

基于粒度计算和遗传算法的 C2 组织结构设计方法*

修保新** 张维明 刘忠 阳东升

国防科技大学信息系统与管理学院 C²ISR 技术国防科技重点实验室, 长沙 410073

摘要 介绍了指挥控制(Command and Control, C2)组织及其组织结构, 基于粒度计算提出了一种新的组织结构设计方法, 设计过程包括3个阶段: 任务集和平台集粒化阶段, 粒内规划阶段和决策分层阶段; 给出了基于遗传算法的平台集粒化方法以及第二个阶段的规划策略及方法. 案例分析和对比试验表明该方法具有较好的性能.

关键词 指挥控制 组织结构 粒度计算 遗传算法

信息和网络技术的发展使得现代组织正面临前所未有的挑战. 由于组织成员个体能力的有限性以及组织使命的复杂性和组织面临环境的不确定性, 未来组织的竞争不再仅限于组织物理资源、组织权威以及组织成员个体能力, 而更多的是通过适时快速地优化组合组织有限能力个体、组织资源以及信息, 以组织成员个体之间的协作、物理资源配置以及组织最优策略的使用来实现组织目标. 这种技术进步对现代高技术战争战场环境中兵力组织的冲击表现得尤为突出. 如何建立起信息化战争条件下高效的指挥体系, 形成与使命环境适应的兵力组织是目前指挥自动化领域的难点之一.

文献[1]建立了指挥控制(command and control, C2)组织的分层描述并提出了有效测度 C2 组织的概念, 在此基础上提出了 C2 组织的设计流程和方法, 其中, C2 组织设计包括过程设计和结构设计. 所谓过程设计是指建立执行组织使命目标的有效行动过程; 结构设计是指设计与环境相适应、与行动过程相匹配的组织结构. 这是两个相对独立的问题. 近几年通过对团队决策的研究表明, 组织结构的优化设计完全在于实际的任务结构和组织运作

环境的关键属性. 这种结论导致了应用系统工程技术来设计人类组织. 以 Levchuk 为代表的研究团队发展了使命模型并提出了组织结构的 3 阶段设计方法^[2,3](简称为 3 阶段方法), 该方法将组织结构问题分解为三个迭代的子优化问题, 依次完成任务到平台的优化配置, 平台到决策实体的分配以及确立决策实体间的层次关系和协作关系. 这种方法虽然克服了问题的计算复杂性, 但仍然存在两个主要的问题: 一是随着问题规模的增大, 第一个阶段的列表规划方法的性能将大大下降; 二是第一个阶段的优化目标会潜在影响后续两个阶段的优化设计, 即影响组织的性能^[4]. 文献[4, 5]对 3 阶段方法进行了改进, 主要是在第一阶段采用改进的列表规划方法优化使命完成时间, 但同样存在原有方法的问题.

如何平衡组织结构问题中的两个重要指标: 使命完成时间和组织结构与使命的适应性测度, 仍然是需要进一步研究的问题. 本文将通过引入信息处理和人工智能领域的新兴研究领域粒度计算, 提出新的解决办法.

2006-06-23 收稿, 2006-09-25 收修改稿

* 国家自然科学基金资助项目(批准号: 70271004, 60504036)

** E-mail: bxxiu@fuzzy@hotmail.com

1 C2 组织和 C2 组织结构

根据组织权变理论, 没有公认一致的普遍适用的最优组织结构和形式, 只有在某一具体使命环境下, 适用于这一具体条件的最优组织^[6]. 建立 C2 组织的实体描述和任务模型, 提出结构与使命的适应性测度是组织结构优化设计的前提.

1.1 C2 组织的实体与任务

Levchuk 等^[2]根据兵力组织的特点提出了决策实体、平台和任务的概念.

定义 1 平台是物理资源载体, 拥有资源能力, 执行特定的功能. 平台集记为 P , 对每一个平台 $P_k (k=1, 2, \dots, K; K$ 是平台数量), 定义它的最大速度 v_k 和它的能力向量 $R(P_k) = [r_{k1}, r_{k2}, \dots, r_{kl}]$, 其中 r_{kl} 规定了平台拥有的功能类型 l 的数量 ($l=1, 2, \dots, L; L$ 是功能类型的数量).

定义 2 任务是一种由一个平台或者一组平台执行的行动, 它由相关功能完成. 任务集记为 T , 通过下列基本属性定义每一个任务 $T_i (i=1, 2, \dots, N; N$ 是任务数量):

- (i) 估计的处理时间 t_i ;
- (ii) 地理限制向量 (例如, 位置 (x_i, y_i) 限定了任务 T_i 和 T_j 的距离 d_{ij});
- (iii) 资源需求向量 $R(T_i) = [R_{i1}, R_{i2}, \dots, R_{il}]$, 这里 R_{il} 表示成功的处理任务 T_i 所需的功能类型 l 的能力.

定义 3 任务图定性描述任务间的依赖关系, 如任务的优先顺序、数据流程以及任务间的输入输出关系等. 任务间优先顺序定义如下:

$$P(i, j) = \begin{cases} 1, & \text{如果任务 } T_i \text{ 必须在任务 } T_j \text{ 执行前完成} \\ 0, & \text{否则} \end{cases} \quad (1)$$

任务集和有向非循环的任务图共同描绘了执行使命 M 的任务计划.

