

文章编号:1001-9081(2010)02-0458-03

基于自组织优化算法的一类多旅行商问题

李天龙,吕勇哉

(浙江大学 智能系统与控制研究所,杭州 310027)

(yongzailu@yahoo.com)

摘要:多旅行商问题作为旅行商问题的一个扩展,是一个经典的组合优化问题,具有更高的复杂性,也具有更广泛的实际意义。针对每个旅行商允许经过的城市数有上限的多旅行商问题,通过引入虚拟城市把多旅行商问题转化为单旅行商问题,并且应用自组织优化算法进行了求解。虚拟城市局部适值的定义很好地处理了此类问题的能力约束,针对多旅行商问题的实例进行的仿真表明自组织优化算法可以很好地求解此类问题。

关键词:多旅行商问题;虚拟城市;自组织优化算法;局部适值

中图分类号: TP30; TP182 **文献标志码:** A

Constrained multiple traveling salesman problem based on self-organizing optimization algorithm

LI Tian-long, Lü Yong-zai

(Institute of Cyber Systems and Control, Zhejiang University, Hangzhou Zhejiang 310027, China)

Abstract: The Multiple Traveling Salesman Problem (MTSP) is an extension of the traveling salesman problem (TSP). MTSP is a classical combinatorial optimization problem with more complexity and applicability. The MTSP with the up-limit of the cities to be visited for each salesman was studied and the self-organizing algorithm (SOA) was introduced to solve the problem. The MTSP was transformed into a TSP by introducing virtual cities and the local fitness of virtual city was defined to deal with the relevant constraints. The computational results with a number of benchmark problems show that SOA can be effectively applied in solving the proposed MTSP with superior performance.

Key words: Multiple Traveling Salesman Problem (MTSP); virtual city; Self-Organizing Algorithm (SOA); local fitness

0 引言

多旅行商问题(Multiple Traveling Salesman Problem, MTSP)是一个典型的组合优化问题,它是旅行商问题(Traveling Salesman Problem, TSP)的一个扩展。MTSP是指:给定 n 个城市, m ($1 < m < n$)个旅行商从同一个城市(或者不同城市)出发,各自经过一条路径回到出发城市,每个城市有且仅有一个旅行商经过(起点城市除外),且每个旅行商至少经过一个非起点城市,并且要求总的路径最短。MTSP比TSP更为复杂,因为它不仅要决定哪个城市由哪个旅行商经过,而且要决定每个旅行商要经过的城市的次序。MTSP也比TSP更有实际意义,很多实际问题如车辆调度问题(Vehicle Scheduling Problem, VSP)^[1]、热轧批次调度^[2]等都可以转化为MTSP。对于MTSP的求解,启发式算法得到了广泛应用,其中有遗传算法^[2-4]、蚁群算法(Ant Colony Optimization, ACO)^[5]等。有关MTSP建模和求解方法的更多详细内容可以参考综述文献^[6]。

本文主要讨论从同一个城市出发的MTSP。通常为了减少总行程长度,会有 $m-1$ 个旅行商,每个旅行商总是只经过最接近起点城市的 $m-1$ 个城市中的一个就回到了出发城市,剩余的 $n-m+1$ 个城市只由一个旅行商遍历。这种情况不符合实际需要,实际中每个旅行商的能力接近,因此我们给每个旅行商附加了能力约束,即每个旅行商允许经过的城市

数不超过一个上限值^[5]。

自组织算法(Self-Organizing Algorithm, SOA)^[7]是基于极值动力学提出的一种新的优化算法,是极值优化(Extremal Optimization, EO)算法^[8]的一种改进形式。EO算法是一种新颖的、基于局部搜索的启发式优化算法。EO算法收敛速度快、局部搜索能力强,且设计简单、容易实现。Lu等人^[9]从统计物理、生物协同进化、生态系统等方面对EO机理进行了阐述。SOA继承了EO的以上优点,并且具有更大概率的广度搜索和深度搜索能力,是一种更为合理的动态优化过程。

