

# 求解 TSP 问题的自逃逸混合离散粒子群算法研究<sup>\*</sup>)

王文峰<sup>1</sup> 刘光远<sup>2</sup> 温万惠<sup>2</sup>

(西南大学计算机与信息科学学院 重庆 400715)<sup>1</sup>(西南大学电子信息工程学院 重庆 400715)<sup>2</sup>

**摘要** 通过对旅行商问题(TSP)局部最优解与个体最优解、群体最优解之间的关系分析,针对 DPSO 算法易早熟和收敛慢的缺点,重新定义了离散粒子群 DPSO 的速度、位置公式,结合生物界中物种在生存密度过大时个体会自动分散迁徙的特性和局部搜索算法(SEC)后,提出了一种新的自逃逸混合离散粒子群算法(SEHDPSO)。自逃逸思想是一种确定性变异操作,能使算法中陷入局部极小区域的粒子通过自逃逸行为进行全局寻优,从而克服算法易早熟的缺陷。仿真结果表明,SEHDPSO 算法比混合蚁群算法(ACS+2-OPT)具有更好的收敛性和搜索效率。

**关键词** 离散粒子群算法,旅行商问题,自逃逸

## Study of a Self-Escape Hybrid Discrete Particle Swarm Optimization for TSP

WANG Wen-Feng<sup>1</sup> LIU Guang-Yuan<sup>2</sup> WEN Wan-Hui<sup>2</sup>

(Faculty of Computer & Information Science, Southwest University, Chongqing 400715)<sup>1</sup>

(Faculty of Electronic Information Engineering, Southwest University, Chongqing 400715)<sup>2</sup>

**Abstract** To deal with the problem of premature convergence and slow search speed, a new algorithm which named the discrete particle swarm optimization algorithm (DPSO) has been proposed based on redefining speed and position of the DPSO, for solving the symmetrical traveling salesman problem (TSP) in this paper. We change the algorithm to self-escape hybrid discrete particle swarm optimization (SEHDPSO) after combining a strategy called self-escape method and local search method. The SEHDPSO uses to explore the global minima thoroughly, which derives from the phenomena that some organisms can escape dynamically from the original cradle when they find the survival density is too high to live. The subsequent experiment result shows that the SEHDPSO can not only speed up the convergence significantly but also solve the premature problem effectively.

**Keywords** Discrete particle swarm optimization algorithm, Traveling salesman problem, Self-escape

## 1 引言

基本粒子群优化算法 PSO (particle swarm optimization) 是由 Kennedy 等于 1995 年提出的<sup>[1]</sup>。PSO 中每个粒子由一个多维向量表示,其下一代粒子的飞行方向和速度由个体最优解和群体最优解向量来修正,PSO 已成功应用于求解连续域问题。为了解决诸多实际工程中的离散问题,Kennedy 等于 1997 年又提出了二进制粒子群算法 BPSO (binary particle swarm optimization)<sup>[2]</sup>。与连续 PSO 相比,BPSO 算法不仅在编码方式上有所改变,更重要的是算法的速度向量不再是位置变化率,而是微粒位置改变的概率。BPSO 适合解决离散问题,但其至今也没有得到很好的推广。文[3]引进了交换子和交换序的概念,对 PSO 进行了修改,改进后的算法在求解 14 个城市的 TSP 问题是可行的,但效率不高。

文中将基本 PSO 中的速度、位置公式重新定义,并引入 S&C<sup>[4]</sup>的数据结构,提出了一种离散粒子群算法 DPSO (discrete particle swarm optimization)。DPSO 更适宜解决离散问题,并且简单、易实现、不需要调整众多参数。但其也存在易早熟,收敛慢的缺陷。

用 DPSO 求解 TSP 问题时,通过对局部最优解、个体最优解和群体最优解中边之间关系的分析,发现局部最优解中有 80% 左右的边和个体最优解、群体最优解的边相同。这就为判断粒子是否陷入局部区域提供了一种思路。结合自然界

中种群密度过大时,个体自动寻找栖息地的习性,提出了一种自逃逸思想:从候选边集合中吸收“新边”,给陷入局部区域的粒子一个变异,使其跳出局部区域。自逃逸思想提高了 DPSO 算法的全局搜索能力,成功地克服了早熟的缺陷。算法中候选边集合由 5-最邻近边法<sup>[5]</sup>生成,它由每个顶点的 5 个最邻近的边所组成。候选边集合几乎囊括了 TSPLIB<sup>[7]</sup>公布的最优解的所有边,而其规模只有最优边数量的三倍左右<sup>[5]</sup>,从而使获得“好边”(公布的最优解中的边)的概率大大增加。另外,DPSO 还采用了 TSP 局部搜索算法 SEC<sup>[4,8]</sup>中的有用机制,大大加快了算法的收敛速度,克服了 DPSO 收敛慢的缺陷,并进一步提高了解的精度。

