

Article

Human Resource Scheduling Model and Algorithm with Time Windows and Multi-Skill Constraints

Zhiping Zuo ¹, Yanhui Li ^{1,3}, Jing Fu ^{2,4,*} and Jianlin Wu ³

¹ School of Management, Wuhan College, Wuhan 430212, China

² School of Business Administration, Wuhan Business University, Wuhan 430056, China

³ School of Information Management, Central China Normal University, Wuhan 430079, China

⁴ Institute of Agricultural Economy and Technology, Hubei Academy of Agricultural Sciences, Wuhan 430064, China

* Correspondence: 20150384@wbu.edu.cn

Received: 12 June 2019; Accepted: 1 July 2019; Published: 4 July 2019



Abstract: In situations where an organization has limited human resources and a lack of multi-skilled employees, organizations pay more and more attention to cost control and personnel arrangements. Based on the consideration of the service personnel scheduling as well as the routing arrangement, service personnel of different skills were divided into different types according to their multiple skills. A mathematical programming model was developed to reduce the actual cost of organization. Then, a hybrid meta heuristic that combines a tabu search algorithm with a simulated annealing was designed to solve the problem. This meta heuristic employs several neighborhood search operators and integrates the advantages of both the tabu search algorithm and the simulated annealing algorithm. Finally, the stability and validity of the algorithm were validated by the tests of several kinds of examples.

Keywords: scheduling problem; human resource; multi-skill; hybrid meta heuristic

1. Introduction

The world's demographic changes have had a significant impact on the economy and business operations. The aging of the population leads to a decrease in the working-age population [1]. On the other hand, due to increasing adult education levels, the quality of laborers has undergone a fundamental change [2]. The "double change" of the quantity and quality of the working-age population leads to the continuous rise of the cost of labor for enterprises, which has formed tremendous pressure on the operation and management of enterprises. Technicians with multiple skills can adapt to a variety of positions and jobs, which can make the organization's personnel arrangement more flexible. Its "a professional with multiple skills" characteristics can also effectively reduce the labor cost of the organization. Therefore, in the management practice of contemporary enterprises, multi-skilled employees are increasingly favored by managers. However, the work arrangements for multi-skilled employees often involve multiple tasks in multiple locations, and each task has its own service time window limit. Therefore, how to arrange the tasks of multiple multi-skilled employees reasonably has become an urgent problem to be solved in human resource management and personnel scheduling or dispatching.

The human resource path problem with time windows and multi-skill constraints is an extension of the classic vehicle routing problem (VRP). Under the premise of meeting the needs of the task, the goal is to reasonably schedule numbers of multi-skilled employees, minimize the operational costs, and improve the operational efficiency of the organization.

VRP is a classical combinatorial optimization problem. The development of VRP, based on the paper written by Dantzig and Ramser [3], was published in 1959. It had extensive theoretical support and practical basis in the research of logistics and supply chain management. Laporte had reviewed the development of the vehicle routing problem in the past 50 years [4]. With the development of VRP, the scope of application of VRP was also constantly developing and expanding. Braekers et al. scientifically classified the research of VRP [5]. According to its classification, the problems studied in this paper could be classified as an extension of the vehicle routing problem with time windows (VRPTW). Many scholars did a lot of research on VRPTW. Savelsbergh proved for the first time that it is an NP-hard problem and designed a local search algorithm to solve it [6]. Bräysy and Gendreau [7] combed the study of VRPTW and introduced the main heuristic algorithms and classical operators for solving this problem, which had provided learning resources for later researchers.

In the early days, VRPTW was named as a special scheduling problem [8], which took into account the need for task time in the scheduling problem (that is the time window of a single point) and the shortest path in the path problem, making the problem more practically significant and gradually becoming a research hotspot in recent years. Among them, Qin et al. [9] and Zhang et al. [10] used an improved heuristic algorithm to analyze and solve the problem and conducted a preliminary exploration of the vehicle routing problem with human resource constraints, but only considered a single type of human resource. The study of scheduling problems with multi-skill constraints first appeared in the multi-skill project scheduling problem (MSPSP). In the subsequent development, Correia and Saldanha-da-Gama carried out an empirical study on the cost factors [11]. Bellenguez and Néron were the first to prove that such problems were NP-hard problems and calculated the lower bound of the problem [12]. Intelligent algorithms, such as the differential evolution algorithm (DE) [13] and the immune genetic algorithm (IGA) [14], were widely used to solve this problem. However, this kind of research mainly considered project work or production systems at the same location [15–17] and did not consider the routing optimization.

In summary, scholars have already done a lot of research on VRPTW and MSPSP, but few studies considered both sides in an integration. According to the actual management practice requirements, this paper considered the above two factors and models and analyzed the vehicle routing problem with multi-skill constraints and time windows (VRPMCTW). According to the characteristics of the model, a heuristic algorithm based on a tabu search was designed to solve the problem as a whole and achieved good results.

2. Problem Description and Model Construction

2.1. Problem Description

In the regional logistics service activities, the logistics service center needs to manage a variety of services such as water supply and drainage, strong electricity, weak electricity (access control, surveillance video, network routing), and home maintenance of buildings in the jurisdiction. Each business is highly professional, and thus technical personnel with professional skills are required to be competent for business processing work, and the geographical distribution of multiple buildings is generally scattered. During the same period, different buildings have different service requests, which constitutes a complex service network.

This paper takes a service network with multi-skill constraints as the research object. The network consists of a service center and multiple building nodes, as shown in Figure 1. The service center is the point numbered 0 in the figure. It dispatches a group of professional technicians to serve the buildings (task points) with different service requirements. Each professional technician serves a path, starting from the service center and returning to it after completing the task. The service request proposed by each task point has time window constraints and different skill requirements (different skill requirements are represented by different shapes in Figure 1).

2.3. Mathematical Model

The problem studied in this paper is an expansion based on the classic VRPTW, and similar to the general VRPTW model, the objective function is the minimum operating cost of the system. In terms of the cost structure, in addition to the path cost considered by the VRPTW, it also includes the salary cost of the personnel. The skill limitations of skills of the professional technicians also need to be considered in the constraints. Therefore, the problem has the following mathematical programming model:

$$\min \sum_{k \in K} \sum_{(i,j) \in E} \sum_{r \in R_k} d_{ij} X_{ijkr} + \sum_{k \in K} \sum_{r \in R_k} f_k \tag{1}$$

$$s.t. : \sum_{k \in K} \sum_{j \in V} \sum_{r \in R_k} X_{ijkr} = 1, \forall i \in N \tag{2}$$

$$\sum_{i \in V} X_{0ikr} = \sum_{i \in V} X_{i0kr} = 1, \forall k \in K, r \in R_k \tag{3}$$

$$\sum_{j \in V} X_{ijkr} = \sum_{j \in V} X_{jikr}, \forall i \in V, k \in K, r \in R_k \tag{4}$$

$$O_i \leq w_{ikr} \leq C_i, \forall i \in N, k \in K, r \in R_k \tag{5}$$

$$w_{ikr} + s_i + t_{ij} - w_{jkr} \leq (1 - X_{ijkr})M, \forall (i, j) \in E, k \in K, r \in R_k \tag{6}$$

$$\sum_{i \in V} X_{imkr} = \sum_{j \in V} X_{mjkr} \leq st_{km}, \forall k \in K, r \in R_k, m \in N \tag{7}$$

$$X_{ijkr} \in \{0, 1\}, \forall i \in V, j \in V, k \in K, r \in R_k. \tag{8}$$

The objective function (1) minimizes the total cost of the service system operation; the constraint (2) ensures that each task point can only be accessed by one path; the constraints (3) and (4) guarantee that the entry of each path at each task point is equal to the degree of outflow, that is, to maintain flow balance; the constraint condition (5) is the time window constraint; the constraint condition (6) prevents the path from generating the sub-ring; the constraint condition (7) is the personnel skill constraint to ensure that each selected task point must meet the demand for professional skills of that point; the constraint (8) defines the decision variable X_{ijkr} as a 0-1 variable.

