第 26 卷第 10 期
2009 年 10 月

计 算 机 应 用 研 究
Application Research of Computers

Vol. 26 No. 10
Oct. 2009

基于递阶遗传算法的多旅行商问题优化*

周辉仁[1,2], 唐万生[1], 牛犇[1,2]

(1. 天津大学 系统工程研究所, 天津 300072; 2. 山东建筑大学 管理工程学院, 济南 250101)

摘 要: 旅行商问题是一个经典的 NP 问题, 对多人旅行商问题的求解则更具有意义。为了解决所有旅行商路径总和最小为优化标准的多旅行商一类问题, 提出了一种递阶遗传算法和矩阵解码方法。该算法根据问题的特点, 采用一种递阶编码方案, 此编码与多旅行商问题一一对应。用递阶遗传算法优化多旅行商问题无须设计专门的遗传算子, 操作简单, 并且解码方法适于求解距离对称和距离非对称的多旅行商问题。计算结果表明, 递阶遗传算法是有效的, 能适用于优化多旅行商问题。

关键词: 递阶遗传算法; 多旅行商问题; 优化; 解码方法

中图分类号: TP301.6; TP18　　文献标志码: A　　文章编号: 1001-3695(2009)10-3754-04

doi: 10.3969/j.issn.1001-3695.2009.10.044

Optimization of multiple traveling salesman problem based on hierarchical genetic algorithm

ZHOU Hui-ren[1,2], TANG Wan-sheng[1], NIU Ben[1,2]

(1. Institute of Systems Engineering, Tianjin University, Tianjin 300072, China; 2. School of Management & Engineering, Shandong Jianzhu University, Jinan 250101, China)

Abstract: Traveling salesman problem is a classical nondeterministic polynomial problem. It is significance to solve multiple traveling salesman problems (MTSP). In order to solve MTSP that employed total-path-shortest as the evaluating rule, this paper proposed a hierarchical genetic algorithm and decoding method with matrix. Its coding method is simple and can effectively reflect the traveling policy, and the methods of crossover and mutation are not special to design. By this method, symmetric and asymmetric multiple traveling salesman problems can be easily solved. The computational results suggest that the hierarchical genetic algorithm is efficient and fit for multiple traveling salesman problems.

Key words: hierarchical genetic algorithm; multiple traveling salesman problem; optimization; decoding method

0 引言

旅行商问题 (traveling salesman problem, TSP) 是一个典型的组合优化难题, 它在许多领域都有着广泛的应用, 已被证明属于 NP 问题[1]。所谓 TSP 问题是指: 有 N 个城市, 要求旅行商到达每个城市各一次, 且仅一次, 并回到起点, 且要求旅行路线最短。而多旅行商问题 (MTSP) 是指 M 个旅行商从同一个城市 (或不同城市) 出发, 分别走一条旅行路线, 使得每个城市有且仅有一个旅行商经过 (出发城市除外), 且总路程最短。有关 TSP 问题的研究在现实问题中有很大的使用价值。例如, 交通运输、管道敷设、路线选择、计算机网络的拓扑设计、邮递员送信等, 均可抽象成 TSP 或 MTSP 问题[2~5]。

美国 Michigan 大学的 Holland 教授提出的遗传算法 (genetic algorithm, GA) 是求解复杂的组合优化问题的有效方法, 其思想来自于达尔文进化论和门德尔松遗传学说。它模拟生物进化过程来从庞大的搜索空间中筛选出较优秀的解, 是一种高效而且具有强鲁棒性方法。所以, 遗传算法在求解 TSP 和 MTSP 问题中得到了广泛的应用。目前, 用遗传算法求解 MTSP 的问题大都是以距离对称的 MTSP 问题, 一般是通过附加虚拟城市的方法将 MTSP 问题转换为 TSP 问题[5,6]。

