

作业车间调度问题的多种群遗传算法

蔡良伟, 张基宏, 李霞

(深圳大学电子工程系, 广东深圳 518060)

摘要: 针对最小化完工时间的作业车间调度问题提出一个多种群遗传算法, 该算法基于工艺约束定义个体的编码方式, 基于工件操作构造遗传算子, 保证了所有个体的可行性; 多种群算法通过各个种群之间的相互竞争和良种共享, 提高了资源的利用效率, 有效地克服个体早熟, 改善了算法的收敛性能。典型测试算例表明该算法是非常有效的。

关键词: 作业车间调度; 多种群; 竞争; 共享; 遗传算法

中图分类号: TP273 **文献标识码:** A **文章编号:** 0372-2112(2005)06-0991-04

A Multi-Population Genetic Algorithm for Job Shop Scheduling Problem

CAI Liangwei, ZHANG Jihong, LI Xia

(Department of Electronic Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China)

Abstract: A multi-population genetic algorithm is presented for Job Shop scheduling problem targeting to minimize the makespan. The coding scheme of chromosomes is defined based on the process constraints, genetic operators are defined according to job operations, so all chromosomes are feasible. By the competition and sharing mechanism among populations, the computing resource is utilized more efficiently, and prematurity is overcome effectively. These improve the convergence of the multi-population genetic algorithm. Experiments on Job Shop Benchmarks show that this algorithm has good performance.

Key words: job shop scheduling; multi-population; competition; sharing; genetic algorithm

1 引言

在最小化完工时间的作业车间调度问题 (Job Shop Scheduling Problem, 简称 JSP)^[1] 中, 给定了一组工件和一组机器, 在任一时刻一台机器最多可加工一个工件, 每个工件都包含一系列的操作, 这些操作必须按照规定的顺序进行, 每个操作需要在一台指定的机器上不间断地加工一预先给定的时间, 目标是要找出一个加工调度方案, 使完成所有工件加工所需时间最少。

实践证明, JSP 问题是一个非常复杂的组合优化问题, 用传统的优化方法往往不能进行有效的求解。遗传算法^[2~6] 是模拟自然界生物进化机理的一种随机搜索并行优化算法, 在包括组合优化等复杂优化问题的求解方面具有独特的优越性。简单的单种群遗传算法存在计算效率低、容易出现早熟而收敛于局部最优解等不足, 因此近年来多种群并行协同遗传算法获得了越来越广泛的关注和应用^[7~11]。用遗传算法求解 JSP 问题^[12~16] 时, 关键是要解决个体的可行性, 保证工艺约束得到严格的满足。本文针对最小化完工时间的 JSP 问题提出一个多种群遗传算法, 基于工艺约束定义个体的编码方式, 基于工件操作构造遗传算子, 保证了遗传进程中所有个

体的可行性; 算法中定义了基于种群最佳适应值和成长性的竞争力指数, 种群根据其竞争力指数获取计算资源, 优化了资源分配; 通过构造优良种群, 使优良个体为各个种群共享, 增加种群中个体的多样性, 改善各种群的品质, 有效避免早熟现象, 改善了算法的收敛性能。典型测试算例表明该算法可以有效地求解 JSP 问题。

2 问题描述

设 J 是所有工件的集合, M 是所有机器的集合, O 是所有操作的集合, $\sigma_1 > \sigma_2 > \dots > \sigma_m$ 表示工件 j 的加工顺序约束, m 是集合 M 中元素的个数, $\sigma_k > \sigma_{k+1}$ 表示操作 σ_{k+1} 只能在操作 σ_k 完成后才能开始, $M(\sigma_k)$ 表示操作 σ_k 所用的机器, $p(\sigma_k)$ 表示操作 σ_k 所需的时间, $t(\sigma_k)$ 表示操作 σ_k 的起始时间, 最小化完工时间的 JSP 问题可描述如下:

$$\min C_{\max} = \min_{\sigma_k \in O} \{ \max(t(\sigma_k) + p(\sigma_k)) \}$$

subject to:

$$\begin{aligned} t(\sigma_k) &\geq t(\sigma_j) + p(\sigma_j), \quad \forall \sigma_j > \sigma_k, j \in J \\ t(\sigma_k) &\geq t(\sigma_j) + p(\sigma_j) \vee t(\sigma_j) \geq t(\sigma_k) + p(\sigma_k), \\ &\quad \forall M(\sigma_j) = M(\sigma_k) \end{aligned}$$

$$t(\sigma_k) \geq 0, \forall \sigma_k \in O$$

3 多种群遗传算法

3.1 编码结构

编码是将问题的解表示成某种形式的个体,合适的编码是设计简单高效遗传操作的关键.对 JSP 问题进行编码的最大困难是必须考虑每个工件的加工顺序.本文采用基于工艺约束的个体编码方式,每个个体都是由基因元素组成的有序集合,一个基因元素就是一个工件在某一机器上的操作:

$$G = \{g_1(j_1, m_1) \dots g_i(j_i, m_i) \dots g_N(j_N, m_N)\}$$

其中, $g_i(j_i, m_i) \in O$, $j_i \in J$, $m_i \in M$.为了保证个体对应的加工方案的可行性,在每个个体有序集合中,对应任一工件 $j \in J$ 的所有基因元素的顺序必须满足该工件的加工顺序约束 $\sigma_1 > \sigma_2 > \dots > \sigma_m$,所有工件的加工顺序约束构成矩阵:

$$\begin{bmatrix} \sigma_1^1 & \dots & \sigma_k^1 & \dots & \sigma_m^1 \\ \sigma_1^2 & \dots & \sigma_k^2 & \dots & \sigma_m^2 \\ \vdots & & \vdots & & \vdots \\ \sigma_1^n & \dots & \sigma_k^n & \dots & \sigma_m^n \end{bmatrix}$$

利用上述矩阵产生初始种群个体的过程如下:

(1) 随机地选取一非空行;

(2) 令 $g_i(j_i, m_i)$ 等于该行的第一个元素,该行其他元素依次右移一位,行末补 $\emptyset$, $i = i + 1$;

(3) 重复(1)和(2),直至所有行变为全 $\emptyset$ 行.

3.2 交叉算子

定义 1 集合 A 和 $\bar{A}$ 称为工件集合 J 的一个划分,如果 $A \cup \bar{A} = J$, $A \cap \bar{A} = \emptyset$, 其中 $\emptyset$ 表示空集.

定义 2 将个体 G 中和工件子集 A 无关的所有元素用空元素 $\emptyset$ 代替,得到一个新的集合,记为 G_A ,

$$G_A = \{g_{a1}(j_1, m_1) \dots g_{ai}(j_i, m_i) \dots g_{aN}(j_N, m_N)\}$$

$$g_{ai}(j_i, m_i) = \begin{cases} \emptyset, & j_i \notin A \\ g_i(j_i, m_i), & j_i \in A \end{cases} \quad i = 1, 2, \dots, N$$

同样地,可以得到 $G_{\bar{A}}$,

$$G_{\bar{A}} = \{g_{\bar{a}1}(j_1, m_1) \dots g_{\bar{a}i}(j_i, m_i) \dots g_{\bar{a}N}(j_N, m_N)\}$$

$$g_{\bar{a}i}(j_i, m_i) = \begin{cases} \emptyset, & j_i \notin \bar{A} \\ g_i(j_i, m_i), & j_i \in \bar{A} \end{cases} \quad i = 1, 2, \dots, N$$

定义 3 对有序集合 R , 若元素 $a(a \in R)$ 为空元素, 其在集合 R 的所有空元素中的顺序号称为它的空序数, 记为 $\tau_R(b)$.

定义 4 对有序集合 R , 若元素 $b(b \in R)$ 为非空元素, 其在集合 R 的所有非空元素中的顺序号称为它的非空序数, 记为 $\lambda_R(b)$.

