

基于自适应克隆启发算法的作业车间调度

徐立芳¹, 莫宏伟²

(1. 哈尔滨工程大学工程训练中心, 哈尔滨 150001; 2. 哈尔滨工程大学自动化学院, 哈尔滨 150001)

摘 要: 将优先权启发式算法获得的最小生产周期倒数作为抗体, 采用实数编码, 给出新的自适应克隆启发算法, 用于求解作业车间调度问题。设计一种新的自适应克隆算子, 基于抗体间距离的大小, 在抗体间自适应地分配抗体激励度和抗体克隆的数量, 模拟免疫系统自适应地调节抗体克隆数量的动态过程。FT10 和 FT06 的仿真实验表明, 该算法性能稳定、效果良好。

关键词: 人工免疫; 克隆选择算法; 免疫调度

Job Shop Scheduling Based on Self-adaptive Clone Heuristic Algorithm

XU Li-fang¹, MO Hong-wei²

(1. Engineering Training Center, Harbin Engineering University, Harbin 150001;

2. Automation College, Harbin Engineering University, Harbin 150001)

【Abstract】 This paper uses the inverse of *Makespan* as antibody, which is calculated by priority heuristic algorithm. It presents a Self-adaptive Clone Heuristic Algorithm(SACHA) based on the prior algorithm to solve Job Shop scheduling problem. SACHA adapts real code and it designs a kind of self-adaptive clone operator based on the distance between antibodies. The amounts of clone and excitation are assigned among antibodies adaptively. It simulates the dynamic process of adjusting the amount of antibody clones adaptively. Simulation tests of FT10 and FT06 show that the algorithm is stable and good.

【Key words】 artificial immune; clone selection algorithm; immune scheduling

1 概述

作业车间调度(Job Shop scheduling)是比较难的组合优化问题之一。目前, 该问题的求解方法主要归为数学规划方法、启发式算法、人工智能方法 3 类。其中比较典型的方法有模拟退火(SA)、遗传算法(GA)、禁忌搜索(TS)、启发式搜索、神经网络等。近些年的研究表明, 并不存在一个全局最优的调度规则。启发式规则由于简单、易于实现、计算复杂度低, 能够用于动态实时调度系统中, 因此一直受到广泛的研究。但它是局部优化方法, 难以得到全局优化结果, 并且不能对得到的结果进行次优性的定量评估。遗传算法具有较强的全局搜索能力, 将启发式算法与遗传算法相结合能够获得优于传统算法的结果。生物免疫系统是大规模并行、自适应信息处理系统的样本。近年来新兴的人工免疫系统理论被用于解决模式识别、计算机安全、自动控制、数据挖掘等多种计算技术问题, 其计算并行性和计算效率往往好于遗传算法, 全局搜索能力强, 可以明显改善遗传算法解过早收敛现象。将免疫算法与启发式算法结合形成混合算法, 可为作业车间调度问题提供一种新的研究途径。

2 Job Shop 调度问题描述

有 m 台不同的机器和 n 个不同的工件, 每个工件包含多个由多道工序组成的工序集合, 工件的工序顺序是预先给定的。每道工序用它所要求的机器和固定的加工时间表示。此外, 对工件和机器有如下约束: (1) 一个工件不能 2 次访问同一台机器; (2) 不同工件的工序之间没有先后约束; (3) 工序一旦进行不能中断; (4) 每台机器一次只能加工一个工件。

Makespan 为调度的最小生产周期, 即所有工件的最大完成时间的最小值, 调度问题要求确定机器上工件加工顺序, 使 *Makespan* 时间达到最小。上述问题的数学描述如下:

$$Makespan = \min_{1 \leq k \leq m} \max_{1 \leq i \leq n} \{c_{ijk}\} \quad (1)$$

$$c_{ik} - t_{ik} + M(1 - a_{ihk}) \quad c_{ih}, \quad i = 1, 2, \dots, n, \quad h, k = 1, 2, \dots, m$$

$$c_{jk} - c_{ik} + M(1 - x_{ijk}) \quad t_{jk}, \quad i, j = 1, 2, \dots, n, \quad k = 1, 2, \dots, m$$

$$c_{ik} \quad 0, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, m$$

$$x_{ijk} = 0 \text{ 或 } 1, \quad i, j = 1, 2, \dots, n, \quad k = 1, 2, \dots, m$$

