha$ and $\beta$.

According to the definition of fuzzy number arithmetic for vector $\Delta T_{x_i}$ of a fuzzy delay time in a given decision vector $X$ the establishment of Equation (6) holds if and only if there exists a clear vector $\Delta T_{x_i}^0$ in the vector $\Delta T_{x_i}$ that satisfies $et_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i}) + \Delta T_{x_i}^0 \leq st_{x_i}$ and $\mu(\Delta T_{x_i}^0) \geq a$, where $\mu(\bullet)$ is the membership function of the fuzzy set. That is to say, if $et_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i}) + \Delta T_{x_i}^0 \leq st_{x_i}$, which uniformly generates vector $\Delta T_{x_i}^0$ from fuzzy vector $\Delta T_{x_i}$ in order to satisfy $\mu(\Delta T_{x_i}^0) \geq a$, then Equation (6) is established. After a given number $N$ of iterations, if vector $\Delta T_{x_i}^0$ does not satisfy $et_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i}) + \Delta T_{x_i}^0 \leq st_{x_i}$, then the decision vector $X$ is considered to be impossible.

As mentioned above, the specific procedure of fuzzy Equation (6) is as follows:

**Step 1:** Uniformly generate vector $\Delta T_{x_i}^0$ from fuzzy vector $\Delta T_{x_i}$ at a horizontal cross-sectional level $\alpha$;

**Step 2:** If $et_{x_{i-1}} + time(dp_{x_{i-1}}, sp_{x_i}) + \Delta T_{x_i}^0 \le st_{x_i}$, then the output is "feasible";

**Step 3:** Repeat step 1 and 2 $N$ times;

**Step 4:** Output "infeasible".

Similarly, the specific procedure of fuzzy Equation (7) is as follows:

**Step 1:** Set $\bar{f} = -\infty$;

**Step 2:** Uniformly generate vector $\Delta T_{x_i}^0$ from fuzzy vector $\Delta T_{x_i}$ at a horizontal cross-sectional level $\beta$;

**Step 3:** If $\bar{f} \le f$, then $\bar{f} = f$;

**Step 4:** Repeat step 2 and 3 $N$ times to calculate $\bar{f}$;

**Step 5:** Output $\bar{f}$.

### 3.4. Formula for modifying position and updating velocity

In PSO, each particle updates itself by tracking the personal extreme value $f_i^*(t)$ and global extreme value $f_g^*(t)$ in every iteration. When all particles gather to the same location (that of the current best particle), its velocity approaches zero due to inertia, personal awareness and social consciousness. Each particle will then refuse to search the solution space in other regions.

To address the inadequacies of PSO, a concept derived from GSO is embedded in the search procedure for GPSO; this concept involves a random movement by a small number of rogue particles, wherein each particle generates a random head angle and distance and then moves to this new position. The number of randomly wandering particles is related to the diversity of the population. According to the process described above, the particle updates its velocity and position formula as follows:

$$V_i(t + 1) = w \cdot V_i(t) + c_1 \cdot rand \cdot (X_i^*(t) - X_i(t))$$
$$+ c_2 \cdot rand \cdot (X_g^*(t) - X_i(t)) \qquad (9)$$

$$X_i(t + 1) = \begin{cases} X_i(t) + V_i(t + 1), \, othes \\ X_i(t) + l_i D_i^t(\varphi^{t+1}), \, rand > \sigma \end{cases} \qquad (10)$$

where $w$, $c_1$ and $c_2$ denote the inertia factor, weight, and global optimal value, respectively; $l_i \in [0, l_{max}]$, $\varphi^{t+1} \in [0, \theta_{max}]$ and $D_i^t(\varphi^{t+1})$ denote a random distance, head angle and the vector of the angle, respectively; and *rand* denotes a random number in the range [0, 1].

**Definition 1.** $\sigma = \frac{1}{N(N-1)} \sum_{i \neq j, i, j=1}^{N} c_{ij}$ denotes the population's diversity where $c_{ij} = \begin{cases} 1, \, f_i \neq f_j \\ 0, \, otherwise \end{cases}$.

### 3.5. Algorithm flow

The detailed procedure for GPSO solving RBSPFTT is described as follows.

**Step 1:** Set the parameters of the algorithm.

**Step 2:** Let t = 0 and initialize the first generation of particle swarm.

**Step 2.1:** For each particle, randomly generate its position and velocity vector $X_i(0)$ and $V_i(0)$, and calculate its fitness value $f_i(0)$.

