

文章编号:1001-9081(2009)06-1527-02

## 柔性工时约束下项目调度及其蚁群算法

应 瑛, 寿涌毅

(浙江大学 管理学院, 杭州 310058)

(yshou@zju.edu.cn)

**摘 要:**针对软件工程项目调度问题,在考虑加班工时的情况下,提出了柔性工时约束下项目调度问题的数学模型,并设计了相应的蚁群算法。模型对项目人力资源的特殊性进行了分析,指出项目人力资源是一种特殊的可更新资源,在允许加班的情况下,人力资源构成特殊的柔性工时约束。针对所设计的数学模型,在并行项目进度生成机制基础上设计了蚁群算法,并通过算例进行验证与分析。

**关键词:**项目调度;工时约束;蚁群算法

**中图分类号:** TP311.5; TB114.1 **文献标志码:** A

### Ant colony algorithm for project scheduling problem under flexible work-hour constraints

YING Ying, SHOU Yong-yi

(School of Management, Zhejiang University, Hangzhou Zhejiang 310058, China)

**Abstract:** A new mathematical model for software engineering project scheduling was proposed. The model took into consideration the over-time of software engineers, which was regarded as a flexible work-hour constraint. The project human resources were analyzed in detail. As a type of special renewable resource, the human resource was added as a new constraint to the traditional resource constrained project scheduling problem. An ant colony algorithm based on parallel schedule generation scheme was suggested to solve this new flexible work-hour constrained project scheduling problem, and an instance was used to verify and analyze the solution.

**Key words:** project scheduling; work-hour constraint; ant colony algorithm

## 0 引言

对于软件工程项目而言,项目人力资源调配是非常重要的问题,直接影响项目进度。资源受限项目调度问题(Resource-Constrained Project Scheduling Problem, RCPSP)从资源角度对这一类问题进行了描述,并给出了主要的求解方法<sup>[1,2]</sup>。多数 RCPSP 问题主要针对可更新资源进行研究<sup>[3]</sup>。对于多数工程项目,尤其是软件工程项目而言,人力资源正是最关键的可更新资源。与一般的可更新资源不同,人力资源有一定的加班余量。因此,其相应的项目调度问题需要重新进行界定,也需要相应的算法进行求解。本文从这一角度出发,对具有上述柔性约束的项目调度问题进行了刻画,定义了数学模型,并给出了相应的求解算法。

## 1 柔性工时约束下项目调度模型

一个项目可以表示为一个节点式(Activity-On-Arrow, AOA)有向网络,并存在两类约束:一类是任务之间存在的逻辑约束,用 AOA 网络中的箭头表示;另一类是有限资源造成的资源约束,限定同一时刻各任务对资源的需求不能超过资源的供应量<sup>[1]</sup>。假设项目包含  $J$  个任务,需要  $K$  种可更新资源,其中第  $k$  种资源的供应量为  $R_k$ 。项目的第  $j$  个任务,其工期为  $p_j$ ,对第  $k$  种资源的需求量为  $r_{jk}$ ,其所有紧前任务的集合记为  $P_j$ ,任务的开始时间标记为  $ST_j$ ,结束时间标记为  $CT_j$ 。在时间  $t$ ,所有进行中任务的集合标记为  $I_t$ 。可以用任务开始时间

向量表示一个项目进度计划  $S = (ST_1, ST_2, \dots, ST_J)$ 。因此,经典的 RCPSP 可以描述为:

$$\min CT_J \quad (1)$$

$$\text{s. t. } ST_j \geq CT_h; \forall j, h \in P_j \quad (2)$$

$$\sum_{j \in I_t} r_{jk} \leq R_k; \forall k, t \quad (3)$$

其中:式(1)表示项目的目标函数,即最小化项目总工期;式(2)表示任务之间的紧前关系,一项任务必须在所有紧前任务完成后才可以开始;式(3)表示资源约束,限定任意时刻项目资源总需求不得超出资源供应量。本文主要分析软件工程项目中的人力资源,因此下文中涉及到的资源均只考虑人力资源,且均以工时为测量单位。