针对旅行商能力带约束的MTSP,本文通过引入虚拟城市把MTSP转化为TSP^[5,10],应用SOA对问题进行求解;通过虚拟城市局部适值的定义,很好地处理了此类MTSP中的约束条件,针对多旅行商问题的实例进行的仿真结果表明SOA可以很好地求解此类MTSP。

1 MTSP描述

假设有 n 个城市,分别记为 $1, 2, \dots, n$,其中点1为起点城市, m 个旅行商,引入 $m-1$ 个虚拟城市,并且把虚拟城市的坐标设为起始点城市的坐标,分别记为 $n+1, n+2, \dots, n+m-1$,则MTSP可以转化为如图1的TSP^[5,10]。

变量定义如下: n 为给定的城市数, m 为旅行商个数, $N = n + m - 1$ 为总城市数, i, j 为城市序号, $i, j \in \{1, 2, \dots, N\}$; C_{ij} 为城市 i 到城市 j 的距离,其中虚拟城市之间以及虚拟城市

收稿日期:2009-08-05;修回日期:2009-09-09。

基金项目:国家创新研究群体科学基金资助项目(NCRGSFC:60721062);国家自然科学基金资助项目(60774081)。

作者简介:李天龙(1983-),男,山东泰安人,硕士研究生,主要研究方向:调度与优化算法;吕勇哉(1937-),男,浙江桐乡人,教授,博士生导师,主要研究方向:系统建模、优化、先进控制。

起点城市之间的距离设为无穷大; L_k 为旅行商经过的城市数 $k \in \{1, 2, \dots, m\}$, L_{\max} 为旅行商允许经过的城市数的上限; $S = \{s_1, s_2, \dots, s_N\}$ 为一条经过所有城市的环路。则 MTSP 的转化为 TSP 后的数学模型为:

$$F(s) = \min \left(\sum_{i=1}^{N-1} C_{s_i, s_{i+1}} + C_{s_N, s_1} \right) \quad (1)$$

$$\text{s. t. } 1 \leq L_k \leq L_{\max}; k \in \{1, 2, \dots, m\} \quad (2)$$

目标函数式(1)表示总的路径之和最短,约束式(2)表示每个旅行商经过的城市数范围。

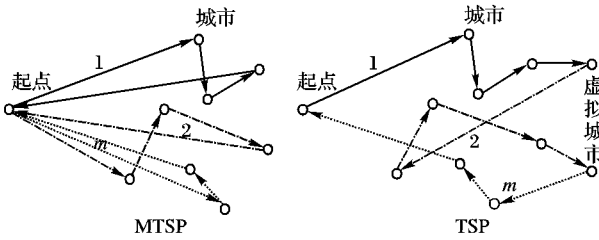


图1 MTSP 转化为 TSP

2 应用 SOA 求解 MTSP

2.1 局部适值函数的定义

首先定义城市序列 S 中城市 s_i 的局部适值函数为 f_i , 由相邻城市距离引起的局部适值为 f_i^1 , 则对于城市 s_1 :

$$f_1^1 = C_{s_1 s_2} + C_{s_1 s_N} - \min_{j \neq 1} C_{s_1 s_j} - \min_{j \neq 1} C_{s_1 s_j}; \quad (3)$$

$j = 2, \dots, N$

对于其他城市:

$$f_i^1 = C_{s_{i-1} s_i} + C_{s_i s_{i+1}} - \min_{j \neq i} C_{s_i s_j} - \min_{j \neq i} C_{s_i s_j}; \quad (4)$$

$i \in \{2, \dots, N\}, j = 1, 2, \dots, N$

其中 $\min_{i \neq j} C_{s_i s_j}$ 和 $\min_{i \neq j} C_{s_i s_j}$ 是城市 s_i 与所有城市之间距离第一小和第二小的值, 则全局目标函数可表示为:

$$\frac{1}{2} \left(\sum_{i=1}^N \min_{i \neq j} C_{ij} + \sum_{i=1}^N \min_{i \neq j} C_{ij} + \sum_{i=1}^N f_i^1 \right) \quad (5)$$

