

## 蚁群算法与粒子群优化算法的混合算法

高 尚<sup>1,2</sup>, 蒋新姿<sup>1</sup>, 汤可宗<sup>1</sup>, 杨静宇<sup>3</sup>

1. 江苏科技大学电子信息学院, 镇江 212003

Email: gao\_shang@hotmail.com

2. 苏州大学江苏省计算机信息处理技术重点实验室, 苏州 215006

3. 南京理工大学计算机系, 南京 210094

**摘要:**根据蚁群算法与粒子群优化算法的特性, 提出了求解旅行商问题的混合算法。首先随机产生若干组比较好的解生成信息素分布, 然后由蚁群算法根据累计更新的信息素找出若干组解后, 再由粒子群算法进行交叉、变异操作, 得到更有效的解。与模拟退火算法、标准遗传算法和蚁群算法进行比较, 16种混合算法效果都比较好, 其中交叉策略B和变异策略B的混合算法效果最好。

**关键词:**蚁群算法, 粒子群优化算法, 旅行商问题

## Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Particle Swarm Optimization

Gao Shang<sup>1,2</sup>, Jiang Xin-zhi<sup>1</sup>, Tang Kezong<sup>1</sup>, Yang Jingyu<sup>3</sup>

1. School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003

Email: gao\_shang@hotmail.com

2. Provincial Key Laboratory of Computer Information Processing Technology, Suzhou 215006

3. Department of Computer, Nanjing University Science and Technology, Nanjing 210094

**Abstract:** By use of the properties of ant colony algorithm and particle swarm optimization , a hybrid algorithm is proposed to solve the traveling salesman problems. First, it adopts statistics method to get several initial better solutions and in accordance with them, gives information pheromone to distribute. Second, it makes use of the ant colony algorithm to get several solutions through information pheromone accumulation and renewal. Finally, by using across and mutation operation of particle swarm optimization, the effective solutions are obtained. Compare with the simulated annealing algorithm, the standard genetic algorithm and the standard ant colony algorithm, all the 16 hybrid algorithms are proved effective. Especially the hybrid algorithm with across strategy B and mutation strategy B is a simple and effective better algorithm than others.

**Key Words:** Ant colony algorithm ,Particle swarm optimization ,Traveling salesman problem

## 1 引言(Introduction)

上世纪50年代中期创立了仿生学, 人们从生物进化的机理中受到启发, 提出了许多用以解决复杂优化问题的新方法, 如遗传算法、模拟退火算法等。目前研究的热点是模拟生物群局生活的特点而提出的群智能算法, 作为一种新兴演化计算技术, 它与人工生命, 特别是进化策略和遗传算法有着极为特殊的联系, 是一种能够有效解决大多数全局优化问题的新方法。在计算智能领域目前主要有两种

群智能的算法: 蚁群算法和粒子群优化算法。前者是对蚂蚁群落食物采集过程的模拟, 由意大利学者Colorni A, Dorigo M和Maniezzo V于1992年首先提出来<sup>[1]</sup>, 目前国内外研究者用蚁群算法研究了旅行商问题、指派问题、调度问题等, 取得了一系列较好的实验结果<sup>[2,3]</sup>, 如蚁群算法在电信路由优化<sup>[4]</sup>、数据聚类分析<sup>[5]</sup>、数据分类规则提取<sup>[6]</sup>等方面效果也很好, 已显示出蚁群算法在求解复杂优化问题(特别是离散优化问题)方面的一些优越性, 证明它是一种很有发展前景的方法。粒子群优化算法也是起源对简单社会系统的模拟, 模拟鸟群觅食的过程, 最早是由Kennedy与Eberhart于1995年提出来的<sup>[7]</sup>。目前粒子群优化算法已广泛应用于函数优化, 神经网络的训练, 模糊系统控制以及其他应用

领域。目前已提出了多种改进的和扩展的粒子群优化算法<sup>[8, 9]</sup>, 如自适应粒子群优化算法、杂交粒子群优化算法、协同粒子群优化算法等。蚁群算法擅长解决离散问题的优化问题, 而粒子群优化算法擅长连续问题的优化问题, 如何把这两种算法结合在一起进行研究, 总结出共同规律, 无疑是一个新的方法。本文尝试用这两种混合方法来解决旅行商问题。

