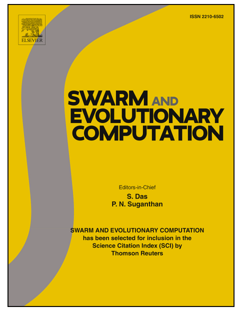
Journal Pre-proof



An Improved Ant Colony Optimization algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice

Yuan Wang, Ling Wang, Guangcai Chen, Zhaoquan Cai, Yongquan Zhou, Lining Xing

PII: S2210-6502(18)30943-X

DOI: https://doi.org/10.1016/j.swevo.2020.100675

Reference: SWEVO 100675

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 19 November 2018

Revised Date: 13 February 2020

Accepted Date: 2 March 2020

Please cite this article as: Y. Wang, L. Wang, G. Chen, Z. Cai, Y. Zhou, L. Xing, An Improved Ant Colony Optimization algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice, *Swarm and Evolutionary Computation BASE DATA* (2020), doi: https://doi.org/10.1016/j.swevo.2020.100675.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier B.V.

Yuan Wang：Conceptualization，Methodology，Software，Writing - Original Draft，Investigation，Formal analysis，Visualization；

Ling Wang：Validation，Writing - Review & Editing，Supervision，Funding acquisition

Guangcai Chen：Investigation，Formal analysis，Visualization；

Zhaoquan Cai：Investigation，Formal analysis；

Yongquan Zhou：Resources，Data Curation；

Lining Xing*：Conceptualization，Validation，Writing - Review & Editing，Supervision，Project administration，Funding acquisition.

Journal Pre-proof

Journal Pre-proof

An Improved Ant Colony Optimization Algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice

Yuan Wang[a], Ling Wang[b], Guangcai Chen[c], Zhaoquan Cai[d], Yongquan Zhou[e], Lining Xing[a,*]

[a]College of Systems Engineering, National University of Defense Technology, Changsha, 410073, P.R. China
[b]Department of Automation, Tsinghua University, Beijing 100084, P.R. China
[c]School of Automation, Guangdong University of Technology, Guangzhou 510006, P.R. China
[d]Department of Information Science and Technology, Huizhou University, Huizhou 516007, P.R. China
[e]College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, 530006, P.R. China

Journal Pre-proof

Abstract

This article addresses a Periodic Vehicle Routing Problem with Time Window and Service Choice problem. This problem is basically a combination of existing Periodic Vehicle Routing Problem with Time Window and Periodic Vehicle Routing Problem with Service Choice. We model it as a multi objective problem. To solve this problem, we develop a heuristic algorithm based on Improved Ant Colony Optimization (IACO) and Simulate Annealing (SA) called Multi Objective Simulate Annealing - Ant Colony Optimization (MOSA-ACO). Improvements are made in following respects: a) a Euclidean distance based solution acceptance criterion is developed; b) a parameter control pattern is designed to generate different initial solutions; c) several local search strategies are added. Benchmark instances generated from Solomon's benchmark instances and Cordeau's benchmarks instances are applied. Comparison algorithms include four population based heuristics and IACO. Computation experiment results show that MOSA-ACO algo-

*Corresponding author.
[1]E-mail address: xinglining@gmail.com

Journal Pre-proof

rithm has a good performance on solving this problem.

Keywords:
Ant Colony Optimization, Multi Objective Optimization, Periodic Vehicle Routing Problem with Time Window, Simulate Annealing, Service Choice

1. Introduction

In this article, we propose a special variant of existing Periodic Vehicle Routing Problem (PVRP) called Periodic Vehicle Routing Problem with Time Window and Service Choice (PVRPTW-SC). We give out a problem model which is an extension of PVRP-SC problem model. Then, a hybrid heuristic algorithm called Multi-Objective Simulation Annealing - Ant Colony Optimization (MOSA-ACO) is applied to this problem. Finally, computation experiment results are reported. Experiment instances include instances generated from VPRTW benchmark instances and PVRPTW benchmark instances. Experiment results show that the MOSA-ACO has a good performance on PVRPTW-SC.

This research is first inspired by Jiting et al. [1]. In this paper, a geostationary orbit (GEO) satellite observation planning problem is introduced. The GEO satellite observation planning problem includes a GEO satellite and several targets on earth to be visited. The aim of GEO satellite observation planning problem is to find an observation plan with lowest cost and highest profit. To solve this problem, a NSGA II based algorithm and a Multi Objective Travelling Salesman Problem (MOTSP) model are implemented. In this paper, we extend this problem in several directions: a) in PVRPTW-SC problem, multi trips in a planning horizon are considered; b) time window constraint is considered; c) a changeable visit frequency of each customer in planning horizon is considered. The PVRPTW-SC problem mainly focus on discussing the influence of changeable visit frequency in PVRPTW scenarios. To give a detailed description of PVRPTW-SC problem, we give out a real-world scenario.

Suppose that there is a beverage warehouse serving n shops in a block and this warehouse has m delivery men. Every week, this warehouse must make a service plan which assigns the distribution routes of each day. Every shop in this block has its unique service preference combinations (2 days a week, for example). Because the daily beverage sales amount of a shop is uncertain, some of these shops may face lacking of beverage stocks even though

Journal Pre-proof

the beverage warehouse follows its distribution plan strictly and all customer preferences are satisfied. Thus, to solve this problem, besides maintaining the lowest visit frequency of every shop, which is indicated by the service preference combinations, the manager of this warehouse can choose to add extra service to any of these shops to earn extra profits. The problem objectives of PVRPTW-SC include visit frequency maximization, travelling distance minimization and fleet size minimization.
The PVRPTW-SC problem is basically a combination of two Periodic Vehicle Routing Problem (PVRP): Periodic Vehicle Routing Problem with Time Window (PVRPTW) and Periodic Vehicle Routing Problem with Service Choice (PVRP-SC). PVRP is a well-known combination optimization problem which is first proposed in Beltrami and Bodin [2]. The main variants of PVRP problem can be seen in Fig.1:

To our best knowledge, this is the first article that studies PVRPTW-SC

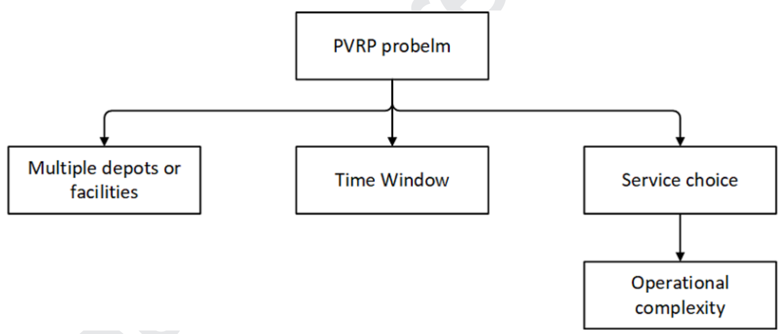


Figure 1: PVRP problem and its variants (Francis et al. [3])

problem. So we give out a brief introduction of articles that studies PVRP-SC and PVRPTW.
As far as we know, articles that study PVRP-SC problem and its variants are rare. The first article that proposes PVRP-SC problem is Francis et al.[4]. In this article, the authors give out a working scenario of interlibrary loan item management in North Suburban Library System. The system manager must decide the visit frequency of each node and daily route plan. The problem constraints include minimum visit frequency of each customer and vehicle capacity constraint. To solve this problem, the authors first decompose PVRP-SC problem into a TSP with Profit sub-problem which deals with the routing problem and a "knapsack" type problem which deals with the vehicle capacity constraint using Lagrangian Relaxation technology. Then a

Journal Pre-proof

branch and bound algorithm is applied to solve it. Another article that studies PVRP-SC variants is Francis et al.[5]. In this article, the authors give out several kinds of operation complexities other than service choice (e.g. schedule options, delivery strategies and arrival span). Tabu search heuristics as well as technologies used in Francis et al.[4] are applied and evaluated.

PVRPTW, which is first proposed in Laporte and Mercier[6] in 2004, is another important variant of PVRP problem. Then, several variants of PVRPTW problem are proposed. Michallet et al.[7] proposes a PVRPTW variant considering time spread constraints. The time spread constraints mean that the daily arrival time of vehicles to a certain customer must be irregular. Thus the service route plan will be unpredictable. And this has a closely connection with the delivery security in special working scenarios (e.g. for Cash In Transit companies). Periodic Green Vehicle Routing Problem with Time Window (PGVRPTW, Mirmohammadi et al.[8]) is another variant of PVRPTW. In PGVRPTW problem, the vehicle travelling speeds on different roads are connected with the traffic time. And also, the travelling speed affects the CO_2 emission, which is one of the problem objectives. Multi-depot Periodic Vehicle Routing Problem (MDPVRPTW) Mingozzi[9] is another kind of PVRPTW variants. In MDPVRPTW, there are several depots. Vehicles can depart from any of these depots and return to any one after finishing service. Recently, Cantu-Funes et al.[10] proposes a variant of PVRPTW considering vehicle renting. This variant is named as Multi-depot Periodic Vehicle Routing Problem with Time Window and Due Date. In this problem, the original fleet size of vehicles are smaller than needed. Thus in every planning horizon, service manager must make decisions of lease extra vehicles to satisfy customers' requirements. Multi-Periodic Vehicle Routing Problem with Time Window (MPVEPTW, Athanasopoulos and Minis[11]) is another variant of PVRPTW. In MPVRPTW, service requirement has two important attributes: requirement period window and requirement time window. For example, a customer may be serviced on Mondays and Tuesdays (period window of two periods) between 8.00 am to 12.00 am (time window).

PVRPTW-SC is different from the original PVRP-SC problem in that it contains time window constraints. For this reason, the method that decomposes the PVRP-SC problem into two sub-problems can be no longer applied. And for the PVRPTW, the visit frequency of each customer is certain. Even though in some Inventory Routing Problem (e.g. Bard and Nananukul[12], Moin et al.[13], Song and Furman[14]), visit frequency of each customer can be changed during planning horizon, the main aim is to satisfy each cus-

tomer's goods demands. And that makes a difference between PVPRTW-SC and IRP problem.
Nature-inspired algorithms (e.g. Ant Colony Optimization, ACO and Artificial Bee Colony, ABC) have been applied to many combination optimization problems, such as Travelling Salesman Problem (TSP, Tasgetiren et al.[15]), Flow Shop Scheduling (FSS, Li et al.[16] Han et al.[17] Pan et al.[18]), Constrained Optimization Problem (COP, Gong et al.[19] Gao et al.[20] Gong et al.[21]), Job Shop Scheduling (JSS, Li et al.[22] Li et al.[23] and Yu and Yang[24]), and proved to be efficient. In this article, a heuristic algorithm namely MOSA-ACO is applied. This algorithm is mainly a combination of Improved Ant Colony Optimization (IACO) and Multi Objective Simulation Annealing (MOSA). The IACO algorithm is basically a variation of MAX-MIN Ant System (MMAS, Thomas and Holger[25]). IACO has been applied to a PVRP problem and proved to be efficient. In this article, we use IACO to generate initial solutions. Considering that PVRPTW-SC problem requires optimization of three problem objectives simultaneously, MOSA algorithm is also implemented. Detailed description of MOSA-ACO can be seen in section 3.
The last of this article are organized as follows: section 2 gives out a detailed PVRPTW-SC problem model; section 3 describes the algorithm appiled to solve PVRPTW-SC called MOSA-ACO; section 4 gives out the computation experiment results and discussions; section 5 gives out the conclusion and future research work needed.