为了有效解决所有旅行商的路程最小及距离对称或者非对称的 MTSP 问题, 本文提出了一种递阶遗传算法 (hierarchical genetic algorithm, HGA) 和矩阵解码方法, 以便确定每个城市由哪个旅行商经过以及各个旅行商的行走路线, 即找到一个最优旅行商分配及行走路线, 在各旅行商行走完后, 使旅行商路程总和最小。仿真结果表明, 本文提出的算法是有效的。

1 多旅行商问题的数学模型

以点 0 表示旅行商的出发城市, 称为源点, 点 1, …, l 表示 m 个旅行商需访问的城市。

定义变量:

$$x_{ijk}=\begin{cases}1 & \text{旅行商 } k \text{ 通过弧 } (i,j)\\ 0 & \text{否则}\end{cases}$$

$$y_{ki}=\begin{cases}1 & \text{旅行商 } k \text{ 访问城市 } i\\ 0 & \text{否则}\end{cases}$$

收稿日期: 2009-01-14; 修回日期: 2009-02-19　　基金项目: 中国博士后科学基金资助项目 (20090450759)

作者简介: 周辉仁 (1972-), 男, 山东高密人, CCF 会员, 博士后在站, 博士, 主要研究方向为智能优化理论与方法、系统管理智能优化与决策等 (huirenzhou@126.com); 唐万生 (1962-), 男, 天津人, 教授, 博导, 主要研究方向为复杂系统建模与控制等; 牛犇 (1969-), 男, 山东济宁人, 博士研究生, 主要研究方向为项目管理等.

其中: c_{ij}表示旅行商经过对应弧段(i, j)的距离。

则得到以下模型:

目标函数为

$$Z=\min(\sum_{k=1}^{m} z_k) \tag{1}$$

其中:

$$z_k=\sum_{i=0}^{l}\sum_{j=0}^{l} c_{ij}x_{ijk}\ (k=1, 2, \cdots, m) \tag{2}$$

约束条件

$$\sum_{k=1}^{m} y_{ki}=\begin{cases} m & i=0 \\ 1 & i=1, 2, \cdots, l \end{cases} \tag{3}$$

$$\sum_{i=0}^{l} x_{ijk}=y_{kj} \tag{4}$$

其中: $j=0, 1, \cdots, l$; $k=1, 2, \cdots, m$。

$$\sum_{j=0}^{l} x_{ijk}=y_{ki} \tag{5}$$

其中: $i=0, 1, \cdots, l$; $k=1, 2, \cdots, m$。

$$X=(x_{ijk})\in S \tag{6}$$

其中: S为支路消去约束, 即消去构成不完整路线的解, 具体含义可参见文献[6]。

在该模型中, 式(1)表示使 m个旅行商的所有路程最小化; 式(2)表示各个旅行商的所遍历的路程; 式(3)表示从指定城市 0出发, 所有城市只有某一个旅行商严格访问一次; 式(4)表示任一条弧的终点城市仅有一个起点城市与之相连; 式(5)表示任一条弧的起点城市仅有一个终点城市与之相连; 式(6)表示消去构成不完整线路的解。

2 递阶遗传算法

在生物学领域, 染色体的结构是一系列基因按层次排列而成的, 一些基因控制着另一些基因。染色体可表示为包括控制基因和参数基因的递阶结构, 参数基因处于最低级, 控制基因处于上级, 下级基因串受上级基因的控制。在基因编码时, 控制基因常采用整数编码, 不同整数信息表示对应的基因处于不同的激活状态, 而与该基因相联系的低级基因串则处于对应的状态。为计算方便和加强遗传算法在解空间的搜索能力, 参数基因采用实数编码, 每个基因用一个实数代表。这样定义染色体结构的遗传算法称为递阶遗传算法, 它比传统遗传算法包含更多的信息, 因而能处理更复杂的问题。目前, 递阶遗传算法已在神经网络、模糊系统、车间调度等得到了较好的应用[7~9]。

