Weapon configuration, allocation and route planning for a fleet of unmanned combat air vehicles

Jiaming Zhang, Jianmai Shi, Hongtao Lei, Zhong Liu*

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, P.R. China, 410073

* jianmaishi@gmail.com

Abstract

There is an increasing trend that Unmanned Combat Air Vehicles (UCAVs) are employed to complete different combat missions in modern wars. This paper investigates a UCAV routing problem, which simultaneously considers the decisions for the configuration of weapons carried by the UCAV subject to its capacity and the allocation of weapons to the targets subject to their destroying requirements. An integer linear programming model is developed to formulate the problem. An adaptive large neighborhood search (ALNS) heuristic is proposed to solve the problem, in which seven neighborhood structures are designed and employed. Randomly generated instances covering the small, medium and large sizes are used to test the proposed ALNS algorithm. CPLEX is also utilized to solve the small-size instances, whose results are compared with that obtained by the ALNS algorithm. And the extensive experimental results confirm the effectiveness and superiority of the proposed ALNS algorithm.

***Keywords**: Unmanned Air Vehicles; Mission planning; Routing; Weapon configuration; adaptive large neighborhood search*

1. Introduction

Unmanned Combat Air Vehicle (UCAV) is usually considered to be a fighter aircraft without a pilot. Due to the advantages of UCAVs in conducting military operations, such as without loss of human life, lower cost and lower probability be detected [1], they have been widely employed in many kinds of military operations in modern wars. When UCAVs are utilized in military operations for attacking multiple targets, the commanders usually have to consider many issues for the mission planning, including which and how many weapons to be equipped on the UCAV, which and how many weapons to be delivered to the target by the UAV, and the route of UCAV to visit the targets. When the decisions for weapon configuration on UCAV and allocation to targets are integrated into the UCAV routing problem, a more complex and new extension to the traditional UAV routing problem is generated.

In a typical UCAV mission planning problem, there are a number of predetermined targets that must be eliminated or neutralized. The geographical coordinates and basic attributes of the targets usually have been obtained from reconnaissance departments. For different military targets, damage demands are different and determined by the commander. Note that, in our paper, each target can only be visited once and must be destroyed in this visit. If not, the target may be repaired and protected during the time between two visits from different UCAVs. Table 1 presents an illustration example for five targets in a mission, which include their position and the damage demand for destroying the target.

Table 1 Damage requirements for different types of target.

Target	Description	Damage demand	Position
T1	Bridge	0.85	(66.27, 88.92)
T2	Communications station	0.81	(29.82, 52.35)
T3	Bunker	0.95	(90.91, 52.35)
T4	Arsenal	0.84	(11.42, 94.11)
T5	Airbase	0.95	(69.79, 25.33)

There are a number of UCAVs that can be potentially used in the mission. Table 2 presents some critical information for a typical UCAV, including the endurance time, the payload, and the number of hanging points, the types of weapons can be carried and so on. There are three types of

weapons that can be carried by this UCAV, whose weights and costs are different and shown in Table 3. For different weapons, the destroying effects are different if they are used to attach the same target. Table 4 presents the destroying effects between weapons and targets. Here, if a UCAV attacks target T5 (airbase) by utilizing weapon W1 (small smart bombs), a damage level of 0.02 units can be obtained. This can be explained by the fact that it's extremely difficult to destroy an airbase with a low-yield bomb. Thus, the UCAV carries different types of weapons, and the combat ability is also quite different.

Table 2 The UCAV basic parameters.

Index	Parameter values
Cost ($ million)	10
Payload capacity (kg)	900
Full-loaded Endurance (h)	14
Cruise speed (km/h)	180
Hardpoints	6
Types of weapons	W1, W2 and W3

Table 3 The weapon basic parameters.

Weapon	Description	Weight (kg)	Cost ($ million)
W1	Small smart bombs	75	0.068
W2	Small precision guided bombs	165	0.084
W3	Laser-guided bomb	240	0.002

Table 4 Weapon-target combat ability matrix.

	T1	T2	T3	T4	T5
W1	0.35	0.70	0.35	0.85	0.02
W2	0.65	0.80	0.86	0.67	0.25
W3	0.95	0.95	0.77	0.75	0.85

From the above illustration of the typical combat mission, we can see that there are two kinds of critical decisions faced by the commander. The first is to determine the targets attacked by each UCVA and its flying route for visiting these targets. The second is to determine which type and how many weapons should be carried by each UCAV and which weapon should allocated to each target, subject to constraints on the targets' damage demand and the limited capacity of UCAV on load and Hardpoints. The commanders have to optimize the decisions on weapon configuration, allocation and routing for multiple UCAVs, while minimizing the overall costs to complete the mission.

In this paper, we address the joint weapon configuration, allocation and routing problem generated in the UCAV mission planning field, which is frequently faced by commanders in modern

wars. We develop an integer linear programming model for the problem. As the scale of the problem becomes large, standard solvers (e.g. CPLEX) cannot obtain an optimal solution efficiently. An adaptive large-scale neighborhood search algorithm is proposed to solve the problem. This study enriches the UCAV routing problem in the following aspects.

(1) This paper is the first to consider a joint optimization problem of UCAVs route planning with weapon configuration and allocation, during which the UCAVs' routing, the weapon configuration for the UCAVs and the weapon allocation for the targets are optimized simultaneously. And we provide an integer linear programming model for the focused problem.

(2) The interactive influence among weapon configuration, allocation and routing makes the problem more complex. For the solving, we develop an adaptive large neighborhood search heuristic, which merges the weapon allocation strategy, several novel neighborhood operators and adaptively learning strategy. A population-based neighborhood search strategy is also employed to improve the search diversification in each iteration. The extensive computational experiments confirm the effectiveness and superiority of the developed algorithm.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review of the UCAVs mission planning problem. Problem description and mathematical formulation are presented in Section 3. Section 4 is devoted to the solution approach developed. Computational results of the solution method on a variety of numerical examples are presented in Section 5. Finally, the conclusions are presented in Section 6.

2. Literature Review

The weapon configuration, allocation and route planning are important mission planning operations for multiple UAVs. In modern wars, more and more military missions are conducted by UAVs, such as attack, intelligence, surveillance/reconnaissance missions. The study of UAV mission planning has gained increasing attention in recent years, and more practical applications are also presented [2-6]. Reviews for earlier research on the UAV mission planning were presented in [7] and [8]. Most of current studies on UAVs' mission planning problem optimize the decisions for determining when to perform their task, how to complete the task, and making it complete with the minimum

cost [9].

The route planning of UAVs is considered to be the most critical and challenging problem in wartime[10], and there has been significant research in this area. The objective is to find out an optimal or near-optimal flight path for each UAV between an initial location and the desired destination under specific constraint conditions [11]. Edison and Shima [12] investigated the integrated task assignment and path planning problem for multiple UAVs, where the UAVs have to perform consecutive tasks cooperatively on ground targets. The UAV's minimum turn radius is considered, and the genetic algorithm is applied to solve the problem. Liu et al. [13] studied the task assignment problem for multiple UAVs to attack targets with dynamic value, which is solved by combining the multi-destination route planning algorithm and the ant colony algorithm. The constraints on flying height, maximum climbing/diving rate, and maximum turning angle are considered during planning the UAVs' route. When planning the path of UAVs, the control features, e.g., flying height, minimum turn radius and flying velocity, are usually considered and optimized.

As the development of UAV's automatic control technologies, most current UAVs in military area can automatically complete the flying path between targets. Thus, recent studies tend to ignore the detail control of UAV in the flying path and optimize the UAV routing problem for visiting targets from the operational level. Shetty et al. [14] studied the priority-based target assignment and routing problem for multiple UCAVs, where the total service to targets is maximized based on their criticality. The problem is divided into a target assignment subproblem and a vehicle routing subproblem, and is solved by the Tabu search heuristic. Mufalli et al. [15] investigated joint sensor selection and routing problem for multiple UAVs, where the endurance of the UAV is considered and depends on the weight of carried sensors. The problem is formulated as a generalization of the team orienteering problem studied in the vehicle routing literature. Evers et al. [16] studied UAV reconnaissance mission planning problem with time windows and time-sensitive targets, which is modeled as the maximum coverage stochastic orienteering problem with time windows. They first construct a tour with maximum expected profits of targets and then develop a fast heuristic to re-plan the tour in the event a time-sensitive target appears.

Location and routing of UAVs is also an important topic in the UAV mission planning. Sarıçiçek and Akkuş [17] studied the hub-location and routing problem for border security, where UAVs are utilized to route a number of points in the land border of Turkey. The problem is solved

by a two-stage approach. The first stage is to select hubs among the current airports as the base of UAVs. The route for visiting the border points are optimized in the second stage. Yakıcı [18] address the location and routing problem for small UAVs at tactical level, and the ant colony algorithm is employed to solve the problem.

From current literature on UAV path/route planning, it can be seen that most of works on this topic contribute on optimizing the flying path between targets or the route visiting targets. Few of them considered the configuration of weapons and the interactive influence on the routing of UAVs.

3. Model Formulation

3.1 Problem Description

In this section we will provide a detailed description of the weapon configuration, allocation and routing problem for UCAVs.

(1) Targets

The problem is given by a set of targets $N=\{1,2,\cdots,n\}$, residing at n different locations, and node 0 denotes the depot. Each pair of locations (i,j), where $i,j\in\{0\}\bigcup N$, and $i\neq j$, is associated with a travel time t_{ij}. Each target i is assigned a specific damage level by the commander, denote by a_i ($i\in N$). Each target can be attacked only once by one of the UCAVs.

