Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 2814 – 2820

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Approach to the Problem of Operation Task and Platform Resource Matching based on MPLDS and PWS

Bai Yun[a,b], Zhang Fengming[b], Zhang JieYong[b], Che Wanfang[a], Wang Shenshen[a,b]

[a]*Key Lab of Complex Aviation System Simulation, Beijing 100076, China*
[b]*Air Force Engineering University, Xi'an 710038, China*

Abstract

The Problem of operation tasks and platform resources matching(MPoTP) is the main topic in the preparation phase of battle. In order to consider the loss of the platform capacity in the process of combat, the loss coefficient is introduced in the process of problem modeling, and the problem model can be more conformable with actual combat. An approach to the problem model based on the multi-PRI list dynamic scheduling (MPLDS) and pair-wise exchange (PWE) is proposed. In the basic of the solution which is obtained by MPLDS algorithm, the PWE method is imported which improves the solution by considering all possible task assignment sequences obtained by exchanging the task at the current place in the assignment sequence with some other task. At last, the superiority and applicability of this approach are illuminated by case analysis.

Keywords: operation research; matching between task and platform; MPLDS; PWE

1. Introduction

The problem of task and platform matching (hereinafter referred to as MPoTP) proposed in Ref.[1,2] is to solve the phase-one problem in the three phases of command and control (C2) organization structure design method. Common solutions include 'branch and bound', 'dynamic scheduling (DP)' and 'dynamic list scheduling (DLS)'. Essentially, MPoTP is a platform scheduling and optimization problem for executing allocated tasks, and multi-dimensional dynamic planning algorithm has been proposed by designating the minimum task completion time as the optimization objective [1,5,6]. The multi-priority

* Corresponding author. Tel.: +86-010-6706-0085.
E-mail address: yunbai13@hotmail.com

doi:10.1016/j.proeng.2012.01.396

list of dynamic scheduling (MPLDS) algorithm has been designed for the problem to achieve more effective task-platform matching scheduling [7-8]. Compared with the MDLS algorithm, MPLDS algorithm focuses on improvements in the phase of platform allocation for selected tasks, instead of changing the order (hereinafter task allocation order) of task allocation platform for the entire mission. Thus, solution obtained through MPLDS algorithm (i.e., mission completion time) may also be a suboptimal solution, because task allocation order in MPLDS algorithm is approximately optimal [1].

Therefore, based on reconciliation of task allocation orders obtained by using the MPLDS algorithm for model solution, this paper introduces the pair-wise exchange (PWE) method to realize the solution improvements by considering all possible tasks allocation orders obtained after the task pair-wise exchange.

2. Modeling the Matching Problem of Task and Platform

2.1 Basic Concepts of MPoTP

The tasks, platforms, and their attributes for achieving the matching need to be described first.

(1) Tasks: By mission decomposition, the mission task sequence diagram can be obtained. where,

Task Set: $\boldsymbol{T}=\{T_i\}$ (i = 1,2, ..., N) (N indicates the number of tasks in task sequence diagram);

Attribute of the Task:

$\boldsymbol{t}_i$ (i=1,2,…,N) - the processing time of T_i;

$\boldsymbol{La}_i = (x_i, y_i)$ - geographical coordinates of implementing T_i;

$\boldsymbol{RE}_i = (R_{i1},R_{i2},\ldots,R_{iL})$ - the resource capacity vector required to successfully implement T_i, where R_{il} is the amount of the lth type of resource capacity required to successfully implement T_i (l=1,2,…,L, L is the amount of resource capacity types).

$\boldsymbol{G}_T$ -Ordinal Relation between Tasks, which describes the dependencies between tasks.

(2) Platform: the carrier of functions. where,

$\boldsymbol{P}$ - Platform Set: $P=\{P_m\}$(m=1,2,…,K) (K is the amount of available platforms).

Attributes of Platform:

$\boldsymbol{PO}_m=(r_{m1},r_{m2},\ldots,r_{mL})$ - the resource capacity vector of P_m during its initialization;

$\boldsymbol{r}_{il}$ – the lth type of resource capacity of P_m during its initialization;

$\boldsymbol{P}_m$- the number of the l^{th} type of resource during its initialization (l=1,2,…,L, L is the amount of resource capacity types);

$\boldsymbol{v}_m$ - the maximum moving speed of P_m;

$\boldsymbol{La}_m=(X_m,Y_m)$ - the initial geographical coordinates of P_m.

2.2 Mathematical Description of MPoTP

The modeling process of MPoTP can be described by the following steps.