3. Algorithm Design

VRPTW has been proved to be an NP-hard problem [6]. The problem studied in this paper is an extension of VRPTW, so it is also a NP-hard problem. There is no ordinary analytical algorithm to find its optimal solution in a reasonable time. It is noted that some heuristic algorithms and intelligent algorithms have achieved good results in solving VRP. This paper intends to propose a heuristic algorithm to solve this problem.

Both tabu search (TS) [18] and simulated annealing (SA) [19] algorithms are both efficient meta-heuristic algorithms, which have been also well applied in solving various NP-hard problems and obtained good results. TS uses the memory-guided algorithm of tabu step to jump out of the local optimal, so to improve the global optimization ability of the algorithm. It usually has a very good performance in solving large complex VRPs [7]. SA accepts a bad solution with a certain probability and jumps out of the local optimal, and then selects a high-quality solution for global search through a certain selection strategy. The algorithm produces different neighborhood structures through a variety of operators, greatly enriching the structural diversity of solutions, so as to improve the overall optimization ability of the algorithm. In this paper, TS is the basic algorithm framework, and the two algorithms of TS and SA are effectively fused by the remove-random operation, and the advantages of

the two algorithms are fully utilized to solve the problem, which effectively improves the efficiency of the algorithm named as the hybrid simulated annealing tabu search (HSATS).

3.1. Path Coding

The coding method of the solution is one of the keys to solve the problem by the heuristic algorithm, which directly affects the efficiency of the algorithm. This paper uses the natural number encoding method commonly used in solving VRP [20,21]. The solution S is coded as a natural number as shown in Equation (9), where S represents the current solution; R_{ij} is the service path, representing the j -th path serviced by the i -th professional technician; U represents the set of points that are not visited:

$$S = \{R_{11}, R_{21}, \dots, R_{i1}, R_{12}, R_{22}, \dots, R_{i2}, \dots, R_{1j}, R_{2j}, R_{ij}, U\}, (j = 2^k - 1). \tag{9}$$

Take the situation shown in Figure 1 as an example, where there are three skill requirements, $k = 3$, then the corresponding professional technicians can be divided into the following seven types as shown in Table 1:

Table 1. Skill types and their needs.

Personnel Type	Mastered Skills
1	A
2	B
3	C
4	A,B
5	B,C
6	A,C
7	A,B,C

Then, one of the feasible solutions in the problem of Figure 1 can be expressed as:

$S = \{R_{11}, R_{41}, R_{61}, R_{71}, U\}$, among them, $R_{11} = \{0, 14, 5, 17, 0\}$; $R_{41} = \{0, 4, 8, 9, 6, 0\}$; $R_{61} = \{0, 2, 3, 15, 18, 0\}$; $R_{71} = \{0, 1, 19, 12, 11, 10, 13, 0\}$; $U = \{7, 16\}$.

3.2. Construction of the Initial Path

This paper first sets a certain number of empty paths $\{0,0\}$, and then uses the Solomon insertion and improved post-processing (IPP) to construct the initial path by iterating all the points into the path.

The basic idea of the Solomon insertion algorithm is to judge the advantages and disadvantages of the insertion position of each point by calculating the evaluation value of the insertion point, to insert the point in the position of higher evaluation value, and to produce the initial solution, which has been widely used in VRP. The Solomon insertion algorithm can be used to generate a series of reasonable routes. The quality of the solution has been greatly improved compared with the simple stochastic, but there is still a lot of room for improvement. Therefore, on the basis of post-processing proposed by Hu [22], this paper improves and designs IPP: (1) add 2-opt* operator; (2) delete the 0-relocate operator procedure.

The operation process of each operator is shown in Figures 2–4 as follows.

The process is as follows ($f(S)$ represents the path length of the problem):

- Step 1 Use 2-relocate, 2-opt*, and 2-exchange operators to exchange between paths to generate a new solution S' , and if $f(S') < f(S)$, accept S' ; otherwise, keep the original solution unchanged;
- Step 2 Use the 1-relocate, or-opt, 2-opt, and 1-exchange operators to exchange in the same path to generate a new solution S' , and if $f(S') < f(S)$, accept S' ; otherwise, keep the last result;
- Step 3 Use the 0-exchange operator to exchange the unserved points with the points in the path to obtain a new solution S' , and if $f(S') < f(S)$, accept S' ; otherwise, keep the last result;

Step 4 If the quality of the solution is not improved after the above three steps, the operation will jump out; otherwise, jump to Step 1.

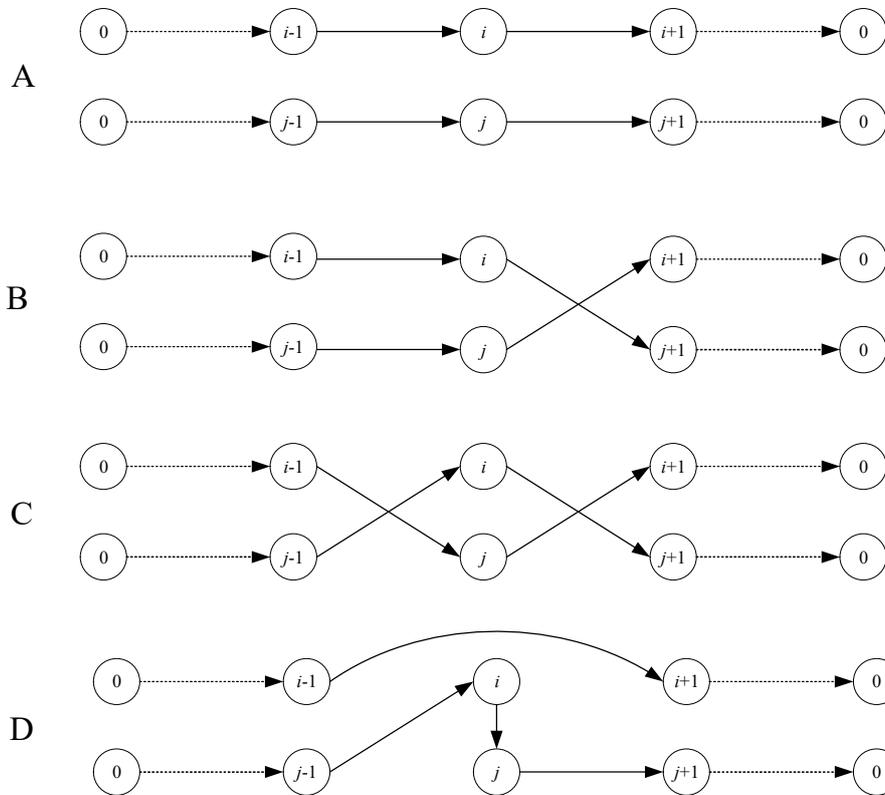


Figure 2. A is the initial solution, B is the result after 2-opt*, C is the result after 2-exchange, D is the result after 2-relocate.