这种采用自逃逸思想和 SEC 中有用机制的 DPSO 算法被称为 SEHDPSO (self-escape hybrid discrete particle swarm optimization)。仿真实验表明,SEHDPSO 比混合蚁群算法(ACS+2-OPT)具有更好的效率和收敛速度,尤其在较大规模的实例上更具优势。

## 2 离散粒子群优化算法的定义及性能分析

### 2.1 TSP 问题及 S&C 结构

TSP 问题的图论描述如下:给定完全无向图  $G = (V; A)$ ,其中  $V$  为顶点集, $A$  为各顶点相互连接组成的弧集,已知各顶点间连接距离,要求确定一条长度最短的 Hamilton 回路  $T$ ,即遍历所有顶点一次且仅一次的最短距离。它可描述为:

<sup>\*</sup>) 教育部项目(104262);重庆市科技计划项目(CSTC-2006BB2328);西南大学校基金(SWNUQ2005005)资助。王文峰 硕士,主要从事智能算法研究;刘光远 教授,博士后,主要研究方向为计算智能和情感计算等;温万惠 硕士,从事智能算法研究。

$$C(T) = \sum_T c_{ij} \quad (i, j = 1, 2, \dots, n) \quad (1)$$

式中,  $c_{ij}$  表示顶点  $i, j$  的距离。一个可行解就是一个 Hamilton 回路, 也是一个有序的边集合。

在求解 TSP 问题时, 一个可行解在转换成另一个可行解的过程中, 涉及到边的增加、删除、子路径的翻转等操作。为了便于在仿真中实现这些操作, 本文采用一种被称为 S&C (stem and cycle) 的数据结构, 如图 1(a) 所示, 它由一段子路径 stem = (vt, ..., vr) 和一个环 cycle = (vr, sl, ..., s2, vr) 两部分组成。利用 S&C 结构可以方便的实现以下操作:

(1) 边的增加、删除。例如: 增加边 (1, 8), 删除边 (7, 8), 图 1(a) 变成图 1(b); 增加边 (1, 4), 删除边 (3, 4), 图 1(a) 变成图 1(c); 图 1(b) 和图 1(c) 仍然是 S&C 结构。

(2) Hamilton 回路和 S&C 结构的相互转换。例如: 删除 Hamilton 回路 (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1) 中的边 (1, 10), 增加边 (5, 10), 就转换成图 1(a) 所示的 S&C 结构; 反过来, 删除图 1(a) 中 S&C 结构的边 (5, 6), 增加边 (1, 6), 就又转换成 Hamilton 回路 (1, 2, 3, 4, 5, 10, 9, 8, 7, 6, 1)。

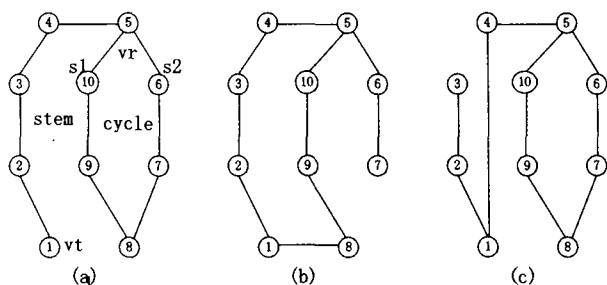


图 1 S&C 的数据结构

## 2.2 离散粒子群算法 DPSO 的定义及性能分析

具有多个可行解的集合称为一个种群, 种群中每个可行解称为一个粒子, 粒子的个数称为种群规模。  $P_i$  表示第  $i$  个粒子的位置, 它是一个有序的边集合。用  $V_i$  表示第  $i$  个粒子的速度,  $P_{ibest}$  表示第  $i$  个粒子所经历的最好位置,  $P_{gbest}$  表示粒子群体所经历过的最佳位置。离散粒子群算法的速度和位置公式可重新定义为:

$$v_i^{t+1} = c_1 * v_i^t + c_2 * (P_{ibest}^t - P_i^t) + c_3 * (P_{gbest}^t - P_i^t) \quad (2)$$

