Journal of Cleaner Production xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems

Jun-qing Li [a, b, *], Yun-qi Han [a], Pei-yong Duan [a, **], Yu-yan Han [b], Ben Niu [a], Cheng-dong Li [c], Zhi-xin Zheng [b], Yi-ping Liu [d]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China
[b] School of Computer, Liaocheng University, Liaocheng, 252059, China
[c] School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, 252101, China
[d] Department of Computer Science and Intelligent Systems, Osaka Prefecture University, 5998531, Japan

ARTICLE INFO

Article history:
Received 3 July 2019
Received in revised form
19 November 2019
Accepted 25 November 2019
Available online xxx

Handling editor: Bin Chen

Keywords:
Vehicle routing problem
Time window
Synchronized visit
Artificial bee colony
Energy consumptions

ABSTRACT

Prefabricated construction has attracted research interest as it can significantly improve the energy, cost, and time efficiency of construction. However, dispatching the required prefabricated components to construction sites in a prefabricated system is challenging. To address this issue, we modeled the dispatching problem as a special type of vehicle routing problem with time windows (VRPTW) and solved it by using an improved artificial bee colony (IABC) algorithm. First, to efficiently solve the cross-synchronization problem that occurs in prefabricated systems, two problem-specific lemmas were derived. Then, a hybrid initialization strategy was developed to generate feasible and efficient solutions and a well-designed encoding repair strategy was utilized to make solutions feasible. Finally, a variable length local search strategy was embedded to enhance the exploitation ability. To verify the performance of the proposed IABC algorithm, 55 instances were generated and used for simulation tests. Three efficient algorithms, including the two-phase genetic algorithm (TPGA), improved tabu search algorithm (ITSA), and adaptive large neighborhood search (ALNS) heuristic, were selected for detailed comparisons. Our simulation results show that, considering the energy consumption metric, the proposed algorithm yields average deviations of about 0.099, 0.096, and 0.143 times the values obtained with the TPGA, ITSA, and ALNS heuristic, respectively. The simulation results confirmed that the proposed algorithm can solve the VRPTW in prefabricated systems with high efficiency.

1. Introduction

The construction and operation of traditional buildings have environmental impacts through waste production and greenhouse gas emission. To improve the efficiency of construction and reduce waste, the prefabrication of buildings has been widely employed. In a prefabricated system, the building components are first produced in a factory and then transported to the construction or customer sites, where the components are assembled (Aye et al., 2012). In both traditional and prefabricated building systems, the distribution of materials is a challenging task in view of ensuring production efficiency, time savings, and cost savings. First, in a typical prefabricated system, there are generally two types of components: precast components and connecting components (Jaillon and Poon, 2009). They are significantly different in both size and weight, and therefore the constraints of different containers should be considered for the dispatching vehicles. Second, vehicles with special equipment should be synchronously dispatched to the assembly sites, so the components can be hoisted and installed. These two constraints should be considered to optimize the dispatching of vehicles in prefabricated systems.

The vehicle routing problem with time windows (VRPTW) has been investigated in many fields (Solomon, 1987; Bräysy and Gendreau, 2005a, 2005b). Solomon (1987) developed two types of insertion heuristics, one of which, the push-forward insertion

* Corresponding author. School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China.
** Corresponding author.
E-mail address: lijunqing@lcu-cs.com (J.-q. Li).

https://doi.org/10.1016/j.jclepro.2019.119464
0959-6526/

heuristic (PFIH), has been widely used to generate initial solutions. Further, Potvin and Rousseau (1993) improved the insertion heuristics in a parallel version. Atkinson (1994) proposed a greedy look-ahead approach to minimize the insertion impact of a newly selected customer. Subsequently, Bräysy and Gendreau (2005a) presented heuristics for the route construction and local search operator. Baniamerian et al. (2018) developed a two-phase genetic algorithm (TPGA) for the problem considering customer satisfaction. Xia and Fu (2018) utilized the improved tabu search algorithm (ITSA) for the VRPTW considering maximization of the satisfied rate. It should be noted that the TPGA and ITSA have been verified to be efficient algorithms for the VRPTW. Very recently, as an extended version of the classical VRPTW, the VRPTW with synchronized visits (VRPTWSyn) has gained attention. Hojabri et al. (2018) solved this problem by using a large neighborhood search with constraint programming. Liu et al. (2019) developed an adaptive large-neighborhood search heuristic. Decerle et al. (2019) proposed a heuristic algorithm hybridizing the ant colony optimization and memetic algorithms to solve the problem. However, there has been little research on the VRPTWSyn in terms of realistic applications considering special features or constraints.

Although there are numerous canonical applications for the VRP, for the analysis of prefabricated buildings and the VRPTW, there remains a gap between the dispatching problems and typical applications. Three critical issues should be solved to bridge this gap. The first is to model the realistic dispatching problem in the prefabricated system, where the two constraints of different vehicle containers and synchronized visits should be considered simultaneously. The second issue is to consider realistic constraints and to design problem-specific heuristics for the problem. The last one is to design an efficient heuristic algorithm for the VRPTWSyn problem in prefabricated systems.

The artificial bee colony (ABC) algorithm is a typical population-based optimization algorithm that has been applied to many types of optimization problems. In the ABC algorithm, there are three types of bees: employed bees, onlooker bees, and scout bees (Karaboga, 2005; Wang et al., 2012; Li et al., 2019a; Li et al., 2019b). The employed and onlooker bees aim to perform the exploitation tasks, while the scout bee completes the exploration tasks. After being applied in different types of optimization problems, such as the continuous optimization problem (Karaboga, 2005), capacitated vehicle routing problem (Ng et al., 2017), crowd evacuation in buildings (Liu et al., 2018), flow shop scheduling problem (Li et al., 2020a) and task scheduling in Cloud system (Li and Han, 2020), the ABC algorithm has been verified to be a competitive optimization algorithm. In this study, considering the efficient application of the ABC algorithm, we propose an improved ABC (IABC) algorithm for the VRPTWSyn problem in prefabricated systems.

The remainder of this report is organized as follows. Section 2 briefly presents the problem formulation. Then, Section 3 describes the proposed algorithm with all of the components. The simulation tests and analysis are presented in Section 4. Finally, Section 5 summarizes the conclusions of this study.

2. Problem formulation

2.1. Problem description

In this study, we modeled the dispatching problem in a prefabricated system as an extension of the VRPTWSyn (hereafter called the E-VRPTWSyn). The dispatching problem in a prefabricated system has the following features: (1) the system contains two types of components (prefabricated and connecting components), two types of vehicles (regular and special vehicles), and two types of construction sites or customers (regular and special customers); (2) all of the construction sites have two types of component demands, a time window constraint and a service duration; (3) regular vehicles dispatch construction components to all of the customers, while special vehicles transfer special equipment; and (4) regular construction sites can be immediately served after the arrival of regular vehicles, whereas special sites need to wait for the synchronized arrival of both regular and special vehicles.

The following assumptions are applied in modeling the E-VRPTWSyn problem:

(1) There is only one depot in the system, and all of the regular and special vehicles must leave from and return to the depot.
(2) Each regular construction site is served exactly once by one regular vehicle.
(3) Each special construction site must be served exactly once by a synchronized visit of one regular vehicle and one special vehicle.
(4) The construction site demand along the routes should not exceed the vehicle capacity.
(5) The return time of each vehicle does not exceed the maximum vehicle route time.
(6) The time window of each construction site should not be violated.

2.2. Problem formulation

The E-VRPTWSyn can be depicted on directed graph G, where $CC = \{1, \ldots, n_1\}$ is the set of nodes representing the regular construction sites, $SC = \{n_1+1, \ldots, n_1+n_2\}$ is the set of nodes showing special sites, and $C = CC \cup SC$ expresses two types of construction sites. Graph $G = (N, A)$ consists of nodes $N = \{0, n_1+n_2+1\} \cup C$ and arc set $A = \{(i,j) : i,j \in N, i \neq j\}$, where nodes 0 and $n_1 + n_2 + 1$ are the depot. Each construction site $i \in N$ is assigned a service duration s_i and time window $[a_i, b_i]$, where a_i and b_i specify the earliest and latest possible service start times for construction site i, respectively. Let t_{ij} represent the trip time and d_{ij} be the trip distance of arc $(i, j) \in A$. $DV = \{1, \ldots, m_1\}$ is the set of regular vehicles, and $HV = \{m_1+1, \ldots, m_1+m_2\}$ is the set of special vehicles. $V = DV \cup HV$ represents all of the vehicles in the system. The route of each vehicle $k \in V$ is to leave from and return to the depot. Each vehicle can be assigned to only one route and must return within the maximum travel distance H. We also set a time window $[a_0, b_0]$ for the depot site.

Considering the special features of the prefabricated system, we define ds_i and dc_i as the demands of construction site i for the prefabricated and connecting components, respectively. Let cs_k and cc_k be the maximum capacities of vehicle k for the prefabricated and connecting components, respectively. e_k is defined as the energy consumption coefficient of vehicle k. The notation used for the E-VRPTWSyn model is summarized in Table 1.

Decision variable:

Z_{ijk}: A binary value that is set to 1 if vehicle k serves the construction site i and travels directly to construction site j; otherwise, Z_{ijk} is set to 0, where $i \neq j; i,j \in N$.

Objective:

$$Min \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} d_{ij} Z_{ijk} e_k \tag{1}$$

Constraints:

$$\sum_{k \in V} \sum_{j \in N} Z_{ijk} = 1 \qquad \forall i \in C \tag{2}$$

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

Table 1
Notation of the E-VRPTWSyn model.

