

抗体修正免疫算法对高维 0/1 背包问题的应用^{*}

庄中文, 钱淑渠

(安顺学院 数学与计算机科学系, 贵州 安顺 561000)

摘 要: 遗传算法极难处理高维约束优化问题, 故借鉴免疫系统机理, 提出一种抗体修正免疫算法解决一类高维约束优化问题。该算法设计的关键在于抗体亲和力由抗体浓度及群体状态决定; 可行抗体被克隆、突变; 非可行抗体的基因按价值密度由小到大逐一修正。选取两种已有的智能算法(ETGA、ISGA), 通过不同约束条件下的高维 0/1 背包问题的仿真比较。结果表明, 该算法较其他算法能更快地跟踪最优值, 具有较强的勘测和开采能力。

关键词: 高维 0/1 背包问题; 约束优化; 抗体修正; 免疫算法

中图分类号: TP10 **文献标志码:** A **文章编号:** 1001-3695(2009)08-2921-03

doi:10.3969/j.issn.1001-3695.2009.08.035

Immune algorithm with antibody-repaired and its application for high-dimensional 0/1 knapsack problem

ZHUANG Zhong-wen, QIAN Shu-qu

(Dept. of Mathematics & Computer Science, Anshun College, Anshun Guizhou 561000, China)

Abstract: It's difficulty to deal with high-dimensional optimization problem constrained for GA, this paper proposed an immune algorithm with antibody-repaired, based on biological immune system's functions, to solve a class of high-dimensional optimization problem constrained. The key of algorithm is: the affinity of antibodies had relation to the antibody's density and current population, the feasible antibodies were cloned and mutated, repaired the infeasible antibodies by means of the increasing sorting of price consistency of antibodies gene. In numerical experiments, selected two existing intelligent algorithms (ETGA, ISGA) to compare with the designed algorithm, tested high-dimensional 0/1 knapsack problems with different constraints. The results indicate that the new algorithm can track rapidly the optimum, and also show the predominant exploitation and exploration capability of algorithm.

Key words: high-dimensional 0/1 knapsack problem; constrained optimization; antibody repair; immune algorithms

0 引言

近来, 背包问题已吸引了广大学者对其理论和实际应用价值的深入研究。在理论上, 尽管背包问题的结构简单, 但它是一类非常经典的 NP 完全问题; 在实际应用中, 许多工业问题均可建模为背包问题, 如资源优化、货舱装载、投资组合等, 都是典型的应用实例。由于该问题的解空间规模同问题输入规模呈指数关系(计算复杂度为 $O(2^n)$)^[1], 动态规划、回溯法、分支定界法等确定性算法不适合求解高维的 0/1 背包问题^[2], 而智能优化算法作为一种高级优化算法, 在求解 0/1 背包问题上得到了广泛的应用。文献[3]引入量子计算思想, 提出了量子遗传算法, 取得较为满意的效果。文献[1]提出主动进化算法用于背包问题的求解。Simões 等人^[4]通过比较大量改进的遗传算法解决 0/1 背包问题的性能, 于 2003 年提出一种新的遗传算法(ETGA)^[5], 该算法引入一种 transformation 算子替换遗传算法中的交叉算子, 同时删除了变异算子。同年, Simões 等人^[6]又将免疫系统机理引入遗传算法, 获得一种免疫系统遗传算法(ISGA)用于动态 0/1 背包问题的求解, 该应用证实了引入免疫系统机理能增强算法的开采能力, 实验效果较好,

但应用的测试实例仅是 17 个物件的 0/1 背包问题, 没有进一步应用于高维的背包问题。

实际上, 背包问题是指从多种物品(一般称为项目)中选择几件物品, 装满背包, 要在不超过背包承重量的前提下, 使装入背包的价值最大。解决背包问题的关键在于约束条件的处理, 目前已有的算法一般采用罚函数的方式处理约束条件, 这种方法当问题规模较大时, 尽管可行, 但搜索效率较低, 甚至有时得到的结果比一般的启发式算法还要差。笔者在此方面也作了部分相关的研究^[7,8]。鉴于此, 本文借鉴生物免疫系统机理, 大量克隆所获得的极少可行抗体, 利用突变算子进行邻域搜索, 对不可行抗体采取修正措施, 进而获得大量可行抗体, 抗体的亲和力不仅由目标值决定, 同时由抗体间的浓度确定, 亲和力较大的抗体被选择参与进化。本文提出了一种抗体修正免疫算法(AIOA)求解大规模背包问题, 算法结构简单, 通过不同约束下的背包问题对算法进行性能比较, 结果表明所获算法具有较快的搜索性能和较强的实用价值。