**Step 2.2:** If $X_i^*(0) = X_i(0)$, then $f_i^*(0) = f_i(0)$ to determine the extreme value of each particle.

**Step 2.3:** Find the global optimal particle $f_g^*(0) = \min\{f_1^*(0), \ldots \ldots, f_n^*(0)\}$.

**Step 3:** Let t = t+1, and update the current status of the *t*-th generation swarm.

**Step 3.1:** For each particle, update its position vector $X_i(t)$ and velocity vector $V_i(t)$, according to Equations (9) and (10).

**Step 3.2:** Recalculate the fitness value $f_i(t)$. If $f_i(t) < f_i^*(t)$, let $f_i^*(t) = f_i(t)$ and $f_i^*(t) = f_i(t)$. If $f_i(t) < f_g^*(t)$, let $f_g^*(t) = f_i(t)$ and $f_g^*(t) = f_i(t)$.

**Step 4:** If the termination condition is satisfied, output $f_g^*(t)$; otherwise return to step 3.

## 4. Analysis of computational examples

A regional bus network is composed of three depots ($capacity_d = 15$, $d = 1, 2, 3$) and five routes. Tables 1–4 depict details of the test instance as follows: Tables 1 and 2 demonstrate the basic information and the fixed schedule of these lines; Table 3 depicts the deadhead time information; and Table 4 depicts the fuzzy delay time between two trips. Using a unified model for bus scheduling, a bus company will assign a certain number of vehicles to the three depots to complete a total of 210 trips while minimizing operation costs.

In order to verify the validity of the model and algorithms, the empirical parameters of GPSO are set as follows: the maximum number of iterations is 200, and the number of particles is 50. Let $w = 0.75$, $c_1 = c_2 = 1.5$, $l_{max} = 5$, $\theta_{max} = 45o$, $a_0 = 500$, $a_1 = 1$, and $a_2 = 2.5$. Matlab was used to program GPSO to solve the

Table 1
Route information

| Route | Period | Departure time (min) | Start depot | End depot | Run time (min) |
|---|---|---|---|---|---|
| 1 | 7:00–20:30 | 15 | A | B | 60 |
| 2 | 6:40–20:40 | 20 | B | A | 65 |
| 3 | 6:30–21:00 | 30 | A | C | 55 |
| 4 | 7:00–20:00 | 20 | C | A | 60 |
| 5 | 6:45–20:25 | 20 | B | C | 65 |

Table 2
Timetable of route information

| Trip | Departure time | Route | Trip | Departure time | Route |
|---|---|---|---|---|---|
| 1 | 7:00 | 1 | … | … | 3 |
| … | … | 1 | 128 | 21:00 | 3 |
| 55 | 20:30 | 1 | 129 | 7:00 | 4 |
| 56 | 6:40 | 2 | … | … | 4 |
| … | … | 2 | 168 | 20:00 | 4 |
| 98 | 20:40 | 2 | 169 | 6:45 | 5 |
| 99 | 6:30 | 3 | … | … | 5 |
| 100 | 7:00 | 3 | 210 | 20:25 | 5 |

Table 3
Deadhead time information

| Start depot | End depot | Deadhead time (min) | Start depot | End depot | Deadhead time (min) |
|---|---|---|---|---|---|
| A | B | 20 | C | A | 20 |
| B | A | 25 | B | C | 25 |
| A | C | 15 | C | B | 20 |

Table 4
Some fuzzy delay time between trips

| Trip | Delay time (min) | Trip | Delay time (min) | Trip | Delay time (min) |
|---|---|---|---|---|---|
| 8 | (23,29,33) | 118 | (27,32,34) | 159 | (25,27,37) |
| 21 | (24,28,35) | 120 | (21,30,36) | 162 | (23,30,36) |
| 24 | (30,36,39) | 124 | (26,31,37) | 163 | (24,28,37) |
| 31 | (21,22,26) | 152 | (31,32,38) | 166 | (20,32,39) |
| 42 | (20,26,29) | 153 | (22,25,34) | 171 | (20,24,25) |
| 92 | (30,32,35) | 155 | (26,28,33) | 193 | (25,30,37) |
| 99 | (29,33,35) | 158 | (37,39,40) | 208 | (21,31,38) |

RBSPFTT, and the simulation as repeated 50 times at confidence levels $\alpha = 1.0$ and $\beta = 0.9$ on the same computer in order to avoid accidental phenomena. Table 5 indicates the simulation results. There are a total of five solutions. The optimal solution of $f$ is equal to 19,950, the mean is equal to 19,964, and the variance is equal to 28.6. The best solution can be identified at 62% probability, a suboptimal solution can be identified at 22% probability, and the worst solution can be identified at 4% probability. Thus, GPSO can find a satisfactory solution in most cases.