## 2基本蚁群算法和粒子群优化算法(Basic Ant Colony Algorithm and Particle Swarm Optimization)

### 2.1 基本蚁群算法(Basic Ant Colony Algorithm)

蚂蚁个体之间是通过一种称之为外激素的物质进行信息传递, 从而能相互协作, 完成复杂的任务。蚂蚁在运动过程中, 能够在它所经过的路径上留下该种物质, 而且蚂蚁在运动过程中能够感知这种物质的存在及其强度, 并以此指导自己的运动方向, 蚂蚁倾向于朝着该物质强度高的方向移动。因此, 由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象: 某一路径上走过的蚂蚁越多, 则后来者选择该路径的概率就越大。蚂蚁个体之间就是通过这种信息的交流达到搜索食物的目的。蚁群算法首先成功应用于旅行商问题, 下面简单介绍其基本算法。

设有  $m$  个蚂蚁, 每个简单蚂蚁有以下特征: 它根据以城市距离和连接边上外激素的数量为变量的概率函数选择下一个城市 (设  $\tau_{ij}(t)$  为  $t$  时刻边  $e(i, j)$  上外激素的强度)。规定蚂蚁走合法路线, 除非周游完成, 不允许转到已访问城市, 由禁忌表控制 (设  $tabu_k$  表示第  $k$  只蚂蚁的禁忌表,  $tabu_k(s)$  表示禁忌表中第  $s$  个元素)。它完成周游后, 蚂蚁在它每一条访问的边上留下外激素。

初始时刻, 各条路径上的信息量相等, 设  $\tau_{ij}(0)=C$  ( $C$  为常数)。蚂蚁  $k$  ( $k=1, 2, \dots, m$ ) 在运动过程中, 根据各条路径上信息量决定转移方向,  $p_{ij}^k(t)$  表示在  $t$  时刻蚂蚁  $k$  由位置  $i$  转移到位置  $j$  的概率,

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_j^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \cdot \eta_s^\beta(t)} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

其中,  $allowed_k = \{0, 1, \dots, n-1\} - tabu_k$  表示蚂蚁  $k$  下一步允许选择的城市, 与实际蚁群不同, 人工蚁群系统具有记忆功能,  $tabu_k$  ( $k=1, 2, \dots, m$ ) 用以记录蚂蚁  $k$  当前所走过的城市, 集合  $tabu_k$  随

着进化过程做动态调整。 $\eta_{ij}$  表示边弧  $(i, j)$  的能见度, 用某种启发式算法算出, 一般取  $\eta_{ij} = 1/d_{ij}$ ,  $d_{ij}$  表示城市  $i$  与城市  $j$  之间的距离。 $\alpha$  表示轨迹的相对重要性,  $\beta$  表示能见度的相对重要性,  $\rho$  表示轨迹的持久性,  $1-\rho$  理解为轨迹衰减度随着时间的推移, 以前留下的信息逐渐消失, 用参数  $1-\rho$  表示信息消逝程度, 经过  $n$  个时刻, 蚂蚁完成一次循环, 各路径上信息量要根据以下式做调整:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (2)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

$\Delta \tau_{ij}^k$  表示第  $k$  只蚂蚁在本次循环中留在路径  $ij$  上的信息量,  $\Delta \tau_{ij}$  表示本次循环中路径  $ij$  上的信息量增量,  $L_k$  表示第  $k$  只蚂蚁环游一周的路径长度,  $Q$  常数。

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁在本次循环经过 } ij \\ 0 & \text{否则} \end{cases} \quad (4)$$

