

一种新的融合分布估计的蚁群优化算法

许 昌 常会友 徐 俊 衣 杨

(中山大学信息科学与技术学院 广州 510275)

摘 要 提出了一种新的融合分布估计的蚁群优化算法。该算法突破了传统蚁群过早收敛的局限性,且蚁群中的每个蚂蚁具有更全面的学习能力,从而能够有效地解决组合优化问题。仿真实验结果表明该算法的性能优于现有的其它几种蚁群优化算法。

关键词 蚁群优化算法,分布估计,旅行商问题,组合优化问题

中图法分类号 TP273 **文献标识码** A

Novel Ant Colony Optimization Algorithm with Estimation of Distribution

XU Chang CHANG Hui-you XU Jun YI Yang

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China)

Abstract In order to improve the performance of the ant colony optimization algorithm, a new ant colony optimization algorithm with estimation of distribution (ACO-ED) was presented. ACO-ED uses probabilistic model based on estimating the distribution of promising solutions in the search space, and adjusts the state transition rule and the global updating rule. Furthermore, ACO-ED is significantly improved by extending with a local search procedure. We applied ACO-ED to TSP problems and compared it with other ant colony optimization algorithms. Simulation results show that ACO-ED is an effective and efficient way to solve combinatorial optimization problems.

Keywords Ant colony optimization, Estimation of distribution, TSP problem, Combinatorial optimization problem

1 引言

蚂蚁系统(Ant System, AS)是一种模拟蚂蚁群体智能行为的仿生优化算法^[1],具有较强的并发性、鲁棒性、正反馈性等^[2]。自1996年意大利学者 M. Dorigo 等提出蚂蚁系统后,很多学者对它进行了研究并提出了很多改进版本,比如 M. Dorigo 等在基本蚂蚁系统的基础上提出了第一个蚁群系统(Ant Colony System, ACS)^[3];德国学者 T. Stutzle 等提出了最大最小蚁群系统(MAX-MIN ant system, MMAS)^[4];吴庆洪等^[5]提出了一种具有变异特征的蚁群算法。M. Dorigo 等人于1999年提出了蚁群优化算法(Ant Colony Optimization, ACO)^[6],该算法给出了蚁群算法的一个一般性框架,其主要思想是:如果求解的问题能够转换为在一个图中寻找最优路径,那么 ACO 就能够通过寻找满足限定条件的最优路径来求解该问题。ACO 算法已经被成功应用于组合优化、网络路由、函数优化以及其他工程领域,但是这些算法普遍存在过早收敛的缺陷。如何避免解空间过早收敛,提高算法性能,是一个重要的研究方向。

近年来,一些研究者从概率统计学的观点出发,将构造性模型引入进化算法,形成一类基于概率分析的新的进化算法,统称为分布估计算法(Estimation of distribution algorithms, EDAs)^[7]。分布估计算法是进化计算领域新兴起的一类随机

优化算法,是当前国际进化计算领域的研究热点。这些算法通过统计学习的手段建立解空间内个体分布的概率模型,是一种全新的进化模式。这种优化技术通过概率图模型对变量之间的关系进行建模,从而有效地解决了多变量相关的优化问题^[8]。

蚁群算法的基本原理是通过更新全局和局部信息量来选择最优路径。分布估计算法的基本思想是依据当前解空间内个体分布的概率模型来产生新解。本文吸取这两种算法各自的优点,提出一种新的融合分布估计的蚁群优化算法(A Novel Ant Colony Optimization Algorithm with Estimation of Distribution, ACO-ED),用于求解组合优化问题。在提出的新算法中,每只蚂蚁依据两种信息来更新信息量:一种是前面经过的蚂蚁留下的信息素;另一种是蚁群最优解的历史统计信息。提出的新算法突破了传统蚁群算法中过早收敛的局限性,蚁群中的每只蚂蚁具有更全面的学习能力,从而能够有效地解决组合优化问题。TSP 问题是典型的 NP 完全问题,许多算法验证及算法效率测试都以 TSP 问题为基础。本文通过解决 TSP 问题来验证算法的有效性。同时,也选取另外两种具有代表性的蚁群优化算法,即蚁群系统(ACS)、最大最小蚁群系统(MMAS),与提出的 ACO-ED 加以对比,实验结果表明新算法的性能优于 ACS 和 MMAS。