Table 5
Results of 50 simulations

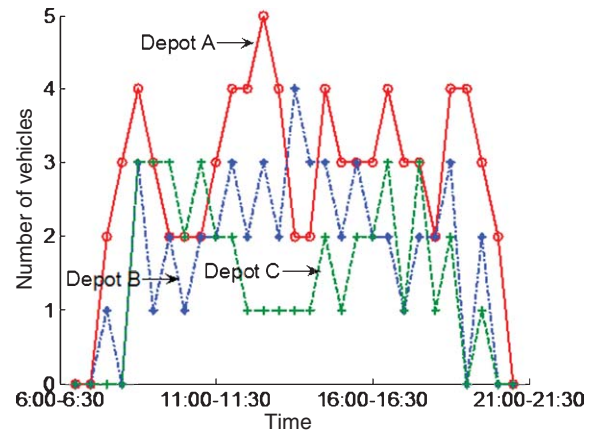| Solution type | Objective function | Waiting time (min) | Deadhead time(min) | Probability |
|---|---|---|---|---|
| Best solution | 19,950 | 4100 | 1740 | 62% |
| Suboptimal solution | 19,975 | 4090 | 1765 | 22% |
| Worst solution | 20,015 | 4065 | 1785 | 4% |



Fig. 1. Number of vehicles in each parking yard between 06:30–21:00.

Table 6 demonstrates the best schedule for the regional buses, and Fig. 1 shows the number of vehicles in each parking yard. Each bus follows a predefined trip sequence and is allowed to return to the starting point of the initial trip. That is to say, if a given trip must be assigned to a vehicle, the vehicle must arrive at a fixed time in advance of its starting time. In order to cover all of the trips in a fixed bus time table, 23 buses are required to execute a total of 85 deadhead trips, wherein:

– 9, 9, and 5 vehicles, respectively, are initially located at depots A, B, and C.
– Deadhead trips occur 23, 6, 10, 4, 21, and 21 times between two depots.
– The total working time for all buses is equal to 12,445 min. Their waiting and deadhead times are equal to 4,100 min. and 1,740 min., respectively.

The influence of the two parameters $\alpha$ and $\beta$ on bus scheduling was also analyzed to study the effect of the fuzzy travel time feature on $f$. As parameter $\alpha$ decreases and parameter $\beta$ increases, the number of pairs of feasible trips completed by a vehicle decreases; this leads to a further reduction in the feasible solution space consisting of these pairs, and the sub-solution space containing