3 递阶遗传算法设计

基于多旅行商问题的特点, 可以设计成二级递阶染色体结构描述多旅行商问题的结构和参数, 控制基因中的每一个等位基因表示城市, 参数基因中的每一个等位基因表示所路过的旅行商。对于给定问题, 其控制基因和参数基因个数是确定的, 都为不含源点的城市个数 l, 控制基因取值为 1~l中互不相等的整数, 参数基因取值为 1~m中的整数, m为旅行商个数, 因此优化多旅行商问题只需确定基因信息。例如, 对于有 8个城市(含源点城市)、旅行商数为 3的 MTSP问题, 假设城市 0为源点, 本文采用二级递阶染色体结构描述。例如控制基因中第 1个等位基因取值为 4 对应的参数基因取值为 2时, 表示城市 4由旅行商 2访问; 控制基因中第 2个等位基因取值为 5 对应的参数基因取值为 3, 表示城市 5由旅行商 3访问, 依此类推。

3.1 群体规模选择

合适的群体规模对遗传算法的收敛具有重要意义。群体太小难以求得满意的结果, 群体太大则计算复杂。根据经验, 群体规模一般取 10~160。

3.2 适值函数

遗传算法在进行选择操作时会出现欺骗问题[10]: 在遗传进化的初期, 通常会产生一些超常的个体, 若按照比例选择法, 这些异常个体因竞争力太突出而控制了选择过程, 影响算法的全局优化性能; 在遗传进化的后期, 即算法接近收敛时, 由于种群中个体适值差异较小时, 继续优化的潜能降低, 可能获得某个局部最优解。适值函数的设计不当有可能造成这种问题的出现。

由于优化目标为最小化路程或费用值, 因此令目标函数作指数变换得到适值函数:

$$f=\alpha \exp(-\beta * Z) \tag{7}$$

其中: α、β为正实数。

3.3 选择

选择是用来确定重组或交叉个体, 以及备选个体将产生多少个子代个体。选择的第一步是计算适值, 采用按比例的适值分配, 是利用比例于各个个体适值的概率决定其子孙的遗留可能性。若某个个体 i, 其适值为 f_i, 则其被选择的概率表示为

$$P_i=f_i/\sum_{k=1}^{M} f_k \tag{8}$$

然后对各个染色体计算它们的累积概率:

$$q_i=\sum_{j=1}^{M} P_j \tag{9}$$

第二步用轮盘赌选择法进行选择。为了选择交配个体, 需要进行多轮选择, 每一轮产生一个[0, 1]均匀随机数, 将该随机数作为选择指针来确定备选个体。

3.4 交叉与变异

递阶遗传算法的交叉、变异操作分为控制基因交叉、变异和参数基因交叉、变异。

交叉在遗传操作中起核心作用, 交叉概率较大可增强遗传算法开辟新搜索空间的能力, 但性能好的基因串遭到破坏的可能性较大, 算法收敛速度降低, 且不稳定; 若交叉概率较小, 则遗传算法搜索可能陷入迟钝状态。对于控制基因串交叉可以采用基于路径表示部分匹配交叉(partially matched crossover, PMX)、顺序交叉(ordered crossover, OX)、循环交叉(cycle crossover, CX)等[10]; 对于参数基因串采用单点交叉。

部分匹配交叉是 Goldberg等人于 1985年针对 TSP提出的基于路径表示的交叉操作, 部分匹配交叉操作要求随机选取两个交叉点, 以便确定一个匹配段, 根据两个父个体中两个交叉点之间的中间段给出的映射关系生成两个子个体; 顺序交叉是 Davis等人于 1985年针对 TSP提出的基于路径表示的交叉操作, 该操作能保留排列并融合不同排列的有序结构单元。两

个父体交叉时, 通过选择父个体 1的一部分, 保存父体 2中代码的相对顺序生成子个体; 循环交叉是 Oliver等人针对 TSP提出的交叉操作, 循环交叉操作中子个体中的代码顺序根据任一父个体产生。

