文章编号: 1001-0920(2001)01-0111-03

一种求解多执行模式资源水平问题的遗传算法

刘士新, 王梦光, 芦宙新
(东北大学 信息科学与工程学院, 辽宁 沈阳 110006)

摘 要: 针对资源受限情况下多执行模式工程调度中资源水平问题的特点, 设计了一种遗传算法。解的编码采用满足紧前关系的工作链表与工作执行模式链表结合的双链表结构, 交叉算子采用修正的一点交叉算法。为保证收敛解的可行性, 在适值函数计算时对不可行解进行惩罚。对标准问题库 PSPLIB 中大量问题的求解实验结果表明, 遗传算法是求解该问题的一种有效算法。

关键词: 工程调度; 资源限制; 多执行模式; 资源水平; 遗传算法
中图分类号: O 224 **文献标识码**: A

Genetic Algorithm for Resource Levelling Problem in Multi-mode Project Scheduling

LIU Shi-xin, WANG Meng-guang, LU Zhou-xin
(School of Information Science and Engineering, Northeastern University, Shenyang 110006, China)

Abstract: A genetic algorithm for resource levelling problem in multi-mode project scheduling is developed. An individual is represented by a pair of precedence feasible activity sequence list and mode assignment list. Modified one-point crossover is taken as the crossover operator. To ensure the feasible solution, infeasible solutions in calculating fitness value is punished. The experiment results show that the genetic algorithm is effective for resource levelling problem.

Key words: project scheduling; resource constraint; multi-mode; resource levelling; GA

1 引 言

资源水平问题(RLP)是工程调度问题的一个重要分类, 它可分为资源不受限和资源受限两种情况。早期的研究主要针对前者, 近几年才开始对后者的研究。文献[1]基于求解 RCPSP 的分支定界算法, 设计出一种精确算法; 文献[2]则提出一种基于优先规则的启发式算法, 并利用大量问题实例对算法进行测试, 得到了较好的结果。但迄今为止, 关于 RLP 的研究都是针对单执行模式的情况, 对于多执行模式RLP(MRLP)的研究还是空白。为此, 本文针对 MRLP 的特点, 设计了一种有效的遗传算法。

2 问题描述

MRLP 问题可描述如下: 一个工程中包含 J 项工作, 由于技术上的要求, 某些工作之间存在着紧前关系, 例如工作 j 在其全部紧前工作 i($i \in P_j$, P_j 为工作 j 的紧前工作集) 完成之前不能开始。整个工程的结构由一张有向网络图表示, 图中节点代表工作,

收稿日期: 1999-09-03; **修回日期**: 1999-10-28
基金项目: 中国科学院现代制造 CAD/CAM 技术开发实验室资助项目(9904)
作者简介: 刘士新(1968—), 男, 辽宁铁法人, 讲师, 博士, 从事生产计划与调度、智能优化方法等研究; 王梦光(1936—), 女, 吉林省吉林市人, 教授, 博士生导师, 从事生产计划与调度、智能优化方法等研究。

弧线代表工作间紧前关系。工作1是唯一最早开始的工作,工作J是唯一最晚完成的工作,均为虚工作(不消耗资源且执行时间为0),表示整个工程的开始和结束。

工作j($j=1,2,\cdots,J$)必须选择M_j种执行模式之一执行,且在执行过程中不能中断或改变执行模式。在第m($1\le m\le M_j$)种模式下执行工作j需要第k种可更新资源量为r^{ρ}_{jmk},需要第n种不可更新资源量为r^{ν}_{jmn},执行时间为d_{jm},工程底线工期为$\overline{T}$。根据各工作在各种执行模式下的最短执行时间,利用传统的PERT/CPM时间参数计算方法,可计算出各工作的最早、最晚完成时间窗口$[EF_j, LF_j]$。第k种可更新资源单位成本为$c\rho_k$,整个工期各阶段内可用量为常量R^{ρ}_k($k=1,2,\cdots,K$),第n种不可更新资源单位成本为$c\nu_n$,整个工期内可用量为R^{ν}_n($n=1,2,\cdots,N$)。

引进如下决策变量

$$x_{jmt}=\begin{cases}1, & \text{如果工作 } j \text{ 选择第 } m \text{ 模式执行且在第 } t \text{ 阶段完成}\\ 0, & \text{否 则}\end{cases}$$

设$R^{\rho\max}_k$为整个工期内可更新资源k的最大消耗水平,$R^{\nu u}_n$为整个工程中不可更新资源n的消耗量,即

$$R^{\rho\max}_k=\max_{t=1,\cdots,\overline{T}}\left(\sum_{j=1}^{J}\sum_{m=1}^{M_j}r^{\rho}_{jmk}\sum_{q=\max\{t,EF_j\}}^{\min\{t+d_{jm}-1,LF_j\}}x_{jmq}\right)$$

$$R^{\nu u}_n=\sum_{j=1}^{J}\sum_{m=1}^{M_j}r^{\nu}_{jmn}\sum_{t=EF_j}^{LF_j}x_{jmt}$$