(2) UCAVs

There are a fleet of homogeneous UCAVs used to attack the targets, noted as $U=\{1,2,\cdots,u\}$ which have the same payload capacity and number of hardpoints. However, each UCAV i can carry different weapons and then have different combat performance. With MQ-9 Reaper, for example, it can carry two 250 kilograms of precision guided bombs or fourteen hell fire anti-tank missiles[19]. The UCAVs leave and return to the depot. Most typical UCAVs in term of military service, such as the one shown in Table 2, have an endurance over than 10 hours, which is much longer than the our mission planning period. Thus, we do not consider the constraints on the endurance of the UCAVs in this paper.

(3) Weapons

There are a number of potential weapons, noted as $W=\{1,2,\cdots,w\}$, which can be equipped in the UCAV. Different types of weapons have different weights and different capabilities, e.g. the weapons shown in Table 3 and Table 4. For weapon $m\in W$, we assume its weight is q_m, its cost is f_m and its capability of destroying target $i\in N$ is b_{im}.

3.2 Mathematical Model

The notations used in the formulation are summarized as follows.

1) Sets

N: the set of targets, and $N=\{1,2,\cdots,n\}$.

U: the set of UCAVs, and $U=\{1,2,\cdots,u\}$.

W: the set of different weapon types, and $W=\{1,2,\cdots,w\}$.

2) Parameters

c: the payload capacity of the UCAV.

g: the number of hardpoints of the UCAV.

a_i: damage demand at target i.

d_{ij}: distance between target i and target j.

f_m: the cost of a weapon of type m, and $m\in W$.

q_m: the weight of a weapon of type m, and $m\in W$.

b_{im}: the combat ability of weapon m on target i, and $m\in W$.

L: a large enough number.

3) Decision Variables

x_{ijk}: binary variable that is equal to 1 if a target j is attacked after target i by UCAV k, and 0 otherwise.

y_{kmi}: integer variable that denotes the number of weapon m on UCAV k used to attack target i, and $y_{kmi}\geq 0$.

The joint weapon configuration, allocation and route planning problem for UCAVs can be formulated as follows：

Minimize $$Z = P_1 \sum_{j=1}^{n} \sum_{k=1}^{u} x_{0jk} + P_2 \sum_{m=1}^{w} \sum_{k=1}^{u} \sum_{i=1}^{n} f_m y_{kmi} + P_3 \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{u} d_{ij} x_{ijk} \quad (1)$$

subject to:

$$\sum_{k=1}^{u} \sum_{i=0, i\neq j}^{n} x_{ijk} = 1, \quad \forall j \in N \quad (2)$$

$$\sum_{k=1}^{u} \sum_{j=0, j\neq i}^{n} x_{ijk} = 1, \quad \forall i \in N \quad (3)$$

$$\sum_{i=0}^{n} x_{ipk} - \sum_{j=0}^{n} x_{pjk} = 0, \quad \forall k \in U, p = 0,1,\cdots,n \quad (4)$$

$$\sum_{j=1}^{n} x_{0jk} \leq 1, \quad \forall k \in U \quad (5)$$

$$\sum_{i=1}^{n} x_{i0k} \leq 1, \quad \forall k \in U \quad (6)$$

$$\sum_{m=1}^{w} q_m \sum_{i=1}^{n} y_{kmi} \leq c, \quad \forall k \in U \quad (7)$$

$$\sum_{m=1}^{w} \sum_{i=1}^{n} y_{kmi} \leq g, \quad \forall k \in U \quad (8)$$

$$\sum_{k=1}^{u} \sum_{m=1}^{w} b_{im} y_{kmi} \geq a_i, \quad \forall i \in N \quad (9)$$

$$y_{kmi} \leq L \sum_{j=1}^{n} x_{ijk}, \quad \forall i \in N, k \in U, m \in W \quad (10)$$

$$y_{kmi} \geq 0, \quad \forall k \in U, i \in N, m \in W \quad (11)$$

$$x_{ijk} \in \{0,1\}, \quad \forall i \in N, j \in N, k \in U \quad (12)$$

Formula (1) is the objective function of the problem. The objective of the model is to minimize the number of UCAVs, the sum of the fixed weapons cost and the distances traveled by all UCAVs. The first term represents the number of UCAVs used, and thus only the non-empty routes are computed. The second term represents the total costs for all weapons used to complete all tasks. The third term represents the total distances traveled by all UCAVs. P_1, P_2 and P_3 are the

coefficients used to adjust the costs of UCAV, weapons, and the flights into the same measure unit.

Constraints (2) and (3) define that every target node can be struck only once by one UCAV. Constraints (4)-(6) state that each UCAV that leaves the depot, after arriving at a target the UCAV leaves again, and it must finally return to the depot. Constraints (7) guarantee that the payload capacity of the UCAV must not be exceeded. Constraints (8) ensure that the number of hardpoints of each UCAV must not be exceeded. Constraints (9) regulate that the damage demand of each target must be fulfilled. Constraints (10) guarantee that the decision variable y_{kmi} can only make sense if a determined UCAV goes by that target. Constraints (11) guarantee that the decision variable y_{kmi} is positive. Constraints (12) assure that the decision variables x_{ijk} to be binary.

With the development of technologies, the endurance of new UCAVs have been significantly improved, even in full load condition, the endurance can reach more than 10 hours[20]. When UCAVs are employed in battlefield, there are two typical mission situations. One is that the information of target is insufficient, and the UCAV has to cruise in an area to wait for the appearance of the target and then starts the stacking action. The other is that the information of the target is sufficient, and the UCAV can fly to the precise position of the target and then attacks in time. In the former situation, we should consider the endurance of UCAV and the taking turns of different UCAVs, while in the later situation, the endurance is long enough for the UCAV to attack several targets and return to the depot. Our problem belongs to the later situation and we do not consider the constraints on the UCAV's endurance [16].

4. Adaptive Large Neighborhood Search Heuristic

In this section, an adaptive large neighborhood search heuristic (ALNS) is developed for solving the problem. ALNS algorithm was proposed by Ropke and Pisinger [21] for solving the pickup and delivery problem with time windows, and then applied to arc routing problem by Laporte et al. [22], vehicle routing problem by Lei et al. [23] and Dayarian et al. [24], and territory design problem by Lei et al. [25], which shows better performance in solving these problem. In the iteration process the ALNS for the proposed problem, a population-based neighborhood search strategy is utilized instead of the general single-solution based search, so as to improve the search diversification.

Population-based neighborhood search strategies have been used to solve complex combinational problems, e.g. the machine total weighted tardiness problem[26], job shop scheduling problem[27] and vehicle routing problem[28], which show good performance on improving the search diversification. Thus, the ALNS for our problem includes two main stages: initial solutions generation in the first stage, where a number of feasible solutions are generated through a constructive heuristic; and the neighborhood search improvement in the second stage, where a group of neighborhood structures are defined and utilized to search better solutions.

4.1 Framework of the ALNS

The framework of the ALNS algorithm for the proposed weapon configuration, allocation and routing problem is illustrated in **Fig. 1**.

In the first stage for generating initial feasible solutions, the construction heuristic includes three parts: firstly, weapons are allocated to each target to satisfy its destroy requirement through randomly selecting one of the three weapon allocation operators; then a tour for visiting all targets are constructed by randomly calling one of the three tour construction operators; at last, the tour is segmented into sub-tours according to the capacities of UCAVs on load and hanging points. In the initial solution generation process, both random operators, such as randomly allocating weapons to targets, and greedy operators, such as Efficiency-cost ratio priority based allocation of weapons, are employed and selected according to the roulette-wheel principle. Thus, a group of initial feasible solutions, G=50, are generated, which include better feasible solutions as well as diverse solutions. The processes of the weapon allocation operators are detailed in Section 4.3.1.

In the second stage, based on the above group of initial solutions, a large neighborhood search strategy with multiple start points is employed, and seven neighborhood structures are defined and used to search the neighborhood solution space related with the decision variables on configuration, allocation and routing. In each iteration of the neighborhood search, a number of G neighbors are generated respectively from the G solutions kept from the last iteration. The neighborhood structures are further introduced in Section 4.4. In the generation process of neighbors, the selection probabilities of the seven neighborhood operators are random, which are dynamically adjusted throughout the iteration process based on an adaptive learning strategy proposed by [21]. We

present a detail introduction of the adaptive learning strategy employed for guiding the neighborhood search in Section 4.5.

