Journal of Systems Engineering and Electronics
Vol. 32, No. 3, June 2021, pp.658 – 667

# A branch and price algorithm for the robust WSOS scheduling problem

LI Ruiyang, HE Ming*, HE Hongyue, WANG Zhixue, and YANG Cheng
Institute of Command and Control Engineering, Army Engineering University of PLA, Nanjing 210007, China

**Abstract:** To analyze and optimize the weapon system of systems (WSOS) scheduling process, a new method based on robust capabilities for WSOS scheduling optimization is proposed. First, we present an activity network to represent the military mission. The member systems need to be reasonably assigned to perform different activities in the mission. Then we express the problem as a set partitioning formulation with novel columns (activity flows). A heuristic branch-and-price algorithm is designed based on the model of the WSOS scheduling problem (WSOSSP). The algorithm uses the shortest resource-constrained path planning to generate robust activity flows that meet the capability requirements. Finally, we discuss this method in several test cases. The results show that the solution can reduce the makespan of the mission remarkably.

**Keywords:** weapon system of systems (WSOS), robust optimization, scheduling decision, branch-and-price, column generation.

**DOI:** 10.23919/JSEE.2021.000056

## 1. Introduction

Weapon system of systems (WSOS) is a higher-level military complex system. It is integrated by various weapon systems that are functionally interconnected and complement [1]. To meet diverse military requirements and maximize overall operational effectiveness, all countries in the world are vigorously developing WSOS under the guidance of national security and military strategy. At present, most of the research is focused on the construction of the WSOS, exploring the optimal construction program to meet the military requirements. However, the research on scheduling in the scheduling process of the WSOS is still scarce. Appropriate WSOS scheduling scheme not only makes rational use of human and material resources, but also preemptively takes the initiative on the battlefield. Thus, at the beginning of the military mission, using scientific models and methods to analyze and optimize the WSOS scheduling process is particularly important [2].

The WSOS scheduling problem (WSOSSP) is defined as: under the premise of meeting the capability requirements of various activities, the limited systems and resources provided by the WSOS are reasonably allocated, the execution strategy of various operational activities is determined, and then the WSOS scheduling scheme in the mission is obtained. This problem is an extension and application of the multi-mode resource-constrained project scheduling problem (MRCPSP) in the military field. The MRCPSP is a classical non-deterministic polynomial hard (NP-hard) problem [3].

Most of the researches on the MRCPSP use the heuristic algorithm or meta-heuristic algorithm. For example, Chen et al. [4] proposed a two-stage genetic algorithm, which considered the activity mode and scheduling scheme. Najid et al. [5] divided the resource constraints in the MRCPSP into renewable and non-renewable, and proposed a new tabu search algorithm. Wang et al. [6] solved the MRCPSP by the estimation of distribution algorithm (EDA). Ayodele et al. [7] combined the EDA with the genetic algorithm to synchronize the different search spaces presented by two subproblems of mode selection and scheduling optimization. According to the non-preemptive MRCPSP, Adamu et al. [8] presented a novel method based on machine learning to obtain the feasible activity priority list of the project’s tasks, and calculated the primary solutions by metaheuristic algorithms. Afshar et al. [9] introduced a new local search method into the genetic algorithm framework to solve the MRCPSP. Roson et al. [10] divided resource constraints into different levels according to their importance, and proposed a hybrid optimization algorithm to solve the problem. In addition to the above, Ratajczak [11], Zsolt [12], Vahdani [13], etc. also solved similar problems

Manuscript received March 17, 2020.
*Corresponding author.
This work was supported by the National Key R&D Program of China (2018YFC0806900), the China Postdoctoral Science Foundation Funded Project (2018M633757), the Primary Research & Development Plan of Jiangsu Province (BE2017616; BE20187540; BE2019762; BE2020729), and the Jiangsu Province Postdoctoral Science Foundation Funded Project (2019K185).

based on the heuristic or meta-heuristic algorithms. These methods can find the approximate optimal solution satisfying the constraint in the polynomial time, but the stability of these algorithms is relatively poor in general.

There are also some researches on solving the MRCPSP with exact algorithms. The branch and bound method proposed by Sprecher et al. [14] is a classical exact algorithm. The algorithm used depth-first search and dominance rules to improve the algorithm efficiency. Mixed integer linear programming proposed by Kyriakidis et al. [15] is also a common method. In addition, the deterministic Boolean satisfiability theory proposed by Jose et al. [16] and Schnell et al. [17], the satisfiability model theory proposed by Bofill et al. [18] and the hybrid method combining branch and bound with a single non-renewable resource constraint proposed by Altintas et al. [19] are all used exact algorithms to solve the MRCPSP.

However, all the methods mentioned above are deterministic optimization methods. Considering the complex and changing characteristics of military missions, uncertainty such as weather changes, enemy strikes, and electromagnetic interference may affect the capabilities of the weapon system, which in turn affects the WSOS scheduling process. It has been proved that the uncertainty may lead to poor or even infeasible solutions. Therefore, we must find an optimization method that can effectively deal with the uncertainty, and ensure that the obtained scheme can be applied reliably. Robust optimization is a suitable method [20]. When the relevant parameters are changed by uncertainty, the scheme can maintain high stability against these changes. Leus et al. [21] proposed that the activities in the project insert a fixed buffer time to reduce the impact of uncertainty, but the value of the buffer time is often based on experience, so it is very subjective. Kaveh et al. [22], Birjandi et al. [23] represented parameters in the scheduling model as fuzzy variables, and Angela [24] used entropy functions to represent uncertain variables in the model and analyze the robustness of the solution. Shan et al. [25] designed a robust counterpart model for solving uncertain variables in the location and path planning problems.

Based on the previous researches, we consider the scenario where the capabilities provided by the weapon system decline during the WSOS scheduling process. Firstly we design a novel robust optimization model based on dynamic transfer equation, in which each column (variable) represents an activity flow on a timeline, and a feasible solution of the WOSSP is composed of multiple such activity flows. Secondly, the model we propose has a tighter lower bound, and each column has a distinct physical meaning. Based on this, we propose a more efficient heuristic branch-and-price algorithm that can help to make a high-quality solution in a certain time.

For the remainder, Section 2 describes the WSOSSP. Section 3 provides a robust mathematical model for the uncertain WSOSSP. Section 4 introduces a heuristic branch-and-price algorithm to solve the model. Section 5 reports computational experiments. Section 6 summarizes the conclusion.