Venn 将调度分为活动调度、半活动调度、非延迟调度^[1], 最优调度产生在活动调度集中。本文将最优调度的产生映射到(0,1)实数区间, 将调度问题的解作为抗体。设抗体 i 邻域搜索的范围为 $Neigh(i) \in (min, max)$, 邻域搜索变化步长为 $Neigh_step$ 。首次搜索令 $Neigh(i) = min$, 下次抗体 i 搜索范围变为 $Neigh(i) = min + Neigh_step$, 通过搜索步长不断调整抗体 i 的搜索范围, 直至完成全部搜索任务。

3 自适应克隆启发算法

3.1 抗体亲和力

将优先分配启发式算法获得的最小生产周期 *Makespan*

基金项目: 国家自然科学基金资助项目(60305007); 黑龙江省归国留学人员基金资助项目(LC05C03); 黑龙江省科技公关基金资助项目(GC06A118)

作者简介: 徐立芳(1973 -), 女, 讲师、硕士, 主研方向: 模式识别, 智能系统; 莫宏伟, 教授、博士

收稿日期: 2008-07-20 **E-mail:** mxlfang@163.com

的倒数作为抗体亲和力评价指标：

$$Ab_aff_i = \frac{k}{Makespan} \quad (2)$$

其中， Ab_aff_i 为抗体 i 的亲和力； k 为比例调节因子。

3.2 抗体编码

采用实数编码，抗体 i 为 $Ab_i = \{ab_1, ab_2, \dots, ab_i, \dots, ab_n\}$ ， $ab_i \in (0, 1)$ ，基因 ab_i 为区间 $(0, 1)$ 上任意实数。其中， $n = Ab_codel$ ； n 为抗体编码长度。抗体中每一个基因对应一个零件加工工序的优先权， n 为全部工序数量。根据优先权的大小将重新产生一个基因降序排列顺序，即调度顺序。

3.3 算法实现

本文的自适应克隆启发算法(Self-adaptive Clone Heuristic Algorithm, SACHA)步骤如下：

步骤 1 抗体邻域初始化

确定抗体变量范围 Ab_var 、解邻域 $Neigh$ 变化范围、邻域搜索变化步长 $Neigh_step$ 。

步骤 2 产生初始解群体

确定解群体中的抗体数 Ab_size 、抗体编码长度 Ab_codel 及抗体克隆总量 Sum_clone 。采用实数编码，随机生成解群体 $Pop = \{Ab_1, Ab_2, \dots, Ab_l\}$ ，其中， $l = Ab_size$ 。

步骤 3 计算抗体亲和力

最小生产周期 $Makespan$ 通过优先分配启发式算法获得，按照 3.1 节定义计算 Pop 中的抗体亲和力。

步骤 4 抗体交叉

在 Pop 中随机选择 2 个抗体，以交叉率 p_c 进行算数交叉，其中， Ab_i^t, Ab_i^{t+1} 分别表示交叉前和交叉后的抗体 i ； μ 是交叉调节参数。

步骤 5 抗体选择

若交叉后抗体的亲和力高于 Pop 中抗体的亲和力，替换 Pop 中较差抗体，更新解群体为 Pop' 。

步骤 6 抗体克隆

将 Pop' 中抗体进行克隆，克隆算子设计如下：

(1) 定义抗体间距离

采用欧几里德距离定义抗体 i, j 间的距离：

$$Ab_dis_{ij} = \left\| Ab_aff_i - Ab_aff_j \right\|^{1/2} \quad (3)$$

(2) 定义抗体激励度

亲和力高的抗体克隆的数量较多，定义抗体激励度，并且抗体克隆数量与抗体的激励度成正比。

$$Ab_stimul_i = \frac{Ab_aff_i}{\sqrt{S_1/S_2 + 1}} \quad (4)$$

$$\begin{cases} S_1 = \sum_{j=1}^{Ab_size-m-1} (Neigh(i) - Ab_dis_{ij})^2 & Ab_dis_{ij} \leq Neigh(i) \\ S_2 = \sum_{j=1}^m (Ab_dis_{ij} - Neigh(i))^2 & Ab_dis_{ij} > Neigh(i) \end{cases}$$

其中， S_1 表示全部抗体间距小于等于搜索邻域的情况； S_2 表示全部抗体间距大于搜索邻域的情况； m 为 S_2 出现的次数，由于 Pop' 中 S_1 与 S_2 的情况对立，因此 S_1 情况出现的次数为 $Ab_size - m - 1$ 。

在 S_1 情况下，抗体 $Ab_dis_{ij} \leq Neigh(i)$ ，此时的抗体 j 与抗体 i 非常近似，减少该类抗体克隆的数量，式(4)中 S_1 与 Ab_stimul_i 成反比。在 S_2 情况下， $Ab_dis_{ij} > Neigh(i)$ ，此时抗体 j 与抗体 i 差异较大，增加该类抗体克隆数量，式(4)中 S_2

