

文章编号:1001-9081(2008)03-0695-04

一种考虑 Agent 截止期限的 CPU 时间片分配算法

刘爱珍,王嘉祯,党 辰,陈立云,张西红

(军械工程学院 计算机工程系,石家庄 050000)

(liuazhen_163@.163.com)

摘 要:提出了一种综合考虑移动 Agent 执行时间片和执行截止期限要求的 CPU 时间片组合拍卖混沌遗传算法。该算法在定义了问题模型的基础上,设计了可减少遗传算法计算复杂度的预选择策略,以及变长的染色体编码方式、基于混沌优化技术的两种交叉算子(同父交叉算子和标准交叉算子),同时设计了基于混沌优化技术的换序算子和换标算子。实验仿真结果表明,该算法可取得优质的最优解。

关键词:移动 Agent; 组合拍卖; 遗传算法; 截止期限; CPU 时间片

中图分类号: TP18; TP301.6 **文献标志码:** A

CPU time slice allocation algorithm considering Agent deadline

LIU Ai-zhen, WANG Jia-zhen, DANG Chen, CHEN Li-yun, ZHANG Xi-hong

(Department of Computer Science, Ordnance Engineering College, Shijiazhuang Hebei 050000, China)

Abstract: A chaos-based genetic algorithm was proposed to solve CPU time slice combinational auction problem in which both agent time slice demand and execution deadline demand were considered. First, the problem model was defined. Then the pre-selecting method was designed to decrease computing complexity of genetic algorithm; and variable chromosome coding, two kinds of chaos-optimal-technology-based crossover operator (one for the same parent chromosome and one for the different parent chromosome), chaos-optimal-technology-based adjusting bid order operator and replacing bid operator were designed to solve it in genetic algorithm. Finally, the experimental results show this algorithm can obtain good optimal solution.

Key words: mobile Agent; combinational auction; Genetic Algorithm (GA); deadline; CPU time slice

0 引言

移动 Agent 作为一种新型的分布式应用,一般在异构网络的主机间自治地迁移来寻找适合其各任务执行要求的主机来执行。因此,资源共享尤其是计算资源、信息资源的共享成为必然,资源消费贯穿始终。由于它代表的是其创建方(用户)的利益,而不是其执行方(主机)的利益,因此其目标是在预算范围内,用最短的时间、最少的花费获得最多最快的资源。而允许移动 Agent 执行的主机系统为保证自己的利益,必须采取有效的资源分配机制来控制 Agent 对其资源的使用。因此,研究有效的资源分配方法(尤其是最具稀缺性的 CPU 时间片)有着积极的意义。

1 相关工作

当前 Agent 平台的 CPU 时间片调度策略主要有三种,1)传统的资源分配调度策略,如先来先分配策略(Aglet 平台);2)以保护主机资源不被滥用或非授权使用为目的的资源分配方法^[1];3)引入市场机制和拍卖机制的调度分配策略^[2-6]。对于第三种策略,移动 Agent 一般以电子货币来购买(交换)对资源的使用权限或数量。文献[5]用第一价格密封拍卖协议对若干 CPU 时间片进行拍卖分配,并用动态规划方法求得以近似最大化收入为目标的资源分配方案。该方法中,Agent 投标特点是对时间片的执行完毕截止时间的要求是

相同的,即在待拍 CPU 时间片内完成即可。没有考虑对执行完毕截止期限有各自具体要求的 Agent 的投标情况。考虑到有些 Agent 对执行时间有严格要求,必须在某个截止期限前执行完毕。本文研究考虑最大截止期限的 CPU 时间片组合拍卖问题。

2 问题描述

假设每轮待拍卖的 CPU 时间片为 $M = mT$ (每个时间片单元记为 T , m 为正整数),有 N 个 Agent 欲投标, N 个投标方的时间片总和总是大于待拍卖的时间片总和。任意 Agent i 可以根据一定的原则(如用户的需要、对当前环境的估计)确定需要的时间片数量 s_i ($s_i \leq mT$ 且是 T 的整数倍)、 s_i 的最大允许执行截止时间 d_i ($d_i \leq mT$) 及其基于上述要求的出价 b_i (记为 $b_i(s_i, d_i)$)。而后进行投标,拍卖方根据收益最大化原则,确定最后的竞胜标。数学公式如下:

$$\max_{X \in A} \sum_{b_i(s_i, d_i) \in X} b_i(s_i, d_i) \quad (1)$$

1) 以最大化拍卖方的收益为目标,采用密封标组合拍卖协议。

2) 式中 X 为一个可行解,满足每个时间片至多可分配给一个标,且满足每个标的截止期限要求。

3) 式中 A 为可行解空间,由所有可行解 X 组成。

4) 可行解确定的竞胜标的时间片总和 $\leq M$ 。

收稿日期:2007-09-12;修回日期:2007-11-29。 基金项目:河北省科技攻关项目(052435179D)。

作者简介:刘爱珍(1972-),女,河北景县人,博士研究生,主要研究方向:移动 Agent 智能算法; 王嘉祯(1944-),男,河北景县人,教授,主要研究方向:计算机网络安全; 党辰(1971-),男,河北石家庄人,博士研究生,主要研究方向:移动 Agent; 陈立云(1969-),男,湖南安仁人,副教授,硕士,主要研究方向:计算机网络; 张西红(1964-),男,河北衡水人,副教授,博士,主要研究方向:传感器网络、战术互联网。

上述问题虽形式简单,但属于 NP 难问题^[5]。和文献[5]的组合拍卖只考虑自身的利益及保证每个时间片只卖给一个投标方的前提相比,基于 Agent 最大截止期限的 CPU 拍卖还必须考虑投标者的最大允许截止期限问题。因此,它的求解更为复杂,用传统算法求解非常困难。遗传算法(Genetic Algorithm, GA)是模拟生物界自然进化和遗传过程的随机搜索算法,具有在解决大空间、非线性、全局寻优等复杂问题时传统方法所不具备的优越性^[7]。混沌是一种普遍存在的非线性现象,其遍历性特点可作为搜索过程中避免陷入局部最小的一种优化机制^[8]。为此,本文设计了一种求解该组合拍卖问题的基于混沌优化技术的遗传算法。

3 算法描述

整个算法分为两个阶段:第一阶段是预选择投标方阶段,该阶段的主要目的是采用各种方法,去掉那些不具竞争力的投标者,以减少运算空间;第二阶段使用 GA 对第一阶段得到的投标方集合进行运算,得到最优解或近似最优解。

3.1 预选择投标方策略

将截止期限 d_i 相同的各标,对于 $s_k > d_i/2$ 的标,每个 s_k 只取出价最高的那个标,这是因为受限于一个时间片只能卖给一个投标方的限制,对于时间片数量要求大于待拍卖时间片数量 M 的各投标方,最多只能有一个中标。同样原因,对于 $d_i/3 < s_k \leq d_i/2$ 的标,每个 s_k 只选取 2 个出价最高的标。对于 $d_i/4 < s_k \leq d_i/3$,每个 s_k 至多选取 3 个出价最高的标,依此类推, $s_k = 1T$ 时,至多选取 d_i 个出价最高的标。这里之所以用“至多”一词,是因为当 d_i 越大, s_k 越小时,除非其他 s_k 较大的标都不选,只选这种标,才会需要选取 d_i/s_k 个投标方,显然,这种情况一般不会出现。因此,为减少搜索空间,可以只选取部分出价较高的标。假设共有 n 个 Agent 参与投标,经预处理后选出 m 个有竞争力的标。显然 m 较 n 越小,则应用 GA 时搜索空间越小。

3.2 遗传算法设计

3.2.1 染色体编码

考虑到各 Agent 的截止期限要求和时间片要求不同,染色体编码直接由选中投标方的 ID 号按序排列组成。染色体中的基因构成一个可行解。每个基因对应于不同投标者的 ID 号: i (i 代表的投标方为 $b_i(s_i, d_i)$)。因为投标者对时间片的需求不同,所以,染色体的基因表达采用变长表示法,即染色体中的基因数不固定,染色体长度可变。假定经过第一阶段预处理后,还有 m 个投标方,记为 $B = \{1, 2, 3, \dots, i, \dots, m\}$,则可用式(2)表示一个染色体。

$$X = (1, 4, 7, i, \dots, j) \quad (2)$$

其中, $i \in \{k \mid k \in B, s'_k = s_k, d'_k \leq d_k\}$,且 X 满足: $s_1 + s_4 + s_7 + s_i + \dots + s_k \leq M$, d'_k 分别为 X 分配给标号为 k 的 Agent 的时间片和实际执行截止期限。