### 2.2 基本粒子群优化算法(Basic Particle Swarm Optimization)

粒子群优化算法是模拟鸟群的捕食行为, 一群鸟在随机搜索食物, 在这个区域里只有一块食物。所有的鸟都不知道食物在那里。但是他们知道当前的位置离食物还有多远, 最简单有效的就是搜寻目前离食物最近的鸟的周围区域。粒子群优化算法从这种模型中得到启示并用于解决优化问题。粒子群优化算法中每个优化问题的解看作搜索空间中的一只鸟, 称之为“粒子”。所有的粒子都有一个由被优化的函数决定的适应值, 每个粒子还有一个速度决定他们飞翔的方向和距离, 然后粒子们就追随当前的最优粒子在解空间中搜索。粒子群优化算法初始化为一群随机粒子(随机解), 然后通过迭代找到最优解。在每一次迭代中, 粒子通过跟踪两个“极值”来更新自己。一个是粒子本身所找到的最优解, 这个解叫做个体极值  $pbest$ , 另一个极值是整个种群目前找到的最优解, 这个极值是全局极值  $gbest$ , 另外也可以不用整个种群而只是用其中一部分为粒子的邻居, 那么在所有邻居中的极值就是

局部极值。在找到这两个最优值时，每个粒子根据如下的公式来更新自己的速度和新的位置：

$$\mathbf{v}_{k+1} = c_0 \mathbf{v}_k + c_1 (\mathbf{pbest}_k - \mathbf{x}_k) + c_2 (\mathbf{gbest}_k - \mathbf{x}_k) \quad (5)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1} \quad (6)$$

其中： $\mathbf{v}_k$  是粒子的速度向量； $\mathbf{x}_k$  是当前粒子的位置； $\mathbf{pbest}_k$  粒子本身所找到的最优解的位置； $\mathbf{gbest}_k$  整个种群目前找到的最优解的位置； $c_0$ ， $c_1$ ， $c_2$  表示群体认知系数， $c_0$  一般取介于 (0, 1) 之间的随机数， $c_1$ ， $c_2$  取 (0, 2) 之间的随机数。

$\mathbf{v}_{k+1}$  是  $\mathbf{v}_k$ ， $\mathbf{pbest}_k - \mathbf{x}_k$  和  $\mathbf{gbest}_k - \mathbf{x}_k$  矢量的之和。在每一维粒子的速度都会被限制在一个最大速度  $\mathbf{v}_{max}$  ( $\mathbf{v}_{max} > 0$ ) 内，如果某一维更新后的速度超过用户设定的  $\mathbf{v}_{max}$ ，那么这一维的速度就被限定为  $\mathbf{v}_{max}$ ，即若  $\mathbf{v}_k > \mathbf{v}_{max}$  时， $\mathbf{v}_k = \mathbf{v}_{max}$  或  $\mathbf{v}_k < -\mathbf{v}_{max}$  时， $\mathbf{v}_k = -\mathbf{v}_{max}$ 。

### 3 蚁群算法与粒子群优化算法的混合 (Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Particle Swarm Optimization)

#### 3.1 蚁群算法的改进 (Improvement of Ant Colony Optimization)

蚁群算法初始化时，各路径的信息素取相同值，让蚂蚁以等概率选择路径，这样使蚂蚁很难在短时间内从大量的杂乱无章的路径中，找出一条较好的路径，所以收敛速度较慢。假如初始化时就给出启发性的信息，可以加快收敛速度。改进的方法是，产生大量的路径（如100条），从中选择比较优的（如30条），使这些路径留下信息素（与路径长度之和成反比），各路径的信息量就不同，以此引导蚂蚁进行选择路径。

蚂蚁每次周游结束后，不论蚂蚁搜索到的解如何，都将赋予相应的信息增量，比较差的解也将留下信息素，这样就干扰后续的蚂蚁进行寻优，造成大量的无效的搜索。改进的方法是，只有比较好的解才留下信息素，即只有当路径长度小于给定的值才留下信息素。为了充分利用各蚂蚁所走过的路径信息，随时记录当前的最好解。

蚁群算法利用了信息素进行传递信息，而粒子群优化算法利用了本身信息、个体极值信息和全局极值三个信息，来指导粒子下一步迭代位置。蚁群算法利用正反馈原理和某种启发式算法的有机结合，容易出现早熟现象以及陷入局部最优解。混合的思路是让蚂蚁也具有“粒子”的特性，首先蚂蚁

按照蚁群算法，完成一次遍历后，再让蚂蚁根据局部最优解和全局最优解进行调整。调整思路如下：对于旅行商问题，其当前的位置是基本路径，若按基本粒子群算法，其速度难于表达。故采用这里采用遗传算法的思想来解决，(5) 式中的  $c_0 \mathbf{v}_k$  项可以看作遗传算法的变异操作，(5) 式中的  $c_1 (\mathbf{pbest}_k - \mathbf{x}_k) + c_2 (\mathbf{gbest}_k - \mathbf{x}_k)$  项可以看作遗传算法的交叉操作，让当前解与个体极值和全局极值分别作交叉操作，产生的解为新的位置。