变异在遗传操作中属于辅助性的搜索操作, 主要目的是维持群体的多样性。较低的变异概率可以防止群体中重要的单一基因丢失, 但降低了遗传算法开辟新搜索空间的能力; 较高的变异概率将使遗传操作趋于纯粹的随机搜索, 降低了算法的收敛速度和稳定性。对于控制基因串采用交换变异, 即交换两个随机位置上的基因; 对于参数基因串采用整数变异, 即对每一基因以一定概率用 [1, m] 之间区别于该基因的整数加以替代。

3.5 解码

如图 1所示的染色体递阶结构, 由以上介绍的染色体递阶结构的含义可知: 旅行商 1访问城市 6、7; 旅行商 2访问城市 1、4; 旅行商 3访问城市 2、3、5。这里, 规定控制基因串从左至右为城市被访问的先后顺序。因此, 可以得到:

旅行商 1访问的城市及顺序为 0-6-7-0;

旅行商 2访问的城市及顺序为 0-4-1-0;

旅行商 3访问的城市及顺序为 0-5-2-3-0。

由于该问题为 8个旅行城市 (包含源点城市 0), 城市之间的距离为 8×8的矩阵, 设该矩阵为 D。为了避免某个旅行商出现 0-0-0的行走路径, 如图 2所示的染色体递阶结构。

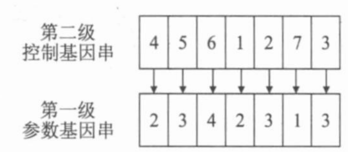

图1 染色体的递阶结构设计

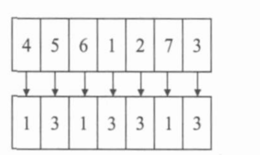

图2 染色体递阶结构

根据递阶结构的含义, 可得到 3个旅行商的行走路径分别为:

0-4-6-7-0;

0-0-0;

0-5-1-2-3-0。

如上所示, 旅行商 2出现了 0-0-0的路径, 因此在矩阵 D中, 要将城市 0-0的距离设为很大的正数 M, 这样可以使出现这种情况的染色体在选择复制过程中被淘汰。

这里, 先设矩阵 D为对称距离矩阵, 并设城市 0-0的距离为 10 000, 对于非对称距离矩阵下面的解码方法同样适用。

$$\begin{array}{cl}
\begin{array}{ccc} \text{城市 }0 & \cdots & \text{城市 }7 \\ \downarrow & & \downarrow \end{array} & \\
D=\begin{bmatrix}
10000 & 13 & 10 & 11 & 9 & 8 & 18 & 21 \\
13 & 0 & 5 & 9 & 14 & 11 & 21 & 7 \\
10 & 5 & 0 & 13 & 6 & 21 & 16 & 17 \\
11 & 9 & 13 & 0 & 11 & 13 & 19 & 12 \\
9 & 14 & 6 & 11 & 0 & 14 & 16 & 18 \\
8 & 11 & 21 & 13 & 14 & 0 & 8 & 19 \\
18 & 21 & 16 & 19 & 16 & 8 & 0 & 22 \\
21 & 7 & 17 & 12 & 18 & 19 & 22 & 0
\end{bmatrix} & \begin{array}{l} \leftarrow \text{城市 }0 \\ \\ \\ \\ \cdots \\ \\ \\ \leftarrow \text{城市 }7 \end{array}
\end{array}$$

对于图 1所示的染色体递阶结构, 在应用计算机具体编程时可采用如下解码步骤:

a) 由旅行商 1的访问路径可得

$$x_{061}=1,\ x_{671}=1,\ x_{701}=1.$$

因此, 旅行商 1的可达矩阵为

$$\begin{array}{cl}
\begin{array}{ccc} \text{城市 }0 & \cdots & \text{城市 }7 \\ \downarrow & & \downarrow \end{array} & \\
X_1=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} & \begin{array}{l} \leftarrow \text{城市 }0 \\ \\ \\ \\ \cdots \\ \\ \\ \leftarrow \text{城市 }7 \end{array}
\end{array}$$