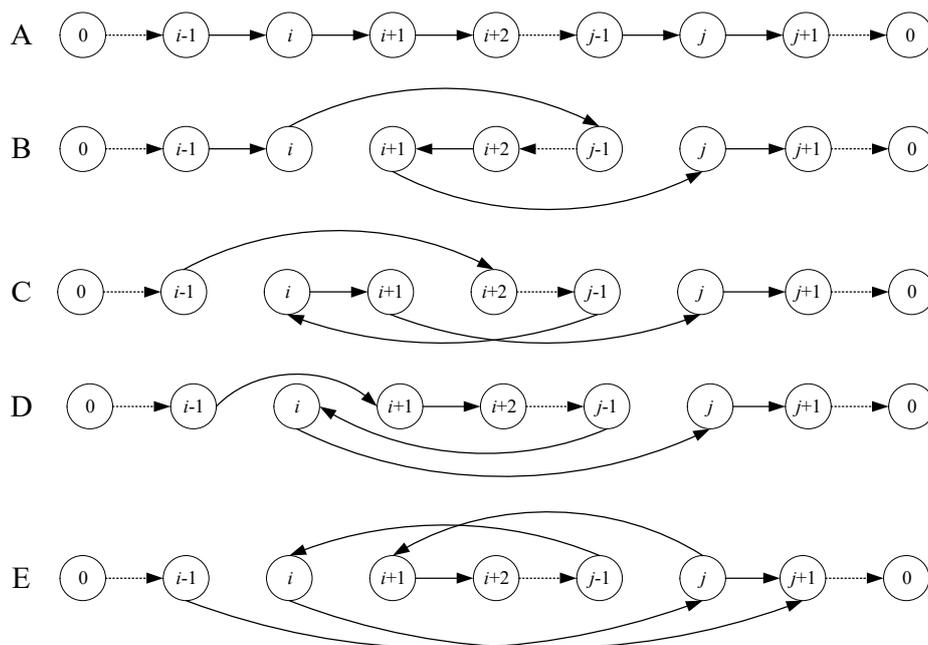


Figure 3. A is the initial solution, B is the result of 2-opt, C is the result of or-opt, D is the result of 1-relocate, E is the result of 1-exchange.

Combined with the above Solomon insertion and IPP, the construction steps of the initial solution are:

- Step 1 Generate n empty paths $\{0, 0\}$, and set the service personnel type to $K = 2^k - 1$ (k is the total skill type, K is the service personnel type);
- Step 2 Traverse all the points and u insert the points into the appropriate positions using the Solomon insert algorithm;
- Step 3 Optimize the path using the IPP for the current solution;
- Step 4 Determine whether all points are inserted. If all points are served, end the iteration and output the current solution; otherwise, return to Step 2.

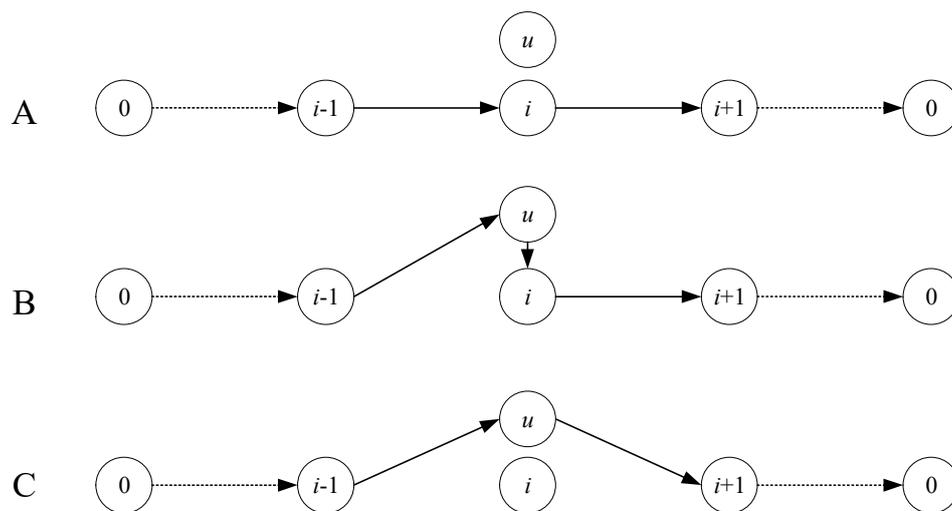


Figure 4. A is the initial solution, B is the result of the Solomon insertion operation, C is the result of the 0-exchange.

3.3. Path Optimization

In this stage, the remove-random perturbation and IPP-based tabu search are used to optimize the path of the initial solution obtained in the previous stage.

3.3.1. Remove-Random (RR) Disturbance

Within the interval $(0, 1)$, randomly generate a number r to determine the size of r and the probability of deletion $rand$. If $r < rand$, delete the currently accessed point and add the deleted point to the set U ; otherwise, the current point is retained without any transformations. Repeat the above operation to traverse each point in each path in the initial solution.

3.3.2. IPP-Based Tabu Search (IPP-TS)

Adding the idea of tabu search on the basis of IPP is to improve the efficiency of path optimization. Drawing on the previous research methods [9], the corresponding tabu object and tabu step are specified for different moving operations, so that the tabu object (node) will not be repeated in a given number of iterations, as shown in Table 2. For example, if the nodes $(i, i + 1, j)$ perform or-opt operations in the n times iterations, they are listed as tabu objects to prevent these nodes from being an or-opt operation again in the future $n+tabu2$ iterations.

Table 2. Tabu objects.

Operational Operator	Tabu Objects	Tabu Step
2-opt, 2-opt*	$(i, j), (i + 1, j + 1)$	<i>tabu1</i>
or-opt	$(i, i + 1, j)$	<i>tabu2</i>
0-exchange, 1-exchange, 2-exchange	(i, j)	<i>tabu3</i>
2-relocate	(r^1, i)	<i>tabu4</i>

¹ *r* is the path before *i* swaps.

The corresponding aspiration criterion rule is that when there is a feasible tabu solution for the candidate solution and it is better than the best solution currently found, the solution is accepted as the current solution and the release is released.

