Task Scheduling Method Based on Feasible Task Execution Sequence and Greedy Strategy

To cite this article: Jieyong Zhang et al 2018 J. Phys.: Conf. Ser. 1060 012051

View the article online for updates and enhancements.
Task Scheduling Method Based on Feasible Task Execution Sequence andGreedy Strategy

Jieyong Zhang1,2,3, Songhua Huang1,3, Peng Sun2 and Guanyu Chen2,*

1 Science and Technology on Information Systems Engineering Laboratory, Nanjing Jiangsu, China
2 College of Information and Navigation, Air Force Engineering University, Fenghao East Road NO.1, Xi’an, Shanxi, China
3 Collaborative Innovation Center for Novel Software and Industrialization, Nanjing

*ChenPTN@163.com

Abstract. In the battlefield environment, pre-war mission plan needs to be made adaptive adjustments in the course of the war. The adjustment of the mission plan is a difficult problem in the field of command and control. Aiming at the task planning adjustment problem in the command and control organization, a task planning adjustment model is established to get the minimum mission completion time as the objective function and to complete the precision as the constraint condition. This paper proposes a task planning adjustment method based on feasible task execution sequence and greedy algorithm (GA), and gives the design idea and detailed steps of the task planning adjustment method. Finally, combined with the simulation of joint landing operations, the simulation experiment verifies the feasibility and effectiveness of the method in dealing with emergencies.

1. Introduction
The method of getting a reasonable and efficient task-platform relation has become a hot issue in the field of command and control [1]. Many scholars study mission planning problem and have achieved some valuable results.

Levchuk proposed a multidimensional dynamic list scheduling (MDLS) algorithm combined with PWE to solve the allocated problem [2]; Yang proposed a multi-priority list dynamic scheduling (MPLDS) algorithm based on the MDLS algorithm [3]. Zhang proposed a task-platform resource assigning method based on dynamic list scheduling and genetic algorithm [4].

The battlefield environment is complex and full of various uncertainties [5]-[7]. When the mission and platform are changed, the task-platform relation planned before change should be adjusted in time to make sure the mission goal will come true successfully.

At present, few studies are made on dynamic planning adjusting during the war. In this paper, we establish a mathematical model of task plan adjustment. Then, we proposes a task planning adjustment method based on feasible task execution sequence and greedy algorithm.

2. Task planning adjustment problem modelling

2.1. Basic concepts
The key entities in military organizations are defined as follows [1].
**Definition 1 Task:** A mission is the goal of a battle, mission is made up of multiple tasks. The completion of a task needs the support of one or more platforms.

**Definition 2 Resource:** A resource is a measurable physical or virtual entity used in the processing of tasks, and each task is specified by resource requirements in each functional resource category.

**Definition 3 Platform:** A platform has specified resource capabilities that can be assigned to execute tasks. The relationship matrix of task and platform is \( M(t)^{JP} = (m(t)^{JP})_{nt} \).

2.2. Variable definition

1) at \( t \) time, if emergency event happened, the set of unprocessed task is \( S_t^{un} (t') \). The set of platform is \( S_p (t') \). Otherwise, it is \( S_t^{un} (t) \) and \( S_p (t) \) dividedly.

2) \( R_i = [R_{i1}, ..., R_{ij}, ..., R_{iL}] \) : resource requirement vector, where \( R_{il} \) is the number of units of resource type \( l \) required for successful processing of task \( i \). \( r_j = [r_{j1}, ..., r_{jl}, ..., r_{jL}] \) : resource capacity vector, where \( r_{jl} \) is the number of units of resource type \( l \) possessed by platform type \( j \).

3) initial task execution sequence is \( TS = [T_{s1}, T_{s2}, ..., T_{sn}] \), \( NT \) is the number of task.

4) BEFORE \( (i) \) is the set of direct predecessor task set of task \( i \).

5) BACK \( (i) \) is the set of direct successor of task \( i \).

6) the platform allocation of task \( i \) is \( y_i = [y_{i1}, ..., y_{ij}, ..., y_{il}]^T \). Where \( y_{ij} \) indicates how many platforms of type \( j \) are allocated to task \( i \).

7) \( m_{ij} \) : assignment variables, if \( P_j \) is assigned to process \( T_i \), then \( m_{ij} = 1 \), otherwise \( m_{ij} = 0 \).

