

基于 IPSO 算法的 TSP 问题求解研究

高 峰 郑 波

(中国民航飞行学院 广汉 618307)

摘 要 为获得旅行商问题(Traveling Salesman Problem, TSP)的最优解,提出利用改进的粒子群优化(Improved Particle Swarm Optimization, IPSO)算法中求解 TSP 问题。IPSO 算法采用了粒子自适应更新机制和继承式判断机制,克服了传统算法易陷入局部最优位置的缺陷以及可调参数和初始位置随机设定对寻优结果不确定性的影响,确保在解空间内获得一致性的全局最优解。通过对不同样本 TSP 问题求解,验证了 IPSO 算法的有效性和稳定性。对比实验表明:IPSO 算法在解决大规模寻优问题时具有突出的全局寻优能力。

关键词 旅行商问题,改进的粒子群优化算法,自适应更新机制,继承式判断机制

中图法分类号 TP302.6

文献标识码 A

Study on TSP Solving Based on IPSO

GAO Feng ZHENG Bo

(Civil Aviation Flight University of China, Guanghan 618307, China)

Abstract In order to obtain the optimal solution of TSP, an improved particle swarm optimization (IPSO) was proposed for solving TSP. By using of adaptive updating mechanism and inheritance judgment mechanism, the IPSO overcomes the shortcomings of traditional algorithm falling into local best position easily and the effect of adjustable parameters and initial position set randomly on the uncertainty of optimization results, to ensure to obtain the consistency global optimal solutions in the solution space. By solving the different samples of TSP, we verified the effectiveness and stability of the IPSO. Comparative experiment show that the IPSO in solving large-scale optimization problem has the highlighted ability about the global optimization.

Keywords TSP, IPSO, Adaptive updating mechanism, Inheritance judgment mechanism

1 引言

旅行商问题(Traveling Salesman Problem, TSP)是著名的优化组合问题,具有广泛的实际应用价值,例如,电力系统中的电网配置优化,物流等行业中的资源配送优化,民航班机进出港排序优化,以及电路板钻孔线路设计等诸多工业领域的优化问题,都可归结为 TSP 问题求解^[1],因此对 TSP 问题求解具有重要的理论和实际意义。

TSP 问题可描述为:已知 n 个城市相互之间的距离,某一旅行商从某个城市出发访问每个城市一次且仅一次,最后再回到出发城市,如何安排才能使其所走路线最短^[2]。TSP 问题是典型的 NP 完全问题,即其求解的时间复杂度会随着问题规模的增大而按指数规模增长,到目前为止还未找出一种多项式时间的有效解析方法^[3]。但在智能优化领域,对 TSP 问题的研究非常活跃,遗传算法(Genetic Algorithm, GA)、人工神经网络(Artificial neural network, ANN)、粒子群优化算法(Particle Swarm Optimization, PSO)、蚁群算法(Ant Colony Algorithm, ACA)等智能方法被国内外学者广泛

应用于求解 TSP 问题,使得 TSP 问题求解取得了突破性进展。但研究表明,这些方法大多存在易陷入局部最优解、收敛速度过慢等缺陷,妨碍了获得全局最优解^[4]。

传统 PSO 算法通过不断更新个体和群体极值来完成寻优过程,具有数学描述简单,可调参数少,迭代过程易于实现等优势,但同样存在易陷入局部最优解的缺陷^[5]。因此本文在传统 PSO 算法的基础上提出一种 IPSO 算法,以保证在 TSP 解空间内获得全局最优解。

2 基于 IPSO 优化算法

PSO 算法源于对鸟类捕食行为的模拟,实质是基于群体叠代的启发式全局搜索算法^[6]。

2.1 传统 PSO 算法^[7]

