

基于改进遗传算法的关键链项目进度计划优化

张琦, 廖良才, 王卫威

(国防科学技术大学 信息系统与管理学院, 湖南 长沙 410073)

摘要:项目进度计划一直是项目管理中的重点和难点,其原因是在资源和工序约束的双重约束下,项目进度计划的求解是一个 NP 难问题,尤其是在多资源约束下更增加了问题的复杂度。文中从关键链这一角度出发,分析了该问题并建立模型。应用自适应遗传算法对模型进行求解,对遗传算法的编码、选择、交叉、变异等操作进行了设计,并在此基础上对传统的遗传算法进行改进。最后,通过案例的求解,验证了关键链方法在研究多资源约束项目进度计划问题上的可行性,以及自适应遗传算法在求解该问题上的优越性。

关键词:项目进度计划;多资源约束;关键链;遗传算法;自适应遗传算法

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2014)04-0001-05

doi:10.3969/j.issn.1673-629X.2014.04.001

Critical Chain Project Schedule Optimization Based on Improved Genetic Algorithm

ZHANG Qi, LIAO Liang-cai, WANG Wei-wei

(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract: Project schedule has been a focus and difficulty of project management, the reason lies in the dual constraints of resources and processes. To solve the project schedule is an NP-hard problem under the constraints, especially in multi-resource constraints adds to the problem complexity. It adopts critical chain theory and method to analyze the problem and construct the model. Apply adaptive genetic algorithm to solve the problem. Genetic coding, selection, crossover and mutation are discussed, and on this basis, the traditional genetic algorithm is improved. Finally, based on the experiment analysis of the instance, verify the validity of the critical chain method in the study of multi-resource project scheduling problem, as well as the feasibility of adaptive genetic algorithm to solve the problem.

Key words: project scheduling; multi-resource constraints; critical chain; genetic algorithm; adaptive genetic algorithm

0 引言

随着世界经济社会的发展,项目管理扮演的角色日趋重要,而项目的进度管理一直是项目管理的重点和难点。在竞争越来越激烈、资源限制日趋明显的今天,传统的项目计划方法如甘特图只是单纯地表明工序的起始时间,并不能表明工序间的相互关系;而 CPM 和 PERT 已不能解决资源约束下的网络规划问题^[1]。另外,传统的项目计划制定方法中也没有将人和环境的因素考虑进去,由于人的惰性等心理行为因素造成的项目延期也使得成本增加,对组织效益产生巨大影响。

针对项目管理问题,Goldratt 博士将约束理论^[2]应用于项目管理为项目管理开创了一种新的方法,即关键链项目管理(CCPM)^[3]。综合考虑了时间、资源以及人的行为等因素,通过设置项目缓冲(PB)、汇入缓冲(FB)和资源缓冲(RB)的方式将安全时间综合利用起来,解决了困扰在项目管理中的不确定性问题。许多学者也对此方法进行了深入的研究,Leach^[4]认为应该找出多个项目共享的关键资源作为约束因素,从而对不同优先权的项目进行不同的管理。Tukel^[5]等综合考虑了项目中资源供应与使用状况和项目的复杂度,提出了资源紧张度自适应程序和密度自适应程序。

收稿日期:2013-07-09

修回日期:2013-10-12

网络出版时间:2014-01-28

基金项目:国家自然科学基金资助项目(61201337)

作者简介:张琦(1988-),女,辽宁阜新人,硕士研究生,研究方向为项目管理、系统优化与系统集成;廖良才,教授,博士,研究方向为系统优化与系统集成、装备采办与项目管理、知识集成与知识管理;王卫威,讲师,博士,研究方向为系统优化与系统集成。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140128.1133.012.html>

徐哲等^[6]对关键链技术的项目进度管理研究进行了综述。文献[7-8]对关键链项目进度管理的定量分析方法进行了研究。文献[9]对关键链在项目管理中的应用进行了探索。杨莉等^[10]提出了一种改进的基于三角模糊数的关键链管理方法。