Symbol	Description
	Depot index
	Indices of the construction site
	Vehicle index
	Set of construction sites in the system
	Set of regular construction sites
	Set of special construction sites
	Set of vehicles in the system
	Set of regular vehicles
	Set of special vehicles
	kth vehicle from the regular vehicle set DV
	kth vehicle from the special vehicle set HV
	Earliest start time of the time window for the rth construction site on the kth route
	Earliest possible service time of construction site i
	Latest possible service time of construction site i
	Distance between nodes i and j
	Maximum travel distance of each vehicle
	Demand for prefabricated components of construction site i
	Demand for connecting components of construction site i
	Maximum capacity for prefabricated components of vehicle k
	Maximum capacity for connecting components of vehicle k
	Service duration for construction site i
	Arrival time of a regular vehicle at construction site i
	Arrival time of a special vehicle at construction site i
	Start time of service for construction site i,$u_i = max(td_i, th_i)$
	Energy consumption coefficient for vehicle k

$$\sum_{j=N} Z_{0jk} = 1 \qquad \forall k \in V \tag{3}$$

$$\sum_{j=N} Z_{j,n_1+n_2+1,k} = 1 \qquad \forall k \in V \tag{4}$$

$$\sum_{i\in N} ds_i \sum_{j\in N} Z_{ijk} \le cs_k \qquad j \ne i \tag{5}$$

$$\sum_{i\in N} dc_i \sum_{j\in N} Z_{ijk} \le cc_k \qquad j \ne i \tag{6}$$

$$a_i \le u_i \le b_i \quad , \quad \forall i \in N \tag{7}$$

$$u_i = \max\{td_i, th_i\} \quad , \quad \forall i \in N \tag{8}$$

$$Z_{ijk} \in \{0,1\} \qquad \forall i,j \in N, k \in V \tag{9}$$

The objective function in (1) minimizes the sum of the energy consumptions of all of the vehicles. Constraint (2) ensures that all of the customer services are fully covered. Constraints (3) and (4) specify that each route starts and ends at the depot. The two special constraints for the prefabricated system including the capacity constraints for the prefabricated components and connecting components are guaranteed in Constraints (5) and (6), respectively. Constraint (7) requires the start time of service for each construction site to be in the range of its time window. Constraint (8) ensures that the regular and special vehicles visit special construction sites synchronously. Constraint (9) imposes a restriction on the decision variable.

2.3. Problem example

Fig. 1 illustrates an example of the E-VRPTWSyn problem, in which five construction sites are set and denoted as $\{c_1, c_2, c_3, c_4, c_5\}$. Special sites c_3 and c_4 are indicated by gray circles. Two regular vehicles, R1 and R2, transport the components for the construction sites. In Fig. 1, three sites are served by R1 and two sites are assigned to R2. Note that two construction sites, c_3 and c_4, are also served by special vehicle R3. The information box beside each construction site specifies the corresponding demand and time window. For instance, d_2 (1,3) at site c_1 indicates that the demands for the prefabricated components and connecting components are 1 and 3 units, respectively. Then, t_2 (3,5) shows that the earliest and latest service times are 3 and 5, respectively. As denoted at the top right corner, the maximum capacities for the prefabricated components and connecting components were set to 5 and 10, respectively, for each regular vehicle.

Considering special construction site c_4, the arrival time of regular vehicle R2 is 1, and the arrival time of special vehicle R3 is 2. Therefore, the service of c_4 should be 2 because of the synchronized visits constraint. The relation between the two types of vehicles is shown in Fig. 2.

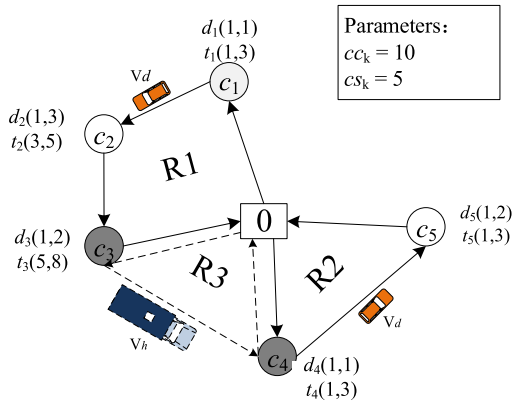


Fig. 1. Simple example of the E-VRPTWSyn problem.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

3. Proposed algorithm

For solving the dispatching problem in prefabricated systems, we propose an improved version of the classical ABC algorithm. The detailed components of this improved algorithm are described in this section.

3.1. Algorithm framework

The framework of the IABC approach is shown in Algorithm 1.

Algorithm 1. Framework of IABC

```
Generate the population by using the initialization strategy (cf. subsection 3.6)
Employed bee phase
for each solution i do
    Generate a neighboring solution j by using the VLLS method (cf. subsection 3.7) for solution i
    if j is better than i then
        Replace the current solution with j
        Update the best solution found so far with j
        Update the local best of i with j
    else
        Update the iteration time without improvement for i
    end
end
Onlooker bee phase
for each solution i do
    Randomly select another solution k from the population
    Select the better solution between i and k as the current solution i
    Perform steps 3-11 listed in the employed bee phase
end
Scout bee phase
for each solution i do
if the number of iterations without improvement for i exceeds L_m then
    Perform the global search heuristic described in subsection 3.8
end
```

3.2. Problem-specific structures

One challenge in the dispatching problem in a prefabricated system is cross-synchronization. Let us suppose that vehicle dv_k visits construction sites i and j sequentially, while vehicle hv_k visits construction sites l and m. However, sites i and m constitute a pair of synchronized construction sites like j and l, which means that i and m or j and l must have the same service start time. Under these circumstances, even with limitless relaxed time windows for these construction sites, i and m cannot be visited simultaneously and cross-synchronization occurs. We propose the following two lemmas to eliminate cross-synchronization.

Lemma 1. *Let the service start time for special construction site SC_i on regular vehicle dv_k be u_i and the arrival time of special vehicle hv_k at SC_i be th_i. Postponing the time u_i contributes to improving the fitness of the solution when hv_k arrives at SC_i later than u_i if $th_i > u_i$.*

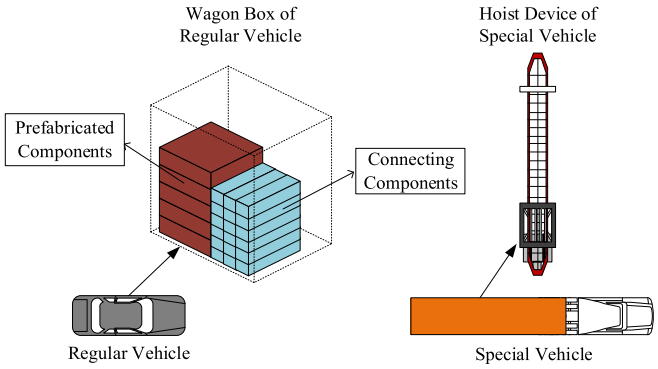


Fig. 2. Relation between the two types of vehicles.

Proof. *Suppose that SC_i is processed in the r^{th} position on dv_k and that the immediate successor of it is denoted as dv_k^{r+1}. The time windows for SC_i and dv_k^{r+1} are $[a_i, b_i]$ and $[a_k^{r+1}, b_k^{r+1}]$, respectively. The relaxed time window $rw_k^{r+1} = \max(a_k^{r+1} - (u_i + s_i + d_{r,r+1}), b_i - u_i)$. Then, we can postpone u_i for $\min(rw_k^{r+1}, th_i - u_i)$ units, and consequently SC_i can be started at any time in $[u_i, u_i + \min(rw_k^{r+1}, th_i - u_i)]$ without violation of any other customers. Therefore, the proof is verified.*

Lemma 2. *Sorting the construction sites according to the service start times u_i of their regular vehicles can prevent cross-synchronization.*

Proof. *If all of the special sites are sorted according to u_i of the regular vehicles, then adjusting the arrival times of special vehicles by using Lemma 1 would obviously improve the performance and prevent cross-synchronization. Therefore, the proof is verified.*

3.3. Solution representation

As in the commonly used encoding method described in the literature (Han et al., 2015; Afifi et al., 2016; Arnold and Sörensen, 2019; Li et al., 2019b; Pan et al., 2019), a two-dimensional vector is embedded for each solution. In addition, a flag vector is included for the solution representation to record whether the corresponding site is regular or special. Fig. 3 shows a solution encoding example, where there are three regular vehicles and one special vehicle. Four construction sites are assigned to the first vehicle, among which site c_1 is a special customer. On the second vehicle, there are two regular sites and one special site. For the third regular vehicle, all the assigned sites are regular. The construction site type is reported in the flag vector, where 0 corresponds to a regular construction site and 1 to a special site.

Considering the problem features in the prefabricated system, two challenges should be addressed to decode solutions. The first challenge is to decide the service time for each construction site, especially for the special sites requiring synchronized visits. The second challenge is to design a repair approach for infeasible solutions.

3.4. Synchronized sorting strategy

To solve the cross-synchronization problem, a native approach is to assign special vehicles for construction sites according to their scheduling sequence in the regular vehicles. Although this method usually generates feasible solutions, more special vehicles will be used, and therefore, the objective value will not be minimized. Fig. 4(a) depicts a result obtained using this simple method in which there are three special sites with time windows of {[0, 10], [20, 30], [10, 20]}. Let *AT* represent the arrival time for each construction site. In Fig. 4(a), according to the scheduling sequence in the regular vehicles, the special vehicle processing order is {1, 2, 3}.

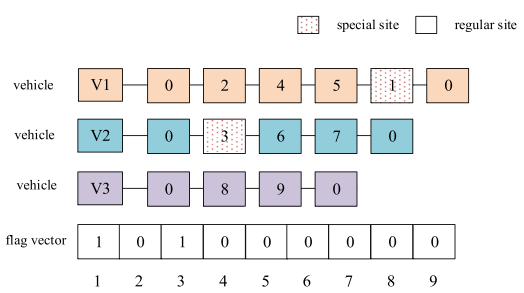


Fig. 3. Example of solution representation.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

It is evident that one special vehicle cannot satisfy all three special sites.