定义 4 决策实体是信息处理的实体. 其能力可控制必要的平台资源来执行任务, 如果任务的执行不会对其能力造成损害. 根据需要可定义决策实体的知识和能力. 决策实体可以是作战指挥中的指挥官, 或者企业组织中的经理人员. 组织中可用的

决策实体集合记为 $DM = \{DM_1, DM_2, \dots, DM_M\}$ (M 是决策实体数量).

组织包括一个决策实体集, 平台到决策实体的分配和决策实体间的协作结构. 组织用来完成执行使命的任务计划, 目标是 minimized 决策实体的内部工作负载和外部工作负载, 以及 minimized 整个使命的完成时间^[2].

1.2 C2 组织结构

组织结构是组织实体之间的 C2 结构关系的体现, 结构决定了组织内的协作, 而这些协作在很大程度上决定了过程执行的好坏. C2 组织实体间结构关系包括: 平台-任务分配关系 R_{P-T} 、决策实体-平台控制关系 R_{DM-P} 、决策实体-任务执行关系 R_{DM-T} 以及决策实体间的协作交流关系 R_{DM-DM} , 即 C2 组织结构可以表示为 $O = (R_{P-T}, R_{DM-P}, R_{DM-T}, R_{DM-DM})$ ^[1].

C2 组织结构设计参数包括任务-决策实体分配矩阵 TD , 平台-决策实体分配矩阵 PD , 任务-平台分配矩阵 TP 和决策实体之间的协作交流矩阵 DD , 分别定义如下:

$$TD(i, m) = \begin{cases} 1, & \text{如果任务 } T_i \text{ 分配到 } DM_m \\ 0, & \text{否则.} \end{cases} \quad (2)$$

$$PD(k, m) = \begin{cases} 1, & \text{如果平台 } P_k \text{ 分配到 } DM_m \\ 0, & \text{否则.} \end{cases} \quad (3)$$

$$TP(i, k) = \begin{cases} 1, & \text{如果平台 } P_k \text{ 分配到任务 } T_i \\ 0, & \text{否则.} \end{cases} \quad (4)$$

$$DD(m, n) = \begin{cases} 1, & \text{如果从 } DM_m \text{ 到 } DM_n \text{ 存在链接} \\ 0, & \text{否则.} \end{cases} \quad (5)$$

一个合理的任务-平台分配矩阵 TP 需要满足下面的约束条件,

$$\sum_{k=1}^K TP(i, k) \cdot r_{kl} \geq R_{il} \quad (6)$$

其中 $i=1, 2, \dots, N; k=1, 2, \dots, K; m=1, 2, \dots, M; l=1, 2, \dots, L$.

1.3 组织结构与使命的适应性测度

组织结构与使命环境和过程的匹配或适应是对组织结构完成使命的性能度量, 是组织最佳运

作的关键^[7]。经过大量的实验验证^[8]结构匹配主要的测度参数包括：工作负载、交流和个体间的依赖关系，C2组织结构的四种关系确定了这些测度参数。

考虑组织完成使命的两个主要的测度参数：决策实体的内部工作负载和外部协作负载^[3]。

DM_m 的内部工作负载 $I(m)$ 是分配到该决策实体的平台的累积负载，

$$I(m) = \sum_{k=1}^K PD(k, m) \quad (7)$$

DM_m 的外部协作负载 $E(m)$ 是该决策实体与其他决策实体的协作的和，

$$E(m) = \sum_{n=1, n \neq m}^M D(m, n) \quad (8)$$

这里， $D(m, n)$ 是决策实体 DM_m 和 DM_n 间的协作量，

$$D(m, n) = \sum_{i=1}^N TD(i, m) \cdot TD(i, n) \quad (9)$$

DM_m 的工作负载 $W(m)$ 是它的内部工作负载和外部协作负载的加权和^[3]，

$$W(m) = W^I \cdot I(m) + W^E \cdot E(m) \quad (10)$$

其中 W^I 和 W^E 是权重。

定义组织结构 O 执行使命 M 的适应性测度为各决策实体工作负载的均方根(RMS)：

$$W_{RMS}(O, M) = \sqrt{\frac{1}{M} \sum_{m=1}^M W^2(m)} \quad (11)$$

该度量同时最小化团队中各成员负载的均值和方差^[9]。

2 组织结构设计与粒度计算

近年来，Zadeh^[10]讨论的粒度计算理论已经成为计算科学、逻辑、人工智能、哲学和其他领域的研究热点之一。所谓粒是指一些元素的集合，这些元素具有不可分辨性、相似性或函数和功能

的内聚性等等^[10]。Zadeh认为，粒度计算是模糊信息粒度理论、粗糙集理论和区间计算理论的超集，是粒度数学的子集，它像一把大伞覆盖了所有有关粒度的理论、方法论、技术和工具的研究。在国内，张钹和张铃在研究问题求解时，独立地提出了商空间理论^[11]和模糊商空间理论^[12]，也属于粒度计算的范畴。粒度计算的基本问题主要有两个方面，一是粒度的构造问题，二是怎样用粒度进行计算。前者研究粒度的形成、表示和语义解释，后者讨论怎样利用粒度去求解问题。虽然粒度计算在理论与应用方法上还存在许多尚待解决的问题，但已经在许多领域，如医疗诊断专家系统、图象处理、数据挖掘、模式识别与智能控制、问题求解中的启发式搜索、路径规划和时间规划等领域取得了很好的应用^[13]。粒度计算的目的是尽量找到计算复杂度最小的足够满意的可行近似解，因此，可以认为粒度计算是降低计算复杂度的有效工具之一。如何在问题求解时选择恰当的粒度层次，以使求解效果达到最佳，这是粒度计算的一个关键内容之一。组织结构问题由于涉及到大量的平台和任务，是典型的复杂问题求解，可以考虑通过引入问题论域的粒度计算来简化问题的求解。