Combined with the simulated annealing process, the algorithm flow of this stage of path optimization is:

- Step 1 Obtain the feasible solution *S* generated in the process of Section 3.2 as the initial solution and set the current iteration number of *iter* and *no_impr* (the un-updated algebra) to 0.
- Step 2 After *S* is disturbed by RR, the points in the set *U* are traversed by the Solomon insertion method, and these points are reinserted to form a new solution, and the new solution is stored as a *local_best*;
- Step 3 Use IPP-TS for local search to form a new solution *S'*, and update the tabu table;
- Step 4 Compare the size of *f(S)* and *f(S')*, if $f(S') < f(S)$, update the current optimal solution "Best" and the local optimal solution "*local_best*", and set "*no_impr*" to zero, go to Step 3, otherwise, jump to Step 5;
- Step 5 If $f(S') < f(S)$ make a judgment. Here we use the cooling strategy commonly used in SA: $T_{k+1} = \alpha \times T_k$ (α is the cooling coefficient), the probability of accepting the inferior solution: $P_{sa} = \exp\{-|f(S') - f(S)|/T\}$ (*S* and *S'* represent the objective function value of the current solution), and then randomly produces a random number *test* (0, 1). If $test \leq P_{sa}$ is satisfied, the inferior solution is used to replace the current solution and it is stored as the local optimal solution *local_best* and let $no_impr = no_impr + 1, iter = iter + 1$;
- Step 6 When *no_impr* is greater than the set value limit, the current best overrides *local_best* to enter the next loop, and *no_impr* is set to zero;
- Step 7 Determine whether the termination condition has been reached. If $iter > max$ terminates the current step and saves the current optimal solution best as the initial solution to the next step, otherwise go to Step 2.

3.4. Conversion of Personnel Types

After the above two steps, the shortest path arrangement is obtained when all paths use the K-type service personnel (the employees with the most skills) without considering the different requirements of the skills at each task point. Design the following algorithm to convert the personnel type:

3.4.1. Transformation of Personnel Types (2-Random Exchange, 2RE)

The transformation process is controlled by two probabilities, P_h and P_l , where P_h determines whether the service personnel will be replaced by a higher level of personnel (the more skills mastered, the higher the level); P_l determines whether it can be replaced by a lower-level staff. The algorithm process is as follows:

- Step 1 Traverse each path in the solution to generate a path R_{tmp} that is different from the current employee type, but whether the modification is accepted is also judged by Step 2 and Step 3;
- Step 2 Generate a random number $test = rand(0,1)$ within an interval (0,1) judged by the following two conditions:

- If the newly generated path type is higher than the current path, and $P_h < test$, the new path type is accepted;
- Accept the new path type if the new path type and current type belong to the same level, or lower than the current path type, and $P_l < test$, the new path type is accepted.

Step 3 If the operations in Step 2 do not meet the probabilistic requirements, the original level remains the same, but can be combined with different skills (that is, when the original path is to master A and B skills of the 2 level staff service, it can be changed to become a level 2 employee who masters B and C skills); After Step 2 and the current step, if the path is changed, the path will not be changed again within the m iterations; (where the size of m here is consistent with the tabu step setting value in the tabu search operation);

Step 4 If there is a change in the Step 2 and Step 3 path, the filter is performed to traverse all the points in the current path and add points that do not meet the skill requirements to the U set; otherwise, jump Step 1 operation and continue traversing the path.

3.4.2. Skill Check (Check)

The solution obtained as a result of the above steps may be the use of redundant personnel or the presence of illegal service points (the service personnel skills cannot meet the requirements of the point). If a path contains only two skill requirements, but uses a person with three skills, that is, redundancy, at this point, the path is reduced to a path that contains only two skilled personnel; and if an illegal service point occurs, the point will be deleted and stored it in the U set, and then the Solomon insertion method is reused to insert the appropriate location.

The algorithm flow of the Check operation is:

- Step 1 Obtain the solution S produced in the process of Section 3.3 as the initial solution of this iteration, and randomly increase the R empty paths to increase the diversity of the solution;
- Step 2 Perform RR and skill check to generate a new solution S' ;
- Step 3 Take advantage of the optimization process in the process of Section 3.3, and change the objective function to $f_1(S)$;
- Step 4 If Step 3 satisfies the termination condition, the current optimal solution outputs and this step terminates, otherwise Step 2 is returned.

3.5. Algorithm Flow

The overall flow of the HSATS algorithm is as follows:

- Step 1 Initialization, using the Solomon insertion method and IPP to obtain the initial solution S ;
- Step 2 Path optimization, using IPP-TS combined with SA for local search for path optimization, resulting in a new solution S' instead of the initial solution S ;
- Step 3 Personnel type conversion, the use of 2RE and Check process for path conversion;
- Step 4 Determine whether the current iteration number of iterations $Iter_num$ reach the maximum number of iterations N . If $Iter_num < N$, then $Iter_num = Iter_num + 1$, return to Step 2, otherwise, go to Step 5;
- Step 5 Meet termination conditions, end the optimization, delete redundant empty path, and output the result.

4. Calculation and Analysis

Due to the lack of industry-recognized standard databases, the test examples in this paper are based on the standard Solomon example library of VRPTW [23]. The Solomon example library contains six types of examples: C1, C2, RC1, RC2, R1, R2. Each example contains: the label of the point (0 represents the service point, the rest are customer points), the coordinates of the point, service time, the demand for goods, the time window. All six examples use the same coordinate points and use different

service times, the demand for different goods, and different time windows. In this paper, the classic Solomon example is used to improve the demand for goods, and the demand for goods is changed to the requirement of different skills. The rest is unchanged according to the original example, in which each time the calculation is carried out, the skill requirement of the point is randomly assigned. The salary gap for personnel corresponding to a wide range of professional skills is 100 (in the case of three skill levels and 100, 200, 300, that is, the salary of employee’s lowest-level is 100, which increases in turn). The experiment uses Java JDK1.7 to configure an Intel(R) Core(TM) i7-4790 CPU @3.6 GHz; RAM: 8.00 GB DDR; OS: Windows 10 desktop.

4.1. Analysis of Parameters and Strategies

In the design of the heuristic algorithm, the value of algorithm parameters has an important influence on the effect and efficiency of the algorithm. The main parameters involved in this algorithm include: the parameters of the Solomon insertion method, the cooling coefficient and starting temperature in SA, the tabu step in TS and *no_impr* in the process of the algorithm, the probability of deleting points in the disturbance process *rand*. Additionally, the selection of two stochastic control variables P_h, P_l in the process of path conversion were used.

The parameters of the Solomon insertion method directly follow the original text parameters [9]. In the preliminary test, the C101-C109 instance in the Solomon VRPTW example library was selected and the C104 group was found to be the most sensitive. The subtle changes in the parameters of the algorithm will have a significant impact on the quality of the solution. Therefore, it was selected as the test example for parameter setting.

