

遗传算法在硬实时系统预调度中的应用¹

张凤芝 任长明 郑红霞 张红线

(天津大学电子信息工程学院 天津 300072)

摘要 针对硬实时系统的特性, 改进了遗传算法的初始群体生成方法及交换变异算子, 实现了硬实时系统任务预调度. 对给出的任务集合找出了最佳调度序列. 通过测试集合仿真, 效果良好.

关键词 硬实时系统, 预调度, 遗传算法

中图分类号 TN711.6

1 引言

广泛应用于航空、航天、军事、国防及工业控制领域的一类计算机系统为硬实时系统, 它必须在明确指定的时间内对外界信号作出响应, 否则会产生严重后果, 为此必须保证系统中的所有任务满足指定的截止时间. 在硬件条件已经确定的情况下, 进程调度就是保证实时性的关键. 由于联机调度不能对整个调度序列动态寻优, 难以保证满足每个进程严格的时间限制, 因而近年来开展了预调度的研究^[1-3], 预调度是系统正式运行之前离线进行的调度, 它要求事先知道系统中各个进程的时间属性和同步关系, 据此找出一个最佳的调度序列. 联机调度根据预调度产生的调度表以时间驱动方式调度各个进程运行, 就可确保系统的实时性.

遗传算法是根据达尔文的随机挑选、优胜劣汰、自然遗传机制的搜索算法. 它不对搜索空间进行任何约束和限制, 例如连续、可导等. 由于它是一种群体搜索操作, 因而抗噪声能力强, 已广泛用于商业、科学和工程等领域. 本文讨论如何使用遗传算法求出最优的调度序列.

2 调度模型

预调度模型主要包括进程描述、处理器集合、最优调度标准几个元素.

我们定义处理器集合为 $V = \{u_1, u_2, u_3, \dots, u_m\}$, $|V| = m$, 将其映射到逻辑时间轴上可形式化为: $R = \{(t, q) | t \in [0, \infty) \wedge (1 \leq q \leq m)\}$.

我们定义进程集合为 $P = \{p_1, p_2, p_3, \dots, p_n\}$, 可以用一个四元组 $(r_p, c_p, d_p, \text{prd}_p)$ 来描述周期性任务 p . 其中 $\text{prd}_p, c_p, d_p, r_p$ 分别表示进程 p 的周期、最大运行时间、截止时间和就绪时间. 周期性进程 p 的第 i 次 ($i = 0, 1, 2, \dots$) 执行有 $r_{pi} = r_p + \text{prd}_p \cdot i$, $d_{pi} = d_p + \text{prd}_p \cdot i$.

异步进程 a 用一个三元组 (c_a, d_a, min_a) 来描述. 其中 c_a, d_a, min_a 分别是 a 最大执行时间、最迟结束时间、两次请求的最小时间间隔. a 的第 i 次执行 a_i , 如果 a_i 的请求时间是 r_{ai} , 则 a_i 的运行截止时间为 $d_{ai} = r_{ai} + d_a$; 如果用 min_a 代表 a 的周期, 则可以统一处理成周期进程.

进程 i 的延迟表示为 $\text{LATENESS}_i = e_i - d_i$, 其中 e_i, d_i 分别为进程 i 的实际结束时间和截止时间. 对于任务集合 $S(P)$, 可行调度必须且只需满足关系:

$$\text{MAX}\{\text{LATENESS}(P[i]) | i \in (0, |S|)\} \leq 0$$

就是说任务集中的任意一个进程的时间延迟均小于等于 0, 即满足它们的截止时间.

