A Distributed Method for Dynamic Adjusting of the Task-Platform Relation

Jieyong Zhang, Songhua Huang, Peng Sun, Guanyu Chen* and Ruijie Wu

ABSTRACT

Dynamic adjusting of the task-platform relation is a hotspot problem in military operational command field during the wartime, and a distributed method for it is present in this paper. Firstly, the constituent part of the distributed military strength organization is represented, and then a distributed coordination framework of the dynamic adjusting of the task-platform relation is designed. This coordination framework has two parts: the intra-module and the external cooperation module. The intra-module combines with an N-best algorithm and a tradeoff strategy to achieve the allocation once again in the inner of decision maker; and the external cooperation module achieves the collaboration between decision makers when task accuracy is less than the desired accuracy. Finally this method for dynamic adjusting of the task-platform relation is proved to be effective through simulation experiment, and the applicability of this method in different situations is also discussed.
INTRODUCTION

Scientific allocation of platforms is the core issue of task planning research [1], [2]. This problem can be described as follows [3], [4]: The goal of the battle is to complete the mission, and tasks are the sub-goals of the mission. There are time constraints between tasks, and a task is an activity that requires a set of relevant resources to be processed; a platform is a physical entity with specified resource capabilities to execute the resulting plan. The problem of task planning is to allocate platforms to tasks substantially.

Reasonable task planning is of great significance for the battle command. To establish mathematical model by minimizing the completion time of the mission, Levchuk proposed a multidimensional dynamic list scheduling (MDLS) algorithm in three-phrase normative design of organization [5]-[7]. Zhang formulated a cycle MDLS algorithm to solve the allocated problem with constraints of the completion time [8]. In general, there is a great environmental uncertainty in the warfare, such as the processing time of the task, the depletion of the platform and so on. Considering this uncertainty, Mo and Zhang put forward a new mathematical model maximizing the probability of mission completion [9], [10].

The above methods belong to centralized task planning. To cope with the uncertain and dynamic mission environment [11], [12], the military strength organization pattern has varied from a centralized control to a more flexible distributed delayering control. Due to the limited of decision-making ability, command and control is centralized in large-scale organization. Distributed decision-making reduces the work load of decision-makers, and increases the scheduling efficiency of platforms which means decision-makers having more autonomy. An important research problem is how to realize the maximization of decision benefits. Park put forward a distributed auction algorithm under different communication structure [13]. On account of large-scale nonlinear planning problem, Han proposed an optimization-based distributed planning algorithm based on blackboard collaborative framework [14].

In this paper, our work focuses on the distributed and dynamic planning adjusting. The paper is organized as follows. We introduce the key concepts describing the distributed planning problem and mathematical formulations in Section 2. We proposed a distributed collaborative framework to realize the planning dynamic adjusting in Section 3. In Section 4, we present the simulation results and empirical results from the experiment. This paper concludes with summary and future directions for research in Section 5.

PLANNING ENVIRONMENT

The key entities in military organizations are as follows.
Definition 1 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, and each platform provides resource capabilities.

Definition 2 Task: A task is an activity that requires a set of relevant resources to be processed. A task $i$ ($i=1, 2 \ldots I$) is characterized by the following properties. $I$ is the number of tasks.

1) $t_{si}$: the start time of task $i$.
2) $t_{pi}$: the processing time of task $i$.
3) $Ril = (Ril, \ldots, RiL)$: resource requirement vector, where $Ril$ is the number of units of resource type $l (l=1, \ldots, L)$ required for successful processing of task $i$.
4) $\rho_i$: priority of task $i$.

Definition 3 Platform: A platform is a physical entity with specified resource capabilities that can be used to process tasks. Each platform type $j (j=1, \ldots, J)$ is defined by its properties as follows.

1) $r_j = (r_{jl}, \ldots, r_{jL})$: resource capability vector, where $r_{jl}$ is the number of units of resource type $l$ possessed by platform type $j$.
2) $n_j$: the number of available platforms for platform type $j$.

Definition 4 DMs: DMs are intelligent entities who utilize knowledge and wisdom of the planning domain to select an appropriate plan among a range of alternative scenarios. Each DM is defined by its properties as follows.