#### 3.2 变异操作 (Mutation Operation)

由路径  $C_0$  变异到另一条路径  $C_1$ ，常用的有以下几种策略，这里假设有  $n$  个城市。

**变异策略A** 在第  $1 \sim n$  个访问的城市中随机地选取第  $j_1$  次和第  $j_2$  次访问的城市，在路径  $C_0$  中交换第  $j_1$  次和第  $j_2$  次访问的城市，其余不变，此时的路径为  $C_1$ 。比如  $C_0=2 3 4 1 5 7 9 8 6$ ， $j_1=2$ （第2次访问的城市是城市3）， $j_2=7$ （第7次访问的城市是城市9），则  $C_1=2 9 4 1 5 7 3 8 6$ 。

**变异策略B** 在第  $1 \sim n$  个访问的城市中随机地选取第  $j_1$  次访问的城市，在路径  $C_0$  中交换第  $j_1$  次和第  $j_1+1$  次访问的城市，其余不变，此时的路径为  $C_1$ 。比如  $C_0=2 3 4 1 5 7 9 8 6$ ， $j_1=2$ ，则  $C_1=4 3 1 5 7 9 8 6$ 。

**变异策略C** 也称逆转策略，在第  $1 \sim n$  个访问的城市中随机地选取第  $j_1$  次和第  $j_2$  次访问的城市，在路径  $C_0$  中第  $j_1$  次到第  $j_2$  次访问的城市之间的子路径以反方向插入，其余不变，此时的路径为  $C_1$ 。比如  $C_0=2 3 4 1 5 7 9 8 6$ ， $j_1=2$ ， $j_2=7$ ，则  $C_1=2 9 4 7 5 1 3 8 6$ 。

**变异策略D** 在第  $1 \sim n$  个访问的城市中随机地选取第  $j_1$  次和第  $j_2$  次访问的城市，假设  $j_1 < j_2$ ，在路径  $C_0$  中将第  $j_1$  次访问的城市安排到第  $j_2$  次访问的城市之后，其余不变，此时的路径为  $C_1$ 。比如  $C_0=2 3 4 1 5 7 9 8 6$ ， $j_1=2$ ， $j_2=7$ ，则  $C_1=4 1 5 7 9 3 8 6$ 。

#### 3.3 交叉操作 (Across Operation)

交叉的方法很多，下面几种方法最常用：

**交叉策略A** 在第二个串中随机选择一个交叉区域；将 old2 的交叉区域加到 old1 前面（或后面），删除 old1 中已在 old2 的交叉区中出现过的城市。例如两父串为：old1=1 2 3 4 5 6 7 8 9，old2=9 8 7

$|6\ 5\ 4\ 3|$  2 1, 若交叉区域为: 6 5 4 3, 交叉后为: newl=6 5 4 3 1 2 7 8 9。

**交叉策略B** 在第二个串中随机选择一个交叉区域; 将old2的交叉区域加到old1对应的位置, 删除old1中已在old2的交叉区中出现过的城市。例如两父串为: old1=1 2 3 4 5 6 7 8 9, old2=9 8 7 |6 5 4 3| 2 1, 若交叉区域为: 6 5 4 3, 交叉后为: newl=1 2 6 5 4 3 7 8 9。

**交叉策略C** 在第二个串中随机选择一个交叉区域; 将old2的交叉区域加到old1的随机的位置, 删除old1中已在old2的交叉区中出现过的城市。例如两父串为: old1=1 2 3 4 5 6 7 8 9, old2=9 8 7 |6 5 4 3| 2 1, 随机产生城市7, 则加入到城市7后, 若交叉区域为: 6 5 4 3, 交叉后为: newl=1 2 7 6 5 4 3 8 9。