通常,第  $k$  种资源在单位时段内的加班工时  $R_k^T$  为标准工时的一个比例:

$$R_k^T = \mu R_k \quad (4)$$

一般而言,从工作效率和员工权益角度出发,需要对加班工时上限进行限制,例如 SA8000 标准明确规定每周加班不超过 12 小时<sup>[4]</sup>。因此,可以认为加班强度  $\mu$  必须小于最大加班强度  $\mu^+$ ,即  $\mu \in [0, \mu^+]$ ,也即:

$$R_k^T \leq \mu^+ R_k \quad (5)$$

对于上述加班工时受限的项目调度问题,需要寻找合适的进度计划  $S$  使各资源的加班时间不超过式(5)设定的上限,保证项目进度计划同时满足紧前关系与资源约束,并在尽可能短的时间内完成项目。

因此,这一类具有实践指导意义的柔性工时约束下项目

收稿日期:2008-12-15;修回日期:2009-02-25。 基金项目:国家自然科学基金资助项目(70401017)。

作者简介:应瑛(1978-),女,浙江金华人,博士研究生,主要研究方向:软件项目管理; 寿涌毅(1974-),男,浙江诸暨人,副教授,博士,主要研究方向:项目管理、运作管理。

调度问题 (Flexible Work-hour Constrained Project Scheduling Problem, FWCPSP) 可以描述为:

$$\min CT_j \quad (6)$$

$$\text{s. t. } ST_j \geq CT_h; \forall j, h \in P_j \quad (7)$$

$$\sum_{j \in I_t} r_{jk} \leq R_k + R_k^T; \forall k, t \quad (8)$$

$$R_k^T \leq \mu^+ R_k, \forall k \quad (9)$$

其中,式(7)表示了任务之间的紧前关系;式(8)、(9)则表示项目的柔性工时约束。

## 2 蚁群算法

由于蚁群算法被广泛应用于各种组合优化问题,也已应用于求解 RCPSP 问题<sup>[5]</sup>,因此本文应用蚁群算法求解 FWCPSP 问题。项目调度蚁群算法主要基于进度生成机制 (schedule generation scheme, SGS) 进行求解,即通过逐步扩展局部进度计划来生成一个完整可行的项目进度计划,并通过反复搜索获得最好解<sup>[6]</sup>。

本文采用并行进度生成机制 (Parallel SGS, PSGS)<sup>[1]</sup> 设计蚁群算法。蚁群中的蚂蚁  $k$  从第 1 个时间点开始搜索。在每一阶段  $g$ ,其时间点为  $t_g$ ,当前已完成任务集合记为  $C_g$ ,包括所有到时间点  $t_g$  为止已完成的任務,正在进行的任务集合  $I_g$  包括所有到时间点  $t_g$  为止正在进行的任务。可执行任务集合  $D_g$  包括所有满足紧前关系和资源约束并可以在时间点  $t_g$  开始的任務。蚂蚁  $k$  从  $D_g$  中选择任务  $j$  的概率为:

$$P_{ij}^k(g) = \begin{cases} \frac{\tau_{ij}(g)^\alpha \times \eta_{ij}^\beta}{\sum_{h \in D_g} \tau_{ih}(g)^\alpha \times \eta_{ih}^\beta}, & j \in D_g \\ 0, & j \notin D_g \end{cases} \quad (10)$$

其中: $\tau_{ij}$  为信息素信息, $\eta_{ij}$  为启发式信息, $\alpha$  和  $\beta$  为控制两类信息权重的参数。

对于 FWCPSP 而言,可行集  $D_g$  依赖于加班强度  $\mu$ 。一般而言,随着  $\mu$  的增加,当前可用资源数量增加,因此可行集中包含的任务数量随之增加,使得蚁群有更大的搜索空间来寻找可行解。