则MRLP可建立如下数学模型

$$\min \sum_{k=1}^{K}c\rho_kR^{\rho\max}_k+\sum_{n=1}^{N}c\nu_nR^{\nu u}_n \tag{1}$$

$$\text{s.t.}\quad \sum_{m=1}^{M_j}\sum_{t=EF_j}^{LF_j}x_{jmt}=1,\quad j=1,2,\cdots,J \tag{2}$$

$$\sum_{m=1}^{M_i}\sum_{t=EF_i}^{LF_i}tx_{imt}\le\sum_{m=1}^{M_j}\sum_{t=EF_j}^{LF_j}(t-d_{jm})x_{jmt}$$

$$j=2,3,\cdots,J,\quad i\in P_j \tag{3}$$

$$R^{\rho\max}_k\le R^{\rho}_k,\quad k=1,2,\cdots,K \tag{4}$$

$$R^{\nu u}_n\le R^{\nu}_n,\quad n=1,2,\cdots,N \tag{5}$$

$$x_{jmt}\in\{0,1\},\quad j=1,2,\cdots,J$$

$$m=1,2,\cdots,M_j,\quad t=EF_j,\cdots,LF_j \tag{6}$$

式(1)为目标函数,代表整个工程的资源消耗费用最小;式(2)表示每项工作只能在工期内以一种执行模式完成一次;式(3)代表紧前关系约束;式(4)保证在每一阶段使用的可更新资源量不能超过其可用量;式(5)保证整个工程消耗的不可更新资源量不能大于其投入总量;式(6)规定了变量的取值范围。一个可行计划是指各项工作的执行模式和开始时间都已确定,且满足约束(2)～(5)的计划。最优计划是使整个工作的资源消耗费用最小的可行计划。

3 求解MRLP的遗传算法

算法首先生成种群规模为偶数PopSize的初始种群POP_0,并随机地将POP_0中的个体两两匹配,经过交叉算子操作生成PopSize个后代,然后对这些后代利用变异算子进行变异,并将其加入到POP_0中,这样便产生规模为$2\times$PopSize的种群。按照适值计算公式计算每个个体的适值,从中选择适值最好的PopSize个个体形成下一代种群。算法重复此过程GEN次结束。

3.1 解的编码方式及适值计算方法

该算法中解的编码方式如表1所示。其中,$\{j^I_1, j^I_2,\cdots,j^I_J\}$是满足紧前关系的工作链表,即$\{j^I_1,j^I_2,\cdots,j^I_J\}=\{1,2,\cdots,J\}$且$P_{j^I_i}\subseteq\{j^I_1,j^I_2,\cdots,j^I_{i-1}\}$;$m^I(j^I_i)$是为工作$j^I_i$指定的执行模式。解码过程就是根据各工作排列顺序和被指定执行模式,按照串行调度方案[3]生成可行计划。

表1 解的编码方式

j^I_1	j^I_2	$\cdots$	j^I_J
$m^I(j^I_1)$	$m^I(j^I_2)$	$\cdots$	$m^I(j^I_J)$

由于工程工期$\overline{T}$和不可更新资源的限制,按以上编码方式和解码规则生成的解(计划)不能保证全部可行,因此在算法中对不可行解进行惩罚。设编码对应计划的完工期为C,W_1和W_2为正数,则编码的适值计算公式为

$$\begin{aligned}f(s)=&\sum_{k=1}^{K}c\rho_kR^{\rho\max}_k+\sum_{n=1}^{N}c\nu_nR^{\nu u}_n+\\&W_1\times\max\{0,C-\overline{T}\}+\\&W_2\times\sum_{n=1}^{N}\max\{0,R^{\nu u}_n-R^{\nu}_n\}\end{aligned}\tag{7}$$