本文第 2 节和第 3 节分别简要地介绍 ACO 和 EDAs;第

到稿日期:2009-03-25 返修日期:2009-06-18

许 昌(1983-),男,博士生,主要研究领域为智能算法、光网络等,E-mail: xuchang_china@gmail.com;常会友(1962-),男,教授,博士生导师,主要研究领域为 ERP、敏捷供应链、工作流等;徐 俊(1982-),男,博士生,主要研究领域为智能算法、ASON 网、工作流等;衣 杨(1967-),女,博士,副教授,硕士生导师,主要研究领域为系统优化与建模、智能优化及软计算。

4 节提出 ACO-ED;第 5 节是仿真实验;最后总结全文。

2 蚁群优化算法

2.1 旅行商问题(TSP)

TSP 问题^[9]可描述为一个带权值的有向完全图 $G=(V, A, \delta)$, 其中 $V=\{1, 2, \dots, n\}$ 是城市集合, $A=\{(r, s): r, s \in V\}$ 是边集合, $\delta(r, s)$ 表示城市 r 与 s 之间的距离, 其中 $(r, s) \in A$ 。 $\forall r, s$, 若 $\delta(r, s)=\delta(s, r)$ 则为对称的 TSP 问题, 否则为不对称的 TSP 问题。若无特别说明, TSP 问题均指对称的 TSP 问题。

TSP 问题中的回路是指顶点序列 $L=x_1, x_2, \dots, x_n, x_1$, 其中 $x_i \neq x_j (i \neq j, 1 \leq i \leq n, 1 \leq j \leq n)$ 。TSP 问题是 NP 完全问题, 其算法复杂度为 $O(n!)$ ^[10], 目标函数为

$$\text{Min}(\sum_{i=1}^{n-1} d(x_i, x_{i+1}) + d(x_n, x_1))$$

2.2 蚁群优化算法(ACO)

蚁群优化算法的基本思想^[3]是:将 m 只蚂蚁随机放于 n 个城市结点;每只蚂蚁通过状态转移规则创建一条回路;在创建路径的过程中,每只蚂蚁通过局部更新规则更新路径上的信息素;当所有蚂蚁都完成路径构造后,通过全局更新规则更新最优路径上的信息素。

(1) 设一只蚂蚁 k , 它的任务是访问所有城市一次且仅一次, 最终回到原点。 $J_k(r)$ 表示待访问城市的集合, 其中 r 为当前城市。蚂蚁 k 依据状态转移规则选择下一个要访问的城市 s 。状态转移规则为

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ \tau(r, u) \cdot \eta(r, u)^\beta \}, \\ \text{if } q \leq q_0 \text{ (exploitation)} \\ S, \text{ otherwise (biased exploration)} \end{cases} \quad (1)$$

其中, $\tau(r, u)$ 表示边 (r, u) 上的信息素; $\eta(r, u) = 1/\delta(r, u)$ 表示边 (r, u) 上的启发因子; β 表示启发因子在蚂蚁选择路径中所起的作用, 是控制可见度相对重要性的参数; q 是 $[0, 1]$ 内的随机数; $q_0 (0 \leq q_0 \leq 1)$ 为初始化参数, q_0 越小, 采用随机概率规则 (random-proportional rule) 选择的机会越大; S 是根据随机概率规则选择的随机变量。随机概率规则为

$$P_k(r, s) = \begin{cases} \frac{\tau(r, u) \cdot \eta(r, u)^\beta}{\sum_{u \in J_k(r)} \tau(r, u) \cdot \eta(r, u)^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

(2) 在创建路径的过程中, 每只蚂蚁通过局部更新规则更新路径上的信息素。局部更新规则为

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \quad (3)$$

其中, $0 < \rho < 1$ 为信息素挥发系数; $\tau_0 = (n * L_m) - 1$, L_m 是根据最近邻居启发 (nearest neighbor heuristic)^[10] 产生的路径长度。

(3) 当所有蚂蚁都完成路径构造后, 通过全局更新规则更新最优路径上的信息素。全局更新规则为

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s) \quad (4)$$

其中,

$$\Delta\tau(r, s) = \begin{cases} (L_{\text{gb}})^{-1}, & \text{if } (r, s) \in \text{globalbest tour} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

这里 $0 < \alpha < 1$ 是信息素衰退参数, L_{gb} 是最优路径长度。

3 分布估计算法(EDAs)