2. Problem Description

A general PVRPTW-SC model can be described as follows: a) an undirected graph $G = (N, E)$, which includes a set of node N (including customer nodes and depot node) and a set of edges E; b) a planning horizon P gives out the maximum planning horizon; c) a depot node d which includes location and maximum fleet size and; d) a customer set $\{i \in N | i = 1, 2, ..., n\}$, which includes locations, service demand information and preference service time window of each customer; e) a set of edges $\{e_{ij} \in E | i, j = 1, 2, ..., n, i \neq j\}$ which correlates with the travelling cost c_{ij} on arc from node i to node j; f) a set of vehicles $\{v_k \in K | k = 1, 2, ..., K\}$, which includes the capacity information q_k of v_k.
To model the PVRPTW-SC problem, we first give out some general notions:

Journal Pre-proof

N donating set of customer nodes; N={0, 1, 2, ..., n}; node 0 donating the depot node;
A donating the set of network arcs; A={(i, j) : i, j $\in$ N};
K donating the set of vehicles;
C donating vehicle capacity (in this article, we assume that capacities of vehicles in the fleet are equal);
T donating the planning horizon; T={1, 2, ..., t}; t donating the length of period;
c_{ij} donating the travelling cost on arc ij;
t_{ij} donating the travelling time cost on arc ij;
m_i donating the demand of goods of customer i;
q_k donating the vehicle capacity of vehicle k;
e_i donating the earliest time point that customer i is willing to be served;
l_i donating the latest time point that customer i is willing to be served;
Sm_i donating the minimum visit frequency that customer i must receive in total planning horizon;
st_{it} donating the service start time at customer node i in planning period t;
s_i donating time needed to finish service at customer node i;
w_{it} donating the waiting time at customer node i in planning period t;
In our model, the decision making disciplines are as follow:

$$x_{ijkt} = \begin{cases} 0, if\ no\ vehicle\ visit\ arc\ ij\ at\ period\ t \\ 1, otherwise, i \neq j, i, j \in \{0, 1, 2, ..., n\} \end{cases}$$

w_{it} donating the waiting time at node i in planning period t;
The optimization objectives are:

$$Minimize \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{i=0}^{N} \sum_{j=0}^{N} c_{ij} x_{ijkt} \tag{1}$$

$$Maximize \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{i=0}^{N} \sum_{j=1}^{N} x_{ijkt} \tag{2}$$

$$Minimize \max(\sum_{j=1}^{N} \sum_{k=1}^{K} x_{ijkt}, k \in K, i = 0) \tag{3}$$

Journal Pre-proof

Subject to:

$$\sum_{j=1}^{N}\sum_{k=1}^{K} x_{ijkt} \leq K, fort \in T, i = 0 \tag{4}$$

$$\sum_{j=0,j\neq i}^{N}\sum_{k=1}^{K} x_{ijkt} \leq 1, fori \in \{1, 2, ...n\}, \forall t \in T \tag{5}$$

$$\sum_{i=0,i\neq j}^{N}\sum_{k=1}^{K} x_{ijkt}\text{-}\sum_{r=0}^{N}\sum_{k=1}^{K} x_{jrkt} = 0, forj \in \{1, 2, ...n\}, \forall t \in T \tag{6}$$

$$\sum_{j=1}^{N} x_{ijkt} \leq 1, fori = 0, t \in T, \forall k \in K \tag{7}$$

$$\sum_{j=1}^{N} x_{ijkt}\text{-}\sum_{j=1}^{N} x_{jikt}{=}0, fori = 0, t \in T, \forall k \in K \tag{8}$$

$$\sum_{i=1}^{N}\sum_{j=0}^{N} x_{ijkt} m_i \leq q_k, for \forall t \in T, k \in K \tag{9}$$

$$\sum_{t=1}^{T}\sum_{i=0}^{N}\sum_{k=1}^{K} x_{ijkt} \geq Sm_j forj \in \{1, 2, ..., n\} \tag{10}$$

$$st_{it} \geq e_i, \forall t \in T \tag{11}$$

$$st_{it}{+}s_i \leq l_i, \forall t \in T \tag{12}$$

$$st_{it} + s_i + w_{jt} + t_{ij} = st_{jt}, for \forall t \in T \tag{13}$$

In PVRPTW-SC problem model, the objectives are: travelling distance minimization (equation (1)); visit frequency maximization (equation (2)) and fleet size minimization (equation (3)). The problem constraints include: constraint (4) ensures that the fleet size in each planning period must not exceed the maximum fleet size; constraint (5) ensures any customer can be visited at most once in one planning period; constraint (6) ensures that a route must not finish at a customer node; constraints (7) and (8) together ensure that any vehicle leaves the depot node must come back when it finish its service

Journal Pre-proof

plan in a planning period; constraint (9) ensures the that the mission payload allocated to a single vehicle must not exceed the maximum capacity of this vehicle; constraint (10) ensures that the minimum visit frequency of each customer must be satisfied; constraint (11) and (12) together ensure that service must start and finish in customer preferred time window; constraint (13) ensures the continuity of service time, waiting time and travelling time in a feasible solution.

Our problem model is different from original PVRP-SC problem Francis et al.[4] in following respects: a) PVRPTW-SC problem takes hard time window constraint into considerations; b) In PVRP-SC problem, there is an implicit constraint that a reasonable set of routes for each driver to perform must be maintained. In PVRPTW-SC problem, we do not maintain this implicit constraint since it is not necessary; c) In PVRPTW-SC problem, three problem objectives with conflictions are considered: travelling distance minimization, fleet size minimization and visit frequency maximization. In original PVRP-SC problem, only travelling distance is considered.

The relations between the three problem objectives are complicated. First, it is obvious that adding extra service frequency is coincident with the growth of travelling distance. Adding extra service frequency can also bring increase to fleet size under the condition that the vehicle capacity constraint is a tight constraint. The relations between travelling distance and fleet size are even more complicated. According to [35], relation between reducing travelling distance and reducing fleet size is not always negative. These two objectives sometimes are positively correlated when the geographical distribution of customers is clustered. Otherwise these two objectives are more likely to be negatively correlated.

3. Multi-Objective Simulation Annealing Ant Colony Optimization

Considering the complexity of PVRPTW-SC problem, we first divide this problem into two sub-problems. In sub-problem 1, we decide service frequency of each customer and assign customer service requirement to different day. This problem is like a classic PVRPTW problem with least visit frequency constraint instead of visit combination constraint. Then, in sub-problem 2, we address the daily routing plan optimization problem. The daily routing plan optimization problem is a typical VRPTW problem. The

Journal Pre-proof

reason is that in PVRPTW-SC problem, a customer only has lowest visit frequency requirement. So, in this article, we first assign the lowest visit frequency of each customer to each day in planning horizon. With the help of a K-mean based heuristic, we ensure that locations of customers assigned to each day are clustered. Then, when an extra visit frequency is added to service plan, it has a higher possibility to be added to the day in which the average travelling distances from this customer to customers that are already assigned to this day are short. When all customer visit frequencies (including lowest visit frequency and extra visit frequency) are assigned to a certain day, this problem becomes a VRPTW problem for each day in the service plan. It should be admitted that dividing PVRPTW-SC problem into these two sub-problems may lose some potential solution space. But based on the experiment results we think that this weakness is acceptable. The algorithm applied in this article is called Multi Objective Simulation Annealing 8C Ant Colony Optimization (MOSA-ACO). The algorithm structure can be seen in Fig.2:

The reasons that we design MOSA-ACO are as follows: a) IACO has been

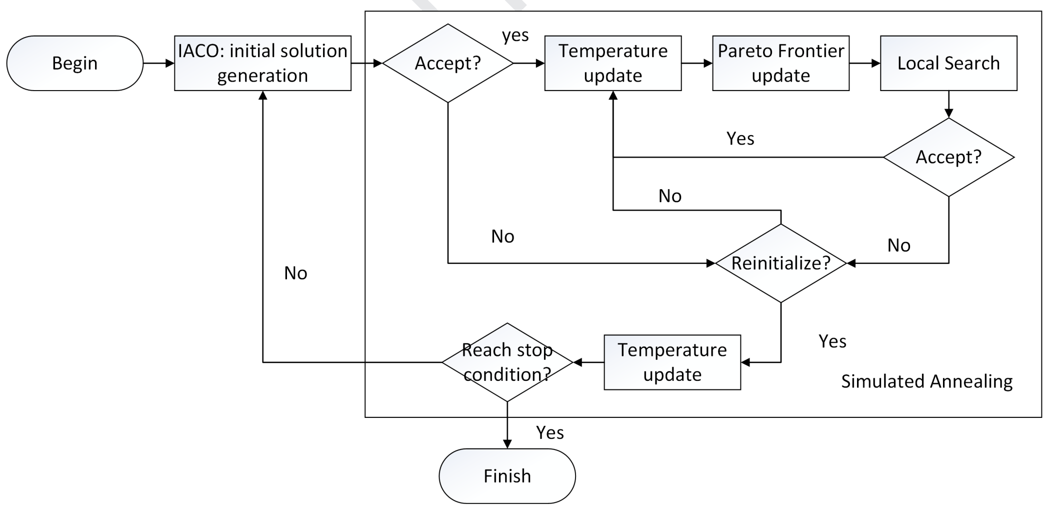


Figure 2: MOSA-ACO algorithm structure

proved to be efficient at solving PVRPTW problem with an acceptable computation cost. Thus, in MOSA-ACO, we use IACO to generate the initial service plan. One thing should be admitted is that the original IACO is a single objective algorithm. To make IACO suitable for the aim of finding service plan with more customers, we build our IACO algorithm with an

Journal Pre-proof

adjustable control parameter. We describe this modification in section 3.1. With this modification, IACO can find solutions with variable customer service frequency with acceptable service cost while the lowest service frequency constraint is satisfied. And, the crossover operator in IACO has the ability of local optimum avoidance in early searching phase. b) In MOSA procedure, we mainly focus on reducing service cost. SA framework has an advantage of local optimum avoidance since it can accept solutions with bad qualities. In MOSA, we design a new multi objective criterion to decide whether the current solution can be accepted. Then, several local search heuristics are applied to the initial service plan. These heuristics have been proved to be efficient at solving VRPTW problem. c) In MOSA-ACO, we also design a reinitialization mechanism to enhance algorithm searching ability. In MOSA procedure, if local search heuristics have run several iterations and no new feasible solution is accepted, IACO with a new control parameter is applied to generate new initial solutions. This will also help MOSA-ACO jump out of local optimum and find new feasible solutions with higher diversity. The MOSA-ACO follows steps below to construct solutions:
Step 1: initial solution is constructed using IACO based heuristics;
Step 2: the initial solution is evaluated using a Euclidean distance based criterion, if the initial solution is accepted, then the initial solution is set to be current solution and go to step 3; else go to step 7;
Step 3: current temperature is updated;
Step 4: Pareto Frontier is updated;
Step 5: four local search algorithms are applied to current solution;
Step 6: current solution is evaluated using a Euclidean distance based criterion, if the current solution is accepted, go to step 3; else go to step 7;
Step 7: current temperature is updated, if the stop condition is reached, then the algorithm is finished; else if the reinitialization condition is reached, go to step 1; else the local search strategies are applied.
In the section below, we give out a detailed description of MOSA-IACO algorithm.

3.1. Improved Ant Colony Optimization

The MOSA-ACO algorithm uses an IACO algorithm to construct initial solutions. The IACO algorithm, which has a good performance in PVRPTW in Yu and Zhang[24], is a combination of Ant Colony Optimization and Genetic Algorithm. The IACO algorithm structure can be seen in Fig.3:

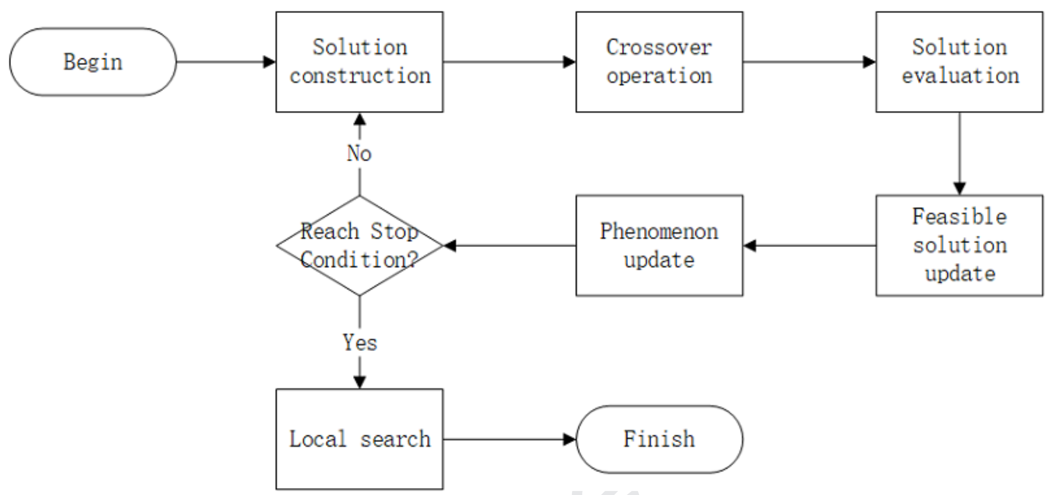


Figure 3: IACO structure

The IACO algorithm of MOSA-ACO follows steps below to construct solutions:

Customer distribution. In PVRPTW-SC problem, every customer has a lowest visit frequency that must be satisfied. Thus, in the beginning of an initial solution construction, we divide these requirements into each planning periods of the planning horizon using a K-means based method. The K-Means method is a very commonly used cluster algorithm. This method considers the similarity of each node in a graph and divides these nodes into k groups. In this article, we divide customers into k groups using a distance-based rule. The number of k is set to be equal to the number of planning periods in a planning horizon. With the help of this method, we ensure that customers that are divided into the same planning period are spatially clustered. Customer nodes are first set to the nearest core. If the lowest visit frequency of a customer is more than one, then it will be set to the first nearest and second nearest core and so on. When the customer distribution phase is finished, the lowest visit frequency constraint of each customer must be satisfied. Fig.4 shows a result of customer distribution.