1) $S_{DM}^T = \{T_1, \ldots, T_{NT}\} = \text{the set of tasks owned by DMs, } NT \text{ is the number of tasks}.$
2) $S_{DM}^P = \{P_1, \ldots, P_{NP}\} = \text{the set of platforms dominated by DMs, } NP \text{ is the number of platforms}.$

DISTRIBUTED PLANNING PROBLEM

The planning problem is one of matching platforms’ capabilities to the tasks’ requirements over the resource categories as shown in Figure 1. However, each DM $p$ has only a limited number of platforms $\phi(p)$ and a set of assigned tasks $\phi(p)$ for which he/she is responsible. We define the allocation relationship between tasks and platforms. The platform allocation of task $i$ is shown as follows.

$$y_i = [y_{i1}, y_{i2}, \ldots, y_{iJ}]^T$$

$y_i \in \mathbb{Z}_+ \quad y_i \geq 0$
Where $y_{ij}$ indicates how many platforms of type $j$ are allocated to task $i$ and $Z$ is the set of integers.

![Figure 1. Distributed planning problem.](image)

**Individual Task Model**

To solve the problem of matching platforms’ capabilities to the individual task requirements, we introduce the concept of accuracy to evaluate how well the platforms’ capabilities match the tasks’ requirement. Each resource category $l$ has a satisfaction degree $z_{il}$ and we compute the ratio of assigned resources to the require resources for each resource category $l$. When the ratio is greater than 1 which means this kind of resource is enough, we set $z_{il} = 1$; Otherwise, we set $z_{il} = \text{radio}$. We seek a task execution accuracy metric $z_i$ as follows.

$$z_i = \min(1, \frac{\sum_{j=1}^{I} r_{ij} y_{ij}}{R_{il}})$$

$$z_i = [z_{i1}, ..., z_{iL}]$$

$z_{il}$ is the satisfaction degree of resource category $l$. To define the accuracy well, we adapt the geometric mean as our accuracy metric. On the one hand, the accuracy is 0 if missing any required resource category; on the other hand, increasing the minimum ratio will increase the accuracy the most.

$$Acc(i) = \left( \prod_{l \in r(i)} z_{il} \right)^{\frac{1}{|r(i)|}}$$

Here, $r(i)$ denotes the set of resource categories required by task $i$. $\|r(i)\|$ is the cardinality of $r(i)$.

The platform allocation problem for a single task $i$ can be formulated as
\[
\max \ Acc(i)
\]
\[s.t. \ 0 \leq \sum_{j=1}^{n} y_{ij} \leq D \]
\[y_{ij} \in \mathbb{Z}, 0 \leq y_{ij} \leq n_j, \forall j\]

\(D\) is the upper bound on the number of platforms that can be allocated to an individual task; \(n_j\) is the available number of platform \(j\). The individual task planning problem in (3) is a mixed integer nonlinear programming problem (MINLP), and this problem can be transformed into a integer linear programming problem \([14]\), so we rewrite (3) as

\[
\max \ \sum_{t \in \tau(i)} z_t
\]
\[s.t. \ z_t \leq \sum_{j=1}^{n} r_{jt} y_{ij}, \ z_t \leq 1\]
\[0 \leq \sum_{j=1}^{n} y_{ij} \leq D \]
\[y_{ij} \in \mathbb{Z}, 0 \leq y_{ij} \leq n_j, \forall j\]

**Platform Pricing Model**

Distributed task planning dynamic adjusting problem can be described like this: When the environment of battlefield gets changed, a DM can’t accomplish tasks only using his own platforms. So we need to adjust the platform allocation plan to adapt the exchange variation of environment. To get tasks executed well, a DM need borrow some platforms from other DMs with a price. So, we write down the platform pricing model to evaluate this price. There is a matching degree \(v_{il}\) of each resource category \(l\) between platform \(j\) and task \(i\). We compute the radio of resources owned by platform \(j\) to the require resources for each resource category \(l\). When the radio is greater than 1 which means this kind of resource is matched completely, we set \(v_{il} = 1\); otherwise, we set \(v_{il} = \text{radio}\). We seek a platform matching degree metric \(v_i\) as follows

\[v_i = \min(l, y_{ij} / R_{il})\]
\[v_i = [v_{i1}, v_{i2}, \ldots, v_{il}]\]