¹ 2001-02-28 收到, 2002-01-04 定稿

将各周期任务周期的最小公倍作为一个调度长度,则在此调度长度上,每个周期任务均执行整数次;每重新执行一次,则看成是一个新任务;这样,我们只需在一个调度长度上求最优解即可。

3 基于遗传算法的实时系统优化调度

3.1 染色体编码

一个染色体就是一个串,我们用它代表一个调度序列。为此,我们没有采用基于 $\{0,1\}$ 符号集的二值编码形式;而是用进程号作为基因进行染色体编码。当进程集大小为 n 时,染色体的最大个数为 $n!$ 。

3.2 适应度函数

适应度值是评判一个解(染色体)优劣程度的标准,我们要用适应度函数对染色体进行比较排序并在此基础上计算选择概率,因此应保证适应度函数的值为正值。我们采用下式计算染色体的适应度 f : $f = \text{SL-LATENESS}$ 。式中 SL 表示调度长度,它等于各周期任务执行周期的最小公倍数,是一个定值。LATENESS 为本次调度的时间延迟,它等于各任务延迟时间的最大值。显然, LATENESS 越小,则 f 越大,适应度就越好。

3.3 初始群体的生成

群体是多个染色体(即调度序列)的集合。为了保证物种的多样性,增加群体操作能力,我们采用了最早截止时间优先、最小宽裕度优先、(宽裕度定义为 $\text{LAXITY} = d_i - (r_i + c_i)$) 和随机选取 3 种方法结合的方式生成初始群体,这样会使初始群体的适应度比完全采用随机方式高。当然,一个有效的初始群体,一定要保证进程间同步关系,并保证每个任务的开始时间大于它的就绪时间。

3.4 控制参数的确定

遗传算法中需要确定的参数主要有群体大小 n_p , 交换概率 p_c , 变异概率 p_m 和遗传代数 n_g 。这些参数的选择影响着遗传算法的性能,目前常用的参数范围是 $p_c = 0.5 \sim 1.0$, $p_m = 0 \sim 0.05$, 我们结合本课题情况经过多次测试选取 $p_c = 0.75$, $p_m = 0.01$; 参数 n_p 和 n_g 的选取与进程集大小线性相关。

3.5 遗传算子设计

3.5.1 交换算子 我们试验了 3 种交换操作: 按序交换、部分匹配交换和循环交换。

(1) 按序交换 给定两个父染色体,子染色体的左半部分是第 1 个(或第 2 个)父染色体的交换点以左的部分,子染色体右半部分的基因也来自第 1 个(或第 2 个)父染色体,但基因顺序按第 2 个(或第 1 个)父染色体中这些基因的出现顺序来定。方法如图 1 所示。

基因位置从 0 开始,交换点在位置 5,生成的子染色体 1 如图 1 所示。子染色体 2 用同样的方法得到。显然,两个合法的父染色体按序交换所产生的子染色体也是合法的。

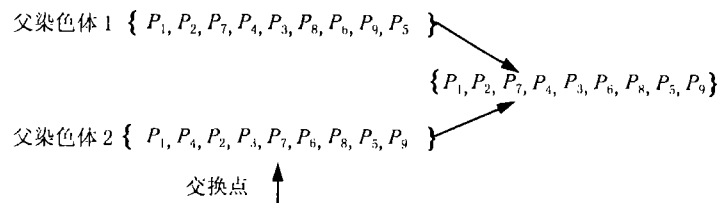


图 1 按序交换

(2) 部分匹配交换 在第 1 个 (或第 2 个) 父染色体中寻找交换点后两个父染色体相应位置的基因对, 然后将第 1 个 (或第 2 个) 父染色体中的这两个基因对互换产生子染色体. 方法如图 2 所示.

这种方法虽然也是基于将交换点后的基因按另一个父染色体的基因顺序进行规范, 但基因对的互换可能会违反进程的同步约束. 如上例, 若 P_4 和 P_3 之间有优先关系, 则生成的新染色体是不合法的.