After customer distribution phase, every planning period has a list of customers that must be visited in this period. This list of customers is called cs_t.

Journal Pre-proof

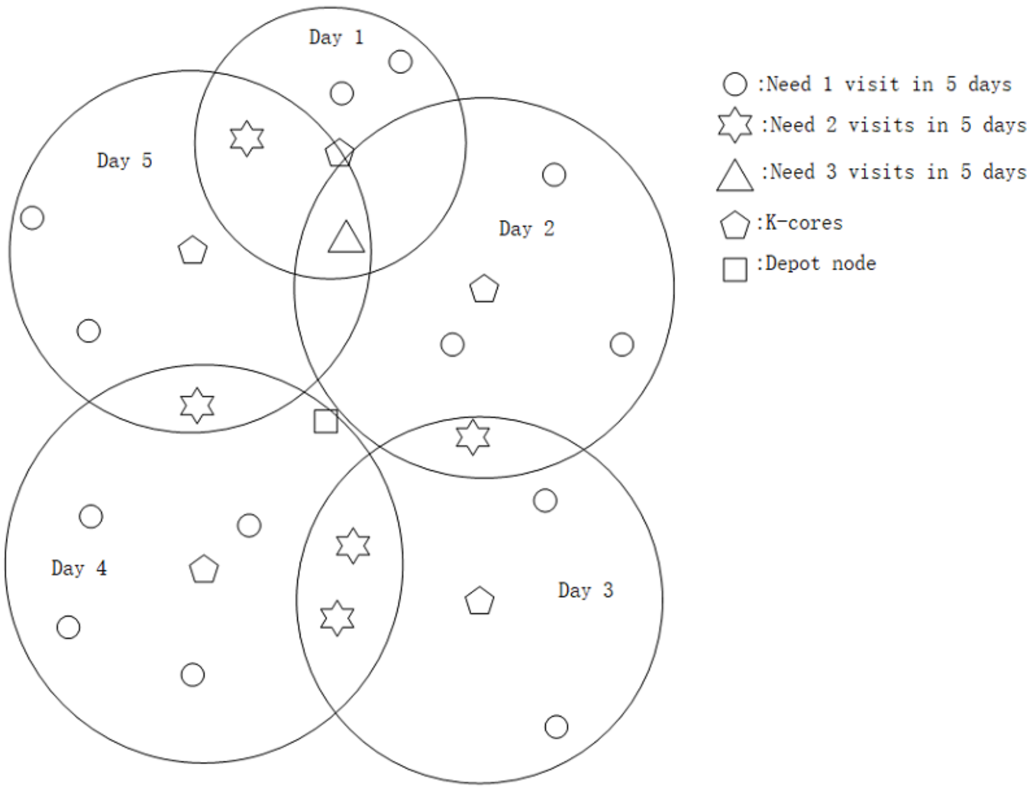


Figure 4: Customer distribution result

Solution construction. At the beginning of each iteration, an artificial ant is put on the depot node. Then, it visits customer nodes following a probabilistic rule donated in equation (14). Customers that are visited by the artificial ant are put into the tabu list. When there is no customer that can be visited, the artificial ant goes back to the depot and a new artificial ant is put on the depot node and begin to build its own route. After all customers in cs_t are visited, the daily plan is built. For each day in planning horizon, IACO follows equation (14) to build daily plan.

$$P_{ij}^{t} = \begin{cases} \frac{\lambda\left(\tau_{ij}^{t}\right)^{\alpha}(\eta_{ij})^{\beta}}{\sum\limits_{s=1}^{N}\lambda\left(\tau_{is}^{t}\right)^{\alpha}(\eta_{is})^{\beta}}, if node j can be selected; \\ 0, otherwise. \end{cases} \tag{14}$$

Journal Pre-proof

In equation (14), P_{ij}^t donates the probability that artificial ant chooses node j to visit next when it is at node i. τ_{ij}^t donates the pheromone information on arc ij at planning period t. η_{ij} donates the heuristic information, which is set to $1/c_{ij}$ in this article. α and β are control parameters which indicate the importance of heuristic and pheromone information. λ is a parameter which controls the willingness of artificial ants to choose a customer node that is not in cs_t to visit. When a customer is in cs_t, λ is set to 1; otherwise λ is set to a constant that is less than 1. The λ parameter is a new mechanism that helps IACO generate solutions with different visit frequency. Every time when MOSA-ACO is reinitialized, λ also changes. The value of λ is between $(0, 1)$. When λ becomes larger, artificial ants are more willing to choose customers that are not in cs_t to visit. The route construction phase is shown in fig.5:

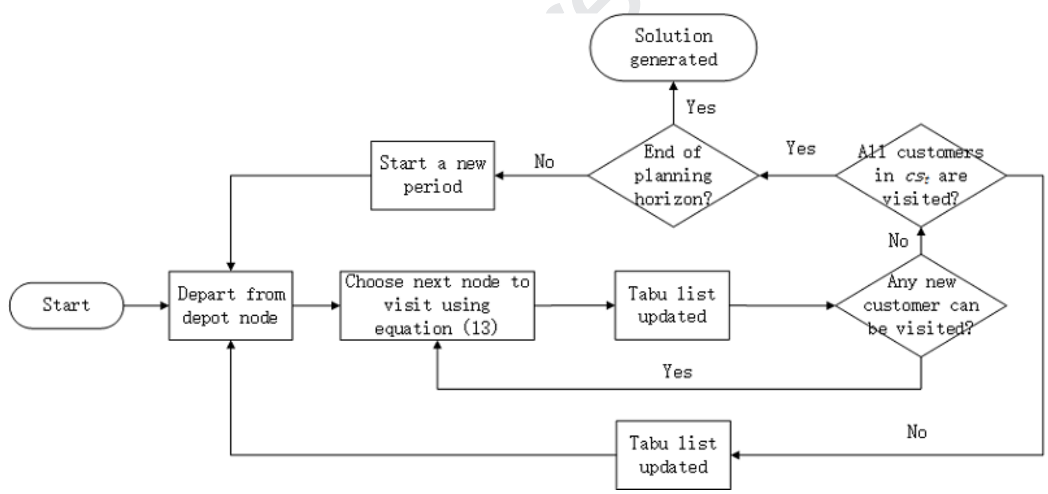


Figure 5: Route Construction

Crossover. IACO algorithm of MOSA-ACO applies a crossover operator to help find better solutions. The crossover operator breaks and relinks routes in a same planning period to help improve solution quality and prevent local optimum. The two crossover operations (i.e. one-point crossover operation and two-point crossover operation) applied in IACO of MOSA-ACO are the same as article Yu and Zhang[24]. Since the aim of crossover operator is to improve the solution quality, only solutions found in crossover operator that are feasible can update the pheromone information.

Journal Pre-proof

Pheromone update. IACO of MOSA-ACO follows a self-adaptive manner to update pheromone information. In every iteration, when crossover phase is over and all new feasible solutions are generated, the algorithm updates the pheromone information following equation (15):

$$\tau_{ij}^{t\ new} = (1-\rho)\tau_{ij}^{t\ old} + \Delta\tau_{ij}^{t\ old}\rho \in [0,1] \tag{15}$$

$\tau_{ij}^{t\ new}$ donates the pheromone information on arc ij in next iteration. $\tau_{ij}^{t\ old}$ donates the current pheromone information on arc ij. ρ donates the pheromone evaporation rate. $\Delta\tau_{ij}^{t\ old}$ donates the pheromone that leaves on arc ij if ij is in current solution. $\Delta\tau_{ij}^{t\ old}$ is decided using equation (16):

$$\Delta\tau_{ij}^{t\ old} = \left(\delta \times \frac{L^{Opt}}{L^{current}}\right) \tag{16}$$

In equation (16), δ is a constant which donates the pheromone increment baseline. L^{Opt} is the total length of current best solution. The initial best solution is generated using nearest neighbor search algorithm (all customers must be visited in each planning period). $L^{current}$ is the total length of current solution. It must be noticed that in original IACO, a punishment factor is applied to reduce the pheromone that leaves on routes in infeasible solutions. However, in IACO of MOSA-ACO, because the infeasible solutions are simply discarded, this factor is removed. δ is decided using equation (17):

$$\delta = \tau_{\max} \times \rho \tag{17}$$

The $\tau_{\max}$ gives the upper bound of pheromone information leaves on an arc. And also, a parameter $\tau_{\min}$ is given to donate the lower bound of pheromone information. The $\tau_{\max}$ is set to the maximum iteration times. And $\tau_{\min}$ is set to 1. The initial pheromone information of each arc is set to $\tau_{\max}$.
Another thing that must be noticed is in IACO of MOSA-ACO, we use a multi-dimensional pheromone matrix to save pheromone information in each planning period. The reason is that the customers who need to be served in different planning periods are different.
Local search. To improve the solution quality of IACO, 2-opt local search heuristic is applied to the solutions generated by IACO. 2-opt is a widely applied local search strategy in VRPTW. Detailed description of 2-opt can be seen in section 3.3. It must be noticed that in MOSA procedure several local search heuristics are applied. However, only 2-opt heuristic is implemented

Journal Pre-proof

in IACO as a result of reduction of computation cost and local optimum avoidance.

3.2. Multi Objective Simulation Annealing

When a new feasible solution a is found, MOSA-ACO follows these steps to decide whether this solution is accepted:
a) if solution a is a non-dominated solution, then it will be accepted directly by MOSA-ACO.
b) if solution a is dominated by current Pareto Frontier solution set, then a probabilistic based rule will be implemented to decide whether solution a can be accepted. This rule can be seen in equation (18) and equation (19).
c) if solution a is accepted, then the current Pareto Frontier set will be checked and solutions that are dominated by solution a will be deleted.
When solution a is dominated by current Pareto Frontier solution set, MOSA first use equation (18) to evaluate the quality of solution a.

$$F(a) = \min_{s \in NS} \sqrt{(Obj_a^1 - Obj_s^1)^2 + \ldots\ldots + (Obj_a^n - Obj_s^n)^2} \tag{18}$$

In equation (18), NS donates the non-dominated solution set. $F(a)$ donates the fitness value of solution a. Obj_a^n donates the n objective value of solution a. Obj_s^n donates the n objective value of solution s. Basically, equation (18) gives out the shortest Euclidean distance of the current solution to the current non-dominated solution set found by MOSA-ACO. The reason that multi objective criterion is introduced into MOSA-ACO is to measure the quality difference between a new feasible solution and global best solutions. In original SA framework, a feasible solution has a high possibility to be accepted when the difference between its fitness value and global best is small. Since PVRPTW-SC problem has three problem objectives, to calculate the difference of solution quality between a new feasible solution and global best solutions, this criterion is applied. Fig.6 gives a visualized description of $F(a)$:

Then, MOSA uses equation (19) to decide whether solution a will be accepted:

$$PA(a) = e^{-F(a)/T_c} \tag{19}$$

Journal Pre-proof

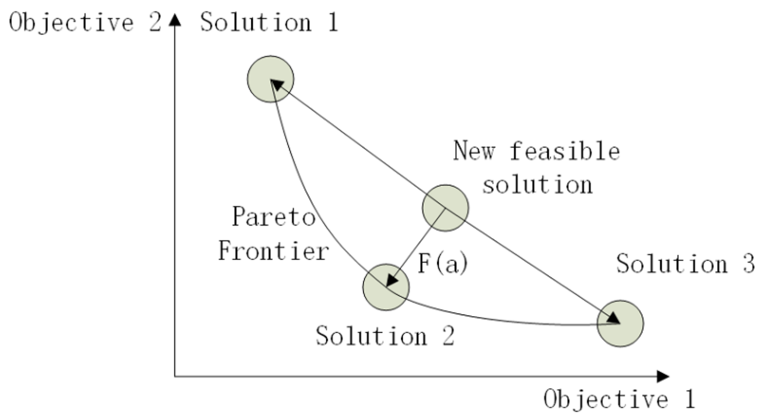


Figure 6: multi objective evaluation criterion (when newly found solution a is dominated by current Pareto Frontier solution set)

In equation (19), T_c donates the current temperature, $PA(a)$ donates the probability that solution a will be accepted. Every time a new solution is constructed, the current temperature is adjusted following equation (20):

$$T_n = T_c \times \varepsilon \tag{20}$$

In equation (20), T_n donates the temperature in next iteration, ε is a parameter which controls the decrease speed of T_c.
To avoid fast convergence, a backfire technology is also implemented. Every time a new solution is accepted, the current temperature will be adjusted using equation (21):

$$T_n = (T_p - T_c) \times \omega + T_c \tag{21}$$

In equation (21), T_p donates the temperature last time when a feasible solution generated by IACO is accepted, ω is a control parameter. In this article, ω is set to 0.5.