2.2.1 Variable Definition of the Matching Process

1. ω_{im} - Allocation Variable of Task-Platform: $\omega_{im}=\begin{cases}1 & P_m \text{ allocated to } T_i\\ 0 & \text{else}\end{cases}$

2. $\boldsymbol{x}_{ijm}$ - Transfer Variable of Platform between Tasks:

$$x_{ijm}=\begin{cases}1 & P_m \text{ allocated to } T_j \text{ after implementing } T_i\\ 0 & \text{else}\end{cases}$$

x_{ijm} doesn't exit if $i=j$, but in order to facilitate the mathematical description below, we set $x_{iim}=x_{jjm}=0$ (i,j =1,2,…,N; m=1,2,…,K) when $i = j$.

Suppose there is a virtual task T_0 which represents the start or end time of all tasks. When x_{ijm} $=x_{0jm}(j=1,2,\ldots,N)$, T_0 indicates that all tasks start, and when $x_{ijm} = x_{i0m}$(i = 1,2, ..., N) , T_0 indicates the termination of all tasks. At the beginning and end of the mission execution, all the platforms are on the virtual task T_0, and the transfer variable $x_{00m}=0(m= 1,2,\ldots,K)$.

3. $\boldsymbol{a}_{ij}$ - Ordinal Variables of Tasks： $a_{ij} = \begin{cases} 0 & T_i \text{ must finish before } T_j \text{ starts to run} \\ 1 & \text{else} \end{cases}$

4. $\boldsymbol{s}_i$ - Time Variable, the start time for running T_i.

5. $\boldsymbol{Y}$ - Mission Completion Time, the time of completing the last task of the mission.

6. $\boldsymbol{PO}'_m$ - Resource Capacity Vector of P_m, as defined in section 1.1. P_m 's available resource capacity vector during its initialization is $\boldsymbol{PO}_m=(r_{m1},r_{m2},\ldots,r_{mL})$. We assumes that during platforms implement their tasks, due to operator fatigue, loss of the platform and other reasons, its available resources capacity gradually decreases with the increase of the amount of tasks.

Therefore, after introducing platform in this problem model, the resources capacity update formula after implementing one task is:

$$r_{ml}^{renewed} = r_{ml} \cdot (1 - W_l \cdot \frac{\tilde{r}_{ml}}{r_{ml}}) \quad l = 1,2,\cdots,L \tag{1}$$

where W_l represents loss factors of the l-th resource capacity(we assumes that W_l is only related to characteristics of the l-th resource capacity, and unrelated to the platform which had capacity loss and unrelated to the task in the process of implementation); r_{ml} represents the amount of the l-th resources capacity that Pm owns before implementing certain tasks; $r_{ml}^{renewed}$ represents the amount of the l-th resources capacity that Pm owns after implementing certain tasks; $\tilde{r}_{ml}$ represents the amount of the l-th resources capacity that Pm actually uses during the process of implementing certain tasks.

From (1) we can see, after the platform implements one task, its resources capacity loss is related to the loss of capacity factor and to the amount of actually-used resources capacity.

After the update, the amount of resource capacity of P_m is $\boldsymbol{PO}_{m.}^{renewed} = (r_{m1}^{renewed}, r_{m2}^{renewed}, \ldots, r_{mL}^{renewed})$. We assume that during the execution of the mission, P_m 's changing resource capacity vector is $\boldsymbol{PO}'_m =(r'_{m1}, r'_{m2}, \ldots, r'_{mL})$.

2.2.2 The Matching Process of Constraint Analysis and Mathematical Description of MPoTP

Constraint Analysis of the Matching Process:

1. Allocation constraint of the platforms

If there is allocation relations between P_m (m=1,2,…,K) and T_i (i=1,2,…,N), ie, ω_{im} =1, P_m would be allocated to implement T_i after executing a certain task (including the virtual starting point of all the tasks task T_0), namely:

$$\sum_{j=0}^{N} x_{jim} - \omega_{im} = 0 \quad i = 1,2,\cdots,N; m = 1,2,\cdots,K \tag{2}$$

And when P_m(m=1,2,…,K) finishes executing T_i(i=1,2,…,N), it will be allocated to the next task (including all the tasks' terminate virtual task T_0), namely:

$$\sum_{j=0}^{N} x_{ijm} - \omega_{im} = 0 \quad i = 1,2,\cdots,N; m = 1,2,\cdots,K \tag{3}$$