## 2. Problem description

As was already mentioned, the WSOSSP can be described as follows: when a military mission is determined, the WSOS begins to perform a series of operational activities. Each of these activities can be completed when it meets specified operational capability requirements, such as communication, reconnaissance, transportation, etc. The system in the WSOS can provide a subset of capabilities. Therefore, different strategies composed of different systems that meet capability requirements can be selected to perform operational activities. In addition, there is a priority relationship between operational activities, and an activity can only begin after all its immediate preceding activities have been completed. Under the capability requirements, resource and time priority constraints, each member system in the WSOS is reasonably scheduled, so that the mission is completed in the shortest time.

Fig. 1 shows an example of a feasible strategy for analyzing activities: there are four systems in the WSOS that can provide certain operational capabilities. For example, Sys1 has C1 capability of 2 and C2 capability of 2. The strategy of (Sys1, Sys3) is able to complete activity V1 because it provides C1 capability of 4, which exceeds V1's minimum capability requirement of 3. Similarly, activity V2 has two feasible strategies (Sys1, Sys2) and (Sys2, Sys4), and activity V3 has two feasible strategies (Sys2, Sys3) and (Sys3, Sys4).

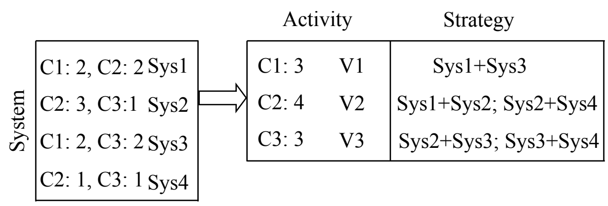


**Fig. 1 Example to analyze the activity feasible strategy**

Fig. 2 shows a simple example of the scheduling process of the WSOS. We use an activity-on-node (AON) to represent the military mission. The activities in the mission are represented by a set of nodes $\{0, 1, 2, \cdots i, \cdots, 11\}$, where nodes 0 and 11 represent the start and end activities of the mission respectively. The priority order of the activities is represented by the directed edges between the connected nodes. The commander rationally allocates the systems in the WSOS to meet the capability requirements of different activities, and ultimately minimizes the makespan of the mission. In fact, each system can be only assigned to one activity at a time. Such as Sys5 in Fig. 2, which can be allocated to activity 7 when activity 4 is completed. In addition, the resources consumed to complete all activities cannot exceed the resource limits that the WSOS can provide.

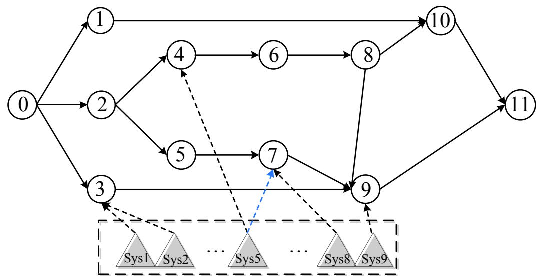


**Fig. 2 Illustrative example of the WSOS scheduling process**

It assumes that all activities in the mission require only one capability. Fig. 3 shows a scheduling scheme for the WSOS. The data is shown in Table 1. In this scheme, the capability requirements and the timing priority for each activity can be satisfied, and a system does not perform two different activities at the same time. Regardless of the resource constraints, the scheme can be considered as a feasible scheduling scheme, and the total time for completing the mission is 14 h.

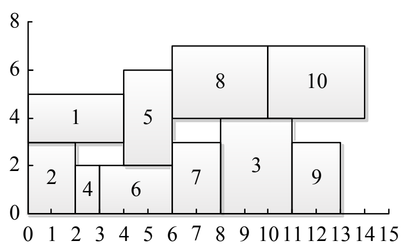


**Fig. 3 Feasible scheduling scheme for the example**

**Table 1 Detailed data of the scheme**

| Activity (requirement) | Participation System (capability) | Duration/h |
|---|---|---|
| 1(2) | Sys3(1), Sys4(1) | 0−4 |
| 2(3) | Sys6(2), Sys7(2) | 0−2 |
| 3(4) | Sys1(2), Sys2(2) | 8−11 |
| 4(2) | Sys5(2) | 2−3 |
| 5(4) | Sys2(2), Sys6(2) | 4−6 |
| 6(3) | Sys3(1), Sys7(2) | 3−6 |
| 7(3) | Sys5(2), Sys8(1) | 6−8 |
| 8(3) | Sys4(1), Sys6(2) | 6−10 |
| 9(3) | Sys9(3) | 11−13 |
| 10(3) | Sys5(2), Sys8(1) | 10−14 |

# 3. Mathematical model

## 3.1 Nominal model of the WSOSSP

We model the scheduling optimization process of the WSOS. In our model, the following assumptions are given:

(i) In the planning horizon, the capability requirements and duration of each operational activity are predetermined. The activities have different feasible strategies, but only one of them can be selected.

(ii) At any time, a system in the WSOS can only provide capabilities for at most one activity.

(iii) The process by which the systems perform an activity is continuous during the mission. In other words, once an activity begins, it cannot be interrupted.

(iv) Starting and ending avtivities are two dummy activities that do not need capabilities and resources, and their duration is 0.

In solving the WSOSSP, the commander should determine how to allocate a limited number of systems in the WSOS to the operational activities. The execution time of the activity is optimized. A mathematical model [P] for the WSOSSP is proposed as follows:

$$\min T = st_e \tag{1}$$

s.t.

$$\sum_{t\in[0,T]} x_i^t \cdot t = st_i,\ \ \forall i\in V \tag{2}$$

$$\sum_{i\in V}\ \sum_{d\in[t-tr_i+1,t]} z_{in}^d \leqslant 1,\ \ \forall t\in[0,T];\ \ \forall n\in N \tag{3}$$

$$st_i + tr_i \leqslant st_j,\ \ \forall i,\ \ j\in V;\ \ i\in P_j \tag{4}$$

$$T_i^{ES} \leqslant st_i \leqslant T_i^{LS},\ \ \forall i\in V \tag{5}$$

$$z_{in}^t \leqslant x_i^t,\ \ \forall i\in V;\ \ \forall t\in[0,T];\ \ \forall n\in N \tag{6}$$

$$x_i^t + y_{in} \leqslant z_{in}^t + 1,\ \ \forall i\in V;\ \ \forall t\in[0,T];\ \ \forall n\in N \tag{7}$$