假设搜索空间为 D 维,由 n 个粒子组成的总群为 $X = (X_1, X_2, \dots, X_n)$,第 i 个粒子位置向量表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,每一粒子代表一个潜在解;单个粒子 i 历史中的最优位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, P_g 为群体的最优位置,即所有 $P_i (i=1, 2, \dots, n)$ 中的最优位置;第 i 个粒子的位置变化率

本文受中国民航飞行学院青年科学基金项目(Q2013-145)资助。

高 峰(1985—),女,助理研究员,主要研究方向为项目优化管理,E-mail:gaof@cafuc.edu.cn;郑 波(1984—),男,博士生,讲师,主要研究方向为设备故障诊断、可靠性优化设计。

(速度)为向量 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。则每个粒子的速度和位置按如下公式进行变化:

$$\begin{cases} v_{id}^{k+1} = \omega \cdot v_{id}^k + c_1 \cdot r_1 \cdot (p_{id}^k - x_{id}^k) + c_2 \cdot r_2 \cdot (p_{gd}^k - x_{id}^k) \\ x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \end{cases} \quad (1)$$

式中, $1 \leq i \leq n, 1 \leq d \leq D; k$ 为当前迭代次数; c_1, c_2 是正常数, 称为加速因子, 一般在 $0 \sim 2$ 之间取值; r_1 和 r_2 为 $[0, 1]$ 之间两个相互独立的随机数; ω 称为惯性权重; 为了避免粒子盲目搜索的可能性, x 和 v 通常会限定在一定范围内。

2.2 IPSO 算法

传统 PSO 算法通过追踪整个搜索空间中的个体极值和群体极值来完成寻优, 但随着迭代次数不断增加, 粒子群在收敛聚集的同时, 各粒子之间的相似性也不断增大, 可能在局部最优解周围无法跳出, 导致最终获得局部最优解^[8], 同时由于传统 PSO 算法的速度向量为连续量, 因此其只适用于连续变化的粒子。基于此, 本文提出适用于离散粒子的自适应 IPSO 算法, 此算法摒弃了传统 PSO 算法中通过跟踪极值来更新粒子的方法, 采用粒子自适应更新和继承式判断机制来获得全局最优解。

2.2.1 粒子离散化编码

每一个粒子都是一个潜在最优解, 因此粒子应体现解的形式。对于 TSP 问题, 粒子应是遍历的所有城市的代码, 例如, 当经历的城市数为 7 时, 则粒子 X_i 编码为 (6, 5, 3, 1, 7, 4, 2), 表示城市遍历从城市 6 开始, 经历城市 $5 \rightarrow 3 \rightarrow 1 \rightarrow 7 \rightarrow 4 \rightarrow 2$, 最终返回城市 6, 形成一个闭合回路, 从而完成 TSP 遍历, 粒子 X_i 为潜在最优解。此时粒子是完全离散化的, 传统的速度更新公式不能满足离散粒子更新要求。

2.2.2 适应度值计算

粒子适应度函数是 PSO 寻优的依据, 其结果表示遍历回路的距离, 计算公式为: $f(X_i) = \sum_{i,j \leq D, i \neq j} path_{i,j}$, 其中, D 为城市数量, $path_{i,j}$ 为城市 i 到 j 间的距离。PSO 通过比较适应度值来确定是否更新极值。

2.2.3 自适应更新机制

为尽量拓展粒子在整个解空间的搜索区域, 保证获得全局最优解, 本文提出自适应更新机制, 通过随机设置更新系数 c_1 , 将粒子 $X_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD})$ 的第 j 位编码与第 $D - j + 1$ 位编码进行调换, 其中 $c_1 < D, j = 1, \dots, c_1$, 从而获得全新的粒子。