All platforms are virtual tasks started from the beginning the mission, and are also in the termination of the virtual completion of the implementation of the mission task, namely:

$$\sum_{j=0}^{N} x_{0jm} = \sum_{i=0}^{N} x_{i0m} = 1 \quad m = 1,2,\cdots,K \tag{4}$$

2. The constraints of task start time

For $P_m(m=1,2,\ldots,K)$, there exists two relationship of between T_i and T_j, ie, $x_{ijm} = 0$ or $x_{ijm} = 1$. When Ti must be completed before Tj starts to be executed, ie, $a_{ij} = 0$, the task start time must satisfy:

$$s_j \geq s_i + t_i \tag{5}$$

Consider the executive relationship of P_m (m = 1,2, ..., K) between T_i and T_j, we can substitute (5) for:

$$s_j \geq s_i + t_i + x_{ijm}\frac{d_{ij}}{v_m} \quad i,j = 1,2,\cdots,N; m = 1,2,\cdots,K \tag{6}$$

where d_{ij} represents the space distance between T_i and T_j, that is,

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{7}$$

When the condition is not satisfied that T_i must be completed before starting the implementation, namely $a_{ij}=1$, the start time of the task must satisfy the following formula:

$$s_j \geq s_i + t_i - Y' \tag{8}$$

where Y' is upper bound of all the task completion time (generally set to a larger value) and $Y' \geq Y$. By combining (6) and (8), it can be obtained that when $a_{ij}=0$ or $a_{ij}=1$, the constraints of task start time need to meet the following formula

$$s_j \geq s_i + t_i + x_{ijm}\left(\frac{d_{ij}}{v_m} + a_{ij}Y'\right) - a_{ij}Y' \quad (i,j = 1,2,\cdots,N; m = 1,2,\cdots,K) \tag{9}$$

3. Resource Requirement Constraints of Tasks

Successful implementation of the resource capacity of Ti requires that in the implementation of this task, all the resource capacity vector of the platform and in every type of resource capacity is not less than the number required by the task corresponding to the type number of the resource capacity, namely:

$$\sum_{m=1}^{K} r'_{ml}\omega_{im} \geq R_{il} \quad i = 1,\cdots,N; l = 1,\cdots,L \tag{10}$$

4. The Constraints of Mission Completion Time

Y should not less than any of the task completion time, namely:

$$Y \geq s_i + t_i \quad i=1,2,\ldots,N \tag{11}$$

Combining the above constraints required by the matching process, the mathematical description of MPoTP is shown as equation *(12)* in the following page.

3. Problem Solution

According to the mathematical description of task-platform matching problem by equation *(12)*, we can see that the task-platform matching problem is a mixed binary linear scheduling problem, which has been proved to be a tough problem with non-deterministic polynomial time (Nondeterministic Polynomial, NP)[1].

$$
\begin{aligned}
&\min\ Y \\
&\begin{cases}
\sum_{j=0}^{N} x_{jim} - \omega_{im} = 0 \quad i = 1,2,\cdots,N; m = 1,2,\cdots,K \\
\sum_{j=0}^{N} x_{ijm} - \omega_{im} = 0 \quad i = 1,2,\cdots,N; m = 1,2,\cdots,K \\
\sum_{j=0}^{N} x_{0jm} = \sum_{i=0}^{N} x_{i0m} = 1 \quad m = 1,2,\cdots,K \\
s_j \geq s_i + t_i + x_{ijm}(\frac{d_{ij}}{v_m} + a_{ij}Y') - a_{ij}Y' \quad i,j = 1,2,\cdots,N; m = 1,2,\cdots,K \\
d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
Y \geq s_i + t_i \quad i = 1,2,\cdots,N \\
\sum_{m=1}^{K} r'_{ml}\omega_{im} \geq R_{il} \quad i = 1,\cdots,N; l = 1,\cdots,L \\
0 \leq Y \leq Y'; s_i \geq 0; x_{jim}, \omega_{im} \in \{0,1\}
\end{cases}
\end{aligned}
\tag{12}
$$

In this paper, we proposed a MPLDS + PWE -based method. This algorithm mainly consists of two steps, i.e., to obtain the task allocation order in the problem model by using MPLDS algorithm, to obtain the task completion time and task-platform matching scheduling under this order, and to improve the model's solution by using the PWE method. Key steps of the method include task selection in the MPLDS algorithm, platform or platform group selection in the MPLDS algorithm, and PWE-based method to improve the model solution, as will be described in detail in the following part of this section.