Considering Tables 3–7, we can set the parameters of the algorithm: (1) In Table 3 when *no_impr* is 10, the average solution, standard deviation, and coefficient of variation are all minimum. It is shown that the algorithm can obtain a better solution under good stable status when *no_impr* = 10. (2) In Table 4 when disturbance operation is not used (*rand* = 0), the quality of the solution and the stability of the algorithm are the worst. When the disturbance operation is used (*rand* ≠ 0), the quality of the solution and the stability of the algorithm decreases gradually with the increase of disturbance probability *rand*. (3) In Table 5 when the contraindication step is 20, the average solution, standard deviation, and coefficient of variation are the minimum. (4) The measurement of simulated annealing parameters refers to the test method in paper [22]. That is to say, different combinations of $T_0-\alpha$ are considered at the same time. Additionally, in Table 6, it is found that the algorithm $T_0 = 0.999$ and $\alpha = 10$ has the best effect on the solution of all. (5) In the probability selection of the 2RE operation, this paper draws on the method of simulating annealing parameter determination. A variety of different combinations are considered, such as considering the combination of P_h and P_l at the same time. In Table 7, the optimal solution, the average, and coefficient of variation will mostly increase with the raising of P_l . When $P_h = 0.3$ and $P_h = 0.5$ are used, the algorithm works better. Although the mean value of $P_h = 0.3$ is a little higher, considering optimum value (O.V.) and coefficient of variation (C.V.) is lower, we chose $P_h = 0.3$. Additionally, the optimal parameters were finally determined as follows: *no_impr* = 10; *rand* = 0.1; *tabu_length* = 20; $T_0 = 0.999$; $\alpha = 10$; $P_h = 0.3$; $P_l = 0.1$.

Table 3. Parameters of *no_impr* in path optimization.

	Best Solution	Average Solution	Standard Deviation	Coefficient of Variation
0	803.5432	805.4517	1.338107	0.016
10	803.5432	804.8631	0.914442	0.001136
20	803.5432	805.2226	1.164162	0.001446
30	803.5432	805.3211	1.172981	0.001457
40	804.2811	805.0887	0.956384	0.001188

Table 4. Perturbation operation probability *rand*.

	Best Solution	Average Solution	Standard Deviation	Coefficient of Variation
0	808.925	843.3944	30.50354	0.036168
0.1	803.5432	804.8631	0.914442	0.001136
0.2	805.7826	807.6978	1.374694	0.001702
0.3	807.114	809.8734	1.893115	0.002338
0.4	809.5577	812.3651	1.767091	0.002175

Table 5. Tabu step.

	Best Solution	Average Solution	Standard Deviation	Coefficient of Variation
5	803.5432	805.8537	1.238797	0.001537
10	803.5432	805.307	1.053678	0.001308
15	803.5432	805.2574	1.075042	0.001335
20	803.5432	804.8631	0.914442	0.001136
25	803.5432	805.337	0.975921	0.001212

Table 6. Determination of simulated annealing parameters ($T_0-\alpha$).

T_0	100			10			1			0.1		
	O.V. ¹	Mean	C.V. ²									
0.9	803.54	805.18	0.09%	803.54	805.21	0.10%	804.28	805.29	0.13%	803.54	805.09	0.12%
0.93	803.54	805.01	0.10%	803.54	805.17	0.17%	803.54	804.87	0.12%	803.54	805.11	0.13%
0.95	803.54	805.52	0.19%	803.54	805.39	0.13%	803.54	805.19	0.13%	804.28	805.41	0.16%
0.97	803.54	805.25	0.14%	803.54	805.14	0.11%	803.54	805.15	0.14%	804.28	805.07	0.07%
0.99	804.28	805.45	0.13%	804.28	805.03	0.09%	804.28	805.24	0.11%	803.54	805.15	0.16%
0.995	803.54	805.31	0.18%	803.54	805.05	0.14%	804.28	805.41	0.16%	803.54	804.95	0.13%
0.999	803.54	805.24	0.08%	803.54	804.90	0.09%	803.54	805.58	0.19%	803.54	804.87	0.10%

¹ O.V. = optimum value; ² C.V. = coefficient of variation.

Table 7. Selection of two variation probabilities in 2-Random Exchange (2RE) operation.

	0.1			0.3			0.5			0.7			0.9		
	O.V. ¹	Mean	C.V. ²												
0.1	2567.60	2591.25	1.61%	2547.08	2599.59	1.42%	2553.17	2598.63	1.70%	2550.42	2602.51	2.59%	2550.07	2599.45	2.22%
0.3	2640.96	2754.60	2.09%	2565.83	2658.83	2.08%	2564.00	2626.52	2.01%	2552.97	2608.82	1.64%	2567.38	2616.51	1.47%
0.5	2691.27	2835.83	2.77%	2577.94	2711.37	4.41%	2556.58	2664.73	2.29%	2570.94	2634.82	1.61%	2570.91	2650.63	2.19%
0.7	2711.30	2846.71	3.40%	2570.56	2705.84	2.56%	2566.86	2654.67	2.59%	2588.38	2674.05	2.43%	2558.52	2682.87	3.56%
0.9	2704.76	2876.77	3.44%	2592.08	2765.29	4.14%	2583.40	2729.64	3.33%	2568.42	2742.43	5.66%	2604.02	2770.43	4.91%

¹ O.V. = optimum value; ² C.V. = coefficient of variation.

4.2. Example Calculation

This section takes the C105 group as an example calculation. Because HSATS is a heuristic algorithm, it is impossible to guarantee that the results are the same every time. Therefore, it runs 30 times to select the best one for analysis, as shown in Tables 8 and 9.

Table 8. Results in the traditional path problem without considering skill requirements.

No.	Personnel Type	Personnel Salary	Path
1	7	300	0-81-78-76-71-70-73-77-79-80-0
2	7	300	0-67-65-63-62-74-72-61-64-68-66-69-0
3	7	300	0-98-96-95-94-92-93-97-100-99-2-1-75-0
4	7	300	0-57-55-54-53-56-58-60-59-0
5	7	300	0-13-17-18-19-15-16-14-12-0
6	7	300	0-20-24-25-27-29-30-28-26-23-0
7	7	300	0-5-3-7-8-10-11-9-6-4-0
8	7	300	0-32-33-31-35-37-38-39-36-34-22-21-0
9	7	300	0-43-42-41-40-44-46-45-48-51-50-52-49-47-0
10	7	300	0-90-87-86-83-82-84-85-88-89-91-0
path length: 822.84		salary: 3000	Total cost: 3822.84

Table 9. Results of the multi-skilled human resource path studied in this paper.

No.	Personnel Type	Personnel Salary	Path
1	1	100	0-20-24-27-29-30-28-22-21-0
2	4	200	0-65-96-94-92-97-100-99-2-1-75-0
3	2	100	0-55-54-53-72-64-68-0
4	4	200	0-43-42-41-62-44-46-45-48-51-50-47-0
5	3	100	0-3-5-25-40-60-61-66-69-0
6	1	100	0-57-63-74-56-58-59-0
7	3	100	0-67-98-95-93-9-26-0
8	7	300	0-13-17-18-19-15-16-14-12-4-0
9	4	200	0-7-8-10-11-6-23-0
10	7	300	0-90-87-86-83-82-84-85-88-89-91-0
11	7	300	0-32-33-31-35-37-38-39-36-34-52-49-0
12	7	300	0-81-78-76-71-70-73-77-79-80-0
path length: 1241.87		salary: 2300	Total cost: 3541.87

The situation is shown in Table 8 where the skill requirements are not considered. At this time, employees are all employees with three skills. Table 9 shows the existence of all types of employees, which is the result of this problem.

