

文章编号: 1000-582X(2011)10-061-07

改进免疫克隆算法的 Job Shop 调度

刘爱军¹, 杨育¹, 邢青松¹, 姚豪¹, 张煜东² 周振宇²

(1. 重庆大学 机械传动国家重点实验室 重庆 400044; 2. 脑图像实验室, 哥伦比亚大学, 美国 纽约 10032)

摘要:提出了基于种群协同进化的并行免疫克隆算法, 将种群中个体的亲和度计算并行在多个计算节点上同时进行。引入免疫记忆机制, 使抗体种群的演化过程和记忆单元的演化过程并行进行, 更好地实现了抗体间的相互协作, 保证了解集从可行域内部和不可行域边缘向着最优解逼近。采用了克隆增殖变异和交叉算子的操作, 增加了种群中优秀个体获得克隆增殖实现亲和度成熟的机会, 提高抗体群分布的多样性, 在深度搜索和广度寻优之间取得了平衡。从而保证了算法较强的收敛性以及搜索空间的多样性。利用标准问题库对算法进行测试, 并分析算法参数对算法结果的影响, 仿真结果表明, 该算法对待寻优空间的全局搜索能力和局部搜索能力以及算法的稳定性与计算速率都要强于简单免疫克隆算法和遗传算法等优化算法。

关键词:生产控制; 调度算法; 并行免疫克隆算法; 协同优化; 克隆激励

中图分类号 TP278

文献标志码: A

Job-Shop scheduling based on improved immune cloning algorithm

LIU Ai-jun¹, YANG Yu¹, XING Qing-song¹, ZHANG Yu-dong², ZHOU Zhen-yu²

(1. State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, P. R. China; 2. Brain Imaging Laboratory, Columbia University, New York, NY 10032, U. S. A.)

Abstract: Parallel immune clone algorithm is proposed based on population coevolution theory and parallel computing affinity of individual at multiple compute nodes. Introducing the immune memory mechanism, the evolution processes of antibody population and memory units are conducted simultaneously, meanwhile, it improves mutual cooperation among antibodies, and ensures solution set approaching optimal solution from the inside of feasible region or infeasible region border. Clone proliferation, high frequency variation and operation of crossover operators increase the chance that better individuals gain affinity maturation by the operation of clone expansion, improve diversity of antibody population distribution, achieve the balance of optimization between depth and range, and ensure the convergence of the algorithm and the diversity of the search range. A computational study for a standard data set is carried out to test the validity of the algorithm, and the effect of algorithm parameters on the results is analyzed. The simulation results show that the global search capability, local search capability, algorithm stability and computing speed of the algorithm are all superior to conventional optimization algorithms such as normal immune clone optimization algorithm, genetic algorithm, etc.

Key words: production control; scheduling algorithms; parallel immunity clone algorithm; coordination optimization; cloning incentive

收稿日期: 2011-04-03

基金项目: 国家自然科学基金资助项目(70601037, 71071173); 新世纪优秀人才支持计划资助项目(NCET-07-0908); 重庆大学“211工程”三期创新人才培养计划建设项目(S-09107); 中央高校基本科研业务费科研专项自然科学基金面上项目(CDJZR10110012); 重庆大学研究生科技创新基金资助项目(201005A1B0010334)

作者简介: 刘爱军(1980-), 男, 重庆大学博士, 主要从事生产调度智能算法与仿真方向研究, (E-mail)liuajun@cqu.edu.cn.

生产调度问题是一类复杂的组合优化问题,属于 NP-hard 问题,随着调度问题规模的增大,解空间呈指数增长,求解难度急剧增加,传统优化方法无法得到优化解或不能保证在有效时间内获得优化解。以模拟退火(simulated annealing, SA)、禁忌搜索(tabu search, TS)、遗传算法(genetic algorithm, GA)和免疫算法(immune algorithm, IA)等为代表的智能优化方法受到研究者的关注^[1-6]。其中,遗传算法在生产调度领域的应用研究较早,相应的研究论文数量最多,但遗传算法存在:局部搜索能力较弱;易早熟收敛、后期搜索缓慢;popsiz (群体规模)、 P_s (选择概率)、 P_c (交叉概率)、 P_m (变异概率)对算法性能有重要影响等问题。尤以,局部搜索能力较差并容易早熟收敛是它的致命弱点。在对遗传算法研究的基础上,研究者将免疫概念及其理论应用于遗传算法并进行了改进,提出了免疫算法。免疫算法通过借鉴和利用生物免疫系统的性质和机制,不仅由于独特的多样性保持机制可以有效抑制算法的早熟现象,还可以通过有效的记忆机制大大加快系统对相同或相似抗原的二次反应速度,与模拟退火、禁忌搜索、遗传算法等几种智能优化方法相比,免疫算法在优化计算和生产调度问题等方面逐步得到应用并取得了良好的效果^[7-10],在免疫算法中,免疫克隆算法能够兼顾全局搜索和局部搜索特性,在函数优化、组合优化和数据挖掘等领域被广泛应用。在处理大规模生产调度问题时,使用免疫克隆这类智能优化算法进行特征优选有一个共同的缺点就是计算亲和度的工作量较大,使其实用性受到了很大限制。为了克服这一缺点,引入并行技术,提出了一种用于特征选择的并行免疫克隆算法,将种群中个体的亲和度计算并行在多个计算节点上同时进行,且保证在深度搜索和广度寻优之间取得平衡,从而能较快地获得较具代表性的特征子集,降低了算法运行时间。最后通过标准调度问题验证,改进免疫克隆算法的全局搜索能力和局部搜索能力以及算法的稳定性与计算速率都要强于简单免疫克隆算法和遗传算法等优化算法,实验结果验证了算法的正确性和有效性。