3. 1 Task Selection

Task selection depends on the connections in the task order diagram, i.e., to select the current task in the current set of T_{ready} (T_{ready} is the set of all previous tasks that have finished with the task of collection) based on the priority coefficient. For the calculation of task priority coefficients, three algorithms have been provided, i.e., the critical path (***CP***) algorithm, the hierarchical allocation (***LA***) algorithm, and the weighted length (***WL***) algorithm.

Of these three algorithms, the task with a priority factor ***WL*** algorithm formula is

$$
WL(i) = t_i + \max_{j \in OUT(i)} WL(j) + \frac{\sum_{j \in OUT(i)} WL(j)}{\max_{j \in OUT(i)} WL(j)} \tag{13}
$$

where $OUT(i)$ is the direct follow-up tasks set for the T_i in task order diagram, and $WL(i)$ is the priority coefficient of T_i obtained with the WL algorithm.

A transformation algorithm for ***WL*** is the weighted critical path ***(WCP)*** algorithm, the calculation formula of which is

$$
WCP(i) = CL(i) + \max_{j \in OUT(i)} CL(j) + \frac{\sum_{j \in OUT(i)} CL(j)}{\max_{j \in OUT(i)} CL(j)} \tag{14}
$$

where $CL(i)$ is the critical path length for T_i to the end of task in the task order diagram, and $WCP(i)$ is priority coefficient of T_i obtained from WCP algorithm.

The greater the priority coefficient of a task, the more preferable for the task to be selected.

3.2 PWE-based methods to improve the solution

The solution obtained from MPLDS algorithm (i.e., the mission completion time) may only be a suboptimal solution, because the task allocation order used in the MPLDS algorithm may only be approximately optimal[1]. Therefore, the PWE method can be used to improve the solution.

The basic idea of solution improvement in PWE-based methods is as follows. On the basis of task allocation orders and results from MPLDS algorithm, all possible task allocation orders are obtained by pair-wise exchanges between tasks, and related method (primarily the platform or platforms group allocation method for selected tasks) in MPLDS algorithm is then used to get solution under the above task allocation orders. Then the model solutions under these orders are received. Finally, the minimum result is chosen to realize the solution improvement.

PWE-based methods of improving the solution of the special process is as follows:

Step1: initialization, that is, the task allocation algorithm assuming MPLDS order $\{i_1,i_2,\ldots,i_N\}$ (the order of a sequence of 1-N), is assigned in the order of the task completion time for the next mission;

Step2: to make $n = 1$;

Step3: to make $m = n$;

Step4: to determine whether task i_n and task i_m can be exchanged in $\{i_1,i_2,\ldots,i_N\}$. If the answer is yes, go to Step5; otherwise go to Step6;

Step5: In $\{i_1,\ldots,i_{n-1},i_m,i_{n+1},\ldots,i_{m-1},i_n,i_{m+1},\ldots,i_N\}$, use MPLDS algorithm to solve the relevant method to get the mission completion time, If $Y'' < Y'$, $Y' = Y''$, turn to Step6;

Step6: $m=m+1$, if $m \le N$, turn to Step4; otherwise go to Step7;

Step7: $n=n+1$, if $n \le N-1$, turn to Step3; otherwise go to Step8;

Step8: The minimum output of the last mission completion time $Y=Y'$ and the corresponding task - platform matching program..

The method of determining whether the tasks in i_m $(n<m)$ in the distribution of tasks in order $\{i_1,i_2,\ldots,i_N\}$ can be exchanged is to determine whether the following two conditions are satisfied.

1. $IN(i_m) \subseteq \{i_1,i_2,\ldots,i_{n-1}\}$; 2. $OUT(i_n) \subseteq \{i_{m+1},i_{m+2},\ldots,i_N\}$.

where $IN(i)$ leads directly to the task set of Ti. If satisfied, they can be exchange; otherwise, they can't be exchanged.

4. Analysis of Cases

The campaign of joint operations in Ref.[9] was designated as a case to validate the proposed MPLDS + PWE -based task-matching platform. The task order diagram of the mission is shown in Fig.1(a). The required number of performed operation tasks is $N = 18$, and the number of platforms available is $K = 20$. The attributes of tasks and platforms data listed in Ref. [9].