Based on the two lemmas discussed in subsection 3.2, we propose a synchronized sorting strategy to solve the cross-synchronization problem. The main steps are as described in Algorithm 2.

Algorithm 2. Synchronized sorting strategy

Input: a solution
Output: the service times for all sites
for each special construction site i **do**
  Record its possible start time u_i only considering the regular vehicles
end
Let $r_i = b_i - s_i$
Sort special construction sites in ascending order of u_i to a set SU
for each special site i in SU **do**
  if the time window constraint is not violated **then**
    Schedule it to the assigned special vehicle
  else
    Assign a new special vehicle for it
  end
end

Fig. 4(b) shows the results obtained after applying the synchronized sorting strategy, where only one special vehicle is required to serve all three special sites, thereby enhancing the performance of the solution.

3.5. Repair strategy

The second task of the decoding process is to repair infeasible solutions. The main idea of the repair strategy is to delete the construction sites whose time windows have been violated and then to insert them into available vehicles. The detailed steps of the repair strategy are given in Algorithm 3. The time complexity of this strategy is $O(n^2m)$. It should be noted that the repair strategy cannot guarantee the conversion of every infeasible solution into a feasible one; therefore, any infeasible solutions are discarded.

Algorithm 3. Repair strategy

Input: an infeasible solution
Output: a feasible solution
for each vehicle i **do**
  for each construction site j who is assigned to i **do**
    if j violates its time window **then**
      Delete construction site j from vehicle i
    end
  end
end
Store all of the deleted construction sites into a set DS
for each construction site j in DS **do**
  Try to insert construction site j into all the current vehicles by **PHIF** in Section 3.6
  if j cannot be inserted **then**
    if the number of vehicles has not been exceeded **then**
      Add a new vehicle and service construction site j
    else
      Discard the solution and stop the procedure
    end
  end
end

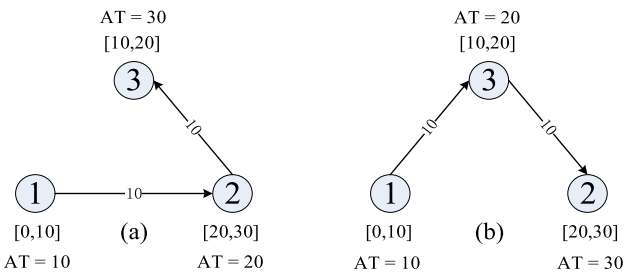


Fig. 4. Comparisons of the two cross-synchronization elimination methods.

3.6. Population initialization method

There are three efficient initialization methods, namely, the insertion heuristic by Solomon (1987), parallel insertion heuristic by Potvin and Rousseau (1993), and greedy insertion heuristic by Ioannou et al. (2001). However, these three methods always generate similar initial solutions and do not guarantee the diversity of the population. By considering the three heuristics simultaneously, an improved PFIH strategy was developed to maintain the diversity of the population. The steps of this strategy are as follows. (1) Generate three initial solutions using the PFIH, parallel, and greedy insertion heuristics separately. (2) Compute the fitness of these solutions and sort each heuristic in ascending order according to its fitness. The resulting sorted set is $HC = \{h_1, h_2, h_3\}$ with fitness values$FC = \{f_1, f_2, f_3\}$, where h_i $(i = 1, 2, 3)$ represents the ith heuristic. (3) Calculate the application times for each heuristic as follows: $AT_i = p_{size} \times \left(f_i \Big/ \sum_{j=1}^{3} f_j \right)$. (4) Apply heuristic h_i to generate AT_i solutions until the population has reached P_{size}.

3.7. Local search strategy

In the classical ABC algorithm, the local search task is implemented by the employed and onlooker bees. In the proposed IABC algorithm, a variable length local search (VLLS) strategy is utilized to enhance the exploitation abilities. The detailed steps of the VLLS are as follows.

Step 1. Randomly generate a number S_L according to the number of iterations. That is, at the first evolutionary stage, S_L is a small value, and it will increase along as the evolutionary iterations proceed. The calculation of S_L is as follows: $S_L = S_{\min} + (S_{\max} - S_{\min})*\max\left(1, \frac{I_c}{I_{\max}}\right)$, where $S_{\min}$ and $S_{\max}$ are set to 0.1 and 0.5, respectively. That is, at least 10% of construction sites are selected to perform the local search, and at most 50% are selected. I_c is the current number of iterations, and $I_{\max}$ is the maximum number of iterations.
Step 2. Randomly select one vehicle k from among all the current vehicles, and randomly select S_L construction sites for the selected vehicle. If the total number of construction sites of the selected vehicle is less than S_L, then randomly select a different vehicle and perform the selection again until enough construction sites are selected.
Step 3. Delete the selected construction sites from their assigned vehicles.
Step 4. Insert the deleted construction sites one by one into all the running vehicles. If none of the vehicles satisfies the deleted site, then dispatch a new vehicle if the total number of vehicles does not exceed the maximum number; otherwise, the solution is infeasible and should be discarded.
Step 5. Apply the repair strategy discussed in subsection 3.5 if the resulting solution is infeasible.

Fig. 5 shows a schematic of the VLLS local search strategy, where Fig. 5(a) displays the original solution representation and the randomly generated number $S_L = 2$. Then, V1 is randomly selected, and construction sites 2 and 4 are randomly selected for it. After being deleted from the assigned vehicles, the two construction sites are inserted into all of the vehicles, where the preferred insertion position for construction site 2 is the third position of the second vehicle, and the preferred insertion position selected for

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

construction site 4 is the first position of the third vehicle. The new solution obtained after applying the local search strategy is shown in Fig. 5(b).

In the proposed algorithm, the local search strategy is embedded into the employed and onlooker bee procedures.

3.8. Global search strategy

The global search of the ABC algorithm is realized by the scout bees. In the conventional ABC algorithm, the scout bee is generally chosen as a randomly generated solution. However, the useful information and knowledge of the current solution will be lost. To utilize the information collected in the previous iterations, a novel global search strategy was developed and embedded into the scout bee procedure. The detailed steps are as follows.

Step 1. Inspired by the particle swarm optimization algorithm (Kennedy and Eberhart, 1995; Marinakis et al., 2019), the best solutions *lbt i* found by the *i*th solution during the evolutionary iterations so far are recorded, where *t* is the current iteration time. The best solution found so far is recorded as gb_t.

Step 2. If there is no update of a solution *u* in the current population after L_m iterations, the formula for generating the scout bee is as follows:

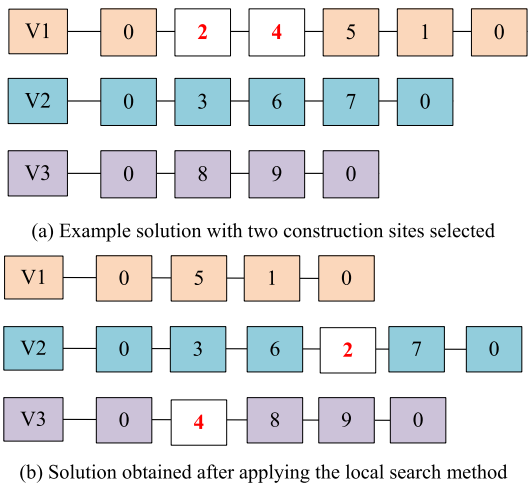
$$\eta_i^t = lb_i^t \otimes gb_t, \tag{10}$$

where the operator ⊗ represents the crossover operator with the following steps: (1) randomly generate a number *c*1 in the range [0, *K*), where *K* is the total number of vehicles; (2) randomly select *c*1 vehicles from *lbt i* and (*K*-*c*1) vehicles from gb_t and copy these selected vehicles directly to $\eta t\, i$; (3) delete the repeated construction sites occurring in the copied vehicles and record the unscheduled construction sites in a set *U*; and (4) insert each construction site U_i in *U* into the vehicles in solution $\eta t\, i$ following two criteria: if none of the vehicles satisfies U_i, then dispatch a new vehicle if the total number of vehicles does not exceed the maximum number; otherwise, the solution is infeasible and should be discarded.

4. Simulation results

4.1. Simulation instances

To reflect the constraints in the prefabricated system more effectively, we extended the canonical Solomon instances (Solomon, 1987) in the simulation test design. The extended instances include 55 instances, each of which contains 100 construction sites. The geographical data are similar to those of the canonical Solomon instances, where three types are included: (1) random instances whose labels begin with "sr1" and "sr2," (2) cluster instances denoted by "scl" and "sc2," and (3) semi-cluster instances beginning with "src 1" and "src 2." The proportions of the prefabricated and connecting components are randomly generated within the range of [1.1, 1.5]. All of the algorithms adopt the same maximum elapsed CPU time of 30 s as a termination criterion.



(a) Example solution with two construction sites selected

(b) Solution obtained after applying the local search method

Fig. 5. Example of the local search strategy.

Table 2
Parameter levels for P_s and L_m.

Parameter	Values			
	1	2	3	4
P_s	30	50	100	200
L_m	5	10	15	20

4.2. Simulation parameters

The experimental parameters included (1) the population size P_s and (2) the maximum number of non-updating iterations L_m. The levels for the two parameters are listed in Table 2.

The DOE (Design of Experiment) Taguchi method (Montgomery, 2005) was used, in which an orthogonal array L_{16} is constructed. For each parameter combination, the proposed algorithm independently ran 30 times, and the average fitness value of this algorithm was collected as the response variable (RV). Fig. 6 presents the main effects plots of the two parameters. According to the results, the proposed algorithm performs best when the population size P_s is set to 50 and L_m is 10.

