

协同免疫克隆算法及其在零等待 flowshop 调度中的应用^①

新费慧^② 顾幸生^③

(华东理工大学自动化研究所 上海 200237)

摘 要 针对零等待 flowshop 调度问题,提出一种叫做协同免疫克隆算法(CICA)的新的有效的算法,该算法将克隆选择机制和免疫系统原理结合起来,在局部操作中引入激励度函数和新的亲和突变操作,使得抗体应答抗原的综合能力不仅和其亲和力有关,而且与其浓度也有关系,从而能增强抗体的多样性,防止早熟。在全局操作中加入了协同进化思想,在进化过程中进行精英迁移,以加快算法的收敛速度,最终达到优化的目的。仿真结果表明,该算法的收敛速度明显优于免疫算法和免疫克隆算法,验证了该算法的有效性和优越性。最后仿真讨论了激励度系数和反馈系数的选取对算法的影响。

关键词 生产调度,克隆选择,免疫算法,协同进化,零等待

0 引 言

生产调度是生产管理的核心内容和关键技术,其任务是在车间有限的资源约束下,确定工件在相关设备上的加工顺序和加工时间,以保证所选定的生产目标最优,使生产任务得以完成。

Flowshop 调度问题是一类具有 NP-hard 特性的重要生产调度问题^[1]。传统的数学规划受到问题维数的制约,无法解决较大规模的生产调度问题。越来越多的人开始探求更新的智能方法。免疫算法(immune algorithm, IA)就是近年来被应用于研究 flowshop 调度问题的一种新兴智能方法。免疫算法是一种确定性选择和随机性选择相结合并具有勘测与开采能力的启发式随机搜索算法,它被认为是对自适应免疫应答中体液免疫的简单模拟。而克隆选择使得亲和力较高的抗体确定性地参与进化,并将其繁殖克隆,为算法的局部搜索提供必要准备。2000 年 De Castro 将克隆选择的原理与独特的免疫原理结合,提出了一种新兴的进化算法——免疫克隆算法(immune clone algorithm, ICA)^[2],其灵感来自于生物获得性免疫的克隆选择原理^[3]。近年来, Wierzbion^[4]从克隆选择原理出发,提出了基于实数

编码的广义克隆选择算法,De Castro 等^[5]将克隆抑制机制与克隆选择机制结合起来,建立了基于实数编码的人工免疫网络模型,具有一定的通用性。随着研究的不断深入,刘丽珏等^[6]提出了基于变异向量集协同进化的免疫克隆算法,保证了算法能记忆和维持好的进化方向,提高了算法的收敛速度。同时,作为一种智能优化算法,免疫算法越来越多地运用于调度问题中,Hart^[7]最早将人工免疫算法用于优化生产调度问题中。近几年来,随着各种调度问题研究的深入,新的免疫算法被应用到各种复杂的生产调度中,徐震浩等^[8]借鉴自然界中生物免疫系统的概念和机理,提出了解决零等待问题的免疫算法,王伟等^[9]将人体内分泌激素调节系统与免疫系统结合起来,提出了解决不确定零等待问题的免疫算法,刘晓冰等^[10]将免疫克隆选择算法应用于柔性调度问题中。

针对免疫克隆算法的特点及不足,受进化算法的启发,本文提出了协同免疫克隆算法(co-evolutionary immune clone algorithm, CICA),该算法在全局操作中引入协同竞争的机制,在局部变异操作中添加了抗体的交叉变异操作和新的激励度评价函数。仿真结果验证了用这种新的算法来解决零等待 flowshop 生产调度问题的优越性和有效性。

① 863 计划(2009AA04Z141),国家自然科学基金(60774078),上海市基础研究重点项目(08JC1408200)和教育部博士点基金(200802510010)资助项目。

② 女,1986 年生,硕士生;研究方向:生产计划与调度,智能算法及应用;E-mail: jfh730@126.com

③ 通讯作者, E-mail: xsq@ecust.edu.cn

(收稿日期:2009-06-30)