多资源背景下关键链的求解是一个 NP 难问题。解决这一问题的方法主要可分为两种,即精确算法和启发式算法^[11]。精确算法可以得到最优解,但随着资源种类和数量的增多、项目复杂度的加大,往往精确算法要花费的时间相当长,在实际项目中就缺少了应用价值。

而基于一定启发式规则的启发式算法,与精确算法不同,其特点是速度快,启发式算法通常能够得到较为理想的解,但不一定能够得到最优解,其与理想解之间往往存在着一定的偏差。

文中采用了改进的遗传算法来解决这一问题,并通过改进算法缩小理想解与最优解的偏差,制定科学合理的进度计划。

1 问题描述

多资源项目进度计划问题主要研究在满足资源约束和工序约束的前提下,通过合理安排工序的开始时间和结束时间,来达到一定的优化目标。而项目工期往往是项目管理者追求的最主要目标,文中即是在给定项目和资源相关信息条件下,综合考虑各种约束,以尽可能少的时间完成项目任务。多资源项目进度计划的数学模型描述如下:

在项目进度计划问题中,共包含着 r 种资源和 n 个工序, $R = \{R_1, R_2, \dots, R_r\}$ 表示资源的集合, $N = \{1, 2, \dots, n\}$ 表示所有工序的集合。

$ST(i)$ 表示工序 i 的开始时间, $ET(j)$ 表示工序 j 的完工时间。 E 为工序 i 的紧前工序的集合。 A_i 表示在 t 时刻进行的工序的集合, B_t 表示在 t 时刻等待开始的工序的集合,工序 i 的完成需要资源 R_k 的数量为 r_{ik} , R_{kt} 表示在时刻 t 资源 R_k 的可用量。 T 表示项目的总工时,也就是项目包含的所有工序均执行完毕的时间。则可建立如下模型:

$$\min T = \{\min t \mid B_t = \emptyset; A_t = \emptyset\} \quad (1)$$

s. t.

$$ST(i) > ET(j) \quad j \in E, i \notin E \quad (2)$$

$$\sum_{j \in A_t} r_{jk} \leq R_{kt} \quad k = 1, \dots, r; t = 1, \dots, T \quad (3)$$

其中,公式(1)为目标函数,表示寻求项目总工期的最短;公式(2)表示项目工序间紧前紧后的约束关系;公式(3)表示任一时刻所有工序使用的某种资源的加和不能大于该资源总的可用量。

2 算法设计

遗传算法 (Genetic Algorithm) 是模拟自然界遗传机制和生物进化规律而产生的一种并行的随机搜索最优化方法。最初由美国 Michigan 大学的 Holland 教授在 1962 年提出。遗传算法是以达尔文的自然选择学说为基础发展起来的,在算法中引入了“优胜劣汰,适者生存”的生物进化原理。其基本思想是:首先对优化参数进行编码形成种群,按适应度函数对种群中的个体进行筛选,经过遗传中的复制、交叉及变异等操作进行筛选,这样适应度高的个体就被保留下来,组成了新的群体,新群体更优于上一代同时也继承了上一代的信息。如此周而复始,种群中个体的适应度不断地提高,直到满足一定的条件。遗传算法的算法简单,可并行处理,并能得到全局最优解。

自从 Holland 教授系统地提出遗传算法完整的理论和结构以来,众多的学者们一直致力于推动遗传算法的发展,从算法的编码方式、参数确定、选择方式和交叉方式等角度进行了深入探究,引入了自适应策略和动态策略以改进遗传算法的性能,提出了改进的遗传算法^[12]。

2.1 算法流程

算法以识别多项目关键链为目标,随机生成初始种群,以迭代次数为终止条件,以适应度为评价准则。

先建立项目进度计划问题的约束规划模型。采用路径表示方法,对项目进度计划进行染色体编码。依据交叉率选择 2 个染色体父体,然后采用部分映射交叉算子对 2 个父体进行交叉操作,得到 2 个子染色体个体。同样根据变异率,然后采用交换变异算子对染色体的两个位置进行变异操作,得到新的染色体,并依据适应度函数,比较新得到的染色体与原来的染色体。