Table 6
Results of bus arrangements

| No. | Sequence case of each bus completing the trip | Time (min) | | |
|---|---|---|---|---|
| | | Traveling | Waiting | Deadhead |
| 1 | A-6-64-DH-68-21-191-154-40-202-166-A | 540 | 137 | 20 |
| 2 | C-134-DH-179-DH-187-DH-80-37-200-DH-123-54-DH-C | 485 | 170 | 125 |
| 3 | B-58-DH-135-14-70-26-DH-32-196-161-124-DH-B | 530 | 185 | 60 |
| 4 | B-172-136-16-186-150-DH-195-160-48-DH-55-B | 555 | 225 | 45 |
| 5 | B-57-102-140-20-76-115-DH-36-201-165-126-DH-B | 580 | 235 | 40 |
| 6 | C-131-12-DH-19-188-152-119-DH-122-52-DH-C | 475 | 213 | 90 |
| 7 | A-3-63-106-141-23-83-41-203-DH-96-128-DH-A | 580 | 265 | 40 |
| 8 | C-132-104-DH-67-110-DH-114-DH-86-44-206-C | 460 | 210 | 60 |
| 9 | B-171-DH-177-143-112-DH-30-82-38-89-51-98-DH-B | 590 | 197 | 60 |
| 10 | A-101-DH-176-DH-15-185-148-31-197-DH-46-DH-167-A | 550 | 133 | 85 |
| 11 | A-2-DH-10-181-145-DH-77-118-DH-45-207-DH-A | 480 | 219 | 65 |
| 12 | A-5-65-108-DH-73-28-81-DH-157-43-DH-50-DH-A | 520 | 155 | 85 |
| 13 | C-130-7-66-18-74-DH-78-116-DH-85-DH-204-DH-209-C | 585 | 165 | 80 |
| 14 | B-59-9-DH-107-144-DH-189-156-DH-88-DH-94-DH-B | 465 | 205 | 105 |
| 15 | B-56-DH-60-11-69-111-149-34-199-DH-92-DH-B | 520 | 215 | 60 |
| 16 | B-170-DH-62-DH-180-DH-22-190-153-39-90-49-210-DH-B | 610 | 171 | 80 |
| 17 | A-1-61-DH-178-DH-71-DH-75-29-DH-117-159-47-208-DH-A | 590 | 128 | 105 |
| 18 | A-100-DH-103-137-DH-183-DH-25-192-155-121-DH-205-168-A | 600 | 132 | 80 |
| 19 | A-4-175-139-109-DH-27-DH-33-84-42-93-53-DH-A | 590 | 134 | 70 |
| 20 | B-173-138-17-72-113-DH-193-158-DH-164-DH-97-DH-B | 535 | 132 | 75 |
| 21 | C-129-DH-133-105-DH-182-146-DH-79-35-87-DH-91-DH-95-DH-C | 580 | 140 | 110 |
| 22 | B-169-DH-174-DH-13-184-147-DH-151-DH-198-162-125-DH-B | 555 | 161 | 95 |
| 23 | A-99-DH-8-DH-142-24-194-DH-120-163-127-DH-A | 470 | 175 | 85 |
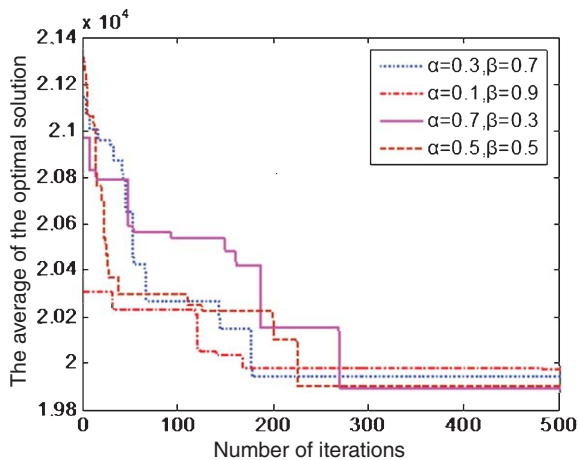


Fig. 2. Influence of parameters $\alpha$ and $\beta$ on bus scheduling.
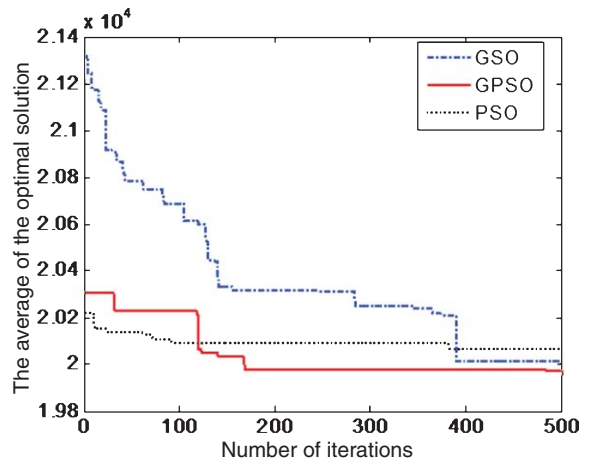


Fig. 3. Comparative analysis of three kinds of algorithms.

better solutions is abandoned. Ultimately, the cost of the scheme increases with less uncertainty. However, the scheme is less disturbed by emergencies, and less cost change occurs in the process of a real-time adjustment scheme. The results shown in Fig. 2 are consistent with visual analysis.

In order to verify the validity of the improved algorithms, a case study was calculated 50 times each with GPSO, PSO, and GSO. Figure 3 comparatively analyzes the average simulation results, with the conclusions as follows:

– GPSO, PSO, and GSO require 210, 95, and 390 iterations, respectively, to converge to their optimal solutions. The differences in optimal solutions among them are 2.1%, 2.5%, and 4.6%, respectively. This indicates that GPSO performs better than PSO and GSO in convergence.
– PSO obtains the optimal solution faster than GSO, but it easily converges to local optima. Hence, GPSO, which combines the advantages of the other

two algorithms, can perform thermal simulations effectively. By judging the degree of prematurity to maintain swarm diversity, GPSO can not only yield a suboptimal solution quickly, but also jump from local optimization solutions by controlling the search space of particles. This indicates that GPSO performs more robustly than PSO and GSO.