It can be seen from Tables 8 and 9 that the reasonable matching of all categories of personnel is more important than the simple path optimization. A good group of personnel can often significantly reduce the operating costs of the organization. Table 9 compared to Table 8, even with the addition of two paths, shows that better results can be obtained in terms of professional skills, total wages, and total operating costs than in Table 8. Therefore, in the management practice, the reasonable combination of different types of multi-skilled employees can make the production organization and personnel dispatching of the enterprise more flexible and efficient, and effectively reduce the production and operation costs.

4.3. Solving Performance

4.3.1. Impact of Algorithm Strategy

The design idea of the algorithm is TS + SA algorithm framework, and TS is a kind of local search algorithm (LS). Similar to the literature [24], the HSATS is compared with the classical LS and LS + SA algorithm frameworks.

Three Java codes were written by the same person using the same data structure and operation operators to ensure, as far as possible, that there were no programming skills and differences in programming levels. The path optimization test was carried out with six types of examples from the well-known Solomon example library, each of which uses the above three algorithms to operate 30 times on the same computer. The calculation results are shown in Table 10.

Table 10 shows a simple comparison of the length of a path without skill constraints, which was used to validate the local search ability of the algorithm. Since the skill constraints were not considered, the problem in this paper is similar to those of traditional VRP, so each algorithm can be compared with the optimal VRP solution of the existing VRP (given in the “Optimal Path” column). The result of the significance test was obtained by comparing HSATS with LS-SA. It can be seen that: (1) For different types of examples, the HSATS can obtain a high-quality path. In some cases, a better solution can be obtained than the existing optimal solution. (2) Compared to the LS and LS + SA two algorithm frameworks, the HSATS is superior to the previous two algorithms in both accuracy and stability of the solution.

4.3.2. Effective Synthesis

Further, considering the situation with multi-skill constraints, select the first three of each type of study in the Solomon example library, and keep the number of salary fixed to 100, 200, and 300. The algorithm parameters were chosen from the previous test parameters, and each algorithm runs independently for 30 times. The pros and cons and stability of the algorithm were analyzed by the three statistics: optimal value, average value, and coefficient of variation. (Due to the same number of iterations in this article, the calculation time gap is small, so the operation times are not listed separately in LS and LS + SA.)

The test results are given in Table 11, from which you can see that: (1) Relative to LS and LS + SA, comparing the second column of the table with the optimal solution of each algorithm, it can be seen that using multi-skilled employees is more cost-saving than using single-skilled employees. (2) HSATS solves different types of examples, has good performance, and can find better solutions. (3) In most samples, the mean value and C.V. of HSATS are the smallest, indicating that HSATS has better stability.

The effect of the number of skills on the performance of the algorithm is given in Table 12. It can be seen from the table that: (1) Under different skill numbers, none of the C.V. of the HSATS algorithm are large. It shows that the stability of the algorithm will not change obviously with the number of skills. (2) Under the same sample, the solution time of the HSATS algorithm increases slightly with the increase of the number of skills, but the increase is small. There is no significant increase in computing time due to exponential growth in the number of combinations, resulting in the algorithm not being available, but also the stability and applicability of the algorithm is also side validated.

Tables 11 and 12 of the comprehensive considerations can be seen—for such problems, from the test examples (including large, medium, and small—three kinds of scale), the HSATS algorithm proposed in this paper to obtain a stable solution time is within half a minute. For VRPTW, since its exact solution is very difficult to obtain, it is particularly important to obtain a reasonable solution in the effective time. Finding a reasonable result in a short time is a solution principle for most VRPs with time windows and a discriminating basis for the performance of the algorithm. From this perspective, the HSATS algorithm is an effective and practical algorithm.

Table 10. Results of different algorithm strategies.

Example	Salary	Path Length	LS			LS + SA			HSATS			Significance Test ³	
			O.V. ¹	Mean	C.V. ²	O.V. ¹	Mean	C.V. ²	O.V. ¹	Mean	C.V. ²	t Value	p Value ⁴
C101	3000	828.94	828.94	835.50	1.68%	828.94	829.48	0.28%	828.94	828.94	0.00%	-1.887	0.024
C102	3000	828.94	821.93	897.10	11.77%	821.93	821.93	0.00%	821.93	821.93	0.00%	N/A	N/A
C103	3000	828.06	816.31	826.98	1.35%	815.44	818.27	0.34%	815.44	816.24	0.22%	-2.909	0.006
R101	5700	1645.79	1649.34	1546.49	12.34%	1217.19	1539.76	12.11%	1215.90	1537.70	12.10%	-1.890	0.023
R102	5100	1648.12	1472.81	1528.25	5.34%	1466.60	1524.19	5.28%	1472.81	1522.57	5.10%	-2.172	0.020
R103	3900	1292.68	1213.62	1368.05	5.60%	1270.59	1366.39	5.63%	1211.00	1303.12	9.28%	-2.389	0.012
RC101	4200	1696.94	1632.90	1653.98	1.11%	1626.16	1646.68	0.65%	1642.88	1643.18	0.03%	-2.872	0.007
RC102	3600	1554.75	1471.16	1476.83	0.36%	1466.60	1474.05	0.30%	1472.81	1475.10	0.30%	-1.669	0.035
RC103	3300	1261.67	1274.82	1318.59	1.26%	1270.59	1313.07	1.45%	1215.00	1221.30	0.26%	-28.671	0.000
C201	900	591.56	591.56	591.56	0.00%	591.56	591.56	0.00%	591.56	591.56	0.00%	N/A	N/A
C202	900	591.56	591.56	594.02	1.25%	591.56	592.27	0.66%	591.56	591.56	0.00%	-2.883	0.007
C203	900	591.17	591.56	616.47	1.96%	591.17	612.49	1.85%	591.17	592.58	0.92%	-13.415	0.000
R201	1200	1252.37	1189.41	1193.30	0.59%	1189.41	1194.67	0.76%	1184.36	1193.15	0.63%	-3.2186	0.002
R202	900	1191.7	1041.10	1045.96	0.62%	1041.10	1046.82	0.51%	1041.10	1050.32	1.52%	-1.796	0.026
R203	900	939.54	879.80	880.51	0.05%	879.80	881.92	0.42%	879.80	881.72	0.43%	-1.652	0.042
RC201	1200	1406.91	1310.44	1350.79	2.47%	1310.44	1318.75	0.84%	1310.44	1319.10	0.91%	-1.913	0.022
RC202	1200	1367.09	1118.66	1126.11	0.70%	1118.66	1122.16	0.43%	1118.66	1120.64	0.34%	-2.525	0.009
RC203	1200	1049.62	937.45	942.65	0.43%	937.45	941.18	0.44%	928.43	939.97	0.39%	-2.682	0.008

¹ O.V. = optimum value; ² C.V. = coefficient of variation; ³ between hybrid simulated annealing tabu search (HSATS) and local search (LS) + simulated annealing (SA); ⁴ significance 2-tailed.

