文章编号: 1001- 2486(2009) 05- 0064- 06

基于 NSGA- ㊐多目标优化的 C2 组织设计*

乔士东, 黄金才, 修保新, 张维明
(国防科技大学 C⁴ISR 技术国防科技重点实验室, 湖南 长沙 410073)

摘 要: 把 NSGA- ㊐算法用于求解 C2 组织设计问题。分析了 C2 组织设计常见处理算法在优化目标处理和算法流程两方面存在的问题, 给出用 NSGA- ㊐算法求解 C2 组织设计问题的算法设置。把 NSGA- ㊐这样一种多目标优化算法引入 C2 组织设计问题, 改变了以往研究此类问题时只能定义单个指标的情况, 使领域专家能定义和研究新的优化目标。针对 C2 组织设计问题的特性做了调整后, 实验结果数据表明 NSGA- ㊐可以迅速地同时得到高质量和富有启发性的一群优化结果。

关键词: C2 组织设计; 遗传算法; 多目标优化算法; NSGA- ㊐
中图分类号: C394; TP24 文献标识码: A

Organizational Design of C2 by NSGA- ㊐Multi-objective Optimization Algorithm

QIAO Shi-dong, HUANG Jin-cai, XIU Bao-xin, ZHANG Wei-ming
(Key Laboratory of C⁴ISR Technology, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: An efficient multi-objective optimization (MO) algorithm named as NSGA- ㊐ is applied to solve the problem of organizational design of C2. Before introducing the settings of NSGA- ㊐, a discussion is made about the shortcomings in forming optimizing objective by weighted method and proceeding procedure like nested GA, which exist in popular algorithms of organizational design of C2. With this MO algorithm, more indexes can be optimized within one procedure which makes it more practical than dealing with single index as usually. Some modifications are made for NSGA- ㊐ to match the specifications of the experiment, a group of appealing and inductive results are reached, which reveals the effectiveness of NSGA- ㊐ in the problem of organizational design of C2.

Key words: organizational design of C2; genetic algorithm(GA); multi-objective optimization(MO); nondominated sorting genetic algorithm ㊐(NSGA- ㊐)

组织执行使命或完成任务的能力与其结构和任务分配方式有密切的关系, 组织的结构与任务越匹配, 则执行效率越高[1- 4, 7]。特别是在军事领域, 如何设计 C2 组织以使组织能够高效地完成使命或执行任务成为一个热点问题。

Levchuk 等提出了任务建模和三阶段方法[5- 6], 把组织设计问题分解成一系列更小规模的子问题。三阶段方法模仿人处理问题的方式, 但在算法原理上已经不能保证得到最优解。

Yu 等提出了基于分组和嵌套 GA 的组织设计方法。在 GA 的各种应用中, 一般来说很难结合应用领域的先验知识, 而 Yu 把分组技术和 GA 结合起来应该说是把问题域很好地结合到 GA 的通用流程中。分组技术虽然满足人的直观感受和经验, 但分组技术是对问题求解空间的限制, 未必能够恰好包含最优解。另一方面, 在嵌套 GA 中, 被嵌套的 GA 实际上是计算外层 GA 种群个体的适应值, 使用两层 GA 往往导致问题求解进度缓慢, 难以满足实际应用对求解速度的需求。

从实用的角度看, C2 组织设计问题中需要关注多个指标, Levchuk 和 Feili Yu 等采取的办法是把这些指标根据先验的权系数合成一个指标。这种做法也是某种意义上的多目标优化, 但在各指标之间存在关联或耦合关系时不能保证得到最优解。

* 收稿日期: 2009- 03- 12
基金项目: 国家自然科学基金资助项目(70701038)
作者简介: 乔士东(1975—), 男, 讲师, 博士。

在组织设计问题中引入多目标优化算法, 则可以避免上述方法的各种弊端。首先, 多目标优化问题有 VEGA、MOGA、NPGA、NSGA、micro- GA、PAES、SPEA 和 PESA[8] 等基于 GA 的算法, 这些算法采用 Pareto 最优等概念把各子目标之间的折中和此消彼长的过程自然地融合到算法流程中, 不必由人先验地确定各子目标之间的权重关系, 可以改变目前的组织设计方法只使用单个优化目标的现状, 增强分析和求解问题的能力; 第二, 可以借鉴 GA 在神经网络结构优化方面的做法。在此类方法中, 把神经网络的结构选取和优化给定结构两方面合并到同一个 GA 流程中, 类似地, 也可以把组织设计中的分组与优化计算过程合并到一个 GA 算法内, 从而简化组织设计的算法流程, 降低求解过程的计算复杂度。