2.1 组织结构中的粒度

给定论域 U 和 U 上的关系 $R: U \rightarrow 2^U, \Rightarrow U = \bigcup_{i \in \tau} G_i$ ，则称每一个 G_i 为一个信息粒， $\{G_i\}_{i \in \tau}$ 是论域 U 的一种粒度。其中， 2^U 表示论域 U 的幂集，关系 R 可以代表等价关系、不可分辨关系、功能关系、函数关系等等。

如果 $\forall i, j \in \tau, i \neq j \Rightarrow G_i \cap G_j = \phi$ ，则称 $\{G_i\}_{i \in \tau}$ 是论域的粒度划分；如果 $\exists i, j \in \tau, i \neq j \Rightarrow G_i \cap G_j \neq \phi$ ，则称 $\{G_i\}_{i \in \tau}$ 是论域的粒度覆盖。

组织结构设计中的问题论域是平台集和任务集。根据资源和任务的特点，可以将平台集和任务集进行粒化，每一个平台粒具有某种功能聚合性，每一个任务粒具有某种功能相似性，由于一个平台只能属于一个决策实体，一个任务可能由多个决策实体协作完成，因此粒化平台集得到的是粒度划分，粒化任务集得到的是粒度覆盖。通过粒化使一个平台粒能够有效的处理一个任务粒，实际上是在

新的粒度层次上求解原问题, 简化了问题求解的复杂度.

2.2 基于粒度计算的 C2 组织结构设计方法

当建立复杂的使命模型并设计相应的组织时, 不同的使命维度以及模型粒度决定了设计过程的复杂程度. 基于粒度计算的组织结构设计方法可以克服这种计算复杂性. 设计过程包括了三个阶段, 将组织结构设计问题分成了三个不同的优化子问题.

阶段 I (任务集和平台集粒化): 分别将平台集和任务集粒化成 M 个信息粒, 建立平台粒到任务粒的对应关系以满足每一个任务粒的资源需求. 这个阶段决定了每个决策实体拥有哪些平台和需要执行哪些任务. 目标是最小化组织结构与使命的适应性测度.

阶段 II (粒内规划): 这个阶段决定了平台到任务的具体配置, 目标是最小化使命完成的总时间. 需要考虑任务序列限制, 同步延迟, 任务资源需求, 资源能力和地理位置等约束. 通过已有平台粒到任务粒的分配, 这一阶段将原规划问题分解为一些独立的具有少量平台和任务的子规划问题.

阶段 III (决策分层): 这一阶段通过指定通讯结构和决策分层来优化决策实体的责任分配和决策实体间的控制协作, 同时平衡所有决策实体的控制工作负载.

下面着重讨论阶段 I 的粒化方法和阶段 II 的规划方法, 阶段 III 可以采用最大跨度树算法进行求解^[3], 本文将不再进行分析.

3 阶段 I: 任务集和平台集粒化

任务集和平台集的粒化可以采用两种策略, 一种是先得到平台集的粒度, 然后根据平台集粒化结果进行任务集的粒化; 另一种是先对任务集进行粒化, 然后根据任务集粒度进行平台集的粒化. 本文采用第一种策略.

3.1 任务集粒化

假设已给定平台集粒度划分 $\{G_i\}_{i \in \tau}$, $\tau = \{1, 2, \dots, M\}$, 并将平台粒 G_i 分配给决策实体 DM_i . 通过决定任务到决策实体的分配来完成

任务集的粒化, 同时可以得到任务粒和平台粒的对应关系, 目标是最小化组织结构与使命的适应性测度.

任务集记为 T , 令 $T^0 \subset T$ 表示可以由某一个平台粒单独完成的任务所组成的集合; 则 $T^l = T/T^0$ 表示需要由多个平台粒共同完成的任务所组成的集合.

3.1.1 T^l 到决策实体的分配 通过最小化组织结构与使命的适应性测度来决定任务子集 T^l 到决策实体的分配, 问题可表示如下:

$$\min W_{RMS}(O, M) = \sqrt{\frac{1}{M} \sum_{m=1}^M W^2(m)} \quad (12)$$

$$\begin{cases} \sum_{m=1}^M TD(i, m) \geq 2, T_i \in T^l; \\ \sum_{m=1}^M TD(i, m) \cdot \sum_{k=1}^K PD(k, m) \cdot r_k \geq R_i, \\ T_i \in T^l; l = 1, 2, \dots, L \end{cases}$$

问题求解采用贪婪算法, 所谓贪婪算法是指逐步构造最优解的方法, 在每个阶段都做出一个看上去最优的决策. 具体步骤如下:

初始参数: $I(i) = |G_i|$, $E(i) = 0$, $i = 1, 2, \dots, M$, 其中 $|\cdot|$ 表示集合的基数;

初始集合: T^l ;

第一步 任取 $T_j \in T^l$, 选取 T_j 到 $\{DM_i\}$ 的最优分配使得当前的 $W_{RMS}(O, M)$ 最小;

第二步 根据第一步的分配结果, 调整相应的 $E(i)$;

第三步 $T^l \leftarrow T^l / \{T_j\}$, 如果 $T^l \neq \emptyset$ 转第一步, 否则算法终止.