由一对父代个体 G 和 H 交叉产生一对后代个体 G' 和 H' 的操作如下:

随机产生工件集合 J 的一个划分 A 和 $\bar{A}$;

生成集合 G_A , $G_{\bar{A}}$, H_A 和 $H_{\bar{A}}$;

用 G_A 和 $H_{\bar{A}}$ 生成个体 G' , 用 $G_{\bar{A}}$ 和 H_A 生成个体 H' ,

$$G' = \{g'_{11}(j_1, m_1) \dots g'_{i1}(j_i, m_i) \dots g'_{N1}(j_N, m_N)\}$$

$$g'_{i1}(j_i, m_i) = \begin{cases} g_{ai}(j_i, m_i), & \text{当 } g_{ai}(j_i, m_i) \neq \emptyset \\ h_{\bar{a}i}(j_i, m_i) \mid \tau_{G_A}(g_{ai}) = \lambda_{H_{\bar{A}}}(h_{\bar{a}i}), & \text{当 } g_{ai}(j_i, m_i) = \emptyset \end{cases}$$

$$i = 1, 2, \dots, N$$

$$H' = \{h'_{11}(j_1, m_1) \dots h'_{i1}(j_i, m_i) \dots h'_{N1}(j_N, m_N)\}$$

$$h'_{i1}(j_i, m_i) = \begin{cases} h_{ai}(j_i, m_i), & \text{当 } h_{ai}(j_i, m_i) \neq \emptyset \\ g_{\bar{a}i}(j_i, m_i) \mid \tau_{H_A}(h_{ai}) = \lambda_{G_{\bar{A}}}(g_{\bar{a}i}), & \text{当 } h_{ai}(j_i, m_i) = \emptyset \end{cases}$$

$$i = 1, 2, \dots, N$$

3.3 变异算子

定义 5 集合 B 称为工件集合 J 的一个非空真子集, 如果 $B \neq \emptyset$, $B \neq J$, $B \subseteq J$.

定义 6 设集合 B 是工件集合 J 的一个非空真子集, 个体 G 中所有与 B 有关的元素按原来顺序构成的集合称为 G 关于 B 的子集, 记为 G^B .

定义 7 设集合 B 是工件集合 J 的一个非空真子集, μ 是个体 G 的元素, 若 $\mu \in G^B$, 记 μ 在 G^B 中的顺序号为 $\rho_{G^B}(\mu)$.

由个体 G 变异产生新个体 G'' 的操作如下:

随机产生工件集合 J 的两个元素数目相等的不相交非空真子集 B 和 C ;

$$G'' = \{g''_{11}(j_1, m_1) \dots g''_{i1}(j_i, m_i) \dots g''_{N1}(j_N, m_N)\}$$

$$g''_{i1}(j_i, m_i) = \begin{cases} g_i(j_i, m_i), & \text{当 } g_i \notin G^B \text{ 且 } g_i \notin G^C \\ g_i(j_i, m_i) \mid g_i \in G^C, \rho_{G^C}(g_i) = \rho_{G^B}(g_i), & \text{当 } g_i \in G^B \\ g_i(j_i, m_i) \mid g_i \in G^B, \rho_{G^B}(g_i) = \rho_{G^C}(g_i), & \text{当 } g_i \in G^C \end{cases}$$

$$i = 1, 2, \dots, N$$

3.4 多种群系统

在自然界中,一方面生物按地域进行分布,不同地域有不同的特点和不同的进化程度,从大自然中争取资源为己所用;另一方面,各不同地域之间又通过信息交换、取长补短、互通有无、共同进步.

本文中给出的多种群遗传算法就是借鉴自然界生物中的这一现象设计出来的.整个算法由计算环境中的多个普通种群和一个优良种群构成,各个普通种群进行竞争,从计算环境中得到计算资源,

一旦某个种群获得计算资源,它便进行一次自己的进化进程.各个普通种群将进化得到的优良个体贡献出来,组成优良种群,普通种群可以从中获取优良个体,以改善本种群的品质,系统结构图如图 1 所示.