这样, 采用 NSGA- ㊣这个优秀的多目标优化(MO) 算法求解组织设计问题就是合适的选择了[10- 11], 但不是简单的算法运用过程, 要对 NSGA- ㊣做多方面的修改才行, 例如 C2 组织设计涉及的资源约束规模相当大, 基本上每个资源都有一定的限制, 需要结合问题的特殊性调整 NSGA- ㊣算法才能达到有效求解问题的目的。NSGA- ㊣算法参见 Deb 的论文[9], 这里不再重复。

1 用 NSGA- ㊣求解 C2 组织设计问题

C2 组织设计问题需要确定: 1) 决策者(Decision maker) 的个数; 2) 任务(Task) - 决策者关联矩阵; 3) 平台(platform) - 决策者关联矩阵; 4) 任务- 平台关联矩阵。C2 组织设计问题带有明显的层次特征, 这里只就第四个问题展开实验, 即研究任务- 平台的最优关联矩阵。其他层次的优化问题与此处的处理方法类似, 只是算法所用编码方法略有调整。

图 1 给出 C2 组织设计问题中使命的任务执行顺序图, 该图定义了使命 M 中任务之间的依赖关系和执行次序。后文的实验针对图 1 所示使命展开, C2 组织设计问题的详细介绍请参考文献[7] 和[12]。实验采用两个子目标组成的目标向量, 即全部任务总的完成时间和平台平均运动距离。

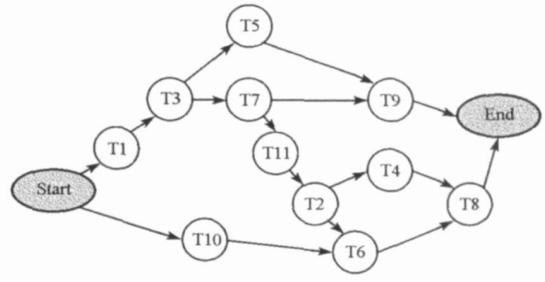

图 1 使命 M 内各任务的执行顺序图

Fig. 1 Precedence task graph for mission M

目标向量的计算满足如下约定:

(1) 任务- 平台关联关系有效当且仅当处理该任务的所有平台提供的资源能力满足该任务的资源需求。这种资源约束是最难满足的约束, 第 3 节会介绍具体处理约束条件的办法。

(2) 各平台执行任务的顺序必须满足图 1 中的处理流程;

(3) 只有指派处理某个任务的所有平台都到齐时, 才能开始工作, 否则已经到达的平台只能等待其他平台;

(4) 平台完成当前任务后, 立即向下一项任务运动。

有些研究者假设平台完成任务后, 会折返到某个指定位置(例如, 该平台处理的所有任务的中心), 再从该位置出发去处理下一个任务。这一假设的合理性不充分, 此处不采用。

2 优化子目标的选取

C2 组织设计问题关注多个设计指标。处理多个指标的常见方法是把各指标加权形成单个指标, 这种处理策略需要事先确定各指标的重要性关系[7, 12], 除非各个指标相互独立, 否则不能保证算法收敛到最优解。采用 NSGA- ㊣这种多目标优化算法后, 就可以直接处理关注的多个优化目标, 而且可以保证算法的最优性。

C2 组织设计问题可以采用的优化指标包括:

(1) 全部任务总的完成时间

这是组织设计问题常用的优化目标。协作往往导致平台之间相互等待, 包括属于同一决策者的平台之间的等待, 特别是属于不同决策者的平台之间的合作还可能付出更大的代价, 这是组织设计中最显