3.3. Local Search Heuristics

To improve the solution quality of MOSA, six kinds of local search heuristics are applied. These local search heuristics include: the well-known 2-opt (Croes [26]) and 3-opt (Lin [27]) heuristics, route elimination and new route operators (Garcia-Najera and Bullinaria [28]) and node exchange and node

Journal Pre-proof

insertion heuristics. The node exchange heuristic randomly selects 2 routes in 1 planning period. Then, node exchange heuristic tries to break these 2 routes into 4 parts and then relinks these 4 parts into 2 new routes. Every possible breaking position of these 2 routes is tested and the new solution with lowest travelling distance is accepted. The node insert heuristic first randomly chooses 2 routes in the same planning period. Then, node insert heuristic randomly removes a customer node in route 1 and insert it into every possible position in route 2. The solution with lowest travelling distance is accepted. The node exchange heuristic is shown in fig.7 and node insertion is shown in fig.8.

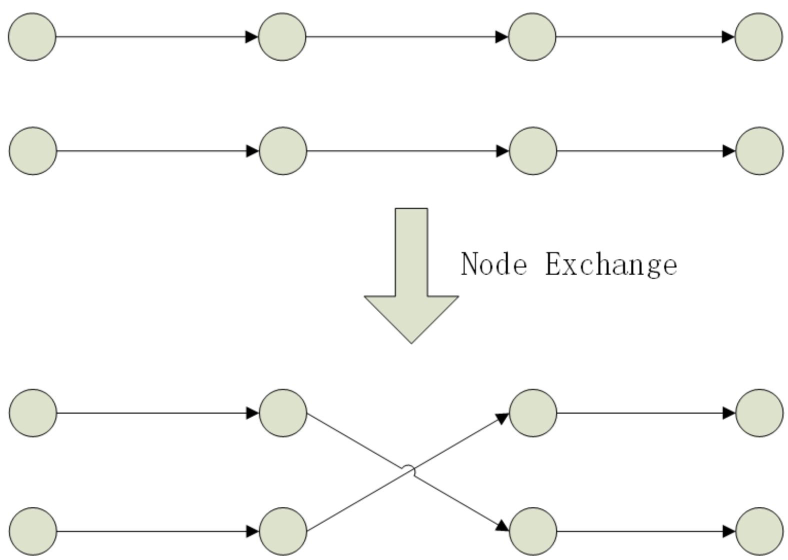


Figure 7: Node Exchange Heuristic

It should be notice that in MOSA-ACO, there are two phases that use local heuristics. To reduce time cost, in IACO phase we only apply 2-opt heuristic, which is the same as the original IACO algorithm.

Journal Pre-proof

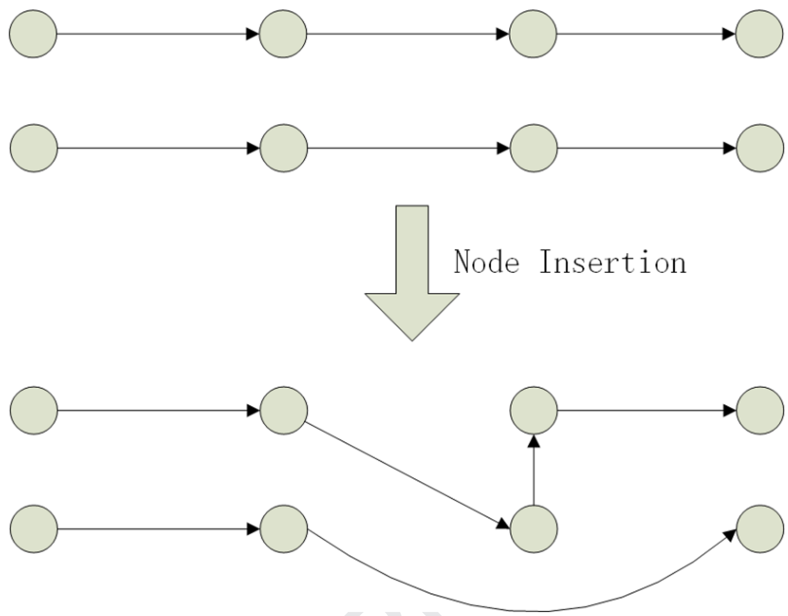


Figure 8: Node Insertion Heuristic

4. Numerical analysis

In this section, the numerical analysis design and results are reported.

4.1. Experiment design

In this paper, we use benchmark instances generated from two widely applied benchmark instance problem sets. One of them is generated from Solomon's 50 benchmark problem instances. The other is generated from Cordeau's PVRPTW problem instances. These instances can be found at http://neo.lcc.uma.es/vrp/vehicle-routing-problem/.
Solomon's benchmark instances have 3 problem scales: 25 customers, 50 customers and 100 customers. In this paper we use the 50 customer instances. According to Baños et al [29], a selected Solomon's benchmark instance set can be used to test the performance of an algorithm since the Solomon's benchmark instances are designed following some certain patterns. In Solomon's benchmark instances, there are six categories: C1 and C2 categories, R1 and R2 categories and RC1 and RC2 categories. In each category, the distribution pattern of locations of customers is the same. In C1 and C2

Journal Pre-proof

categories, locations of customers are clustered. In R1 and R2 categories, locations of customers are randomly generated. And instances in RC1 and RC2 categories have mixtures of random and clustered customers. Instances in C1, R1 and RC1 have smaller time windows than instances in C2, R2 and RC2. In each category, the customer geographical locations, demands and service times are the same, while they differ in the percentage of customers that have time windows and the time window intervals.
Since the Solomon's benchmark instances are designed as VRPTW instances, we generate our instances based on Solomon's benchmark instances: the planning horizon is set to 5; the customer locations, demands and time windows of each selected instances remain unchanged; for each customer, a minimum visit frequency is given based on a probabilistic rule. The range of minimum service is set to 1 to 3. For each customer, the probability that its minimum visit frequency is set to 1 is 85%, while the probabilities of 2 and 3 are 15% and 5%, respectively.
The PVRPTW-SC problem is also very similar to the PVRPTW problem, so Cordeau's C-PVRPTW problem instances are also applied to test the performance of MOSA-ACO algorithm. In this paper, we only use instances in Cordeau's problem instances which have total customers less than 150. And in Cordeau's problem instances, all customers have possible visited combinations. Since in PVRPTW-SC, we only focus on the visit frequency of each customer, the possible visited combinations are no longer needed.
An example of PVRPTW-SC problem instance can be seen in Table 1.

Table 1: An example of PVRPTW-SC problem instance

Time Period	5 days	Vehicle capacity	200	Max. fleet size	25		
Customers:							
Cust. No.	X Coord.	Y Coord.	Demand	Ready time	Due date	Service time	Min. frequency
1	45	45	10	700	1200	90	2
2	30	45	5	200	800	90	3
......							

In this paper, comparison algorithms include four widely applied population based multi objective optimization algorithms, namely NSGA II [30], SPEA2 [31], MOEA/D [32] and NSGA III [33]. MOEA based algorithms are very commonly used meta-heuristics for many kinds of multiobjective optimization problems [34]. The algorithm frameworks are open source codes which can be gotten from https://yarpiz.com/category/multiobjective

-optimization. To analyze the improvement of local search heuristics and MOSA framework, computation experiments using original IACO algorithm are included as well. For each algorithm, the customer minimum visit frequency is first randomly distributed to the planning horizon. The daily plan is initialized using a time-window insertion heuristic (TWIH) which is described in Baños et al [29] and proved to be efficient for VRPTW problems. The crossover operator, which is described in Garcia-Najera and Bullinaria [28], is a widely applied crossover operator in VRP problems. The mutation operators include ten efficient operators, namely random relocation, customer best relocation, customer random migration, customer best migration, customer random exchange, customer best exchange, customer exchange with coincident time window, route partitioning, new route and route elimination, for VRP problems to reduce travelling distances and fleet size. These operators are selected from Tan et al.[35], Alvarenga et al.[36], El-Sherbeny.[37] and Garcia-Najera and Bullinaria [28]. To achieve adding extra visit frequency to customers, two other operators are designed. One of them adding extra visit frequency to customers that have lowest visit frequency. The newly added service frequencies are put into the daily plan that has lowest number of customers. In this operator, up to 5 randomly chosen customers can be chosen in one time. Another operator is designed to randomly remove visit frequencies from daily plan that has largest number of customers while the lowest visit frequency constraint must not be broken. This operator removes one visit of a customer in one time. All the 12 mutation operators have the same probability to be chosen in one iteration. Other parameter tuning methodologies are as follows.

For MOSA-ACO, there are 2 parts that should be tuned:

a) in the MOSA part, three parameters (initial temperature, target temperature and delta) need to be tuned. The initial temperature is set using a simple but efficient principle which is suggested in [38]. The initial temperature is set such that a solution that is w% worse than the initial solution can be accepted at a possibility of more than 50%. Thus, in this article, we set the initial temperature at 800. We also follow a simple principle that the algorithm can be terminated when a new solution can be accepted at a possibility of lower than 1%, so we set the target temperature at 10. Then, the ε is set to 0.99, which is a commonly used ε of simulated annealing algorithm.

b) for IACO in MOSA-ACO, the control parameters (α and β) are suggested by the original article [24]. The maximum iterations from 100 to 500 are tested and 200 is chosen considering both algorithm performance and com-

putation costs.
For population-based algorithms (NSGA II, SPEA2, NSGA III, MOEA/D), the number of solutions generated by MOSA-ACO during 1 run is measured. Then, the population size of 50 and total running time of 300 are chosen to make sure that the solutions generated by MOSA-ACO and comparison algorithms are nearly the same.
For IACO, the maximum running iterations are set to 1000 for instances that contain 50 customers, 2000 for instances that contain customers that are more than 50 but less than 100 and 3000 for instances that contain customers that are more than 100. This will result in nearly the same computation costs of IACO and MOSA-ACO.
All tests are running on a computer with Intel® $Core^{TM}$ i5–4460 central processing unit (four cores, 3.2 GHz) with 4 GBs of memory. All of the algorithms are coded using MATLAB 2014a. The $\{\alpha, \beta\}$ is set to $\{2, 1\}$, which is the same as it is in article Yu and Zhang[24]. ρ is set to 0.01. Maximum running iteration of IACO in MOSA-ACO is set to 200.