$$P_i^{t+1} = P_i^t \oplus v_i^{t+1} \quad (3)$$

式(2)中  $(P_{ibest}^t - P_i^t)$  表示边集合  $P_{ibest}^t, P_i^t$  的差集; “+”表示两个边集合的交集; “\*”表示依概率从边集中选取边。  $P_i^t \oplus v_i^{t+1}$  表示把  $v_i^{t+1}$  中的边添加到粒子  $P_i^t$  中。  $c_1, c_2$  和  $c_3$  是学习参数, 其中  $c_1 + c_2 + c_3 = 1$ 。式(2)、(3)可简述为: 根据  $c_1, c_2, c_3$  的大小, 依概率在三个边集合中选择一些边, 组成新的边集合  $v_i^{t+1}$ , 把  $v_i^{t+1}$  中的边添加到粒子  $P_i^t$  中, 并利用 S&C 结构调整成为一个可行解  $P_i^{t+1}$ 。

重新定义的 DPSO 比 PSO 更适合求解离散问题。它也继承了 PSO 的本质特性: 利用本身、个体最优解和全局最优解三个信息, 来指导粒子下一步迭代位置。式(2)中的第 1 部分称为动量部分, 表示粒子对当前自身运动状态的信任, 为粒子提供了一个必要动量, 使其依据自身速度进行惯性运动。第 2 部分称为认知部分, 代表了粒子自身的思考行为, 鼓励粒子飞向自身曾经发现的最佳位置。第 3 部分称为社会部分, 表示粒子间的信息共享与相互合作, 它引导粒子飞向粒子群中的最佳位置。

文[6]的理论结果显示, PSO 算法并不能保证收敛到全

局最优, 而仅仅是收敛到种群最好点。由式(2)、(3)可以看出, 粒子  $P_i^t$  从  $P_{ibest}^t, P_{gbest}^t$  中吸收“好边”, 随着迭代次数的增加,  $P_i^t$  将逼近  $P_{ibest}^t, P_{gbest}^t$ 。取 TSPLIB 中的实例 eil51、kroall50、tsp225、pcb442 为实验样本, 用  $\rho$  表示边集合  $P_i^t, P_{ibest}^t, P_{gbest}^t$  的共同边所占的比例。用 DPSO 算法运行 100 次的实验得到的  $\rho$  分别为: 79.1%, 83.8%, 78.4%, 80.3%。这些结果有一个共同的特性是, 局部最优解中有 80% 左右的边是  $P_{ibest}^t, P_{gbest}^t$  中的边。于是有如下简单的结论: 当陷入极小区域时, 粒子  $P_i^t$  中有五分之四左右的边与  $P_{ibest}^t, P_{gbest}^t$  中的边相同。这一结论为判断粒子是否陷入极小区域提供了方法, 可描述如下:

$$|L_i| = |E(P_i) \cap E(P_{ibest}) \cap E(P_{gbest})| \quad (4)$$

$$l = (1/m - 1) \sum_{i=1}^m |L_i| \quad (5)$$

这里,  $E(P_i)$  表示  $P_i$  的边集合;  $|E(P_i)|$  表示边集合  $E(P_i)$  中边的数目;  $|L_i|$  表示边集合  $P_i, P_{ibest}, P_{gbest}$  的交集中边的数目。  $l$  表示群体中所有不同边的平均数, 其中  $m$  是群体规模。当  $|L_i| > l$  时, 粒子  $P_i$  陷入局部极小区域。

## 3 自逃逸混合离散粒子群算法

### 3.1 自逃逸的基本思想

粒子怎样有效地跳出局部区域是算法的关键, 这一点可以从物种的进化过程中得到一些启示。在自然界中, 当某一物种种群密度过于拥挤时, 一些动物, 例如: 蚂蚁, 蜜蜂等, 就会分开并寻找新的栖息地繁殖下一代。相对应到本文算法中的粒子, 当它陷入局部区域时, 它会自动逃离并寻找新的地方, 同时它还记着群体中的目前最好的位置。这就是自逃逸思想, 可描述为:

$$\text{if } |L_i| > l \text{ then } V = c_4 * M, P_i = P_{gbest} + V \quad (6)$$

这里,  $c_4$  是一个常数,  $M$  是 5-最邻近边法生成的候选边集合。上式可以简述为: 对于任意一个粒子  $P_i$ , 如果  $|L_i| > l$ , 粒子  $P_i$  就会被一个新的粒子所替代。这个新的粒子是这样生成的: 从  $M$  中任意选择  $c_4 * M$  条边, 增加这些边到中  $P_{gbest}$ , 利用 S&C 结构, 调整成合法粒子来替代  $P_i$ 。