3.1.2 T^0 到决策实体的分配 T^0 中的任务可以由某一个平台粒单独完成, 它们对组织结构与使命的适应性测度没有影响, 但可以影响整个使命的完成时间. 通过优化 T^0 中所有任务完成的最终时间来决定 T^0 到决策实体的分配, 对于这个小规模的规划问题可以通过全局搜索或其他一些列表规划方法^[4]求解.

3.2 平台集粒化

在平台集所有可能的粒度划分中寻找最优的粒

度划分是一个 NP 完全问题, 对此类问题, 目前还没有一般意义上有效的算法. 遗传算法是根据自然选择和遗传进化的机理进行的随即搜索技术, 具有较强的全局搜索能力, 适用于解决各类组合优化问题. 文献[14, 15]采用遗传算法求解带约束的集合划分问题和集合覆盖问题, 表现了良好的性能. 同样采用遗传算法求解组织结构设计中的平台集粒化问题.

3.2.1 编码 采用直接基于划分结果的整数编码. 如果将 K 个平台划分为 M 个子集(对应于 M 个决策实体), 则一条染色体就是一个长度为 K 的由 1 到 M 的整数组成的串. 例如, $K=4, M=2$, 则染色体 $(1, 2, 2, 1)$ 表示平台 P_1 和 P_4 被划分给 DM_1 , 平台 P_2 和 P_3 被划分给 DM_2 .

3.2.2 生成初始种群 遗传算法向全局最优解的逼近程度和速度不仅与初始种群中个体数目有关系, 而且与初始种群在取值域上的分布状态有很大关系, 应该使初始种群在取值域上尽量均匀的分布. 初始种群的数量通常取 50—200.

3.2.3 适应度函数的构造 如果某个染色体是不可行的(例如 M 个子集中存在空集), 则这个染色体的适应度设为最小值 0. 否则, 假设一个染色体对应了一个平台集粒度划分 $\{G_i\}_{i \in \tau}$, 则用当前划分下通过(12)式所求得的最小的 $W_{RMS}^*(O, M)$ 来决定该染色体的适应度值, 其形式为:

$$f = 1/W_{RMS}^*(O, M) \quad (13)$$

$W_{RMS}^*(O, M)$ 越小, 适应度越高, 相应的染色体被选中的概率越大.

3.2.4 选择、交叉和变异 选择(或称为复制)的目的是把当前群体中适应度较高的染色体按照某种规则遗传到下一代群体中. 采用轮盘法选择优良个体, 同时采用最优保存策略, 即群体中适应度最大的染色体不参与交叉、变异操作, 用来替换掉下一代群体中经过交叉、变异操作后所产生的适应度最低的染色体.

交叉运算是遗传算法中产生新染色体的主要操作过程, 以某一概率相互交换两个染色体之间的部

分基因. 采用两点交叉的方法, 交叉概率 p_c 的取值范围是 0.5—1.0.

变异运算是染色体上的某一个或几个基因座上的基因按某一较小的概率(p_m 通常取 0.0001—0.1)进行改变, 也是产生新染色体的一种方法. 首先确定各个染色体的基因变异位置, 然后依照变异概率将变异点随机选择一个与原来不同的划分结果作为新的基因值.

3.2.5 算法终止条件 采用规定最大迭代次数(通常取 300—500), 当遗传算法的操作次数达到最大迭代次数时, 即终止运算, 输出结果.

平台集粒化的遗传算法流程如图 1 所示.

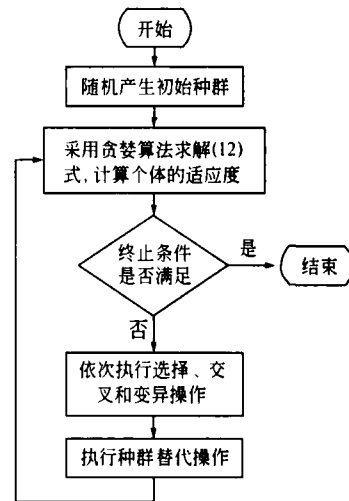


图 1 平台集粒化的遗传算法流程图

4 阶段 II: 粒内规划阶段

这一阶段决定了每个决策实体控制的平台到需要执行的任务的具体配置, 目标是 minimized 使命完成的总时间. 这类问题通常是 NP 完全问题^[3], 解决方法大多数可以归为列表规划方法^[4], 但这些方法的性能会随着平台和任务的增多而大大下降. 由于通过阶段 I 已经得到平台粒到任务粒的分配, 本阶段将原规划问题分解为一些独立的子规划问题, 并采用全局搜索或者其他一些列表规划方法求解, 由于规模较小, 能够得到更加优化的结果.