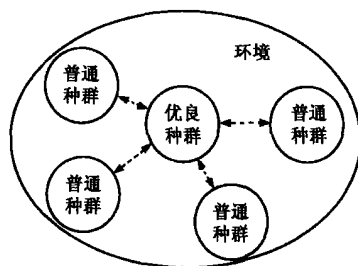


图 1 多种群系统

3.4.1 竞争机制

在算法中,种群的竞争力取决于种群的最佳适应值和成

长性这两个因素,竞争力强的种群将有较大的获得计算资源的概率。在寻找最大适应值的问题中,当一个种群的最佳适应值较小时,表明该种群仍处于进化进程的初始阶段,距离最优解较远,应将资源较多地分配给距离最优解较近的种群;另一方面,当一个种群再无成长性或成长性较差时,如果该种群还没有达到最优解,表明它可能出现早熟现象,趋于局部最优,这时将资源较多地分配给成长性好的种群;这样,一个最佳适应值大同时成长性好的种群将优先获得计算资源分配。

定义种群的成长性为:

$$D_i = \frac{F_{\max}^i(k) - F_{\max}^i(k-1)}{F_{\max}^i(k-1)}, \quad i = 1, 2, \dots, N_P$$

其中, $F_{\max}^i(k)$ 表示第 i 个种群当前进化进程的最佳适应值, $F_{\max}^i(k-1)$ 表示第 i 个种群上一次进化进程的最佳适应值, N_P 为普通种群的个数。当种群采用保留最佳个体的进化策略时,其成长性必为非负数。

定义种群的竞争力指数为:

$$R_i = \beta \frac{F_{\max}^i(k)}{\sum_{j=1}^{N_P} F_{\max}^j(k)} + (1 - \beta) \frac{D_i}{\sum_{j=1}^{N_P} D_j}, \quad j = 1, 2, \dots, N_P$$

其中, β 为权重系数, $\beta \in [0, 1]$, β 愈大, 种群最佳适应值的影响愈大; 反之, β 愈小, 则种群成长性的影响就愈大。

R_i 是反映种群最佳适应值和成长性的综合指数, 一个种群的 R_i 愈大, 它获得计算资源分配的概率就愈大。

资源分配算法如下:

第一步: 计算 $r_i = R_i / \sum_{j=1}^{N_P} R_j, i = 1, 2, \dots, N_P$ 。

第二步: 计算 $\pi_i = \sum_{j=1}^i r_j, i = 1, 2, \dots, N_P$ 。

第三步: 产生一个在 $[0, 1]$ 内均匀分布的随机数 φ , 令 $k = \{\min(i) | \pi_i > \varphi, i = 1, 2, \dots, N_P\}$ 。

第四步: 第 k 个种群获得本次资源分配, 执行一次进化进程。

3.4.2 良种共享机制

优良种群由各个普通种群中适应值较优的若干个体组成, 是一个优良种子库, 供各种群的进化进程共享。共享机制有两种: 良种迁移和良种交叉。

良种迁移: 普通种群在自己的进化进程中, 直接从优良种群中引进若干优良种子个体代替本种群中的较差个体。

良种交叉: 普通种群在自己的进化进程中, 从优良种群中选取若干优良种子个体和普通种群中的个体进行交叉繁殖, 用交叉生成的较优个体取代普通种群中原来的个体。

3.5 种群内部的遗传进化进程

普通种群内部的遗传进程包括选择、交叉、变异、良种迁移等步骤, 其中交叉有内部交叉和良种交叉两种。一次内部遗传进程的结构如下:

step1: 使用 Roulette Wheel 方法选择个体;

step2: 在种群内随机选择若干个体两两配对, 进行内部交叉操作;

step3: 在种群中随机选择若干个体, 同时在优良种群中随机选择相同数量的个体, 两组个体进行一一配对, 进行良种交叉;

step4: 在种群内随机选择若干个体, 进行变异操作;