根据染色体上有效基因的信息以及基因对应的任务的属性,计算每条染色体的适应度函数,选择适应度较大的个体作为下一代遗传操作的染色体父体。同时,为了保存种群中的优秀个体,还采用了精英策略。最后,若达到了最大的迭代次数,算法终止,否则重复算法步骤。其算法的流程图如图 1 所示。

2.2 自适应遗传算法

在遗传算法的参数中,交叉概率 P_c 和变异概率 P_m 的选择是影响遗传算法性能和行为的关键所在, P_c 越大,新个体产生的速度就越快,然而 P_c 过大时遗传模式被破坏的可能性也同样越大,使得具有较高适应度的个体很快就会被破坏;但如果 P_c 过小,就使得搜索过程缓慢,停滞不前。对于 P_m 如果过小,就不容易产生新的个体,如果 P_m 取值过大,那么遗传算法就变成了随机搜索算法。Srinivas 等提出了一种自适应遗传算法, P_c 和 P_m 能随着适应度自动改变。算法的基

本思想是:对适应度高于群体平均适应度的个体,赋以相对较低的 P_c 和 P_m , 使该解得以保护能够进入下一代;而对低于平均适应度的个体,赋以较高的 P_c 和 P_m , 使得该解被淘汰^[13-14]。

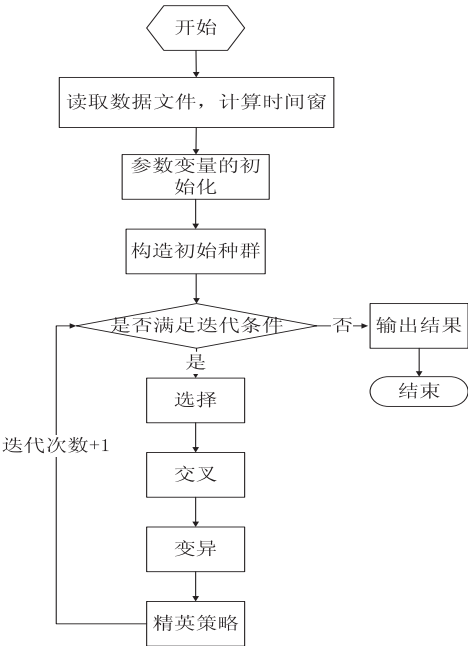


图1 算法流程图

在自适应遗传算法中, P_c 和 P_m 按如下的公式进行自适应地调整:

$$P_c = \begin{cases} k_1 \frac{(f_{\max} - f')}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_2, & f < f_{\text{avg}} \end{cases}$$
$$P_m = \begin{cases} k_3 \frac{(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_4, & f < f_{\text{avg}} \end{cases}$$

其中,
 f_{\max} 表示群体中最大的适应度值;
 f_{avg} 表示群体平均的适应度值;
 f' 表示进行交叉操作的两个个体中较大的适应度值;
 f 表示要进行变异操作个体的适应度值。

从上式可以看出, 当要进行操作的个体的适应度值越接近最大适应度值时, 其交叉率和变异率就越小, 而当其等于最大适应度值时, 交叉率和变异率为零。这种调整策略对群体处于进化后期时是比较合适的, 但在进化初期时是不利的, 原因在于进化初期时群体中的较优个体变化较小, 使进化走向局部最优解的可能性较大。因此, 可以作进一步的改进, 令群体中最大适应度值个体的交叉率和变异率分别为 P_{c2} 和 P_{m2} 。为了使每一代的最优个体不被破坏, 通过采取精英选择策略, 使它们能够直接复制到下一代。