$$\sum_{n=1}^{N} c_{nl}\cdot y_{in} \geqslant CR_{il},\ \ \forall i\in V;\ \ \forall l\in L \tag{8}$$

$$\sum_{i=0}^{e} tr_i \cdot \sum_{n=1}^{N} rr_n \cdot y_{in} \leqslant R,\ \ \forall i\in V \tag{9}$$

where the objective function (1) minimizes the parameter $T$ which represents the makespan of the whole military mission, and $st_e$ is the start time of the ending activity. Constraint (2) denotes that the start time of each activity $st_i$, $V$ is the set of operational activities in the mission, and $x_i^t$ represents the start time of activity $i$ is $t$. Constraint (3) guarantees that any system in the WSOS can only be assigned to one activity at a time, $N$ is the set of member systems in the WSOS, $z_{in}^d$ represents system $n$ is assigned to activity $i$, and the start time of activity $i$ is $t$. Constraint (4) denotes that the priority relationship between activities, that is, any activity must start after all its preceding activities are completed, $tr_i$ is the duration time of activity $i$, and $P_j$ is the set of all preceding activities of the activity $j$. Constraint (5) denotes the time window for the activity, $T_i^{ES}$ and $T_i^{LS}$ are the earliest start time and the latest start time of the activity $i$. Constraints (6) and (7) construct the logical relationship among the three decision variables, if and only if $x_i^t=1\wedge y_{in}=1$, $z_{in}^t=1$, $y_{in}$ represents system $n$ is assigned to activity $i$. Constraint (8) denotes that the capability requirements of the activity must be satisfied by the participating systems, $L$ is the set of operational capability, $c_{nl}$ is the value of the capability $l$ provided by system $n$, and $CR_{il}$ is the requirement of activity $i$ for capability $l$. Constraint (9) denotes that the amount of consumable resources used cannot exceed the resource limit provided by the WSOS, $rr_n$ is the resource consumption by system $n$ in unit time, and $R$ is the maximum resource consumption provided by the WSOS.

### 3.2 Robust model of the WSOSSP

Each system in the WSOS participating in the operational activities will provide specific capabilities. There are many methods for evaluating the capabilities of a weapon system, such as expert scoring, fuzzy comprehensive evaluation, Lanchester equation, Monte Carlo [26]. When considering the influence of uncertain factors in the battlefield, the capability of the systems is often not fixed, and usually fluctuate within a range. According to the robust optimization, it is defined in an uncertain set to optimize the worst case in the set. In this paper, we use a symmetric and bounded random variable $\tilde{c}_{nl}\in[\bar{c}_{nl}-\hat{c}_{nl},\bar{c}_{nl}+\hat{c}_{nl}]$ to represent the capability of the systems, where $\bar{c}_{nl}$ denotes the expected value of the capability $l$ for the system $n$, and $\hat{c}_{nl}$ is the maximum deviation from the expected capability. In the model [P], only constraint (8) which contains uncertain variables needs to be reformulated. Therefore, constraint (8) can be represented as

$$\sum_{n=1}^{N}\bar{c}_{nl}\cdot y_{in}+\eta_l\cdot\sum_{n=1}^{N}\hat{c}_{nl}\cdot y_{in}\geqslant CR_{il},\ \ \forall i\in V;\ \ \forall l\in L. \quad (10)$$

In (10), we define a random variable $\eta_l$, which takes value according to an unknown but symmetric distribution $[-1,1]$. When $\eta_l=-1$, the various capabilities of the systems are the minimum, which is equivalent to the min-max absolute robust criterion [27]. Although the obtained scheduling scheme can face any variation of the system capabilities, the scheme is too conservative and may cause the WSOS waste a lot of time. In practice, the commander will comprehensively adjust the degree of risk preference and the robustness of the scheme according to their personality characteristics and battlefield situation. We consider the slack robustness criterion [28]. A control parameter $\Gamma_l$ is introduced to adjust the robustness of the scheme. This parameter denotes the number of the weapon systems that can provide the capability $l$ which takes the minimum value. $\Gamma_l$ takes value in $[0,|J_l|], J_l=\{n:\hat{c}_{nl}>0\}$. The larger the value, the higher the robustness. When $\Gamma_l$ is determined, it means that up to $\lfloor\Gamma_l\rfloor$ of systems with capability $l$ take a value of $\bar{c}_{nl}-\hat{c}_{nl}$, one system takes a value of $\bar{c}_{nl}-(\Gamma_l-\lfloor\Gamma_l\rfloor)\hat{c}_{nl}$, and the remaining systems take the expected value of $\bar{c}_{nl}$. Then, the constraint (10) can be formulated as

$$\sum_{n\in N}\bar{c}_{nl}\cdot y_{in}-\lambda_l(\boldsymbol{y},\Gamma_l)\geqslant CR_{il},\ \ i\in V;\ \ \forall l\in L \quad (11)$$

where $\boldsymbol{y}$ denotes the set of the decision variables $y_{\rm in}$, and $\lambda_l(\boldsymbol{y},\Gamma_l)$ is a nonlinear function, according to the strong duality, it can be transformed into an equivalent linear constraint as follows:

$$\lambda_l(\boldsymbol{y},\Gamma_l)=\max_{\{S_l\cup\{r_l\}|S_l\in J_l,|S_l|=\lfloor\Gamma_l\rfloor,\,r_l\in J_l/S_l\}}\left\{\sum_{n\in S_l}\hat{c}_{nl}\cdot y_{in}+(\Gamma_l-\lfloor\Gamma_l\rfloor)\cdot\hat{c}_{nl}\cdot y_{in}\right\}$$

where $\boldsymbol{S}$ represents the set of systems with capability $l$ takes a value of $\bar{c}_{nl}-\hat{c}_{nl}$, $\boldsymbol{J}$ represents the set of all systems with uncertain capability $l$, and $r$ represents the set of remaining systems with uncertain capability $l$.

## 4. Solution method

Although the formulation [P] could be solved by CPLEX directly, it is difficult to solve practical problems, even for a moderate size instance. A Dantzig-Wolfe decomposition method is used in this paper to obtain a master problem and a pricing subproblem. We propose a column generation approach because the cardinality of the variables is extremely large.

In this mathematical model, a column $r$ (i.e., an activity flow) represents a series of activities performed along a timeline. Hence, a feasible solution is a convex combination of multiple activity flows that satisfy the constraints, which is shown in Fig. 4. Two red arrows represent two different activity flows (0-1-5-8-10-11, 0-2-4-6-7-3-9-11).