著的影响因素。决策者之间协作往往由于情报传递、消息确认、资源准备和命令通信等导致延迟。

(2) 平台的平均运动距离

根据自动目标识别领域的经验, 若目标处于运动状态, 则其被发现的概率会比处于静止状态时增加百分之九十以上, 所以, 减少平台运动距离就是降低被发现的可能性, 进而保证平台未被瘫痪或击毁, 顺利地到达任务所在的地点。

(3) 平台能力余度

该指标度量平台的能力超出任务要求的资源数量, 前提是这些平台能处理该任务, 所以该指标是非负的。

考虑到作战过程中任务对资源数量的要求经常变化, 平台能力超出任务需求越高, 则该任务- 平台关联关系的适应能力越强, 反之可能无法处理任务资源需求增长的情况。

实验采用两个子目标组成的目标向量, 即全部任务总的完成时间和平台平均运动距离。

3 NSGA- ㊣算法的设置

C2 组织设计问题有自己的特殊性, 例如涉及大规模的约束条件, 所以要对 NSGA- ㊣算法作如下几点调整:

(1) 个体编码方案

采用布尔型编码方案, 即用一个布尔矩阵表达任务- 平台关联矩阵。

NSGA- ㊣求解测试问题时采用的是浮点编码方案。C2 组织设计问题关注的是任务- 平台关联矩阵, 这是一个 0/1 矩阵, 采用浮点方案会导致算法的进化效率低: 一方面, 遗传操作很难使个体的基因位置发生 0~ 1 的转变, 经常使遗传操作失效, 而布尔型编码只要变化一个基因位置, 就可以使个体有变化; 另一方面, 浮点编码遗传操作的执行速度比布尔型编码的执行速度慢得多。

(2) 资源约束处理

C2 组织设计问题的资源约束规模与“任务个数× 资源个数”相当, 种群个体要满足如此众多的约束条件是十分困难的。文献[9] 处理约束条件的方法是把不满足约束的个体的适应值设置成尽量小的数, 使只有满足约束条件的可行个体才能进入下一代。按照该方法, 由于 C2 结构设计问题涉及的约束规模庞大, 个体很难满足所有的约束, 导致进化过程长期缺少可行的个体或者可行个体的数量过少, 使得进化效率低。

针对这种情况, 实验采用强制修正个体的方法, 即若个体不满足某项资源约束, 则给该任务随机指派某个平台, 直到该个体满足所有资源约束为止。这种操作的目的是把所有不可行的个体都强制转化成可行的个体, 至于得到的个体的优秀程度, 不是此处关心的问题, 至少经过该操作后整个种群都满足约束。

(3) 精简平台操作

在每代进化之后, 从每个任务的所有处理平台中尝试去掉某个平台。若去掉某个平台后, 该任务仍能被顺利执行, 或者说个体仍然是可行的, 则继续尝试去掉其它平台, 直到该个体变成不可行的为止。

在实验中引入精简平台的操作后, 可以观察到种群质量迅速提高的现象。

(4) 个体强制变异操作

在每代进化之后, 比较任意两个个体, 若两个个体完全相同, 则对其中某个个体执行变异操作, 而且只对一个基因位置执行变异。

引入该操作的原因是在实验中观察到出现大量重复个体的现象, 使得种群的进化速度缓慢。NSGA- ㊣在处理 FON 和 KUR 等测试问题时表现良好, 是因为测试问题的 Pareto 解数量众多而且比较均匀地分布, 适合 NSGA- ㊣搜索能力强和优秀个体种群分布均匀的特点。但在 C2 组织设计问题中, 优秀个体的个数只有几个, 而且趋于集中, 所以在实验中出现上述问题。采用个体强制变异操作后, 使算法在进化过程中不断探索解空间, 提升了算法的搜索能力。

4 实验结果分析

NSGA- ㊦算法采用 Visual C++ 6.0编写, 并在 Intel(R) Core2 Duo CPU 2.33GHz 2G 内存的台式计算机上运行。

NSGA- ㊦算法的种群规模为 100 个个体, 其优秀个体保存策略要求的优秀个体数目为 100 个。采用按进化代数的停止策略, 在上述环境下每执行 100 次进化需要约3~ 8s。从问题的规模看, 算法的计算速度是很迅速的。