改进后, P_{c2} 和 P_{m2} 按下式进行自适应调整:

$$P_{c2} = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f_{\max} - f')}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ P_{c1}, & f < f_{\text{avg}} \end{cases}$$
$$P_{m2} = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ P_{m1}, & f < f_{\text{avg}} \end{cases}$$

2.3 染色体的表示与评价

在文中的项目进度计划问题中, 采用路径表示方法。路径表示是表示进度计划序列的基因编码的最自然、最简单的一种表示方法。例如, 一个进度计划方案 Activity6→Activity2→Activity9→Activity7→Activity4→Activity1→Activity5→Activity8→Activity3, 可表示为 6-2-9-7-4-1-5-8-3。依据路径表示编码方法, 要求进度计划方案的染色体编码中不允许有重复基因码。

每条染色体表示一个进度计划方案, 染色体上的基因代表相应的工序, 并且这种映射是一一对应的。

适应度是用来度量染色体对于环境的适应程度, 适应度较高的染色体将会获得更多的繁殖机会, 而适应度较低的染色体其繁殖机会就较少, 甚至会逐渐灭绝。在该问题中的适应度是评价项目进度计划优劣的标准, 是有关项目总工期的函数, 因此为非负。对于给定的染色体, 适应度函数定义如下:

$$F \propto \frac{1}{T}$$

式中, T 表示项目的总工期。

2.4 选择和交叉算子

交叉算子采用部分映射交叉 (Partially Mapping Crossover, PMX) 操作。首先随机选取 2 个交叉点, 交换两个父代个体交叉点之间的片段, 而对交叉点以外的基因, 若它不与交换过来的片断冲突则保留下来, 若冲突则通过部分映射策略来确定直到没有冲突的基因, 从而获得后代染色体个体。PMX 的算子在一定程度上满足了 Holland 图式定理的基本性质, 子串能继承到父串的有效模式^[15]。

例如, 2 个父代个体分别为 P_1 : [0864 | 7952 | 31], P_2 : [0458 | 1276 | 39], 若交叉位置为 4 和 8, 则片段 |7952| 和 |1276| 将交换。对于 P_1 的剩余基因, 由于 8 不与 |1276| 冲突则直接填入, 6 存在冲突, 6 的映射基因为 2 仍存在冲突, 2 的映射基因为 9 不存在冲突, 则将 9 填入后代个体 Q_1 的相应位置。依此类推, 得到的后代个体 Q_1 : [0894 | 1276 | 35], 类似的可以得到 Q_2 : [0452 | 1876 | 93]。

变异算子通常包括替换式或扰动式变异^[15], 根据项目进度计划问题的特点, 要求设计的变异算子满足: 对任一染色体进行变异操作后, 产生的新个体要能对应一个具有实际意义的项目进度计划方案。因此, 文

中采用替换式变异中的交换变异操作。交换变异是指交换染色体中 2 个随机选取的基因,从而产生出一个新的进度计划序列。例如,对 Q_1 : [894 | 1276 | 35] 进行交换变异操作,基因 9 与基因 6 进行交换,得 Q_1' : [264 | 1873 | 95]。

3 应用案例

为了测试文中方法的可行性与算法的有效性,文中采用 Visual Studio C#语言编程进行实现。

文中选取的案例是一个典型的多资源项目调度问题,共包括三种资源和 34 个工序,资源总量为 $(R_1, R_2, R_3) = (9, 9, 9)$ 。项目信息如表 1 所示。

