

# 网络拓扑进化算法

高永超,李歧强

GAO Yong-chao, LI Qi-qiang

山东大学 控制科学与工程学院, 济南 250061

Department of Control and Engineering, Shandong University, Ji'nan 250061, China

GAO Yong-chao, LI Qi-qiang. Network topology evolving optimization. Computer Engineering and Applications, 2007, 43(27): 91-94.

**Abstract:** Based on the model of topology evolving networks describing evolution, we designed Network Topology Evolving Optimization (NTEO), which can automatically control the diversity of populations. The structure of topology evolving makes the population scale of NTEO variable, and the new solutions in the process of evolving enhance the exploration of problem space. The same parts of solutions may directly denote the relations of units (solutions) of evolving networks with the introduction of backbone of solutions of combinatorial optimizations. Coding continues variables into binary and denoting the same parts of solutions using binary bits with the same values, we may optimize continues problems by NTEO. Moreover, NTEO does not fix the mutation mechanism of solutions, so it may be combined with others various technologies and has broad applicability. The experiments show that NTEO has better searching performance.

**Key words:** diversity of population; network topology evolving; backbone of solution; extremal optimization

**摘要:**借助拓扑进化网络描述生物进化的模型,设计出网络拓扑进化算法。算法的拓扑进化结构使其具有可变的种群规模,而进化中产生的新解提高了探测空间的能力。通过在算法中引入组合优化问题解的 backbone 概念,可以用解的相同部分来直观表示进化网络中单元(解)之间的联系。将连续变量转换成二进制编码后,以相同取值的二进制“位”表示解的相同部分,也可以用该算法求解连续函数优化问题。网络拓扑进化算法不规定解的变异方式,可以结合各种现有的技术,具有广泛的应用性。仿真实验表明算法具有较强的空间搜索能力。

**关键词:**种群多样性;网络拓扑进化;解的 backbone;极值优化

**文章编号:**1002-8331(2007)27-0091-04 **文献标识码:**A **中图分类号:**O224

## 1 引言

遗传算法、粒子群算法多数搜索算法在搜索过程中不断繁殖好解并择优保留个体,导致群体多样性不断降低,形成最优解的有效成分缺失,从而使算法失去了探索新空间区域的能力,因而算法易于陷入局部最优。为此,众多研究者提出了一系列基于原始算法的改进算法(包括基于种群<sup>[1,2]</sup>和控制参数优化<sup>[3-5]</sup>等的改进)及与其它优化算法相结合的混合算法,取得了较好的效果,并应用到实际工程中。通过控制种群多样性来提高算法性能的方法,主要集中在分析解之间及与最优解之间的距离<sup>[6-8]</sup>、引进新的操作算子<sup>[9-12]</sup>、对解增加随机扰动<sup>[13,14]</sup>等方面。

在算法运行过程中权衡算法的空间探测和开发能力,有效控制搜索的进行,才能使算法具有较好的搜索性能,提高收敛速度和解的质量。因此要得到高性能的算法,除了改进现有的算法,设计更为合理的搜索机制也有重要的意义。本文以新型的生物进化模型——网络拓扑进化模型为基础,以拓扑进化网络的结构为框架,设计网络拓扑进化算法(Network Topology Evolving Optimization, 简称 NTEO)。算法具有可变的种群规模,用解的 backbone 表示种群中解之间的联系,实现新解的产生和差解的消亡,自动控制种群的多样性,增强搜索能力。