蚂蚁  $k$  逐一从可执行任务集  $D_g$  中选取任务,并安排该任务在时间  $t_g$  开始。然后将该任务剔除出可执行任务集合  $D_g$ ,并继续选取和安排任务,直到可执行任务集合  $D_g$  为空。之后,蚁群进入 PSGS 的下一个阶段,从时间点  $t_{g+1}$  开始更新三个任务计划  $C_g, I_g, D_g, t_{g+1}$  为所有在阶段  $g$  中安排的全部任务的最早结束时间。

蚁群算法的效率受式(10)中的信息素信息与启发式信息两个参数影响。启发式信息表示蚂蚁在搜索决策中可以利用的直观信息,如 TSP 中候选节点的距离。在项目调度问题中,一般用优先规则来构造启发式信息,如常见的最迟完工 (Late Finish, LF) 时间<sup>[7]</sup>:

$$\eta_{ij} = \max_{h \in D_g} LF_h - LF_j + 1 \quad (11)$$

信息素是蚁群在搜索过程中逐步积累的历史信息,有效利用信息素信息可以加速蚁群的学习和搜索。其更新机制是蚁群算法的重点,包括两个部分,即信息素的挥发和累积。在蚁群完成一次完整的搜索后,整个信息素矩阵的更新可采用如下机制:

$$\tau_{ij}(g+1) = (1-\rho) \times \tau_{ij}(g) + \rho \times \frac{1}{CT_j} \quad (12)$$

其中: $\rho$  为信息素的挥发率,蚁群所得调度方案的总工期越小,相应路径增加的信息素就越多。

当蚂蚁  $k$  从第 1 个任务逐步搜索到第  $J$  个任务,就构成一个完整的任务列,而每个阶段的时间点  $t_g$  则成为该阶段所分配各任务的开始时间。因此,蚂蚁  $k$  就得到一个可行的项目进度计划。如果蚁群在已有信息素基础上反复进行搜索,并不断更新信息素矩阵,就可逐步获得最好进度计划<sup>[7]</sup>。

因此,对于 FWCPSP 可以设计如下的蚁群算法:

```

input FWCPSP instance
initialize coefficients
set pheromone matrix  $\tau_{ij} = 0$ 
for  $\mu = 0 \rightarrow \max\_mu$ 
  for every ant generation
    for every ant colony
      for every ant  $k$ 
        for every stage  $g$ 
          calculate  $t_g$ 
          estimate  $D_g$ 
          while  $D_g$  is not empty
            select  $j$  from  $D_g$  according to 式(10)
            set  $ST_j = t_g$ 
            update  $D_g$ 
          end while
        end for
      end for
    record current schedule  $S$ 
    if  $S$  is better than  $S^*$ 
      update  $S^*$ 
    end if
    update pheromone  $\tau_{ij}$  by 式(12)
  end for
end for
output  $S^*$ 

```

## 3 算例分析

本文选取文献[8]中的一个项目作为实例进行分析。该项目共有 9 个任务,包括虚的启动任务 A 和结束任务 I,项目只涉及 1 种人力资源,每周可用工时为 40 小时,该项目的 AON 图如图 1 所示。