1 背包问题数学模型

背包问题的数学模型实质是一类 0-1 整数规划问题。现

收稿日期: 2008-11-19; 修回日期: 2008-12-23 基金项目: 安顺学院青年一般项目基金资助(20080514)

作者简介: 庄中文(1962-), 男, 贵州安顺人, 副教授, 硕士研究生, 主要研究方向为最优化方法(shuquqian@163.com); 钱淑渠(1978-), 男, 安徽枞阳人, 讲师, 硕士, 主要研究方向为智能优化算法、系统建模与仿真。

有 k 个物件,令第 i 个物件的重量和价值分别记为 w_i 和 p_i ($i = 1, 2, \dots, k$),背包的最大容纳重量记为 $C = \alpha \sum_{i=1}^k w_i$ ($\alpha < 1$),当物件 i 被装入背包时,定义变量 $x_i = 1$;否则 $x_i = 0$ 。现在考虑 k 个物件的选择与否,则背包内所选物件总重量为 $\sum_{i=1}^k x_i w_i$,物件的总价值为 $\sum_{i=1}^k p_i x_i$,即如何决定变量 x_i ($i = 1, 2, \dots, k$) 的值(即确定一个物件组合),使背包内物件总价值为最大。故其数学模型可描述为

$$\begin{aligned} \max f(X) &= \sum_{i=1}^k p_i x_i \\ \text{s. t. } \sum_{i=1}^k x_i w_i &< C; x_i \in \{0, 1\}; i = 1, 2, \dots, k \end{aligned}$$

2 算法设计及步骤

设抗体 $x \in A$ 为抗体群 A 中的任一抗体,其亲和力设计为 $\text{aff}(x) = 1.0 / e^{d(x)} + e^{[f(x) - \min(A)] / [\max(A) - \min(A) + \varepsilon]}$ 其中: $d(x)$ 为抗体 x 的浓度; $f(x)$ 为抗体 x 的目标值; $\max(A)$ 、 $\min(A)$ 分别为抗体群 A 中抗体的最大、最小目标值; ε 为一个任意小的正数(ε 是为防止抗体群中所有抗体都相同而使分母为 0 现象)。此设计表明抗体 x 的亲和力既由其浓度决定又与当前进化群体 A 的分布状况有关。在此亲和力的设计下,算法的流程图如图 1 所示。其中: n 为算法的迭代数; G 为算法的最大迭代数;群体规模设置为 N 。

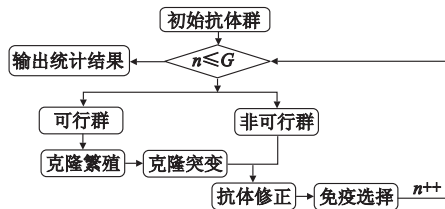


图1 AIOA流程图

2.1 算子的设计

2.1.1 抗体编码

对每个抗体施行 0、1 编码,抗体 x 定义为 $x = (x_1 \ x_2 \ \dots \ x_k)$ 。其中 $x_i = 0$ 或 1 , $x_i = 1$ 表明第 i 个物件被装入背包,否则表明其不被装入背包。

2.1.2 克隆繁殖

克隆繁殖算子只作用于可行的抗体。每个抗体克隆数目为 4,则克隆抗体群的最大规模为 $4N$ 。

2.1.3 抗体修正^[9]

抗体修正是使不可行抗体修正为可行抗体,在算法中起到非常关键作用。假设抗体 $x = (x_1 x_2 \dots x_k)$ 为非可行抗体,其具体修正步骤如下:

a) 设 $\Omega = \{i | x_i = 1, i = 1, 2, \dots, k\}$,计算基因 x_i 的价值密度 $\rho_i = \rho_i / w_i$ ($i \in \Omega$),若 $\rho_m = \min_{i \in \Omega} \{\rho_i\}$,则令 $x_m = 0$,获第一次修正的抗体 $x = (x_1 x_2 \dots x_k)$ 。

b) 判断 $x = (x_1 x_2 \dots x_k)$ 是否可行,若可行,则结束;否则,转入步骤 a)。

2.1.4 抗体浓度

设抗体 $x = (x_1 x_2 \dots x_k)$,抗体 $y = (y_1 y_2 \dots y_k)$,则抗体 x 的浓度为

$$d(x) = | \{ x | s(x, y) > r \} | / N$$

其中: $s(x, y) = | \{ x_i | x_i = y_i \} | / k$; r 为浓度半径。

2.2 算法步骤

根据图 1,算法步骤可描述如下:

- a) 随机产生初始群体 A_1 及 k 物件的价值及重量,并设置参数 $n = 1$ 。
- b) 判断 $n \leq G$? 若是,执行群体分离,获可行群 B 及非可行群 C ; 否则,输出统计结果。
- c) 克隆繁殖算子作用于 B ,获克隆群 D 。
- d) 对 D 施行克隆突变,获突变群 E 。
- e) 修正算子分别作用于 C 和 E 获可行群 C' 和 E' ,并将其组合后删除相同抗体,获群体 F 。
- f) 计算 F 中所有抗体的浓度及亲和力,并根据亲和力选择较好的 N 个抗体,获新抗体群 H 。
- g) 置 $n \leftarrow n + 1, A_n \leftarrow H$,转入步骤 b)。

该算法结构简单,关键是亲和力的设计既兼顾抗体本身又兼顾当前群体性能,因此增强了被选取的抗体多样性;同时免疫算子仅作用于可行的抗体,加速算法的收敛速度,增强算法的开采和探测能力;对于非可行的抗体进行修复,大大提高了算法的执行效率。

3 数值实验

实验中,选取物件数目分别为 100、200 为测试例子,分别记为问题 P1 和问题 P2。 p_i, w_i 均为 1 ~ 100 中随机产生的整数,对于问题 P1 权重约束中 α 分别选取 0.2、0.5、0.8; 对于问题 P2 权重约束中 α 分别选取 0.3、0.6、0.9。

选取参与比较的两种算法为 ETGA^[5]、ISGA^[6]。为减少随机性对算法性能的影响,各算法对每一测试问题独立执行 $M = 30$ 次,每次独立执行的最大迭代数为 $G = 200$,以便分析它们的统计特征。在实验中,对同一测试问题,参与比较的三种算法的适应度计算公式均相同。对于 ETGA 和 ISGA 的参数值均选自相应文献。

算法 ETGA 和 ISGA 群体规模均为 100,而 AIOA 的群体规模为 25,此设计是为了使三算法具有相同的评价次数。ETGA 的转换率为 90%,取代率为 40%,基因段长度为 5,基因池个数为 4; ISGA 的转换率为 90%,选择率为 50%,每个抗体的克隆数目为 2,取代率为 40%,基因段长度为 4,基因池个数为 8,记忆池大小为 5。

对于 AIOA,突变概率为 $1.0/k$, k 为染色体长度,浓度半径 $r = 0.5$,每个可行抗体克隆数目为 4。

性能比较标准为算法在 M 次独立执行中所获的最好适应度值(max)和 30 次独立执行所获群体的平均适应度值(Aver)、Acc^[6]、Ada^[6]、运行时间的平均值及所获可行最优解的次数。其中:

$$\begin{aligned} \text{Acc} &= 1/M \sum_{i=1}^M [1/G \sum_{n=1}^G ((\text{optimum} - \text{best}_{in}) / (\max \text{fit}_{in} - \min \text{fit}_{in}))] \\ \text{Ada} &= 1/M \sum_{i=1}^M ((\text{optimum} - \text{best}_{iG}) / (\max \text{fit}_{iG} - \min \text{fit}_{iG})) \end{aligned}$$

其中:optimum 表示理论最优值,best_{in} 为算法在第 i 次执行中第 n 代所获的最好适应度值;max fit_{in}、min fit_{in} 分别表示在第 i 次执行中第 n 代群体的最好可行抗体和最差可行抗体的适应度值。由此可知,Acc、Ada 的值越小,表明算法所获最好值越接近理论最优值;相反则表明算法收敛性差。特别地,若算法不能获得可行的解,则可能导致这两个值出现无穷大(∞)现象。