式(7)右边第1项为可更新资源费用,第2项为不可更新资源费用,第3项为拖期惩罚,第4项为破坏不可更新资源约束惩罚。

初始种群生成过程为:随机为各工作选定一执行模式,然后基于SRD优先规则生成满足紧前关系的工作链表。重复此过程PopSize次,产生初始种群。

3.2　遗传算子

交叉算子采用两点交叉算法, 即随机地从当代种群中选择两个个体, 记为母亲 M 和父亲 F, 生成两个随机整数 $p_1, p_2, 1 < p_1 < p_2 < J$, 由 M 和 F 通过在 p_1 和 p_2 的两点交叉运算产生两个后代, 分别为女儿 D 和男儿 S。在 D 的编码中, 前 p_1 项工作继承于 M, 即

$$j_i^D := j_i^M, \quad i = 1, 2, \cdots, p_1$$

$$m^D(j_i^D) := m^M(j_i^D), \quad i = 1, 2, \cdots, p_1$$

$i = p_1 + 1, \cdots, p_2$ 位置的工作则继承于 F, 并保持 F 中各工作的相对位置, 即

$$j_i^D := j_k^F, \quad i = p_1 + 1, \cdots, p_2$$

$$k = \min\{k \mid j_k^F \notin \{j_1^D, \cdots, j_{i-1}^D\}, \ k = 1, 2, \cdots, J\}$$

$$m^D(j_i^D) := m^F(j_i^D), \quad i = p_1 + 1, \cdots, p_2$$

$i = p_2 + 1, \cdots, J$ 位置的工作再继承于 M, 并保持 M 中各工作的相对位置, 即

$$j_i^D := j_k^M, \quad i = p_2 + 1, \cdots, J$$

$$k = \min\{k \mid j_k^M \notin \{j_1^D, \cdots, j_{i-1}^D\}, \ k = 1, 2, \cdots, J\}$$

$$m^D(j_i^D) := m^M(j_i^D), \quad i = p_2 + 1, \cdots, J$$

S 的工作链表形成过程与 D 相似, 但继承顺序则相反。

变异算子的操作过程分两步进行:

1) 对 $i = 2, 3, \cdots, J - 1$, 根据变异概率 P_{m1} 交换工作 j_i^I 和 j_{i+1}^I, 如果交换后的工作序列不满足紧前关系约束, 则恢复原来位置;

2) 对 $i = 2, 3, \cdots, J - 1$, 根据变异概率 P_{m2} 随机地改变工作 j_i^I 的执行模式。

4　实验结果

第 3 节介绍的遗传算法应用 Java 语言编写, 运行于 Windows98 操作系统的 Pentium 133/32M IBM 兼容机上。初始参数如下: 种群规模 PopSize = 20, 30, 40, 变异概率 $P_{m1} = P_{m2} = 0.05$, 迭代代数与 PopSize 相对应, 分别为 GEN = 50, 33, 25。

利用问题库 PSPLIB[4] 中工作数为 10 和 20(不包括虚工作) 的两组问题对算法进行测试。每组问题中有两种可更新资源和两种不可更新资源, 每项工作有 3 种执行模式, 执行时间是[1, 10] 之间的随机整数。两种问题中工程工期的上限 $\bar{T}$ 均为无资源约束情况下最短工期的 1.5 倍。

由于无法知道这些问题的最优解, 所以把每个问题在各种 PopSize 和 GEN 组合下求得的最好解作为比较标准。两组问题的实验结果列于表 2, 其中平均偏差为最后一代种群中最坏解与最好解目标函数值相对差的均值。

表 2　两组问题的实验结果

种群规模	迭代代数	$J = 12$ 平均偏差	$J = 12$ 最好解比例	$J = 12$ CPU (s)	$J = 22$ 平均偏差	$J = 22$ 最好解比例	$J = 22$ CPU (s)
20	50	2.2%	98%	5.8	3.8%	97%	10.2
30	33	2.8%	100%	5.5	4.7%	100%	9.8
40	25	3.7%	99%	6.1	5.3%	99%	11.0

由表 2 可见, 在 PopSize 和 GEN 的 3 种组合下, 最后一代种群中最坏解与最好解目标函数值相对差均小于 6%; 在 PopSize = 30, GEN = 33 的组合下, 求得最好解的比例最大, 而运行时间不超过 11s。

5　结　　论

本文针对 MRLP 的特点, 设计了一种遗传算法。采用该算法对标准问题库 PSPLIB 中大量问题进行求解实验结果表明, 遗传算法是求解该问题的一种有效算法。该算法可在以下几方面进行改进: 1) 设计更好的交叉和变异算子, 改进后代种群的生成方法, 从而改进解的质量; 2) 改变目标函数的形式, 求解多目标的 MRLP。

参考文献:

[1] Demeulemeester E. Minimizing resource availability costs in time-limited project networks[J]. Management Science, 1995, 41(10): 1590-1598.

[2] Neumann K, Zimmermann J. Resource levelling for projects with schedule-dependent time windows[J]. European J of Operational Research, 1999, 117(3): 591-605.

[3] Kolisch R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation[J]. European J of Operational Research, 1996, 90(2): 320-333.

[4] Kolisch R, Sprecher A. PSPLIB——A project scheduling problem library[J]. European J of Operational Research, 1996, 96(2): 205-216.