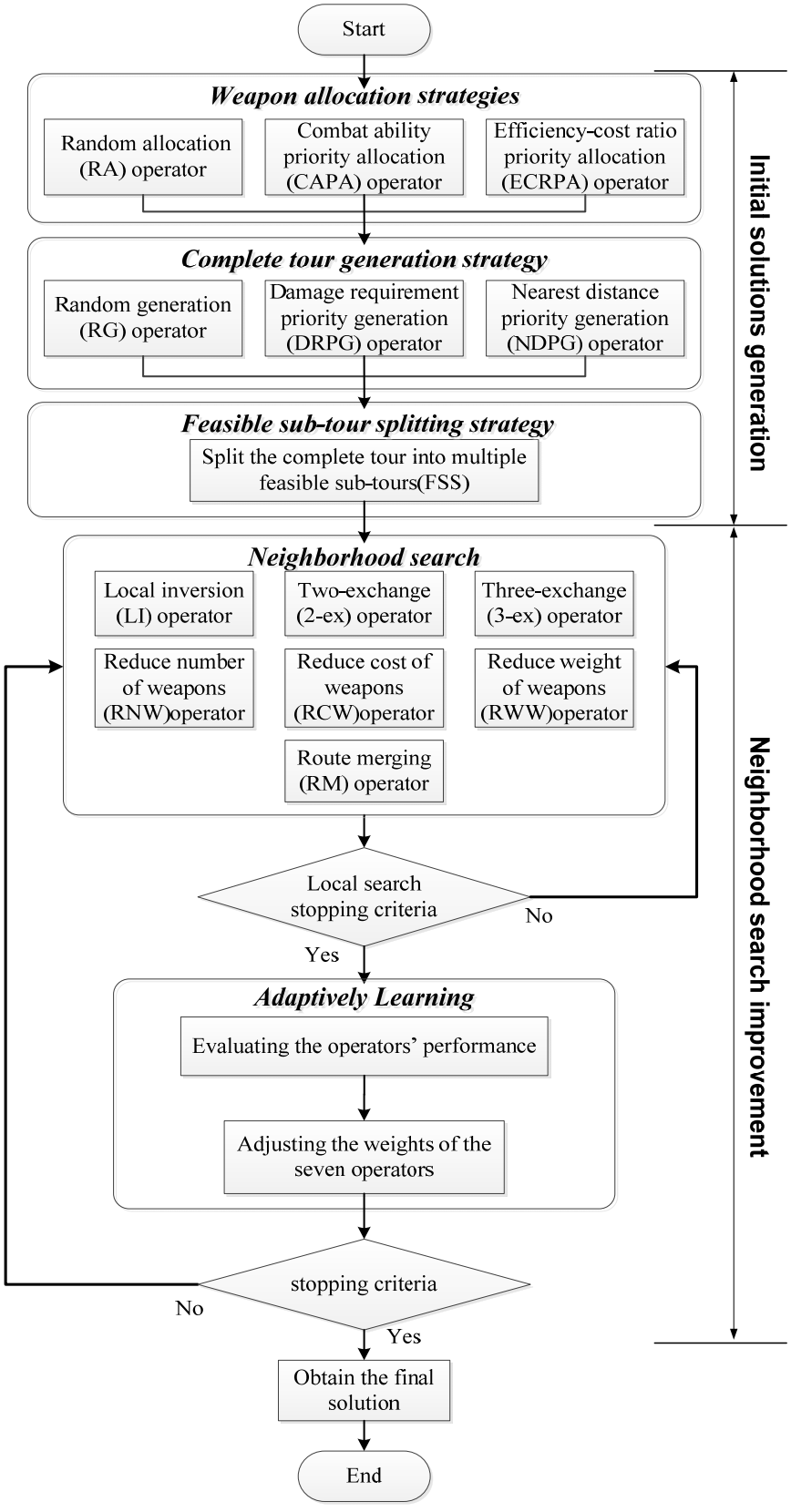

Fig. 1. Framework of the ALNS.

4.2 Definition of the solution

4.2.1 Encoding

The encoding scheme is important to the ALNS algorithm for our problem, and an effective encoding of the solution can facilitate the adjusting of decisions on both weapons and UCAV routing, which impacts the conduction of neighborhood operators. We utilize a $(w+1)\times n$ matrix to represent the visiting sequence of targets and the allocation of weapons. **Fig. 2** presents an encoding illustration of an instance with 9 targets and 3 weapons. The first row presents the targets and the attack sequence by UCAVs, which forms a complete tour by connecting the first and last cells to the depot respectively. In each column of the matrix, the cells from the second row to the (w+1)*th* row represent the weapon allocation strategy for the target in the first row. For instance, the first column of the matrix in **Fig. 2**, (6, 1, 0, 1) means that one Weapon 1 and one Weapon 3 should be delivered to Target 6.

Targets	6	2	5	7	1	4	3	9	8
Weapon 1	1	1	0	0	1	1	0	2	0
Weapon 2	0	1	1	0	0	1	1	0	1
Weapon 3	1	0	0	1	1	0	2	0	0

Fig. 2. Example of solution representation.

4.2.2 Decoding

The above encoding scheme only determines the visiting sequence of all targets and their weapon allocation strategies, and cannot present the weapon configuration and route of each UCAV. Thus, a splitting heuristic is developed to decoding the chromosome into feasible solution. The decoding process is to split the complete tour presented in the chromosome into sub-tours according to the capacities of UCAVs on load and hanging points. The splitting heuristic is detailed in Section 4.3.3.

4.3 Construction heuristic for initial solutions

The construction heuristic is proposed to generate a group of initial solutions as the starting points of the following neighborhood search. We construct the feasible solution based on an inverse process. We first determine which weapon and how many should be delivered to the target, and then construct a complete tour for all targets. At last, the UCAV starts from the depot to pick up the weapons along the complete tour until the payload capacities and hardpoints are sufficiently utilized, and then returns to the depot. The heuristic includes both greedy and random search ideas, so as to ensure the initial population contains some better feasible solutions as well as some diverse solutions.

4.3.1 Weapon allocation strategy

The weapon allocation strategy is used to determine which weapon and how many should be delivered to the target, so as to satisfy the targets' destroy requirement. Three operators are used in the weapon allocation process, which are presented as follows. When weapons are allocated to the target, the three operators are selected randomly with equal probabilities.

(1) Random allocation (RA)

The RA operator assigns weapons to each target one by one randomly until the destroy requirement of the target is satisfied. The total weight of weapons allocated to the target must be no more than the load capacity of the UCAV, and the total number of weapons must be no more than the number of hanging points in the UCAV, that is, the weapon allocation solution can not violate constraints (7) and (8). If one of or both the constraints are violated, the solution is abandoned and the weapons are reassigned again randomly until a feasible solution is obtained. A pseudo code of the RA operator is given in **Fig. 3**.

Algorithm 1: RA operator.

Input:

a_i : damage requirement of target i;

b_{im} : the combat ability of weapon m ($m \in W$) to against target i ;

q_m : the weight of a weapon of type m , and $m \in W$;

c : the load capacity of the UCAV;

g : the number of hanging points of the UCAV.

Output:

wat_{im} : the number of weapon m to against target i.

$CWH = -1$. *//check if constraints (7) and (8) are satisfied.*

while ($CWH < 0$) **do**

  $wat_{im} = 0$;

  $a_i' = a_i$;

  while($a_i' > 0$) **do**

    randomly select a weapon m from W;

    wat_{im}++;

    $a_i' = a_i' - b_{im}$;

  end

  if ($\sum_{m=1}^{M} q_m wat_{im} \le c$ and $\sum_{m=1}^{M} wat_{im} \le g$) **do**

    $CWH = 1$;

  end

end

Return $wat_{im}(m \in W)$.

Fig. 3. The pseudo code of the RA operator.

(2) Combat ability priority based allocation (CAPA)

The CAPA operator is to choose a set of weapons for target i according to their combat ability. In other words, give priority to the most powerful combat weapons assigned to the target. The pseudo code of CAPA is given in **Fig. 4.**

Algorithm 2: CAPA operator.

Input:

a_i : damage requirement of target i;

b_{im} : the combat ability of weapon m ($m \in W$) to against target i.

Output:

wat_{im} : the number of weapon m to against target i.

Set $wat_{im} = 0 (m \in W)$;

$m^* = \arg\max\{b_{im}, m \in W\}$;

$wat_{im^*} = \left\lfloor a_i / b_{im^*} \right\rfloor$;

$m' = \arg\min\left\{ f_m \middle| m \in W \text{ and } a_i - wat_{im^*} b_{im^*} - b_{im} \le 0 \right\}$;

$wat_{im'}$++;

Return $wat_{im}(m \in W)$.

Fig. 4. The pseudo code of CAPA.

(3) Efficiency-cost ratio priority based allocation (ECRPA)

Here, the efficiency-cost ratio of weapon m to target i is calculated as follows:

$$eff_{im} = \frac{b_{im}}{f_m}. \tag{13}$$

ECRPA operator uses the efficiency-cost ratio as the weapon allocation criteria and gives priority to the weapons with higher efficiency-cost ratio in the weapon allocation process. The pseudo code of ECRPA operator is given in **Fig. 5.**

Algorithm 3: ECRPA operator.

Input:

eff_{im}: the efficiency-cost ratio of weapon m to target i;

q_m: the weight of weapon m;

c: the load capacity of the UCAV;

g: the number of hanging points of the UCAV.

Output:

wat_{im}: the number of weapon m to against target i.

$CWH = -1$. *//check if constraints (7) and (8) are satisfied.*

Set $wat_{im} = 0 \, (m \in W)$;

while $(CWH < 0)$ **do**

  $m^* = \arg\max \{eff_{im}, m \in W\}$;

  $wat_{im^*} = \left\lceil a_i / b_{im^*} \right\rceil$;

  if $(\sum_{m=1}^{M} q_m wat_{im} \le c$ and $\sum_{m=1}^{M} wat_{im} \le g)$ **then**

    $CWH = 1$;

  end

  else

    $wat_{im^*} = 0$;

    $W = W / m^*$;

  end

end

Return $wat_{im} (m \in W)$.

Fig. 5. The pseudo code of ECRPA operator.