表 1 项目信息表

工序编号	紧后工序	工期	关键链法对	所需资源		
			工期估计	R_1	R_2	R_3
0	1,6,13,18,27	0	0	0	0	0
1	2,3	0	0	0	0	0
2	4,5	6	3	3	2	1
3	4	10	5	2	4	2
4	33	12	6	3	1	2
5	33	4	2	4	3	1
6	7,8	0	0	0	0	0
7	9	10	5	3	1	1
8	9	10	5	3	2	3
9	10,11	10	5	2	1	3
10	12	6	3	1	1	1
11	12	4	2	3	2	1
12	33	6	3	1	2	3
13	14	0	0	0	0	0
14	15,16	4	2	2	2	2
15	17	12	6	3	2	2
16	17	6	3	3	2	1
17	33	6	3	3	3	2
18	19,20	0	0	0	0	0
19	21,22	4	2	2	1	2
20	22,23	12	6	2	3	2
21	24	8	4	3	1	2
22	25	4	2	3	1	1
23	25	6	3	3	2	1
24	26	6	3	2	2	1
25	26	8	4	1	1	3
26	33	10	5	3	2	1
27	28,29,30	0	0	0	0	0
28	31	2	1	2	2	1
29	31	4	2	2	3	2
30	31,32	12	6	1	1	3
31	33	6	3	3	2	1
32	33	4	2	0	1	1
33	null	0	0	0	0	0

应用文中提出的算法对此问题进行求解。参数选取如下:种群规模 PopSize = 1 000;迭代次数 MaxCount

= 1 000;取 $P_{c1} = 0.9$ 、 $P_{c2} = 0.6$ 、 $P_{m1} = 0.1$ 以及 $P_{m2} = 0.001$ 。

3.1 模拟结果及分析

3.1.1 确定关键链

以最小化完工时间 $\min T$ 为目标进行求解。

根据自适应遗传算法优化求得的最优染色体为: 0-1-2-3-18-19-20-13-4-5-6-7-14-16-27-30-8-21-22-24-9-11-23-29-15-28-10-25-12-26-17-31-32-33,关键链为 0-1-3-20-8-9-11-26-33,关键链时间为 28。

非关键链 1:2-4;汇入关键链上的工序 33。

非关键链 2:6-7-5;汇入关键链上的工序 33。

非关键链 3:6;汇入关键链上的工序 8。

非关键链 4:10-12;汇入关键链上的作业 33。

非关键链 5:13-14-15(-16)-17;汇入关键链上的作业 33。

非关键链 6:18;汇入关键链上的作业 20。

非关键链 7:18-19-21-24;汇入关键链上的作业 33。

非关键链 8:22(-23)-25;汇入关键链上的作业 26。

非关键链 9:27-28(-29)(-30)-31-32;汇入关键链上的作业 33。

非关键链 10:27-30-32;汇入关键链上的作业 33。

3.1.2 设置缓冲量

目前对缓冲区大小的设置方法主要有两种,即剪贴法(cut and paste)和根方差法(rootsquare error)。剪贴法的优点是简单易行,缺点是缓冲区长度随着工作链的长度呈线性增长;根方差法基于各工作执行时间相互独立的假设,为关键链工作以及整个项目提供恰当的缓冲区保护。文中采用根方差法对缓冲区大小进行设置,以项目缓冲 PB 为例,其计算方法为:

$$\begin{aligned} PB &= (t^2(0) + t^2(1) + t^2(3) + t^2(20) + t^2(8) + \\ &\quad t^2(9) + t^2(11) + t^2(26) + t^2(33))^{1/2} = \\ &\quad (0^2 + 0^2 + 5^2 + 6^2 + 5^2 + 5^2 + 2^2 + 5^2 + \\ &\quad 0^2)^{1/2} = 11.83 \end{aligned}$$

项目缓冲 PB 大小为 11.83,项目总时间为 39.83,即 40 天。以此方法,各汇入缓冲区大小如下:

FB¹ 大小为 6;FB² 大小为 5.39;FB³ 大小为 0;FB⁴ 大小为 3;FB⁵ 大小为 7;FB⁶ 大小为 0;FB⁷ 大小为 5.39;FB⁸ 大小为 5;FB⁹ 大小为 6.71;FB¹⁰ 大小为 6.32。