分布估计算法的基本框架^[7]如下。

第一步 从随机产生的解中选取若干优质解作为初始解空间。

第二步 采用概率模型评估解空间, 构建真实的概率分布模型, 并依据该模型产生新解。

第三步 将产生的新解加入原来的解空间, 替换旧的解。

第四步 判断是否满足终止条件。若满足, 则新的解空间即为所求, 否则转至第二步。

分布估计算法中最简单的情况是变量无关的^[11], 其中具有代表性的算法包括 PBIL (Population Based Incremental Learning)^[12], UMDA (Univariate Marginal Distribution Algorithm)^[13] 等。PBIL 是公认最早的分布概率模型^[8]。

在 PBIL 算法中, 表示解空间分布的概率模型是一个概率向量 $p(x) = (p(x_1), p(x_2), \dots, p(x_n))$, 其中 $p(x_i)$ 表示位置 i 上取值为 1 的概率。PBIL 中的概率模型为

$$p_{l+1}(x) = (1 - \alpha) p_l(x) + \alpha \frac{1}{N} \sum_{k=1}^N x_k^l \quad (6)$$

其中, $p_l(x)$ 表示第 l 次解空间的概率向量, $x_1^l, x_2^l, \dots, x_n^l$ 表示选择的 N 个优质解, x_i^l 表示位置 i 上的概率取值, α 表示学习速率。

近年来, 包括 PBIL 在内的分布估计算法已成为算法研究中的一个热点。PBIL 算法作为最简单的 EDA 模型引起了学术界的广泛关注。本文提出的 ACO-ED 也是受到 PBIL 思想的启发。

4 融合分布估计的蚁群优化算法(ACO-ED)

针对现有的蚁群算法存在容易陷入局部最优解的缺点, 把分布估计算法思想引入到蚁群优化算法中, 提出了一种新的融合分布估计的蚁群优化算法(ACO-ED), 它保持了蚁群系统的简单有效性。另外, 在 ACO-ED 的基础上增加了局部搜索策略, 提高了算法性能。

4.1 算法设计

ACO-ED 对蚁群系统的改进主要包括以下 3 个方面。

第一, 在状态转移规则中增加各条边上的概率分布因子。

第二, 全局更新规则中考虑最优路径上各条边的概率分布因子。

第三, 当蚂蚁选择下一个访问城市时, 根据随机概率规则对各条边上的概率分布因子进行评估。

在 ACO-ED 中状态转移规则为

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ p(r, u) \cdot \tau(r, u) \cdot \eta(r, u)^\beta \}, \\ \text{if } q \leq q_0 \text{ (exploitation)} \\ S, \text{ otherwise (biased exploration)} \end{cases} \quad (7)$$

其中, $p(r, u)$ 表示边 (r, u) 被访问的次数。 $p(r, u)$ 的更新规则为

$$p(r, u) \leftarrow p(r, u) + \Delta p(r, u)$$

其中, $\Delta p(r, u) = \begin{cases} 1, & \text{if } p(r, u) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$

相应地, 随机概率规则为

$$P_k(r, s) = \begin{cases} \frac{p(r, u) \cdot \tau(r, u) \cdot \eta(r, u)^\beta}{\sum_{u \in J_k(r)} p(r, u) \cdot \tau(r, u) \cdot \eta(r, u)^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

改进后的状态转移规则将选择距离短、信息素多、概率分

布高的城市。参数 q_0 用来调整开发 (exploitation) 和探索 (biased exploration) 之间的相对重要性。当 $q \leq q_0$ 时, 选择当前转移概率最大的那个城市; 当 $q > q_0$ 时, 则按照随机概率规则选择下一个城市。

全局更新规则: 在 ACO-ED 中只有最优路径上的蚂蚁才能更新信息素, 更新规则同式 (4), 但是 $\Delta\tau(r, s)$ 与蚁群系统不同, 为

$$\Delta\tau(r, s) = \begin{cases} \frac{\rho(r, s)}{n} \cdot \frac{1}{L_{gb}}, & \text{if } (r, s) \in \text{global best tour} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