8) MIT: mission completion time, mission completion time is the time when last task is completed. It can be obtained by the equation (1). \( t_{s,j} \) is the start time of task \( I \), \( t_{p,j} \) is the processing time of task \( i \).

\[
MIT = \max_{i=1,2,..,N} (t_{s,i} + t_{p,i})
\]  

2.3. Objective function

With the constraint condition above, the plan whose completion time is the shortest is the best plan. Thus, the minimized mission completion time is taken as the objective function.

\[
\min MIT
\]  

2.4. Constraints

1) Accuracy

To define the accuracy well, we adapt the geometric mean as our accuracy metric. Formally.

\[
Acc(i) = \prod_{\gamma(i) \cap \gamma(j) \neq \emptyset} \min_{1 \leq l \leq N} \left( \frac{1}{\sum_{\gamma(i)} r_{lj} y_{lj}} R_{lj} \right)^{\frac{1}{\gamma(i)}}
\]  

Here, \( Acc(i) \) denotes the accuracy of task \( i \), \( \gamma(i) \) denotes the set of resource categories required by task \( i \), \( |\gamma(i)| \) is the cardinality of \( \gamma(i) \). The \( Acc(i) \) is 0, if missing any required resource.

\[
Accuracy = \sum_{i=1}^{NT} w_i Acc(i)
\]  

\( w_i \) is the weight of task \( i \). \( Acc(i) \) and \( Accuracy \) should greater than threshold \( \gamma \) and \( \delta \) respectively.

2) New added task allocation order

Battlefield situation changes at \( t \) time. The newly added task is \( T_{add} \). After inserting \( T_{add} \) into \( TS_{new} = [T_{s1}, T_{s2}, ..., T_{sn}, T_{s1}, T_{s2}, ..., T_{sn}] \), the new task sequence is \( TS' = [T_{s1}, ..., T_{sadd}, ..., T_{sn}] \). In this sequence,
the order of \( T_{add} \)'s direct predecessor task is in front of \( T_{add} \)'s order, the order of \( T_{add} \)'s direct predecessor is behind \( T_{add} \)'s order.

\[
\begin{align*}
BEFORE(T_{add}) & \subseteq \{ T_n, \ldots, T_{n-1}, T_{add} \} \\
BACK(T_{add}) & \subseteq \{ T_{add}, T_{n-1}, \ldots, T_n \}
\end{align*}
\]

(5)

3) The stability of adjusting

The set of damaged platform is \( S^\text{broken}_{p} = S_p(t) - S_p(t) \cap S_p(t') \). The set of impacted task is \( S^\text{impact}_{t} = \{ T_p | P_j \in S^\text{broken}_{p}, m(t)_{ij} = 1 \} \cap S_t(t') \). The tasks whose \( Acc(i) \) above threshold belongs to \( S^\text{keep}_{t}(t') = S^\text{wait}_{t}(t') - S^\text{add}_{t} - S^\text{impact}_{t} \) won’t be reassigned platform.

2.5. Problem modelling

Combined with the objective function and various constraints, the mathematical model for the adjustment problem is as follows.

\[
\begin{align*}
\text{min} & \quad \text{MIT} \\
\text{s.t.} & \quad Acc(i) \geq \gamma \\
& \quad (i = 1, 2, \ldots, NT) \\
& \quad \text{Accuracy} \geq \delta \\
& \quad BEFOR(E(T_{add}) \subseteq \{ T_n, \ldots, T_{n-1}, T_{add} \} \\
& \quad BACK(T_{add}) \subseteq \{ T_{add}, T_{n-1}, \ldots, T_n \} \\
& \quad m(t)_{ij} = m(t)_{ij} \text{ for } (T_p, P_j) \\
& \quad \text{for } (T_p, P_j) \in S^\text{keep}_{t}(t'), P_j = P_j \\
& \quad \text{for } (T_p, P_j) \in S^\text{wait}_{t}(t') \cap S_p(t), P_j = P_j
\end{align*}
\]

(6)

3. Solving method based on feasible task execution sequence and greedy strategy

The insertion position of new task that meets the constraint may be one or more. It means \( TS' \) may be one or more. The set of \( TS' \) is \( \text{Sequence} = \{ TS_1', TS_2', \ldots, TS_{SQ}' \} \), \( SQ \) is the total number of sequence. Using traversal method to find out all feasible \( TS' \), if new task appear.