3.2 结果分析

3.2.1 资源利用情况分析

资源 R_1 、 R_2 、 R_3 的利用情况如图 2 所示。
对项目进度计划问题,项目采取的是工序间交叉

进行的方式,体现了项目执行过程中综合利用有限资源、实现项目整体目标最优的思想。分析图2,对三种资源在整个进度计划过程中的利用情况进行分析。可知,三种资源利用率的最大值均达到了100%,平均利用水平均在50%以上,分别为90.1%、69.8%和70.6%。说明算法充分发挥了有限资源的作用,达到了较高的利用水平。

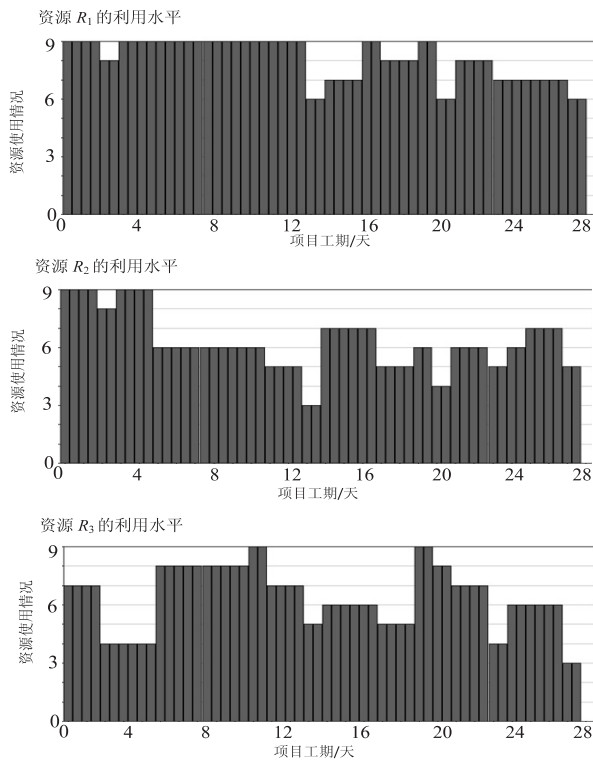


图2 三种资源利用水平

3.2.2 算法稳定性、有效性分析

由于遗传算法自身存在的随机性,每次迭代得到的最优解也存在着一一定的波动性。运行程序240次,对得到的优化结果进行统计分析。文中提出的算法达到了最优解28,且得到的解出现在(28~32)区间的概率为70.42%,解出现在(33~37)区间的概率为25.83%,因此算法能够得到较好解所占的比例为96.25%。证明了文中提出的算法的稳定性和有效性。

3.3 对比分析

3.3.1 方法对比

文中采用Goldratt博士提出的基于约束理论的关键链方法解决项目进度计划问题,这是一种综合考虑资源问题以及人的行为的方法,是对项目进度进行计划新的诠释。针对该案例,关键链方法得出的多项目总时间为40。用传统的网络图方法计算最长的一条路径为0-18-20-23-25-26-33,总时间长为36。网络图方法得出的时间虽然较小,但其在实际操作中并不可行。首先关键路径方法没有考虑资源问题,在该案例中由于资源的有限,工序对资源需求存在冲突,一

些工序并不能够同时进行。工序不能按时开始,就造成了项目工期的拖延。其次,关键路径法未能对人的行为因素以及环境因素进行考虑,因而在项目实际执行中,提前完工的工序并不能促使其紧后工序的提前开始,但一旦有工序未能按时完工,就会造成其随后工序的推迟,造成了进度的拖延。综上,关键链方法综合考虑了多方因素,其进度计划在实际操作中是有效可行的。

3.3.2 算法对比

针对该项目问题,运用文中提出的自适应算法与传统的遗传算法相比较,参数选择相同,分别运行200次。虽然两种算法均能达到最优解,但所需运行时间和运行次数相差较大,结果分别如图3、图4所示。