4.3. Effectiveness of the proposed components

4.3.1. Efficiency of the synchronized sorting strategy

To investigate the performance of the synchronized sorting strategy, we implemented two types of IABC algorithms: one including all of the components discussed in Section 3 (hereafter

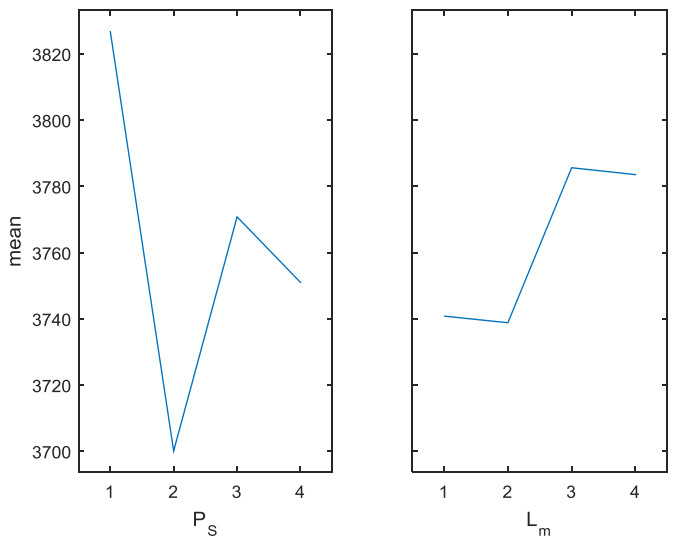


Fig. 6. Main effects of P_s and L_m.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

called the IABC algorithm) and the other with all of the other components except the synchronized sorting strategy embedded (hereafter called the IABC_NS algorithm).

Over 30 independent runs, the results obtained by the two compared algorithms were collected. The performance measure was the relative percentage increase (RPI), which is given by

$$RPI(C) = \frac{C_c - C_b}{C_b} \times 100, \quad (11)$$

where is the best solution found by all of the compared algorithms and is the best solution obtained by a specified algorithm.

To evaluate whether the difference between the two methods was significant, we performed multifactor analysis of variance (ANOVA), in which the compared methods are considered as factors. Fig. 7(a) shows the means and 95% least-significant difference (LSD) intervals for the fitness values of the two compared methods. The *p*-value is close to zero; hence, there are significant differences between the compared methods. It can be concluded that the proposed synchronized sorting strategy improves the performance of the proposed algorithm significantly. The main reason for this effect is that by sorting each special construction site according to the start times, cross-synchronization can be avoided. Furthermore, the trip times of the affected vehicles can be decreased.

4.3.2. Efficiency of the local search strategy

To verify the effectiveness of the proposed local search strategy, we performed detailed comparisons of the two algorithms, namely, the algorithm with all of the components of the proposed IABC algorithm except for the proposed VLLS method (hereafter called the IABC_CL algorithm) and the proposed IABC algorithm with all of the components. It should be noted that in the IABC_CL algorithm, the local search procedure is performed using a random method, that is, randomly selecting a random number of construction sites to delete from the current solution and then inserting them into suitable vehicles. Fig. 7(b) shows the ANOVA results obtained by comparing these strategies. It can be concluded from Fig. 7(b) that the proposed VLLS strategy can enhance the local search ability of the proposed algorithm. The main reason for this effect is that by using a variable local search length, the algorithm can adaptively change the search strength during different evolutionary iterations, that is, it can perform a detailed exploitation in the first part of the evolution and conduct a wide range of searches in the latter stage of evolution.

4.3.3. Efficiency of the global search strategy

To verify the efficiency of the proposed global search strategy further, we performed detailed comparisons of the two algorithms, namely, the algorithm with all the components except for the proposed global search component discussed in subsection 3.7 (hereafter called IABC_CG) and the proposed IABC algorithm with all of the components. It should be noted that in the IABC_CG algorithm, the global search is performed using a random method, that is, randomly generating a solution as the scout bee. Fig. 7(c) presents the ANOVA results obtained by comparing the two algorithms. It can be concluded from Fig. 7(c) that the proposed global search strategy can enhance the exploration ability of the proposed algorithm and obtain significantly better results. The main reason for this effect is that by considering the useful information collected by the local best and global best solutions found so far, the newly generated solution can enhance the search and exploration abilities.

4.4. Comparisons with TPGA and ITSA

To evaluate the performance of the IABC algorithm, the TPGA (Baniamerian et al., 2018) and ITSA (Xia and Fu, 2018) were selected for comparison. The main reasons for selecting these algorithms are as follows: (1) the TPGA algorithm was proposed to solve the VRPTW and verified to be an efficient algorithm for this type of problem and (2) the ITSA is also an efficient algorithm for solving the VRPTW. Because the E-VRPTWSyn considered in this study is an extended version of the VRPTW, the two compared algorithms can be extended to solve this problem with a simple change. Therefore, in this study, we recoded the compared TPGA and ITSA using the main components proposed in the relevant literature and adapted them to solve the E-VRPTWSyn.

After 30 independent runs, the fitness values obtained using the three compared algorithms were collected for 55 extended E-VRPTWSyn instances. The average RPI values were 0.84, 8.47, and 8.73 for the TPGA, ITSA, and IABC algorithm, respectively. The results clearly demonstrate that our algorithm yields average RPI values of about 0.099 and 0.096 times those obtained using the TPGA and ITSA, respectively. The detailed comparison results can be found in Table 1 in the Appendix.

Fig. 7(d) shows the results of multiple comparisons among the three algorithms, which clearly verify the superior performance of the proposed IABC algorithm.

4.5. Comparison with the ALNS

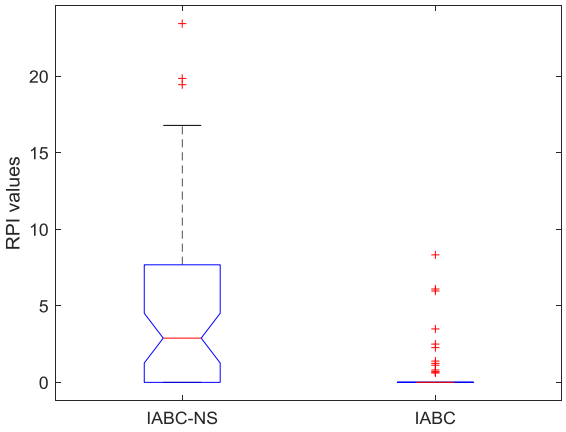
To further evaluate the performance of the proposed IABC, a comparison with the existing VRPTWSyn algorithm, the adaptive large neighborhood search (ALNS) heuristic of Liu et al. (2019), was performed. It should be noted that the ALNS heuristic is an efficient means of solving the VRPTWSyn problem. To adapt the ALNS heuristic to solve the prefabricated system problem considered, we recoded it and adapted the main components and parameters from Liu et al. (2019). In addition, for fair comparison, the synchronized sorting strategy discussed in subsection 3.3 was embedded into the ALNS algorithm. The two compared algorithms were used to perform 50 independent runs, adopting the same maximum elapsed CPU time of 50 s as the termination criterion. The minimum, maximum, and average fitness values for each instance were collected for detailed comparison.

Based on the results obtained over 30 independent runs, it can be concluded that: (1) compared with the ALNS heuristic, the proposed IABC algorithm obtained 39 better values out of the considered 55 benchmarks, which is better than the ALNS algorithm; (2) the IABC algorithm yielded an average RPI value of 1.07, which is approximately 1/7 of that resulting from using the ALNS algorithm; and (3) the ANOVA results in Fig. 7(e) and (f) demonstrate that the IABC algorithm shows significantly better performance than the ALNS. The detailed comparison results can be found in Table 2 the Appendix.

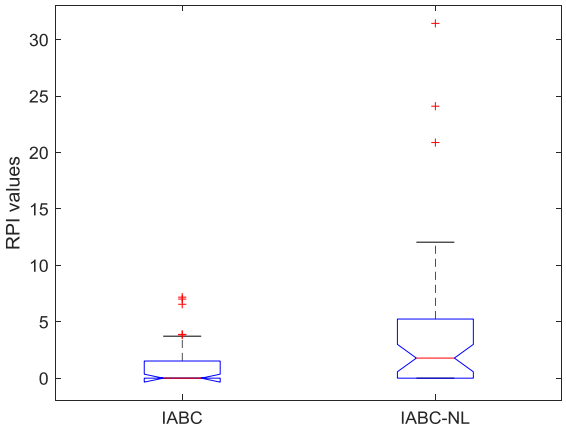
Furthermore, to demonstrate the convergence ability of the proposed algorithm, we conducted detailed comparisons between the IABC and ALNS algorithms, and the convergence curves for the four instances with different features are depicted in Fig. 8. The four curves of the selected instances, sc102, sr103, sr202, and src 103, are shown in Fig. 8(a), (b), (c), and (d), respectively. The convergence curves for different types of instances demonstrate that the proposed algorithm has efficient convergence abilities for solving the E-VRPTWSyn.