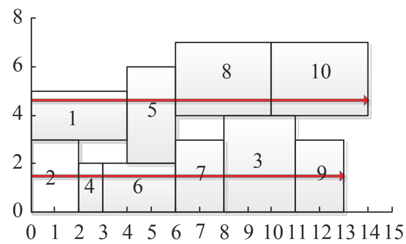


**Fig. 4 Illustrative example of the activity flow**

### 4.1 Master problem

This paper transforms the compact formulation [P] into a set partitioning formulation [MP] (Nannicini et al. [29]), which is regarded as the MP. Using the Dantzig-Wolfe decomposition, a set of constraints ((2)−(9)) in [P] can be replaced by a convex hull.

The MP [MP] can be formulated as follows:

[MP]

$$\min \gamma \tag{12}$$

s.t.

$$\gamma \geqslant T_r \cdot \theta_r,\ \ \forall r \in \varOmega \tag{13}$$

$$\sum_{r\in\varOmega} \alpha_r^i \cdot \theta_r \geqslant 1,\ \forall i \in V \tag{14}$$

$$\sum_{r\in\varOmega} \delta_{rn}^t \cdot \theta_r \leqslant 1,\ \forall d \in [0,T];\ \ \forall n \in N \tag{15}$$

$$\left(\sum_{r\in\varOmega} t_{ri}^S + tr_i\right)\cdot \theta_r \leqslant \sum_{r\in\varOmega} t_{rj}^S \cdot \theta_r,\ \ \forall i,\ \ j \in V;\ \ i \in P_j \tag{16}$$

$$\sum_{r\in\varOmega} I_r \cdot \theta_r \leqslant R \tag{17}$$

$$\theta_r \in \{0,1\},\ \forall r \in \varOmega \tag{18}$$

$$\gamma \geqslant 0 \tag{19}$$

where the objective function (12) and constraint (13) minimize $\gamma$ which represents the makespan of the longest lasting activity flow in a feasible solution, $\varOmega$ is the set of feasible activity flows, $T_r$ is the makespan of the activity flow $r$, and $\theta_r$ represents that the activity flow $r$ is selected. Constraint (14) denotes that all activities must be included in the selected activity flows, each activity must be completed only once, and $\alpha_r^i$ represents that the activity $i$ is selected in the activity flow $r$. Constraint (15) denotes that any system in the WSOS can be only assigned to one activity at a time, and $\delta_{rn}^t$ represents the system $n$ is selected by the activity flow $r$ at time $t$. Constraint (16) denotes that the priority relationship between activities, in other words, any activity can only start when all its preceding activities are completed, and $t_{ri}^S$ represents the start time of the activity $i$ in the activity flow $r$. Constraint (17) denotes that the amount of consumable resources used cannot exceed the resource limit provided by the WSOS, and $I_r$ is the resource consumption by the activity flow $r$. Constraints (18) and (19) clarify the domains of variables.

Given that the activity flow set $\varOmega$ is extremely large, this paper considers a subset $\varOmega'$ of $\varOmega$. This subset only contains those variables (i.e., activity flows) generated by solving the pricing subproblem. The master problem only considering the subset $\varOmega'$ is called the restricted master problem (RMP).

### 4.2 Pricing subproblem

Let $(\mu_r, \pi_i, \rho_n^t, \kappa_{(i,j)}, \upsilon)$ denote the dual variables corresponding to the constraints (13)−(17). If $(\mu_r, \pi_i, \rho_n^t, \kappa_{(i,j)}, \upsilon)$ satisfies complementary slackness, solving RMP provides an optimal solution of master problem. Otherwise, there must exist one activity flow $r \in \varOmega/\varOmega'$ such that $T_r \cdot \mu_r +$ $\sum_{i\in V} \alpha_r^i \cdot \pi_i + \sum_{d\in T}\sum_{n\in N} \delta_{rn}^t \cdot \rho_n^t + \sum_{\forall i,j\in V, i\in P_j} \left(t_{ri}^S + tr_i - t_{rj}^S\right)\cdot \kappa_{(i,j)} + \mathrm{I}_r \cdot \upsilon \leqslant 0$

Thus, the PP is to find one activity flow $r \in \varOmega/\varOmega'$ that achieves the optimal value of the following:

[PP]

$$\varpi_r = \mathrm{Min} \sum_{i\in V}\sum_{j\in V} \chi_{ij} \cdot \pi_i + \sum_{i\in V}\sum_{j\in V} \chi_{ij} \cdot y_{in} \cdot \sum_{t=\tau_i}^{t=\tau_i+tr_i} \rho_n^t + \sum_{\forall i,j\in V, i\in P_j} \chi_{ij} \cdot \left(t_{ri}^S + tr_i - t_{rj}^S\right)\cdot \kappa_{(i,j)} + \beta \cdot \mu_r + \eta \cdot \upsilon \tag{20}$$

s.t.

$$\sum_{j\in V} \chi_{oj} = 1 \tag{21}$$

$$\sum_{i\in V} \chi_{ij} - \sum_{i\in V} \chi_{ji} = 0,\ \forall j \in V \tag{22}$$

$$\sum_{i\in V} \chi_{ie} = 1 \tag{23}$$

$$\sum_{j\in V} \chi_{ij} \leqslant \sum_{n\in N} y_{in},\ \forall i \in V \tag{24}$$

$$(st_i + tr_i)\cdot \chi_{ij} \leqslant st_j,\ \ \forall i,\ \ j \in V;\ \ i \in P_j \tag{25}$$

$$T_i^{ES} \leqslant st_i \cdot \sum_{j\in V} \chi_{ij} \leqslant T_i^{LS};\ \ \forall i \in V \tag{26}$$

$$\sum_{n\in N} \tilde{c}_{nl} \cdot y_{in} \geqslant CR_{il},\ \forall i \in V;\ \ \forall l \in L \tag{27}$$

$$\beta \geqslant \sum_{i,j\in V} \chi_{ij} \cdot st_j \tag{28}$$

$$\eta \geqslant \sum_{i,j\in V} x_{ij} \cdot tr_i \cdot \sum_{n\in N} rr_n \cdot y_{in} \tag{29}$$

$$\chi_{ij} \in (0,1),\ \forall i,\ \ j \in V \tag{30}$$

$$st_i \geqslant 0,\ \forall i \in V \tag{31}$$

$$\beta \geqslant 0 \tag{32}$$

where the objective function (20) minimizes the reduced cost $\varpi_r$. Constraints (21)–(23) denote a flow equilibrium relationship. Constraint (24) denotes the relationship between variables $\chi_{ij}$ and $y_{in}$. Constraints (25)–(26) denote the time window constraints for each activity. Constraints (27)–(29) respectively denote the capability requirements, makespan of the whole mission constraints and the consumable resource constraints. Constraints (30)–(32) clarify the domains of variables. The binary variable $\chi_{ij}$ is equal to 1 if and only if the activities $i$ and $j$ are completed continuously. The PP is also NP-hard [30], it will consume most of the computing time in the process of column generation. Therefore, this paper proposes a tailor dynamic programming to solve it.