2.2.4 继承式判断机制

本文提出继承式判断机制, 用于对个体极值和群体极值的追踪和复制, 其继承过程为: 通过随机设置继承系数 c_2 和 c_3 , 其中 $c_2 < c_3 < D$, 将个体极值 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 或者群体极值 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 的 c_1 到 c_2 位间的编码复制到粒子 X_i 的最后 $c_2 - c_1 + 1$ 位, 对继承后的粒子 X_i , 若有重复编码, 则删除, 并用未出现的编码替代, 这样就产生了保留个体或群体极值信息的新粒子 X_i 。例如, 当经历的城市数为 12 时, 随机产生的继承系数 $c_1 = 4$ 和 $c_2 = 9$, 则由继承产生的新粒子过程为: 原粒子 $X_i = (1, 4, 7, 3, 6, 9, 8, 2, 10, 11, 5, 12)$, 群体极值 $P_g = (3, 7, 8, 9, 12, 4, 6, 5, 11, 2, 1, 10)$, 继承后的粒子 $X'_i = (1, 4, 7, 3, 6, 9, 9, 12, 4, 6, 5, 11)$, 但第 5 位

和第 6 位重复, 则删除, 并用编码 8 和 2 替换, 得粒子 $X_{new i} = (1, 4, 7, 3, 8, 2, 9, 12, 4, 6, 5, 11)$ 。

对于新继承的粒子是否具有继续进化的价值, 则采用优秀个体策略进行判断, 判断方式为: 若 $f(X_i) > f(X_{new i})$, 则是有效继承, 并用 $X_{new i}$ 替换 X_i , 否则是无效继承, 放弃 $X_{new i}$ 。IPSO 算法流程如图 1 所示。

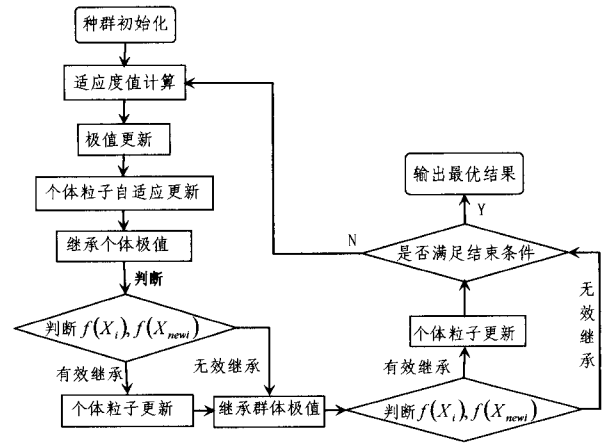


图1 IPSO 算法流程图

将粒子自适应更新和继承式判断机制应用到 PSO 算法中, 形成 IPSO 算法, 具有以下优点:

(1) 拓展 PSO 算法应用空间, 使算法能够处理离散粒子群。

(2) 采用粒子自适应更新, 每代进化时, 更新离散粒子, 可增强粒子拓展全局的能力, 确保完成对整个解空间的搜索。

(3) 继承式判断机制能确保最优极值信息被保留下去, 确保在进化过程中获得全局最优解。

(4) 自适应更新和继承式判断方式由于采用“片段式”更新方式, 因此保证了对整个粒子群“社会信息”良好的继承性, 使得在其他区域搜索寻优时, 避免了冗长的重新搜索过程, 加快了收敛速度。

(5) 与传统 PSO 算法相比, IPSO 算法可调参数少, 只需要确定进化代数 and 粒子种群数量, 因此计算精度更为客观准确。

3 基于 IPSO 算法的 TSP 问题求解

3.1 对小样本 TSP 问题求解研究

以遍历 14 座城市作为小样本研究对象, 其解空间包含约 3.114×10^9 种遍历路径, 14 座城市的坐标如表 1 所列, 图 2 是随机的遍历路径, 其距离为 43.4197km, 这样的遍历路径不具有数学应用意义, 但如何从众多路径中找出一条最佳路径是比较困难的问题。本文将用 IPSO 算法求解出一条最佳路径使得遍历所有城市后距离最短。

表1 城市位置坐标表(单位: km)