图2 给出一次实验的目标向量分布图, 该图是采用修改的 NSGA- ㊦算法执行 1200 次进化后得到的。图中横坐标表示优化目标向量中的第一个分量 OPT- 1, 纵坐标表示第二个分量 OPT- 2。图中有部分个体的目标向量十分接近, 有的甚至完全重合, 这说明要达到相同或相近的目标可以有多种任务- 平台关联方法。

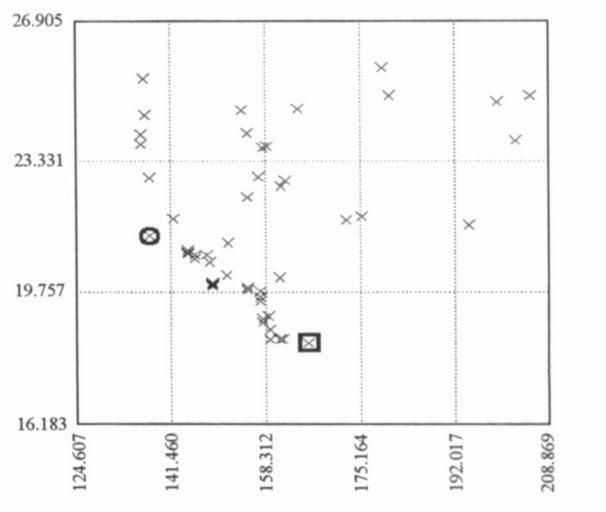

图 2 第 1200 代进化得到的父代 P_{1200}

Fig. 2 The population distribution after 1200generation

4.1 实验分析 iv

考察图中圆圈标记的个体。该个体完成使命 M 的时间为 OPT- 1= 137.655, 处理使命 M 过程中平台的平均运动距离为 OPT- 2= 21.302。由图中种群的分布情况可知, 该个体的任务完成时间是比较短的, 但其平台平均运动距离比较大。

表1 给出该个体表达的任务- 平台分配关系矩阵。其中, 符号“|”表示对应平台处理该任务, 符号“_”表示对应平台与该任务无关; 表2 给出各个任务的开始处理和完成时间, 其中, 任务的完成时间= 任务的开始时间+ 处理任务所需的时间; 图3 根据该个体表达的任务- 平台分配关系矩阵给出使命 M 的甘特图。

表 1 任务 - 平台分配关系矩阵

Tab.1 Platform-task allocation for mission M

序号	任务名称	平台 ID	
		1 2 3 …	18 19 20
1.	压制抢滩	_ _|_ _ _ _ _ _|_ _ _ _ _ _ _ _|_	
2.	抢占北滩	|_ _ _ _ _ _ _|_ _ _ _ _ _ _ _ _ _|	
3.	抢占南滩	|_ _|_ _ _ _ _|_ _ _ _ _ _ _ _|	
4.	北区防御	_ _|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|	
5.	南区防御	|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	
6.	北路行进	_ _ _ _ _ _|_ _ _ _ _ _ _ _ _ _|__	
7.	南路行进	_ _ _ _ _ _ _ _||_ _ _ _ _ _ _ _ _ __	
8.	占领港口	_ _ _ _|_|_ _ _ _ _|_ _ _ _|_ _	
9.	占领机场	_ _ _|_ _ _ _ _|_ _ _ _ _ _|_|_	
10.	随机任务 1	_|_ _ _ _ _ _ _ _ _|_ _ _ _ _ _ _ _	
11.	随机任务 2	_|_ _ _ _ _ _ _ _|_ _ _ _|_ _ _ _	

表 2 使命 M 内各个任务的处理时间

Tab.2 Processing time of each task for mission M

序号	任务名称	平台序号	开始时间	结束时间
1.	压制抢滩	3,10,9	8.67015	17.27015
2.	抢占北滩	1,9,20	90.43714	99.23714
3.	抢占南滩	1,4,10,20	26.77015	36.47015
4.	北区防御	3,20	111.11212	126.01212
5.	南区防御	1	40.99784	48.19784
6.	北路行进	7,18	101.57957	111.27957
7.	南路行进	9,10	41.27899	48.97899
8.	占领港口	5,7,13,18	128.35455	137.65455
9.	占领机场	4,10,17,19	78.02888	91.62888
10.	随机任务 1	2,12	1.83249	14.43249
11.	随机任务 2	2,11,16	69.95541	82.15541