1 改进免疫克隆算法

克隆选择原理最先由 Jerne 提出, LN De Castro 基于免疫系统的克隆选择理论提出了克隆选择算法^[11]。免疫克隆算法把实际求解问题当作抗原,相应的解当作抗体,经过克隆增殖、交叉和变异趋向于亲合度成熟(affinity maturation)的过程^[12],即对抗

原亲合度较低的个体在克隆选择机制的作用下,经历增殖复制和变异操作后,其亲合度逐步提高而“成熟”的过程。由于亲合度计算的工作量较大,算法运行时间较长,实用性受到了很大限制,为此,本文提出了融入并行计算和种群协同竞争思想的改进免疫克隆算法。算法流程如图 1 所示:

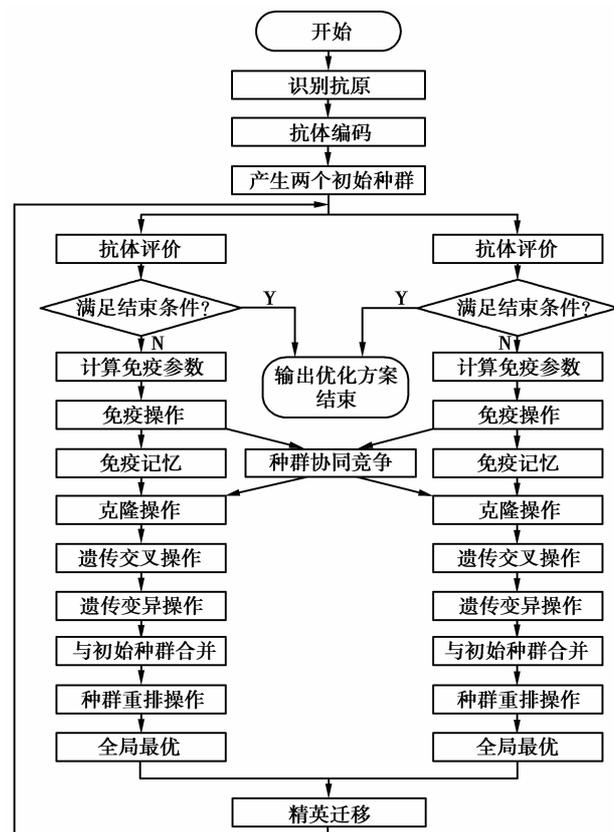


图 1 改进免疫克隆算法

首先产生 2 个初始种群,经过免疫操作以后,比较两种种群解集的特性,有选择地通过克隆复制来增加种群数目,扩大搜索空间,再经过遗传交叉和变异操作来增加种群的多样性,跳出局部最优,获得种群全局最优。经过精英迁移操作加快整体收敛速度。其中,在变异过程中,保持初始个体不参与变异操作,以防止破坏最优解。对变异后的个体进行选择操作,将最优的个体作为新的初始种群来加快收敛速度^[13-17]。改进免疫克隆算法是通过采用交叉、变异等遗传算子和相应的群体控制机制实现的,算法具体描述如下:

步骤 1 初始化抗体群 A_{b1} 和 A_{b2} , 分别随机产生 N 个抗体,生成初始群体;

步骤 2 分别计算 A_{b1} 和 A_{b2} 中抗体与抗原的亲合力 A_{if_1} 和 A_{if_2} , 及抗体与抗体之间的亲合力 A_{yy_1}

和 A_{yy_2} ;

步骤 3 引入协同竞争机制,通过两种群每个抗体目标函数值的大小对比分析来调整抗体与抗原的亲合力;