## 2 网络拓扑进化模型

生物系统一直是优化算法机理的重要来源,遗传算法等进化算法是以达尔文的生物进化论为基础,采用复制、变异、交叉等操作来体现适者生存的自然选择进化规则。20世纪80年代,生物进化开始被作为一个自组织系统来研究。Paczuski 认为自然进化是一个自组织的过程,不仅受变异和选择的影响,而且受物种灭亡的影响。在生物进化过程中系统存在间歇的平衡,也有大的物种灭亡事件,如恐龙的灭绝<sup>[15]</sup>。自组织系统<sup>[16]</sup>可以不受外界的约束而组织自己到达临界稳定状态,其模型常常依赖于极值过程,其中幂率分布是自组织过程的重要特征。Bak & Sneppen 以极值过程原则建立了简单的生物进化模型——Bak & Sneppen 模型,简称为 BS 模型<sup>[17]</sup>。Boettcher 将 BS 模型改进后构造了极值优化算法<sup>[18-20]</sup>,幂函数概率选择变异进行变化对搜索过程起关键作用,并决定了参数(幂函数  $P_i \propto k^{-\tau}$  的指数  $\tau$ )具有使搜索性能最好的最优取值<sup>[21]</sup>。

Slanina & Kotrla 认为,实际系统中的单元之间的连接拓扑在进化,并且单元数目也在变,由此建立了网络拓扑进化模型,简称为 SK 模型<sup>[22]</sup>,该模型比 BS 进化模型更能体现生物进化过程中的间歇平衡。SK 模型考虑一个由数目变化的单元连

接成网络的系统,在生物进化中,这些单元是物种。设初始网络共有  $N$  个单元,构成单元集合  $I=\{1,2,\dots,N\}$ ,对应每个单元  $i$  的适应值为  $\lambda_i$ ,其中最小适应值  $\lambda^*=\min\{\lambda_i\}$ ,以  $M(i)$  表示变异函数。单元  $i$  与单元  $j$  之间的联系定义为  $l_{ij}$ ,单元  $i$  的邻居即与单元  $i$  有联系的单元集合  $U_i=\{j|\exists l_{ij},j\in I \& j\neq i\}$ ,新生成单元为  $e$ 。

网络拓扑进化的动态规则如下:

- (1)如果  $\lambda_i=\lambda^*$ ,则  $\lambda'_i=M(i),\lambda'_j=M(j), (j\in U_i)$ ;
- (2)如果  $\lambda_i=\lambda^* \& \lambda'_i>\lambda'_j (j\in U_i)$ ,则  $\lambda_e=random(), U_e=\{j|\exists l_{ij},j\in I \& j\neq i\}, I=e\cup I$ ;
- (3)如果  $\lambda_i=\lambda^* \& \lambda'_i<\lambda'_j (j\in U_i)$ ,则  $I=I-i, U_j=U_j-l_{ij}, (j\in U_i)$ ,若  $U_j=U_j-l_{ij}=\emptyset, (j\in U_i)$ ,则  $I=I-i$ 。

图1遵照上述动态规则描述了网络拓扑的进化过程。在初始网络(a)中,共有8个单元。若适应值最小的单元变异后的适应值高于与其连接的其它单元,则产生一个新单元,新生单元继承其母体单元与其它单元的连接。例如(a)→(b)的进化,适应值最小的单元1变异后的适应值高于与其连接的单元2~8,产生一个新单元9,单元9同时继承了单元1与其它单元2~8之间的连接。当单元变异后的适应值小于它的邻居单元时,该单元灭亡,它与其它单元的连接也消失,如果此时出现孤立单元,则该单元也随之灭亡。