同理, 旅行商 2、3的可达矩阵分别为

$$X_2=\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}\quad
X_3=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

b) 则每个旅行商的行走路程。

对于旅行商 1将矩阵 X_1与矩阵 D点乘得到矩阵 D_1, 矩阵 D_1中不为 0的数值之和即为旅行商 1所访问城市的距离。设旅行商 1、2、3所访问城市的距离分别为 z_1、z_2、z_3。

$$D_1=X_1\times D$$

$$D_1=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 \\
21 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$z_1=18+22+21=61$$

同理

$$z_2=9+14+13=36$$

$$z_3=8+21+13+11=53$$

c) 最后求得所有旅行商的行走路程:

$$Z=z_1+z_2+z_3=150$$

4 实例仿真及比较

以上介绍了递阶遗传算法优化 MTSP问题的步骤, 本章通过 TSPLIB中的距离非对称的数据进行仿真。

4.1 17个城市 (br17.atsp)的 MTSP问题

br17.atsp数据总共有 17个城市, 并且其距离矩阵是非对称的。设旅行商数为 3, 进行仿真。

这里, 对于控制基因串分别采用循环交叉 (CX)、顺序交叉 (OX)和部分匹配交叉 (PMX), 交叉概率为 0.75, 变异采用交换变异且变异概率为 0.25; 对于参数基因串采用单点交叉且交叉概率为 0.75, 变异采用整数变异且变异概率为 0.01; 种群规模为 100, 最大迭代次数为 200。用 MATLAB语言编程, 随机运行 10次, 所得结果如表 1所示。

表 1 Br17. atsp的仿真结果 (HGA)

交叉算子	最优解	最差解	平均解
CX	42	52	48.6
OX	42	53	47.8
PMX	42	52	47.3

本文所求的一个最优解的路线分别为

1-6-7-16-15-5-4-9-8-17-1;

1-11-13-10-2-3-14-1;

1-12-1。

为了便于比较, 对该实例用标准遗传算法进行仿真。交叉算子同样分别采用循环交叉 (CX)、顺序交叉 (OX)和部分匹配交叉 (PMX), 交叉概率为 0.75, 变异采用交换变异且变异概率为 0.25, 种群规模为 100, 最大迭代次数为 200。用 MATLAB语言编程, 随机运行 10次, 所得结果如表 2所示。

表 2 Br17. atsp的仿真结果 (GA)

交叉算子	最优解	最差解	平均解
CX	44	60	49.2
OX	42	55	47.2
PMX	42	55	47.9

本实例是一个小规模的 MTSP问题, 通过表 1、2数据比较可以看出, 递阶遗传算法所求的结果是比较稳定的, 优于标准遗传算法的结果。

4.2 100个城市 (kro124p)的 MTSP问题

Kro124p数据总共有 100个城市, 并且其距离矩阵是非对称的。设旅行商数分别为 4、8、12, 进行仿真。

这里, 对于控制基因串分别采用循环交叉 (CX)和部分匹配交叉 (PMX), 且交叉概率为 0.65, 变异采用交换变异且变异概率为 0.25; 对于参数基因串采用单点交叉且交叉概率为 0.75, 变异采用整数变异且变异概率为 0.003, 种群规模为 100, 最大迭代次数为 800。用 MATLAB语言编程, 旅行商数分别为 4、8、12时, 各随机运行 10次, 所得结果如表 3所示。

表 3 Kro124p的仿真结果 (HGA)

旅行商数	交叉算子	最优解	最差解	平均解
4	CX	65 115	73 555	69 005
	PMX	63 971	70 530	66 925
8	CX	65 963	75 537	71 960
	PMX	67 142	73 117	70 339
12	CX	71 536	81 204	75 167
	PMX	72 870	80 188	75 584