3.2.2 初始种群的生成

1) 从 B 中,随机选取投标方 i 作为首基因,则解(染色体)的第一个基因为 i ,此时 $X = (i)$ 。

2) 如果 $\sum_{i \in X} s_i < M$,则从 B 中寻找满足条件 $s_k \leq M - \sum_{i \in X} s_i$ 和 $d_k > \sum_{i \in X} s_i$ 的所有标集合 U 。如果 $U = \emptyset$,转 5); 否则转 3)。

3) 如果 $U \neq \emptyset$,从 U 中随机选取一个标 k ,若满足 $d_k \geq$

$\sum_{i \in X} s_i + s_k$,则将其加入到 X ,此时 $X = (i, k)$,并计算此时解 X 的时间片数量和 $\sum_{i \in X} s_i = \sum_{i \in X} s_i + s_k$,转 2)。否则转 4)。

4) 从 U 中删除标 k ,若 $U \neq \emptyset$,则转 3) 继续寻找满足条件要求的标。否则转 5)。

5) 如果 $U = \emptyset$,表明已不存在满足条件的标,染色体(解)生成完毕。

6) 重复 1) ~ 5) 直到达到种群规模为止。

由上述可知,染色体的长度由依次选取的投标方的截止期限要求和时间片数量和共同决定。

3.2.3 适应度函数

因目标函数是求拍卖收益最大值,所以染色体的适应度函数定义为:

$$F(X) = k \cdot f(X) \quad (3)$$

其中 $f(X)$ 为目标函数值, $0 < k \leq 1$ 为定标参数。

3.2.4 遗传算子

由于解用变长染色体表示,且每个基因对应的时间片不同,因此应用常规的 PMX 、 CX 、 OX 等交叉算子,很难产生合法解。为使杂交算子生成合法解,设计如下两种交叉算子。

同父交叉算子

所谓同父交叉算子,是指若随机选择的两个父个体 P_1 、 P_2 相同时要应用的交叉算子,如下:

1) 随机产生整数 r ($1 \leq r < \text{len}(P_1)$, $\text{len}(P_1)$ 代表染色体 P_1 的长度)。

2) 计算前 r 个基因相应标的时间片数量之和 L ;

新解 P'_1 保留 P_1 的前 r 个基因(即前 r 个标),后面的基因按如下原则选取:

a) 寻找标集 $U = \{k \mid s_k \leq M - L, d_k > L, k \notin P'_1, k \in B\}$ 。

b) 若 $U = \emptyset$,转 d)。

c) 随机选取一个标 $k \in U$,如果其 s_k 、 d_k 满足 $d_k \geq L + s_k$,则将 k 加入到解 P'_1 ,并更新解 P'_1 的时间片数量和: $L = L + s_k$ 。若 $L = M$,转 d),若 $L < M$,转 a)。如果 $d_k < L + s_k$,则从 U 中删掉 k ,转 b)。

d) 输出新染色体 P'_1 。

4) 新解 P'_2 保留 P_2 的第 r 基因之后的基因,前面的基因按如下原则选取:

a) 设 $L' = 0$; $i = 0$ 。

b) 寻找标集 $U = \{k \mid s_k \leq L - L', d_k > L', k \notin P'_2, k \in B\}$ 。若 $U = \emptyset$,转 c)。若 $U \neq \emptyset$,则随机选取一个标 $k \in U$,若满足 $d_k \geq L' + s_k$,则 $i = i + 1$,选中 k 作为 P'_2 的第 i 个基因,并计算前 i 个基因的时间片数量和 $L' = L' + s_k$,若 $L' = L$ 转 c),否则转 b)。

c) 将 P_2 的第 r 基因之后的基因依次加入到新解 P'_2 的第 $i + 1$ 及之后的基因位。输出新染色体 P'_2 。

应用该算子一般可得到新解,增强种群的多样性。

标准交叉算子

所谓标准交叉算子,是指随机选择的两个个体 $P_1 \neq P_2$,且其染色体长度均 > 1 时(可记为 $\min(\text{len}(P_1), \text{len}(P_2))$)应用的交叉算子,如下:

1) 随机产生整数 r ($1 \leq r < \min(\text{len}(P_1), \text{len}(P_2))$)。

2) 分别计算 P_1 、 P_2 前 r 个基因相应标的时间片数量和 L_1 、 L_2 ,不妨假设 $L_1 \leq L_2$ 。

3) 计算 $L' = L_2 - L_1$ 。

4) 若 $L' = 0$, 分别查找 P_1, P_2 的基因位 r 之后的标是否已在另一个染色体的前 r 个基因出现, 若无重复标, 则交叉后的新染色体 $P'_1 =$ 原染色体 P_1 的前 r 个标 + 原染色体 P_2 的 r 之后的标; $P'_2 =$ 原染色体 P_2 的前 r 个标中的 x 个标 + 原染色体 P_1 的 r 之后的标, 转 6)。否则, 转 5)。

5) 交叉后的新染色体 P'_1 的生成过程如下:

a) P_1 的前 r 个标作为新染色体 P'_1 的前 r 个标。

b) 将染色体 P_2 的 r 之后的所有标中, 满足不重复出现且满足截止期要求和时间片数量要求的标, 依次加入新染色体 P'_1 。

c) 计算此时新染色体 P'_1 的时间片数量和 L'_1 。

d) 若 $L'_1 < M$, 则从尚未被 P'_1 调度的标集合中, 寻找 $d > L'_1$ 且 $s \leq M - L'_1$ 的标集合 U ; 如果 $U \neq \emptyset$, 从 U 中, 随机选取一个标 k , 若其满足 $d_k \geq L'_1 + s_k$, 则将 k 加入到 P'_1 , 并计算此时的时间片数量和 $L'_1 = L'_1 + s_k$, 转 d)。否则转 6)。

6) 输出新染色体 P'_1 , 结束。

新染色体 P'_2 依同样原则生成。

换标算子

设计本算子是为了使染色体的收益变大。如下:

1) 选定一个混沌序列初值 x_n , 根据式(4) 生成一个 $[0, 1]$ 区间内来回变化的混沌序列 x_{n+1} , 将 x_{n+1} 作为动态控制该算子操作频率的依据:

$$x_{n+1} = 4x_n(1 - x_n) \quad (4)$$

2) 根据 x_{n+1} 及特定概率动态选择某染色体;

3) 依次对选定染色体的各基因, 分别从 B 中其他标中寻找时间片数量小于等于当前标, 但出价更高的标集, 如果标集的各标满足下列条件之一, 则应用本算子:

a) 截止期大于等于当前标的截止期, 则替换;

b) 截止期小于当前标的截止期, 但若替换当前标可满足其截止期要求, 则替换。

换序算子

1) 选定一个混沌序列初值 x_n , 根据式(4) 生成混沌序列 x_{n+1} , 将 x_{n+1} 作为控制该算子操作频率的依据;

2) 根据 x_{n+1} 及特定概率动态选择某染色体;

3) 从选定染色体的标中, 寻找时间片数量相同的标集, 若标数 > 1 , 将它们按各标对应的截止期限从小到大排序, 若解的这些标对应的基因位置与此排序不符, 则按此顺序替换各标在解中的基因位。

该算子不会改变染色体选中的标, 只会改变标的选中顺序, 从而使染色体的各标截止期的要求在得以满足的条件下, 整个解的截止期要求放宽, 进而放宽应用其他算子的限制。