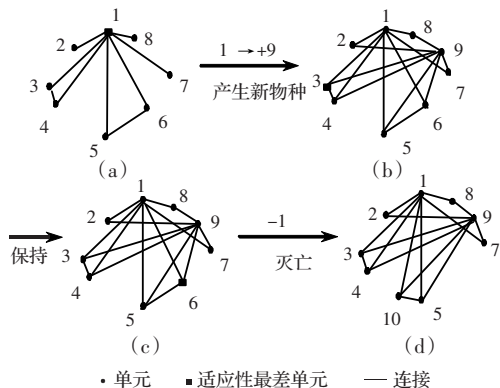


图1 网络拓扑进化示意图

### 3 网络拓扑进化算法

以网络拓扑进化模型的动态规则为基础,设计一种新的优化算法——网络拓扑进化算法(NTEO)。如果将优化问题中的每个候选解视为一个物种,那么  $N$  个候选解组成物种种群  $W$ ,从而构成进化网络,这时每个物种相当于网络中的单元。但是BS模型中拓扑变化的自由度太大,与实际生物进化仍有差距,因此需要寻找某些规则限制网络模型的拓扑,允许单元(物种)被新的代替<sup>[22]</sup>。因此在构造算法的时候,为种群规模设定一个上限,在未达到此限时,保留产生的新解,当达到规模上限时  $N_{up}$ ,用幂函数概率选择一个解用新解替换。解的适应度根据目标函数进行相应定义,由于适应度之间只是进行比较,其单位量级不影响计算,只要保证好解的目标函数适应度高即可。当种群只剩下一个孤立单元时,保留该单元。为了加快算法速度,每次迭代中所有解都发生变异。

为了直观表达解之间的联系,引入解的 backbone 的概念,并将其引申和扩展。下面首先介绍解的 backbone 以及如何产生新解,并分析新解可能到达的搜索范围。然后给出算法的执行步骤,进行分析和实验。

#### 3.1 解的 backbone

backbone 是从统计物理借鉴来的一个概念。简单地讲,

backbone 指所有最优解中相同的部分、结构或模式,如 TSP 问题的最优解中具有相同城市连接的路径部分。Slaney & Walsh<sup>[23]</sup>研究了 backbone 与问题难度、搜索费用的关系,发现可以利用组合优化问题解的 backbone 发展启发式算法。将 backbone 引申为搜索得到的解的相同部分。对于含有连续变量或整数变量的一般优化问题,将变量编码成二进制后比较变量的二进制“位”,把具有相同取值的“位”构成的集合定义为解的 backbone。

#### 3.2 新解的产生

直接求解组合优化问题时,常常将问题的物理量作为变量。如 TSP 问题,把每个城市看作一个变量,定义目标函数为:从一个城市出发经过其它城市且仅经过一次,最后回到起点城市所走过的最短路径。因此 TSP 问题的一个解可用城市号表示为(以 10 城市为例):1-4-3-7-5-9-2-6-8-10-1。假设有 3 个解,其路径连接和路径长度分别为:

$$P_1:1-10-3-8-6-7-9-2-4-5-1, L_1=130$$

$$P_2:1-3-8-2-9-5-4-6-10-7-1, L_2=114$$

$$P_3:1-2-9-10-5-4-8-7-3-6-1, L_3=142$$

这 3 个解的相同连接部分为:3-8,2-9,4-5,从而新解中将连接片段 3-8,2-9,4-5 分别作为一个城市在路径中连接,随机排列 3-8,2-9,4-5 与剩余的城市可以产生一个新解 1-6-3-8-2-9-10-4-5-7-1。

连续变量或整数变量转换成二进制后,几个解中对应的变量的相同“位”保持不变,“位”值不同的以概率置为 0 或 1,当所有变量处理完以后形成一个新解。

#### 3.3 新解的搜索范围

本节分析新解可能到达的搜索范围。仍以 TSP 为例,组合优化问题解的  $n$  个变量中若有  $m$  个相同,则可能产生的新解个数为  $(n-m)!$ ,占解空间总数的  $(n-m)!/n! \times 100\%$ ,表 1、表 2 列出了  $n$  和  $m$  取不同值时的比较。设一个连续变量按要求精度转换成二进制后,二进制“位”数为  $n$ ,不同解中对应的该变量有  $m$  位相同,则可能形成的新解个数为  $2^{n-m}$ ,占解空间总数的  $2^{n-m}/2^n \times 100\%$ 。