step5: 良种迁移。

3.6 多种群遗传算法整体结构

多种群遗传算法的整体结构如下:

step1: 初始化参数;

step2: 产生各种群的初始种群;

step3: 计算各普通种群的竞争力指数, 确定本次计算资源分配方案;

step4: 获得计算资源分配的种群执行一次内部遗传进程;

step5: 更新优良种群;

step6: 判断是否满足收敛条件。如满足, 算法运行结束, 输出结果; 如不满足, 则返回 step3。

4 经典问题计算

采用文中定义的编码结构、交叉和变异算子、竞争和共享机制, 构造单种群遗传算法 (SPGA) 和多种群遗传算法 (MPGA), 分别对 Fisher and Thompson 的三个经典 Job Shop 问题^[1]进行求解, 各运行 10 次。算法中, 每个种群的规模为 50, 遗传进化代数 1000, 交叉概率 0.8, 变异概率 0.02, MPGA 的种群个数为 3。计算结果

如表 1 所示。图 2、图 3 和图 4 分别是三个问题各 10 次运行中最好一次的进化过程。

表 1 SPGA 和 MPGA 的结果比较

Problem		SPGA			MPGA				
n	m	C_{best}	P_{best}	Var	G_{min}	C_{best}	P_{best}	Var	G_{min}
6	6	55	70%	0.2%	73	55	80%	0.1%	52
10	10	930	30%	1.2%	437	930	50%	0.3%	353
20	5	1165	40%	1.7%	283	1165	60%	0.5%	217

注: n 为工件数, m 为机器数, C_{best} 为求得的最优解, P_{best} 为求得最优解的概率, Var 为平均偏差, G_{min} 为 10 次运行中得到最优解时的最小进化次数。

可看出: SPGA 和 MPGA 都能得到三个问题的最优解, 说明所定义的编码结构、交叉和变异算子是很有效的; 同 SPGA 相比, MPGA 能以更大的概率, 更小的偏差和更快的速度收敛

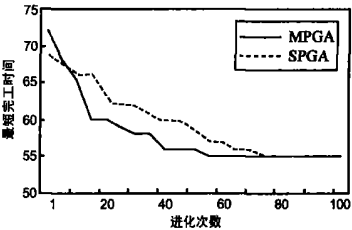


图 2 FT6×6调度问题

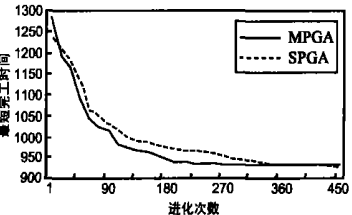


图 3 FT10×10调度问题

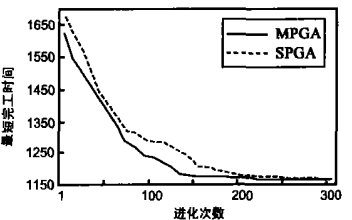


图 4 FT20×5调度问题

到最优解,证明多种群的竞争和共享机制具有明显的优越性.

5 结论

本文针对 JSP 调度问题定义了个体的编码方式以及交叉和变异算子,构造了具有竞争和共享机制的多种群遗传算法,通过 Fisher and Thompson 著名的三个经典 JSP 问题测算,取得满意的结果,表明了该算法能有效地求解复杂的作业车间调度问题.

参考文献:

- [1] Muth J F, Thompson G L. Industrial Scheduling[M]. New Jersey: Prentice Hall, 1963.
- [2] Goldberg D E. Genetic Algorithms in Search, Optimization, and Machine Learning[M]. USA: Addison Wesley Publishing Company, 1989.
- [3] Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs[M]. Berlin: Springer, 1996.
- [4] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithm[J]. IEEE Trans on Systems, Man and Cybernetics, 1994, 24(4): 656- 667.
- [5] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999.
Zhou Ming, Sun Shudong. The Theory and Application of Genetic Algorithm[M]. Beijing: National Defense Industry Press, 1999.
- [6] 刘勇, 康立山, 陈毓屏. 非数值并行算法——遗传算法[M]. 北京: 科学出版社, 1998.
Liu Yong, Kang Lishan, Chen Yuping. Nonnumeric Parallel Algorithm——Genetic Algorithm[M]. Beijing: Science Press, 1998.
- [7] Muhlenbein H, Schomish M, Born J. The parallel genetic algorithm as function optimizer[J]. Parallel Computing, 1991, 17(3): 619- 632.
- [8] J H Nang. A simple parallelizing scheme of genetic algorithm on distributed memory multiprocessors[J]. International Journal of High Speed Computing, 1994, 6(3): 451- 474.
- [9] 江瑞, 罗予频, 等. 一种基于多 Agent 协同的准并行遗传算法[J]. 电子学报, 2002, 30(10): 1490- 1495.
Jiang Rui, Luo Yupin, et al. A multi agent cooperating approach to quasi parallel genetic algorithms[J]. ACTA ELECTRONICA SINICA, 2002, 30(10): 1490- 1495.
- [10] 曾建潮, 徐玉斌. 多群体优良模式自学习遗传算法[J]. 系统工程与电子技术, 2000, 22(6): 81- 83.
Zeng Jianchao, Xu Yubin. A good schema self learning genetic algorithm with multi population[J]. Systems Engineering and Electronics, 2000, 22(6): 81- 83.
- [11] 曾宏坤, 沈德耀. 基于多智能体的新型遗传算法及其在复杂系统中的应用研究[J]. 信息与控制, 2003, 32(3): 277- 280.
Zeng Hongkun, Shen Deyao. Study of a new multiagent based genetic algorithm and its application in complex system[J]. Information and

Control, 2003, 32(3): 277- 280.

- [12] Federico Della Croce, Roberto Tadei, Giuseppe Volta. A genetic algorithm for the job shop problem[J]. Computers & Operations Research, 1995, 22(1): 15- 24.
- [13] Runwei Cheng, Mitsuo Gen, Yasuhiro Tsujimura. A tutorial survey of job shop scheduling problems using genetic algorithms[J]. Computers Industry Engineering, 1996, 130(4): 983- 997.
- [14] Shi G. A genetic algorithm applied to a classic job shop scheduling problem[J]. International Journal of Systems Science, 1997, 28(1): 25- 32.
- [15] Imed Kacem, Slim Hammadi, Pierre Borne. Approach by location and multiobjective evolutionary optimization for flexible job shop scheduling problems[J]. IEEE Trans on Systems, Man and Cybernetics, Part C, 2002, 32(1): 1- 12.
- [16] 尹文君, 刘民, 吴澄. 带工艺约束并行机调度问题的一种新的遗传算法[J]. 电子学报, 2001, 29(11): 1482- 1485.
Yin Wenjun, Liu Min, Wu Cheng. A new genetic algorithm for parallel machine scheduling with process constraint[J]. ACTA ELECTRONICA SINICA, 2001, 29(11): 1482- 1485.

作者简介:



蔡良伟 男, 1964 年 4 月生于广东阳江, 1988 年于清华大学自动化系获得工学硕士学位, 1998~ 1999 年在英国利物浦大学做访问研究, 现为深圳大学信息工程学院副教授, 主要研究方向为进化算法、智能控制与系统优化. E-mail: cailw@szu.edu.cn.



张基宏 男, 1964 年 6 月生于江苏南通, 1992 年于东南大学无线电系获得博士学位, 先后在英国和日本进行合作研究, 现为深圳大学信息工程学院教授, 主要研究方向为智能计算、信息论与编码理论, 发表论文五十余篇. E-mail: zhangjh@szu.edu.cn.



李霞 女, 1968 年 10 月生于四川乐山, 1997 年于香港中文大学获得博士学位, 现为深圳大学信息工程学院教授, 主要研究方向为智能优化方法及其在图像处理中的应用. E-mail: lixia@szu.edu.cn.