The task that need to be reallocated is \( S^\text{assign}_{t} = T_{add} \cup S^\text{impact}_{t} = \{ T_{assign}, T_{assign}, \ldots, T_{assign}, \ldots, T_{assign} \} \). The platform set assigned to \( T_{assign} \) is \( S^\text{add}_{p} \). \( M_{BEST} \) is scheduling scheme corresponding to each \( TS' \). \( O \) is the set of scheduling scheme \( O = \{ M_{BEST}, M_{BEST}, \ldots, M_{BEST} \} \). The specific steps are as follows.

Step 1: Let \( v = 1 \).

Step 2: Select the task sequence \( TS'_v \) from \( \text{Sequence} \).

Step 3: Assign all of platform to task in \( S^\text{assign}_{t} \), when the task being executed. It means \( \forall T_{assign} \in S^\text{assign}_{t} \), \( S^\text{temp}_{p\text{temp}} = S_p(t) \). This is initial scheduling plan \( M_0 \), calculate the mission completion time \( MIT_0 \), \( as = 0 \).

Step 4: If \( as \leq AS \), select \( T_{assign} \) from \( S^\text{assign}_{t} \) to delete redundant platform, \( as = as + 1 \); Otherwise, go to step 7.

Step 5: If \( as = 1 \), then \( M_0 = M_{best} \), \( MIT_0 = MIT_{best} \). Let \( M_{temp} = M_{best} \), \( MIT_{temp} = MIT_{best} \), delete one redundant platform from \( S^\text{add}_{p} \). The principle of deleting platform is delete this platform will reduce
mission complete time most. After the redundant platform is deleted, the accuracy of $T_{assign}$ should fulfill $Acc \geq \gamma$. The scheduling plan is $M_{best}$, the complete time is $MIT_{best}$.

Step 6: If deleting any platform in $S_P^m$, the accuracy of $T_{assign}$ will can’t meet $Acc \geq \gamma$, or the mission complete time does not get less, then stop deleting, return to step 4. If the stop condition is not fulfilled, then return to step 5.

Step 7: The optimal solution recorded in the allocation solution set $O$; $v = v + 1$, if $v \geq SQ$, output the mission distribution strategy with the $M_{best}$ from the set $O$; Otherwise, return to step 2.

4. Case analysis
This paper simulates the landing operation in [8]. In this case, the operational mission was dismantled into 18 sub-tasks, and the relationship between the tasks is shown in figure 1.

The MDLS algorithm is applied to pre-war task planning [9]-10, and WL algorithm is selected as priority function [2], and the following task assignment plan is obtained according to the information of task and platform in [8], as shown in figure 2. The execution accuracy of each task is 1.

Under the initial operational conditions, the task execution sequence is $TS=[T_5,T_6,T_1,T_3,T_4,T_{17},T_{18},T_2,T_7,T_9,T_{11},T_{13},T_{10},T_{12},T_{15},T_{14},T_{16}]$, and the mission completion time is 192.9529.

4.1. Simulation Analysis of Task Added Event
When the mission is carried out to $t=60$, the battlefield situation changed and the operational mission was adjusted. A new increase is task $T_{19}$, and $T_{19}$ is inserted between $T_{14}$ and $T_{16}$ in Figure 2. The
location of $T_{19}$ is $(5, 95)$, the vector of resource requirement is $[0,0,0,0,8,0,6]$, and the processing time of task is 20.

It can be seen from figure 2 that when $t = 60$, $T_1$, $T_3$, $T_4$, $T_5$, $T_7$ and $T_{18}$ have already been executed. In figure 2, $TS_{init} = [T_7,T_8,T_9,T_{11},T_{13},T_{16},T_{12},T_{15},T_{14},T_{16}]$. After new task $T_{19}$ is added to $TS_{init}$, $TS_{init}$ changes to $TS' = [T_7,T_8,T_9,T_{11},T_{13},T_{16},T_{12},T_{15},T_{14},T_{10},T_{16}]$.

Using the adjustment method above to assign platform to $T_{19}$, the task-platform relationship after adjusting is shown in figure 3. The thresholds $\gamma$ and $\delta$ are both set to 1.