### 3.2 自逃逸混合离散粒子群算法的实现

- step1 初始化群体、参数;
- step2 产生  $P_{ibest}$  和  $P_{gbest}$ ;
- step3 计算  $|L_i|, l$ ;
- step4 如果  $|L_i| > l$ , 运用自逃逸方法生成新的粒子, 替代原来的粒子  $P_i^{t+1}$ ;
- step5 根据公式(2)、(3)分别计算粒子的速度、新的位置;
- step6 用局部算法 SEC 计算;
- step7 计算每个粒子的适应度;
- step8 更新  $P_{ibest}$  和  $P_{gbest}$ ;
- step9 如果终止条件不满足, 返回 step3。

## 4 实验结果及分析

测试实例来自 TSP 标准测试集<sup>[7]</sup>, 实验环境是 (Pentium IV 2GHz CPU, Win2000, Matlab6.5)。

图 2 图示了 SEHDPSO 算法在实例 eil51 上的一个完整搜索过程。从图中可以看出, SEHDPSO 算法经过两次逃逸就发现了 TSPLIB 公布的最优解。

(下转第 195 页)

实验结果显示,对于含缺省属性值少的数据集,矩阵规则提取算法能获得更高的规则匹配率。对于含缺省属性值多的数据集,将来要考虑采用合适的规则推理策略,来提高规则的适应能力。

**结束语** 本文提出了一种新的基于粗糙集的从不完备信息系统获取规则的矩阵算法。该算法具有以下优点:1)不改变初始不完备信息系统结构;2)获取的规则不受缺省值的影响。实验结果表明,所获得的规则简洁,规则集的规模小,具有较好的可理解性和较强的泛化能力。进一步的工作,研究限制非对称相似关系模型下规则的增量获取以及如何定义规则的可信度、覆盖度来解决规则冲突消解问题。

**参考文献**

1 Komorowski J. Öhrn A. Skowron A. The ROSETTA Rough Set Software System; In: Handbook of Data Mining and Knowledge

Discovery, London: Oxford University Press, 2002  
 2 Clark P, Niblett T. The CN2 induction algorithm. Machine learning, 1989(3): 261~283  
 3 Grzymala-Busse J W. On the unknown attribute values in learning from examples, proc of the ISMIS-91. In: 6<sup>th</sup> International symposium on Methodologies for Intelligent Systems, 1991, Lecture Notes in Artificial Intelligence, vol 542, Springer-Verlag, Berlin Heidelberg New York, 1991. 368~377  
 4 Stefanowski J, Tsoukias A. Valued Tolerance and Decision Rules. In: W. Ziarko, Y. Yao, eds, Rough Sets and Current Trends in Computing, Berlin: Springer, 2002. 271~278  
 5 Kryszkiewicz M. Rough set approach to incomplete information systems. Information Sciences, 1998, 112(1-4): 39~49  
 6 王国胤著. Rough 集理论与知识获取. 西安:西安交通大学出版社, 2001

(上接第 144 页)

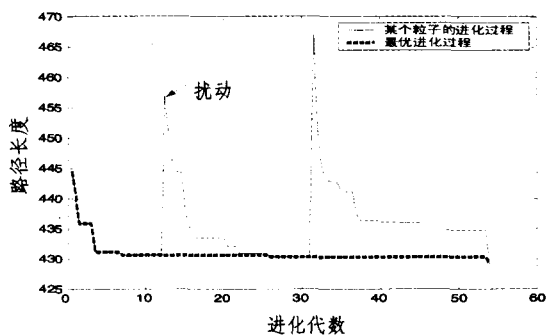


图 2 一个完整的搜索过程

表 1 算法 ACS+2-OPT 和 SEHDPSO 的比较 (各运行 10 次)