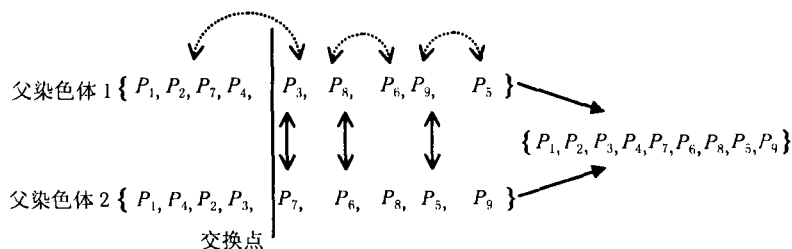


图 2 部分匹配交换

(3) 循环交换 从第 1 个父染色体的位置 1 开始, 将此位置的基因复制给子染色体; 检查第 2 个父染色体位置 1 的基因, 在第 1 个父染色体中寻找这个基因, 假定在位置 i 找到, 则将此基因复制到子染色体的位置 i ; 同样, 第 2 个父染色体位置 i 的基因不能被复制, 需在第一个父染色体中定位此基因并将其复制给子染色体, 一直重复这个过程, 直到在第 2 个父染色体中遇到的基因已经从第一个父染色体中复制给了子染色体, 这样就结束了一个循环. 另一个循环从子染色体未被占用的最小位置开始, 而且复制是从第 2 个父染色体进行的. 图 3 是循环交换的一个例子.

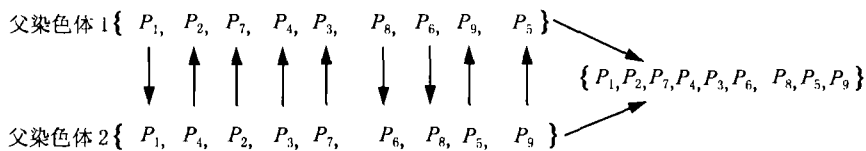


图 3 循环交换

这种方法仍有可能产生不合法子染色体, 如对任务集 $\{p_1, p_2, p_3, p_4, p_5\}$, 任务间需满足 $p_1 \propto p_2 \propto p_4, p_1 \propto p_3 \propto p_5$ (\propto 表示优先), 图 4 中示出了这种情况.

根据以上的分析, 我们采用了可靠程度最高的按序交换算子.

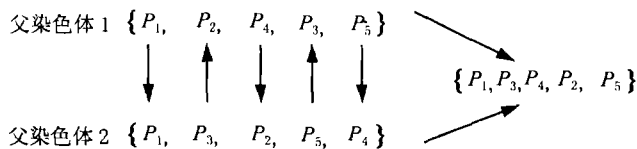


图 4 不符合条件的子染色体

3.5.2 变异算子 鉴于以上原因, 我们不采用常规的简单互换两个位置上的基因方法以免导致不合法基因的产生, 而是采用计算候选集的方法找出适合后移的基因, 将它放到变异基因之后, 原先在候选基因和变异基因之间的基因依次前移。例如在染色体 $\{p_1, p_2, p_4, p_3, p_5\}$ 中, p_1 是候选基因, p_5 是变异基因, 变异算子会产生新的染色体 $\{p_2, p_4, p_3, p_5, p_1\}$ 。

3.6 算法描述

本遗传算法的算法描述如下: (1) 生成有效初始群体 ParentPopu(g), 它满足指定的同步约束条件。(2) 计算群体中每个染色体的适应度。(3) 满足结束条件则转 (7), 否则继续。(4) 按适应度值进行选择, 产生下一代。(5) 进行交换和变异操作。(6) 计算本代中各染色体的适应度值, 保存适应度最好的染色体。转 (3)。(7) 输出最优的染色体, 即调度序列, 结束。

结束条件可以有以下几种情况: 已找到可行解; 已满足遗传代数; 操作员按下强迫结束按钮。由于解是在不断优化的, 任何时候结束都会找到优于初始解的解。

4 仿真结果

7 个任务, 均在同一处理机上运行, 它们之间的同步限制为 $F \propto G, B \propto E, A \otimes C, A \otimes E, B \otimes F$ (\propto 表示优先, \otimes 表示互斥), 任务 D 和其它进程间无约束关系。各任务时间属性如表 1 所示, 时间单位为秒。