4.3.2 Complete tour generation

When the weapon allocation is completed, the next step is to optimize the routing of the UCAVs. A complete tour for visiting all targets are first constructed, which the UCAVs can fly along this tour to attack targets in sequence. The following three operators are utilized to construct

the complete tour.

1) Random generation (RG): The RG operator randomly arranges the visiting sequences of all targets to form a complete tour.
2) Damage requirement priority generation (DRPG): In real-world scenarios, targets are characterized by their priority or importance level[14], some targets may need be destroyed first. The DRPG utilizes the damage requirement of the target as the indicator for constructing the tour. That is, the targets are visited following the order from the one with highest damage requirement to the one with lowest damage requirement.
3) Nearest distance priority generation (NDPG): The NDPG operator is inspired by the nearest neighbor heuristic for the traveling salesman problem. However, in this operator, we randomly select a target as the first one and then visiting the other targets following the nearest neighbor principle.

4.3.3 Feasible sub-tour splitting (FSS)

Usually, one UCAV cannot complete the attacking tasks for all targets due to the limitation on load and hang point capacities. Thus, we have to split the complete tour for all targets into multiple sub-tours, and ensure the total weapons allocated to the targets in each sub-tour do not inviolate the constraints (7) and (8). A feasible sub-tour splitting heuristic is proposed to help each UCAV find a suitable sub-tour. The basic idea of this heuristic is to let the UCAV fly along the complete tour and pick up the weapons allocated to the targets in his flying route until its capacities are sufficiently consumed and then return to the depot. The next UCAV continues to visit the targets whose weapons are not picked up along the complete tour, and then one UCAV is followed by another one until all targets are visited. The calculating process of FSS operator is given in **Fig. 6**, while an illustrative example is shown in **Fig. 7** where the UCAV is assumed to have 6 hanging points and a load capacity of 900 kg.

Algorithm 4: FSS operator.

Input:

E : the encoding matrix, where the first row is obtained by complete tour generation operators and the rows from 2 to w+1 are obtained by the weapon allocation operators, and its element is noted as e_{lr} ;

q_m : the weight of a weapon of type m , and $m \in W$;

g : the number of hanging points for the UCAV;

c : the load capacity of the UCAV.

Output:

y_{kmi} : the number of weapon m on UCAV k used to attack target i .

Initialize k=1, $g' = g$, $c' = c$, $y_{kmi} = 0$;

for($l = 1, 2, \cdots, n$) **do**

  $g' = g' - \sum_{r=2}^{w+1} e_{lr}$;

  $c' = c' - \sum_{r=2}^{w+1} e_{lr} q_{r-1}$;

  if ($g' < 0$ **or** $c' < 0$) **then**

    l--;

    k++;

    $g' = g$;

    $c' = c$;

  end

  else

    $i = e_{l1}$;

    for ($m = 1, 2, \cdots, w$) **do**

      $y_{kmi} = e_{l,m+1}$;

    end

  end

end

Return y_{kmi} .

Fig. 6. The pseudo code of FSS operator.

Targets	6	2	5	7	1	4	3	9	8
Weapon 1 (75kg)	1	1	0	0	1	1	0	2	0
Weapon 2 (165kg)	0	1	1	0	0	1	1	0	1
Weapon 3 (240kg)	1	0	0	1	1	0	2	0	0
Sum of the numbers	2	2	1	1	2	2	3	2	1
Sum of the weights(kg)	315	240	165	240	315	240	645	150	165

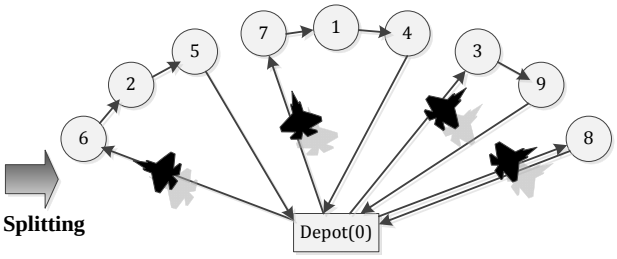

Fig. 7. Illustration of the results after splitting.

4.4 Neighborhood Operators

The above construction heuristic can propose a number of feasible solutions for the problem, which have to be further improved. In this section, we develop five neighborhood structures, which

utilize the initial feasible solutions as starting points to search the solution space and find a much better solution of the problem. Among the seven neighborhood operators, local inversion, two exchange and three exchange are random search operators, while the other four operators are greedy search operators.

4.4.1 Local inversion (LI)

The purpose of the LI operator is to change the attacking order of targets in the complete tour. In the LI operator, two targets are randomly selected and exchanged, and the visiting sequences of the targets between them are also exchanged. An example is depicted in **Fig. 8**.

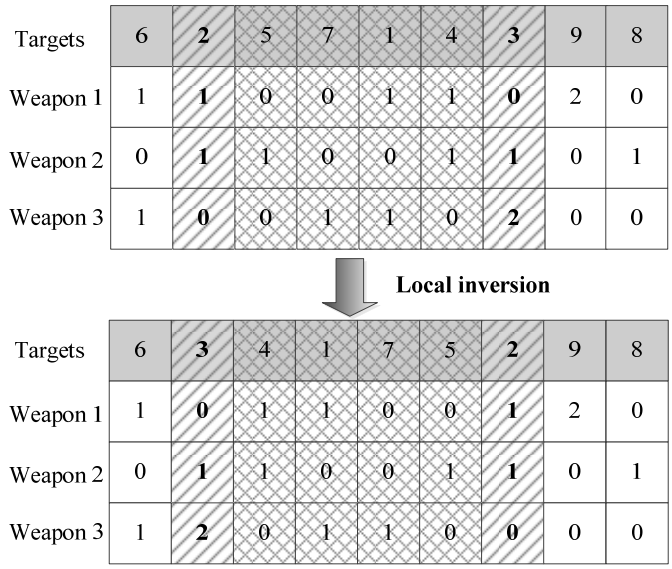

Fig. 8. Illustration of LI operator.

4.4.2 Two-exchange (2-ex)

The aim of this operator is to change the visit order of targets, which generates as many feasible solutions as possible on the basis of initial feasible solutions. Randomly generate two unequal integers $ex_1, ex_2 \in [0, n-1]$, which represent the position of the two genes of the chromosome, and then exchange the gene information of the two positions. Taking the problem in **Fig. 2** as an example, a 2-exchange process is illustrated in **Fig. 9**. After 2-exchange operation, the attack order of targets becomes: T6→T2→T9→T7→T1→T4→T3→T5→T8.

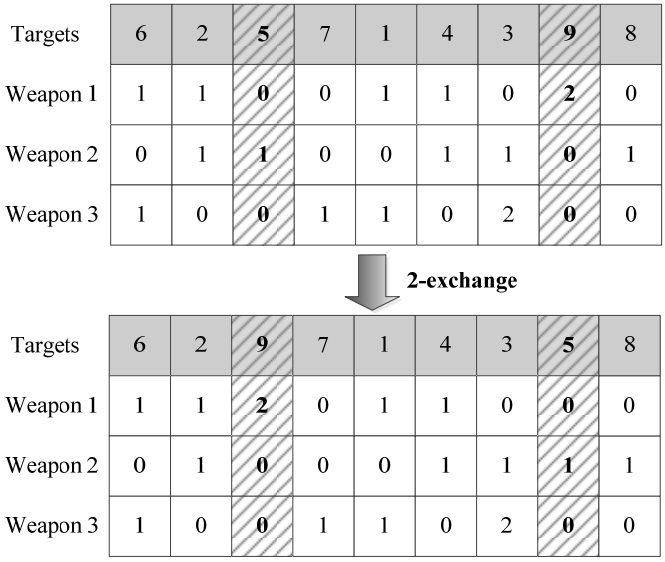

Fig. 9. Illustration of 2-ex operator.

4.4.3 Three-exchange (3-ex)

The 3-exchange operator is a variant of the 2-exchange operator. Randomly generate three unequal integers $ex_1, ex_2, ex_3 \in [0, n-1]$, which represent the position of the three genes of the chromosome, and then exchange the three positions of the genetic information. Taking the problem in **Fig. 2** as an example, a 3-exchange process is illustrated in **Fig. 10**. After 3-exchange operation, the attack order of targets becomes: T9→T2→T5→T6→T1→T4→T3→T7→T8. In contrast with a 2-exchange, in a 3-exchange, where three genes are mutated, there are several possibilities to construct a new initial solution from the 3-exchange process [29].

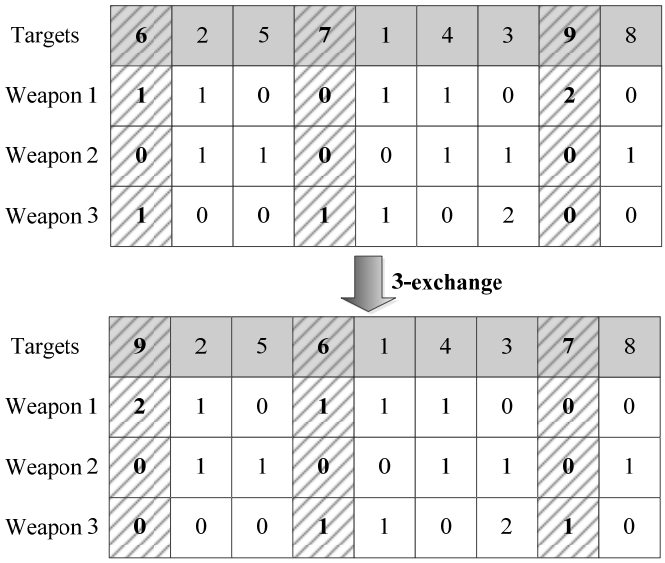

Fig. 10. One possible case of 3-ex operator.