Fig. 9 shows the construction site service time Gantt diagram of the sc101 instance, where “V1” represents the first vehicle, each rectangular frame corresponds to a construction site, and the number in the rectangular frame is the construction site number. For example, the construction sites for the first car service are {20, 24, 25, 27, 30, 28, 26, 23, 22, 21}, for a total of 10 construction sites. The two numbers below each construction site specify the times at

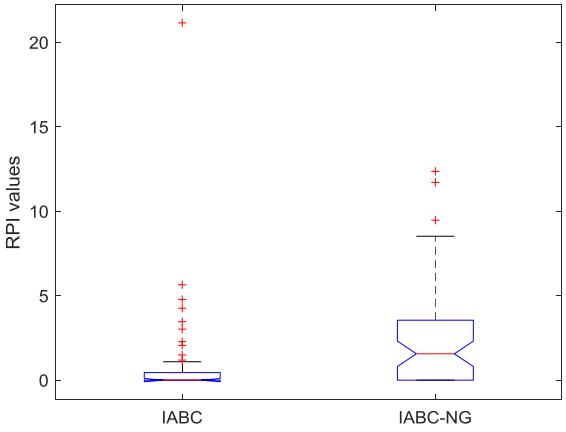
Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464



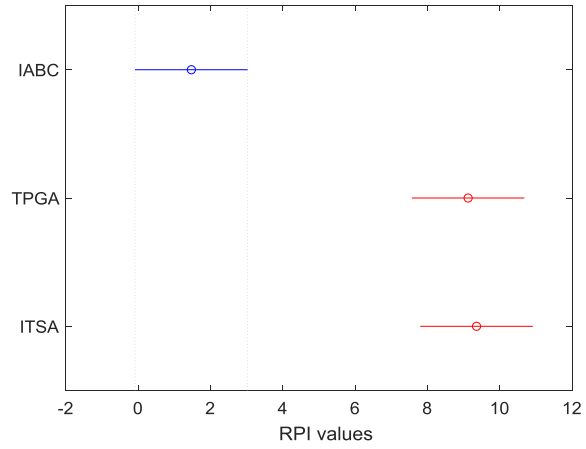
(a) ANOVA of the synchronized sorting strategy



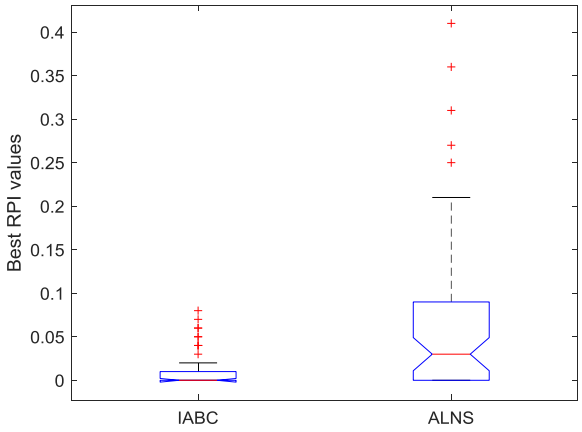
(b) ANOVA of the local search strategy



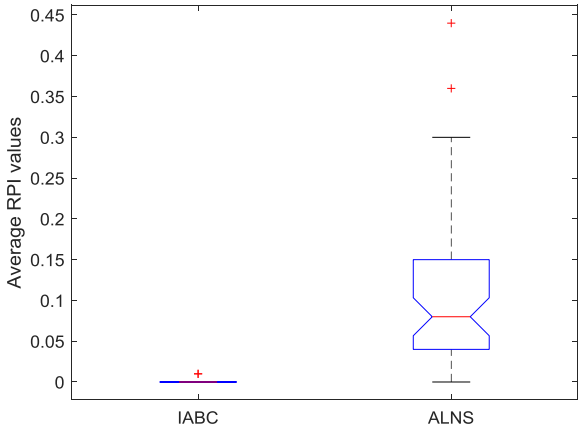
(c) ANOVA of the global search strategy



(d) ANOVA of the IABC algorithm, TPGA, and ITSA



(e) ANOVA of the minimum performance of the IABC and ALNS algorithms



(f) ANOVA of the average performance of the IABC and ALNS-III algorithms

Fig. 7. Means and 95% LSD intervals for the compared algorithms.

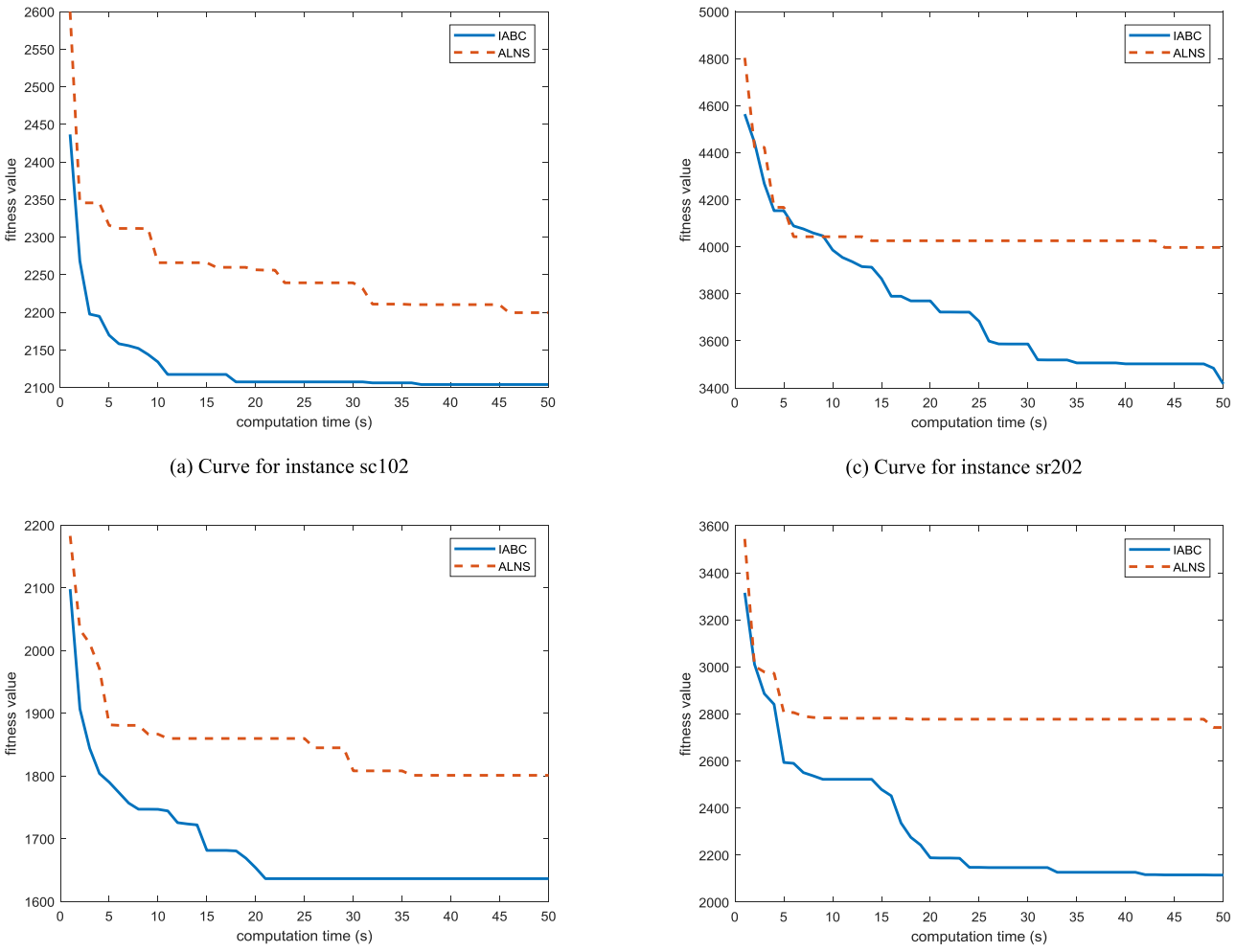
Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

which service begins and ends, respectively, at the construction site point. For instance, the start and end service times of construction site 21 are 915 and 1007, respectively. The service start time of each client point is within the specified service time window, and the scheduling scheme is feasible and effective. Meanwhile, special construction sites begin on V15—V22, where each construction site has the same service start and completion times for the corresponding regular vehicles. For example, construction site 35 has the same service start and end times for special and regular vehicles. Figs. 10 and 11 show the charts for the instance sc101, which verifiethat the proposed algorithm is effective

5. Conclusions

Dispatching components in prefabricated systems is challenging, yet efficient scheduling can significantly improve the energy, cost, and time efficiency of construction. Thus, it is important to design an optimization algorithm for the dispatching problem. However, the existing research has generally been limited to optimizing the dispatching problem without considering the constraints specific to prefabricated systems, creating a gap between prefabricated construction and VRP algorithms. Therefore, in this study, we first modeled the dispatching problem as an extension of the classical VRPTW, denoting it as E-VRPTWSyn. Two constraints were considered simultaneously in the mathematical model, i.e., the synchronized visits of regular and special vehicles, and the different capacities of the two types of components. To solve the cross-synchronization problem that occurs in prefabricated systems, two problem-specific lemmas were defined. An IABC algorithm was developed to solve the complex optimization problem. In this algorithm, each solution is represented by using a two-dimensional vector with a flag vector. Then, to decode each solution for feasible scheduling, we designed a repair strategy. Based on the encoding and decoding approach, we also developed a novel initialization heuristic that can generate a diverse solution population. After initialization, the algorithm goes through numerous iterations. In each evolution, an efficient local search method is used by both the employed and onlooker bee procedures. Then, if some solutions have not been updated after a certain number of iterations, a scout bee is generated to replace one of these solutions, where the proposed global search strategy is stimulated. The best solution found so far is updated at each evolution. Finally, when the stop condition is satisfied, the best solution becomes the last schedule for the system.