It notes that the parameter $\tilde{c}_{nl}$ in constraint (27) is a random variable, thus it needs to be linearized. We refer to the method of Munari et al. [31] to define new state variables $u^{\varepsilon}_{inl}$, which denote the maximum value of the capability $l$ that can be provided when system $n$ performs activity $i$. Therefore, constraint (27) can be extended as follows:

$$\mu^{\varepsilon}_{in'l} \geqslant u^{\varepsilon}_{inl} + c_{nl}\cdot y_{in'} - CR_{il}(1-y_{in'}),\quad \forall n,n' \in N; \forall i \in V; \varepsilon = 0,\cdots,\Gamma_l; \forall l \in L \tag{33}$$

$$u^{\varepsilon}_{in'l} \geqslant u^{\varepsilon-1}_{inl} + (c_{nl}+\hat{c}_{nl})\cdot y_{in'} - CR_{il}(1-y_{in'}),\quad \forall n,n' \in N; \forall i \in V; \varepsilon = 1,\cdots,\Gamma_l, \forall l \in L \tag{34}$$

$$u^{\varepsilon}_{inl} \leqslant CR_{il};\ \forall n \in N; \forall i \in V; \varepsilon = 0,\cdots,\Gamma_l; \forall l \in L. \tag{35}$$

Hence, for the robust labeling algorithm, an implementation of the dynamic programming-based algorithm is used [32], which is called exact dynamic programming (EDP). In the dynamic programming-based algorithm, the start node and end node (i.e., activities) are represented by the same node. We build the new activity flows from the start node toward a sink node, which are encoded by labels. A state associated with a feasible activity flow from the start node 0 to the node $i$ is defined as a multi-dimensional label $\sigma_i = (S_i, \psi^{\varepsilon}_{inl}, \tau_i, \varphi_i, i)$, where $S_i$ is the set indicating which nodes have already been visited by $\sigma_i$; $\psi^{\varepsilon}_{inl}$ is the minimum value of the capability $l$ provided by system $n$ performing activity $i$, when up to $\varepsilon \leqslant \Gamma_l$ capability $l$ attains it worst case; $\tau_i$ is the start time at the node $i$ in the activity flow $\sigma_i$; $\varphi_i$ is the least reduced cost of the activity flow $\sigma_i$; $i$ is the last reached node in $\sigma_i$.

At the start node 0, the set $S_0$ is initialized at $\phi$, and $\psi^{\varepsilon}_{inl}$, $\tau_i$ and $\varphi_i$ are initialized at 0. Each node may have multiple labels and the optimal solution to the pricing subproblem can be achieved by identifying the labels with the smallest $\varphi(r)$ at the node $e$. Note that for the activity flow $r$, and each possible schedule scheme can be performed only at a time. Once a certain time is chosen to perform the schedule scheme by a label, the successors will perform the schedule pattern on the same time. When a feasible label $\sigma_i = (S_i, \psi^{\varepsilon}_{inl}, \tau_i, \varphi_i, i)$ is associated with the node $i$, it can be extended to a node $j \in V$ along an arc $(i, j)$, yielding a new label $\sigma_i$. The extension functions are

$$\begin{cases} S_j = S_i \cup \{j\} \\ \psi^{\varepsilon}_{in'l} = \max\{u^{\varepsilon}_{inl} + c_{nl}, u^{\varepsilon-1}_{inl} + c_{nl} + \hat{c}_{nl}\} \\ \tau_j = \max\{T^{ES}_j, \tau_i + tr_i\} \\ \varphi_j = \varphi_i + \varpi_{ij} \end{cases}. \tag{36}$$

For each time $t$, the label $\sigma^1_i$ dominates the label $\sigma^2_i$ if the following conditions hold:

$$\begin{cases} S^1_i \subseteq S^2_i \\ \psi^{1\varepsilon}_{inl} \leqslant \psi^{2\varepsilon}_{inl},\ \forall \varepsilon = 0,\cdots,\Gamma_l \\ \tau^1_i \leqslant \tau^2_i \\ \varphi(r_1) \leqslant \varphi(r_2) \end{cases}. \tag{37}$$

In this algorithm, $TL_i$ and $UL_i$ denote the sets of treated and untreated labels at the node $i \in V$, respectively. First, we initialize the label in lines 1 to 5. Line 6 is the start of the main loop and all untreated nondominated labels will be extended next. In line 7, a label $L_i$ is chosen in $UL_i$ following a certain rule (the shortest time component first). Then, the label $L_i$ is extended to its successor $j$ to generate a new label $L_j$ according to the expansion function in line 8 and line 9. If label $L_j$ is feasible, we use the dominance rules to examine whether the label $L_j$ is dominated by a label in $UL_j \cup TL_j$ or whether the label $L_j$ dominates a label in $UL_j \cup TL_j$ in line 10 to line 13. Finally, we update the sets of treated and untreated labels in line 16 and get a label $L_{n+1}$ which provides the feasible activity flow with the minimal reduced cost from the node 0 to node $n+1$ performed on the time $t^*$.

The pseudo code of the proposed exact dynamic programming is given in Algorithm 1.

**Algorithm 1** The proposed exact dynamic programming

Create an initial label $L_0 = (\{0\}, 0, 0, 0, \varphi(r), (P_i, \varnothing))$.
  Set $UL^t_0 \leftarrow \{L_0\}$ and $TL^t_0 \leftarrow \{\varnothing\}$.
  **For each** node $i$ **in** $N/\{0\}$
    Set $UL^t_i \leftarrow \{\varnothing\}$ and $TL^t_i \leftarrow \{\varnothing\}$.
  **End for**
  **While** $\bigcup_{i \in V} UL^p_i \neq \varnothing$ **do**
    Choose a label $L_i(t) \in UL^t_i$.
  **For each** $(i, j)$ **in** $V * V$
    Using extension functions, extend $L_i(t)$ along the arc $(i, j)$ to create a label $L_j(t)$.
  **If** $L_j(t)$ is feasible **then**
    Set $UL^t_j \leftarrow UL^t_j \cup \{L_j(t)\}$.
    Discard the labels dominated from the set

$UL_j^t \cup TL_j^t$ according to the dominance rule.