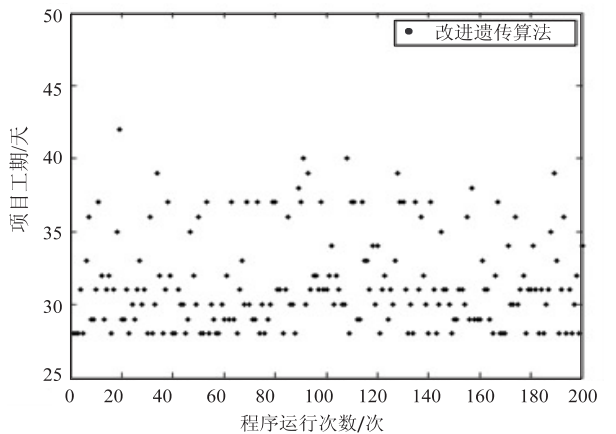


图3 自适应遗传算法结果

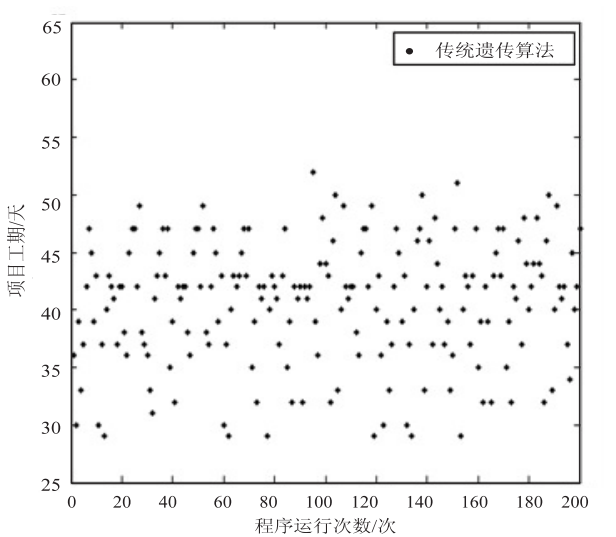


图4 传统遗传算法结果

分析图3和图4可知,将文中提出的算法与传统的遗传算法相比较,可以看到求得的项目总工时的平均值较小,与最优解的偏差较小,计算效率可靠,误差较小。且用文中算法求解,最优解出现的频率明显大于用传统的遗传算法。综上,说明了文中提出的算法求得的解更为优秀,且更具稳定性。

长包的数据内容。当缓存深度是100的情况下,每个超长包的组包消耗时间约为14.8 μs,满足监控超长包时间精度要求。

5 结束语

通过对1394总线网络监控器数据处理关键技术的研究,文中提出了一种高性能的数据监控处理方法,并对该方法的硬件设计实现做了较详细的介绍。经过1394总线网络综合测试结果表明,可正确地对总线上传输的数据信息进行错误检错、故障分析和信息显示,且监控工作稳定可靠,满足时间精度要求,易实现模块化和通用化,为以后的相关测试系统的研制开发提供了设计思路。

参考文献:

[1] 郑斐,田泽,马宁,等. 1394与RS422总线桥的设计与实现[J]. 计算机技术与发展,2013,23(7):253-256.

[2] 蒋国峰,白红. 1553B总线监控器的设计与实现[J]. 电子设计工程,2011,19(17):98-100.

[3] 王治,田泽. 一种高性能AFDX监控卡的实现技术研究[J]. 计算机技术与发展,2010,20(8):217-220.

[4] Song Jiayou. Design of monitor system for general 1553B bus [J]. Electronic Design Engineering,2011(3):90-92.

(上接第5页)

4 结束语

文中通过分析项目进度计划问题中资源间的约束关系以及项目内部工序的约束关系,应用关键链方法的理论,建立多资源项目进度计划问题的约束规划模型,提出了基于自适应遗传算法对模型的求解。结果表明,文中提出的基于关键链方法建立的模型和采用的自适应遗传算法进行的求解,能够充分地考虑到项目进度计划问题以及多资源约束问题的特点,能够以较快的速度找到有效的项目进度计划方案。

参考文献:

[1] 蔡晨,万伟. 基于PERT/CPM的关键链管理[J]. 中国管理科学,2003,11(6):35-39.