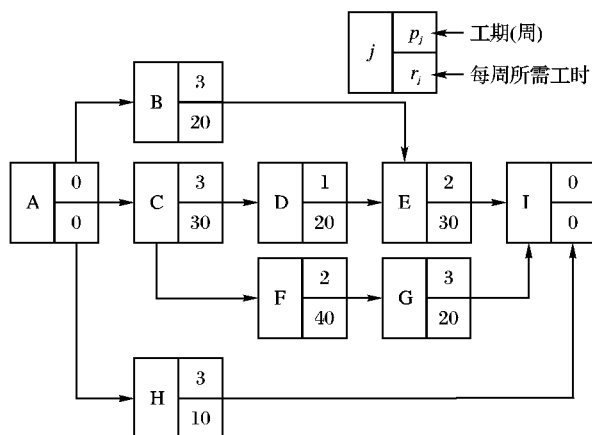


图 1 项目实例

在不考虑加班的情况下,该问题为一个典型的 RCPSP 问题,其最优项目进度计划如图 2 所示,所需总工期为 11 周。

假如项目允许加班,每周最多加班 10 小时,即最大加班系数为 0.25。设定加班强度的步长为 0.05,利用上述蚁群算法逐步求解。在加班系数小于等于 0.20 时,项目最短总工期

(下转第 1568 页)

地调整。试验数据表明,变异区间自适应调整的遗传算法在几种测试函数上均优于基本遗传算法,能有效提高遗传算法的搜索速度和精度。变异区间自适应调整的遗传算法为解决高维函数的优化问题及优化算法的改进和发展提供了一条新的思路。

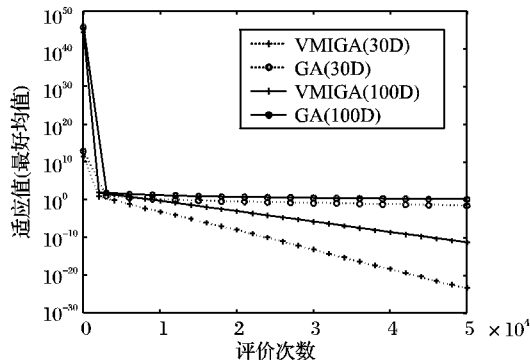


图2 VMIGA 和 GA 在 30 维和 100 维的  $f_1$  上的比较

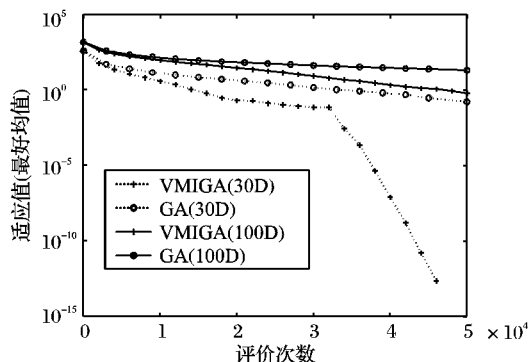


图3 VMIGA 和 GA 在 30 维和 100 维的  $f_2$  上的比较

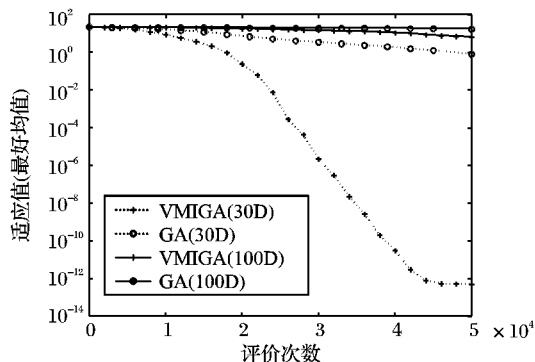


图4 VMIGA 和 GA 在 30 维和 100 维的  $f_3$  上的比较

#### 参考文献:

- [1] JOHN H. Adaptation in natural and artificial systems [M]. Ann Arbor, MI: The University of Michigan Press, 1975.
- [2] PAUL S H. Evolution and optimization seeking [M]. New York: John Wiley & Sons, 1995.
- [3] ZBIGNIEW M. Genetic algorithms + data structures = Evolutionary programs [M]. Berlin: Springer-Verlag, 1992.
- [4] PANZHENG - JUN, KANG LI - SHAN, NIE SI - XIANG. Evolving both the topology and weights of neural networks [J]. Parallel Algorithms and Applications, 1996, 9(3/4): 299 - 307.
- [5] HEINZ M, DIRK S V. Predictive models for the breeder genetic algorithm [J]. Evolutionary computation, 1993, 1(1): 25 - 49.
- [6] 钟伟才, 刘静, 焦李成. 遗传算法中自适应伸缩搜索空间的方法 [J]. 系统工程与电子技术, 2004, 26(2): 245 - 247.
- [7] 潘正君, 康立山, 陈毓屏. 演化计算 [M]. 北京: 清华大学出版社, 1998.
- [8] 刘静, 钟伟才, 刘芳, 等. 组织进化数值优化算 [J]. 计算机学报, 2004, 27(2): 157 - 167.
- [9] 王欣星, 周晖. 基于混合遗传-模拟退火算法的日前购电策略研究 [J]. 水电能源科学, 2007, 25(5): 122 - 126.