**交叉策略D** 在第二个串中随机选择一个交叉区域, 如交叉区域为: 6 5 4 3; 将old2的交叉区域加到old1的城市6的位置, 删除old1中已在old2的交叉区中出现过的城市。例如两父串为: old1=1 2 3 4 5 6 7 8 9, old2=9 8 7 |6 5 4 3| 2 1, 交叉后为: newl=1 2 6 5 4 3 7 8 9。

### 3.4 混合算法 (Hybrid Algorithm)

解TSP问题的混合的算法如下:

步骤1  $nc \leftarrow 0$ , ( $nc$  为迭代步数或搜索次数), 初始化, 产生大量的路径(如100条), 从中选择比较优的(如30条), 使这些路径留下信息素, 将  $m$  个蚂蚁置于  $n$  个顶点上;

步骤2 根据当前位置计算适应值(各路径的长度), 设置当前适应值为个体极值  $ptbest$ , 当前位置为个体极值位置  $pcbest$ , 根据各个粒子的个体极值  $ptbest$ , 找出全局极值  $gtbest$  和全局极值位置  $gcbest$ ;

步骤3 将各蚂蚁的初始出发点置于当前解集中, 对每个蚂蚁  $k(k=1,2,\dots,m)$ , 按概率  $p_{ij}^k$  移至下一顶点  $j$ , 将顶点  $j$  置于当前解集;

步骤4 对每个蚂蚁进行如下操作, 第  $j$  个蚂蚁路径  $C_0(j)$  与  $gcbest$  交叉得到  $C'_1(j)$ ,  $C'_1(j)$  与  $pcbest$  交叉得到  $C''_1(j)$ ,  $C''_1(j)$  以一定概率变异得到  $C_1(j)$ , 根据当前位置计算路径长度, 若新的目标函数变好, 接受新值, 否则就拒绝, 第  $j$  个粒子路径  $C_1(j)$  仍然为  $C_0(j)$ , 重新找出各个蚂蚁子的个体极值  $ptbest$  和极值位置  $pcbest$ , 找出全局极值  $gtbest$  和全局极值位置  $gcbest$ ;

步骤5 计算各蚂蚁的路径的总长度  $L_k(k=1,2,\dots,m)$ , 记录当前的最好解;

步骤6 对路径长度  $L_k$  小于给定值的路径, 按更新方程(2)修改轨迹强度;

步骤7  $nc \leftarrow nc + 1$ ;

步骤8 若  $nc <$  预定的迭代次数且无退化行为(即找到的都是相同解)则转步骤2;

步骤9 输出目前最好解。

## 4 仿真结果及分析(Simulation Results and Analysis)

为了检验算法的有效性, 分别采用模拟退火算法、遗传算法和混合算法来解决30个城市的TSP问题, 数据来源于文献[10]。模拟退火算法采用的算法, 起始温度  $T = 100000$ , 终止温度  $T_0 = 1$ , 退火速度  $\alpha = 0.99$ ; 遗传算法程序采用MATLAB的遗传算法工具箱<sup>[12]</sup>, 参数如下: 染色体个数  $N = 30$ , 交叉概率  $P_c = 0.2$ , 变异概率  $P_m = 0.5$ , 迭代次数 100; 混合算法参数:  $\alpha = 1.5$ ,  $m = 30$ ,  $\beta = 2$ ,  $\rho = 0.9$ , 混合算法分别采用4种交叉策略和4种变异策略组合起来16种方法进行比较。对各种算法测试20次, 结果如表1所示。图1是混合算法最好的解, 总路程为 423.7406。

表1 几种算法测试结果

算法	平均值	最好解	最差解
模拟退火算法	438.5223	424.6918	479.8312
遗传算法	483.4572	467.6844	502.5742
基本蚁群算法	550.0346	491.9581	599.9331
交叉策略A+变异策略A	439.4948	425.6490	456.7721
交叉策略A+变异策略B	441.9257	428.7296	455.2382
交叉策略A+变异策略C	437.0028	426.6002	446.2394
交叉策略A+变异策略D	438.7750	425.4752	455.2929
交叉策略B+变异策略A	438.9350	424.6354	457.9062
交叉策略B+变异策略B	431.4987	423.7406	447.6865
交叉策略B+变异策略C	435.4220	424.9003	447.3223
交叉策略B+变异策略D	439.4777	426.1972	465.9935
交叉策略C+变异策略A	444.1723	429.3803	459.4925
交叉策略C+变异策略B	438.5871	426.3076	455.5854
交叉策略C+变异策略C	440.4201	427.6016	454.8674
交叉策略C+变异策略D	439.5524	424.4643	461.7948
交叉策略D+变异策略A	439.0477	424.6727	451.8001
交叉策略D+变异策略B	436.0081	423.7406	460.6230
交叉策略D+变异策略C	438.8091	425.8201	455.4830
交叉策略D+变异策略D	436.4577	423.9490	457.3155