其中, $\rho(r, s)/n$ 表示最优路径上的随机向量。

4.2 算法步骤和分析

基于上述设计思想, 给出了 ACO-ED 算法的具体流程和步骤, 描述如下。

第一步 初始化, 将蚁群中所有蚂蚁置于一个初始城市。

第二步 每只蚂蚁通过状态转移规则选择下一个城市, 逐步构造出一条回路。在此过程中, 每只蚂蚁通过局部更新规则更新路径上的信息素和概率向量。

第三步 当所有蚂蚁都完成路径构造后, 通过全局更新规则更新最优路径上的信息素和概率向量。

第四步 判断是否满足终止条件。若满足, 则退出, 否则转至第二步。

新算法中, 由于采用的是简单的一阶统计量的概率分布模型, 因此更新概率向量的计算是线性时间复杂度的。算法的主要开销还是在下一个城市的选择上。

在每一次迭代过程中, 路径上的信息素和概率向量不断被更新。随着迭代的进行, 最优路径上的概率分布越来越大, 而高概率分布的路径又有更多的机会被选中。这样反复进行, 最终求得最优解。在整个过程中, 概率分布模型与信息素一起不断地引导蚁群向最优路径集中, 每只蚂蚁具有更全面的学习能力, 避免了解空间过早收敛。

4.3 局部查找策略

解决 TSP 问题的启发算法大体分为两类^[3], 分别是路径构造启发和局部优化启发。路径构造启发通常是从城市集合中随机选择一个初始城市, 然后根据启发规则选择下一个城市, 逐步构造出一条 TSP 回路。而局部优化启发是从一条给定路径出发, 通过启发规则交换边来缩短路径长度, 直到找到局部最优路径为止。最常用的局部优化启发规则是 2-opt 和 3-opt^[14]。相应地, 2-opt 交换两条边, 3-opt 交换 3 条边。本文采用 3-opt 局部搜索策略。

在 ACO-ED 中增加局部搜索策略主要基于两方面的考虑。一方面, 通过局部搜索可以提高算法性能, 尽可能在早期选出高质量的解, 并且把学习机制较准确地引导到最优解方向。另一方面, 通过调整参数, 可以使 ACO-ED 产生的初始路径有利于局部搜索, 进而找到最优路径的解。具有 3-opt 局部搜索的 ACO-ED (ACO-ED-3-opt) 与 ACO-ED 的主要区别仅在于当所有蚂蚁都完成路径构造后通过 3-opt 启发把每只蚂蚁转移引导到局部最短路径。ACO-ED-3-opt 算法的具体流程如下。

第一步 初始化, 将蚁群中所有蚂蚁置于初始城市。

第二步 每只蚂蚁通过状态转移规则选择下一个城市,

逐步构造出一条回路。在此过程中, 每只蚂蚁通过局部更新规则更新路径上的信息素和概率向量。

第三步 当所有蚂蚁都完成路径构造后, 通过 3-opt 启发引导每只蚂蚁转移到局部最短路径, 通过全局更新规则更新最优路径上的信息素和概率向量。

第四步 判断是否满足终止条件。若满足, 则退出, 否则转至第二步。

5 仿真实验

为了充分显示算法的性能, 选取两种具有代表性的蚁群优化算法, 即蚁群系统 (ACS)、最大最小蚂蚁系统 (MMAS), 与提出的 ACO-ED 加以对比。对比的 3 种算法都采用了 3-opt 局部搜索策略, 相应地依次为 ACS-3-opt, MMAS-3-opt, ACO-ED-3-opt。

5.1 参数设置

ACO-ED 主要有 4 个参数, β 是控制可见度相对重要性的参数。 β 值越大, 启发因子在蚂蚁选择路径中所起的作用越大。为了让算法具有全面的学习能力, 设置 $\beta=2$ 。参数 q_0 用来平衡算法的开发和探索能力, 为了让算法具有较好的搜索能力, 即一方面保证算法具有一定的全局探索能力, 在全局范围内探索到较好的路径, 另一方面又能够使得算法在较优路径周围进行精细的局部开采, 设置 $q_0=0.9$ 。 ρ 为信息素挥发系数, 挥发的目的是防止算法陷入局部最小值, 但是挥发系数不能太大, 否则成了随机搜索, 因此挥发系数取较小值, $\rho=0.1$ 。 α 是全局信息素衰退参数, 与挥发系数类似, 设 $\alpha=0.1$ 。当然, 参数的具体取值往往需要进行一些初步实验调整而得, 对于不同的优化问题参数的取值可能也不同。但是我们已经讨论过新算法中每个参数的取值范围、意义和影响, 所以即使对于不同的优化问题也只需在小范围内进行调整。