4.2. Experiment result

We choose 20 benchmark problems to test the performance of MOSA-ACO. λ is set to 1 for customers in cs_t. For customers that are not in cs_t, factor λ is set following the steps below: a) λ is set to 0.5 when MOSA-ACO is initialized; b) every time when a new solution generated by IACO is accepted by MOSA, λ is reduced by 0.1; c) while $\lambda < 0$, it is reset to 0.5.
In this section, the average performance of three algorithms are reported. Table 3, 4 and 5 give the average value of three objectives of non-dominate solutions found by the three algorithms in each instance. In tables below, "T" is short for "Travelling Distance", "FS" is short for "Fleet Size", "VF" is short for "Visit Frequency". To help understanding the PVRPTW-SC problem results, an visualized example is shown in Fig.9.

In VRPTW instances, MOSA-ACO algorithm shows a good ability of finding solutions with higher total visit frequency and lower travelling distance. In 10 of 12 test instances, MOSA-ACO finds solutions with lowest average travelling distance. And in 6 of 12 instances, MOSA-ACO finds solutions with largest total visit frequency. In C type and RC type instances, MOSA-ACO finds solutions with lowest travelling distance and highest total

Table 2: Mean numbers of travelling distance, fleet size and visit frequency in Solomon's instances

		MOSA-ACO			IACO[24]			NSGA II[30]			SPEA 2[31]			MOEA/D[32]			NSGA III[33]		
		Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.
C103-50																			
	TD	**913.92**	1.6%	0.0%	1094.71	0.7%	19.8%	1150.33	6.1%	25.9%	1078.14	10.5%	18.0%	1038.10	6.9%	13.6%	963.15	7.2%	5.4%
	FS	3.55	2.3%	15.5%	4.38	5.8%	31.5%	3.29	5.6%	8.8%	3.01	13.8%	0.3%	3.21	5.0%	6.4%	**3.00**	14.4%	0.0%
	VF	97.77	2.7%	5.7%	102.58	2.9%	1.1%	**103.73**	3.8%	0.0%	93.02	7.8%	10.3%	98.06	4.5%	5.5%	99.30	6.9%	4.3%
	CPU(s)	1063.29			1145.69			88.52			81.19			191.57			198.58		
C108-50																			
	TD	1216.65	1.6%	11.4%	1222.49	1.2%	12.0%	1335.57	2.7%	22.3%	**1091.9**	5.9%	0.0%	1331.75	4.9%	22.0%	1280.36	2.4%	17.3%
	FS	4.28	1.7%	33.5%	5.11	2.5%	44.3%	2.95	8.3%	3.5%	3.45	7.7%	17.4%	3.31	5.5%	14.0%	**2.85**	0.3%	0.0%
	VF	**127.43**	2.9%	0.0%	124.75	2.2%	2.1%	105.2	5.5%	17.4%	99.03	5.2%	22.3%	101.58	3.6%	20.3%	108.32	2.9%	15.0%
	CPU(s)	2585.71			1587.14			93.8			80.18			191.33			202.99		
C203-50																			
	TD	**1086.5**	0.9%	0.0%	1109.55	0.7%	2.1%	1223.64	4.5%	12.6%	1225.23	6.4%	12.8%	1207.61	7.0%	11.2%	1169.10	8.5%	7.6%
	FS	2.73	4.2%	45.9%	3.00	0.0%	50.8%	1.77	9.8%	16.6%	1.52	18.7%	2.9%	**1.48**	13.8%	0.0%	2.20	14.4%	32.9%
	VF	**115.21**	2.1%	0.0%	111.88	2.1%	2.9%	99.98	3.3%	13.2%	98.87	5.7%	14.2%	103.92	3.1%	9.8%	107.48	3.4%	6.7%
	CPU(s)	1244.32			1635.39			81.05			59.31			185.72			173.75		
C208-50																			
	TD	**1113.8**	0.8%	0.0%	1230.13	0.4%	10.4%	1201.4	3.7%	7.9%	1134.5	3.2%	1.9%	1207.61	7.0%	8.4%	1184.93	8.4%	6.4%
	FS	2	0.0%	43.5%	2.00	0.0%	43.5%	1.53	6.9%	26.1%	**1.13**	13.6%	0.0%	1.38	13.8%	17.9%	1.34	6.3%	15.8%
	VF	**113.78**	2.7%	1.0%	114.92	2.7%	0.0%	98.03	3.2%	14.7%	99.37	4.9%	13.5%	103.92	3.1%	9.6%	102.12	3.7%	11.1%
	CPU(s)	3072.09			2296.34			70.82			49.8			185.17			170.99		
R103-50																			
	TD	**1791.4**	1.5%	0.0%	2067.54	1.0%	15.4%	1859.74	3.9%	3.8%	1800.47	6.3%	0.5%	1833.07	6.1%	2.3%	1816.77	7.3%	1.4%
	FS	5.2	2.3%	10.6%	5.35	3.7%	13.2%	5.26	9.4%	11.6%	5.35	11.3%	13.1%	**4.65**	9.1%	0.0%	4.88	16.9%	4.7%
	VF	107.93	2.5%	6.1%	**114.95**	1.1%	0.0%	105.67	3.7%	8.1%	103.89	5.4%	9.6%	103.99	4.0%	9.5%	103.39	9.1%	10.1%
	CPU(s)	2355.44			2290.14			99.95			82.1			183.66			193.54		
R108-50																			
	TD	**1384.7**	0.8%	0.0%	1512.31	0.6%	9.2%	1519.16	4.1%	9.7%	1425.32	2.6%	2.9%	1469.62	8.1%	6.1%	1473.38	8.2%	6.4%
	FS	3.33	4.5%	9.9%	3.64	3.6%	17.5%	3.54	3.4%	15.3%	**3**	5.4%	0.0%	3.28	4.9%	8.6%	3.30	11.3%	9.0%
	VF	98.9	1.7%	3.7%	**102.68**	2.0%	0.0%	100.37	2.3%	2.2%	95.16	5.1%	7.3%	94.62	6.3%	7.8%	95.26	7.5%	7.2%
	CPU(s)	2812.24			2250.75			99.59			99.52			195.45			202.62		
R203-50																			
	TD	1697.94	1.7%	4.3%	1946.29	2.6%	19.5%	1706.23	2.6%	4.8%	1676.23	4.5%	2.9%	**1628.61**	6.0%	0.0%	1636.54	5.9%	0.5%
	FS	2.96	3.6%	49.0%	3.05	3.4%	50.5%	2.07	4.4%	27.1%	1.6	5.9%	5.6%	1.83	13.1%	17.7%	**1.51**	1.7%	0.0%
	VF	**126.31**	2.5%	0.0%	126.02	2.4%	0.2%	110.51	3.3%	12.5%	111.64	2.9%	11.6%	108.36	8.2%	14.2%	113.94	4.9%	9.8%
	CPU(s)	769.78			1581.57			183.44			99.52			175.49			199.44		
R208-50																			
	TD	**1084.1**	0.9%	0.0%	1108.53	4.7%	2.3%	1106.23	4.5%	2.0%	1107.52	3.4%	2.2%	1143.26	11.9%	5.5%	1090.23	7.1%	0.6%
	FS	1.29	8.2%	22.1%	1.18	17.8%	14.8%	1.13	14.7%	11.1%	1.13	14.5%	11.1%	1.05	8.5%	4.3%	**1.00**	0.9%	0.0%
	VF	91.4	2.3%	3.9%	**95.09**	5.7%	0.0%	91.2	2.3%	4.1%	90.66	3.3%	4.7%	88.93	5.8%	6.5%	89.71	6.0%	5.7%
	CPU(s)	1264.67			1443.28			122.98			224.01			183.35			168.62		
RC103-50																			
	TD	**1705**	0.8%	0.0%	1767.38	1.1%	3.7%	1716.29	2.9%	0.7%	1793.1	6.0%	5.2%	1791.78	4.3%	5.1%	1756.95	4.5%	3.0%
	FS	4.5	2.9%	6.0%	5.00	0.0%	15.4%	4.51	5.5%	6.2%	**4.23**	9.1%	0.0%	4.53	2.6%	6.6%	4.43	9.9%	4.5%
	VF	**96.65**	2.0%	0.0%	96.59	1.8%	0.1%	95.85	4.9%	0.8%	94.72	5.7%	2.0%	92.01	3.7%	4.8%	92.60	5.9%	4.2%
	CPU(s)	857.56			1108.96			152.41			97.09			170.38			190.08		
RC108-50																			
	TD	**1439.9**	0.8%	0.0%	1596.24	0.7%	10.9%	1507.51	3.7%	4.7%	1531.46	3.6%	6.4%	1587.47	3.6%	10.2%	1478.36	5.3%	2.7%
	FS	3.99	3.1%	21.6%	4.57	3.0%	31.5%	3.81	5.9%	17.8%	**3.13**	3.5%	0.0%	3.59	8.5%	12.7%	3.33	12.6%	6.0%
	VF	93.77	2.1%	0.7%	**94.39**	4.0%	0.0%	93.58	6.9%	0.9%	90.36	3.6%	4.3%	90.87	3.7%	3.7%	89.01	10.6%	5.7%
	CPU(s)	965.5			1205.23			170.16			108.15			182.53			188.81		
RC203-50																			
	TD	**1739.7**	3.6%	0.0%	1942.25	1.4%	11.6%	1796.1	5.2%	3.2%	1766.47	7.1%	1.5%	1782.17	5.0%	2.4%	1752.38	8.6%	0.7%
	FS	3	6.8%	37.0%	3.51	6.2%	46.1%	2.02	2.3%	6.4%	**1.89**	11.1%	0.0%	2.31	6.5%	18.1%	2.04	3.0%	7.3%
	VF	125.06	6.3%	0.4%	**125.58**	1.9%	0.0%	107.86	4.8%	14.1%	108.18	6.9%	13.9%	111.41	5.9%	11.3%	112.90	11.6%	10.1%
	CPU(s)	818.89			1429.20			207.06			92.08			191.48			216.05		
RC208-50																			
	TD	**1365**	14.3%	0.0%	1640.13	1.8%	20.2%	1468.03	2.9%	7.5%	1449.87	9.5%	6.2%	1484.93	12.0%	8.8%	1365.60	4.9%	0.0%
	FS	1.84	2.0%	19.2%	2.00	0.0%	25.6%	1.54	11.8%	3.4%	1.63	15.0%	8.7%	1.80	7.4%	17.3%	**1.49**	3.2%	0.0%
	VF	**107.48**	11.9%	0.0%	106.18	4.4%	1.2%	101.21	5.7%	5.8%	99.33	8.5%	7.6%	102.35	5.5%	4.8%	102.07	6.1%	5.0%
	CPU(s)	994.67			1365.51			133.53			350.15			180.33			173.70		

Table 3: Mean numbers of travelling distance, fleet size and visit frequency in Solomon's instances(averaged by type of instances

		MOSA-ACO		IACO[24]		NSGA II[30]		SPEA 2[31]		MOEA/D[32]		NSGA III[33]	
		Total	Dif.	Total	Dif.	Total	Dif.	Total	Dif.	Total	Dif.	Total	Dif.
C-type	TD	**4330.8**	**0.0**	4656.9	7.5	4910.9	13.4	4529.8	4.6	4724.1	9.1	4597.5	6.2
	FS	12.6	38.5	14.5	59.8	9.5	5.2	**9.1**	**0.0**	9.4	3.3	9.4	3.5
	VF	**454.2**	**0.0**	454.1	0.0	406.9	10.4	390.3	14.6	405.1	10.8	417.2	8.1
R-type	TD	**5958.2**	**0.0**	6634.7	11.4	6191.4	3.9	6009.5	0.9	6074.6	2.0	6016.9	1.0
	FS	12.8	19.6	13.2	23.7	12.0	12.3	11.1	3.7	10.8	1.2	**10.7**	**0.0**
	VF	424.5	3.2	**438.7**	**0.0**	407.8	7.1	401.4	8.5	395.9	9.8	404.3	7.8
RC-type	TD	**6249.6**	**0.0**	6946.0	11.1	6487.9	3.8	6540.9	4.7	6615.2	5.9	6353.3	1.7
	FS	13.3	22.5	15.1	38.5	11.9	9.2	**10.9**	**0.0**	11.5	5.5	11.3	3.7
	VF	**423.0**	**0.0**	422.7	0.1	398.5	5.8	392.6	7.2	399.4	5.6	396.6	6.2