假设由阶段 I 得到, 决策实体 DM_m 拥有 p_m 个平台, 平台集记为 $DM_m(P)$; 需要执行 q_m 个任务,

任务集记为 $DM_m(T)$, 其中

$$DM_m(P) = \{P_{k_h^m} | k_h^m \in \{1, 2, \dots, K\} \\ h = 1, 2, \dots, p_m\}, \quad (14)$$

$$DM_m(T) = \{T_{i_j^m} | i_j^m \in \{1, 2, \dots, N\} \\ j = 1, 2, \dots, q_m\}. \quad (15)$$

DM_m 的能力向量记为 $R(DM_m) = [\bar{r}_{m1}, \bar{r}_{m2}, \dots, \bar{r}_{mL}]$, \bar{r}_{ml} 表示该决策实体可控制的功能类型 l 的数量, 其中

$$\bar{r}_{ml} = \sum_{h=1}^{p_m} r_{k_h^m l}. \quad (16)$$

定义 5 记由任务集 $DM_m(T)$ 诱导出的新任务集为 $DM'_m(T) = \{T_{i_j^m}^m\}$, $T_{i_j^m}^m$ 的估计处理时间和地理限制向量与 $T_{i_j^m}$ 相同; 资源需求向量为 $[R_{i_j^m}^m, R_{i_j^m}^m, \dots, R_{i_j^m}^m]$, 其中

$$R_{i_j^m}^m = \min(R_{i_j^m}^m, \bar{r}_{ml}) \quad (17)$$

新任务间的优先顺序关系不变, 即:

$$P^m(i_{j_1}^m, i_{j_2}^m) = \begin{cases} 1, & \text{如果任务 } T_{i_{j_1}^m}^m \text{ 必须在} \\ & \text{任务 } T_{i_{j_2}^m}^m \text{ 执行前完成} \\ 0, & \text{否则.} \end{cases} \\ = P(i_{j_1}^m, i_{j_2}^m) \quad (18)$$

其中, $j_1, j_2 = 1, 2, \dots, q_m$.

由(17)式容易看出, 新任务集 $DM'_m(T)$ 可由决策实体 DM_m 单独执行.

定义 6 对于任务集 $DM'_m(T)$ 和平台集 $DM_m(P)$, 一个合理的任务-平台分配矩阵 TP^m 满足如下条件:

$$(1) TP^m(i_j^m, k_h^m) = \begin{cases} 1, & \text{如果平台 } P_{k_h^m} \text{ 分配} \\ & \text{到任务 } T_{i_j^m}, \\ 0, & \text{否则} \end{cases} \quad (19)$$

$$(2) \sum_{h=1}^{p_m} TP^m(i_j^m, k_h^m) \cdot r_{k_h^m l} \geq R_{i_j^m}^m. \quad (20)$$

其中, $j = 1, 2, \dots, q_m$; $h = 1, 2, \dots, p_m$; $m = 1, 2, \dots, M$; $l = 1, 2, \dots, L$.

定理 1 假设已有 M 个合理的任务-平台分配矩

阵 TP^m , $m = 1, 2, \dots, M$, 则满足(21)式的任务-平台分配矩阵 TP 是一个合理的分配矩阵:

$$TP(i, k) = \begin{cases} 1, & \text{如果存在 } m, \text{ 满足 } TP^m(i, k) = 1 \\ 0, & \text{否则.} \end{cases} \\ = \sum_{m=1}^M TD(i, m) \cdot PD(k, m) \cdot TP^m(i, k) \quad (21)$$

其中, $i = 1, 2, \dots, N$; $k = 1, 2, \dots, K$.

证明 任取 T_i , 不妨设由阶段 I 得到 T_i 分配给其中 n 个决策实体, 即 $T_i \in DM_{m_1}(T), \dots, T_i \in DM_{m_n}(T)$, 则有:

$$\sum_{s=1}^n \bar{r}_{m_s l} \geq R_{i l} \quad (22)$$

其中, $l = 1, 2, \dots, L$.

由(17), (20), (21)和(22)式可以得到:

$$\sum_{k=1}^K TP(i, k) \cdot r_{kl} = \sum_{k=1}^K \left(\sum_{m=1}^M TD(i, m) \cdot PD(k, m) \cdot TP^m(i, k) \right) \cdot r_{kl} \\ = \sum_{s=1}^n \sum_{h=1}^{p_{m_s}} TP^{m_s}(i, k_h^{m_s}) \cdot r_{k_h^{m_s} l} \geq \sum_{s=1}^n R_{i l}^m = \sum_{s=1}^n \min(R_{i l}, \bar{r}_{m_s l}) \geq R_{i l},$$

其中 $l = 1, 2, \dots, L$.

由于 T_i 选取的任意性, 定理得证.

通过定理 1, 可以将具有大量平台和任务的规划问题分解为一些独立的具有少量平台和任务的子规划问题. 我们的研究表明改进的 MDLS 算法对于小规模规划求解具有良好的性能^[4], 因此本阶段采用该算法求解这些子规划问题. 粒内规划阶段的具体过程为:

第一步 由任务集 $DM_m(T)$ 诱导出新任务集 $DM'_m(T)$;

第二步 对于任务集 $DM'_m(T)$ 和平台集 $DM_m(P)$, 采用改进的 MDLS 算法^[4] 得到任务-平台分配矩阵 TP^m ;

第三步 由(21)式计算最终的任务-平台分配矩阵 TP , 并得到任务完成的总时间.

5 案例分析

本节通过案例说明我们的方法, 该案例是设计一个组织完成一个由 11 个任务组成的使命 M, 任务图如图 2 所示. 考虑下列 8 种资源需求/能力: AAW, ASUW, ASW, GASLT, FIRE, ARM, MINE 和 DES. 假设组织有 5 个可利用的决策实体和 20 个平台资源. 平台和任务的参数分别见表 1 和表 2. 关于案例的详细描述以及任务和平台的描述

可参见文献[9].