To verify the performance of our algorithm, we selected three efficient algorithms, the TPGA, ITSA, and ALNS algorithm, to conduct detailed simulation tests. Because this is the first study in which the dispatching problem in a prefabricated system has been



(a) Curve for instance sc102

(c) Curve for instance sr202

(b) Curve for instance sr103

(d) Curve for instance src103

Fig. 8. Comparisons of the convergence curve between the IABC and ALNS algorithms.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

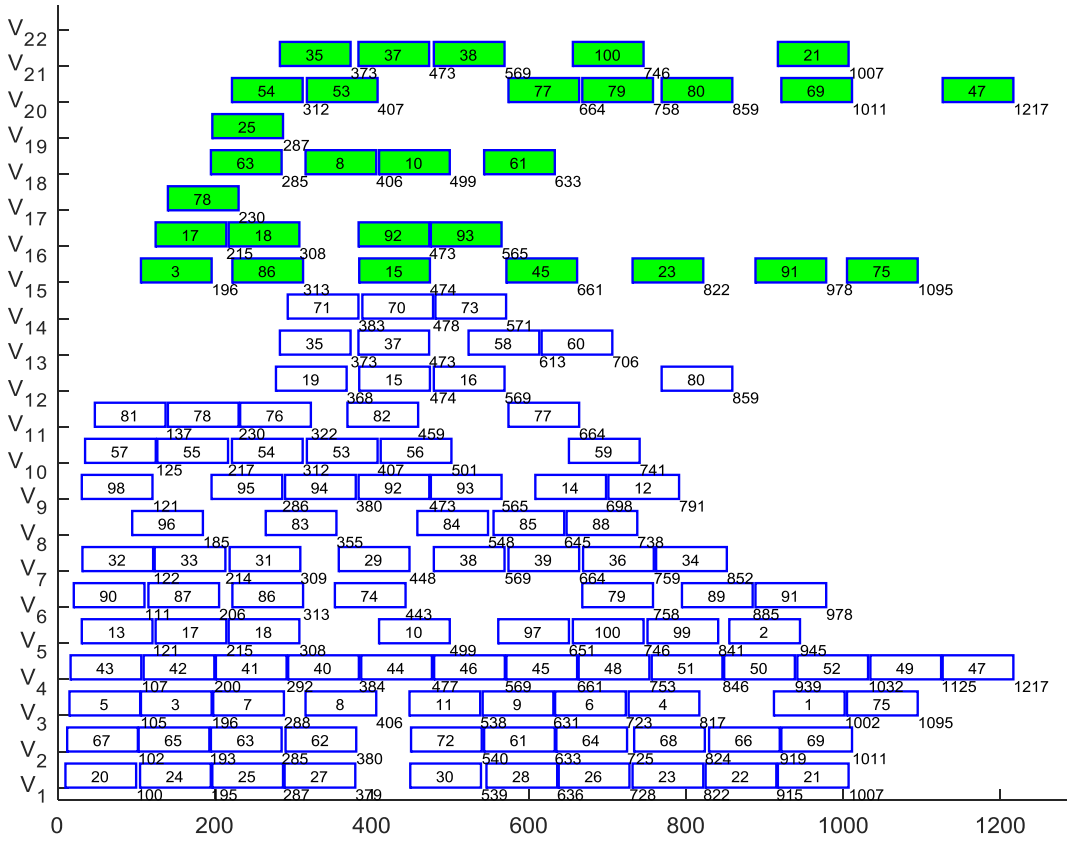


Fig. 9. Gantt chart for the best individual for sc101.

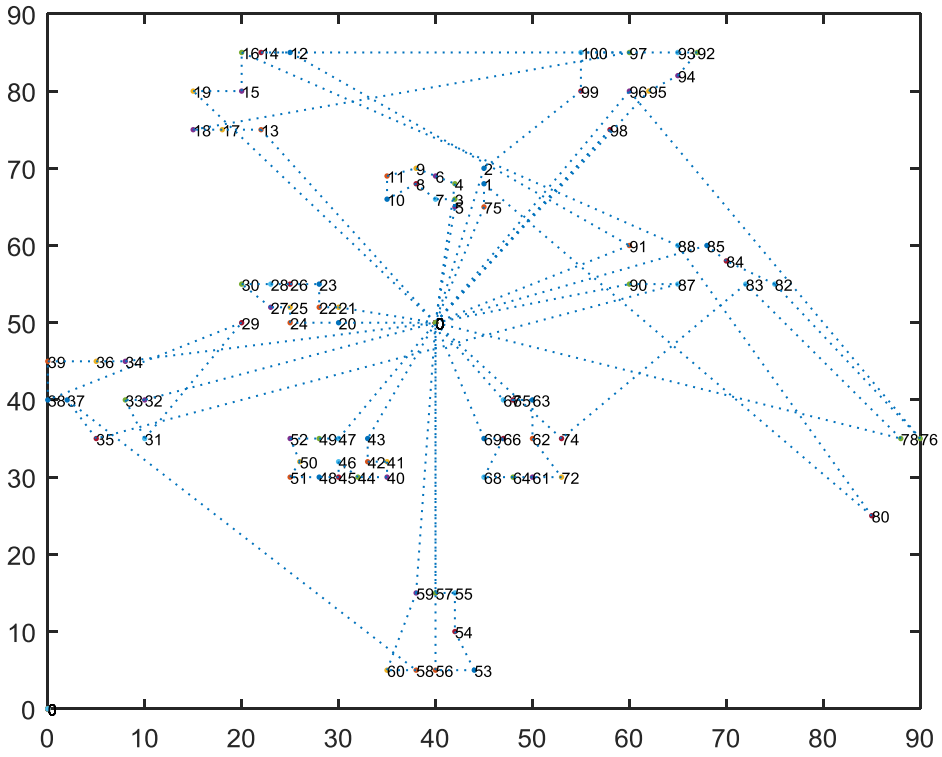


Fig. 10. Chart for sc101 without consideration of the synchronized visits.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

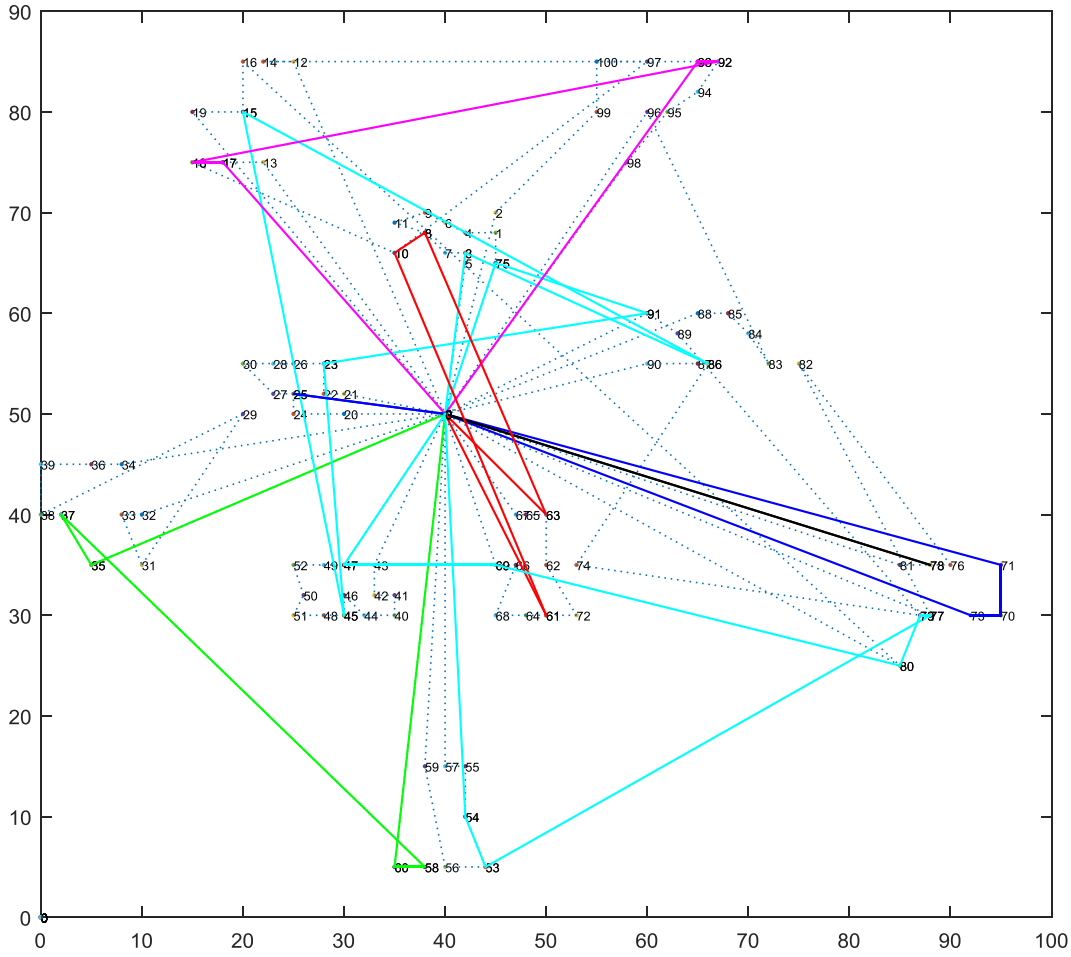


Fig. 11. Chart for sc101 considering synchronized visits.

addressed as an extension of the VPRTW, there is no benchmark to test. Therefore, we extended the canonical Solomon benchmarks and generated 55 instances for the prefabricated system. For each instance, 30 independent runs were performed with the four compared algorithms and the average fitness values were used to calculate the PRI values. The simulation results showed that, considering the energy consumption metric, our algorithm yields average deviations of about 0.099, 0.096, and 0.143 times the values obtained using the TPGA, ITSA, and ALNS algorithm, respectively. The multifactor ANOVA results and convergence curve comparisons also verified the efficiency of the proposed algorithm. In summary, our algorithm is a competitive means of solving the dispatching problem in prefabricated systems. In the future, we will focus on following aspects: (1) to consider the interval dispatching time (Sun et al., 2020) and design a multi-objective framework with two or more objectives; and (2) considering the other constraints such as economic dispatch optimization (Zheng et al., 2020), and apply the proposed algorithm to solve VRPTW problems with more realistic constraints.

Author contributions

1. Jun-qing Li's contributions are as follows: (1) proposed an improved artificial bee colony algorithm for the considered problem; (2) design several heuristics for the problem; (3) design the simulation tests; and (4) make a detailed analysis for the results;

2. Yun-qi Han's contributions are as follows: (1) code the compared algorithms and test the simulation; (2) design several heuristics for the problem; and (3) test all the compared algorithms.

3. Pei-yong Duan's contributions are as follows: (1) give a detailed analysis of the prefabricated system; (2) propose the VRP problem in the prefabricated system; and (3) analyzed the algorithm's performance.

4. Yu-yan Han's contributions are as follows: (1) analyze the problem features and proposed problem-specific heuristic; and (2) design the algorithm parameter and test them.