Table 11. Total cost in different algorithm strategies.

Example	Non Multi-Skill ¹	LS ²			LS + SA ²			HSATS ²			Number of Vehicles Required	Solving Time (s)
		O.V. ³	Mean	C.V. ⁴	O.V. ³	Mean	C.V. ⁴	O.V. ³	Mean	C.V. ⁴		
C101	3828.94	3668.39	3742.94	1.17%	3404.65	3557.09	3.02%	3398.13	3565.57	3.30%	12.03	13.79
C102	3828.94	3605.70	3746.78	2.71%	3372.87	3505.16	2.01%	3322.70	3413.64	2.56%	12.40	18.11
C103	3828.06	2918.17	3019.06	1.87%	3145.51	3565.80	5.98%	2902.96	2963.92	0.03%	11.73	20.64
R101	7345.79	5499.05	6482.29	6.21%	4728.88	4964.37	2.92%	4737.71	4905.07	3.36%	23.30	13.95
R102	6748.12	4586.25	5579.13	8.02%	4542.94	4605.69	0.88%	4513.38	4565.15	0.84%	21.17	18.22
R103	5192.68	3499.03	3551.96	0.81%	3476.91	3576.80	3.07%	3460.70	3503.75	0.80%	16.10	20.41
RC101	5896.94	4408.44	5015.56	6.60%	4432.44	4668.37	3.08%	4346.76	4447.54	1.64%	19.47	15.26
RC102	5154.75	3930.85	4250.21	3.25%	3921.06	4137.01	3.08%	3926.82	4033.47	1.44%	17.17	19.37
RC103	4561.67	3476.22	3678.32	3.91%	3482.32	3607.96	2.88%	3429.77	3503.05	1.16%	14.77	20.30
C201	1491.56	1491.56	1491.56	0.00%	1491.56	1491.56	0.00%	1491.56	1491.56	0.00%	3.10	14.40
C202	1491.56	1491.56	1491.56	0.00%	1491.56	1491.56	0.00%	1491.56	1491.56	0.00%	3.03	19.59

Table 11. Cont.

Example	Non Multi-Skill ¹	LS ²			LS + SA ²			HSATS ²			Number of Vehicles Required	Solving Time (s)
		O.V. ³	Mean	C.V. ⁴	O.V. ³	Mean	C.V. ⁴	O.V. ³	Mean	C.V. ⁴		
C203	1491.17	1491.17	1498.65	0.68%	1491.17	1492.06	0.33%	1491.17	1491.17	0.00%	3.07	21.73
R201	2452.37	2427.48	2538.71	2.81%	2372.59	2487.25	1.28%	2350.23	2462.41	1.57%	4.73	14.07
R202	2091.7	2138.74	2292.70	3.18%	2146.09	2253.94	2.30%	2137.84	2204.57	1.62%	4.97	16.96
R203	1839.54	1743.87	1902.00	4.40%	1782.73	1880.76	2.54%	1757.78	1776.16	0.76%	3.97	17.86
RC201	2606.91	2572.36	2750.00	2.77%	2543.08	2621.48	2.21%	2525.18	2632.17	2.64%	4.73	17.40
RC202	2467.09	2373.03	2522.44	2.74%	2362.33	2522.44	2.85%	2362.33	2391.38	1.01%	4.37	19.94
RC203	2049.62	1969.99	2111.62	2.42%	1987.01	2071.75	2.83%	1983.15	2060.58	1.86%	3.87	21.65

¹ Without considering multi-skill requirements; ² consider multi-skill requirements; ³ O.V. = optimum value; ⁴ C.V. = coefficient of variation.

Table 12. Performance of the algorithm when the skill number *k* is equal to 3, 4, and 5, respectively.

Example	<i>k</i> = 3				<i>k</i> = 4				<i>k</i> = 5			
	O.V. ¹	Mean	C.V. ²	A.T. ³	O.V. ¹	Mean	C.V. ²	A.T. ³	O.V. ¹	Mean	C.V. ²	A.T. ³
C101	3398.13	3565.57	3.30%	9.88	3511.17	3721.28	3.19%	15.94	3658.00	3792.51	2.57%	17.19
C102	3322.70	3413.64	2.56%	14.81	3440.76	3542.14	2.56%	17.10	3621.13	3735.28	2.41%	17.53
C103	2902.96	2963.92	1.11%	17.14	3005.30	3087.38	1.24%	19.45	3055.29	3156.36	1.61%	19.88
R101	4737.71	4905.07	3.36%	13.98	4849.42	5034.01	3.25%	16.25	4903.55	5103.30	3.31%	16.72
R102	4513.38	4565.15	0.84%	18.09	4634.77	4693.55	0.94%	20.34	4683.33	4770.90	0.92%	20.85
R103	3460.70	3503.75	0.80%	19.75	3572.01	3629.25	0.86%	22.00	3650.36	3707.68	0.84%	22.52
RC101	4346.76	4447.54	1.64%	4.45	4467.53	4571.64	1.69%	18.19	4547.43	4646.25	1.73%	18.61
RC102	3926.82	4033.47	1.44%	19.86	4028.88	4160.53	1.52%	22.13	4116.73	4237.59	1.59%	22.61
RC103	3429.77	3503.05	1.16%	21.39	3556.40	3625.34	1.32%	23.70	3610.20	3700.38	1.56%	24.17
C201	1491.56	1491.56	0.00%	12.43	1491.56	1491.56	0.00%	14.68	1491.56	1491.56	0.00%	17.49
C202	1491.56	1491.56	0.00%	16.80	1491.56	1491.56	0.00%	19.02	1491.56	1491.56	0.00%	19.53
C203	1491.17	1491.17	0.00%	19.79	1491.17	1491.17	0.00%	22.02	1491.17	1491.17	0.00%	22.54
R201	2350.23	2462.41	1.57%	16.70	2478.35	2591.36	1.64%	18.92	2578.15	2665.51	1.78%	19.44
R202	2137.84	2204.57	1.62%	19.35	2259.38	2326.10	1.52%	21.62	2291.80	2404.34	1.88%	22.13
R203	1757.78	1776.16	0.76%	20.85	1859.27	1901.14	1.06%	23.08	1909.08	1980.59	1.67%	23.59
RC201	2525.18	2632.17	2.64%	17.63	2644.61	2755.00	2.46%	19.87	2692.69	2824.93	2.72%	20.41
RC202	2309.00	2436.63	2.95%	2.44	2415.95	2560.97	2.88%	21.95	2503.37	2632.30	2.67%	22.40
RC203	1983.15	2060.58	1.86%	22.05	2091.75	2187.46	1.85%	24.27	2183.92	2260.86	1.86%	24.80

¹ O.V. = optimum value; ² C.V. = coefficient of variation; ³ A.T. = average time.