Journal Pre-proof

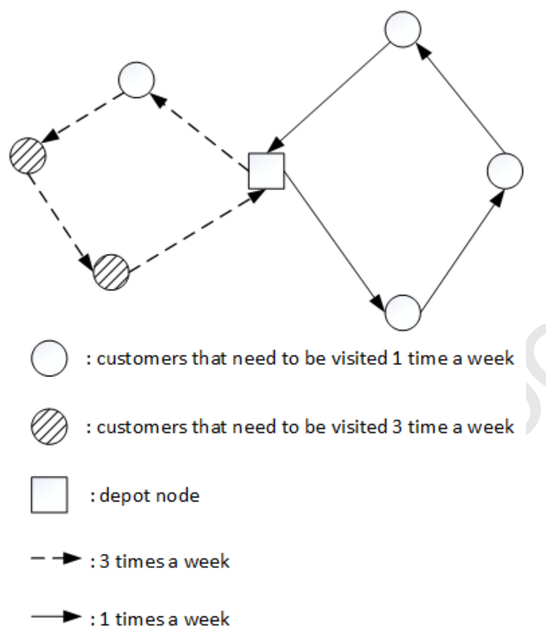


Figure 9: Example of visualized solution

visit frequency in total (shown in Table 3). And in R type instances, MOSA-ACO finds solutions with lowest travelling frequency in total. Compared with SPEA 2 which finds solutions with lowest total fleet size in C and RC type instances, MOSA-ACO finds solutions with average 3.37% lower on total travelling distance and 10.11% higher on total visit frequency. Compared with NSGA III which finds solutions with lowest total fleet size in R type instances, MOSA-ACO finds solutions with average 2.93% lower on total travelling distances and 7.41% higher on total visit frequency. And, MOSA-ACO also shows a robust performance in most tested Solomon's instances. Besides RC208-50, the maximum divisions of travelling distance, fleet size and visit frequency are 3.6% (RC203-50), 8.2% (R208-50) and 6.3% (RC203-50), respectively. Another thing should be noticed is that in RC208-50 instance, MOSA-ACO, SPEA 2, NSGA II and MOEA/D show more unstable performances than they show in other instances. The reason may be that in RC2 type instances, the original problem instances have conflict objectives, which means that reducing fleet size will increase travelling distance. And in RC2 type instances, customers have relatively long service time windows. Thus, customers will have more possibility to be added extra visit frequency. And

Journal Pre-proof

the algorithm performances become unstable. In VRPTW problem instances, MOSA-ACO has a longer computation time cost compared with other algorithms (except IACO). In all tested instances, the computation time cost of MOSA-ACO is 5 to 10 times longer compared with other algorithms, while other algorithms have almost the same computation time costs in all tested instances. We also test performances of MOSA-ACO and original IACO algorithms with Solomon's instances. Compared with IACO, MOSA-ACO finds solutions with 10.09% lower travelling distance, 13.8% lower fleet size and 1.03% lower total visit frequency on average. Since the difference of IACO and MOSA-ACO is whether the local search strategies are implemented, this phenomenon indicates that applying these local search heuristics brings reduction of service costs at the price of a slightly reduction of service profits.

Table 4: Mean numbers of travelling distance, fleet size and visit frequency in Cordeau's instances

		MOSA-ACO			IACO[24]			NSGA II[30]			SPEA 2[31]			MOEA/D[32]			NSGA III[33]		
		Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.	Ave.	Dev.	Dif.
pr01																			
	TD	4045.9	2.2%	2.0%	4500.45	1.1%	11.9%	4122.6	3.0%	3.9%	4087.8	5.2%	3.0%	4052.2	9.2%	2.2%	**3963.2**	8.9%	0.0%
	FS	5.4	3.7%	25.9%	5.91	2.2%	32.3%	**4**	6.8%	0.0%	**4**	6.3%	0.0%	4.1	6.1%	1.3%	4.2	16.2%	5.5%
	VF	**144.7**	2.7%	0.0%	142.52	1.6%	1.5%	137.1	3.7%	5.5%	135.1	4.3%	7.1%	131.1	7.4%	10.4%	132.8	7.7%	8.9%
	CPU(s)	2130.7			1405.7			144.3			143.2			221.7			232.0		
pr02																			
	TD	**7399.4**	2.9%	0.0%	8157.9	0.8%	9.3%	7663.6	4.4%	3.4%	7542.8	2.7%	1.9%	7506.8	8.8%	1.4%	7484.6	6.3%	1.1%
	FS	10.4	2.0%	23.2%	11.1	2.5%	27.9%	9	6.5%	11.2%	**8.8**	4.4%	9.2%	9.0	6.1%	11.2%	**8.0**	6.6%	0.0%
	VF	**307.6**	2.4%	0.0%	301.2	1.5%	2.1%	284.6	2.4%	8.1%	290.9	2.4%	5.7%	283.0	6.6%	8.7%	277.7	4.3%	10.8%
	CPU(s)	5532.8			4158.5			588.3			588.8			905.1			744.1		
pr03																			
	TD	**10802**	0.3%	0.0%	11787.8	1.0%	8.4%	11605.5	2.1%	6.9%	11924.8	2.0%	9.4%	11858.4	6.0%	8.9%	11700.3	2.7%	7.7%
	FS	14.3	2.1%	11.2%	15.4	1.4%	17.7%	12.8	2.8%	0.8%	13.1	8.1%	3.1%	**12.7**	6.8%	0.0%	13.3	5.4%	4.3%
	VF	**455.7**	0.5%	0.0%	454.8	1.0%	0.2%	422.9	1.2%	7.8%	424.3	1.4%	7.4%	432.0	5.4%	5.5%	448.1	2.1%	1.7%
	CPU(s)	10379			9078.8			1207.7			1347.5			1824.1			1513.4		
pr07																			
	TD	9543	1.6%	0.3%	10363.5	0.7%	8.2%	10979.5	2.9%	13.3%	10287.8	4.5%	7.5%	10772.6	5.7%	11.7%	**9516.6**	5.1%	0.0%
	FS	9.9	3.2%	19.9%	10.4	1.7%	23.8%	8.1	3.8%	2.1%	8.2	5.3%	3.3%	8.4	3.2%	5.8%	**7.9**	11.4%	0.0%
	VF	**347.2**	2.9%	0.0%	342.9	0.6%	1.2%	322.1	1.7%	7.8%	320.5	2.5%	8.3%	335.1	2.9%	3.6%	324.3	4.0%	7.1%
	CPU(s)	4695.1			3641.5			522.8			576.2			869.8			984.9		
pr08																			
	TD	**14790**	1.7%	0.0%	16401.2	0.6%	9.8%	15358.9	3.0%	3.7%	15778.2	3.8%	6.3%	15681.1	12.8%	5.7%	15241.5	5.2%	3.0%
	FS	15.2	2.0%	17.0%	16.6	1.1%	23.8%	13.2	3.4%	4.4%	13.1	3.7%	3.7%	12.6	5.3%	0.1%	**12.6**	7.5%	0.0%
	VF	679.5	1.6%	0.6%	**683.8**	0.5%	0.0%	625.9	0.9%	9.3%	623.2	1.3%	9.7%	626.5	4.7%	9.2%	635.2	1.6%	7.7%
	CPU(s)	12379.3			10206.3			2009.9			2351.7			2298.1			2399.0		
pr11																			
	TD	**3109.6**	1.3%	0.0%	3396.6	1.5%	8.4%	3362.1	3.8%	7.5%	3327.2	3.5%	6.5%	3294.1	8.7%	5.6%	3216.4	6.9%	3.3%
	FS	3.6	4.3%	2.8%	4.0	1.3%	12.9%	**3.5**	4.4%	0.0%	**3.5**	5.1%	0.0%	3.6	2.9%	3.8%	4.0	18.1%	13.0%
	VF	139	2.3%	1.9%	**141.7**	1.7%	0.0%	131.4	2.9%	7.8%	132.8	3.3%	6.7%	132.7	5.0%	6.7%	141.2	4.0%	0.3%
	CPU(s)	2778.2			2217.0			174.4			164.7			263.0			276.5		
pr12																			
	TD	**5865.9**	1.7%	0.0%	6313.3	1.2%	7.1%	6240.4	4.4%	6.0%	6572.3	3.8%	10.7%	6355.3	8.5%	7.7%	6114.8	6.8%	4.1%
	FS	7.8	2.7%	12.8%	8.2	2.2%	17.0%	**6.8**	5.4%	0.0%	6.9	4.4%	1.4%	7.1	5.9%	4.6%	7.3	8.0%	6.3%
	VF	295.9	2.2%	1.6%	**300.6**	1.1%	0.0%	274.6	2.0%	9.5%	278.8	2.1%	7.8%	283.6	3.9%	6.0%	282.8	2.3%	6.3%
	CPU(s)	5858.5			3962.4			745.8			780			938.3			863.8		
pr13																			
	TD	**8691.4**	0.8%	0.0%	9396.5	0.9%	7.5%	9204.9	2.9%	5.6%	9007.9	3.6%	3.5%	9366.7	8.4%	7.2%	9141.7	4.6%	4.9%
	FS	10.4	1.1%	11.9%	11.1	1.5%	17.7%	9.3	3.7%	1.4%	9.5	3.5%	3.5%	9.9	7.1%	7.8%	**9.2**	4.0%	0.0%
	VF	445.1	1.0%	0.3%	**446.4**	1.1%	0.0%	402.7	1.2%	10.9%	410.5	1.9%	8.8%	417.5	4.2%	6.9%	427.1	2.7%	4.5%
	CPU(s)	9972.4			8395.6			1622			1586.9			1851.7			1905.8		
pr17																			
	TD	**7533.2**	1.4%	0.0%	8004.0	0.9%	5.9%	8157.4	1.5%	7.7%	8502.5	5.9%	11.4%	8289.9	7.7%	9.1%	8150.3	4.4%	7.6%
	FS	6	3.2%	7.0%	6.4	2.0%	13.1%	5.7	2.8%	2.1%	5.8	4.4%	3.8%	5.8	5.5%	3.2%	**5.6**	3.3%	0.0%
	VF	333.1	2.8%	0.3%	**334.2**	0.9%	0.0%	315.7	1.3%	5.9%	329.4	3.3%	1.5%	317.3	5.0%	5.3%	322.3	3.1%	3.7%
	CPU(s)	5219.2			5322.9			668.8			730.6			883.3			902.5		
pr18																			
	TD	**12260**	1.1%	0.0%	12935.1	0.8%	5.2%	13044.5	3.0%	6.0%	13451	2.6%	8.9%	13025.2	8.4%	5.9%	12859.7	5.3%	4.7%
	FS	10.8	2.4%	14.1%	11.6	1.2%	20.1%	9.8	4.0%	5.3%	9.7	3.3%	4.4%	10.4	5.4%	11.2%	**9.3**	6.4%	0.0%
	VF	**675.1**	0.8%	0.0%	672.8	0.7%	0.3%	621	0.7%	8.7%	621.8	0.9%	8.6%	637.3	3.2%	5.9%	646.2	1.4%	4.5%
	CPU(s)	12001.1			12733.2			2426.9			2587.4			2715.4			2915.8		

In Cordeau's instances, MOSA-ACO also shows a good performance of

Journal Pre-proof

finding solutions with lower travelling distance and higher total visit frequency. In 4 of 10 tested instances, MOSA-ACO finds solutions with the lowest average travelling distance and highest total visit frequency (shown in Table 4). In 8 of 10 instances, MOSA-ACO finds solutions with lowest travelling, and in 6 instances MOSA-ACO finds solutions with minimum travelling distances. Compared with MOEA/D, MOSA-ACO finds solutions with 7.3% average lower travelling distance, 5.9% higher average visit frequency while total fleet size rises about 10.8%. Compared with MOEA/D, MOSA-ACO finds solutions with 4.0% average lower travelling distance, 4.8% higher average visit frequency while total fleet size rises about 13.3%. Compared with IACO, MOSA-ACO finds solutions with 7.4% lower fleet size and 8.5% lower travelling distance while the visit frequencies are almost the same. This result is coincidence with the result of Solomon1instances. In all tested 10 instances, MOSA-ACO shows a robust performance. The maximum divisions of travelling distance, fleet size and visit frequency are 2.9% (in pr02), 4.3% (in pr11) and 2.8% (in pr17). In Cordeau's instances, MOSA-ACO also has a higher computation cost compared with other algorithms. Detailed comparisons can be seen in Table 4.