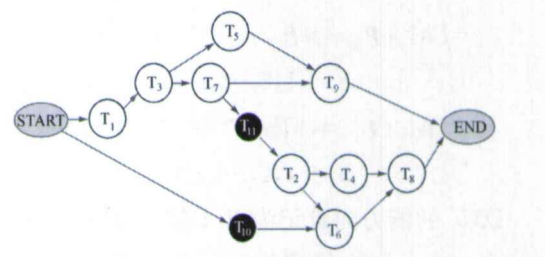


图 2 使命 M 的任务图

表 1 平台参数

序号	平台类型	资源能力								速度
		AAW	ASUW	ASW	GASLT	FIRE	ARM	MINE	DES	
1	DDG	10	10	1	0	9	5	0	0	2.00
2	FPG	1	4	10	0	4	3	0	0	2.00
3	CG	10	10	1	0	9	5	0	0	2.00
4	ENG	0	0	0	2	0	0	5	0	4.00
5	INFA	1	0	0	10	2	2	1	0	1.35
6	SD	5	0	0	0	0	0	0	0	4.00
7	AHI	3	4	0	0	6	10	1	0	4.00
8	CAS1	1	3	0	0	10	8	1	0	4.00
9	CAS2	1	3	0	0	10	8	1	0	4.00
10	CAS3	1	3	0	0	10	8	1	0	4.00
11	VF1	6	1	0	0	1	1	0	0	4.50
12	VF2	6	1	0	0	1	1	0	0	4.50
13	VF3	6	1	0	0	1	1	0	0	4.50
14	SMC	0	0	0	0	0	0	10	0	2.00
15	TARP	0	0	0	0	0	0	0	6	5.00
16	SAT	0	0	0	0	0	0	0	6	7.00
17	SOF	0	0	0	6	6	0	1	10	2.50
18	INF (AAV-1)	1	0	0	10	2	2	1	0	1.35
19	INF (AAV-2)	1	0	0	10	2	2	1	0	1.35
20	INF (AAV-3)	1	0	0	10	2	2	1	0	1.35

表 2 使命 M 的任务参数

序号	任务名称	资源需求								位置	处理时间
		AAW	ASUW	ASW	GASLT	FIRE	ARM	MINE	DES		
1	压制高地	0	1	1	9	16	14	0	0	[22,55]	8.6
2	抢占北滩	0	0	0	10	13	10	1	0	[23,82]	8.8
3	抢占南滩	0	1	1	11	15	13	0	0	[28,92]	9.7
4	北区防御	2	0	0	3	0	6	1	0	[24,66]	14.9
5	南区防御	0	1	0	0	0	5	0	0	[29,83]	7.2
6	北路行进	1	1	0	2	1	11	2	0	[27,39]	9.7
7	南路行进	0	1	0	0	1	11	2	0	[9,89]	7.7
8	占领港口	0	0	0	18	11	4	2	0	[26,42]	9.3
9	占领机场	2	1	0	18	8	4	0	0	[9,92]	13.6
10	随机任务	5	2	2	0	0	0	0	0	[18,44]	12.6
11	随机任务	3	2	6	0	0	0	0	5	[30,38]	12.2

5.1 使命实现:设计一致性组织

假设内部和外部工作负载权重相等,即 $W^I = W^E = 1$. 应用本文提出的方法建立匹配使命 M 的最优组织结构 O(不考虑组织结构的决策分层设计,该设计不影响下面的各种比较).

(1) 阶段 I:种群数量取 100,进化代数取 500,交叉概率取 0.6,变异概率取 0.05,经过遗传算法运算,得到平台-决策实体分配和任务-决策实体分配如图 3 所示. 组织结构 O 与使命 M 的适应性测度由(11)式得到: $W_{RMS}(O, M) = 4.05$.

(2) 阶段 II:针对每个决策实体控制的资源和需要执行的任务,采用改进的 MDLS 算法求解这些子规划问题,得到最终的任务-平台分配,具体分配结

果见图 4. 整个使命的完成时间是 98.9 时间单位.