1 零等待 flowshop 调度问题的数学模型

在流程工业中,间歇(批量)过程占有很大的比重。间歇化工过程在间歇级之间可以设立中间储罐来增加生产过程操作的灵活性^[1],依照中间储罐设置情况的不同,可以分为无限中间储罐、有限中间储罐、无中间储罐、零等待和混合中间储罐等 5 种方式。在有化学反应发生的间歇操作中,经常要求中间产物在某个设备处理完毕后,立即转移到下一个加工设备中去,不能有延误或者中间存储过程,这时生产就应当采用零等待方式。

零等待 flowshop 调度问题可以这样描述:有 N 个产品需要加工;可供选用的设备单元有 M 个,第 i 个被加工的产品在第 j 个设备上需要的加工时间为 T_{ij} ,每个产品的加工工序都相同,并且以相同的次序在各设备上加工;定义 S_{ij} 和 C_{ij} 分别表示工件 i 在设备 j 上的开始加工时间和完工时间, S_u 和 C_u 分别是工件 i 的最后一道工序的开始加工时间和处理时间,以最小化总加工周期为调度目标。

首先作如下假设:(1)所有产品在每个加工单元上的操作次序相同,即为排列排序;(2)产品之间没有优先性;(3)一个设备不能同时处理多种产品,一种产品不能同时在多个设备上处理;(4)对最终的产品有足够的存储容量;(5)产品的加工过程不允许中断。

以上的定义和假设用如下的数学模型来表示:

$$\min\{Z = \max(S_u + C_u)\} \quad (1)$$

$$\text{s.t. } S_{ij} = S_{i(j-1)} + T_{i(j-1)} \quad (2)$$

$$S_{ij} \geq S_{(i-1)j} + T_{(i-1)j} (i \in N, j \in M) \quad (3)$$

$$S_{ij} \geq 0 \quad (4)$$

由于产品的加工方式是零等待,在第一个加工单元上的产品的开始操作时间需要适当延迟。令加工顺序中产品 s 和产品 t 相邻,则两个产品的延长时间为

$$d_{st} = \max_{m=2, \dots, M} \left\{ 0, \sum_{k=2}^m T_{sk} - \sum_{k=1}^{m-1} T_{tk} \right\} (m = 2, \dots, M) \quad (5)$$

其中, k 为加工单元上处理产品的排序号。

由上述式子可推知所求解的目标函数为

$$\min(\text{makespan}) = \min(Z) \quad (6)$$

2 协同免疫克隆算法

2.1 免疫克隆算法

免疫克隆算法与一般的免疫算法相比具有以下的特点:(1)细胞克隆及亲和突变的协作体现了邻域搜索及并行搜索特性;(2)抗体的选择受其亲和力的制约;(3)搜索过程处于探测、选择、自我调节的协调合作过程,体现了免疫应答中抗体学习抗原的行动特性;(4)搜索过程开放,随时有自我抗体被加入进化群体,增强了群体的多样性。

在免疫克隆算法中,抗原与抗体代表着优化问题或进化群体中的最好解和优化问题的可行解,记忆细胞代表进化群体中较好的解,自我抗体代表优化问题的可行解。研究免疫克隆算法时,需要弄清下述相似度、亲和力、抗体浓度、激励度、克隆选择、细胞克隆等概念。