步骤 4 对 A_{b1} 中的抗体按照亲和力由大至小按降序排列,从中选取前 T 个抗体按照克隆扩增操作进行克隆,得到规模为 MM 的抗体群 A_{bc} ;

步骤 5 计算克隆后各抗体的亲和力 A_{ff_MM1} ,然后克隆交叉;

步骤 6 计算交叉后各抗体的亲和力 A_{ff_MM3} ,然后针对每个抗体比较 A_{ff_MM1} 和 A_{ff_MM3} ,选择亲和力高的抗体来代替当前种群中亲和力低的抗体;

步骤 7 进行变异,计算变异后各抗体的亲和力 A_{ff_MM2} ,然后针对每个抗体对比 A_{ff_MM1} 和 A_{ff_MM2} ,选择亲和力高的抗体来代替当前种群中亲和力低的抗体;

步骤 8 对抗体群 A_{bc} 中的抗体按照浓度由大至小按降序排列,进行删除操作,得到规模为 N_c 的抗体群 A_{be} ;

步骤 9 合并抗体群 A_b 和 A_{be} ,得到数量 N_{N_c} ($N_{N_c} = N + N_c$) 个抗体组成抗体群 A_{bc} ;

步骤 10 随机产生规模为 N_r 的抗体群 A_{br} ,选出亲和力最高的 N_s 个抗体组成抗体群 A_{bs} ;

步骤 11 用 A_{bs} 代替 A_{bc} 中亲和力最低的 N_s 个抗体,形成规模为 N 的抗体群 A_b ;

步骤 12 计算抗体群 A_b 目标函数,并且保存最优抗体 A ;

步骤 13 抗体群 A_{b2} 进行步骤 4-12 的操作,并且保存最优抗体 B ;

步骤 14 进行最优抗体 A 和 B 的交换;

步骤 15 判断是否满足终止条件,不满足则转至步骤 2 继续执行,满足则结束计算。

其中结束条件:①如果迭代次数 IterativeTime 大于某个最大阈值(即最大迭代次数),则算法结束;②如果最优值几代不变,则算法结束。

1.1 抗体与抗原之间亲和力计算

抗体 x_i 与抗原的亲合力 a_{fi} ($i=1,2,\dots,N$) 表示抗原与抗体的结合程度。

$$a_{fi} = \frac{1}{1 + o_{pti}} \quad (1)$$

如果生产调度问题的目标函数 o_{bj} 为最小值,则 $o_{pti} = o_{bj}$,如果为求最大值,则 $o_{pti} = 1/o_{bj}$ 。

1.2 抗体之间的亲和力

抗体和抗体之间的匹配程度可以用抗体基因间的亲和力表示。设有 N 个抗体,每个抗体有 M 个基因,如图 2 所示,每个抗体对应着一个解。抗体 x_i

和 x_j 的亲合力 $a_{yy_{ij}}$ ($i,j=1,2,\dots,N$) 表示抗体之间的相似程度,可以借用信息熵的理论进行计算。图中标为 K_1, K_i 和 K_N 的单元称为来自第 j 个基因的等位基因,其中 $\{K_1, \dots, K_i, \dots, K_N\} \in S$, 这样,整个抗体群中所有抗体的第 x 位基因的信息熵可以用式(2)计算^[12-13]

$$H_x(N) = - \sum_{i=1}^N p_i(x) \log p_i(x) \quad (2)$$

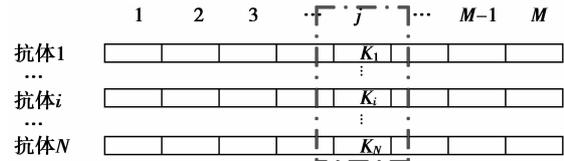


图 2 基因的信息结构

式中: $x \in \{1, 2, \dots, M\}$, $p_i(x)$ 是字符集 S 中的某个字符在第 i 个抗体的第 x 位基因出现的概率,如果在位置 x 上所有的等位基因都相同,那么 $H_x(N) = 0$,多样性的平均信息熵为

$$H(N) = \frac{1}{M} \sum_{x=1}^M H_x(N) \quad (3)$$

根据熵的定义,得到两个抗体 x_i 和 x_j 的亲合力

$$a_{yy_{i,j}} = \frac{1}{1 + H(2)} \quad (4)$$

其中 $H(2)$ 为抗体 x_i 和 x_j 的平均信息熵,当 $H(2)$ 为 0,即抗体 i 与 j 完全相同时,其亲合力达到最大值 1。 $a_{yy_{i,j}}$ 的取值范围是 $(0, 1]$, $a_{yy_{i,j}}$ 越大表示两个抗体基因相似度越高, $a_{yy_{i,j}} = 1$ 则表示两者的基因完全相同。