问题 P1 随机产生的物件价格和重量值(表 1),总的重量 $W = 4998$; 问题 P2 物件价格和重量值(表 2),总重量为 $W =$

9 594。表 3 是统计值的比较,其中 optimum 是在当前约束下问题的最优值。

表 1 问题 P1 的 100 个物件的价格及其重量值

	94	60	88	18	39	57	4	74	86	77	59	45	74	99	46	68	99	83	23	85
	80	41	58	11	35	73	100	2	79	58	70	40	6	9	26	2	7	92	40	45
p_i	65	50	80	53	37	84	14	14	72	41	46	76	13	78	77	77	49	29	63	61
	32	2	6	47	31	46	35	85	39	64	52	24	25	26	50	81	89	61	44	95
	40	27	83	81	85	32	60	91	44	54	31	48	50	94	32	83	24	40	43	11
	94	81	8	62	21	83	85	45	48	81	4	64	59	97	96	14	21	59	34	68
	36	2	3	65	26	48	25	17	11	97	43	23	24	48	56	73	54	15	98	99
w_i	47	93	78	68	24	52	8	89	100	7	9	46	8	40	77	46	76	78	7	32
	92	11	93	75	6	60	64	15	99	30	99	61	17	3	31	34	76	68	79	91
	95	25	73	43	89	9	12	31	71	24	19	70	76	14	50	85	40	78	12	6

表 2 问题 P2 的 200 个物件的价格及其重量值

	68	4	33	95	74	1	63	23	15	29	90	20	74	40	4	82	69	93	26	12
	1	88	72	25	58	7	53	43	52	85	68	24	54	35	13	67	54	69	35	84
	62	85	94	87	78	17	84	91	62	79	47	58	4	68	17	35	61	78	54	14
	61	23	38	83	24	14	81	70	12	20	35	58	51	25	91	33	25	100	17	70
p_i	35	56	99	96	18	5	53	58	22	54	41	47	44	100	70	73	6	87	15	80
	75	68	14	44	11	70	49	35	36	17	5	68	36	72	97	65	12	34	30	14
	6	67	100	66	88	55	47	77	69	36	36	34	95	76	13	19	73	37	44	99
	57	48	38	4	58	7	98	30	56	75	42	30	20	17	92	18	34	4	35	55
	96	35	99	24	57	87	56	63	21	41	83	85	34	53	62	43	40	37	90	42
	81	31	46	74	85	4	77	43	28	20	84	72	79	34	29	65	4	95	55	30
	80	29	84	77	39	8	35	52	63	74	69	82	12	52	78	24	32	93	23	70
	5	17	71	4	59	40	23	98	8	7	10	19	85	51	61	49	71	74	10	44
	52	24	21	95	63	50	100	73	29	62	69	78	30	55	93	28	20	30	83	29
	11	50	72	96	20	18	67	68	26	21	86	3	20	38	71	5	93	42	12	44
w_i	52	24	1	41	53	69	42	3	15	40	94	14	48	63	47	42	78	42	58	56
	24	74	57	52	12	48	99	63	67	23	60	67	46	24	13	29	72	5	39	54
	92	51	43	98	36	67	57	40	29	35	61	94	86	74	61	23	23	30	17	34
	59	97	55	38	62	72	10	10	98	82	93	84	32	65	100	5	18	36	15	15
	92	73	74	85	55	71	27	57	27	22	34	11	68	58	1	2	20	94	4	37
	40	30	77	4	66	22	75	95	4	88	16	98	24	96	91	57	7	7	71	2

表 3 三算法在不同的约束下独立执行30次所获最好适应度值、群体平均适应度值、Acc及Ada值的平均统计值比较

算法	$\alpha=0.20$, Optimum=2 656				$\alpha=0.50$, optimum=4 142				$\alpha=0.80$, optimum=4 985			
	Max	Aver	Acc	Ada	Max	Aver	Acc	Ada	Max	Aver	Acc	Ada
ETGA	2 532	2 148	∞	0.152	3 881	3 388	0.331	0.150	4 915	4 460	0.161	0.025
P1 ISGA	2 577	2 245	∞	0.156	3 979	3 751	∞	0.367	4 964	4 901	0.811	0.114
AIOA	2 656	2 641	0.062	0.004	4 142	4 009	0.066	0.000	4 985	4 779	0.000	0.000
算法	$\alpha=0.30$, optimum=6 298				$\alpha=0.60$, optimum=8 800				$\alpha=0.90$, optimum=9 949			
	Max	Aver	Acc	Ada	Max	Aver	Acc	Ada	Max	Aver	Acc	Ada
ETGA	5 719	6 014	∞	∞	8 326	7 937	0.415	0.130	9 870	9 361	0.337	0.015
P2 ISGA	6 149	6 205	∞	∞	8 475	8 419	∞	0.898	9 887	∞	1.386	∞
AIOA	6 298	6 176	0.166	0.003	8 796	8 484	0.103	0.001	9 940	9 531	0.115	0.001