(1)相似度是指抗体与抗体的相似程度,采用信息熵来表示:如果群体中有 N 个抗体,每个抗体由 M 位基因组成,每位可供选择的字母表中共有 s 个字母,于是群体中基因座 j 的信息熵定义为

$$H_j(N) = - \sum_{i=1}^s p_{ij} \log p_{ij} \quad (7)$$

其中 p_{ij} 为 N 个抗体第 i 个符号出现在基因座 j 上的概率。

群体的平均信息熵可以表示为

$$H(N) = \frac{1}{M} \sum_{j=1}^M H_j(N) \quad (8)$$

则抗体与抗体之间的相似度可以表示为

$$ay_{v,w} = \frac{1}{1 + H(2)} (v, w \in N) \quad (9)$$

(2)亲和力指抗体与抗原的匹配程度,即每个可能解与目标函数的匹配程度。可以表示为

$$ay_v = \frac{1}{1 + opt_v} (v \in N) \quad (10)$$

式中 opt_v 表示抗体与抗原之间的结合度,本文采用总生产周期来表示 opt_v ,即 $opt_v = \text{makespan}$ 。所以对于生产调度问题,亲和力表示为

$$ay_v = \frac{1}{1 + \text{makespan}} (v \in N) \quad (11)$$

(3)抗体浓度指抗体在抗体群中与其相似的抗体所占的比例。可以表示为

$$C(u) = \frac{|\{v \in N \mid ay_{v,w} \leq \eta\}|}{N} \quad (12)$$

其中 η 为浓度阈值, $0 \leq \eta \leq 1$,在此称为浓度抑制半径。

(4)激励度^[12]是指抗体群中抗体应答抗原和被其他抗体激活的综合能力。其表示为

$$act(v) = ax_v e^{-\frac{C(v)}{\beta}} \quad (13)$$

其中 β 为调节因子, $\beta \geq 1$, 此函数的设计是为了调节抗体群的多样性, 而 β 的选取对算法性能有一定的影响, 在其后的仿真研究中将对其进行讨论。

(5) 克隆选择是指在给定的选择率 α 下, $0 < \alpha < 1$, 在抗体群中选择亲和力最高的 $N_0 = \text{round}(\alpha \times N)$ 个抗体进行克隆繁殖与突变。

(6) 细胞克隆是指在给定的繁殖数 M 下, 对选择出的 N_0 个个体按其亲和力及繁殖率繁殖 M 个抗体。繁殖率函数表示为

$$m_i = M \cdot \frac{ax(i)}{\sum_{i=1}^{N_0} ax(i)} \quad (14)$$

则实际克隆数为

$$M = \sum_{i=1}^{N_0} m_i \quad (15)$$

2.2 协同进化思想

协同进化算法 (co-evolutionary algorithm, CEA) 借鉴生态学的种群协同理论, 应用种群间自动调节、自动适应原理来构造彼此关联的种群, 共同进化求解。目前协同进化算法大致分为合作协同进化算法 (co-operative co-evolutionary algorithm, CoopEA) 和竞争协同进化算法 (competitive co-evolutionary algorithm, CompEA) 两种。竞争协同进化是近年来针对传统进化算法的不足而兴起的一个研究热点, 意指多个种群通过适应度的关联同时进化, 个体的适应度通过与来自另一种群的个体直接竞争得到。CompEA 多不对问题进行分解, 每个个体都是一个完整解, 因而导致个体编码过长, 搜索空间过大, 使得到的最终解质量不够优秀, 在此基础上文献 [13] 提出了结合竞争和合作的 Coop&CompEA 算法。

对种群 S_i 的每个个体 p_{ij} , 其竞争对手为 $S_{i'} (i' \neq i)$ 的所有个体 g 。定义 p_{ij} 的竞争结果值为

$$c(p_{ij}) = \sum_{g \in S_{i'}} \begin{cases} 1 & \text{如果 } p_{ij} \text{ 比 } g \text{ 好} \\ 0 & \text{否则} \end{cases} \quad (16)$$

可见 $c(p_{ij})$ 是一整数, 在 $[0, N]$ 区间, N 为种群 $S_{i'}$ 的大小 (含的个体的个数)。对种群中的所有个体计算其竞争结果值, 用于下一步的反馈。