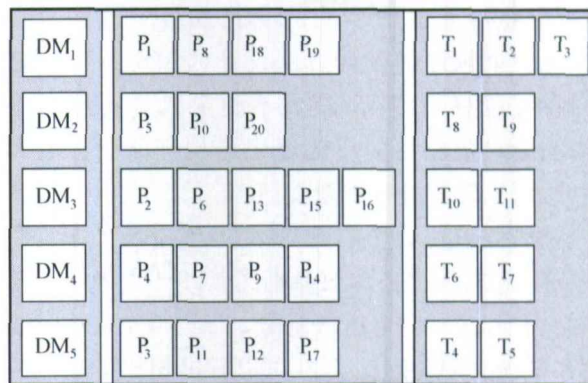


图 3 组织 O 执行使命 M 的决策实体-平台-任务分配

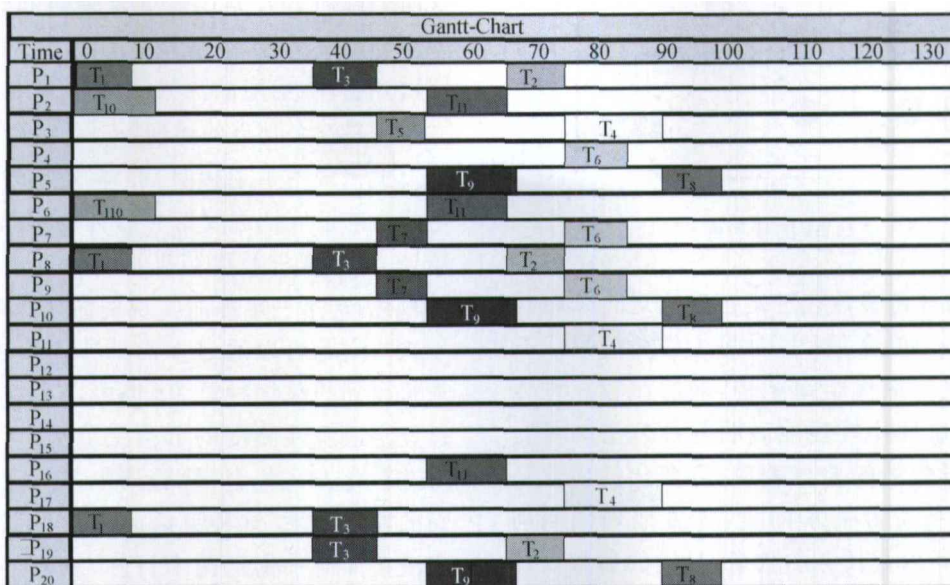


图 4 任务-平台的分配结果

5.2 本文方法和现有方法的比较

分别应用 3 阶段方法^[2,3,9]及其改进方法^[4,5]建

立与使命 M 相匹配的最优组织结构 O₁ 和 O₂. 3 种方法的适应性测度和使命完成时间见表 3.

表 3 基于粒度计算和遗传算法的组织结构设计方法和现有方法的比较

使命	O		O ₁		O ₂	
	适应性测度	使命完成时间	适应性测度	使命完成时间	适应性测度	使命完成时间
M	4.05	98.9	7.18	172.5	8.32	121.4
M'	7.46	181.6	9.04	270.7	10.67	213.3

可以看出,在这两项指标上本文的方法均优于 3 阶段方法和其改进方法. 已有的方法是先优化使命完成时间,再优化适应性测度,通过实验可以看出前者对后者的影响很大;另外列表规划方法随着问题

规模的增大而越来越偏离最优解. 本文方法通过问题论域的粒化,在新的粗粒度层次上优化组织的适应性测度,在粒度内的细化层次上优化使命的完成时间,一定程度上同时优化了这两项指标.

另一方面,组织在执行使命的过程中,不确定的外界因素将导致使命环境的变化或组织约束的变化.因此一个较好的设计方法设计的组织结构应该是鲁棒的.这样,当使命参数不确定或者动态变化时,由初始参数获得的优化组织结构仍能保持较高的性能,而不需要进行重新设计.

这里,假设使命场景 M 改变为使命场景 M' ,使命任务图见图 5,任务参数参见文献[9].当仍然采用组织结构 O, O_1 和 O_2 执行使命 M' 时,得到的适应性测度和使命完成时间见表 3.可以看出相比于其他两个组织结构,组织结构 O 在新的使命环境中仍能保持较高的性能.

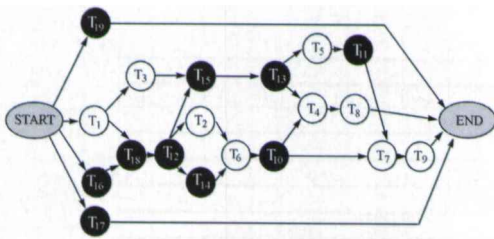


图 5 使命 M' 的任务图

最后对使命环境参数在一定范围内进行随机扰动,随机生成 50 个使命 $\{M_i\}$,对每个使命分别用三种方法求得相应的最优组织结构,最后比较三种方法的平均性能.这种比较更能说明各种方法的可行性以及在性能指标上的差异表现.可变化的使命环境参数包括:任务资源需求向量,任务地理限制向量,任务处理时间以及任务的数量,具体的参数设计参见文献[9].三种方法设计的组织结构对使命 $\{M_i\}$ 的平均适应性测度见图 6.

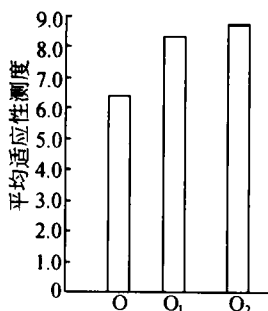


图 6 3 种方法的平均适应性测度

通过图 6 可以看出,改进的 3 阶段方法是在牺牲了组织的适应性测度的情况下减少使命完成时间

的.我们的方法在平均性能上仍然表现最好,这也说明了对于组织结构设计这种复杂问题,采用粒化方法在不同的粒度层次上求解可以简化问题求解的复杂度并得到理想的优化结果.

6 结论与讨论

组织结构设计问题是一个大规模、多目标的复杂性问题,主要目标包括优化使命完成时间和组织结构与使命的适应性测度,目前的方法不能同时优化这两个目标.本文基于粒度计算和遗传算法提出了一种新的设计方法,该方法首先在粗粒度层次上优化组织的适应性测度,然后在细粒度层次上优化使命的完成时间,从而能够在一定程度上同时优化这两个目标.“人类在解决复杂问题时,通常是先把问题分解或简化,忽略其中的细节,然后从较抽象的层次开始,一层层的深入到其中的细节”^[11],本文的方法正是这一观点的体现.

由于采用了遗传算法,本文方法对于组织结构与使命的适应性测度这一目标能够给出近似最优的结果.但与这一最优结果相对应的平台集和任务集的粒化,却不一定都能实现使命完成时间的优化.对于这一问题可以通过下面的方法解决:在遗传算法求解得到的最终种群中(也可以考虑中间种群),通过设定适应性测度的可接受范围,可以得到对应的可选粒化方案集合,然后从中寻找完成使命时间最短的粒化结果.这种灵活的方法可以根据实际需要来决定如何平衡这两个目标,从而得到适合的组织结构.