[2] Goldratt E M, Cox J. The goal [M]. 2nd ed. Aldershot: Gower, 1993.

[3] Goldratt E M. Critical chain [M]. MA: The North River Press, 1997.

[4] Leach L P. Critical chain project management [M]. Boston, London: Artech House, 2000.

[5] Yukel O I, Rom W O, Eksioglu S D. An investigation of buffer sizing techniques in critical chain scheduling [J]. European Journal of Operational Research, 2006, 172(2): 401-416.

[5] Brajou F, Ricco P. AFX-based flight test computer concept [J]. IEEE Instrumentation & Measurement Magazine, 2005 (3):55-58.

[6] ARINC664 Aircraft Data Network Part7 (draft3): Deterministic network [S]. [s.l.]: ARINC Airlines Electronic Engineering Committee, 2005.

[7] 马进, 王伟, 杨煜普. 基于DMA的1394总线高速数据传输实现[J]. 计算机工程与设计, 2010, 31(20): 4410-4413.

[8] 赵莹, 闫娟娟, 郑铮. 基于IEEE1394的光纤通道技术[J]. 航空电子技术, 2008, 39(4): 15-22.

[9] 张大朴, 王晓, 张大为, 等. IEEE1394协议及接口设计[M]. 西安: 西安电子科技大学出版社, 2004.

[10] 李肇庆, 朱险峰. IEEE1394接口技术[M]. 北京: 国防工业出版社, 2004.

[11] 龚东磊, 胡继波. IEEE1394高速串行总线及其应用[J]. 计算机工程, 2002, 28(11): 237-239.

[12] 牛少平, 田泽. 基于SoPC的智能监控卡设计与实现[J]. 计算机技术与发展, 2012, 22(8): 167-169.

[13] 廖寅龙, 田泽. FC网络通信中PCIe的接口的设计与实现[J]. 航空计算技术, 2010, 40(4): 127-130.

[14] 李建兵, 徐向辉. 基于PCI-E总线的高速数据传输卡的设计与实现[J]. 计算机测量与控制, 2011, 19(10): 2581-2583.

[6] 徐哲, 王黎黎. 基于关键链技术的项目进度管理研究综述[J]. 北京航空航天大学学报(社会科学版), 2011, 24(2): 54-59.

[7] 马国丰, 尤建新. 关键链项目群进度管理的定量分析[J]. 系统工程理论与实践, 2007, 27(9): 54-60.

[8] 刘士新, 宋健海, 唐加福. 关键链——一种项目计划与调度新方法[J]. 控制与决策, 2003, 18(5): 513-516.

[9] Rand G K. Critical chain: the theory of constraints applied to project management [J]. International Journal of Project Management, 2000, 18(5): 173-177.

[10] 杨莉, 李南. 软件项目进度的关键链管理[J]. 计算机工程, 2010, 36(7): 42-44.

[11] 刘士新. 资源受限工程调度问题的优化方法研究[D]. 沈阳: 东北大学, 2000.

[12] 熊伟清, 刘明达, 魏平. 遗传算法的基因定位算子[J]. 控制理论与应用, 2005, 22(3): 491-494.

[13] Srinivas M, Patnaik L. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. IEEE Transactions on System, Man and Cybernetics, 1994, 24(4): 656-667.

[14] 乐秀璠, 覃振成, 尹峰. 基于自适应模拟退火遗传算法的多目标最优潮流[J]. 继电器, 2005, 33(7): 10-15.

[15] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2002.

基于改进遗传算法的关键链项目进度计划优化

作者：[张琦](#)，[廖良才](#)，[王卫威](#)，[ZHANG Qi](#)，[LIAO Liang-cai](#)，[WANG Wei-wei](#)
作者单位：[国防科学技术大学 信息系统与管理学院, 湖南 长沙, 410073](#)
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(4)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201404001.aspx