![Figure 3. Task-platform relationship after adjusting (add task)](image)

In figure 3, $P_{12}$, $P_{13}$, $P_{17}$, $P_{18}$, $P_{19}$ and $P_{20}$ are assigned to $T_{19}$. The allocated platform resources meet the requirements of $T_{19}$. The platforms to which other tasks are assigned are the same as the original allocation.

After new task is added, the completion time of the mission is increased from 192.9529 to 206.1240, the completion time increases by 6 seconds. The processing time of newly added task is 20. The adjustment algorithm solves the problem of task added effectively.

4.2. Simulation Analysis of Platform Damage Event

When the mission is carried out to $t=60$, platform $P_7$ is damaged. $T_{11}$ and $T_{12}$ whose accuracy is lower than the threshold should be reassigned platform with $\gamma$ and $\delta$ both are set to 1.

From figure 2, before the platform $P_7$ is damaged, the platform assignment of tasks $T_{11}$ and $T_{12}$ is shown in table 1. After platform $P_7$ is destroyed, the mission plan adjustment is carried out, and the platforms reassigned for $T_{11}$ and $T_{12}$ are shown in table 2.

| Task   | Assigned platform |
|--------|-------------------|
| $T_{11}$ | $P_7,P_4$         |
| $T_{12}$ | $P_{14},P_7$     |

| Task   | Assigned platform |
|--------|-------------------|
| $T_{11}$ | $P_1,P_2,P_3,P_4$ |
| $T_{12}$ | $P_1,P_2,P_3,P_4$ |

The adjusted Gantt chart is shown in figure 4 under the condition that the platform-related constraints are satisfied. The mission completion time changes to 206.2094 after adjustment. Compared with the initial mission completion time, the new mission completion time increased by 14, which meets the requirements, and solving the task plan adjustment problem after the platform loss.
5. Conclusion
This paper studies the adjustment of mission plan and establishes the mission plan adjustment model with objective of minimizing mission completion time. A solution to the model is proposed based on the feasible task execution sequence and greedy algorithm. At last, the experimental simulation of landing operation is carried out. The mission plan is adaptively adjusted after battlefield environment changed. The experimental results demonstrate the effectiveness of the method for mission planning adjustment.

Acknowledgments
Supported by Natural Science Foundation (Grant No. 61573017) and Natural Science Foundation of Shanxi Province (Grant No. 2017JM6062).

References
[1] Xie B, Lin H. 2013 Generation time analysis of operation resource planning Command Control & Simulation vol 35 no 6 pp 21-25.
[2] Levchuk G M, Levchuk Y N, Luo J, et al. 2002 Normative design of organizations - part I: Mission planningIEEE Trans. on Systems, Man, and Cybernetics vol 32 no 3 pp 346-359.
[3] Yang D S, Zang W M, Lu Y L. 2006 Research on Mathematical description and solving algorithms of tasks scheduling for campaign Systems Engineering-Theory & Practice vol 1 pp 26-34.
[4] Zhang J Y, Yao P Y, Zhou X X, Wang X. 2012 Approach to operation task and platform resource matching based on DLS and GA Systems Engineering and Electronics vol 34 no 5 pp 947-954.
[5] Liu H X. 2010 Design and adjust methodology of command and control organization Changsha: National University of Defense Technology.
[6] Xiu B X. 2006 Design methodology of C2 organizational structure and its analysis of robustness and adaptivity Changsha: National University of Defense Technology.
[7] Zhang J Y, Yao P Y, Li F. 2012 Task-platform relation design model and its algorithm under completion time constraint Systems Engineering and Electronics vol 34 no 8 pp 1621-1629.
[8] Yu F, Tu F, Pappipati K R. 2008 Integration of a holonic organizational control architecture and multiobjective evolutionary algorithm for flexible distributed scheduling IEEE Transactions on Systems, Man, and Cybernetics vol 38 no 5 pp 1001-1017.
[9] Zhou X X, Yao P Y, Zhang J Y, Wang X. 2012 Platform resource scheduling method based on DLS and ACO Computer Science vol 39 no 6 pp 98-103.
[10] Lu Y L, Yang D S, Liu Z, Dai C H. 2006 Research on algorithm of resource allocating in joint operation Fire Control and Command Control vol 2 pp 12-16.