由表 3 获知,对于问题 P1:在不同的 α 下,AIOA 所获的最好值比其他算法都大,且在给定的代数内都搜索到了最优值,所获平均适应度值当 $\alpha=0.8$ 比 ISGA 稍小,其余都比其他算法大;由 Acc 值比较获知,ISGA 和 ETGA 当 $\alpha=0.2$ 都为 ∞ ,是由于两算法至少在某一代或某一次独立执行中没有找到可行的解。同样由 Ada 值获知,ISGA 当 $\alpha=0.5$ 时为 ∞ ,是由于该算法在 30 次独立执行中至少有一次没有搜索到可行解;而由 AIOA 所获 Acc 和 Ada 值可知,其所获值都比较接近于 0,当 $\alpha=0.2$ 时,效果稍差;当 $\alpha=0.8$ 时,Acc 和 Ada 的值都为 0,表明该算法在每一代或每一次独立执行中都能搜索到最优值。由整体分析可知,当 α 较小时,参与比较的算法搜索到最优值相对困难,而当 α 较大时算法较易搜索到最优值。由表 4 算法运行的平均时间比较获知,AIOA 相对其他算法运行的时间稍长。同样对于问题 P2,当 $\alpha=0.3$ 时 ISGA 和 ETGA 所获效果很差,能搜索到可行解非常困难,致使 Ada 和 Acc 值出现 ∞ ,当 $\alpha=0.9$ 时,ISGA 中的 max 出现比 AIOA 大,这是由于 ISGA 所获解是不可行的解;同时由问题 P1 和 P2 所获结果比较获知,

随着物件数目的增多,参与比较的算法的搜索更加困难,而对 AIOA 性能影响较小。

由表 5 可知,算法 AIOA 每一次执行都能获得可行的最优解,而其他算法所获可行的最优解次数较少,当 α 较大(约束较弱)时稍好。图 2~4 是物件数为 100 时三算法独立执行 30 次所获最好抗体的适应度平均变化曲线。由该曲线可知,当搜索到 60 代左右时,AIOA 便能跟踪到最优曲线,而其他算法在 200 代以内都未能达到最优曲线,其中 ISGA 较 ETGA 稍好,可知其他算法搜索速度较慢。图 5~7 是物件数为 200 时三算法独立执行 30 次所获最好抗体的适应度平均变化曲线,由此可知 AIOA 搜索到 100 代左右便能接近最优曲线,而其他算法极难跟踪最优曲线,其中 ETGA 跟踪较慢。但在图 7 中出现了三算法平均搜索效果较相近,这是由于当时问题的约束较弱,使得问题的求解容易,很难显示出算法的优越性。总体分析可知,在约束较强(较小)的条件下,两种参与比较的算法易陷入局部搜索现象(图 2~6),ETGA 和 ISGA 搜索进程非常慢,而 AIOA 跟踪最优曲线速度较快。

表 4 三算法在不同约束下独立执行30次所获平均时间(s)比较

算法	$\alpha=0.20$	$\alpha=0.50$	$\alpha=0.80$
P1 ETGA	1.675	1.748	1.625
ISGA	1.624	1.674	1.447
AIOA	2.189	2.988	2.096
算法	$\alpha=0.30$	$\alpha=0.60$	$\alpha=0.90$
P2 ETGA	2.980	3.959	2.903
ISGA	2.761	2.944	2.627
AIOA	2.966	2.152	2.959

表 5 三算法在30次独立执行中获得可行最优解的总次数比较

算法	$\alpha=0.20$	$\alpha=0.50$	$\alpha=0.80$
P1 ETGA	6	15	19
ISGA	7	12	16
AIOA	30	30	30
算法	$\alpha=0.30$	$\alpha=0.60$	$\alpha=0.90$
P2 ETGA	4	12	25
ISGA	8	16	23
AIOA	30	30	30