与 Ab_stimul_i 成正比，丰富 Pop' 中抗体的多样性。

(3) 确定抗体克隆数

抗体依据自身激励度 Ab_stimul_i ，按一定比例分配预设抗体克隆数，抗体 i 进行克隆的数量为

$$Ab_clone_num_i = Int \left(Sum_clone \times \frac{Ab_stimul_i}{\sum_{j=1}^{Ab_size} Ab_stimul_j} \right)$$

其中， Sum_clone 为抗体克隆总量，即抗体预设克隆规模； $Int()$ 为对括号内的数向下取整。

步骤 7 抗体变异

对克隆后的抗体以变异率 p_m 进行均匀变异。

步骤 8 抗体选择

重新计算变异后每个抗体的亲和力，若变异后抗体的亲和力高于 Pop' 中抗体的亲和力，替换 Pop' 中较差抗体，更新解群体为 Pop'' 。

为了保证抗体的多样性，按概率 $Accept_rate$ 保留变异后部分亲和力较低抗体，替换 Pop'' 中较差的抗体，再次更新解群体为 Pop''' 。

步骤 9 变邻域

按邻域变化步长 $Neigh_step$ 改变下一代抗体邻域搜索范围，即改变抗体群中各抗体基因数值的取值范围，同时令 $Pop''' = Pop$ 。

步骤 10 判断是否满足进化代数

如满足进化代数，转步骤 11；否则，转步骤 4。

步骤 11 抗体译码，获得问题的解。

4 实验

4.1 实验参数

表 1 为 SACHA 相关实验参数的设置。

表 1 SACHA 的实验参数

	比例因子 k	Ab_var	$Neigh$	$Neigh_step$
FT06	10	(0, 1)	(0.005 0, 0.05)	0.000 50
FT10	100	(0, 1)	(0.000 5, 0.05)	0.000 05
	p_m	p_c	Ab_codel	Ab_size
FT06	0.001	0.3	36	50
FT10	0.001	0.3	100	30
	代数 G	μ	$Accept_rate$	Sum_clone
FT06	100	0.5	0.000 1	60
FT10	1 000	0.5	0.000 1	100

4.2 实验结果

表 2 和表 3 为实验结果。模拟退火(SA)和禁忌搜索(TS)方法实验对比数据来源于文献[2-3]，遗传启发算法(GHA)是本文将优先启发算法与遗传算法结合形成的混合算法。

表 2 FT06 实验结果

	SA	TS	GHA	SACHA
最小 $Makespan$	55	55	55	55
平均 $Makespan$	-	55	55	55
标准差	-	0	0	0

表 3 FT10 实验结果

	SA	TS	GHA	SACHA
最小 $Makespan$	1 089	963	1 024	960
平均 $Makespan$	-	985.90	1 045.75	970.25
标准差	7.34	10.46	18.64	8.12

图 1 和图 2 分别为 SACHA 用于 FT06, FT10 问题时最小生产周期 $Makespan$ 的搜索情况。

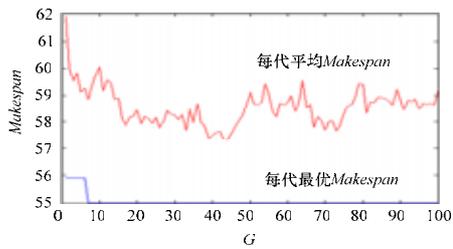


图1 FT06 最小生产周期 Makespan 搜索

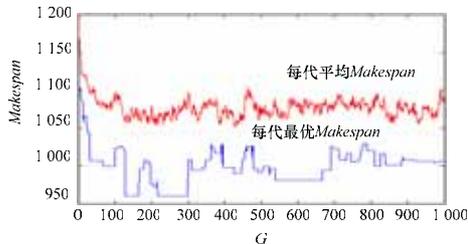


图2 FT10 最小生产周期 Makespan 搜索

4.3 实验分析

如表 2 和图 1 所示,在解决规模为 6×6 的 FT06 问题时, SACHA 能够在种群进化 10 代内准确找到 *Makespan* 的全局最小值。因此,对于解决小规模 Job Shop 问题, SACHA 性能优秀。

如表 3 和图 2 所示, SACHA 能够在 200 代内找到 *Makespan* 全局最小值 960, 优于模拟退火、禁忌搜索方法及遗传启发算法的结果。

如图 1 和图 2 所示, SACHA 搜索到全局最小 *Makespan* 后, 每代抗体的平均最小生产周期变化仍很活跃, 表明 SACHA 维持种群多样性能力较强, 能在一定程度上避免解

(上接第 206 页)