表 1 新解搜索范围示例  $n=10$

| $n$ | $n!$    | $m$ | $(n-m)!$ | $\frac{(n-m)!}{n!} \times 100\%$ |
|-----|---------|-----|----------|----------------------------------|
| 10  | 362 880 | 2   | 40 320   | 11.11                            |
|     |         | 3   | 5 040    | 1.39                             |
|     |         | 4   | 720      | 0.20                             |
|     |         | 5   | 120      | 0.03                             |
|     |         | 6   | 24       | -                                |
|     |         | 7   | 6        | -                                |
|     |         | 8   | 2        | -                                |
|     |         | 9   | 1        | -                                |

表 2 新解搜索范围示例  $n=100$

| $n$ | $n!$                      | $m$ | $(n-m)!$                  | $\frac{(n-m)!}{n!} \times 100\%$ |
|-----|---------------------------|-----|---------------------------|----------------------------------|
| 100 | $9.332\ 6 \times 10^{55}$ | 2   | $9.426\ 9 \times 10^{53}$ | 1.01                             |
|     |                           | 3   | $0.096\ 2 \times 10^{53}$ | 0.01                             |
|     |                           | 4   | $0.001\ 0 \times 10^{53}$ | -                                |
|     |                           | 5   | -                         | -                                |

算法前期,解之间相同部分较少,新解随机性较大。当群体收敛到最优解附近时,新解的较大部分与已有解相同,其作用是在最优解域内的小范围内进行细搜索。

### 3.4 算法的执行步骤

(1) 设定种群规模上限  $N_{up}$ , 产生  $N_{up}/2$  个随机解作为初始状态; 比较解并获得解的 backbone (解之间的联系), 具有 backbone 的解构成子群, 记为  $w_1, w_2, \dots, w_i, \dots, w_s, 1 \leq s \leq N_{up}$ 。若所有解存在 backbone, 则所有子群合并为整个种群  $W$ 。

(2) 种群中的所有解进行变异。若某变异后的解差于变异前子群中的其它解, 该解消亡; 好于变异前子群中的其它解, 产生一个新解; 其余解保留为变异后的解, 构成新的种群。若种群变异后产生新解, 而种群规模达到上限, 则在产生新解的个体所在的子群  $w_i$  中, 根据解对应的目标函数值形成等级  $k=1 (\lambda_i^k = \max\{\lambda_{w,i}\}), k=2, \dots, k=N_i (\lambda_i^k = \min\{\lambda_{w,i}\}), N_i$  为该子群解的个数) 的排列, 以幂函数概率  $P(k) \propto k^{-\tau}$  ( $\tau$  为常数) 选择一个解替换为新解。根据幂函数变化趋势, 在此取  $\tau=4$ , 排序在  $k>5$  之后的解被选中的概率就很小了。

(3) 停止准则: 目标函数值无明显改善则停止运行。

### 3.5 算法的收敛性分析

迭代中产生的新解仅与当前群体有关, 而与之之前的各代解无关, 且各解之间的转换概率与时间起点无关, 满足无后效性。因而状态的转换过程可以看作马尔可夫过程, 其过程状态构成时齐马尔可夫链:

$$P_{ij}(s, s+1) = P\{X(s+1)=j | X(s)=i\} = P_{ij}$$

其中  $i, j \in I$  为状态,  $s$  为起始时刻。并且初始分布对 Markov 链极限行为无影响, 所以初始分布可任意。

(1) 在进化过程中, 若某个解一直存在, 该解从某一状态  $i \in I$ , 经过变异的连续作用转变为状态  $j \in I$ 。算子的转移概率为  $P_m$ , 构成相应的随机矩阵  $M = \{m_{ij}\}$ , 则解的状态变化矩阵为:  $R = M \cdot \{r_{ij}\}$ 。  $M$  是随机矩阵, 并且  $P_k = k^{-\tau} \prod_{s \in n} (1 - k_s^{-\tau}) > 0$ , 因此  $R$  是正定的。所以从任一状态出发转移到任一其它状态的概率大于 0, 也包括最优解在内, 如果不将最优解保留下来, 则算法满足“完全”遍历性, 因而不收敛<sup>[24]</sup>。