用于比较的 ACS 和 MMAS 算法的参数按照已有文献的方法设置, 具体参数取值如下: 在 ACS^[3] 中, $\beta=2$, $q_0=0.9$, $\alpha=\rho=0.1$; 在 MMAS^[4] 中, α 表示信息素在选择路径中所起的作用, 取值为 $\alpha=1$, $\beta=2$, $q_0=0.9$, $\rho=0.1$ 。所有的算法中蚂蚁数量均为 25, 循环次数为 100, 分别进行 10 次独立实验。

5.2 实验结果

本文所用的测试用例均来自 TSPLIB: <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html>。算法采用 ANSI C 编写, 在具有 P IV 2.4GHz CPU、1GB 内存、Windows XP 操作系统的机器上运行。

表 1 是 3 种算法的实验结果, 其中 ACS-3-opt 的数据来自文献[3, 15], MMAS-3-opt 的数据来自文献[4]。

表 1 3 种算法的结果比较

Problem	Optimum	ACS-3-opt			MMAS-3-opt			ACO-ED-3-opt		
		best	average	%Error	Best	average	%Error	best	average	%Error
eil51	426	426	426.0	0.00%	426	426.0	0.00%	426	426.0	0.00%
kroA100	21282	21282	21282.0	0.00%	21282	21282.0	0.00%	21282	21282.0	0.00%
d198	15780	15780	15780.1	0.00%	15780	15780.3	0.00%	15780	15781.2	0.00%
lin318	42029	42029	42086.5	0.00%	42029	42069.4	0.00%	42029	42065.6	0.00%
pcb442	50778	50778	50914.7	0.03%	50778	50899.1	0.00%	50778	50902.4	0.00%
at532	27686	27715	27761.4	0.10%	27699	27714.5	0.05%	27695	27718.2	0.03%

图 1 是 3 种算法的相对误差 (%Error) 结果比较, 计算公式为 $\%Error = (\text{best} - \text{optimum}) / \text{optimum}$, 其中, best 为求得的最好解, optimum 为已知最优解。从结果可知, ACO-ED-3-

(下转第 211 页)

随机初始聚类中心 d_6, d_9, d_{22}	D1	85.71%	60%	2
	D2	0%	0%	
	D3	52.63%	100%	
随机初始聚类中心 d_4, d_6, d_8	D1	100%	100%	6
	D2	80%	40%	
	D3	60%	90%	
离簇中心最近的点 d_8, d_{16}, d_{25}	D1	100%	100%	2
	D2	81.82%	90%	
	D3	88.89%	80%	
三个簇的 中心点	D1	100%	100%	2
	D2	83.33%	100%	
	D3	100%	80%	
基于模糊粒度计算的 K-means 算法初始聚类中心	D1	100%	100%	2
	D2	83.33%	100%	
	D3	100%	80%	

实验中,当粒度 $d_\lambda = 0.05 \sim 0.04$ 之间时,得到初始聚类,计算其中心得到优化的 K-means 初始值。由表 1 可知,基于模糊粒度计算的方法得到的 K-means 初始聚类中心,正确率高,迭代次数少。其聚类结果与理想的最优效果基本一致。

结束语 1) 通过实验对比证明,基于模糊粒度计算的初始聚类中心有效地消除了 K-means 算法对于初始输入的敏

(上接第 188 页)

opt 的总体性能要优于 ACS-3-opt 和 MMAS-3-opt。

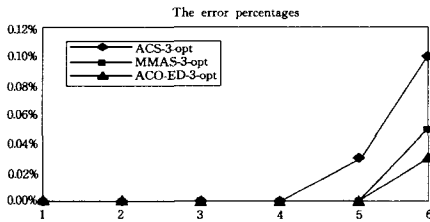


图 1 3 种算法的 Error% 结果比较

图 2 是 ACO-ED-3-opt 求解 pcb442 问题的结果,图中曲线表示每次迭代中的最好解。从图上可以看出解空间没有过早收敛的情况,解的质量较高。

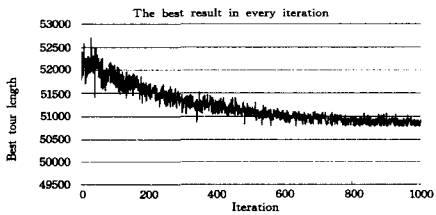


图 2 ACO-ED-3-opt 求解 pcb442 问题的结果

结束语 本文提出了一种新的融合分布估计的蚁群优化算法(ACO-ED),用于求解组合优化问题。ACO-ED 把分布估计算法思想首次引入到蚁群优化算法中,避免了解空间过早收敛的局限性。另外,在 ACO-ED 算法中采用了局部搜索策略,进一步提高了算法性能。新算法具有结构简单、搜索效率高、求解速度快且能够获得高质量的全局近似最优解等优点。仿真实验表明该方法可行性强,性能结果令人满意。