4.4.4 Reducing number of weapons (RNW)

RNW operator selects the target with the largest number of weapons and checks if the target can be destroyed by less number of weapons with higher capability. It is worth noting that the reduction of weapons must meet the requirement of constraint(9). If the number of weapons allocated to the target is reduced, then the UCAV may visit more targets, which provides a potential to reduce the number of UCAV utilized and the overall flying distance. An illustrative example is given in **Fig. 11**.

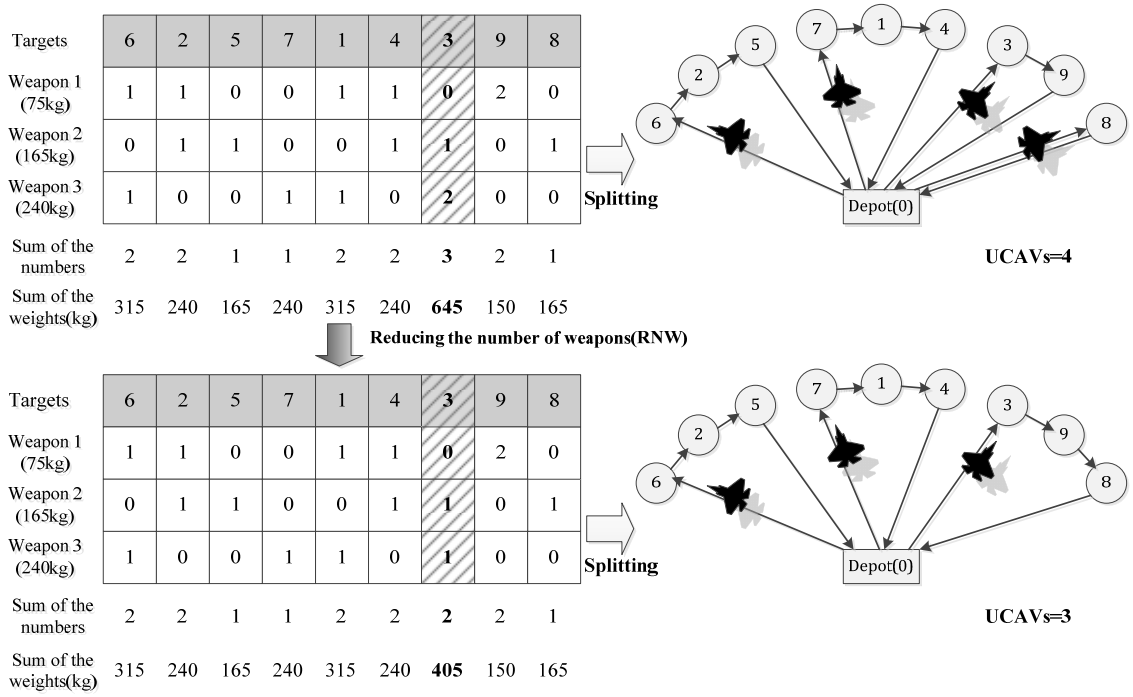

Fig. 11. Illustration of procedure RNW operator.

4.4.5 Reducing cost of weapons (RCW)

RCW operator selects the target whose total cost of allocated weapons is the highest one among all targets, and then checks if the weapons allocated to this target can be changed into cheaper ones. The requirements of constraints (7), (8) and (9) must be met when the weapon is exchanged. If the overall cost of the selected targets can be reduced, the better solution may be found. However, RCW operator cannot ensure that the solution be improved every time, and sometimes the reducing on cost of weapons may increase the number of weapons and the total weight of weapons, which may cause a solution with higher number of UCAVs and total flying

distance. Thus, RCW operator just provides a potential way to optimize the current solution. An example is given in **Fig. 12**.

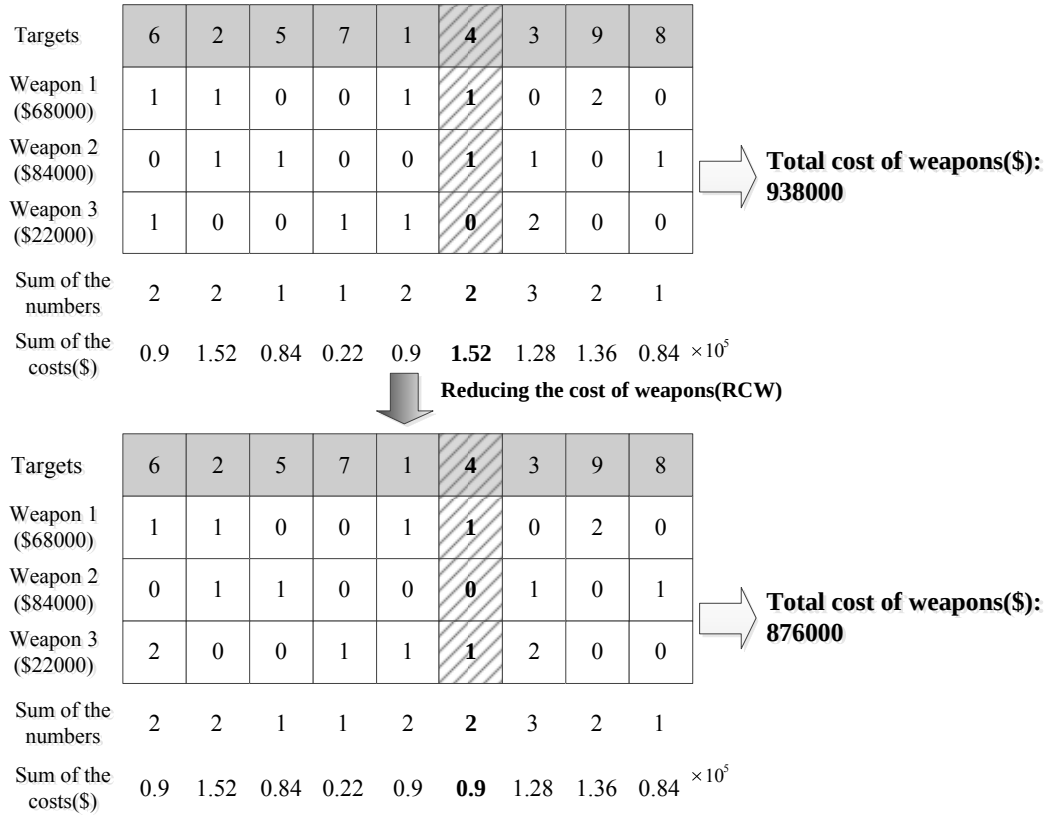

Fig. 12. Illustration of procedure RCW operator.

4.4.6 Reducing weight of weapons (RWW)

The RWW operator is to select the target whose total weight of all weapons is the highest, and change some of weapons with higher weight with some lighter ones. It should be pointed out that the replacement of weapons must meet the requirements of constraints (7)-(9). When the UCAV cannot visit more targets due to the constraints on load, the RWW operator provides a potential to let the UCAV visit more targets and then may reduce the overall number of UCAV utilized and the flying distance. An illustrative example is given in **Fig. 13**.

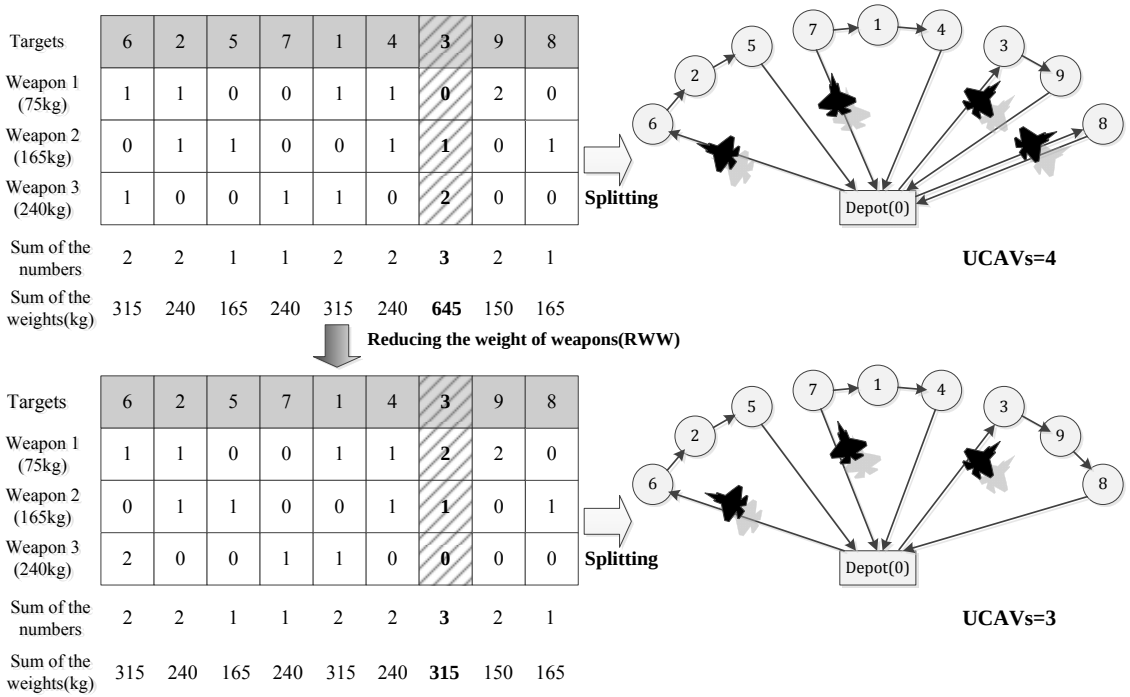

Fig. 13. Illustration of procedure RWW operator.