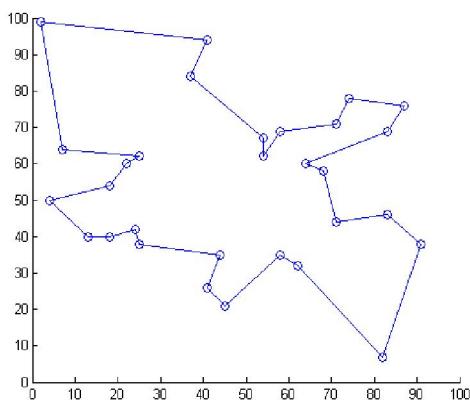


图1 用混合算法解30个城市的TSP最好的解

从表1可以看出，混合粒子群算法的16种组合大都比较好，特别交叉策略B和变异策略B的混合粒子群算法是最好的算法。交叉策略B比其他交叉策略效果好，因为其他交叉策略将交叉区域加入到随机的位置，有可能增加了部分差的解。在变异策略中，变异策略B效果比较好，其变异的幅度比较小，避免出现比较差的解。另外这里的粒子群交叉与遗传算法交叉不一样，遗传算法解采用随机策略，虽然保证解分布比较均匀，但个体的质量不能保证，解群中的个体很大一部分远离最优解，这样在经过交叉操作后的个体的性能不能保证比原来的好；而本文采用的交叉操作，是与个体极优和群体极优进行交叉，无疑会改变下一代群体的质量，有助于提高搜索效率。

## 5 结束语 (Conclusion)

利用蚁群算法和粒子群优化算法特性，提出的混合算法可以显著提高计算效率，具有较大的实用价值。蚁群算法和粒子群优化算法研究处于初期，还有许多问题值得研究，如算法的收敛性、理论依

据等。本文用混合算法解决了离散性优化问题，对于连续性优化问题有待于进一步研究。

## 参考文献 (References)

- [1] Colomi A, Dorigo M, Maniezzo V. An investigation of some properties of an ant algorithm[A]. Proc. Of the Parallel Problem Solving from Nature Conference (PPSN'92)[C]. Brussels, Belgium: Elsevier Publishing, 1992: 509-520.
- [2] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36 (10): 1240-1245.
- [3] 马良, 项培军. 蚁群算法在组合优化中的应用[J]. 管理科学学报, 2001, 4 (2): 32-37.
- [4] Gunes M, Sorges U, Bouazizi I. ARA the ant colony based routing algorithm for MANETs[A]. Proceedings International Conference on Parallel Processing Workshops[C]. Uuncouver, B C, Canada, 2002:79-85.
- [5] Lumer E, Faleta B. Diversity and adaptation in populations of clustering ants[A]. Proc of the 3 Conf On Simulation of Adaptive Behavior[C]. MIT Press, 1994:499-508.
- [6] Parpinelli R S, Lopes H S, Freitas. Data mining with an Ant Colony optimization algorithm[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4):321-332.
- [7] Eberhart R. C. , Kennedy J.. A new optimizer using particles swarm theory[C]. Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995:39-43.
- [8] Shi Y. H. , Eberhart R. C.. A modified particle swarm optimizer[c]. IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998:69-73.
- [9] 李爱国, 覃征, 鲍复民等. 粒子群优化算法[J]. 计算机工程与应用, 2002, 38 (21) :1-3.
- [10] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001: 195-211.
- [11] 康立山, 谢云, 尤矢勇等. 模拟退火算法[M]. 北京: 科学出版社, 1994: 150-151.
- [12] 高尚. 基于 MATLAB 遗传算法优化工具箱的优化计算 [J]. 微型电脑应用, 2002,18(8):52-54.