Here, \(V_{il}\) denotes the matching degree between platform \(j\) and task \(i\). Then, we compute the price of platform \(j\) to task \(i\) as follows.
\[ E_j(i) = \frac{\sum_{i \in \text{set}(i)} V_{ij} \cdot \text{num}(j)}{\| r(i) \|} \quad (6) \]

\( \text{num}(j) \) denotes the number of platform \( j \). Platform \( j \) has different price to different task. Finally, we compute the weighted average of them to get the final platform’s price.

\[ \text{price}(j) = \frac{\sum_{i=1}^{NT} \omega_i E_j(i)}{\omega_1 + \omega_2 + ... + \omega_{NT}} \quad (7) \]

\( \omega_i \) is the weight of task \( i \), and \( \text{price}(j) \) is the price of platform \( j \).

**Optimizing Target**

ADM must pay the price if he/she need borrow some platforms from other DMs, and formulation (7) measures this price. The set of platforms borrowed from other DMs is \( S_{\text{borrow}} = \{P_1, ..., P_N\} \); \( N \) is the number of platforms. The final benefit of a DM equals to the sum of all tasks’ accuracy subtracting the price of platforms that borrowed from other DMs. Revenue function of a DM is as follows.

\[ W = \sum_{i,j \in S_{\text{borrow}}} \omega_i \text{Acc}(i) - \sum_{j=1}^{N} \text{price}(j) \quad (8) \]

In distributed environment, global profit is the sum of all DMs’ profits. So the final target of this model is to maximize the global profit.

\[ \max V = \sum_{m=1}^{M} W(m) = \sum_{m=1}^{M} \{ \sum_{i,j \in S_{\text{borrow}_m}} \omega_i \text{Acc}(i) - \sum_{j=1}^{N} \text{price}(j) \} \quad (9) \]

**COLLABORATION FRAMEWORK**

Prewar planning has formed the military strength organization, and a DM can’t accomplish tasks only relay on his/her own platforms when the parameters of environment, task requirement and platform ability get changed. To adapt to these changes quickly, this paper put forward a distributed collaboration framework, and the whole framework includes two components as illustrated in Figure 2.
The first part is an inner platform allocation module. When these parameters have changed, a DM tries to adjust the allocation plan by using his/her own platforms to meet the requirements of tasks.

The second part is an external collaboration module. If the accuracy of a task is less than the desired accuracy threshold via inner module, the DM send out a request to other DMs to borrow necessary platforms to improve the accuracy to the desired accuracy threshold with least price.

**Inner Allocation Module**

This module is a platform allocation algorithm including a N-best algorithm and a tradeoff strategy.

**STEP 1** The priorities of tasks are calculated by using a weight length algorithm [15]. Once the priorities are calculated, the task list is sorted in decreasing order of priorities and we select the task with highest priority to generate N-best allocation plans.

**STEP 2** Compare the task accuracy with the accuracy threshold. If the best allocation plan provides a greater accuracy than the threshold, go to step 3; otherwise, go to external collaboration module.

**STEP 3** Determine the final allocation plan by using a tradeoff strategy.
**N-best ALGORITHM**

N-best algorithm [16] can generate N plans for a task, and it can provide DMs with more choices and help them decide which plan is better.

The basic process about N-best algorithm is illustrated as follows.

- **STEP 1** Partition the problem into several sub-problems with constraints according to the current best answer [2].
- **STEP 2** Solve for each sub-problem and set the solution to candidate set.
- **STEP 3** Select the solution from the candidate set which can maximize the target function in (2), and set it as the next best solution.

**TRADEOFF STRATEGY**

High priority tasks are usually allocated platforms preferentially which can result in poor accuracy performance on later tasks. This paper put forward a tradeoff strategy to solve this problem. We apply the tradeoff strategy to balance the platform allocation plans among tasks. For each task, tradeoff strategy generates N allocation plans and assigns best available platforms for the remaining tasks using a greedy strategy. Finally, we select the most suitable plan from N plans by calculating which plan maximizes the mission objective in (10).

\[
B(k) = Acc(i,k) + \sum_{i=1}^{f} Acc(i,k) \\
\]

\[
k^* = \arg \max \{ B(k) | k = 1, \ldots, N \}
\]

Here, \(k\) donates \(k^{th}\) \((1 \leq k \leq N, k \in \mathbb{Z})\) allocation plan, \(B(k)\) is the objective function of \(k^{th}\) allocation plan, and allocation plan \(k^*\) donates the best plan which can bring biggest global objective.