表 1 各任务时间属性

任务号	任务名称	就绪时间 (s)	运行时间 (s)	截止时间 (s)	运行周期 (s)	处理机号
1	A	80	10	200	200	0
2	B	30	20	100	100	0
3	C	10	20	50	100	0
4	D	50	10	90	200	0
5	E	20	30	100	200	0
6	F	70	10	100	200	0
7	G	55	20	150	200	0

由表 1 可知: 调度长度 $SL=200$, 任务 B 和任务 C 在此调度长度上需要调度两次, 第一次调度的任务号分别为 2 和 3, 第二次调度的任务号分别为 8 和 9。经过 42 代遗传操作后, 终于找出最佳调度序列, 本算法在 PII 350 机上花费 70(s) 时间^[4]。结果如表 2 所示。

表 2 优化调度结果

最大适应度	最佳调度序列	产生最佳序列的遗传代数	花费 CPU 时间 (s)
200	3 2 4 5 6 7 1 9 8	42	70

由适应度函数 $f = SL - LATENESS$ 可知, 本调度的延迟时间: $LATENESS = 200 - 200 = 0$, 调度序列的时间关系见图 5。

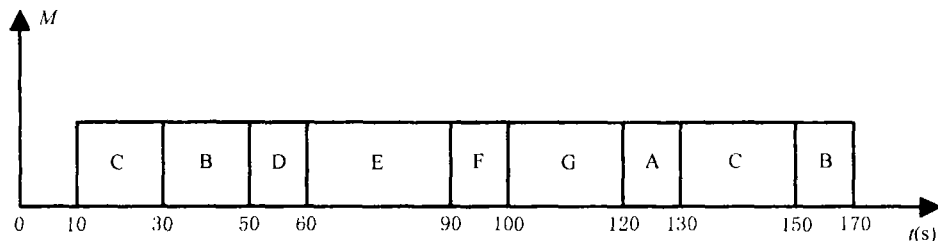


图 5 调度序列时间关系

5 结束语

本文将遗传算法应用于硬实时系统任务预调度, 经用多个测试实例进行仿真, 均取得了良好的仿真效果, 这将有助于对硬实时系统任务预调度领域的进一步研究。

参 考 文 献

- [1] J. XU, D. L. Panas, Pre-run-time scheduling of process with exclusion relations on nested or overlapping critical sections, In proc. 11th. Annu. IEEE Int. Phoenix Conf. on Computers and Communications(Ipccc-92), Scottsdale, AZ, 1992, April, 1-3, 774-782.
- [2] S. K. Tripathi, V. Nirkhe, Pre-scheduling for synchronization in hard real-time systems, Proc. International Workshop on Operating Systems of the 90s and Beyond, Berlin, 1991, 102-108.
- [3] T. Shepard, M. Gagné, A model of the F18 mission computer software for pre-run-time scheduling, Proc. 10th. International Conference on Distributed Computing Systems, Los Alamitos, 1990, 62-69.
- [4] 郑红霞, 硬实时系统任务预调度研究, [硕士论文], 天津, 天津大学电信学院, 2000.

GA APPLICATION IN HARD REAL-TIME SYSTEM PRE-RUN-TIME SCHEDULE

Zhang Fengzhi Ren Changming Zheng Hongxia Zhang Hongxian

(*School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China*)

Abstract According to the features of hard real-time systems, GA is improved in making initial group, crossover and mutation operator, etc.. A pre-run-time scheduler of hard real-time systems is implemented, and the process sequence is optimized for the given task set in order to meet the deadlines. Finally, the validity of the GA is verified through simulation with testing case.

Key words Hard real-time system, Pre-run-time schedule, Genetic algorithm

张凤芝: 女, 1946年生, 副教授, 研究领域为调度算法, 实时系统设计.

任长明: 男, 1942年生, 教授, 研究领域为智能控制, 系统建模.

郑红霞: 女, 1974年生, 助工, 研究领域为应用软件设计.

张红线: 女, 1976年生, 硕士, 研究领域为实时软件设计.