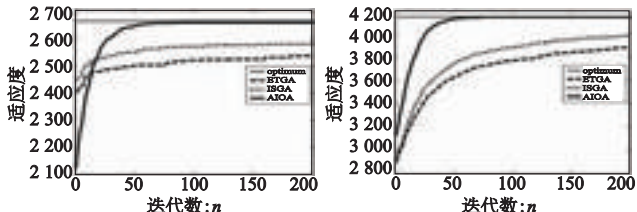


图2 问题P1: $\alpha=0.20$ 时各算法独立执行30次所获平均适应度搜索曲线比较

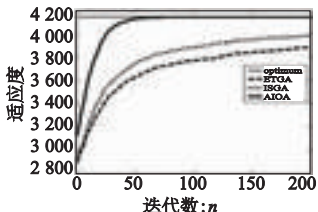


图3 问题P1: $\alpha=0.50$ 时各算法独立执行30次所获平均适应度搜索曲线比较

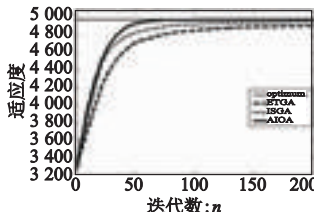


图4 问题P1: $\alpha=0.80$ 时各算法独立执行30次所获平均适应度搜索曲线比较

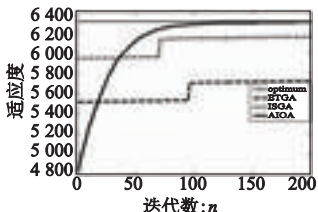


图5 问题P2: $\alpha=0.30$ 时各算法独立执行30次所获平均适应度搜索曲线比较

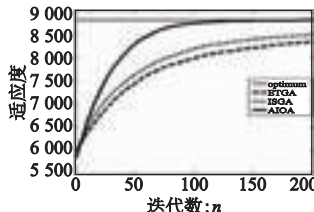


图6 问题P2: $\alpha=0.60$ 时各算法独立执行30次所获平均适应度搜索曲线比较

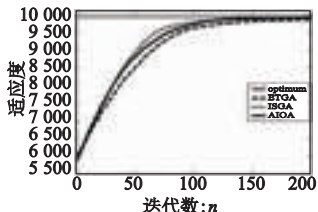


图7 问题P2: $\alpha=0.90$ 时各算法独立执行30次所获平均适应度搜索曲线比较

4 结束语

本文借助生物免疫系统的学习、抗体克隆、基因突变等功能,设计了一种抗体修正免疫算法,并用于高维 0/1 背包问题;在不同的约束条件下比较所设计算法和其他 (下转第 2930 页)

作为父染色体,对每对父染色体进行一次交叉操作,但搜索速度慢,得到的最优效果不是很好。本文每次进行遗传算法进化时都是在确保种群多样性的前提下进行的,加大进化强度,提出三个父辈个体交叉,生成六个子个体,因此保证遗传算法能够快速进化。

5 实验测试

本文采用随机产生初始种群的方法,优化目标是最短加工完成时间,采用最大进化代数作为停止准则。对于上述设计的算法应用 Fisher 等人提出的 FT 问题和 Lawrence 提出的 LA 几个典型问题进行测试,采用 MATLAB 编程进行了实现,程序运行在 P4 1.8 GHz CPU,内存 512 MB 的计算机上。其种群规模为 50、迭代次数为 200 或 400,交叉概率 0.6,变异概率 0.2,疫苗库疫苗个数为 $50 \times 10\% \times 6 = 30$;阈值取值 0.9;抗体注射疫苗率为 0.3。将本文算法的优化指标与 C. A. Coello 等人的人工免疫算法 AIS^[7]及传统遗传算法 SGA 作比较,统计结果如表 1 所示。

表 1 SGA、AIS 与 GAI 的比较

problem	C **	SGA			AIS			GAI		
		C **	B	T	C **	B	T	C **	B	T
FT6 × 6	55	55	0	5.1	55	0	53.2	55	0	6.25
FT10 × 10	930	997	8.1	213	941	1.2	410	930	1.4	223
LA10 × 10	945	979	6.3	149	945	0	722	945	0	277
LA15 × 10	977	996	7.2	177	1122	4.6	903	981	2.1	385
LA20 × 10	1216	1428	5	301	1217	5	1162	1216	1.8	543