**External Collaboration Module**

However, the accuracies of some tasks are still less than the desired accuracy threshold via inner module’s adjusting. In this situation, DMs need ask for necessary platforms from others to increase their accuracy.

In 2.2, we have introduced the concept of platform price. Obviously, a DM wants to pay the least price to borrow platforms from others to get his/her accuracy greater than the desired accuracy threshold. Here, the platform allocation plan of task i is \(\{y_{i1}, \ldots, y_{in}\}\), and the borrowed platforms set is \(\{y_{i1}, \ldots, y_{im}\}\) \((1 \leq N \leq J)\), \(D\) denotes the upper bound on the number of platforms that can be allocated to a single task. \(\overline{n_j}\) is the number of available platforms j. This platform borrowing mechanism is as follows.
To realize the collaboration between DMs effectively, we introduce a coordinator as an information pool. First, a DM sends his request message to the information pool in a broadcast way. Other DMs can receive this request and make a decision about which platforms can be lent. Each DM provides platform prices using the platform pricing model. Finally, each DM gets his/her borrowed platforms via (11).

**SIMULATION**

**Scenario Description**

The task and platform data is abstracted from the MOC experiment [17]. There are I=11 tasks, and J=14 different types of platforms. The whole simulation is running on a computer with Intel(R) Core i5-2450M CPU 2.5GHz.

**Result Analysis**

Experiment 1. In the first experiment, we test the distributed collaboration framework on MOC scenario. We set $\omega_i = 1$, and the accuracy threshold is set at 75%. We find that the accuracies of T3, T9 cannot reach the desired accuracy by using each individual DM’s platforms before coordination. The accuracies of T3, T9 are improved after coordination; T3 from 0 to 95.32%, T9 from 53.13% to 79.37%. The average accuracy is improved from 84.47% to 94.13%, and the average profit of DMs is improved from 3.097 to 3.253. However, there is an exception that the accuracy of T11 is decreased from 1 to 84.68%. This inflects that the distributed collaboration framework is an optimizing process balancing global accuracies.

Experiment 2. We conducted 200 experiments to evaluate the statistical property of accuracies. There are four cases including task requirement changed, platform capability changed, platform destroyed, and new task added. Figure 3 indicates that the convergence and average of case 1, case 2 are better.
Experiment 3 Prewar platform allocation is a centralized planning problem, and genetic algorithm (GA) is an effective way to solve this large-scale nonlinear problem. In this section, we conducted five group experiments to compare our distributed algorithm with centralized GA algorithm to verify the feasibility of our distributed framework, and the average task accuracy is shown as Figure 4. The first group is getting task requirement changed; the second group is getting platform capability changed; the third group is getting one platform destroyed; the fourth group is getting 4 platforms destroyed; the last group is adding a new complex task. The former three groups belong to soft changes and the latter two groups are violent changes.

The average accuracy of the first group is increased from 86.4% to 89.8%; the second group from 95.5% to 97.5%; the third group from 93.1% to 98%; the fourth group from 75% to 90.5%; the last group from 86% to 97%. In general, the average
accuracy of centralized planning is better than that of distributed planning. However, there are still obvious disadvantages for centralized planning. On the one hand, centralized planning can alter the command and control structure which is undesired to commanders; on the other hand, centralized planning has a greater computation complexity than distributed way and this cannot meet the requirement of timeliness in task planning dynamic adjusting.

CONCLUSIONS

In this paper, we have introduced a platform pricing model and a distributed collaboration framework to solve the task planning dynamic adjusting problem during the war. One is that the average accuracy of tasks and the average profit of DMs are increased with our distributed collaboration framework; the other is that our adjusting plans have a strong adaptability on general changes occurring during the war using our algorithm. In our future work, we plan to adopt our distributed model to study distributed command and control problems, and to optimize the structure of the command and control.

ACKNOWLEDGEMENTS

Supported by Natural Science Foundation of Shanxi Province of China (Grant No. 2017JM6062) and the key laboratory fund for Equipment pre-research (Grant No. 61421010201).

REFERENCES