**End if**

**End for**

Set $UL_i^t \leftarrow UL_i^t \backslash \{L_i(t)\}$ and $TL_i^t \leftarrow TL_i^t \cup \{L_i(t)\}$.

**End do**

$L_{n+1}(t^*)$ provides the minimal reduced cost from the node 0 to node $n+1$.

### 4.3 Branching strategy

Since the column generation algorithm solves the linear relaxation model of the master problem, it cannot guarantee that the obtained solution is an integer solution [33]. If the obtained optimal solution is not an integer solution, a branching strategy needs to be adopted to properly modify the non-integer solution. Branching the activity flows in the RMP is a traditional branching strategy. Its disadvantage is the imbalance of the branch tree and the difficulty in solving the pricing subproblem. According to the characteristics of the model in this paper, we use the sum of the flow variables on arcs between operational activities to branch. This branching strategy neither increases the difficulty of solving the node subproblems, nor does it cause imbalance between the left and right subtrees.

Let $\beta_{ijr}$ represent whether the activity flow $r$ passes through the arc $(i,j)$, if the arc $(i,j)$ is included in the activity flow $r$, $\beta_{ijr}=1$, otherwise $\beta_{ijr}=0$. $\bar{x}_{ij}=\sum_{r\in\Omega}\beta_{ijr}\bar{\theta}_r$ be the activity flow of arc $(i,j)$. The arc $(i,j)$ between operational activities is branched according to the formula $(\bar{i},\bar{j})=\arg\min|\bar{x}_{ij}-0.5|$ by creating two left and right child nodes. At the left node, the arc $(i,j)$ is not allowed to appear in the optimal solution, that is, the constraint $\bar{\chi}_{ij}=0$ is added. It is equivalent to assign a sufficiently large positive weight to the arc $(i,j)$ in the subproblem. At the right node, the arc $(i,j)$ must appear in the optimal solution, that is, the constraint $\bar{\chi}_{ij}=1$ is added. It is equivalent to leave only arc $(i,j)$ among all arcs related to node $j$ in the subproblem.

## 5. Computational study

### 5.1 Test bed

In this paper, we design several test instances to verify the effectiveness of the robust WSOSSP model and the performance of the proposed branch-and-price algorithm. Since the number of operational activities in the mission (i.e., $|V|$), the types of capabilities (i.e., $|L|$) and the number of weapon systems in the WSOS (i.e., $|N|$) will all affect the complexity of the WSOSSP, therefore, we select a subset of the iMOPSE dataset, and generate 10 test instances of different scales [34]. The iMOPSE dataset is a standard test dataset which is jointly created after a large number of data analysis and processing by many experienced project managers in Volvo's IT department. Compared to the well-known PSPLIB benchmark dataset [35], the iMOPSE dataset not only includes the execution time of the scheduling plan, but also the economic cost and capability types. What is more, iMOPSE dataset can completely reflect the robust WSOSSP model.

The selected test instances in this paper has been presented in the Table 2, which are divided into two groups. The first group contains 100 activities and another group contains 200 activities. Within each group, different instances are distinguished according to the number of weapon systems and the number of priority relationships between operational activities. For each case, 9 or 15 different capability types have been set up. Each of these activities can be completed when it meets specified operational capability requirements and different system combinations that meet the capability requirements can perform different activities, consuming some resources at the same time. Because of the difference in the number of weapon systems and preceding relations, the computational complexity for each instance is varied.

**Table 2 iMOPSE dataset instances**

| Group | Instance | Activity | System | Relation | Capability |
|---|---|---|---|---|---|
| G1 | 100-10-48-15 | 100 | 10 | 48 | 15 |
| | 100-20-46-15 | 100 | 20 | 46 | 15 |
| | 100-5-20-9-D3 | 100 | 5 | 20 | 9 |
| G2 | 200-10-50-9 | 200 | 10 | 50 | 9 |
| | 200-10-85-15 | 200 | 10 | 85 | 15 |
| | 200-20-97-9 | 200 | 20 | 97 | 9 |
| | 200-40-133-15 | 200 | 40 | 133 | 15 |
| | 200-40-45-9 | 200 | 40 | 45 | 9 |
| | 200-40-90-9 | 200 | 40 | 90 | 9 |
| | 200-40-91-15 | 200 | 40 | 91 | 15 |

### 5.2 Computational results

We use the CPLEX 12.6 solver and the branch-and-price algorithm to calculate the optimal solutions for these 10 test instances. The calculation results are shown in Table 3 and the contents of the indicators in each column are explained as follows.

In Table 3, the three columns belong to the solution indicators of the CPLEX: the average calculation time of the algorithm is represented by "ACT", the model optimal solution is represented by "OPT" and the number of explored nodes when the calculation is terminated is represented by "Node". The following columns are related to the branch-and-price algorithm. The number of columns generated in the MP is represented by “col. in MP”. The average calculation time in the MP and the algorithm total time are represented by “ACT in MP” and “ACT”. “LB” represents the lower bound of the objective function, its corresponding solution is a relaxed real number solution, “OPT” represents the objective function obtained by the branch pricing algorithm, and its corresponding solution must be a feasible integer solution. The column “Node” represents the number of nodes searched by the branch-and-price algorithm. The calculation process is automatically terminated when the calculation time reaches 1 h. The last column “GAP” reflects the relative difference between the objective function values of the two algorithms. The “GAP” formula is given as follows:

$$\text{GAP} = \frac{\text{OPT}_{\text{B\&P}} - \text{OPT}_{\text{CPLEX}}}{\text{OPT}_{\text{CPLEX}}}.$$