Suppose the initial position of all platforms is La_m=(85,40) and the loss factor of resource capacity is the vector $\boldsymbol{W}$=[0.1,0.1,0.1,0.1, 0.05,0.05,0.1,0] . We choose $\boldsymbol{WL}$ algorithm as the method of calculating the task priority coefficient in the simulation experiment. Under the condition set in the above experiment, when the MPLDS algorithm is applied, the result of task-platform matching solution is with Mission completion time Y = 166.4101 and the corresponding tasks distribution sequence being {5,6,1,3,4, 17, 2,7,18,8,9,13,11,12,10,15,14,16}. The task-platform matching solutions obtained by the MPLDS + PWE method is with mission completion time Y = 153.0722 and the corresponding tasks distribution sequence being {1,6,5,3,4,17,2,7,18 , 8,9,13,11,12, 10,15,14,16}. According to the simulated analytical results shown in Fig.1(b), the proposed MPLDS + PWE-based task-platform matching algorithm has improved

the model solution by considering a variety of task allocation orders, demonstrating the superiority of this method. This method provides new solution to the task-platform matching problem.

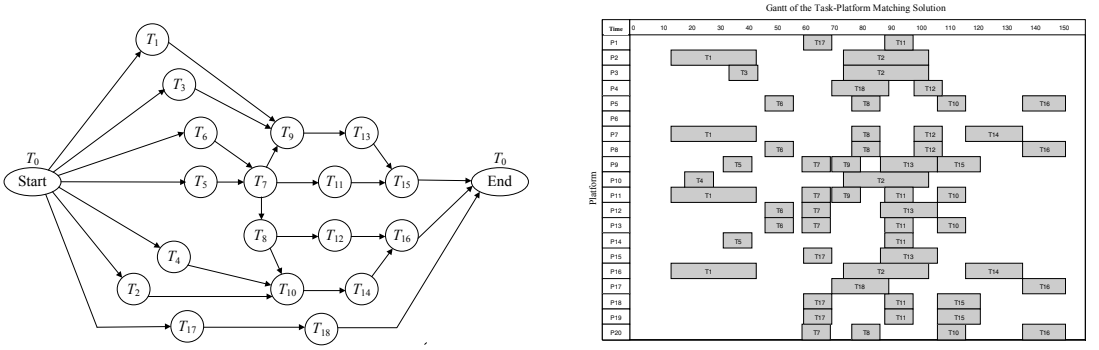

Fig. 1. (a) Mission task sequence diagram; (b) MPLDS + PWE method of getting the task - platform matching program, the mission deadline: 153.0722.

5. Conclusions

By combining MPLDS and PWE, this paper proposes a new task-platform matching method, in which the MPLDS is utilized to get the model solution under task allocation order and the PWE is used to improve the solution. The major work includes:1. Improving the mathematical model of the task-platform matching through the introduction of loss factor of resource capacity; 2. Improving the solution through the introduction of PWE method based on the model results from MPLDS algorithm; 3. Analyzing and validating the superiority of the MPLDS + PWE matching method from a case which designated as joint-operation battle.

References

[1] Levchuk GM, Levchuk YN, Luo J, et al. Normative Design of Organizations-Part I: Mission Planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 2002; 32(3): 346-359.

[2] Levchuk GM, Levchuk YN, Luo J,, et al. Normative Design of Organizations-Part II: Organizational Structure. *IEEE Transactions on Systems, Man, and Cybernetics*, 2002, 32(3): 360-375.

[3] Bui H, Han X, Mandal S, et al. Optimization-based Decision Support Algorithms for a Team-in-the-Loop Planning Experiment, In: *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, 2009.

[4] Mandal S, Han X, Pappipati KR, et al. Agent-Based Distributed Framework for Collaborative Planning. In: Proceedings of the 2010 IEEE International Conference on Aerospace conference, Big Sky, MT, 2010.

[5] Lu YL, Yang DS, Liu Z, et al. Research on Algorithm of Resource Allocating in Joint Operation, *Fire Control and Command Control*, 2006; 31:12-16.

[6] Liu HF, Yang DS, Liu Z, et al. Research on Methodology of Resource Tailoring and Clustering in Battlefields. *Journal of System Simulation*, 2006, 18:2598-2603.

[7] Yang DS, Zhang WM, Liu Z,, et al. Research on Mathematical Description and Solving Algorithms of Tasks Scheduling for Campaign. *Systems Engineering Theory & Practice*, 2006; 6:26-34.

[8] Yang DS, Peng XH, Zhang WM, et al. Design of C2 Organizational Structure: Designing Relations among Platforms and Tasks. *Fire Control and Command Control*, 2006; 31:9-13.

[9] Yu F, Tu F, Pappipati KR. Integration of a Holonic Organizational Control Architecture and Multiobjective Evolutionary Algorithm for Flexible Distributed Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics*, 2008; 38: 1001-1017.