5. Ben Niu modeled the dispatching problem as a special type of vehicle routing problem with time windows (VRPTW).

6. Cheng-dong Li's contributions are as follows: (1) give detailed report of the related literatures; and (2) written the problem descriptions and examples.

7. Zhi-xin Zheng's contributions are as follows: (1) verify the mathematical model; and (2) recode and test the compared algorithms.

8. Yi-ping Liu's contributions are as follows: (1) give a detailed comparison analysis of different compared algorithms; and (2) give a carefully check for the grammar in the paper.

Declaration of competing interest

No conflict of interest exits in the submission of this manuscript.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

Acknowledgements

This research was partially supported by National Science Foundation of China (61773192, 61803192, and 61773246), Shandong Province Higher Educational Science and Technology Program (J17KZ005), the State Key Laboratory of Synthetical Automation for Process Industries (PAL-N201602), the Special Fund for Local Science and Technology Development Lead by Central Authority and Major Basic Research Projects in Shandong (ZR2018ZB0419), and a Grant of Key Laboratory of Intelligent Optimization and Control with Big Data.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jclepro.2019.119464.

Appendix

Table 1
Comparison results for the IABC, TPGA, and ITSA

Inst	Best	Fitness			PRI		
		IABC	TPGA +	ITSA #	IABC	TPGA	ITSA
sc101	3869.28	**3869.28**	3877.37	3960.49	**0.00**	0.21	2.36
sc102	2934.22	**2934.22**	3261.23	3522.41	**0.00**	11.14	20.05
sc103	2287.22	**2287.22**	2534.54	3053.41	**0.00**	10.81	33.50
sc104	1715.12	**1715.12**	1968.68	2380.99	**0.00**	14.78	38.82
sc105	2777.75	**2777.75**	2929.4	3372.51	**0.00**	5.46	21.41
sc106	2876.93	**2876.93**	3044.09	3351.22	**0.00**	5.81	16.49
sc107	2295.35	**2295.35**	2331.89	2699.19	**0.00**	1.59	17.59
sc108	2668.26	**2668.26**	2914.71	3462.17	**0.00**	9.24	29.75
sc109	2351.18	**2351.18**	2642.34	3340.11	**0.00**	12.38	42.06
sc201	3773.74	**3773.74**	3807.31	4330.38	**0.00**	0.89	14.75
sc202	4663.5	**4663.5**	4750.42	4857.97	**0.00**	1.86	4.17
sc203	2783.45	**2783.45**	2923.51	3182.62	**0.00**	5.03	14.34
sc204	2323.17	**2323.17**	2382.59	2460.57	**0.00**	2.56	5.91
sc205	4758.71	**4758.71**	4903.25	4822.64	**0.00**	3.04	1.34
sc206	2330.74	**2330.74**	2734.02	3065.03	**0.00**	17.30	31.50
sc207	3322.8	**3322.8**	3382.97	4149.59	**0.00**	1.81	24.88
sc208	2052.41	**2052.41**	2268.74	2515.48	**0.00**	10.54	22.56
sr101	2905.33	2905.33	2948.03	2908.77	**0.00**	1.47	0.12
sr102	2490.2	**2490.2**	2541.96	2512.05	**0.00**	2.08	0.88
sr103	1636.34	**1636.34**	1702.87	1682.55	**0.00**	4.07	2.82
sr104	1551.98	**1551.98**	1704.8	1576.49	**0.00**	9.85	1.58
sr105	2198.63	2227.56	**2198.63**	2303.65	1.32	**0.00**	4.78
sr106	1680.91	**1680.91**	1742.69	1736.36	**0.00**	3.68	3.30
sr107	1560.18	**1560.18**	1665.98	1618.37	**0.00**	6.78	3.73
sr108	1226.43	**1226.43**	1406.81	1337.26	**0.00**	14.71	9.04
sr109	1555.38	**1555.38**	1642.74	1609.92	**0.00**	5.62	3.51
sr110	1461.62	**1461.62**	1575.79	1482.46	**0.00**	7.81	1.43
sr111	1667.42	**1667.42**	1726.75	1707.62	**0.00**	3.56	2.41
sr112	1560.68	**1560.68**	1732.2	1710.52	**0.00**	10.99	9.60
sr201	2301.45	2377.39	2558.21	**2301.45**	3.30	11.16	**0.00**
sr202	2000.8	**2000.8**	2370.5	2279.32	**0.00**	18.48	13.92
sr203	1566.6	1687.57	1812.87	**1566.6**	7.72	15.72	**0.00**
sr204	1431.84	**1431.84**	1694.65	1499.49	**0.00**	18.35	4.72
sr205	1694.38	1697.85	1805.34	1694.38	0.20	6.55	**0.00**
sr206	1304.87	**1304.87**	1507.47	1428.26	**0.00**	15.53	9.46
sr207	1738.97	**1738.97**	1926.29	1773.41	**0.00**	10.77	1.98
sr208	1142.23	**1142.23**	1294.97	1219.15	**0.00**	13.37	6.73
sr209	1843.57	**1843.57**	2010.9	1895.89	**0.00**	9.08	2.84
sr210	1503.52	**1503.52**	1708.97	1516.9	**0.00**	13.66	0.89
sr211	1228.57	**1228.57**	1441.05	1335.28	**0.00**	17.29	8.69
src 101	2583.64	2684.03	2656.82	**2583.64**	3.89	2.83	**0.00**
src 102	2295.58	**2295.58**	2361.17	2431.17	**0.00**	2.86	5.91
src 103	1860.58	**1860.58**	1983.92	1957.19	**0.00**	6.63	5.19
src 104	1458.4	**1458.4**	1648.97	1580.65	**0.00**	13.07	8.38
src 105	2335.58	2371.61	2408.5	**2335.58**	1.54	3.12	**0.00**
src 106	1864.18	1874.15	1929.45	**1864.18**	0.53	3.50	**0.00**
src 107	1735.4	1878.09	1907.97	**1735.4**	8.22	9.94	**0.00**
src 108	1604.01	1698.04	1728.31	**1604.01**	5.86	7.75	**0.00**
src 201	2384.99	2431.23	2545.1	**2384.99**	1.94	6.71	**0.00**
src 202	2222.58	**2222.58**	2704.34	2425.09	**0.00**	21.68	9.11
src 203	1964.37	**1964.37**	2104.82	2040.02	**0.00**	7.15	3.85
src 204	1283.49	1435.92	1542.01	**1283.49**	11.88	20.14	**0.00**
src 205	2582.73	**2582.73**	2815.98	2586.46	**0.00**	9.03	0.14
src 206	1558.58	**1558.58**	1809.4	1650.55	**0.00**	16.09	5.90
src 207	1976.35	**1976.35**	1985.72	2127.66	**0.00**	0.47	7.66
mean	2159.02	**2173.77**	2318.36	2360.24	**0.84**	8.47	8.73

+: **TPGA** (Baniamerian et al., 2018).
#: ITSA (Xia and Fu, 2018).