抗体浓度也可以用抗体间的亲和力来表示。抗体 i 与抗体群中其他抗体的亲和力之和为:

$$a_{yy_i} = \sum_{j=1, j \neq i}^N (a_{yy_{ij}}) \quad (5)$$

这样,抗体 i 的浓度可以表示为:

$$N_{ong_i} = a_{yy_i} / \sum_{j=1}^N a_{yy_j} \quad (6)$$

1.3 种群协同竞争伪代码

为了实现改进免疫克隆算法的快速收敛,引入了种群间协同竞争机制,以 2 个种群的目标函数值作为比较的对象,考虑反馈系数,采用抗体与抗原的亲合力为衡量的标准。种群间协同竞争伪代码如下:

```
Define feed1, feed2, g1=0, g2=0;
While 不满足条件 do
```

如果, $OBJV1(1,i) < OBJV2(1,j)$

$g1 = g1 + 1;$

$feed1(1,i) = Pr * g1 * Aff1(1,i)/N;$ Aff1

$(1,i) = Aff1(1,i) + feed1(1,i);$

While 不满足条件 do

如果, $OBJV2(1,i) < OBJV1(1,j)$

$g2 = g2 + 1;$

$feed2(1,i) = Pr * g2 * Aff2(1,i)/N;$ Aff2

$(1,i) = Aff2(1,i) + feed2(1,i);$

其中 Pr 表示反馈概率, 取为 1。OBJV1 和 OBJV2 分别表示种群 1 和种群 2 的目标函数值。

1.4 克隆竞争扩增操作

$$N_i(k) = \text{round} \left[N \times \frac{a_{\text{ff}}(A_i(k))}{\sum_{j=1}^T a_{\text{ff}}(A_j(k))} \right]. \quad (7)$$

其中, N_i 为第 i 个抗体的克隆规模, $\text{round}(\cdot)$ 为取整函数, 且 $i=1, 2, \dots, T$

$$M_M = \sum_{i=1}^T N_i. \quad (8)$$

其中 M_M 为克隆后的抗体数量, 亲和力越高的抗体, 克隆规模越大, 克隆出的抗体就越多。经过竞争克隆后, 原来的 T 个优秀的抗体就分别扩张成为了 T 个规模分别为 $N_i (i=1, 2, \dots, T)$ 的小抗体群。则优秀抗体群 A_{b1} 变为了规模为 N_c 的克隆抗体群

$$A_{bc} = \{A_{bc1}, A_{bc2}, \dots, A_{bcT}\}. \quad (9)$$

其中 $A_{bc i} (i=1, 2, \dots, T)$ 表示 A_{b1} 中第 i 抗体 $A_{b1 i} (i=1, 2, \dots, T)$ 经过克隆后形成的小抗体群, 其包含的抗体数量为 N_i 。

根据抗体的浓度 N_{omg_i} 先排除所得激增抗体群中的具有最高抗体, 然后再计算所得激增抗体群中剩余抗体的浓度, 依次排除浓度最高的抗体, 直到激增抗体群中剩余 N 个抗体为止, 此时产生的抗体群成为新一代。

2 Job-Shop 调度问题描述

Job-Shop 调度问题研究 n 个工件在 m 台机器上加工, 已知各操作的加工时间和各工件在各机器上的加工次序约束, 要求确定与工艺约束条件相容的各机器上所有工件的加工次序, 使所有工件的最大完成时间 (makespan) 最小的调度方案的数学描述如下^[16-17]: $J = (j_1, j_2, \dots, j_n), i=1, 2, \dots, n$ 为所有工件的一个排序, Π 为所有排序的集合, $C(j_i, k)$ 代表工件 j_i 在机器 k 上的完成时间, 则

$$C(j_1, 1) = T_{j_1, 1}. \quad (10)$$

$$C(j_1, k) = C(j_1, k-1) + T_{j_1, k}, k = 2, \dots, m. \quad (11)$$

$$C(j_i, k) = C(j_{i-1}, 1) + \sum_{l=1}^k T_{j_i, l} + \Delta(j_i),$$

$$i = 2, \dots, n, k = 1, 2, \dots, m. \quad (12)$$

其中

$$\Delta(j_i) = \max \{0, \max_{2 \leq k \leq m} [C(j_{i-1}, k) - (C(j_{i-1}, 1) + \sum_{l=1}^{k-1} T_{j_i, l})]\}, i = 2, \dots, n. \quad (13)$$

$$C_{\max}(J) = C(j_n, m). \quad (14)$$

$$J^* = \arg \{C_{\max}(J) = C(j_n, m)\} \rightarrow \min, \forall J \in \Pi. \quad (15)$$