4.4.7 Route merging (RM)

The aim of RM operator is to find the best possible merging route. The main idea of this operator is to reduce the number of UCAVs to achieve the objective function optimization. For each route, there will always be some UCAV that is not loaded when it returns. The RM operator sorted the total number of UCAVs mounted weapons in descending order. Then try merging the two routes, and the UCAVs that fly on both routes are relatively less armed. An example is depicted in **Fig. 14**, in which the UCAV 1 and UCAV 4 are not fully loaded, after the calculation of RM operator the corresponding two routes can be merged.

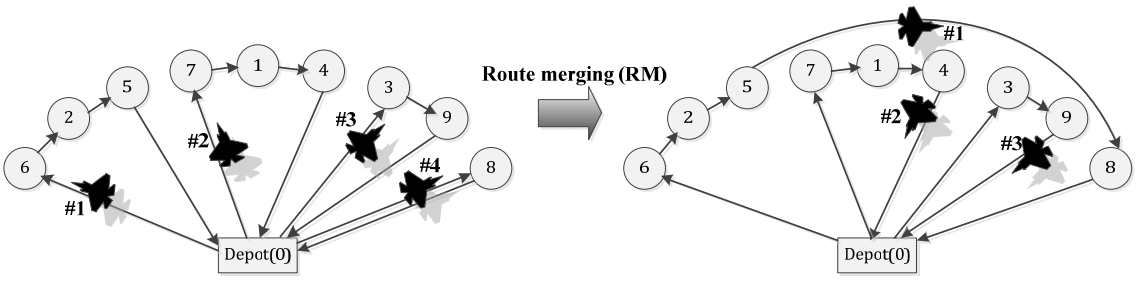

Fig. 14. Illustration of procedure RM operator.

4.5 Adaptively Learning Strategy

The feasible solution optimization heuristics are each associated a weight which dynamically changes throughout the algorithm. The general idea behind the learning selection mechanism, which was introduced by [23]. At each iteration, the selection of a removal heuristic and of an insertion heuristic is based on the roulette-wheel principle selection. Given h operators with weights ω_i, operator j is chosen with probability $\omega_j \Big/ \sum_{i=1}^{h} \omega_i$. The weight of each operator is updated every $\varphi(\varphi = 4)$ iterations. Initially, all operators have a weight of 1. The weight ω_{ij} of operator i at the j th sequence of φ iterations is computed as

$$\omega_{i,j+1} = \omega_{ij}(1-r) + r\frac{\sigma_{ij}}{\varepsilon_{ij}} \tag{14}$$

where ε_{ij} and σ_{ij} are the number of times operator i has been used in the j th sequence of φ iterations, and the score of operator i in the j th sequence of φ iterations, respectively, and $r(r \in [0,1])$ a is an arbitrary parameter equal to 0.1 in our implementation.

At the start of each learning, the scores of all operators are set to zero. The scores are increased by $\sigma_{ij}^1 = 30$ if a new best solution is found, by $\sigma_{ij}^2 = 10$ if the new solution is better than the current solution and by $\sigma_{ij}^3 = 6$ if the new solution is not improve upon the current solution but is accepted.

4.6 Acceptance and stopping criteria

We use the acceptance criteria introduced by [23] to define the acceptance criterion for a new solution. Let f^* be the value of the best current solution, and δ is a unique and positive parameter. Let R be a solution, R' a neighbor of R, and $f_{R'}$ the objective value of solution R'. Solution $R^{'}$ is accepted if $f_{R'} < f^* + \delta$, and f^* is updated if $f_{R'} < f^*$. We set $\delta = 0.1 f^*$ in our implementation. This ALNS algorithm will not stop unless the predetermined number of iterations θ is conducted.

5. Computational experiments

In this section, computational experiments are presented to evaluate the performance of ALNS algorithm. The proposed ALNS algorithm was coded with Visual C# 4.0 and the test environment is set up on a computer with Intel Core i7-4790 CPU, 3.60 GHz, 32GB RAM, running on Windows 7. One hundred and twenty instances are designed to test the algorithm, which cover the problem in different sizes including 10, 20, 50 and 100 targets respectively. For the small-size instances, the computational results of our ALNS algorithm are also compared with that obtained by CPLEX.

5.1 Experiment design and instance generation

In order to estimate the value of parameters related with the UCAV, we investigated typical UCAVs employed by armies in word wide, e.g, the MQ-9 Reaper which has six hardpoints and can carry four AGM-114 Hellfire airborne anti-tank missiles, and carry two GBU-12 Paveway II aerial laser-guided bombs [19]. We consider three types of payload capacity for UCAV, which are 600kg, 900kg and 1,200kg, and also three types of the number of hardpoints, which are 4, 6 and 8. It is assumed that three types of weapons can be armed in the UCAV. The details of the parameters are illustrated in Table 5, which can reflect the capabilities of UCAVs applied in different battlefields of modern wars.

Table 5 Basic data for UCAV and weapon.

		Index	Parameter values
UCAV		Payload capacity of the UCAV(kg)	600; 900; 1,200
		Number of hardpoints	4, 6, 8
		Type of weapons	W1, W2 and W3
		Cruise speed (km/h)	180
Weapons	W1	Weight (kg)	75
		Cost ($ thousand)	68
	W2	Weight (kg)	165
		Cost ($ thousand)	84
	W3	Weight (kg)	240
		Cost ($ thousand)	22

As mentioned earlier, our mathematical model does not consider UCAV endurance, and assumes that the all attacking missions can be completed within the UCAV endurance. Accordingly,

considering the UCAV endurance and combat radius, three scales of the mission region of the battlefield are considered, which are 300 by 300 km, 500 by 500 km and 800 by 800 km and discretized to grids with size of 300 × 300, 500 × 500 and 800 × 800, respectively. In practical war scenarios, the depot must keep a safe distance far away from the enemy targets, and we note this safe distance as φ. In this paper, we suppose that the safe distance φ is equal to the UCAV uniform cruise one hour flight distance, which is $\varphi = 180$ km. Without loss of generality, suppose that the depot position is at (1, 1), and all the targets are randomly generated in the region and are away from the depot at least the safe distance φ. Related with different scales of the planning region, four sizes for the number of targets are considered, which are 10, 20, 50 and 100. The experimental scales are set in Table 6.

Table 6 Summary of experiments.

Instance size	number of targets	Region (km^2)
Small	10 and 20	300 × 300
Medium	50	500 × 500
Large	100	800 × 800

The damage demand a_i of each target is randomly generated in the range $[0.85,1]$ following the uniform distribution. The combat ability of weapon *m* on target *i*, b_{im} is randomly generated in the range [0.1, 1] following the uniform distribution. In the considered mission planning situation, each target can only be visited once, and thus for any target *i*, the UCAV must be able to destroy it when carrying the weapons with highest ability to this target, that is, $\max\{b_{im}g, m \in W\} \geq a_i$ for $i \in N$. If this condition cannot be satisfied, the randomly generated value of b_{im} is unreasonable. We check all b_{im} and find the unreasonable ones to regenerate randomly until this condition holds. As mentioned earlier, the objective function coefficients reflect the UCAV combat cost. Therefore, the setting of the objective function coefficients in the experiment varies slightly according to the UCAV payload capacity. The objective function coefficients setting is shown in Table 7.

Table 7 The objective function coefficients setting in experiment.

Armament	objective function coefficients
4 hardpoints & 600 kg payload capacity	P_1=1000； P_2=1； P_3=1000
6 hardpoints & 900 kg payload capacity	P_1=1200； P_2=1； P_3=1200
8 hardpoints & 1200 kg payload capacity	P_1=1440； P_2=1； P_3=1440

5.2 Computational results analysis

In the experiment, each instance is solved ten times, and the average value of the ten results is taken as the final result of the tested instance. The computation times are measured in seconds.

5.3.1 Small-size instances

Thirty small-size instances with 10 targets and thirty with 20 targets are randomly generated based on the above experiment design, which are solved both by the proposed ALNS and CPLEX. Table 8 presents the computational results for the instances with 10 targets. Columns 4 and 7 of Table 8 and Table 9 display the computational time of two approaches, respectively. And the average computational time of CPLEX in 10 targets is 0.59 seconds, which is shorter than the average computational time of ALNS. But in 20 targets, the average computational time of CPLEX is 4002.20 seconds, which is much larger than the average computational time of ALNS. Columns 8 in Table 8 and Table 9 show the improvement by the ALNS algorithm on the initial solutions, and the average solution improvement is about 48.21% and 41.10%, respectively. Columns 9 of Table 8 and Table 9 display the gap between the two approaches, in this paper, where gap means the relative gap between the objective value obtained by CPLEX and the ALNS, which is equal to (the objective obtained by ALNS – the objective obtained by CPLEX)/the objective obtained by ALNS. The average gaps between the two approaches are only 1.22% and 1.98%, respectively on the instances with 10 targets and 20 targets.

Table 8 Performance characteristics of CPLEX and ALNS tested using 10 targets instances.