式(5)括号中第一项和第二项均为常数, 因此, 式(5)中局部适值之和的最小即为目标函数值最小, 问题转化为求使得局部适值的和最小的解 S 。

对于约束条件, 通过起点城市和虚拟城市的局部适值来解决, 虚拟城市的坐标均设置为起始点的坐标。起点城市和虚拟城市在城市序列中的序号记为 o_1, o_2, \dots, o_m , 则每个旅行商经过的除起点城市外的城市数为:

$$L_k = \begin{cases} o_{k+1} - o_k - 1, & k \in \{1, 2, \dots, m-1\} \\ N - o_k, & k = m \end{cases} \quad (6)$$

那么因旅行商能力约束引起的局部适值 f_i^2 定义为:

$$f_i^2 = \begin{cases} 0, & i \neq o_k \text{ 或 } (i = o_k \text{ 且 } 1 \leq L_k \leq L_{\max}) \\ \infty, & \text{其他} \end{cases} \quad (7)$$

其中 $i, k \in \{1, 2, \dots, m\}$ 。

城市 s_i 的局部适值函数为:

$$f_i = f_i^1 + f_i^2; i \in \{1, 2, \dots, N\} \quad (8)$$

2.2 求解 MTSP 的 SOA 流程

求解 MTSP 的 SOA 流程如下。

步骤1 随机产生一个初始解 s , 即为包含所有城市的城市序列, 并将 s 设置为当前最优解 s_{best} , 计算 s 的目标函数 $F(s)$, 设置最小目标函数值 $F_{\text{best}} = F(s)$ 。

步骤2 针对当前解 s , 进行如下操作:

1) 根据适值函数定义计算每个城市的局部适值, 按此适

值由大到小排序得到新的城市序列, 按概率 $P(k) \propto k^{-\alpha}, 1 \leq k \leq N$ 选择某个城市;

2) 通过 2-opt 邻域操作改变被选中城市的状态, 产生 $N-1$ 个候选解;

3) 计算所有候选解的全局适应度并由小到大排序, 依概率分布 $P(k) \propto k^{-\beta}, 1 \leq k \leq N-1$ 选择一个新的解 s' ;

4) 用新的解 s' 代替当前解 s , 并比较全局目标函数, 如果 $F(s') < F(s)$, 则设 $F_{\text{best}} = F(s'), s_{\text{best}} = s'$ 。

步骤3 如果满足结束条件, 停止搜索, 输出最优解 s_{best} 及其适值 F_{best} ; 否则转步骤2。

根据大量实验研究表明: $\alpha \approx 1 + 1/\ln(N), \beta \approx 3$ 时, SOA 算法具有良好的优化性能^[7]。

3 仿真结果

采用 Matlab 2008a 仿真平台, 在 Pentium 4, 2.8 GHz, 512 MB 内存的微机进行了仿真实验。针对 76 个城市, 5 个旅行商, 每个旅行商允许经过的城市数上限为 20 时, 图 2 给出了 SOA 在参数 α 的不同取值条件下搜索深度值, β 均取 3, 迭代 10000 次。

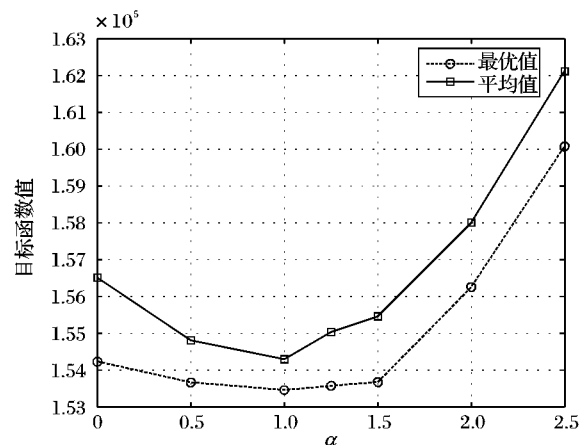


图2 SOA 在参数 α 的不同取值条件下搜索深度值 (pr76)