观察图 3, 可以得到如下两点结论: 1) 任务的执行顺序满足使命 M 的要求(图 1) , 说明当前个体表达的任务– 平台关系矩阵是可行解; 2) 尽可能地提高任务之间的并行程度, 可以有效地缩短使命的完成时间(OPT – 1) 。

例如, 图中 T5 和 T7 几乎同时开始, 其开始时间的微小差异可能是平台之间相互等待或者等待平台到达任务执行地点所造成的。同样地, T9 和 T11 的开始时间也很接近。

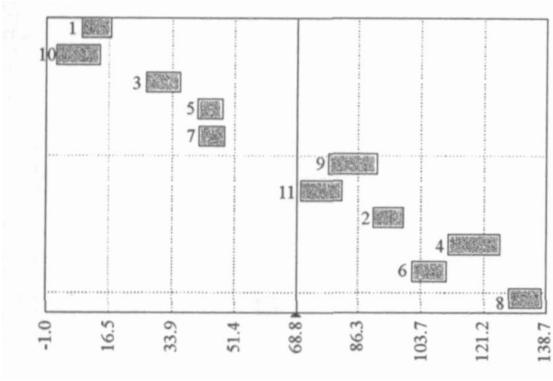

图 3 圆圈标记个体(任务– 平台关系矩阵) 对应的甘特图

Fig. 3 Gantt chart of the circle-marked individual

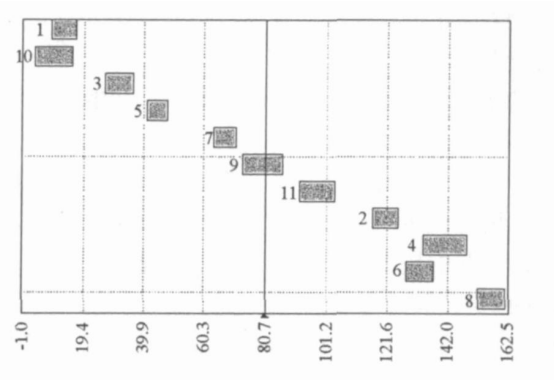

图 4 方框标记个体(任务– 平台关系矩阵) 对应的甘特图

Fig. 4 Gantt chart of the rectange-marked individual

4.2 实验分析 ㊣

考察图 2 中用方框标记的个体。该个体完成使命 M 的时间为 OPT – 1 = 161. 470, 处理使命 M 过程中平台的平均运动距离为 OPT – 2 = 18. 463, 其任务完成时间偏长, 但其平台平均运动距离比较小。

表 3 给出各个任务的开始处理和完成时间。为了方便比较当前个体与实验结果 I 中的个体, 在表 3 中把相同的量用斜体标记。可以看出, 即使两者关于平台的分配方案很相近, 完成使命 M 的时间仍然存在明显的差异。

表 3 使命 M 内各个任务的处理时间

Tab. 3 Processing time of each task for mission M

序号	任务名称	平台序号	开始时间	结束时间
1.	压制抢滩	*3, 10, 9*	*8. 67015*	*17. 27015*
2.	抢占北滩	*1, 9, 20*	117. 05412	125. 85412
3.	抢占南滩	*1, 4*, 9, *20*	*26. 77015*	*36. 47015*
4.	北区防御	1, *3*, 17	133. 86973	148. 76973
5.	南区防御	*1*	*40. 99784*	*48. 19784*
6.	北路行进	*7, 18*	128. 19655	137. 89655
7.	南路行进	*9*, 19, 20	63. 43352	71. 13352
8.	占领港口	*5*, 10, *18*	152. 17010	161. 47010
9.	占领机场	9, *19*, 20	73. 35574	86. 95574
10.	随机任务 1	*2, 12*	*1. 83249*	*14. 43249*
11.	随机任务 *2*	*2, 11, 16*	92. 10993	104. 30993