Armament	No.	CPLEX		ALNS [a]				Gap (%)
		Obj.Val.	Time (s)	Obj.Val.	Initial solution[b]	Time (s)	Improve (%)	
4 hardpoints & 600 kg payload capacity	1	2424597.98	0.12	2457572.51	3888657.08	14.13	36.80	1.36
	2	2380102.02	1.92	2394144.62	3873176.37	14.84	38.19	0.59
	3	2060957.84	0.20	2091047.82	3608727.74	13.65	42.06	1.46
	4	2352257.91	0.44	2393422.42	3724640.19	14.83	35.74	1.75
	5	2114529.56	0.65	2148150.58	3777908.04	14.32	43.14	1.59
	6	2243336.12	0.53	2274518.49	4068900.50	14.45	44.10	1.39
	7	2155791.79	0.40	2193733.73	3751946.02	15.65	41.53	1.76
	8	2141811.63	0.34	2159802.85	3806186.38	15.54	43.26	0.84
	9	2041426.68	0.27	2054695.95	3776444.99	14.93	45.59	0.65
	10	2360302.65	0.55	2378476.98	3923007.34	15.92	39.37	0.77
6 hardpoints & 900 kg payload capacity	11	2409043.58	1.79	2415788.90	4878869.49	15.18	50.48	0.28
	12	2451521.48	0.22	2455198.76	4672567.67	15.70	47.46	0.15
	13	2150123.74	0.20	2177430.31	4289204.62	14.47	49.23	1.27
	14	2223192.75	1.43	2232974.80	4544061.43	15.04	50.86	0.44
	15	2396833.35	2.06	2444290.65	4402794.64	14.96	44.48	1.98
	16	2371673.97	0.21	2383769.51	4471151.19	15.44	46.69	0.51
	17	2367222.82	0.17	2408412.50	4824156.59	14.04	50.08	1.74
	18	2293715.92	0.12	2390969.47	4761927.64	13.68	49.79	4.24
	19	2164985.48	0.14	2182521.86	4428984.09	13.78	50.72	0.81
	20	2198422.64	0.64	2212492.54	4654435.57	13.24	52.46	0.64
8 hardpoints & 1200 kg payload capacity	21	2232777.94	0.62	2273637.78	4942157.64	13.45	54.00	1.83
	22	2170507.96	0.31	2181143.45	4903109.18	13.52	55.52	0.49
	23	2133621.41	0.35	2186961.95	5173273.95	13.19	57.73	2.5
	24	2322472.15	0.44	2372637.55	4834518.65	13.49	50.92	2.16
	25	2169989.42	0.23	2185396.34	5112405.26	13.68	57.25	0.71
	26	2297950.78	0.52	2323687.83	5536982.21	13.31	58.03	1.12
	27	2210491.30	1.75	2242543.42	4830348.62	13.35	53.57	1.45
	28	2466338.72	0.34	2472997.83	5262662.30	13.33	53.01	0.27
	29	2318171.53	0.49	2348307.76	4953336.47	13.29	52.59	1.3
	30	2451781.82	0.28	2466737.69	5111780.87	14.08	51.74	0.61
Avg.		—	**0.59**	—	—	**14.28**	**48.21**	**1.22**

[a] the iteration number of the algorithm is 2000.

[b] the minimum value of the objective function in the initial population.

Table 9 Performance characteristics of CPLEX and ALNS tested using 20 targets instances.

Armament	No.	CPLEX		ALNS [a]				Gap (%)
		Obj.Val.	Time (s)	Obj.Val.	Initial solution[b]	Time (s)	Improve (%)	
4 hardpoints & 600 kg payload capacity	31	4899610.22	3343.28	5063697.54	7492363.51	172.83	32.42	3.35
	32	5034765.67[c]	7200.00	5059977.76	7952110.85	181.88	36.37	0.50
	33	5520572.73	2101.08	5581197.55	8262242.54	174.34	32.45	1.10
	34	4782554.74[c]	7200.00	4814935.01	7740820.10	159.98	37.80	0.68
	35	5126144.84	3342.66	5221043.14	8521157.63	181.46	38.73	1.85
	36	4983406.04	4139.49	5062217.82	7714785.85	177.81	34.38	1.58
	37	5426823.76	2861.28	5493187.76	8244558.53	202.44	33.37	1.22
	38	4768546.48	3295.02	4816711.65	7385545.18	159.35	34.78	1.01
	39	5082171.68	2296.35	5243736.55	8077279.46	214.01	35.08	3.18
	40	4893166.56	2864.81	5079180.63	8007398.00	175.60	36.57	3.80
6 hardpoints & 900 kg payload capacity	41	6072122.90[c]	7200.00	6097756.98	10133896.72	169.87	39.83	0.42
	42	5169130.61	4139.63	5251331.52	9273408.90	331.76	43.37	1.59
	43	6223474.21	4452.23	6413740.53	11126924.68	207.58	42.36	3.06
	44	5468838.52	2175.06	5528572.84	9717544.13	196.23	43.11	1.09
	45	5439860.50	3658.69	5638837.13	9352402.83	186.67	39.71	3.66
	46	5274467.03	3493.83	5403594.43	9749993.90	161.46	44.58	2.45
	47	5374977.48[c]	7200.00	5422746.29	9191694.26	155.84	41.00	0.89
	48	6192449.97	2218.25	6307882.98	10540562.03	175.84	40.16	1.86
	49	6257508.70	4313.64	6400498.98	10622605.75	165.88	39.75	2.29
	50	6202262.45[c]	7200.00	6263963.85	10172959.25	174.07	38.43	0.99
8 hardpoints & 1200 kg payload capacity	51	6308856.25	3009.60	6390566.83	12201417.94	160.00	47.62	1.30
	52	5480492.21	2080.21	5641612.05	11202576.31	155.18	49.64	2.94
	53	6070380.20[c]	7200.00	6117562.61	12239885.06	164.25	50.02	0.78
	54	5522360.80	2815.72	5696050.07	11141702.42	157.08	48.88	3.15
	55	5533082.06	3096.89	5709138.24	9902122.84	177.69	42.34	3.18
	56	6512213.86	2760.13	6578654.03	11533613.36	152.47	42.96	1.02
	57	5833734.84	4166.18	6071186.17	11812193.19	163.27	48.60	4.07
	58	6443927.95	3942.82	6611908.41	11734979.34	159.43	43.66	2.61
	59	5933455.78	3558.42	6052242.16	11705717.35	168.15	48.30	2.00
	60	6010477.97	2740.77	6121263.88	11505725.93	162.27	46.80	1.84
Avg.		—	**4002.20**	—	—	**178.16**	**41.10**	**1.98**

[a] the iteration number of the algorithm is 10000.

[b] the minimum value of the objective function in the initial population.

[c] optimality was not verified within a time-limit of 7200 s.

From the overall results, it can be seen that the proposed ALNS solved all the small-size instances in reasonable time and obtained near optimal solutions which are quite close to that obtained by CPLEX. However, the computational time of CPLEX increases greatly as the number of targets increases, and for some instances with 20 target, e.g. Instance 32, 34, 41 etc, CPLEX

cannot obtain the optimal solution in 7200 seconds.

5.3.2 Medium size and large size instances

As indicated by the results for instances with 10 and 20 targets, the computational time of CPLEX increases greatly with the number of targets. And the further computation of the instances with 50 targets shows that CPLEX cannot obtain satisfactory solutions in several hours, which cannot satisfy the practical military requirement in current wars. Thus, we only present the computational results for the medium and large-size instances, which are obtained by the ALNS algorithm. Table 10 and Table 11 report the results obtained by ALNS for solving instances with 50 targets and 100 targets, respectively.

The computational results in Table 10 for the medium instances show that, compared with the initial solutions obtained by the constructive heuristic, the final solutions obtained the ALNS algorithm are improved by an average of 42.04%, which is similar to the performance on the small-size instances. The average computational time is 901.12 seconds, which is acceptable.

The computational results in Table 11 for the large instances show that, compared with the initial solutions obtained by the constructive heuristic, the final solutions obtained the ALNS algorithm are improved by an average of 42.13%, which is similar to the performance on the small-size and medium instances. The average computational time for large instances is 3588.58 seconds, about one hour, which is also acceptable.

Form the computational results for all instances with different sizes, it can be seen that the ALNS algorithm shows good performance on both solution quality and computational time, which indicates that the neighborhood structures designed for the joint weapon configuration, allocation and route planning problem are effective.

Table 10 Performance characteristics of ALNS tested using 50 targets instances.