(上接第 1528 页)

保持为 11 周,无法利用加班压缩项目总工期。当加班系数为 0.25 时,所得最优解如图 3 所示,其项目总工期为 10 周。可见该项目如在第 1~3 周及第 6 周安排加班 10 个工时,则能保证项目提前 1 周完成。

因此,对于 FWCPSP 问题而言,在允许适当加班的情况下,如何加速项目进度以实现最小化工期就成为一个新的问题。本文所建立的模型与算法有助于求解此类问题。

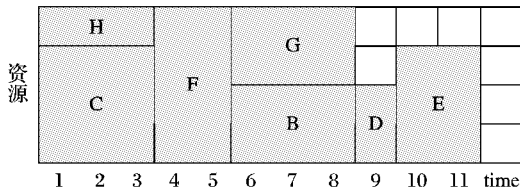


图2 最优项目进度计划

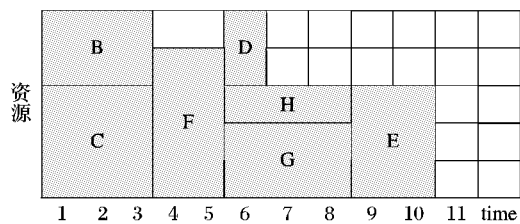


图3 允许加班情况下的项目进度计划

## 4 结语

本文结合软件工程项目实践情况,提出了一类新的项目调度问题,即在考虑加班的情况下如何合理调配人力资源使得项目在最短工期内完工。同时基于并行项目进度机制设计

了将蚁群算法用于求解这一类柔性工时约束下的项目调度问题。本文研究尚可以进一步深入,例如尚未考虑加班费,如果考虑到加班费则还需要在支付加班费和提前完工之间进行权衡,这也将是后续研究的主要方向。

#### 参考文献:

- [1] KOLISCH R, SPRECHER A, DREXL A. Characterization and generation of a general class of resource constrained project scheduling problems [J]. Management Science, 1995, 41(10): 1693 - 1703.
- [2] VALIS V, BALLESTIN F, QUINTANILLA S. Justification and RCPS: A technique that pays [J]. European Journal of Operational Research, 2005, 165(2): 375 - 386.
- [3] ALVAREZ-VALDES R, TAMARIT J M. Heuristic algorithm for a resource-constrained project scheduling: A review and an empirical analysis [C]// Advances in Project Scheduling. Amsterdam: Elsevier, 1989: 113 - 134.
- [4] 胡明娟, 丁建定. 全球化、SA8000 与我国企业社会责任标准的建立 [J]. 华中科技大学学报: 社会科学版, 2005, 19(5): 115 - 120.
- [5] 刘士新, 宋健海, 唐加福. 蚁群最优化: 模型、算法及应用综述 [J]. 系统工程学报, 2004, 19(5): 496 - 502.
- [6] HARTMANN S, KOLISCH R. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem [J]. European Journal of Operational Research, 2000, 127(2): 394 - 407.
- [7] MERKLE D, MIDDENDORF M, SCHMECK H. Ant colony optimization for resource-constrained project scheduling [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 333 - 346.
- [8] PATTERSON J H. A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem [J]. Management Science, 1984, 30(7): 854 - 867.