竞争完成后, 把得到的竞争结果值反馈给相应的个体。给出竞争反馈值 $feed(p_{ij})$ 为

$$feed(p_{ij}) = r \cdot \frac{c(p_{ij})}{N} \cdot Aff(p_{ij}) \quad (17)$$

其中, $Aff(p_{ij})$ 为当前的亲和力, r 称为反馈因子, 目的是调节反馈的大小。 $\frac{c(p_{ij})}{N}$ 是一个 $[0, 1]$ 区间的数。

竞争之后修正的亲和力为

$$Aff'(p_{ij}) = Aff(p_{ij}) + feed(p_{ij}) \quad (18)$$

3 协同免疫克隆算法 (CICA) 在零等待调度中的应用

协同免疫克隆算法是在免疫克隆算法的基础上采用了新的亲和突变操作, 使得进行变异后的优秀抗体间进行部分基因位交叉, 相对传统的抗体单一的自身变异过程, 能促进优秀抗体间信息的交流, 进一步提高算法的收敛速度和局部搜索的能力。加入激励度评价函数进行克隆抑制, 使得抗体应答抗原的综合能力不仅和其亲和力有关, 而且与其在群体中的浓度也有关系, 提高了抗体应答抗原的综合能力。同时加入种群间协同竞争的思想, 应用种群间自动调节、自动适应原理来构造彼此关联的种群, 共同进化求解。这样保证了群体多样性, 有效克服个体早熟的现象。本文算法主要由初始化、细胞克隆、细胞变异和交叉、克隆抑制和免疫选择、募集新成员和协同竞争这几部分组成, 具体步骤如下:

(1) 编码。由于调度问题是一类有序的操作, 它的目标函数值不仅与解的数值有关, 而且与其在编码字符串中的位置有关, 所以在 CICA 调度算法中, 采用字符编码的方法来表示抗体。即根据调度问题的生产特性, 以每个字符代表一个加工产品, 编码中的每个产品只能出现一次, 字符在编码中出现的顺序就是加工产品的顺序。

(2) 参数设置。定义抗体初始规模、期望克隆规模、克隆选择率、抑制半径、激励度、免疫选择率、最大迭代次数等参数。

(3) 初始化。随机产生 N 个抗体构成初始抗体群 A_1, A_2 。利用式 (9)、(11) 计算得到两个种群中各抗体的亲和力和相似度, 并开始对两个种群进行如下相同操作。

(4) 细胞克隆。根据亲和力大小对抗体进行排序, 以选择概率在 A_n 选择 N_0 个抗体构成克隆群体 B_n , 利用式 (14)、(15) 繁殖 M 个克隆, 形成新的抗体群 C_n 。

(5) 亲和力突变, 克隆交叉。本算法中的突变原则与传统的突变不同, 对 C_n 中的个体分别进行部分基因突变和全变异, 部分基因突变为任意产生两个变异位, 对变异位之间的基因任意改变其排列方式。全变异方式是抗体的所有基因位依次向左移一位, 将子代亲和力优于父代的抗体保留。然后从保留下来的优秀子抗体中随机选择两个抗体进行单位置次

序交叉操作,即随机产生一个交叉位置,保留 P_1 个体交叉位置前的基因片段,并在 P_2 个体中删除 P_1 中保留的基因,进而将剩余的基因片段填入交叉位置后,从而得到子代个体。

(6)克隆抑制,指在抗体群中依据抗体的亲和力和相似度抑制部分抗体。计算 C_n 中各抗体的相似度,将浓度高于阈值的抗体个体抑制去掉,并将 C_n 与 B_n 结合形成 D_n 。

(7)免疫选择。利用式(13)计算 D_n 中的各抗体的激励度,并以某种概率从 D_n 中选出激励度最高且不同的 N 个抗体,构成新的抗体群 E_n 。

(8)募集新成员。随机产生个数为 S 的新抗体群,利用式(9)计算其亲和力,将 E_n 中亲和力较低的一部分个体,用 S 中亲和力较高的个体替代,同时求得当前种群最优解。