式(14)即为最大完成时间, 式(15)表示最小化最大完成时间的调度, 同时需要满足以下约束条件。

1) 资源约束, 同一设备上上一个工序结束以后才能进行下一个工序。

$$\begin{cases} T_{s, J_i, j} \geq T_{f, J_p, q} \\ z_{m, J_i, l} = z_{m, J_p, q} = 1 \end{cases} \text{ 或者 } \begin{cases} T_{s, J_p, q} \geq T_{f, J_i, j} \\ z_{m, J_i, l} = z_{m, J_p, q} = 1 \end{cases}. \quad (16)$$

式(16)中

$$z_{m, J_i, l} = \begin{cases} 1, \text{操作 } Q_{J_i, l} \text{ 在设备 } m \text{ 上加工;} \\ 0, \text{其他.} \end{cases} \quad (17)$$

式(17)中: $Q_{J_i, l}$ 为第 j_i 工件第 l 个工序的操作; $T_{s, J_i, l}$ 为操作 $Q_{J_i, l}$ 的开始时间; $T_{f, J_i, l}$ 为操作 $Q_{J_i, l}$ 的完成时间; $T_{J_i, l}$ 为操作 $Q_{J_i, l}$ 的加工时间。

2) 工序约束: 一个工件只能在完成前一道工序以后才可进行后一道工序。

$$C_{J_i, l, k} - C_{J_i, (l-1), j} \geq P_{J_i, l, k} + S_{J_i, l, k}. \quad (18)$$

其中, $z_{k, J_i, l} = z_{j, J_i, (l-1)} = 1$ 。

3) 处理约束: 不同工件的工序之间没有先后约束, 同一工件的工序之间有先后约束。

4) 机器约束: 同一时刻一台机器只能加工一个工件。

2.1 调度编码

目前已经提出了基于工序、优先表、工件对关系、工件、完成时间、优先规则、非连接图、机器和随机键等 9 种 JSSP 编码方法。这里采用基于工序的表达法进行编码, 给所有同一工件的工序指定相同的符号, 然后根据它们在给定染色体中出现的顺序加以解释。对于 n 工件 m 机器问题, 一个染色体包括 $n * m$ 个基因, 每个工件出现在染色体中 m 次, 如在 $4 * 4$ 问题中, 假设给定的染色体为 [1 3 2 1 1 2 1 2 3 4 3 4 2 3 4 4], 其中第 1 个 1 表达工件 1 的第一个工序, 第 2 个 1 表示工件 1 的第 2 个工序。因为每个 1 件有 4 道工序, 所以每个工件在一个染色体中刚好出现 4 次。

2.2 遗传算子

基本免疫算法采用的是随机交叉和变异机制,这种方法不利用保留优秀个体。笔者融入了遗传交叉和变异算子,提高优良个体的继承性,从而缩短算法的运行时间,提高了收敛速度和质量。

目前遗传算子设计方法有单点交叉,基于位置的交叉、多点交叉、部分匹配交叉、插入变异、互换变异等。由于编码方法中基因的位置表示工序的优先调度顺序,所以只要改变各染色体中基因的位置,就可得到一种新的调度方案。具体的遗传算子方法如下。

根据生物遗传规律可知,若父母均为优良个体,则后代具备优良性质的可能性会很大。在交叉过程中,首先依交叉概率选择出需进行交叉的个体。设交叉概率为 P_c ,群体规模是 P_{opsize} ,则在区间 $[0, 1]$ 内选取 P_{opsize} 个随机数 $\text{rand}_i (i = 1, 2, \dots, P_{\text{opsize}})$,将 P_c 和 rand_i 进行比较,若 $P_c \geq \text{rand}_i$,则父子群体和母子群体中的第 i 个个体分别作为需进行交叉的两个父体,反之若 $P_c < \text{rand}_i$,则该个体对直接作为交叉后代进入变异过程。

对于得到的需进行交叉的两个父体,选取单亲交叉方法,从而根据相应的交叉过程得到两个交叉后代分别作为交叉后得到的父、母子代。

这种交叉方法中,仅有一个父代个体参与交叉,通过交叉过程的产生的子代个体也是一个。具体的交叉过程是:对于交叉父代 X ,随机选取一个交叉点,互换交叉点前后的基因得到新的交叉后代 X' 。按照上述交叉方法可得到群体规模为 P_{opsize} 的两个交叉后代群体。

设对于 6 个工序的问题,所选择的交叉的父、母体分别为 $X = [1, 3, 6, 2, 4, 5]$, $Y = [1, 2, 3, 6, 5, 4]$,则单亲交叉方法得到的后代 X' , Y' 如图 3 所示。