3.2.5 选择策略

采用两代个体共同选优策略, 从上代 n 个个体 + 本代 n 个个体中, 选择 n 个适应度较好的个体作为下一代种群。

3.2.6 GA 算法框架

1) 按 3.2.2) 节生成一个规模为 N 的初始种群。

2) 评价初始种群。

3) 置群体代数 $g = 1$ 。

4) 依次实施交叉算子、换标算子、换序算子, 初步得到新一代种群。

5) 评价新一代种群的适应度, 按选择策略从两代个体中, 按适应度从大到小选择前 n 个作为确定的新一代群体。

6) 检验收敛准则, 若满足, 转 7); 否则 $g = g + 1$, 转 4)。

7) 输出最好个体。

4 拍卖协议整体框架

图 1、图 2 给出了拍卖方和投标方协商的具体过程。

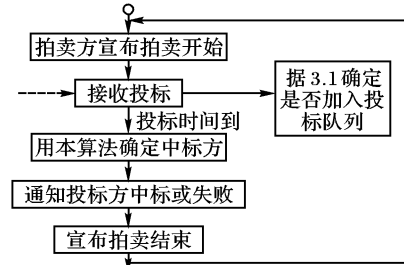


图 1 拍卖方有限状态图

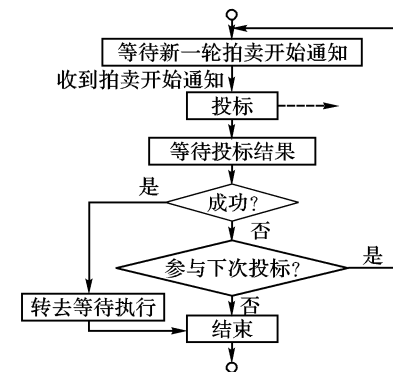


图 2 投标方有限状态图

5 实验分析及结论

为检验本算法的性能, 通过 Matlab 仿真在笔记本电脑 (Intel CPU 双核 T7200 2.0 GHz, RAM 2 GB) 进行了 4 组共 7 个实验。每个实验分别运行 20 次, 运行指定代数, 最后给出实验的平均结果, 并同时给出了采用传统 FCFS 算法的结果, 以便比较。其中第 2 组、第 4 组测试种群规模和迭代次数对解的影响。所有实验的投标方参数用矩阵 A 表示, $A_{ij} =$ (时间片数量, 截止期限, 出价, 标号); 各标的时间片数量为 $[1, M]$ 区间内的随机整数, 截止期限为 $[$ 各标的时间片数量, $M]$ 区间内的随机整数 (该范围可保证各标的最大截止时间不小于其所需的时间片数量), 出价空间为 $[2, M \times 5]$ 区间内的随机整数。标号 ID 设为由到达次序确定的整数。其余参数设置如表 1 所示。GA 各参数经多次实验调整, 交叉概率 = 1, 换标算子的操作概率 = 0.8, 换序算子的操作概率 = 0.8, 适应度定标参数 $k = 1$ 。

表 1 实验参数设置

序号	实验组别	拍卖时间片数 M	投标人 数 N	种群 规模	指定运 行代数	备注
1	1	10	10	10	200	
2	2-1	10	50	10	200	组实验投标方
3	2-2	10	50	20	200	参数矩阵相同
4	3	50	50	10	200	
5	4-1	50	200	10	200	
6	4-2	50	200	20	200	组实验投标方
7	4-3	50	200	20	500	参数矩阵相同

表 2 给出了 7 个实验的各种运行结果, 从每个实验运行 20 次所得到的最优解、最差的最优解、获得最优解的次数、平

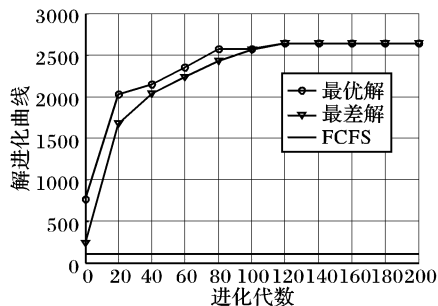
均值、标准差、收敛平均代数、到指定运行代数的平均运行时间等方面来反映本算法的性能。同时给出了传统 FCFS 算法所得到的解的结果。图 3 给出了第 4 组实验随机产生的初始种群的最优解及最差解随进化代数进化的整个过程。

由表 2 和图 3 可以看出,该算法不仅收敛速度快、鲁棒性强,而且在 4 组不同规模的实验中均表现出了良好的适应性,其解的质量较传统 FCFS 算法有着不可比拟的优势。从表 2 可

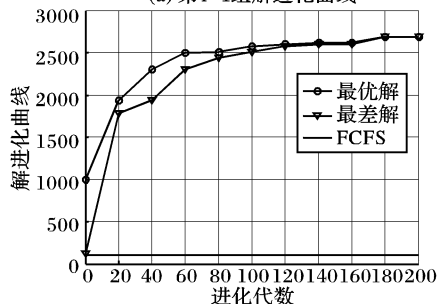
以看出,在待拍卖时间片数量及投标者较少的第一组实验中,20 次实验均得到了相同的最优解,且该值是问题的最优解。另外从表 2 中可以看出种群规模对算法所得最优解的影响不太明显,但算法运行时间增加量较明显,所以种群规模取 10~20 之间均可。从结果对比看,进化代数越多,得到的解的质量越高,花费的计算时间也越长,因此应综合平衡二者的关系。从其他几组实验也可以看出,该算法求得解的质量很高。