Table 5: Instance information of Cordeau's instances

	pr01	pr02	pr03	pr07	pr08
Number of days	4	4	4	6	6
Number of customers	48	96	144	72	144
	pr11	pr12	pr13	pr17	pr18
Number of days	4	4	4	6	6
Number of customers	48	96	144	72	144

Finally, we can infer from table 4 and 5 that the computation cost of MOSA-ACO shows a linear growth trend as problem scale becomes larger. Table 5 shows the problem scales of all Cordeau's instances tested. For example, the problem scale of pr02 is 2 times of problem scale of pr01, and the computation cost of MOSA-ACO in pr02 is around 2.5 times of computation cost of pr01.

Journal Pre-proof

4.3. Hyper-volume performance analysis

Hyper-volume (HV) analysis is an efficient method to compare the performances of several multi objective optimization heuristic algorithms. HV analysis can compare qualities of non-dominated solution sets of different multi objective heuristics even when the actual Pareto Frontier is unknown. In this paper, we applied a Monte-Carlo method based HV indicator since the exact Pareto Frontier of each instance is unknown. This indicator can be obtained from https://ww2.mathworks.cn/matlabcentral/fileexchange/30785-hypervolume-computation. This indicator randomly generates points in an area between a given lower bound and a given upper bound. Then, it calculates the percentage of points dominated by a given solution set. A graphic description of this indicator is as follows.

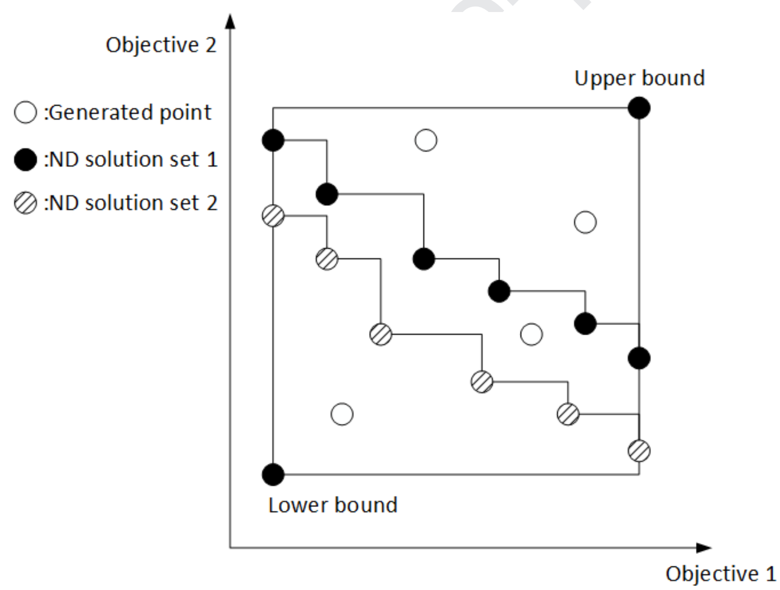


Figure 10: Monte-Carlo based HV analysis: ND solution set 1 dominates 2 points, ND solution set 2 dominates 3 points. Thus the HV coverage performance of ND solution set 1 is 50%, and the HV coverage performance of ND solution set 2 is 75%.

In this paper, the lower bound of each instance is set to the best values of three objectives found by all algorithms during 10 runs. The travelling distance and fleet size upper bounds of each instance are set to twice the total travelling distance and fleet size of the solution found by TWIH heuristic, which is a commonly applied lower bound of fleet size and travelling distance in multi objective VRP problems. And the visit frequency upper bound of each instance is set to the sum of minimum visit frequency of each customer. The sample time of each run is set to 1000000. And the simulation runs 10 times for each instance of all tested algorithms. Since the actual Pareto Optimum of each instance is unknown, we only analyze the coverage performances of the three algorithms. The average HV and analysis are shown as follows.

Table 6 and 7 show the average, maximum and minimum HV coverage percentages of the three algorithms. In Cordeau's instances, MOSA-ACO finds solutions with best average coverage performances in 7 of 10 tested instances. When the average coverage performances are considered, MOSA-ACO shows a good coverage performance of 78.7% on average of 10 tested instances. MOSA-ACO is followed by MOEA/D (78.2% on average) and NSGA III (74.0% on average). When the maximum coverage performances are considered, NGSA III and MOEA/D both have coverage performances of 81.6% on average of 10 tested instances. NSGA III and MOEA/D are followed by MOSA-ACO (81.2% on average). When the minimum coverage performances are considered, MOSA-ACO has the best performance of 75.9% on average. Compared with IACO, the average coverage performance of MOSA-ACO rises about 6.1% (78.7% to 72.6%). The experiment results of Cordeau's instances also indicate that MOSA-ACO has a robust performance on Cordeau's instances. Detailed experiment results of Cordeau's instances can be seen in Table 6.
In Solomon's instances, MOSA-ACO finds solutions with best average coverage performances in 8 of 12 tested instances. When the average coverage performances are considered, MOSA-ACO shows a good coverage performance of 64.6% on average of 12 tested instances. MOSA-ACO is followed by MOEA/D (63.9% on average) and NSGA III (62.5% on average). When the maximum coverage performances are considered, MOEA/D has the best coverage performances of 72.2% on average of 12 tested instances. MOEA/D is followed by MOSA-ACO (72.0% on average) and NSGA III (71.3% on average). When the minimum coverage performances are considered, MOSA-ACO has the best performance of 57.8% on average. In VRPTW instances, the coverage performance of MOSA-ACO rises about 9% (64.6% to 55.0%) when it is compared with IACO algorithm. In VRPTW instances, all 6 algorithms show more unstable coverage performances. The reason may be that

Table 6: HV analysis of Cordeau's instances(shown in percentage)

		MOSA-ACO	IACO[24]	NSGA II[30]	SPEA 2[31]	MOEA/D[32]	NSGA III[33]
pr01	Ave	56.2	52.4	53.6	53.5	**61.6**	56.2
	Max	57.4	55.2	61.4	56.0	66.0	**66.2**
	Min	**55.3**	48.9	48.2	49.8	54.5	48.9
	Dif.	8.8%	15.0%	13.0%	13.2%	**0.0%**	8.8%
pr02	Ave	**67.1**	62.7	62.8	65.0	65.6	62.8
	Max	68.7	64.5	67.7	67.5	67.7	**70.8**
	Min	**64.1**	59.5	60.4	59.4	63.0	55.2
	Dif.	**0.0%**	6.6%	6.4%	3.1%	2.2%	6.5%
pr03	Ave	80.9	76.2	72.8	70.6	**81.9**	76.0
	Max	83.2	78.5	75.5	74.8	**83.9**	82.4
	Min	**78.0**	73.7	69.1	69.1	78.4	68.7
	Dif.	1.3%	7.0%	11.2%	13.8%	**0.0%**	7.2%
pr07	Ave	79.7	68.8	74.9	78.3	**81.7**	75.7
	Max	80.7	70.6	79.8	81.9	**82.5**	81.4
	Min	77.6	67.3	74.5	76.9	**80.9**	69.8
	Dif.	2.4%	15.7%	8.3%	4.1%	**0.0%**	7.3%
pr08	Ave	**75.3**	72.2	73.9	71.9	74.3	75.2
	Max	77.5	74.8	75.0	72.9	**78.4**	**78.4**
	Min	**73.4**	69.6	71.5	69.8	72.6	70.4
	Dif.	**0.0%**	4.1%	1.9%	4.5%	1.4%	0.1%
pr11	Ave	**87.4**	78.4	77.2	76.3	86.3	79.1
	Max	93.8	86.1	86.5	84.7	93.6	**94.0**
	Min	**82.8**	74.0	65.7	67.1	82.5	70.8
	Dif.	**0.0%**	10.3%	11.7%	12.7%	1.2%	9.5%
pr12	Ave	**87.3**	78.1	76.4	78.6	85.5	80.9
	Max	**89.8**	82.7	80.3	69.9	88.3	89.7
	Min	**84.3**	78.7	72.5	61.8	82.9	72.7
	Dif.	**0.0%**	10.5%	12.5%	10.0%	2.1%	7.4%
pr13	Ave	**81.1**	80.0	76.7	77.1	79.1	76.2
	Max	84.7	79.9	78.2	79.0	**85.4**	83.3
	Min	**77.3**	76.4	75.0	76.4	73.2	72.2
	Dif.	**0.0%**	1.3%	5.4%	4.9%	2.4%	6.0%
pr17	Ave	**91.1**	83.4	84.2	83.6	85.6	81.1
	Max	92.5	85.2	89.5	**92.8**	88.7	91.0
	Min	**88.8**	80.3	80.1	74.1	80.2	73.7
	Dif.	**0.0%**	8.5%	7.6%	8.2%	6.0%	11.0%
pr18	Ave	**80.5**	73.7	71.0	72.8	79.4	76.7
	Max	**83.4**	77.2	73.4	74.8	81.9	78.4
	Min	**77.6**	72.9	68.2	71.5	75.4	74.2
	Dif.	**0.0%**	8.4%	11.8%	9.6%	1.4%	4.7%
Average		78.7	72.6	72.4	72.8	78.1	74.0
	Dif.	**0.0%**	7.7%	8.0%	7.5%	0.7%	5.9%

Journal Pre-proof

Journal Pre-proof

Table 7: HV analysis of Solomon's instances(shown in percentage)

		MOSA-ACO	IACO[24]	NSGA II[30]	SPEA 2[31]	MOEA/D[32]	NSGA III[33]
C103-50	Ave	73.6	63.1	73.0	67.9	**75.4**	72.8
	Max	81.0	68.1	80.0	86.2	**82.3**	80.5
	Min	69.1	59.4	66.9	58.6	**70.4**	66.4
	Dif.	2.4%	16.3%	3.1%	9.9%	**0.0%**	3.5%
C108-50	Ave	**59.6**	53.8	50.3	50.7	57.2	54.3
	Max	62.9	58.0	54.8	58.5	**64.2**	58.7
	Min	**56.0**	50.6	43.9	43.5	49.9	49.1
	Dif.	**0.0%**	9.7%	15.6%	14.9%	4.0%	8.9%
C203-50	Ave	**75.8**	66.2	72.6	69.3	74.8	74.0
	Max	**96.8**	69.7	82.9	88.9	82.0	81.1
	Min	50.9	63.3	58.6	51.1	**68.5**	68.1
	Dif.	**0.0%**	12.7%	4.2%	8.6%	1.4%	2.4%
C208-50	Ave	**61.3**	49.4	47.1	46.3	53.1	51.7
	Max	63.9	53.5	53.9	57.5	**66.4**	58.7
	Min	**59.2**	45.6	40.2	37.6	49.5	44.6
	Dif.	**0.0%**	19.4%	23.2%	24.5%	13.3%	15.6%
R103-50	Ave	**68.4**	65.5	62.9	63.8	66.1	64.3
	Max	71.8	69.8	**75.1**	74.0	72.8	72.6
	Min	**62.6**	59.5	50.8	53.1	60.6	56.4
	Dif.	0.0%	4.2%	8.0%	6.7%	**3.3%**	6.1%
R108-50	Ave	62.0	57.5	62.9	61.9	**64.0**	63.2
	Max	67.6	66.5	69.1	67.5	**70.4**	71.3
	Min	**58.4**	51.9	55.5	56.4	56.3	56.5
	Dif.	3.1%	10.1%	1.6%	3.2%	**0.0%**	1.1%
R203-50	Ave	57.7	52.7	61.4	53.6	63.2	**63.5**
	Max	59.9	62.9	65.6	59.2	**71.7**	67.0
	Min	**55.6**	46.6	54.3	49.1	54.7	45.7
	Dif.	9.1%	17.0%	3.3%	15.6%	0.5%	**0.0%**
R208-50	Ave	**73.4**	67.0	70.4	68.1	72.6	70.2
	Max	80.7	**83.3**	82.6	72.5	78.5	81.4
	Min	**65.7**	52.8	55.9	34.2	62.5	59.1
	Dif.	**0.0%**	8.7%	4.1%	7.2%	1.1%	4.4%
RC103-50	Ave	55.1	37.3	58.5	54.8	**62.0**	60.2
	Max	54.1	40.6	63.1	60.7	70.8	**76.0**
	Min	52.5	32.9	54.3	52.4	**57.5**	55.1
	Dif.	11.2%	39.8%	5.7%	11.6%	**0.0%**	2.9%
RC108-50	Ave	**74.4**	52.3	67.2	68.1	66.0	65.9
	Max	80.3	58.3	79.8	**83.5**	76.6	73.4
	Min	**65.9**	46.8	53.5	59.5	61.0	58.0
	Dif.	**0.0%**	29.7%	9.7%	8.5%	11.4%	11.4%
RC203-50	Ave	**61.1**	54.0	58.0	59.4	60.6	59.7
	Max	67.6	56.0	64.8	76.1	72.2	**72.5**
	Min	**58.1**	50.3	44.5	41.0	53.8	52.0
	Dif.	**0.0%**	11.6%	5.1%	2.8%	0.9%	2.3%
RC208-50	Ave	**52.2**	40.9	44.8	45.6	51.6	50.3
	Max	**76.9**	44.1	57.3	56.2	58.1	62.0
	Min	39.3	36.8	39.4	38.0	**43.5**	39.4
	Dif.	**0.0%**	21.6%	14.2%	12.6%	1.2%	3.6%
Average		64.6	55.0	60.8	59.1	63.9	62.5
	Dif.	**0.0%**	14.8%	5.9%	8.4%	1.1%	3.2%

R1, R2, RC1 and RC2 types of instances have contradictory objectives, while C1 and C2 types of instances have positive correlating objectives (which is suggested in Garcia-Najera and Bullinaria [28]). Detailed experiment results of Solomom's instances can be seen in Table 7.