可以看出,对于测试函数 f_1 和 f_5 , HOGA 算法得到的标准差虽然大于 LEA 算法和 OGA/Q 算法,即本文算法得到的最优解的波动性比其他 2 种算法大,但是函数的平均计算次数分别减少 68% 和 35% 左右,且获得的最优解的精度明显优于其他 2 种算法,这说明 HOGA 算法的收敛速度和搜索精度明显优于其他 2 种算法。其中,测试函数 f_1 存在多个局部极值点,测试函数 f_5 拥有 100! 个局部极值点,搜索过程易陷入局部最优。因此, HOGA 算法还具有较强的多峰搜索能力。对于 f_6 和 f_7 , 本文的算法同 LEA 和 OGA/Q 算法比,函数的平均计算次数分别减少了 78% 和 46% 左右,而函数寻优精度却显著提高。对于测试函数 f_2, f_3, f_4 , 本算法优化的结果不论在函数的平均计算次数、最优平均值还是标准方差方面,都要优于其他 2 种算法。

综上所述,将正交交叉算子和本文提出的聚类局部搜索算子有机结合起来而形成的混合正交遗传算法,在求解复杂的高维函数优化中,显示出了良好的性能。

4 结束语

本文在正交遗传算法的基础上引入局部搜索策略,提出一种新的聚类局部搜索算子。实验结果表明,在高维的全局函数优化中,新的算法不仅具有好的全局搜索能力,而且表现了良好的收敛速度、搜索精度和多峰搜索能力。如何将新的局部搜索算子同约束处理技术及多目标处理技术相结合,

早熟。

寻找车间调度的最优解从理论上是 NP 完全问题, FT10 问题历来是较难处理的复杂调度问题,很难找到最优解,表 3 中的 4 种算法都未能找到最优解。由于调度问题的复杂性,实际生产调度的目标应定为寻找一个好的、可行的解决方案,即寻找最优解或次优解。目前还没有一个算法能很好地解决调度问题, SACHA 算法的实现为调度问题的解决提供了新的思路。

5 结束语

本文提出了一种新的自适应克隆启发算法用于 Job Shop 问题的求解。SACSA 采用实数编码,避免了二进制编码面临的“维数灾难”问题;设计了一种新的自适应克隆算子,模拟免疫系统自适应地调节抗体克隆数量过程。在种群进化环节,算法保留了基因遗传的特点,采用交叉和变异来进一步改善种群中抗体的多样性,扩大抗体搜索范围。SACHA 有效地利用了其良好的全局搜索和局部搜索特性,维持种群多样性能力较强,可以对复杂问题进行深度搜索,在一定程度上避免了目标函数值陷入局部极小。通过 FT10 和 FT06 仿真实验表明, SACHA 性能稳定,效果良好,可以作为一种求解 Job Shop 问题的新方法。

参考文献

- [1] Baker K, Scudder G. Sequencing with Earliness and Tardiness Penalties: A Review[J]. Operations Research, 1990, 38(10): 22-36.
- [2] 余建军, 孙树栋, 郑峰. 基于动态评价免疫算法的车间作业调度研究[J]. 机械工程学报, 2005, 41(3): 25-30.
- [3] 潘全科, 王文宏, 朱剑英. 一类解决 Job Shop 问题的改进遗传算法[J]. 中国机械工程, 2006, 8(17): 866-869.

来处理高维的约束优化和多目标优化问题,是进一步研究的方向。

参考文献

- [1] Wang Yuping, Dang Chuangyin. An Evolutionary Algorithm for Global Optimization Based on Level-set Evolution and Latin Squares[J]. IEEE Trans. on Evolutionary Computation, 2007, 11(5): 579-595.
- [2] Sun Jianyong, Zhang Qingfu, Tsang E P K. DE/EDA: A New Evolutionary Algorithm for Global Optimization[J]. Information Sciences, 2005, 169(3): 249-262.
- [3] Leung Yiuwing, Wang Yuping. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization[J]. IEEE Trans. on Evolutionary Computation, 2001, 5(1): 41-53.
- [4] Tsai J T, Liu T K, Chou J H. Hybrid Taguchi-genetic Algorithm for Global Numerical Optimization[J]. IEEE Trans. on Evolutionary Computation, 2004, 8(4): 365-377.
- [5] 曾三友, 魏巍, 康立三, 等. 基于正交设计的多目标演化算法[J]. 计算机学报, 2005, 28(7): 1153-1162.
- [6] Wang Yong, Liu Hui, Cai Zixing. An Orthogonal Design Based Constrained Optimization Evolutionary Algorithm[J]. Engineering Optimization, 2007, 39(6): 715-736.