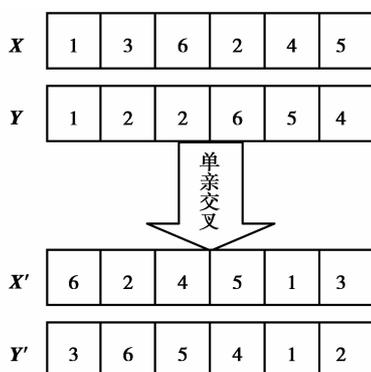


图 3 单亲交叉过程,交叉点为 3

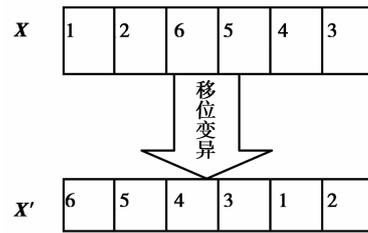


图 4 移位变异过程

变异过程同样引入竞争的机制,对于依变异概率 P_m 对需进行变异的个体采用移位变异。随机选择染色体中的一个子串,将该子串插入到一个随机的位置上。变异操作中进行互换的数目将由变异率决定,变异率越高,互换的数目将越高。

3 实验数据与分析

基于以上改进免疫克隆算法的研究,结合 job shop 生产调度问题的数学模型,用抗原表示待调度的生产计划,抗体为高效的生产调度结果,以 FT 调度标准数据为输入,以最小 Makespan 为目标,与协同粒子群、遗传算法和标准免疫克隆算法进行对比研究。

在生产调度仿真中,种群规模、交叉概率和变异概率对算法的搜索性能有重要影响,经研究发现:种群规模太小,搜索范围小,很难找到最优值,规模太大,增加搜索时间,影响算法效率;交叉概率太小则搜索过程缓慢,增加搜索时间,降低算法效率,交叉概率过大则可能对最优个体的结构造成破坏,从而无法搜索到全部最优个体;变异率太小则变异操作不明显,群体的多样性不能保证,变异率太大则群体的稳定性差,难以长时间稳定地收敛。

由上述原因,对算法参数进行如下设置:协同粒子群和遗传算法参数如表 1 所示;免疫克隆算法及其改进算法参数如表 2 所示。在此基础上,对协同粒子群、遗传算法、标准免疫克隆算法和改进免疫克隆算法分别进行 20 次运算,得到如表 3 和 4 所示的性能分析结果(其中 C 表示 FT 问题的理论最优解, J^* 表示算法运行 20 次搜索到的最优值, \bar{t} 表示算法运行 20 次的平均时间耗费, \bar{j} 表示运行 20 次的目标平均值, S^2 表示目标方差)及如图 5 和 6 所示的基于遗传算法及改进免疫克隆算法的 FT10 调度结果,且由表和图可知,改进免疫克隆算法的调度结果明显优于其他几种算法,由此,证明了改进算法的优良的搜索性能。在上述研究的基础上,对算法运算过程中平均亲合力和平均目标值进行跟踪,得出如

图 7-8 所示的运行结果,可知,运算过程实现了种群多样性与快速收敛的平衡。

表 1 CPSO GA 算法参数

算法	P_{size}	交叉概率(P_c) 或权重 w	变异(P_m)或 协同选择概率 P_a	迭代次数 g_{en_max}
CPSO	100	0.4	0.35	100
GA	40	0.8	0.6	250

表 2 免疫克隆算法及其改进算法参数

算法	P_{size}	替换 抗体 N_s	克隆选 择概 率 P_s	克隆抑 制半 径 P_y	交叉 概率 P_c	变异 P_m	迭代 次数 g_{en_max}
ICA	70	30	0.7	80	0.7	0.2	250
IICA	70	30	0.7	80	0.7	0.2	250

表 3 CPSO 和 GA 性能对比

参量	N, M	C	CPSO				GA			
			J^*	\bar{t}	\bar{j}	S^2	J^*	\bar{t}	\bar{j}	S^2
FT06	6,6	55	55	12.44	57	1.2	55	57.05	57.5	1.99
FT10	10,10	930	945	42.15	950	8.2	997	129.72	1001	12.4
FT20	20,5	1 165	1 190	47.11	1 195	9.2	1 247	141.42	1 250	14.4

表 4 ICA 和 IICA 性能对比

参量	N, M	C	ICA				IICA			
			J^*	\bar{t}	\bar{j}	S^2	J^*	\bar{t}	\bar{j}	S^2
FT06	6,6	55	55	11.54	56	0.8	55	12.79	55	0
FT10	10,10	930	936	40.54	940	8	930	70.45	930.5	1.5
FT20	20,5	1 165	1 180	47.16	1 187	14	1 167	204.73	1 169	7.6