5. Conclusion

In this article, we discuss a new variant of existing PVRP-SC problem called PVRPTW-SC. We first give out the problem model based on the existing PVRP-SC problem model. Then, to solve this problem, we develop a hybrid multi-objective heuristic algorithm called MOSA-ACO. MOSA-ACO algorithm is a combination of MOSA algorithm and IACO algorithm. Main algorithm modifications include a new solution acceptation criterion of MOSA and a new route construction method of IACO. To help MOSA-ACO find solutions with better quality, a K-Mean based customer distribution method and 6 local search strategies are also implemented. Benchmark instances generated from Solomon's benchmark instances and Cordeau's instances are used to test the performance of MOSA-ACO. Comparison algorithms include 4 widely applied population-based heuristics and original IACO algorithm. Experiment results show that MOSA-ACO has a good and robust performance on 2 benchmark instance sets with relatively higher computation costs. High computation cost of MOSA-ACO is caused by that MOSA-ACO follows a constructive heuristic to build the initial solution. During the constructive process, the constraint check runs on every possible move to make sure that the solution is feasible. That means if an instance has 100 customers, the constraint check operator runs 100! times. Though the population-based heuristics also have constraint check process, the process is only applied after a solution is generated. Thus, if an instance has 100 customers, the constraint check operator runs only 100 times at most in one move. Though the best acceptance principle is applied in mutation operators which brings increases in computation cost, population-based heuristics are more computational efficient than MOSA-ACO. Considering that when the temperature is low and MOSA-ACO need to reinitialize frequently, the high computation costs are reasonable.

Further research can be carried out in these directions: a) Techniques dealing with customer selection is needed. In MOSA-ACO, we implement a K-mean based customer selection technique to deal with customer selection problem of PVRTW-SC. However, other factors such as average length of customer time windows may also affect the selection of customers. b) Problem decomposition should be further discussed. In this article, we divide PVRPTW-SC problem into a PVRPTW problem with lowest visit frequency constraint and a VRPTW problem. This decomposition may lose some solution space. So, improvement can be made on problem model or problem decomposition method. c) Relations between enlarging visit frequency and reducing service costs need to be further analyzed. d) Powerful heuristic operators need to be implemented to improve algorithm performance. In this paper, heuristic operators implemented in MOSA-ACO are designed to solve PVRP problems and VRPTW problems. To further improve algorithm performance, powerful operators should be designed.

[1] L. Jiting, S. Zhang, X. Liu and R. He, "Multi-objective evolutionary optimization for geostationary orbit satellite mission planning," in *Journal of Systems Engineering and Electronics*, Vol.28(5), pp.934-945.

[2] E. J. Beltrami and L. D. Bodin, "Networks and vehicle routing for municipal waste collection," in *Networks,* Vol.4(1), pp. 65-94, 1974.

[3] P. M. Francis, K. R. Smilowitz, and M. Tzur, "The Period Vehicle Routing Problem and its Extensions," in The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, vol.43, B. Golden, S. Raghavan, and E. Wasil, Ed., 1st ed. Springer, Boston, US, 2008, pp. 73-102.

[4] P. M. Francis, K. R. Smilowitz, and M. Tzur, "The period vehicle routing problem with service choice," in *Transportation Science,* vol.40, no.4, pp. 439–454, 2006.

[5] P. M. Francis, K. R. Smilowitz, and M. Tzur, "Flexibility and complexity in periodic distribution problems," in *Naval Research Logistics,* vol.54, no.2, pp. 136–150, 2010.

[6] G. Laporte, and A. Mercier, "Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows," in *Journal of the Operational Research Society,* vol.55, no.5, pp. 542–546, 2004.

Journal Pre-proof

[7] J. Michallet, C. Prins, L. Amodeo, F. Yalaoui, and G. Vitry, "Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services," in *Computers & Operations Research,* vol.41, no.41, pp. 196–207, 2014.

[8] S. H. Mirmohammadi, E. Babaee Tirkolaee, A. Goli and S. Dehnavi 8C Arani, "The Periodic Green Vehicle Routing Problem with Considering of Time-dependent Urban Traffic and Time Windows," in *International Journal of Optimization in Civil Engineering,* vol.7, no.1, pp. 143–165, 2017.

[9] A. Mingozzi, "The multi-depot periodic vehicle routing problem," presented at the *6th International Symposium on Abstraction, Reformulation and Approximation,* pp. 347–350, Airth Castle, UK, July 26-29, 2005.

[10] R. Cantu-Funes, M. A. Salazar-Aguilar, and V. Boyer, "Multi-depot periodic vehicle routing problem with due dates and time windows," in *Journal of the Operational Research Society,* vol.69, no.2, pp. 296–306, 2018.

[11] T. Athanasopoulos, and I. Minis, "Efficient techniques for the multi-period vehicle routing problem with time windows within a branch and price framework," in *Annals of Operations Research,* vol.206, no.1, pp. 1–22, 2013.

[12] J. F. Bard, and N. Nananukul, "A branch-and-price algorithm for an integrated production and inventory routing problem," in *Computers & Operations Research,* vol.37, no.12, pp. 2202–2217, 2010.

[13] N. H. Moin, S. Salhi, and N. A. B. Aziz, "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem," in *International Journal of Production Economics,* vol.133, no.1, pp.334–343, 2011.

[14] J. H. Song, and K. C. Furman, "A maritime inventory routing problem: practical approach," in *Computers & Operations Research,* vol.40, no.3, pp. 657–665, 2013.

[15] M. F. Tasgetiren, P. N. Suganthan, and Q. K. Pan, "An ensemble of discrete differential evolution algorithms for solving the generalized

Journal Pre-proof

traveling salesman problem," in *Applied Mathematics & Computation,* vol.215, no.9, pp. 3356–3368, 2010.

[16] J. Q. Li, Q. K. Pan, K. Mao, and P. N. Suganthan, "Solving the steelmaking casting problem using an effective fruit fly optimisation algorithm," in *Knowledge–Based Systems,* vol.72, no.5, pp. 28–36, 2014.

[17] Y. Han, D. Gong, Y. Jin, and Q. Pan, "Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns," in *IEEE Transactions on Cybernetics,* vol.99, pp. 28–36, 2017.

[18] Q. K. Pan, L. Wang, and Q. J. Li, "A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimization," in *OMEGA–International Journal of Management Science,* vol.45, pp. 42–56, 2014.

[19] W. Gong, and Z. Cai, "A multiobjective differential evolution algorithm for constrained optimization," presented at the *IEEE Congress on Evolutionary Computation,* pp. 347–350, Hong Kong, China, July 1-6, 2008.

[20] L. Gao, Y. Zhou, X. Li, Q. Pan, and W. Yi, "Multi–objective optimization based reverse strategy with differential evolution algorithm for constrained optimization problems," in *Expert Systems with Applications,* vol.42, no.14, pp. 5976–5987, 2015.

[21] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," in *IEEE Transactions on Cybernetics,* vol.45, no.4, pp. 716–727, 2015.

[22] J. Li, Q. Pan, and S. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," in *Applied Mathematics & Computation,* vol.218, no.18, pp. 9353–9371, 2012.

[23] J. Li, Q. Pan, and S. Xie, "Flexible job shop scheduling problems by a hybrid artificial bee colony algorithm," in *Evolutionary Computation,* vol.6, pp. 78–83, 2011.

[24] B. Yu, and Z. Z. Yang, "An ant colony optimization model: the period vehicle routing problem with time windows," in *Transportation Research Part E,* vol.47, no.2, pp. 166–181, 2011.

Journal Pre-proof

[25] S. Thomas, and H. H. Holger, "MAX-MIN Ant system," in *Future Generation Computer Systems,* vol.16, no.9, pp. 889–914, 2000.

[26] Croes, G. A., " A method for solving traveling-salesman problems," in *Operations Research,* 6(6), 791-812, 1958.

[27] Lin, S., "Computer solutions of the traveling salesman problem," in *Bell Labs Technical Journal,* Vol.44(10), 2245-2269, 1965.

[28] Garcia-Najera, A., and Bullinaria, J. A., "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," in *Computers and Operations Research,* Vol.38(1), 287-300, 2011.

[29] Baños, R., Ortega, J., Gil, Consolación, Márquez, Antonio L., and De Toro, F., "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," in *Computers & Industrial Engineering,* Vol.65(2), pp. 286-296, 2013.

[30] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation,* Vol.6(2), 182-197, 2002.

[31] Zitzler, E., Laumanns, M., and Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *proceedings of evolutionary methods of design, optimization, and control, Barcelona,* Barcelona, Spain, 95-100, 2001.

[32] Zhang, Q. and Li, H., "Moea/d: a multiobjective evolutionary algorithm based on decomposition." in *IEEE Transactions on Evolutionary Computation,* Vol.11(6), 712-731, 2007.

[33] Deb, K. and Jain, H., "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints." in *IEEE Transactions on Evolutionary Computation,* 18(4), 577-601, 2014.

[34] Zou, J., Fu, L. W., Yang, S. X., Zheng, J. H., Ruan, G., Pei, T. R., Wang, L., "An adaptation reference-point-based multiobjective evolutionary algorithm," in *Information Science,* Vol.488, 41-57, 2019.

Journal Pre-proof

[35] Tan, K. C., Chew, Y. H., and Lee, L. H., "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," in *Computational Optimization & Applications,* Vol.34(1), 115-151, 2006.

[36] Alvarenga, G. B., Mateus, G. R., and Tomi, G. D., "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows," in *Computers & Operations Research,* Vol.34(6), 1561-1584, 2007.

[37] El-Sherbeny, N. A., "Vehicle routing with time windows: an overview of exact, heuristic and metaheuristic methods," in *Journal of King Saud University Science,* Vol.2(3), pp. 123-131, 2010.

[38] R., Masson, F., Lehuédé, O., Péton, "An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Transfers," in *Transportation Science,* Vol. 47(3), 295-454, 2013.

Journal Pre-proof

Journal Pre-proof

AUTHOR DECLARATION TEMPLATE

We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work. [OR]
We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We further confirm that any aspect of the work covered in this manuscript that has involved either experimental animals or human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript. [CAN BE DELETED IF NOT RELEVANT]

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from (PLEASE CONFIRM WHAT E-MAIL ADDRESS YOU WISH TO INSERT HERE)

Signed by all authors as follows:

[LIST AUTHORS AND DATED SIGNATURES ALONGSIDE]

Yuan Wang 2019.5.6
Ling Wang 2019.5.6
Ling Xing 2019.5.6
Guangcai Chen 2019.5.6
Zhenquan Cai 2019.5.6
Yongquan Zhou 2019.5.6