(2) 在进化过程中若有一个新解产生并被保留, 则该解遵循(1)的状态转移规律。

(3) 在进化过程中某解变异后确定为消亡解, 则对该解的处理自动消失。

因此采用 Elitist 策略, 保留每代的最好解, 以保证状态转移到最优解时不脱离, 从而使算法能以概率 1 收敛到最优解。

## 4 仿真实验

在算法执行步骤(2)中, 解的变异方式可以因问题而异, 在此采用极值优化算法。对组合优化问题 TSP, 定义变量(城市)适应度为  $\lambda_i = 3/(\alpha_i + \beta_i)^{1/3}$ , 表示该城市连接到距其第  $\alpha$  近和第  $\beta$  近的城市, 理想情况下适应度为 1; 对连续函数优化, 将连续变量编码成二进制。在极值优化算法中, 每一个二进制“位”作为一个变量, 定义变量适应度为二进制“位”翻转前后目标函数之差。因为只是通过比较适应度的好坏对变量进行排序, 所以不要求适应度一定为正。如果解的变异采用其它方式, 如采用高斯变异, 则连续变量变异后再编码成二进制比较解是否存在 backbone。

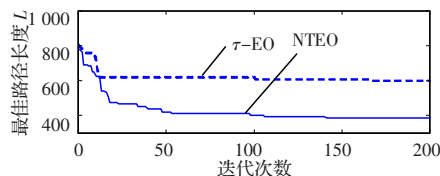
### 4.1 极值优化算法

极值优化(Extremal Optimization, 简称 EO)算法以及  $\tau$ -EO

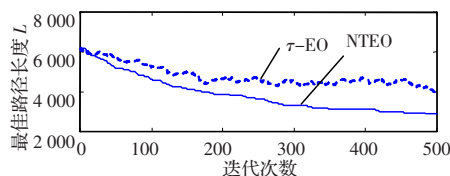
算法是由 Boettcher 等对 BS 进化模型进行改进后提出的。不同于遗传算法等繁殖优选的操作方式,  $\tau$ -EO 算法不断以幂函数概率  $P_k \propto k^{-\tau}$  选择适应性较差的变量进行变异(其中  $k$  为变量按适应度的排序号,  $\tau$  为指定参数), 逐步清除构成解的较差成分, 从而很快地靠近近优解。若参数  $\tau$  趋于无穷大, 则总是选择适应性最差的变量进行变异, 这时  $\tau$ -EO 还原为 EO 算法。 $\tau$ -EO 采用单解迭代且只有一个参数  $\tau$  需要调节, 因而简单易实现, 现已应用到模式识别<sup>[25]</sup>、信号滤波<sup>[26]</sup>等领域, 与模拟退火、遗传算法等相比具有较强的竞争力。特别是对于一般算法难以求解的某些 NP 难问题,  $\tau$ -EO 算法具有较好的性能<sup>[27]</sup>。Sousa & Ramos<sup>[28]</sup>通过将变量编码成二进制, 每一个二进制“位”看作一个变量, 实现了算法到连续函数优化、包含各种类型变量的一般优化问题的扩展。

### 4.2 TSP 问题仿真

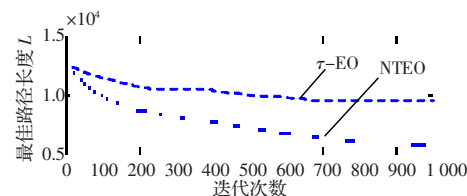
随机产生城市数分别为  $n=2^4, 2^5, 2^6, 2^7, 2^8$  的 TSP 问题, 城市点的坐标值为  $[0, 100]$  间的随机数, 分别用  $\tau$ -EO、NTEO 算法求解。图 2(a)~(c)列出了 16、128 和 256 城市 TSP 问题的最短路径长度随迭代次数变化的典型曲线。从图中可以看出, NTEO 的搜索能力有很大程度的提高。