(9)协同竞争。对种群1中的每个个体 p_{ij} 和种群2中的个体进行种群间竞争,按式(15)、(16)将竞争结果反馈给个体,利用式(18)当前亲和力值进行修正,同时交换两种群的最佳个体。

(10)判断是否满足终止条件,如果满足,则结束运算,否则转到(4)。

4 仿真研究

在仿真实验中,以10个加工产品、5个处理单元的调度问题为例,表1是产品的处理时间。采用第3节中介绍的协同免疫克隆算法(CICA)对零等待flowshop问题进行仿真。

表1 产品处理时间

处理单元	加工产品									
	1	2	3	4	5	6	7	8	9	10
1	25	17	41	74	37	72	11	31	32	27
2	15	41	155	12	95	34	77	39	92	114
3	12	22	83	24	72	62	31	141	42	21
4	40	36	121	48	52	32	26	56	74	90
5	60	58	160	78	153	162	32	79	102	52

算法中初始解群体的规模 $Popsiz$ = 40,克隆规模 N_c = 80,最大的迭代次数 $MaxGen$ = 80,激励度系数 β = 10(经参数讨论确定),反馈因子 r = 1(经参数讨论确定),交叉率为0.6,抗体部分基因变异率为0.3,抗体基因全变异率为0.05,抗体浓度抑制阈值和克隆抑制阈值都取为0.7。对问题进行了多次仿真,得到的结果如图1、图2所示。图1是解群体

的最优值收敛曲线和平均目标值收敛曲线,图2是调度问题的甘特图。从图1中可以看出,随着算法的不断演化,目标函数值和平均目标值不断下降,趋于稳定,说明了算法的收敛性,由此可以确定产品的加工顺序为1-2-4-6-9-5-8-3-10-7。从图2可以看出,每个工件都满足零等待的约束。

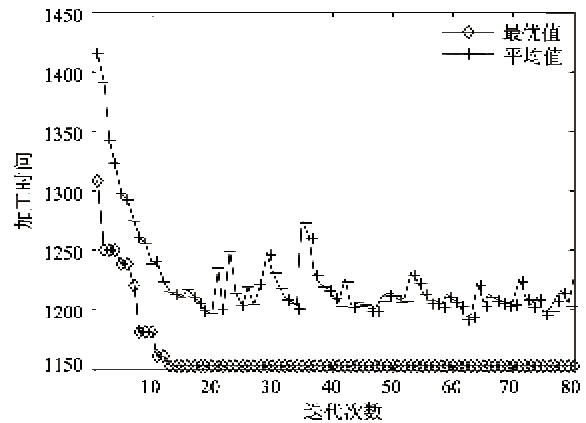


图1 最优值演化曲线

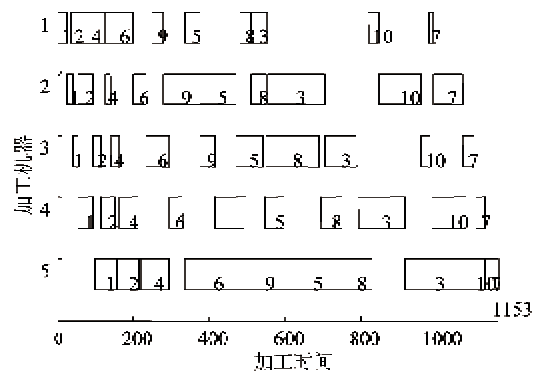


图2 最优解的调度问题甘特图

为了进一步验证算法CICA的优越性,以上述调度问题为例,在参数设置不变的条件下,与免疫克隆算法(ICA)和文献[8]中的免疫算法(IA)重复运行100次的结果进行比较,结果如表2所示,文献[8]中的算法已经与改进的遗传算法进行了比较,体现了该算法的优越性。