Armament	No.	ALNS [a]			
		Obj.Val.	Initial solution[b]	Time (s)	Improve (%)
4 hardpoints & 600 kg payload capacity	61	19639702.79	26418858.49	894.16	25.66
	62	20226587.53	28091423.00	905.93	28.00
	63	19075602.57	29497875.83	827.38	35.33
	64	19562878.93	28270698.47	885.11	30.80
	65	21225258.16	33479351.74	968.09	36.60
	66	21245526.49	30937990.54	903.73	31.33
	67	19025215.37	30255708.29	951.10	37.12
	68	18279376.77	25863988.21	934.00	29.32
	69	21500859.25	32517817.92	921.50	33.88
	70	19837561.10	27110162.60	843.02	26.83
6 hardpoints & 900 kg payload capacity	71	23757260.36	38863692.78	864.34	38.87
	72	23626293.40	45297246.03	839.71	47.84
	73	25086301.95	43649491.24	897.37	42.53
	74	23803380.80	40704101.47	957.38	41.52
	75	24953042.70	44968659.67	960.44	44.51
	76	20252252.48	33375639.30	951.77	39.32
	77	21484026.19	36297375.83	947.17	40.81
	78	20961277.39	34038993.24	962.00	38.42
	79	21290151.48	38375687.63	965.33	44.52
	80	23262131.65	43412205.51	871.07	46.42
8 hardpoints & 1200 kg payload capacity	81	26411759.24	51683908.96	916.99	48.90
	82	25927669.48	49991044.91	839.00	48.14
	83	28654411.71	64790006.23	925.85	55.77
	84	25173709.93	53936381.18	834.87	53.33
	85	24846209.02	51853298.48	943.74	52.08
	86	26687074.82	54588557.26	941.93	51.11
	87	26893376.17	59066981.34	843.04	54.47
	88	24835318.46	47791483.44	839.85	48.03
	89	29422705.85	62766002.15	849.10	53.12
	90	27632781.35	63730779.37	848.59	56.64
Avg.		—	—	**901.12**	**42.04**

[a] the iteration number of the algorithm is 15000.

[b] the minimum value of the objective function in the initial population.

Table 11 Performance characteristics of ALNS tested using 100 targets instances.

Armament	No.	ALNS [a]			
		Obj.Val.	Initial solution[b]	Time (s)	Improve (%)
4 hardpoints & 600 kg payload capacity	91	56713612.00	77833338.02	3658.66	27.13
	92	58819736.63	88335510.39	3939.26	33.41
	93	61499268.73	90584129.84	3620.54	32.11
	94	59363550.94	81817562.91	3480.32	27.44
	95	60015455.25	87037253.27	3427.54	31.05
	96	59031436.97	79440954.08	3691.88	25.69
	97	54676897.55	82662092.88	3331.97	33.85
	98	57308366.75	93759087.00	3532.63	38.88
	99	58273504.04	93494742.44	3337.27	37.67
	100	60034343.70	82882886.05	3913.52	27.57
6 hardpoints & 900 kg payload capacity	101	74474010.78	120333006.01	3994.62	38.11
	102	70858242.62	129567870.06	3787.14	45.31
	103	63311816.71	117370935.19	3334.90	46.06
	104	63549955.83	105734877.09	3598.18	39.90
	105	80182413.87	134483508.76	3715.91	40.38
	106	60589994.73	101644904.13	3806.19	40.39
	107	74023035.54	123394701.99	3506.24	40.01
	108	72685374.15	139422975.33	3571.98	47.87
	109	68584173.73	131933825.44	3564.44	48.02
	110	70937380.62	127674074.42	3954.26	44.44
8 hardpoints & 1200 kg payload capacity	111	71011557.80	138470904.86	3455.29	48.72
	112	79348300.75	188831368.78	3335.35	57.98
	113	87032222.17	174639693.52	3548.56	50.16
	114	78536295.79	155542205.60	3307.42	49.51
	115	79126976.77	169402740.94	3984.86	53.29
	116	81119272.13	162194167.94	3545.99	49.99
	117	80819186.75	191885864.20	3407.36	57.88
	118	89209526.06	177191089.51	3516.22	49.65
	119	78658940.98	166693102.07	3353.06	52.81
	120	84250485.60	163825165.08	3435.85	48.57
Avg.		—	—	**3588.58**	**42.13**

[a] the iteration number of the algorithm is 20000.

[b] the minimum value of the objective function in the initial population.

6. Conclusions and future work

In this paper, we introduced a joint weapon configuration, allocation and route planning

problem for a fleet of unmanned combat air vehicles, which is a new extension of the UAV/vehicle routing problem. The problem is formulated into an integer linear programming model, and an adaptive large neighborhood search heuristic is developed to solve the model. A problem specified constructive heuristic is designed to obtain initial feasible solutions of the problem, and seven neighborhood structures are defined and employed to improve the quality of the solutions. Computational results based on random instances with different sizes show that CPLEX can only solve the small-size instances, while the proposed ALNS algorithm can obtain good solutions for all instances and the average computational time for large instances with 100 targets is about one hour. The overall computational results indicate that the proposed ALNS algorithm is efficient and applicable in practical military mission planning.

The joint Weapon configuration, allocation and route planning problem is a new topic in military UCAV mission planning area, and there are many extensions required to be studied in future research, such as the situations with multiple deports, with time window and with random losses of UCAVs. Another valuable research on this topic is to develop solution algorithms based on other metaheurisitcs, e.g. Tabu search, greedy randomized adaptive search, and evolutionary algorithm, and then compare their performances.

Acknowledgment

This work was supported by Natural Science Foundation of China with Grant no. 71471174 and 71771215.

References

[1] V.K. Shetty, M. Sudit, R. Nagi, Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles, Computers & Operations Research, 35 (2008) 1813–1828.

[2] D.L. Grecu, P.G. Gonsalves, Agent-Based Simulation Environment For UCAV Mission Planning And Execution, American Institute of Aeronautics and Astronautics, (2000) 1-11.

[3] S. Karaman, E. Frazzoli, Linear temporal logic vehicle routing with applications to multi-UAV mission planning, Int J Robust Nonlin, 21 (2011) 1372-1395.

[4] L.G. B, UAV Swarm Mission Planning Development Using Evolutionary Algorithms and Parallel Simulation-Part II SCI-195, Rta.nato.int, (2008).

[5] A.J. Pohl, G.B. Lamont, Multi-objective UAV mission planning using evolutionary computation, in: Conference on

Winter Simulation, (2008), pp. 1268-1279.
[6] J. Tian, L.C. Shen, Y.X. Zheng, Genetic algorithm based approach for multi-UAV cooperative reconnaissance mission planning problem, Lect Notes Artif Int, 4203 (2006) 101-110.
[7] L. Evers, A.I. Barros, H. Monsuur, A. Wagelmans, UAV Mission Planning: From Robust to Agile, 56 (2015) 1-17.
[8] A.N. Kopeikin, S.S. Ponda, L.B. Johnson, J.P. How, Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams, All Publications, 1 (2013).
[9] X. Wang, J. Wu, H. Huang, S. Deng, Multiple tasks scheduling algorithm for UAV attacking in uncertain environment, in: Systems and Informatics (ICSAI), 2012 International Conference on, IEEE, 2012, pp. 735-739.
[10] E. Sonmezocak, S. Kurt, Optimum route planning and scheduling for unmanned aerial vehicles, in, Monterey, California. Naval Postgraduate School, 2008.
[11] Xu C, Duan H, L. F, Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle path planning, Aerosp Sci Technol, 14 (2010) 535-541.
[12] E. Edison, T. Shima, Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms, Computers & Operations Research, 38 (2011) 340-356.
[13] Z. Liu, Q. Luo, W. Ding, S. Qiao, J. Shi, W. Zhang, A Task Assignment Algorithm for Multiple Aerial Vehicles to Attack Targets With Dynamic Values, IEEE Transactions on Intelligent Transportation Systems, 14 (2013).
[14] V.K. Shetty, M. Sudit, R. Nagi, Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles, Computers & Operations Research, 35 (2008) 1813-1828.
[15] F. Mufalli, R. Batta, R. Nagi, Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans, Computers & Operations Research, 39 (2012) 2787-2799.
[16] L. Evers, T. Dollevoet, A.I. Barros, H. Monsuur, Robust UAV mission planning, Annals of Operations Research, 222 (2014) 293-315.
[17] İ. Sarıçiçek, Y. Akkuş, Unmanned Aerial Vehicle hub-location and routing for monitoring geographic borders, Appl Math Model, 39 (2015) 3939-3953.
[18] E. Yakıcı, Solving location and routing problem for UAVs, Computers & Industrial Engineering, 102 (2016) 294-301.
[19] S.M. V, The Reaper Harvest, Air Force Magazine, 94 (2011) 36-39.
[20] V. BEŇO, F. ADAMČÍK Jr., Unmanned combat air vehicle: MQ-9 Reaper, Scientific Research & Education in the Air Force-AFASES, (2014).
[21] S. Ropke, D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transportation science, 40 (2006) 455-472.
[22] G. Laporte, R. Musmanno, F. Vocaturo, An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands, Transportation Science, 44 (2010) 125-135.
[23] H. Lei, G. Laporte, B. Guo, The capacitated vehicle routing problem with stochastic demands and time windows, Computers & Operations Research, 38 (2011) 1775-1783.
[24] I. Dayarian, T.G. Crainic, M. Gendreau, W. Rei, An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem, Transportation Research Part E: Logistics and Transportation Review, 95 (2016) 95-123.
[25] H. Lei, G. Laporte, Y. Liu, T. Zhang, Dynamic design of sales territories, Computers & Operations Research, 56 (2015) 84-92.
[26] X. Wang, L. Tang, A population-based variable neighborhood search for the single machine total weighted tardiness problem, Computers & Operations Research, 36 (2009) 2105-2110.
[27] D. Lei, Population-based neighborhood search for job shop scheduling with interval processing time, Computers & Industrial Engineering, 61 (2011) 1200-1208.
[28] S. Liu, A hybrid population heuristic for the heterogeneous vehicle routing problems, Transportation Research Part E: Logistics and Transportation Review, 54 (2013) 67-78.

[29] M.W.P. Savelsbergh, An efficient implementation of local search algorithms for constrained routing problems, Eur J Oper Res, 47 (1990) 75-85.