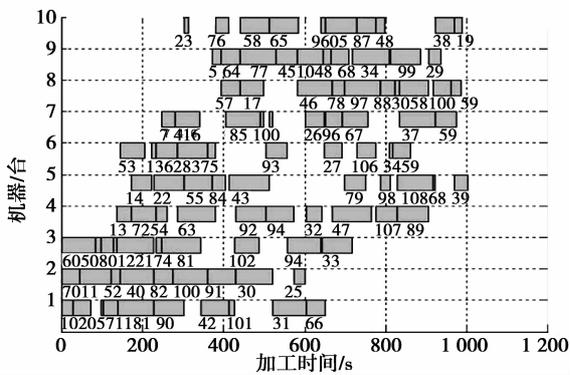


图 5 基于遗传算法的 FT10 调度结果

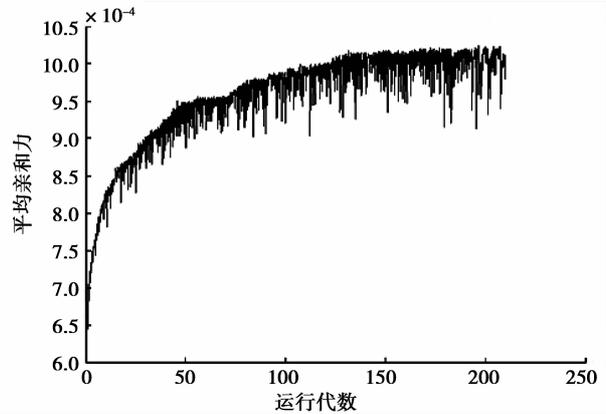


图 7 基于改进免疫克隆算法的平均亲和力

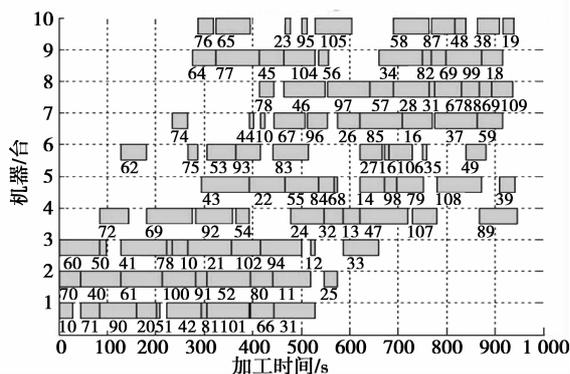


图 6 基于改进免疫克隆算法的 FT10 调度结果

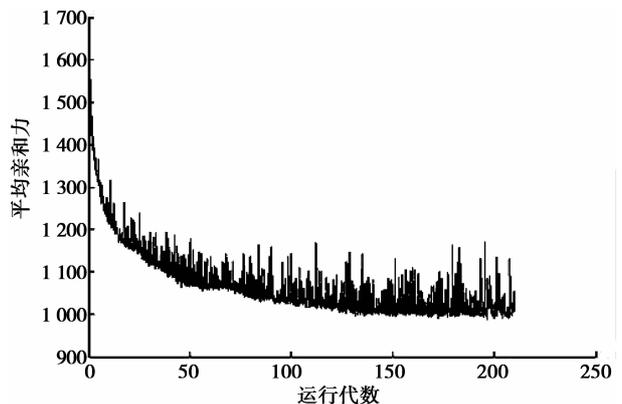


图 8 基于改进免疫克隆算法的平均目标值

4 结 论

提出基于种群协同进化的并行免疫克隆算法,实现了快速收敛与抗体群分布的多样性及深度搜索和广度寻优之间的平衡,很好地解决了维数灾难,降低了 Job Shop 问题的求解难度,提高了问题的求解效率,仿真结果证实了该算法适合求解大规模 Job Shop 调度问题。

参考文献:

- [1] ZHANG R, WU C. A hybrid immune simulated annealing algorithm for the job shop scheduling problem [J]. *Applied Soft Computing*, 2010, 10(1): 79-89.
- [2] ESWARAMURTHY V P. Tabu search strategies for solving job shop scheduling problems [J]. *Journal of Advanced Manufacturing Systems*, 2007, 6(1):59-75.
- [3] ZHANG C Y, RAO Y Q, LI P G. An effective hybrid genetic algorithm for the job shop scheduling problem [J]. *International Journal of Advanced Manufacturing Technology*, 2008, 39:965-974.
- [4] 郑忠,朱道飞,高小强. 钢厂炼钢-连铸生产调度及重计划方法[J]. *重庆大学学报*, 2008, 31(7):820-824.
ZHENG ZHONG, ZHU DAO-FEI, GAO XIAO-QIANG. An approach of production scheduling and replanning in a steelmaking-continuous casting plant [J]. *Journal of Chongqing University*, 2008, 31(7):820-824.
- [5] BAGHERI A, ZANDIEH M, MAHDAVI I, et al. An artificial immune algorithm for the flexible job-shop scheduling problem [J]. *Future Generation Computer Systems*, 2010, 26(4):533-541.
- [6] LUHA GUAN-CHUN, CHUEHB CHUNG-HUEL. A multi-modal immune algorithm for the job-shop scheduling problem [J]. *Information Sciences*, 2009, 179(10): 1516-1532.
- [7] 左兴权,莫宏伟. 免疫调度算法综述[J]. *控制与决策*, 2009, 24(12): 1761-1768.
ZUO XING-QUAN, MO HONG-WEI. Survey on immune scheduling algorithms [J]. *Control and Decision*, 2009, 24(12):1761-1768.
- [8] 张会红,顾幸生,汪鹏君. 基于免疫算法的生产调度现状与展望[J]. *计算机集成制造系统*, 2008, 14(11): 2081-2091.
ZHANG HUI-HONG, GU XING-SHENG, WANG PENG-JUN. Status & outlook of immune algorithm based production scheduling [J]. *Computer Integrated Manufacturing Systems*, 2008, 14(11): 2081-2091.
- [9] HART, E, ROSS P, NELSON J. Producing robust schedules via an artificial immune system [C]// *Proceedings of IEEE International conference on Engineering Computation*. AK, USA: [s. n.], 1998: 464-469.
- [10] HART, E, ROSS P. The evolution and analysis of a potential antibody library for job-shop scheduling[M]. London: *New Ideas in Optimization*. D. Corne, M. Dorigo & F. McGraw-Hil, 1999.
- [11] DE CASTRO L N, VON ZUBEN F J. The colonel selection algorithm with engineering applications [C]// *Proceedings of GEECCO, workshop on Artificial Immune Systems and the Applications*, San Francisco: Morgan Kaufmann Publishers, 2000:36-37.
- [12] 刘晓冰,吕强. 免疫克隆选择算法求解柔性生产调度问题[J]. *控制与决策*, 2008, 23(7): 781-785.
LIU XIAO-BING, LV QIANG. Immune clonal selection algorithm for flexible job-shop scheduling problem [J]. *Control and Decision*, 2008, 23(7): 781-785.
- [13] 杜广宇,王莉. 一种改进的人工免疫算法在模糊 Flow-shop 调度问题上的应用[J]. *鞍山科技大学学报*, 2005, 28(6):432-435.
DU GUANG-YU, WANG LI. Application of a modified artificial immune algorithm to fuzzy flow-shop scheduling problem[J]. *Journal of Anshan University of Science and Technology*, 2005, 28(6):432-435.
- [14] 余建军,孙树栋,刘易勇. 基于免疫算法的多目标柔性 job-shop 调度研究[J]. *系统工程学报*, 2007, 22(5): 511-519.
YU JIAN-JUN, SUN SHU-DONG, LIU YI-YONG. Multi-objective flexible job-shop scheduling based on immune algorithm [J]. *Journal of Systems Engineering*, 2007, 22(5): 511-519.
- [15] 杨建国,丁慧敏,李蓓智. 解决多目标 Flow-shop 问题的生物免疫调度算法[J]. *机械设计与研究*, 2002, 18(4): 28-31.
YANG JIAN-GUO, DING HUI-MIN, LI BEI-ZHI. An immune scheduling algorithm for solving multi-objective flow-shop problem[J]. *Machine Design and Research*, 2002, 18(4): 28-32.
- [16] 余建军,孙树栋,郝京辉. 免疫算法求解多目标柔性作业车间调度研究[J]. *计算机集成制造系统*, 2006, 12(10): 1643-1650.
YU JIAN-JUN, SUN SHU-DONG, HAO JING-HUI. Multi-objective flexible job-shop scheduling based on immune algorithm [J]. *Computer Integrated Manufacturing Systems*, 2006, 12(10): 1643-1650.
- [17] 马佳,高立群,石刚,等. 求解柔性作业车间调度问题的免疫遗传算法[J]. *东北大学学报:自然科学版*, 2008, 29(7):936-939.
MA JIA, GAO LI-QUN, SHI GANG, et al. Immune genetic algorithm the flexible job-shop scheduling problem [J]. *Journal of Northeastern University: Natural Science*, 2008, 29(7): 936-939.