实例	ACS+2-OPT			SEHDPSO		
	平均误差 %	最小误差 %	CPU 时间 (s)	平均误差 %	最小误差 %	CPU 时间 (s)
Eil51	0.00	0.00	0.00	1	0.00	0.00
Berlin52	0.00	0.00	1	0.00	0.00	3.2
St70	0.40	0.00	26	0.00	0.00	6.4
Eil76	0.00	0.00	3	0.00	0.00	9.8
Pr76	0.00	0.00	3	0.05	0.02	8.7
Kroc100	0.00	0.00	41	0.01	0.00	37.5
Krod100	0.06	0.00	19	0.05	0.12	27.2
Krob100	0.23	0.00	55	0.09	0.00	41.8
Rd100	0.77	0.45	9	0.25	0.30	54.1
Eil101	0.64	0.08	28	0.02	0.00	16.3
Lin105	0.00	0.00	9	0.03	0.00	12.4
Pr107	0.40	0.30	15	0.18	0.00	16.4
Pr124	0.05	0.00	9	0.23	0.30	47.5
Bier127	0.48	0.10	136	0.10	0.00	101
Ch130	0.45	0.21	236	0.29	0.11	239
Kroa150	0.01	0.00	661	0.05	0.00	66
Krob150	0.05	0.05	93	0.18	0.10	45.7
U159	0.47	0.19	223	0.00	0.00	153
Krob200	0.35	0.03	1459	0.25	0.05	254
D198	0.61	0.55	229	0.13	0.03	198
Tsp225	0.18	0.10	34	0.08	0.02	168
Tsp225	0.56	0.40	673	0.23	0.00	95
A280	0.86	0.38	1514	0.15	0.03	561
Rd400	1.34	1.18	4581	0.25	0.16	1250
P654	1.10	0.72	13582	0.93	0.41	7004
U724	2.26	1.85	24340	0.64	0.22	2381
平均值	0.42	0.24	1845	0.16	0.07	492

为了检验 SEHDPSO 的有效性,本文的算法和文[9]中的蚁群混合算法(ACS+2-OPT)进行了比较,仿真结果见表 1。

从表 1 中的数据对比来看,在实例规模 51 个城市到 124 个城市之间,算法 SEHDPSO 和算法 ACS+2-OPT 的效果相当。但随着规模的增加,SEHDPSO 显示出明显的优势,尤其在实例 U724 上,它不仅搜索精度远高于 ACS+2-OPT,而且收敛的时间不到它的十分之一。表 1 的最后一行列出了 26 个实例的平均值,从平均误差、最小误差、运行时间来看,SEHDPSO 分别是 ACS+2-OPT 的 35%,29%,26.6%。

**结论** 本文提出的 SEHDPSO 利用自逃逸思想很好地保持了群体的多样性,利用局部搜索算法能加快收敛速度。和混合蚁群算法的比较表明,SEHDPSO 算法是有效的。传统理论分析和实验数据显示,速度公式(2)的系数变化对算法的结果有很大的影响。所以,进一步的工作将放在以下两个方面:(1)调整各系数,协调算法局部搜索和全局搜索。(2)利用种群信息,从候选边集中选择更好的边,找出更有效的逃逸行为,增强算法的全局搜索能力。

**参考文献**

1 Eberhart R, Kennedy J. A New Optimization Using Particles Swarm Theory. Proc Sixth International Symposium on Micro Machine and Human Science Nagiya [C]. Japan: IEEE Service Center, Piscataway, 1995. 39~43  
 2 Kennedy J, Eberhart R. A discrete binary version of the particle swarm optimization [C]. In: Proc. IEEE Int Conf. on Neural Networks. Perth, Australia, 1997. 4104~4108  
 3 WANG KANG-PING. Particle swarm optimization for traveling salesman problem [J]. In: Proceedings of the Second International on Machine Learning and Cybernetics, Xi'an, November 2003. 2~5  
 4 Glover F. Ejection chains, reference structures and alternating path methods for traveling salesman problems [J]. Discrete Applied Mathematics, 1996, 65:223~253  
 5 Helsgaun K. An effective implementation of the Lin-Kernighan traveling salesman heuristic [J]. European Journal of Operational Research, 2000, 126:106~130  
 6 Ozcan E, Mohan CK. Particle swarm optimization: Surfing the waves [J]. In: Proc. of the IEEE Int'l Conf. on Evolutionary Computation. Washington: IEEE Inc, 1999. 1939~1944  
 7 www. iwr. uniheidelberg. de/groups/comopt/software/TSPLIB95/tsp/  
 8 Rego C. Relaxed tours and path ejections for the traveling salesman problem [J]. European journal of Operational Research, 1998, 106: 522~538  
 9 LE Louarn F, Gendreau M. Geni Ants for the Traveling Salesman Problem [J]. Annals of Operations Research, 2004, 121: 187~201  
 10 Croes G A. A method for solving traveling salesman problem [J]. Operations Research, 1958, 6:791~812