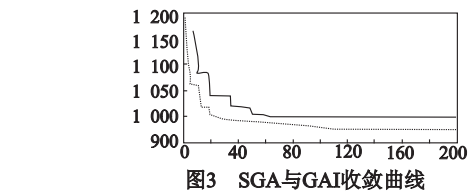
注:C ** 为最优值;B 为标准差;T 为平均时间。

对于较简单的 FT6 × 6 问题,本文算法都在十几代以内收敛到最优解,而对于难度较大的问题,本文算法也能得到最优解。从表 2 可以看出,本文算法 GAI 所得最优解比 SGA、AIS 要好,算法所用时间明显比 AIS 少,而且标准差小于 5,说明 GAI 算法不但有较强的搜索能力,而且有较强的鲁棒性。图 3 为算法 FT10 × 10 的收敛曲线,实线为 SGA 收敛曲线图,虚线为 GAI 收敛曲线图。从图中可看出,本文设计的算法有较快的收敛速度。通过以上实验结果不难看出,本文算法有更强的搜索能力。

(上接第 2923 页)算法的跟踪速度及搜索效果,实验结果论证了该算法处理高维 0/1 背包问题具有较明显的优越性,其能快速定位最优解的位置,具有较好的搜索效果和效率。另外,本文仅探讨了解决一类特殊的约束规划问题,其他情形的研究工作将继续作报道。关于该算法的广泛应用和理论研究,需进一步开展。

参考文献:

[1] 史亮,董槐林,王备战,等. 求解大规模 0/1 背包问题的主动进化遗传算法[J]. 计算机工程, 2007,33(13):31-33.
[2] 陈国良,王煦法,庄镇泉,等. 遗传算法及其应用[M]. 北京:人民邮电出版社,1996.
[3] HAN K H, KIM J H. Genetic quantum algorithm and its application to combinatorial optimization problem[C]//Proc of IEEE Congress on Evolutionary Computation. 2000:1354-1360.
[4] SIMÕES A, COSTA E. Using genetic algorithms to deal with dynamic environments: a comparative study of several approaches based on



6 结束语

本文针对 job-shop 调度问题,设计了一种有效的算法,三个个体交叉的遗传算法与改进的人工免疫算法相结合,既加快了进化速度,又预防了局部收敛,通过对五个典型问题的仿真结果表明,本文算法无论是在最优解的获取能力、还是在实验结果的平均相对偏差以及收敛速度等方面,都明显优于传统的遗传算法和人工免疫算法,具有较强的实用性。

参考文献:

[1] LENSTRA J K, RINNOOY K A H G, BRUCKER P. Complexity of machine scheduling problem[J]. Ann Discr Math, 1997, 9(1): 343-362.
[2] JAIN A S, MEERAN S. Deterministic job-shop scheduling past, present and future[J]. European Journal of Research, 1999, 113(2):390-434.
[3] GONCALVES J F, De MAGALHAES MENDES, RESENDE M G C, et al. A hybrid genetic algorithm for the job-shop scheduling problem[J]. European J of Operational Research, 2005,167(1): 77-95.
[4] 廖章珍,陈强. 人工免疫系统的基本理论及其应用[J]. 自动化与仪器仪表, 2008(1):5-15.
[5] 张超勇,饶运清,李培根,等. 求解作业车间调度问题的一种改进遗传算法[J]. 计算机集成制造系统,2004, 10(8): 966-970.
[6] CHEN S, CHANG E S, ALKADHIMI K. Regularized orthogonal least squares algorithm for constructing radial basis function networks[J]. Int J Control, 1996,64(5):829-837.
[7] COELLO C A, CORTES R D, CRUI C N. Use of an artificial immune system for job-shop scheduling[M]. Berlin Heidelberg: Springer-Verlag, 2003.
[8] promoting diversity[C]//Proc of Genetic and Evolutionary Computation (GECCO' 02). New York: Morgan Kaufmann Press,2002: 9-13.
[5] SIMÕES A, COSTA E. Improving the genetic algorithm's performance when using transformation[C]//Proc of the 6th International Conference on Neural Networks and Genetic Algorithms (ICANNGA '03). 2003:175-181.
[6] SIMÕES A, COSTA E. An immune system-based genetic algorithm to deal with dynamic environment: diversity and memory[C]//Proc of the 6th International Conference on Neural Networks and Genetic Algorithms (ICANNGA '03). 2003: 168-174.
[7] 张著洪,钱淑渠. 自适应免疫算法及其对动态函数优化的跟踪[J]. 模式识别与人工智能, 2007,20(1):85-95.
[8] 钱淑渠,张著洪. 动态多目标免疫优化算法及其性能研究[J]. 智能系统学报, 2007,2(5):68-78.
[9] BASU S K, BHATIA A K. A naive genetic approach for non-stationary constrained problems[J]. Soft Computing,2006,10(2):152-162.