序号	X 坐标	Y 坐标	序号	X 坐标	Y 坐标
1	16.47	96.10	8	17.20	96.29
2	16.47	94.44	9	16.30	97.38
3	20.09	92.54	10	14.05	98.12
4	22.39	93.37	11	16.53	97.38
5	25.32	97.24	12	21.52	95.59
6	22.00	96.05	13	19.41	97.13
7	20.47	97.02	14	20.09	92.55

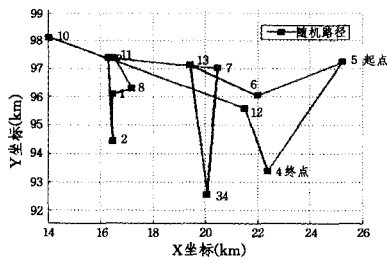


图2 14座城市随机遍历路径图

对于 IPSO 算法,由于需要遍历 14 座城市,因此个体粒子 $X_i=(x_{i1},x_{i2},\cdots,x_{i14})$ 为 14 维向量,由于只有 14 座城市,城市样本数较少,因此取进化代数 50 代,粒子总群数量为 100,所有参数设置完成。按照图 1 所示的流程让 100 个粒子在整个解空间进行搜索,并根据适应度值判断是否更新极值。进化 50 代后,得到图 3 所示的最优规划路径,遍历过程为:13→7→12→6→5→4→3→14→2→1→10→9→11→8→13,距离为 29.3405km。图 4 为 IPSO 算法的适应度值变化曲线,经过 34 次进化后,得到全局最优解。

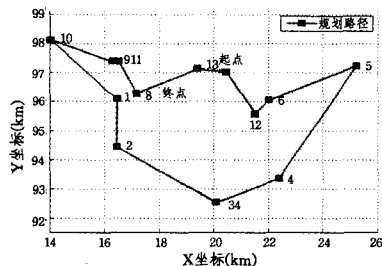


图3 14座城市最优规划路径图

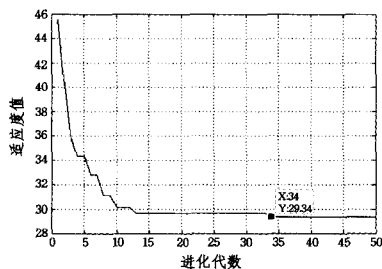


图4 IPSO算法适应度值变化曲线

将 IPSO 算法与 ACA 算法和传统 PSO 算法对小样本 TSP 问题求解结果进行比较,比较结果如表 2 所列。表 2 表明:在 10 次求解中,IPSO 算法和 ACA 算法都能获得全局最优解,而 PSO 算法只有 8 次获得全局最优解,其余 2 次获得局部最优解。在对小样本 TSP 问题求解中,各算法都能获得全局最优解,这是因为小样本 TSP 问题的解空间相对较小,从而局部极值数量较少且形成局部极值的解的范围较小^[9],因此各算法易跳出局部极值而获得全局最优解。

表2 实验结果对比表

计算方法	求解次数	最优解次数	最优解值	平均值
IPSO	10	10	29.3405	29.3405
PSO	10	8	29.3405	29.7544
ACA	10	10	29.3405	29.3405

3.2 对大样本 TSP 问题求解研究

对于大样本 TSP 问题求解,以著名的中国 TSP 问题为研究对象,我国 31 个直辖市、省会和自治区首府的遍历路径应用约 1.326×10^{32} 种^[10],较小样本 TSP 问题,其解空间要庞

大得多。31 座城市的坐标如表 3 所列,图 5 是随机的遍历路径,其距离为 31352.5028km,同样不具有实际指导意义,因此 IPSO 算法将从庞大的解空间中求解出一条最短路径。

表3 城市位置坐标表(单位:km)