(a) 16 城市 TSP 问题的求解结果比较



(b) 128 城市 TSP 问题的求解结果比较



(c) 256 城市 TSP 问题的求解结果比较

图 2 TSP 问题的最佳路径长度随迭代次数的变化曲线

### 4.3 连续优化仿真

#### 4.3.1 Schaffe 函数<sup>[29]</sup>

$$\min f(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -10 \leq x_i \leq 10$$

已知上式的全局最优点是(0,0), 最优值为-0.5, 在距全局最优点附近有无限多个局部最优解。由于函数强烈的振荡性以及全局最优解被局部最优解包围的特性, 一般优化算法很难找到它的全局最优解。分别用分组比较遗传算法<sup>[9]</sup>、线性下降粒子群算法<sup>[3]</sup>和 NTEO 算法求解该函数, 得到的最好函数值如表 3 所列。图 3 是迭代中不保留最好解(不采用 Elitist 策略)的 NTEO 算法得到的目标函数值曲线。可以看到, 由于新解扩展了搜索范围, 目标函数也因此呈阶段性改变。但迭代中不保留每代中

的最好解,算法找到最优解后仍会继续搜索,目标函数有振荡。

表3 三种算法求解结果比较

| Algorithms | $f(x)$   |
|------------|----------|
| GC-GA      | -0.481 3 |
| PSO        | -0.5     |
| NTEO       | -0.5     |

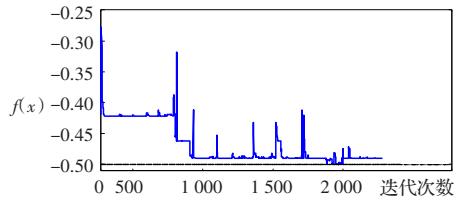
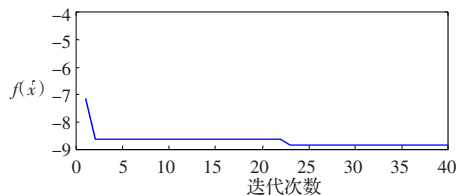


图3 NTEO 求解 Schaffer 函数的最小值曲线  
(不使用 Elitist 策略)

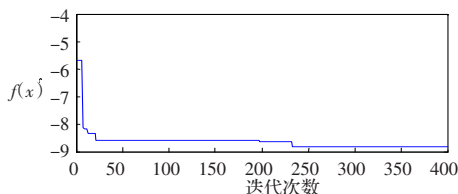
#### 4.3.2 函数极小化问题

$$f(x) = -2\pi x - \sum_{i=0}^{10} \sin(2^{i+1}\pi x), x \in [0, 1]$$

该函数在 $[0, 1]$ 上有很多局部极值,但有唯一全局最小值 $f(0.566\ 713) = -8.817\ 893$ 。采用退火进化规划<sup>[12]</sup>和 NTEO 算法分别求解该函数。退火进化规划算法只包含解的高斯变异操作,以退火概率接受劣解来改善种群多样性。NTEO 在迭代中产生新解,从而扩展了搜索范围,所花费的时间也较多,得到的最好解可达到 $-8.817\ 8$ 。图4给出了用 AEP 和 NTEO 求解到的最小函数值曲线。



(a)退火进化规划的最小函数值曲线



(b)NTEO 的最小函数值曲线

图4 最小函数值曲线

经过实验,NTEO 算法的种群规模上限对搜索结果的影响不大。但是由于要比较解之间的相同部分,种群规模对算法计算量的影响较大,并且找到最优解的时间相比而言要长一些。但对于求解空间结构复杂的优化问题而言,该算法更具有搜索能力上的优势。