Table 2
Comparison results for the IABC and ALNS

Inst	Best	IABC ALNS						dev	
		min	max	avg	min	max	avg	IABC ALNS	
sc101	3700.79	3869.28	4417.78	4219.96	**3700.79**	4554.57	4176.95	4.55	**0.00**
sc102	2934.22	**2934.22**	3442.3	3218.79	3199.39	4256.35	3685.27	**0.00**	9.04
sc103	2287.22	**2287.22**	2718.12	2515.36	2913.04	3904.87	3280.74	**0.00**	27.36
sc104	1715.12	**1715.12**	2122.78	1886.81	2417.24	2885.49	2604.49	**0.00**	40.94
sc105	2777.75	**2777.75**	3192.05	2962.4	3021.29	3773.56	3480.73	**0.00**	8.77
sc106	2876.93	**2876.93**	3419.57	3163.92	3334.37	3996.76	3642.47	**0.00**	15.90
sc107	2295.35	**2295.35**	2737.54	2518.31	2618.27	3373.89	3020.93	**0.00**	14.07
sc108	2668.26	**2668.26**	3267.01	2961.77	3231.65	3975.89	3635.32	**0.00**	21.11
sc109	2351.18	**2351.18**	2988.13	2637.33	3196.88	3638.12	3394.77	**0.00**	35.97
sc201	3773.74	**3773.74**	4264.3	3939.55	4525.14	5529.74	5011.4	**0.00**	19.91
sc202	4663.5	**4663.5**	5202.57	4918.52	4990.03	5682.85	5277.16	**0.00**	7.00
sc203	2783.45	**2783.45**	3562.72	3216.75	3208.14	4362.89	3748.32	**0.00**	15.26
sc204	2323.17	**2323.17**	2543.26	2411.48	2403.26	2864.46	2661.99	**0.00**	3.45
sc205	4758.71	**4758.71**	5205	5022.45	4902.85	5522.55	5233.68	**0.00**	3.03
sc206	2330.74	**2330.74**	3366.7	2867.26	3049.78	3882.92	3472.23	**0.00**	30.85
sc207	3322.8	**3322.8**	4111.25	3662.66	4018.29	4931.01	4435.52	**0.00**	20.93
sc208	2052.41	**2052.41**	3048.84	2503.14	2568.33	3589.51	2985.21	**0.00**	25.14
sr101	2819.26	2905.33	3059.07	2978.69	**2819.26**	3042.72	2940.37	3.05	**0.00**
sr102	2467.28	2490.2	2720.33	2638.19	**2467.28**	2826.22	2637.75	0.93	**0.00**
sr103	1620.87	1636.34	1844.2	1753	**1620.87**	1929.33	1797.69	0.95	**0.00**
sr104	1551.98	**1551.98**	1792.51	1688.29	1583.81	1850.21	1730.58	**0.00**	2.05
sr105	2175.41	2227.56	2391.6	2325.23	**2175.41**	2466.71	2308.81	2.40	**0.00**
sr106	1680.91	**1680.91**	1867.64	1788.92	1732.42	1972.43	1878.66	**0.00**	3.06
sr107	1560.18	**1560.18**	1733.67	1664.41	1565.7	1921.15	1741.54	**0.00**	0.35
sr108	1226.43	**1226.43**	1483.5	1394.58	1304.42	1687.52	1474.33	**0.00**	6.36
sr109	1555.38	**1555.38**	1761.1	1676.37	1594.2	1851.28	1704.34	**0.00**	2.50
sr110	1461.62	**1461.62**	1716.5	1596.98	1479.88	1767.18	1620.25	**0.00**	1.25
sr111	1667.42	**1667.42**	1811.35	1737.74	1727.41	1968.42	1825.2	**0.00**	3.60
sr112	1560.68	**1560.68**	1872.13	1710.64	1667.72	2004.5	1851	**0.00**	6.86
sr201	2224.45	2377.39	2783.69	2597.02	**2224.45**	2967.96	2621.53	6.88	**0.00**
sr202	2000.8	**2000.8**	2535.59	2287.6	2178.61	2742.45	2444.49	**0.00**	8.89
sr203	1687.57	**1687.57**	2053.72	1857.66	1743.5	2355.32	1969.72	**0.00**	3.31
sr204	1431.84	**1431.84**	1855.27	1650.95	1512.88	1932.89	1758.1	**0.00**	5.66
sr205	1628.96	1697.85	2197.72	1980.61	**1628.96**	2267.43	1961.37	4.23	**0.00**
sr206	1304.87	**1304.87**	1672.13	1454.53	1397.29	1890.43	1618.97	**0.00**	7.08
sr207	1738.97	**1738.97**	2223.94	1966.01	1900.22	2288.42	2079.28	**0.00**	9.27
sr208	1142.23	**1142.23**	1466.53	1332.65	1227.9	1509.43	1392.99	**0.00**	7.50
sr209	1755.61	1843.57	2307.3	2066.49	**1755.61**	2284.68	2053.21	5.01	**0.00**
sr210	1503.52	**1503.52**	1859.81	1682.76	1553.8	1977.52	1745.78	**0.00**	3.34
sr211	1228.57	**1228.57**	1624.6	1460.83	1270.51	1850.54	1497.56	**0.00**	3.41
src 101	2572.37	2684.03	2908	2794.41	**2572.37**	2912.65	2752.62	4.34	**0.00**
src 102	2295.58	**2295.58**	2487.37	2385	2305.7	2801.68	2452.13	**0.00**	0.44
src 103	1860.58	**1860.58**	2142.87	2032.82	1889.17	2251.1	2099.89	**0.00**	1.54
src 104	1458.4	**1458.4**	1772.33	1640.6	1535.16	2114.93	1712.03	**0.00**	5.26
src 105	2186.78	2371.61	2516.13	2451.18	**2186.78**	2545.9	2406.02	8.45	**0.00**
src 106	1802.3	1874.15	2080.21	2001.9	**1802.3**	2210.8	2001.62	3.99	**0.00**
src 107	1765.71	1878.09	2124.06	1971.81	**1765.71**	2143.93	1943.66	6.36	**0.00**
src 108	1606.97	1698.04	1876.2	1783.6	**1606.97**	1910.5	1767.54	5.67	**0.00**
src 201	2412.98	2431.23	2808.98	2542.64	**2412.98**	2960.67	2703.62	0.76	**0.00**
src 202	2222.58	**2222.58**	3013.47	2714.48	2636.47	3241.29	2831.06	**0.00**	18.62
src 203	1964.37	**1964.37**	2343.54	2177.98	1978.52	2567.78	2292.45	**0.00**	0.72
src 204	1426.37	1435.92	1693.27	1554.46	**1426.37**	1942.4	1657.78	0.67	**0.00**
src 205	2570.99	2582.73	2985.12	2807.02	**2570.99**	3224.79	2841.63	0.46	**0.00**
src 206	1558.58	**1558.58**	2159.41	1820.06	1683.9	2317.54	1906.9	**0.00**	8.04
src 207	1976.35	**1976.35**	2468.01	2207.63	2038.71	2588.83	2326.55	**0.00**	3.16
mean	2150.75	**2173.77**	2574.96	2378.22	2324.42	2869.36	2583.57	**1.07**	7.47

References

Afifi, S., Dang, D.C., Moukrim, A., 2016. Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. Opt. Lett. 10 (3), 511–525.

Arnold, F., Sörensen, K., 2019. Knowledge-guided local search for the vehicle routing problem. Comput. Oper. Res. 105, 32–46.

Atkinson, J.B., 1994. A greedy look-ahead heuristic for combinatorial optimization: an application to vehicle scheduling with time windows. J. Oper. Res. Soc. 45 (6), 673–684.

Aye, L., Ngo, T., Crawford, R.H., Gammampila, R., Mendis, P., 2012. Life cycle greenhouse gas emissions and energy analysis of prefabricated reusable building modules. Energy Build. 47, 159–168.

Baniamerian, A., Bashiri, M., Zabihi, F., 2018. Two phase genetic algorithm for vehicle routing and scheduling problem with cross-docking and time windows considering customer satisfaction. J. Ind. Eng. Int. 14 (1), 15–30.

Bräysy, O., Gendreau, M., 2005a. Vehicle routing problem with time windows, Part I: route construction and local search algorithms. Transp. Sci. 39 (1), 104–118.

Bräysy, O., Gendreau, M., 2005b. Vehicle routing problem with time windows, Part II: Metaheuristics. Transp. Sci. 39 (1), 119–139.

Decerle, J., Grunder, O., El Hassani, A.H., Barakat, O., 2019. A hybrid memetic-ant colony optimization algorithm for the home health care problem with time window, synchronization and working time balancing. Swarm Evol. Comput. https://doi.org/10.1016/j.swevo.2019.02.009.

Han, Y., Gong, D., Sun, X., 2015. A discrete artificial bee colony algorithm incorporating differential evolution for flow shop scheduling problem with blocking. Eng. Optim. 47, 927–946.

Hojabri, H., Gendreau, M., Potvin, J.Y., Rousseau, L.M., 2018. Large neighborhood search with constraint programming for a vehicle routing problem with synchronization constraints. Comput. Oper. Res. 92, 87–97.

Ioannou, G., Kritikos, M., Prastacos, G., 2001. A greedy look-ahead heuristic for the

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464

vehicle routing problem with time windows. J. Oper. Res. Soc. 52 (5), 523–537.
Jaillon, L., Poon, C.S., 2009. The evolution of prefabricated residential building systems in Hong Kong: a review of the public and the private sector. Autom. ConStruct. 18 (3), 239–248.
Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization, vol. 200. Erciyes University, Engineering Faculty, Computer Engineering Department. Technical report-tr06.
Kennedy, J., Eberhart, R., 1995. Particle swarm optimization (PSO). In: Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948.
Li, J.Q., Bai, S.C., Duan, P.Y., Sang, H.Y., Han, Y.Y., Zheng, Z.X., 2019a. An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system. Int. J. Prod. Res. 57, 6922–6942.
Li, J.Q., Tao, X.R., Jia, B.X., Han, Y.Y., Liu, C., Duan, P., Zheng, Z.X., Sang, H.Y., 2019b. Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots. Swarm Evol. Comput. https://doi.org/10.1016/j.swevo.2019.100600.
Li, J.Q., Han, Y.Q., 2020b. A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in Cloud computing system. Clust. Comput. 23, 1–15. https://doi.org/10.1007/s10586-019-03022-z.
Li, J.Q., Song, M.X., Wang, L., Duan, P.Y., Han, Y.Y., Sang, H.Y., Pan, Q.K., 2020a. Hybrid artificial bee colony algorithm for a parallel batching distributed flow shop problem with deteriorating jobs. IEEE Trans. Cybernet. https://doi.org/10.1109/TCYB.2019.2943606.
Liu, H., Xu, B., Lu, D., Zhang, G., 2018. A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. Appl. Soft Comput. 68, 360–376.
Liu, R., Tao, Y., Xie, X., 2019. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. Comput. Oper. Res. 101, 250–262.
Marinakis, Y., Marinaki, M., Migdalas, A., 2019. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. Inf. Sci. 481, 311–329.
Montgomery, D.C., 2005. Design and Analysis of Experiments. John Wiley & Sons, Arizona.
Ng, K.K.H., Lee, C.K.M., Zhang, S.Z., Wu, K., Ho, W., 2017. A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion. Comput. Ind. Eng. 109, 151–168.
Potvin, J.Y., Rousseau, J.M., 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. Eur. J. Oper. Res. 66 (3), 331–340.
Pan, Q.K., Gao, L., Li, X.Y., Framinan, M., 2019. Effective constructive heuristics and meta-heuristics for the distributed assembly permutation flowshop scheduling problem. Appl. Soft Comput. 81, 105492.
Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.
Sun, J., Miao, Z., Gong, D., Zeng, X.J., Li, J., Wang, G., 2020. Interval multiobjective optimization with memetic algorithms. IEEE Trans. Cybernet. https://doi.org/10.1109/TCYB.2019.2908485.
Wang, L., Zhou, G., Xu, Y., Liu, M., 2012. An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. Int. J. Adv. Manuf. Technol. 60 (9–12), 1111–1123.
Xia, Y., Fu, Z., 2018. Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate. Clust. Comput. https://doi.org/10.1007/s10586-018-1957-x.
Zheng, Z.X., Li, J.Q., Han, Y.Y., 2020. An improved invasive weed optimization algorithm for solving dynamic economic dispatch problems with valve-point effects. J. Exp. Theor. Artif. Intell. https://doi.org/10.1080/0952813X.2019.1673488.

Please cite this article as: Li, J.-q et al., Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems, Journal of Cleaner Production, https://doi.org/10.1016/j.jclepro.2019.119464