**Table 3 Performance comparison of the brand-and-price algorithm and CPLEX**

| Instance | CPLEX | | | Branch-and-price | | | | | | GAP/% |
|---|---|---|---|---|---|---|---|---|---|---|
| | ACT | OPT | Node | col. in MP | ACT in MP | ACT | LB | OPT | Node | |
| 100-10-48-15 | 1751 | 792 | 9701 | 961 | 145 | 1134 | 792 | 792 | 14 | 0.00 |
| 100-20-46-15 | 1954 | 600 | 14984 | 1443 | 180 | 1395 | 600 | 600 | 38 | 0.00 |
| 100-5-20-9-D3 | 1648 | 1200 | 12505 | 727 | 135 | 749 | 1200 | 1200 | 25 | 0.00 |
| 200-10-50-9 | 2561 | 1560 | 19400 | 1626 | 202 | 1683 | 1560 | 1560 | 112 | 0.00 |
| 200-10-85-15 | 2948 | 1464 | 22441 | 3548 | 242 | 1990 | 1464 | 1464 | 154 | 0.00 |
| 200-20-97-9 | 3474 | 888 | 27832 | 5137 | 212 | 2451 | 888 | 888 | 189 | 0.00 |
| 200-40-133-15 | 3600 | 720 | 31576 | 7126 | 295 | 2846 | 648 | 648 | 265 | −10.00 |
| 200-40-45-9 | 3600 | 696 | 37072 | 9552 | 362 | 3399 | 624 | 624 | 378 | −10.34 |
| 200-40-90-9 | 3600 | 696 | 39098 | 11660 | 450 | 3600 | 576 | 600 | 466 | −13.79 |
| 200-40-91-15 | 3600 | 672 | 41687 | 14235 | 481 | 3600 | 552 | 576 | 555 | −14.29 |

In terms of average calculation time and the optimal solution, the performance of the branch-and-price algorithm is better than CPLEX. Firstly, we compare the average calculation time, as shown in the column “ACT”, the CPLEX solver obtained the optimal solution for 6 of the 10 instances, and the branch-and-price algorithm obtained the optimal solution for 8 instances within 1 h. For the same instance, the branch-and-price algorithm solves faster than CPLEX. Although the average calculation time of two algorithms increases sharply with the increase of the problem scale, the branch-and-price algorithm increases relatively slowly. The main reason is that CPLEX directly solves the linear relaxation model of the original model, so the lower bound at the root node is poor, and it takes plenty of time to find the optimal integer solution in the branch and bound process.

Secondly, in 4 of the 10 test instances, the gap is less than 0. This is because a tight lower bound is provided by the set partition formula in the subproblem. A tight lower bound cannot only improve the branching effect, but also save a lot of time for the branch and bound process. In instances 200-40-133-15 and 200-40-45-9, CPLEX does not find the optimal solution, while the branch-and-price algorithm find it in a limited time. In instances 200-40-90-9 and 200-40-91-15, neither algorithm finds the optimal solution, but it is obvious that the branch-and-price algorithm can get a better upper bound than CPLEX.

In addition, we compare the number of nodes explored in the two algorithms. As shown in the column “Node”, in the same instance, the number of nodes explored by CPLEX is much larger than the branch pricing algorithm. The two main reasons are as follows. On the one hand, the branch-and-price algorithm proposes a compact set partition method, which defines the activity flow in the scheduling process as columns. Compared with the intensive variables in the original model, the new model definition form can provide a tighter lower bound for calculation. On the other hand, due to the constraints of priority relationships and capability requirements, only activity flows that meet the constraints will be generated. Therefore, the scale of the MP is small, and the lower bound of the set partition model always corresponds to a relaxed solution.

As the instance size increases, the space and time complexity of the branch-and-price algorithm will increase exponentially. By comparing the calculation time of different stages in the algorithm, we find that solving the subproblem model will consume most of the time. That is because with increase in the scale of instances, the length of an activity flow in the subproblem grows dramatically (in some cases, an activity flow contains 20 elements). In other words, the process of dynamic programming in the subproblem model will take up a lot of calculation time. We are actually exploring a subproblem where the size of the network is growing exponentially. At the same time, it can be seen that as the problem scale increases, the number of “col. in MP” also increases exponentially, and the branch-and-price algorithm still has difficulties in solving large-scale instances.

## 6. Conclusions

The WSOS scheduling decisions are important for efficiently completing military missions. In this paper, we design a robust model to optimize the WSOSSP. At the same time, a slack robustness criterion is used to describe the uncertainty of the capabilities of the weapon system. A branch-and-price algorithm is developed to solve the model. We propose a novel set partitioning formulation, in which an activity flow is defined as a column. Thus, the subproblem is to identify the activity execution order and the distribution of member systems. The performance of the proposed algorithm is tested by several instances based on the iMOPSE dataset.

The results show that after optimizing, the makespan of the WSOSSP is significantly reduced. From the perspective of computational efficiency, the solution of the branch-and-price algorithm is extremely stable, and its performance is significantly better than CPLEX, which can solve medium-scale instances in a reasonable time.