表 2 各组实验的综合结果

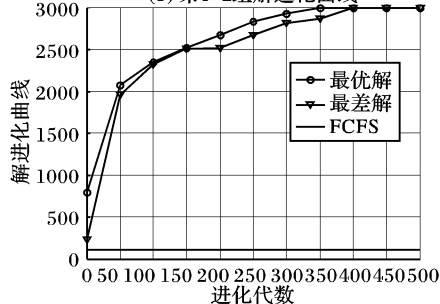
序号	最优解	最差解	平均值	获得最优解次数	标准差	指定运行代数 平均运行时间/s	预选择后 的标数	达到最优解平 均运行代数	FCFS 解
1	125	125	125.0	20	0.00	0.24	10	7	45
2	178	176	176.4	2	0.56	0.35	30	26	68
3	178	176	176.6	4	0.80	0.63	30	23	68
4	1545	1424	1510.0	12	42.03	0.59	47	84	545
5	2642	2217	2442.2	1	123.80	1.50	180	113	111
6	2685	2287	2506.2	1	112.00	2.61	180	117	111
7	2994	2418	2629.4	1	144.30	6.67	180	229	111



(a) 第4-1组解进化曲线



(b) 第4-2组解进化曲线



(c) 第4-3组解进化曲线

图 3 测试结果

鉴于 FCFS 算法一般不计算运行开销,该算法的解是按投标方的到达顺序依次调度执行,且满足其截止时间要求的那些标的解值。而本文算法是根据收益最大化原则来确定投标方的调度顺序,因此需要计算运行开销。

另外本实验的算法运行代数是指定的。实际解决问题时,可将收敛条件设为:若连续 10 代最优解没有进化,则随意加入两个新解,替代种群中的最差解,继续进化,若连续三次加入新解,种群最优解依然没有进化,则认为解群已收敛,退出。

从计算空间上看,假设有 N 个投标方,预选择阶段后筛选

出 k 个有竞争力的标。待拍卖的时间片数量为 M 。则本算法解的生成空间由 $N \times (N-1) \times \dots \times (N-M-1)$ 降为 $kk = k \times (k-1) \times \dots \times (k-M-1)$ 。以第 2 组实验为例,生成空间可减少到原来的 $30 \times 29 \times \dots \times 21/50 \times 49 \times \dots \times 41 = 12\%$ 左右。显然 M 越大, k 越小,本算法计算复杂度越小。

6 结语

本文设计了一种组合拍卖竞标遗传算法来改善 Agent 系统平台现有 CPU 时间片分配策略面临的资源分配问题。该算法可在保证主机系统利益最大化的前提下,综合权衡多个来访 Agent 的截止时间要求和 CPU 时间片数量要求,进而高效地确定分配方案。实验表明,该算法设计的各算子是有效的,获得的解是满意的,尤其是拍卖时间片和投标人数较少时可获得理论最优解。

参考文献:

- [1] LAI M, PANAY R. A scheduling scheme for controlling allocation of CPU resources for mobile programs[J]. Autonomous Agents and Multi-Agent Systems, 2002(5): 7-43.
- [2] 张凡, 房鼎益. 基于市场机制的移动 Agent 系统资源管理[J]. 西北大学学报: 自然科学版, 2003, 33(6): 649-652.
- [3] AWALDSPURGER C, HOGG T. Spawn: A distributed computational economy [J]. IEEE Transactions on Software Engineering, 1992, 18(2): 103-117.
- [4] WELLMAN M P, JEFFREY K, MASON M. Exploring bidding strategy for market-based scheduling [C]// Proceedings of the 4th ACM conference on Electronic commerce. San Diego: ACM Press, 2003: 115-124.
- [5] BREDIN J L. Market-based control of mobile-agent systems[D]. Dartmouth University, 2002.
- [6] REGGERS B. Markets for computational resources[D]. Universiteit Maastricht, 2004.
- [7] 王小平, 曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002.
- [8] WANG NING, LIU LI-MING, LIU LING-LING. Genetic algorithm in chaos[J]. Operations Research Transactions, 2001, 5(3): 1-10.