由图2 搜索深度值可知, 参数 α 取值 1 附近时, SOA 算法具有良好的深度搜索能力, 随着 α 的增大, 算法的搜索能力下降。为了算法的普适性, 本文采用上面提到的经验公式: $\alpha = 1 + 1/\ln(N)$ 。

针对 pr76, 图 3 给出了参数 β 在不同取值 (1, 2, 3, 100) 条件下的 SOA 的动态搜索过程, 100 代表 ∞ , 设置 $\alpha = 1 + 1/\ln(80)$ 。

由图 3 可知, SOA 求解 MTSP 时很快收敛到了局部最优解附近, 说明 SOA 具有快速的深度搜索能力; 曲线在局部最优解附近有很多波动出现, 说明 SOA 可以不断跳出局部最优解, 从而增加了广度搜索能力。随着参数 β 的不断增大, 优化算法在搜索过程中波动幅度逐渐减小; 当 $\beta \rightarrow \infty$ 时, 搜索过程直接收敛于一个局部最优解; 而当 $\beta = 3$ 时, SOA 在广度搜索和深度搜索都有比较好的效果。

SOA 所求得的优化路径如图 4 所示。由图 4 可知, SOA 中起点城市和虚拟城市局部适值的定义很好地解决了此类 MTSP 的约束条件, 并得到了较好的优化解。

我们采用文献[5]中的例子来测试 SOA 的有效性, 并和文献[5]中的结果进行了对比。对于其中的 6 个问题, 均设置参数 $\alpha = 1 + 1/\ln(N), \beta = 3$, 迭代 10000 次, 每个问题均

运行 10 次。优化结果如表 1 所示。由表 1 可知,SOA 能够取得比文献[5]中 ACO 更好的优化结果。

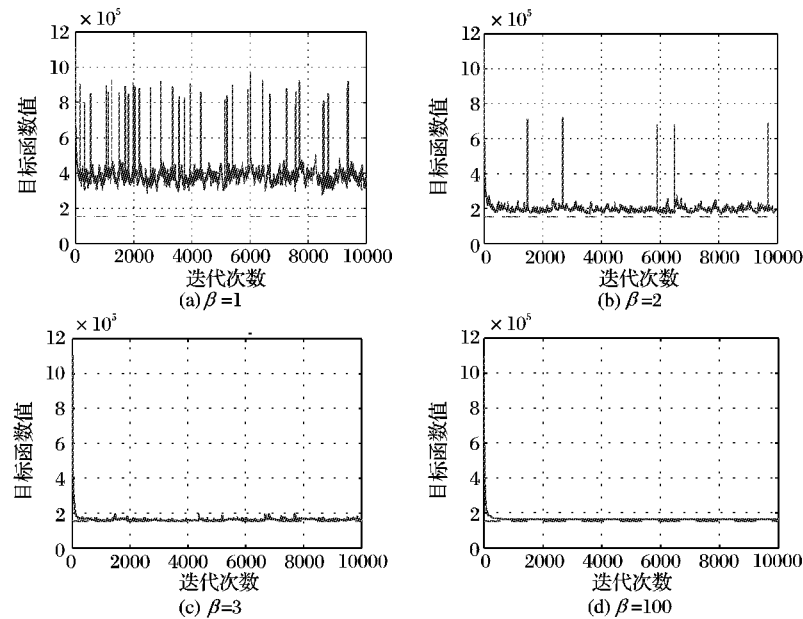


图3 参数 β 在不同取值条件下的 SOA 的动态搜索过程(pr76)