定义:寻优率 P = 寻到最优解次数/总寻优次数。

平均目标值误差 MP = (平均目标值 - 最小目标值)/最小目标值;

最差目标值误差 RP = (最差目标值 - 最小目标值)/最小目标值。

表2给出了调度问题运算结果。从表中可以看出,在参数设置相同的情况下,CICA的寻优效果明

显优于免疫算法和免疫克隆算法, CICA 得到的 *makespan* 的平均值, 寻到的最差解均远好于免疫算法和免疫克隆算法求得的 *makespan* 的平均值和寻到的最差解。且在收敛速度上也有明显的改善, CICA 基本在 50 代之内都可以寻到最优解, 但是免疫

算法的最大迭代次数为 150。这主要是 CICA 引入了克隆机制和协同竞争的机制, 实现了局部并行搜索和种群间信息的交流, 使得抗体群具有更好的多样性和更好的全局寻优能力, 加快了收敛速度。

表 2 调度问题运算结果

算法名称	目标函数	寻优率 <i>P</i>	平均目标值偏差 <i>MP</i> (%)	最差目标值偏差 <i>RP</i> (%)
	最优值/平均值/最差值			
IA(10 次) ^[8]	1154/1153.3/1153	0.7	0.026	0.087
ICA(10 次)	1156/1153.3/1153	0.9	0.026	0.26
CICA(10 次)	1153/1153/1153	1	0	0
IA(100 次) ^[8]	1196/1155.3/1153	0.52	0.199	3.729
ICA(100 次)	1157/1153.4/1153	0.88	0.035	0.347
CICA(100 次)	1156/1153.1/1153	0.95	0.0086	0.26

讨论激励度系数选取对算法性能的影响, 在其他参数不变的情况下仿真 10 次。

表 3 给出了 β 取值不同时的仿真结果。从中可以看出, 当 $\beta = 1$ 时, 对抗体的相似度没有反比例的抑制作用, 抗体种群的多样性不能保证, 使得算法容易陷入局部最优。但是当 β 取值过大时(如 $\beta = 100$), 此时对相似度的抑制过大, 使得某些相似度比较高而并不完全一样的优秀抗体过早被抑制, 收敛速度过快, 使得陷入了某些局部最优点。

讨论反馈因子选取对算法性能的影响, 在其他参数不变的情况下仿真 10 次。

表 3 β 取值不同时的调度问题的仿真运算结果

激励度系数	目标函数	寻优率 <i>P</i>
	最优值/最差值	
$\beta = 1$	1153/1156	0.5
$\beta = 5$	1153/1156	0.8
$\beta = 10$	1153/1153	1
$\beta = 15$	1153/1153	1
$\beta = 20$	1153/1153	1
$\beta = 100$	1153/1156	0.6

表 4 给出了 r 取值不同时的仿真运算结果。从中可以看出, 当 $r = 0$ 时, 反馈为 0, 此时竞争的效果不存在, 但是由于 CICA 引入了精英迁移的策略, 使得两种群之间仍有信息的交流, 所以此时算法仍优于免疫算法和免疫克隆算法。但是当反馈因子取值稍大时, 此时反馈比较激烈, 竞争效果比较明显, 此算法类似于完全竞争型的 CompEA, 收敛速度过快, 使得产生的解不够优秀, 容易陷入局部最优。

表 4 r 取值不同时的调度问题的仿真运算结果

反馈因子	目标函数	寻优率 <i>P</i>
	最优值/最差值	
$r = 0$	1153/1153	1
$r = 0.5$	1153/1153	1
$r = 1$	1153/1153	1
$r = 3$	1153/1157	0.6
$r = 5$	1153/1156	0.5