根据该个体表达的任务– 平台分配矩阵, 图 4 给出使命 M 的甘特图。对比图 3 可知, 图 4 中的任务并行程度较低。例如, T5 和 T7 一前一后执行, 而两者按照使命 M 的执行顺序图(见图 1) 是可以同时开始的。

在执行任务过程中, 如下几个因素会导致时间延迟:

(1) 由于必须等所有关联平台到齐之后才能执行任务, 所以可能出现平台之间相互等待的情况。

平台未能迅速赶到的原因可能是: 正在来任务所在地点的途中, 或者该平台正在处理其他任务。

(2) 按照使命 M 的任务执行顺序关系图, 后续任务必须等待前面的所有任务处理完成之后才能开始执行。

(3) 平台运动到任务所处地点也需要时间。

上述三类因素与任务- 平台分配矩阵结合起来导致当前个体的使命完成时间有明显的延迟。

5 结 论

应用 NSGA-Ⅱ求解 C2 组织设计问题。引入多目标优化算法, 一方面为组织设计问题处理多个目标提供了有效的求解工具, 方便领域专家提出和研究新的优化目标; 另一方面, 实验表明应用 NSGA-Ⅱ可以迅速地得到高质量和富有启发性的一群优化结果, 为后续的决断和分析提供了素材, 大量可行解的出现为人们处理问题提供良好的启发。

实验只关注平台- 任务关联矩阵, 没有解决完整的 C2 组织设计问题。但此处的算法流程和问题处理方法适用于整个组织设计问题, 可能需要修改的地方就是决策者之间的协作关系要采用整数编码方案。

实际的 C2 组织设计问题涉及的平台/任务都具有相当大的规模, 下一步要结合问题的特性并结合协同策略和并行算法等使该方法尽可能地实用化, 并探索粒子群算法[11,13-14] 等其他类型的多目标优化算法。

参 考 文 献:

[1] Nadler D A, Tushman M L. Competing by Design[M]. New York: Oxford, 1997.

[2] Levchuk G M, Levchuk Y N, et al. Normative Design of Organizations-Part Ⅷ: Modeling Congruent, Robust, and Adaptive Organizations[J]. IEEE Transactions on SMC, 2004, 34(3): 337- 350.

[3] Burton R M, Obel B. Strategic Organizational Diagnosis and Design: Developing Theory for Application (2nd Ed.)[M]. Boston, MA: Kluwer Academic Publishers, 1998.

[4] Mackenzie K D. Organizational Design: The Organizational Audit and Analysis Technology[M]. Alex Pub. Co., Norwood, New Jersey, 1986.

[5] Levchuk G M, et al. Kleinman. Normative Design of Organizations-Part Ⅳ: Mission Planning[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2002, 32(3): 346- 359.

[6] Levchuk G M, et al. Normative Design of Organizations-Part Ⅱ: Organizational Structure[J]. IEEE Transactions on SMC, 2002, 32(3): 360- 375.

[7] Yu F L, Levchuk G M, et al. A Novel Congruent Organizational Design Methodology Using Group Technology and a Nested Genetic Algorithm[C]// Proceedings of the 9th Command & Control Research & Technology Symposium, CA, 2004.

[8] 崔逊学. 多目标规划算法及其应用[M]. 北京: 国防工业出版社, 2006: 109.

[9] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-Ⅱ[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182- 197.

[10] Srinivas N, Deb K. Multiobjective Function Optimization Using Nondominated Sorting Genetic Algorithms[J]. Evol. Comput., 1995, 2(3): 221 - 248.

[11] 云庆夏, 黄光球, 王战权. 遗传算法和遗传规划——一种搜索寻优技术[M]. 北京: 冶金工业出版社, 1997.

[12] Xiu B X, et al. A Novel Organizational Design Methodology Based on the Theory of Information Granulation[C]// Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 2005: 1- 6.

[13] 夏蔚军, 吴智铭, 张伟, 等. 微粒群优化在 Job2shop 调度中的应用[J]. 上海交通大学学报, 2005, 39(3): 381- 385.

[14] 张维存, 郑丕谔, 吴晓丹. 基于蚁群粒子群算法求解多目标柔性调度问题[J]. 计算机应用, 2007, 27(4): 936- 941.