## 5. Conclusions

The primary contribution of this paper is to propose a chance-constrained programming model for regional bus scheduling with fuzzy travel times. Compared to existing studies on RBSPUTT with random or grey travel times, the proposed model can avoid the defect of requiring a large amount of traffic data to determine the distribution function of uncertain travel times, making it easier to optimally solve in practice. The test case analyzed the influence of the fuzzy feature of travel time on bus scheduling, and results indicate that the scheme with less fuzziness of travel time is less disturbed by emergencies and has less cost change in a real-time adjustment scheme.

This paper also contributes to the design of a new hybrid GPSO algorithm, which combines the advantages of both PSO and GSO by embedding the concept of a small portion of particles to take a random walk out of the group into GSO, as occurs in the search procedure for PSO. The simulation indicates that, compared to PSO and GSO in convergence and robustness of the algorithm, GPSO can not only converge to a suboptimal solution quickly, but also jump from a local optimization solution by controlling the search space of particles.

## Acknowledgments

## References

[1] A.A. Bertossi and P. Carraresi, On some matching problems arising in vehicle scheduling models, *Networks* **17** (1987), 271–281.

[2] A.A.R. Hosseinabadi, et al. GELS-GA: Hybrid metaheuristic algorithm for solving multiple traveling salesman problem, *The 14th IEEE ISDA*, 2014, pp. 76–81.

[3] A. Haghani and M. Banihashemi, Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints, *Transportation Research* **36** (2002), 309–333.

[4] C.C. Ribeiro and F.A. Soumis, Column generation approach to the multiple-depot vehicle scheduling problem, *Operations Research* **42** (1994), 41–52.

[5] D.Y. Wang, X.Y. Zang and H.X. Wang, Research on situation and developing trend for optimization theory and method of public transit vehicle scheduling problem, *Journal of Beijing Jiaotong University* **32**(3) (2008), 42–47.

[6] D. Huisman, R. Freling and A.P.M. Wagelmans, A robust solution approach to the dynamic vehicle and crew scheduling problem, *Transportation Science* **38** (2004), 447–458.

[7] D. Huisman, R. Freling and A.P.M. Wagelmans, A solution approach for dynamic vehicle and crew scheduling, *European Journal of Operational Research* **172** (2006), 453–471.

[8] H.X. Wang and J.S. Shen, Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints, *Applied Mathematics and Computation* **190** (2007), 1237–1249.

[9] J.Q. Li, P.B. Mirchandani and D.A. Borenstein, Lagrangian heuristic for the real-time vehicle rescheduling problem, *Transportation Research Part E* **23** (2009), 419–433.

[10] J.J. He and K. Zhang, A hybrid distribution algorithm based on membrane computing for solving the multi-objective multiple traveling salesman problem, *Fundamental Information* **136**(3) (2015), 199–208.

[11] M. Wei, W.Z. Jin and B. Sun, Model and algorithm for regional bus scheduling with stochastic travel time, *Journal of Highway and Transportation Research and Development* **28**(10) (2011), 151–156.

[12] M. Wei, W.Z. Jin and B. Sun, Model and algorithm for regional bus scheduling with grey travel time, *Journal of Transportation Systems Engineering and Information Technology* **12**(0) (2012), 134–151.

[13] M. Wei, B. Sun and W.Z. Jin, Bi-level programming model for uncertain regional bus scheduling problem, *Journal of Transportation Systems Engineering and Information Technology* **12**(2) (2013), 1–8.

[14] N. Kliewer, T. Mellouli and L.A. Suhl, Time space network based exact optimization model for multi-depot bus scheduling, *Eurpoean Journal of Operational Research* **175** (2006), 1616–1627.

[15] Q.D. Wu, Q. Kang, L. Wang and J.S. Lu, *An Introduction to Natural Computation*, Shanghai Science and Technology Press, Shanghai, 2001.

[16] S. He, H. Wu and J.R. Saunders, Group search optimizer: An optimization algorithm inspired by animal searching behavior, *IEEE Transactions on Evolutionary Computation* **33**(5) (2009), 973–1001.

[17] X.F. Xie, W.J. Zhang and Z.L. Yang, Overview of parti-cle swarm optimization, *Control and Decision* **18**(2) (2003), 129–135.

[18] Y.K. Liu and S.M. Wang, *Fuzzy Stochastic Optimization The-ory*, China Agricultural University Press, Beijing, 2006.

[19] Z.G. Liu and J.S. Shen, Regional bus operation bi-level programming model integrating timetabling and vehicle scheduling, *Systems Engineering Theory & Practice* **11** (2007), 135–141.