## References

[1] EDWARD C, AMIT S. Next-generation smart environments: from system of systems to data ecosystems. IEEE Intelligent Systems, 2018, 33(3): 69–76.
[2] ZHAO Q S, DING J Y, LI J C, et al. Mission-oriented scheme generation method for weapon system of systems. IEEE Access, 2020, 8: 70981–70996.
[3] WANG H, LAPPAS N H, GOUNARIS C E. Multi-mode resource constrained project scheduling with alternative prerequisites: new models and computational studies. Industrial & Engineering Chemistry Research, 2019, 58(39): 18253–18266.
[4] CHEN P H, WENG H J. A two-phase GA model for resource-constrained project scheduling. Automation in Construction, 2009, 18(4): 485–498.
[5] NAJID N M, ARROUB M. An efficient algorithm for the multi-mode resource constrained project scheduling problem with resource flexibility. International Journal of Mathematics in Operational Research, 2010, 2(6): 748–761.
[6] WANG L, FANG C. An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem. Computers & Operations Research, 2012, 39(2): 449–460.
[7] AYODELE M, MCCALL J, REGNIER C O. RK-EDA: a novel random key based estimation of distribution algorithm. Proc. of the International Conference on Parallel Problem Solving from Nature, 2016: 849−858.
[8] ADAMU P I, AGARANA M C, OKAGBUE H. Learning heuristic for solving multi-mode resource-constrained project scheduling problems. Proc. of the International Multiconference of Engineers and Computer Scientists, 2018: 14−16.
[9] AFSHAR M R, SHAHHOSSEINI V, SEBT M H. A genetic algorithm with a new local search method for solving the multimode resource-constrained project scheduling problem. International Journal of Construction Management, 2019, 19(3): 1–9.
[10] ROSON J H, KULEJEWSKI J E. A hybrid approach for solving multi-mode resource-constrained project scheduling problem in construction. Open Engineering, 2019, 9(1): 7–13.
[11] RATAJCZAK R E. Experimental evaluation of agent-based approaches to solving multi-mode resource-constrained project scheduling problem. Cybernetics and Systems, 2018, 49(2): 1–21.
[12] ZSOLT T K, SZALKAI I. Multimode resource-constrained project scheduling in flexible projects. Journal of Global Optimization, 2020, 76(1): 211–241.
[13] VAHDANI H, SHAMS A. Multi-mode capital-constrained project payment scheduling model considering bonus-penalty structure. International Journal of Management Science & Engineering Management, 2020, 15(1): 17–25.
[14] SPRECHER A, DREXL A. Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. European Journal of Operational Research, 1998, 107(2): 431–450.
[15] KYRIAKIDIS T S, KOPANOS G M, GEORGIADIS M C. MILP formulations for single- and multi-mode resource-constrained project scheduling problems. Computers & Chemical Engineering, 2012, 36(1): 369–385.
[16] JOSE C, MARIO V. Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers. European Journal of Operational Research, 2011, 213(1): 73–82.
[17] SCHNELL A, HARTL R F. On the efficient modeling and solution of the multi-mode resource-constrained project scheduling problem with generalized precedence relations. OR Spectrum, 2016, 38(2): 283–303.
[18] BOFILL M, COLL J, SUY J, et al. Solving the multi-mode resource-constrained project scheduling problem with SMT. Proc.of the IEEE International Conference on Tools with Artificial Intelligence, 2017: 239–246.
[19] ALTINTAS C, AZIZOGLU M. A branch and bound algorithm for a multi-mode project scheduling problem with a single non-renewable resource. International Journal of Information Technology, 2020, 11(2): 55–70.
[20] LI R Y, WANG Z X, YU M G, et al. Multi-objective portfolio optimization of system of systems based on robust capabilities. Systems Engineering and Electronics, 2019, 41(5): 103–111. (in Chinese)
[21] LEUS R, HERROELEN W. The complexity of machine scheduling for stability with a single disrupted job. Operations Research Letters, 2005, 33(2): 151–156.
[22] KAVEH A, KHANZADI M, ALIPOUR M. Fuzzy resource constraint project scheduling problem using CBO and CSS algorithms. International Journal of Civil Engineering, 2016, 14(5): 325–337.
[23] BIRJANDI A, MOUSAVI S M. Fuzzy resource-constrained project scheduling with multiple routes: a heuristic solution. Automation in Construction, 2019, 100(4): 84–102.
[24] ANGELA C, YUN C L, JOSE D P. An entropy-based upper bound methodology for robust predictive multi-mode RCPSP schedules. Entropy, 2014, 16(9): 5032–5067.
[25] SHAN W X, PENG Z X, LIU J M, et al. An exact algorithm for inland container transportation network design. Transportation Research Part B: Methodological, 2020, 135:

41–82.

[26] DRAUDVILIENE L, MESKUOTIENE A, RAISUTIS R, et al. The capability assessment of the spectrum decomposition technique for measurements of the group velocity of lamb waves. Journal of Nondestructive Evaluation, 2018, 37(2): 29–42.

[27] LIAO W Z, FU Y X. Min-max regret criterion-based robust model for the permutation flow-shop scheduling problem. Engineering Optimization, 2020, 52(4): 687−700.

[28] LI R Y, HE M, HE H Y, et al. Heuristic column generation for designing an express circular packaging distribution network. Operational Research, 2020, 1: 1–24.

[29] NANNICINI G, SARTOR G, TRAVERSI E, et al. An exact algorithm for robust influence maximization. Mathematical Programming, 2020, 183(1): 419–453.

[30] REIHANEH M, GHONIEM A. A branch-cut-and-price algorithm for the generalized vehicle routing problem. Journal of the Operational Research Society, 2018, 69(2): 307–318.

[31] MUNARI P, MORENO A, JONATHAN D, et al. The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. Transportation Science, 2019, 53(4): 1043–1066.

[32] WEI Q, WU Y. Dynamic programming algorithms for two-machine hybrid flow-shop scheduling with a given job sequence and deadline. IEEE Access, 2020, 8: 89964–89975.

[33] WANG Q, LIU C C, ZHENG L. A column-generation-based algorithm for a resource-constrained project scheduling problem with a fractional shared resource. Engineering Optimization, 2020, 52(5): 798–816.

[34] MYSZKOWSKI P B, MAREK E S, OLECH U P, et al. Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. Soft Computing, 2015, 19(12): 3599–3619.

[35] EYNDE R, VANHOUCKE M. Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling. Journal of Scheduling, 2020, 23: 301–325.

## Biographies

**LI Ruiyang** was born in 1992. He is a Ph.D. student in Institute of Command and Control Engineering, Army Engineering University of PLA. After graduating from college in 2014, He was admitted to the PLA University of Science and Technology, his profession was system analysis and integration. In the master’s period, his major research interests were operational optimization, operational effectiveness assessment and artificial intelligence algorithm. Then he continued to pursue a doctorate in software engineering. His major research interests are requirement engineering and model and method of system of systems planning and management.
E-mail: 15601591998@163.com

**HE Ming** was born in 1978. He is a professor in Institute of Command and Control Engineering, Army Engineering University of PLA. He received his B.Sc., M.Sc. and Ph.D. degrees from PLA Science and Technology University in 2000, 2003 and 2007, respectively. His main research interests focus on emergency command, big data analytics, multi-agent control and situation awareness.
E-mail: heming@126.com

**HE Hongyue** was born in 1985. He received his M.S. degree in 2007 and Ph.D. degree in 2014 from PLA University of Science and Technology. Now, he is a lecturer of Army Engineering University of PLA. His research interests are system of systems engineering, focusing on specification.
E-mail: hehy2008@sina.com

**WANG Zhixue** was born in 1961. He is a professor in Institute of Command and Control Engineering, Army Engineering University of PLA. He received his M.S. degree from National University of Defense and Technology, and was a visiting researcher in Faculty of Information Technology, University of Brighton, England. His research interests are software engineering, requirements engineering, theory and technology of command automation, currently focusing on domain specific modeling and formal verification.
E-mail: wzxcx@163.com

**YANG Cheng** was born in 1990. He is a master student in Institute of Command and Control Engineering, Army Engineering University of PLA. His major is computer technology. His main research interests are machine learning and the internet of things.
E-mail: 978008436@qq.com