为了便于比较, 对该实例用标准遗传算法进行仿真。交叉算子同样分别采用循环交叉 (CX)和部分匹配交叉 (PMX), 交叉概率为 0.75, 变异采用交换变异且变异概率为 0.25, 种群规模为 100, 最大迭代次数为 800。用 MATLAB语言编程, 随机运行 10次, 所得结果如表 4所示。

表 4 Kro124p的仿真结果 (GA)

旅行商数	交叉算子	最优解	最差解	平均解
4	CX	66 733	78 482	71 944
	PMX	63 465	73 784	67 333
8	CX	71 287	81 084	76 011
	PMX	67 780	76 447	71 225
12	CX	72 154	85 455	79 710
	PMX	73 062	84 095	78 282

本实例是一个大规模的 MTSP问题, 从表 3、4的比较可以看出, 对于给定的旅行商数和交叉算子在最优解和平均解方面, 递阶遗传算法总体上要优于标准遗传算法。

5 结束语

本文对最小化所有旅行商路程问题进行了研究, 并提出递阶遗传算法来优化多旅行商问题。该算法基因编码简洁, 能清楚地反映出旅行商与城市的关系及访问顺序, 并且研究了解码方法, 适合于优化距离对称和非对称的多旅行商问题。

参考文献:

[1] 李军民, 林淑飞, 高让礼. 用混合遗传算法求解多目标 TSP问题 [J]. 西安科技大学学报, 2006, 26(4): 515-518.

[2] 陶然, 吕红霞, 陈广秀. 基于 MTSP的机车周转图编制模型与算法 [J]. 西南交通大学学报, 2006, 41(5): 653-657.

[3] 卢厚清, 王辉东, 黄杰, 等. 任务均分的多旅行商问题 [J]. 系统工程, 2005, 23(2): 22-24.

[4] 唐立新. 热轧调度并行处理策略的多旅行商模型 [J]. 东北大学学报, 1999, 20(2): 148-150.

[5] 黄可伟, 汪定伟. 热轧计划中的多旅行商问题及其计算方法 [J]. 计算机应用研究, 2007, 24(7): 43-45.

[6] 李军, 郭耀煌. 物流配送车辆优化调度理论与方法 [M]. 北京: 中国物资出版社, 2001: 63-73.

[7] 周辉仁, 郑丕谔. 基于递阶遗传算法的并行多机调度优化 [J]. 计算机应用, 2007, 27(9): 2273-2275.

[8] 周辉仁, 郑丕谔. 四层 BP网络的一种结构设计方法及应用 [J]. 系统仿真学报, 2008, 20(9): 2325-2328.

[9] 周辉仁, 郑丕谔. 模糊系统的一种结构设计方法及经济系统应用 [J]. 系统工程理论与实践, 2008, 28(4): 101-107.

[10] 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: 28-38.

(上接第 3737页)

参考文献:

[1] 王美仙, 李明. 先进战斗机控制分配方法研究综述 [J]. 飞机设计, 2006(3): 17-19.

[2] ANDREW R T, JAMES M B. Anti-windup for an F-16's daisy chain control allocator [C] // Proc of AIAA GNC Conference. New Orleans: [s. n.], 1997: 748-754.

[3] KENNEDY J, EBERHART R. Particle swarm optimization [C] // Proc of International Conference on Neural Networks. Piscataway: USA [s. n.], 1995: 1942-1947.

[4] 王介生, 王金城, 王伟. 基于粒子群算法的 PID控制器参数自整定 [J]. 控制与决策, 2005, 20(1): 73-76.

[5] DURHAM W C. Computationally efficient control allocation [J]. Journal of Guidance, control and Dynamics, 2001, 24(3): 519-524.

[6] BORDIGNON K A. Constrained control allocation for systems with redundant control effectors [D]. Blacksburg, VA: Department of Aerospace and Ocean Engineering, Virginia Polytechnic Inst and State Univ, 1996.