5 结 论

本文针对流程工业中零等待的调度问题, 提出一种新的解决零等待生产调度问题的协同克隆免疫算法(CICA), 通过仿真讨论了激励度系数和反馈因子的选取并验证了算法的优越性和有效性。激励度系数不仅和抗体的亲和力成正比, 且和抗体的相似度成反比, 对相似抗体具有一定的抑制作用, 以此来保证种群的多样性。激励度系数不能选取过大, 当抑制作用过于激烈时, 可能会将一部分优秀抗体过早抑制, 使得算法陷入局部最优。反馈因子是为了调节种群间的竞争激烈度, 以此将具有更高的竞争力的个体更多地被克隆, 加快算法的收敛速度, 但是仿真结果表明, 当反馈因子较大时, 竞争效果被放大, 此时算法效果并不理想。综上所述, 在选取了适当的参数时, CICA 算法具有更好的性能, 并能够更好地应用于实际的生产调度问题中。当然, 怎样将此算法用于解决更加复杂的生产调度问题, 也是今后需要继续研究的方向。

参考文献

- [1] Gamay M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness M. San Francisco: W. H. Freeman and Company, 1979
- [2] De Castro L N, Von Zuben F J. Learning and optimization using the colonial selection principle. *IEEE Transaction on Evolutionary Computation*, 2002, 6(3): 239-251
- [3] Burnet F M. The Colonial Selection Theory of Acquired Immunity. Cambridge: Cambridge University Press, 1959
- [4] Wierzbicki S. Function optimization by the immune metaphor. *Task Quarterly*, 2002, 6(30): 1-16
- [5] De Castro L N, Timmins J. An artificial immune network for multimodal function optimization. In: Proceedings of the 2002 World on Congress on Computational Intelligence, Honolulu, USA, 2002, 1: 699-704
- [6] 刘丽珏, 蔡自兴. 变异协同进化的免疫克隆算法. 控制与决策, 2008, 23(6): 613-618
- [7] Hart E, Ross P, Nelson J. Producing robust schedules via an artificial immune system. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA, 1998. 464-469
- [8] 徐震浩, 顾幸生. 具有零等待的 flow shop 问题的免疫调度算法. 化工自动化及仪表, 2005, 32(1): 10-12
- [9] 王祎, 陈为栋, 顾幸生等. 基于内分泌激素调节机制的免疫算法的 flowshop 调度问题. 系统仿真学报, 2008, 20(13): 3425-3430
- [10] 刘晓冰, 吕强. 免疫克隆算法求解柔性生产调度问题. 控制与决策, 2008, 23(7): 781-785
- [11] 宋建成. 间歇过程计算机集成过程系统. 北京: 化学工业出版社, 1999
- [12] 张著洪, 黄席樾. 多目标约束优化免疫算法研究及其应用. 模式识别与人工智能, 2003, 16(4): 453-458
- [13] 张新征. 结合竞争与合作的新型协同进化算法: [硕士学位论文]. 合肥: 中国科学技术大学计算机科学技术系, 2005. 16-20

The co-evolutionary immune clone algorithm and its application in zero-wait flowshop scheduling

Jin Feihui, Gu Xingsheng

(Research Institute of Automation, East China University of Science and Technology, Shanghai 200237)

Abstract

For solving the zero-wait flowshop problem, the paper proposes an effective approach called co-evolutionary immune clone algorithm (CICA). The algorithm combines the colonial selection mechanism with the principle of immune system, and adds in a new operational of affinity mutation and a new function of activity in local operation, to make the capacity of antibodies to antigens not only relate with their affinity but also concern with their concentration, consequently improving the diversity of the antibodies and avoiding their prematurity. In collective operation, a thinking of co-evolutionary and the elite migration process in evolution are used to accelerate the convergence and achieve the purpose of optimization. The simulation result demonstrates the search precision of the CICA is more effective and highly advantageous than that of the immune algorithm and the immune colonial algorithm, thus verifies its validation and excellence. Finally, the paper discusses the influence of the incentive coefficient and the feedback coefficient on the algorithm's performance by simulation.

Key words: scheduling, colonial selection, immune algorithm, co-evolutionary, zero wait