可以预见,粒度计算将在复杂问题求解领域发挥巨大的作用.在以后的研究中,我们将进一步研究任务集的粒化方法,另外扩展该方法设计鲁棒的和适应性的组织结构也是一个重要的研究方向.

参 考 文 献

- 1 阳东升,张维明,刘忠,等. C2 组织的有效测度与设计. 自然科学进展, 2005, 15(3): 349—356
- 2 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
- 3 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
- 4 Yang DS, Lu YL, Liu Z. Research on algorithms of task scheduling. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics. Shanghai, 2004. New York: IEEE Press, 2004, 42—47
 - 5 Yang DS, Peng XH, Zhang WM, et al. An approach to designing C2 organization. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics. Shanghai, 2004. New York: IEEE Press, 2004, 37—41
 - 6 Pete A, Kleinman DL, Pattipati KR. Structural congruence of tasks and organizations. In: Proceedings of the 1994 Symposium on Command and Control Research and Decision Aids. NPS, Monterey, CA, 1994, 168—175
 - 7 Diedrich FJ, Entin EE, Hutchins SG, et al. When do organizations need to change-Part I: Coping with organizational incongruence. In: The 8th International Command and Control Research and Technology Symposium, Washington, DC, 2003
 - 8 Kleinman DL, Levchuk GM, Hutchins SG, et al. Scenario design for the empirical Testing of organizational congruence. In: The 8th International Command and Control Research and Technology Symposium, Washington, DC, 2003
 - 9 Levchuk GM, Levchuk YN, Meirina C, et al. Normative design of project-based organizations - Part III: Modeling congruent, robust, and adaptive organizations. IEEE Transactions on Systems, Man, and Cybernetics, 2004, 34(3): 337—350
 - 10 Zadeh LA. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems, 1997, 90(2): 111—127
 - 11 张 钹, 张 铃. 问题求解理论及应用. 北京: 清华大学出版社, 1990, 1—86
 - 12 张 钹, 张 铃. 模糊商空间理论(模糊粒度计算方法). 软件学报, 2003, 14(4): 770—776
 - 13 李道国, 苗夺谦, 张东星, 等. 粒度计算研究综述. 计算机科学, 2005, 32(9): 1—12
 - 14 林智勇. 带约束集合划分优化问题及其求解. 计算机工程与科学, 2005, 27(7): 98—100
 - 15 陈 亮, 任世军. 一种遗传算法在集合覆盖问题中的应用研究. 哈尔滨商业大学学报(自然科学版), 2006, 22(2): 67—70

(上接第 661 页)

3. 基于网络的发现和创新

千万亿次量级(petascale)(即每秒钟可进行千的五次方运算)计算机、海量数据流、巨型数据库的迅速发展,这些前所未有的挑战对我们能力的要求,远不是今天所能够满足的. 2008 年度 NSF 的预算请求中,包括有一项新的涉及其各部门的 5200 万美元投资计划,用于通过开发新一代基于计算的概念与工具,分析源于所有学科前沿发现的复杂的含有大量数据的交互系统,借此拓展国家的创新能力. 网络环境下的科学发现与技术创新(CDI)计划旨在探索极具创新的新概念、新方法和新工具,以解决计算、生物与物质世界之间相互交叉所带来的挑战.

4. 网络和信息技术研发

CDI 计划是对相关联机构间合作开展的网络和信息技术研发(NITRD)计划活动的补充. 2008 年度 NSF 在 NITRD 计划下的研究领域将增加 9000 万美元的投资,这些领域的研究总额将接近 10 亿美元. 特别增加的投入将覆盖高端计算基础设施及应用、大型网络建设以及信息技术的社会、经济和人力方面等领域.

5. 纳米技术

2008 年,NSF 对国家纳米技术计划(NNI)的投入增加了近 1700 万美元,总投入达到 3.9 亿美元. 增加的投入主要集中在基础纳米尺度研究、纳米材料开发,以及纳米技术开发对环境、健康和安全性影响. 其中有 2580 万美元用于支持粒子、纳米材料与生物界相互作用的跨学科研究.

6. 海洋研究

理解社会与海洋的关系,对于确保海洋环境的清洁、健康与稳定具有重要作用. 今年 1 月发布的“海洋研究优先计划”,确定了理解海洋动力学、预报海洋事件和管理海洋资源所需的重要研究挑战. 2008 年度 NSF 新增加了 1700 万元的预算,将用于以下几个方面研究:控制和调控海洋生态系统过程的复杂动力学,大西洋经向翻转环流的变异性,海岸生态系统对极端和突发事件的响应,改进海洋观测的新型传感器开发.

7. 高级激光干涉引力波天文台

2008 年度 NSF 提出 3275 万美元的预算请求,开始建造高级激光干涉引力波天文台(高级 LIGO). 作为激光干涉引力波天文台(LIGO)的升级设备,高级 LIGO 项目将改进激光、悬浮液、地震绝缘体和光学子系统,以减少地球上因引力波通过而引起时空误差所带来的局限. 高级 LIGO 将极大地增强该设备的灵敏性,以确保引力波的探测,并建立起引力波天文学研究的新领域.

(下转第 700 页)