In summary, the HSATS algorithm proposed in this paper has obvious advantages in accuracy, stability, and practicability.

5. Conclusions

Under the current general trend of the decline of the labor force population and the improvement of the quality of laborers in the world, enterprises are more inclined to employ laborers with multiple skills, in order to improve the flexibility of enterprise human resource management and reduce the cost of enterprise operation. For example, the U-shaped cell presents in the just-in-time production line, in which multi-skilled employees are widely used [25]. While the flexibility of human resource management is improving, and the operating costs of enterprises is reducing, there are also difficulties in personnel dispatching and scheduling. Aiming at this kind of problem, this paper establishes an optimization model of a human resource path problem with time window and multi-skill resource constraints. Because this kind of problem is NP-hard, in this paper, a heuristic algorithm was designed based on the TS + SA algorithm framework of various local search operators.

Based on the typical instances in the classical Solomon instances library, the example calculation and algorithm performance analysis show that: (1) The HSATS algorithm proposed in this paper is more suitable for solving the optimization problem proposed in this paper than the LS and LS + SA algorithms, and there is a significant difference between HSATS and LS + SA in solving performance. (2) Compared with the single skilled employee model, the use of multi-skilled employees can effectively reduce the overall cost of the enterprise. (3) With the increase of the number of employee skills, the solving time of the algorithm will increase slightly, but an optimal solution of this kind of problem can still be obtained in a relatively short time.

The research in this paper is a preliminary exploration of such problems. In the future, we can do further research in the following aspects: (1) This paper considers the static optimization problem. In the future we can consider multi-cycle and multi-stage dynamic optimization problems. For example, the multi-skilled personnel scheduling problem between multiple engineering projects at different construction stages. (2) In the design of the algorithm, combined with the existing and advanced path optimization operators, there is still room for improvement in the path conversion strategy of this paper. Through the continuous improvement of the above-mentioned problems and algorithms, we shall constantly enrich and perfect the research and application of this problem.

Author Contributions: Conceptualization, Y.L. and J.F.; methodology, Z.Z.; software, J.W.; validation, J.F., Y.L., and Z.Z.; writing—original draft preparation, Z.Z. and J.W.; writing—review and editing, J.F.; project administration, J.F.; funding acquisition, Y.L.

Funding: This research was supported by National Natural Science Foundation of China, grant number 71471093; Postdoctoral Research Foundation of China, grant number 2019M652611; Doctoral Funding of Wuhan Business University, grant number 2016KB004, and Scientific Research Innovation Team Construction Project of Wuhan College, grant number KYT201805.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Sheiner, L.; Sichel, D.; Slifman, L. *A Primer on the Macroeconomic Implications of Population Aging*; Divisions of Research & Statistics and Monetary Affairs: Washington, DC, USA, 2007; pp. 5–7.
2. U.S. Census Bureau. Available online: <http://www.census.gov/newsroom/press-releases/2017/cb17-51.html> (accessed on 31 May 2019).
3. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [[CrossRef](#)]
4. Laporte, G. Fifty years of vehicle routing. *Transport. Sci.* **2009**, *43*, 408–416. [[CrossRef](#)]
5. Braekers, K.; Ramaekers, K.; Nieuwenhuysse, I.V. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [[CrossRef](#)]

6. Savelsbergh, M. Local search in routing problems with time windows. *Ann. Oper. Res.* **1985**, *4*, 285–305. [[CrossRef](#)]
7. Bräysy, O.; Gendreau, M. Vehicle routing problem with time windows part I: Route construction and local search algorithms. *Transport. Sci.* **2005**, *39*, 104–118. [[CrossRef](#)]
8. Solomon, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **1987**, *35*, 254–266. [[CrossRef](#)]
9. Qin, H.; Ming, W.; Zhang, Z.; Xie, Y.; Lim, A. A tabu search algorithm for the multi-period inspector scheduling problem. *Comput. Oper. Res.* **2015**, *59*, 78–93. [[CrossRef](#)]
10. Zhang, Z.; Che, O.; Cheang, B.; Lim, A.; Qin, H. A memetic algorithm for the multiperiod vehicle routing problem with profit. *Eur. J. Oper. Res.* **2013**, *229*, 573–584. [[CrossRef](#)]
11. Correia, I.; Saldanha-Da-Gama, F. The impact of fixed and variable costs in a multi-skill project scheduling problem: An empirical study. *Comput. Ind. Eng.* **2014**, *72*, 230–238. [[CrossRef](#)]
12. Bellenguez, O.; Néron, E. Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. *Lect. Notes Comput. Sci.* **2005**, *3616*, 229–243.
13. Kazemipoor, H.; Tavakkoli-Moghaddam, R.; Shahnazari-Shahrezaei, P.; Azaron, A. A differential evolution algorithm to solve multi-skilled project portfolio scheduling problems. *Int. J. Adv. Manuf. Technol.* **2013**, *64*, 1099–1111. [[CrossRef](#)]
14. Cheng, L.; Liao, G.R.; Liu, Z.Y. A multi-objective immune genetic algorithm for project scheduling on multi-skill resources. *Appl. Mech. Mater.* **2015**, *7*, 719–720. [[CrossRef](#)]
15. Maghsoudlou, H.; Afshar-Nadjafi, B.; Niaki, S.T.A. Multi-skilled project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search. *Appl. Soft Comput.* **2017**, *54*, 46–61. [[CrossRef](#)]
16. Rong, C.; Liang, C.; Gu, D.; Leung, J.Y.-T. A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution. *Int. J. Prod. Res.* **2017**, *55*, 6207–6234.
17. Lian, J.; Liu, C.G.; Li, W.J.; Yin, Y. Multi-skilled worker assignment in seru production systems considering worker heterogeneity. *Comput. Ind. Eng.* **2018**, *118*, 366–382. [[CrossRef](#)]
18. Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [[CrossRef](#)]
19. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
20. Vignaux, G.A.; Michalewicz, Z. A genetic algorithm for the linear transportation problem. *IEEE Trans. Syst. Man Cybern. Syst.* **1991**, *21*, 445–452. [[CrossRef](#)]
21. Li, Y.; Guo, H.; Zhang, Y. An integrated location-inventory problem in a closed-loop supply chain with third-party logistics. *Int. J. Prod. Res.* **2018**, *56*, 3462–3481. [[CrossRef](#)]
22. Hu, Q.; Lim, A. An iterative three-component heuristics for team orienteering problem with time windows. *Eur. J. Oper. Res.* **2014**, *232*, 276–286. [[CrossRef](#)]
23. VRPTW Benchmark Problems. Available online: <http://w.cba.neu.edu/~msolomon/problems.htm> (accessed on 16 March 2019).
24. Lim, A.; Zhang, X. A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. *INFORMS J. Comput.* **2007**, *19*, 443–457. [[CrossRef](#)]
25. Jacobs, F.R.; Chase, R.B. *Operations and Supply Chain Management*, 14th ed.; McGraw-Hill Education: New York, NY, USA, 2013; pp. 338–354.