我们进一步的工作是:(1)把 ACO-ED 和局部搜索方法合理结合并应用到其他的组合优化问题中;(2)用高阶概率分布模型来评估优质解的分布情况,并尝试应用到更多难解的优化问题中。

参考文献

[1] Colnani A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies[C]// Proceedings of ECALSI - European Conference on Artificial Life. Elsevier Publishing, 1991; 134-142

感性,提高了算法的稳定性和准确率。2) 本算法利用模糊粒度计算的理论基础,定义归一化距离函数,通过粒度的控制得到优化的初始聚类中心,方法简单,便于在实际应用中操作;并且由文本间归一化距离产生的相似矩阵与文本的属性维数无关,算法的时间复杂度和空间复杂度都较低,算法的实用性得到了保证。

参考文献

- [1] Zadeh L A. The key roles of information granulation and fuzzy logic in human reasoning[C]// Proc of the Fifth IEEE International Conference on Fuzzy Systems, 1996(1); 8-11
- [2] 肖位枢. 模糊数学基础及应用[M]. 北京: 航空工业出版社, 1992; 50-54
- [3] Yao Yiyu. Granular computing: Basic issues and possible solutions[C]// Proc of the 5th Joint Conference on Information Sciences. Atlantic(NJ, USA), 2000; 186-189
- [4] 张铃, 张钊. 问题求解理论及应用(第二版)[M]. 北京: 清华大学出版社, 2007; 66-80
- [5] 张铃, 张钊. 模糊商空间理论(方法)[J]. 软件学报, 2003, 14(4): 770-776
- [2] Dorigo M, Manieszo V, Colnani A. The Ant System: Optimization by a Colony of Cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics - Part B, 1996, 26(1); 29-41
- [3] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[R]. 96-05, INDIA. UniversitC Libre de Bruxelles, 1996
- [4] Stutzle T, Hoos H. Improvements on the Ant System: Introducing MAX-MIN Ant System[M]// Smith G D, Steele N C, eds. Artificial Neural Networks and Genetic Algorithms. 1998; 245-249
- [5] 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
- [6] Dorigo M, Di Caro G. The ant colony optimization meta-heuristic [C]// Corne D, Dorigo M, Glover F, eds. New Ideas in Optimization, London, UK; McGraw-Hill, 1999; 11-32
- [7] Larranaga P, Lozano J A. Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation[M]. Boston: Kluwer Academic Publishers, 2002
- [8] 周树德, 孙增圻. 分布估计算法综述[J]. 自动化学报, 2007, 33(2): 113-124
- [9] Reinelt G. The Traveling Salesman: Computational Solutions for TSP Applications [C] // LNCS. volume 840, Springer Verlag, 1994
- [10] Rosenkrantz D J, Stearns R E, Lewis P M. An analysis of several heuristics for the traveling salesman problem[J]. SIAM J. Comput., 1977, 6; 563-581
- [11] Pelikan M, Goldberg D E, Lobo F. A Survey of Optimization by Building and Using Probabilistic Models[R]. IlliGAL Report No. 99018. Urbana, Illinois; University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 1999
- [12] Baluja S. Population-based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning [R]. CMU-CS-94-163. Pittsburgh, PA: Carnegie Mellon University, 1994
- [13] Mühlenbein H, Paass G. From recombination of genes to the estimation of distributions I. Binary parameters. Parallel Problem Solving from Nature[C]// PPSN IV, Berlin, 1996; 178-187
- [14] Lin S. Computer solutions of the traveling salesman problem [J]. Bell Syst. J., 1965, 44; 2245-2269
- [15] Gambardella L, Dorigo M. Solving Symmetric and Asymmetric TSPs by Ant Colonies[C]// IEEE Conference on Evolutionary Computation (ICEC'96). IEEE Press, 1996