表 1 计算结果

| 问题描述 | | | | ACO ^[5] 计算时间/s | | SOA 计算时间/s | | | | 改进效果/% |
|--------|-------|------|-------|---------------------------|-------|------------|---------|---------|-------|--------|
| 名称 | 城市数目 | 旅行商数 | 城市数上限 | 平均值 | 运行时间 | 最优值 | 平均值 | 最差值 | 运行时间 | |
| pr76 | 76 | 5 | 20 | 180 690 | 43 | 152 378 | 154 522 | 155 801 | 51 | 14.48 |
| pr152 | 152 | 5 | 40 | 136 341 | 91 | 121 562 | 122 659 | 123 956 | 149 | 10.03 |
| pr226 | 226 | 5 | 50 | 170 877 | 165 | 148 780 | 152 233 | 154 714 | 273 | 10.91 |
| pr299 | 299 | 5 | 70 | 83 845 | 288 | 72 133 | 75 032 | 76 192 | 480 | 10.51 |
| pr439 | 439 | 5 | 100 | 165 035 | 563 | 140 129 | 143 900 | 146 754 | 927 | 12.81 |
| pr1002 | 1 002 | 5 | 220 | 387 205 | 2 620 | 338 690 | 340 565 | 342 577 | 4 419 | 12.04 |

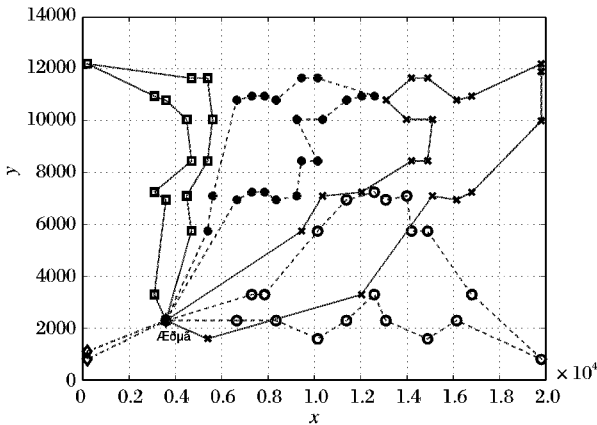


图4 SOA 对 MTSP 的优化结果路径图

4 结语

针对一类旅行商能力带约束的 MTSP,本文通过引入虚拟城市以及虚拟城市局部适值的定义,很好地处理了此类 MTSP 中的约束条件;应用 SOA 对问题进行了求解,仿真结果表明 SOA 可以很好地求解此类 MTSP。此外,对于 MTSP 求解结果的进一步优化,SOA 算法中幂率分布函数的选取,以及自组织优化算法在其他领域的应用都值得深入研究。

参考文献:

[1] PARK Y B. A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines [J]. International Journal of Productions Economics, 2001, 73(2): 175 – 188.
[2] TANG L X, LIU J Y, RONG A Y, et al. A multiple traveling sales-

man problem model for hot rolling scheduling in Shanghai Baoshan iron & steel complex [J]. European Journal of Operational Research, 2000, 124(2): 267 – 282.
[3] CARTER A E, RAGSDALE C T. A new approach to solving the multiple traveling salesperson problem using genetic algorithms [J]. European Journal of Operational Research, 2006, 175(1): 246 – 257.
[4] ZHAO F G, DONG J Y, LI S J, et al. An improved genetic algorithm for the multiple traveling salesman problem [C]// CCDC 2008: Chinese Control and Decision Conference. [S. l.]: IEEE, 2008: 1935 – 1939.
[5] PAN JUNJIE, WANG DINGWEI. An ant colony optimization algorithm for multiple travelling salesman problem[C]// ICICIC06: First International Conference on Innovative Computing, Information and Control. Washington, DC: IEEE Computer Society, 2006, 1: 210 – 213.
[6] BEKTAS T. The multiple traveling salesman problem: An overview of formulations and solution procedures[J]. Omega, 2006, 34(3): 209 – 219.
[7] 陈玉旺. 基于极值动力学的自组织优化理论、算法与应用研究 [D]. 上海: 上海交通大学, 2007.
[8] BOETTCHER S, PERCUS A G. Nature's way of optimizing [J]. Artificial Intelligence, 2000, 119(1/2): 275 – 286.
[9] LUYONG - ZAI, CHEN MIN - RONG, CHEN YU - WANG. Studies on extremal optimization and its applications in solving real world optimization problems[C]// Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence. Hawaii: IEEE, 2007: 162 – 168.
[10] BELLMORE M, HONG S. Transformation of multisalesmen problem to the standard traveling salesman problem[J]. Journal of Association for Computing Machinery, 1974, 21(3): 500 – 504.