## 5 结论

以拓扑进化网络模型为框架的优化算法 NTEO,具有可变的种群规模。通过引入组合优化问题解的 backbone,以解的相同部分表示种群中解之间的联系,在迭代中通过新解的产生和差解的消亡来控制种群的多样性,增强了对空间的探测能力。将连续变量转换成二进制编码后,以相同取值的二进制“位”表示解的相同部分,则 NTEO 除了能求解组合优化问题也可应用于连续函数优化。仿真实验表明了算法的有效性。而且 NTEO

算法不设定解的变异方式,因而可以采用各种现有的技术,具有广泛的应用性和可结合性。此外,对约束优化问题而言,算法迭代过程中产生的新解可能是由不可行解产生的可行解或更接近可行域的解,因而该算法也具有求解约束优化问题的价值,这也是下一步要尝试研究的内容。(收稿日期:2006年12月)

## 参考文献:

- [1] van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 1-15.
- [2] 巩敦卫, 孙晓燕. 变搜索区域多种群遗传算法[J]. 控制理论与应用, 2006, 23(2): 256-260.
- [3] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]//Proceeding of Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1945-1949.
- [4] Zheng Yong-ling, Ma Long-hua, Zhang Li-yan, et al. On the convergence analysis and parameter selection in particle swarm optimization[C]//International Conference on Machine Learning and Cybernetics, 2003, 3(2/5): 1802-1807.
- [5] 刘宜春, 喻寿益. 局部快速微调遗传算法[J]. 计算机学报, 2006, 29(1): 100-105.
- [6] Lovbjerg M, Krink T. Extending particle swarm optimizers with self-organized critically[C]//Proceeding of the IEEE Int'l Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1588-1593.
- [7] Kazemi BAL, Mohan CK. Multi-phase generalization of the particle swarm optimization algorithm[C]//Proc of the IEEE Int'l Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 489-494.
- [8] 赫然, 王永吉, 王青, 等. 一种改进的自适应逃逸微粒群算法及实验分析[J]. 软件学报, 2005, 16(12): 2036-2045.
- [9] 周永华, 毛宗源. 求解约束优化问题的分组比较遗传算法[J]. 华南理工大学学报: 自然科学版, 2003, 31(2): 38-43.
- [10] Zhang WJ, Xie XF. DEPSO: hybrid particle swarm with differential evolution operator[C]//Proc of the IEEE Int'l Conf on Systems, Man and Cybernetics. Washington: IEEE Inc, 2003: 3816-3821.
- [11] 毕惟红, 任红民, 吴庆标. 一种新的遗传算法最优保存策略[J]. 浙江大学学报: 理学版, 2006, 33(1): 32-35.
- [12] 高永超, 李歧强. 退火进化规划算法及其收敛性[J]. 系统仿真学报, 2006, 18(3): 577-580.
- [13] Xie XF, Zhang WJ, Yang ZL. A dissipative particle swarm optimization[C]//Proc of the IEEE Int'l Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1456-1461.
- [14] 郝晋, 石立宝, 周家启. 求解复杂 TSP 问题的随机扰动蚁群算法[J]. 系统工程理论与实践, 2002(9): 88-91.
- [15] Paczuski M, Maslov S, Bak P. Avalanche dynamics in evolution, growth, and depinning models[J]. Phys Rev E, 1996, 53: 414-443.
- [16] Bak P, Tang C, Wiessenfeld K. Self-organized criticality: an explanation of 1/f noise[J]. Physical Review Letters, 1987, 59(4): 381-384.
- [17] Bak P, Sneppen K. Punctuated equilibrium and criticality in a simple model of evolution[J]. Phys Rev Lett, 1993, 71(24): 4083-4086.
- [18] Boettcher S. Extremal optimization: heuristics via coevolutionary avalanches[J]. Computing in Science and Engineering, 2000, 2(6): 75-82.
- [19] Boettcher S, Percus A G. Nature's way of optimizing[J]. Artificial Intelligence, 2000, 119(1/2): 275-286.