序号	X 坐标	Y 坐标	序号	X 坐标	Y 坐标
1	1304	2312	17	3918	2179
2	3639	1315	18	4061	2370
3	4177	2244	19	3780	2212
4	3712	1399	20	3676	2578
5	3488	1535	21	4029	2838
6	3326	1556	22	4263	2931
7	3238	1229	23	3429	1908
8	4196	1004	24	3507	2367
9	4312	790	25	3394	2643
10	4386	570	26	3439	3201
11	3007	1970	27	2935	3240
12	2562	1756	28	3140	3550
13	2788	1491	29	2545	2357
14	2381	1676	30	2778	2826
15	1332	695	31	2370	2975
16	3715	1678			

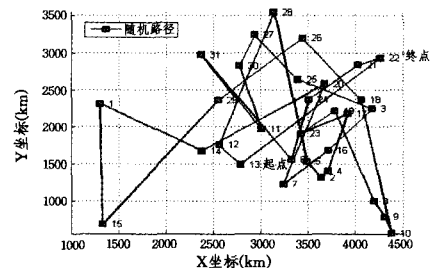


图5 31座城市随机遍历路径图

由于需要遍历 31 座城市,因此 IPSO 算法的个体粒子 $X_i=(x_{i1},x_{i2},\cdots,x_{i31})$ 为 31 维向量,取进化代数为 400 代,粒子总群数量为 500。进化 400 代后,得到图 6 所示的最优规划路径,遍历过程为:24→11→23→16→4→8→9→10→2→5→6→7→13→12→14→15→1→29→31→30→27→28→26→25→20→21→22→18→3→27→19→24,距离为 15377.7113km。据最新研究成果表明,中国 TSP 问题的最优解为 15378km^[11],因此,IPSO 所搜索的最优规划路径为全局最优解。

对于大样本 TSP 问题,由于更新系数 c_1 以及继承系数 c_2 、 c_3 为随机设置的,变量的随机性易影响寻优结果,造成获得局部最优解。为验证 IPSO 算法对大样本 TSP 问题的全局搜索能力,本文多次调用 IPSO 算法求解 TSP 问题,其结果均为 15377.7113km,图 7 为其中任意两次求解的适应度值变化曲线,图 7 表明:分别在经历 190 次和 307 次进化后,IPSO 获得全局最优解。因此 IPSO 算法能够确保获得全局最优解。

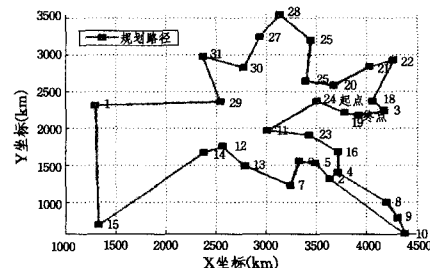


图6 31座城市最优规划路径图

(下转第 82 页)

from a moving vehicle[J]. Machine Vision and Applications, 1999,11(4):203-212

- [4] Bar-Shalom Y, Li X R, Kirubarajan T. Estimation with applications to tracking and navigation: theory algorithms and software [M]. John Wiley & Sons, 2004
- [5] Gandhi T, Trivedi M. Motion analysis of omni-directional video streams for a mobile sentry[C]// First ACM SIGMM international workshop on Video surveillance. ACM, 2003:49-58
- [6] FUJITSU[OL]. <http://www.fujitsu.com/us/semiconductors/gdc/products/omni.html>
- [7] Gandhi T, Trivedi M. Motion based vehicle surround analysis using an omni-directional camera[C]// Intelligent Vehicles Symposium, 2004 IEEE. IEEE, 2004:560-565
- [8] Broggi A, Bertozzi M, Fascioli A, et al. Shape-based pedestrian detection[C]// Proceedings of the IEEE Intelligent Vehicles

Symposium. 2000;215-220

- [9] Shashua A, Gdalyahu Y, Hayun G. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance[C]// Intelligent Vehicles Symposium, 2004 IEEE. IEEE, 2004:1-6
- [10] Bertozzi M, Broggi A. GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection[J]. IEEE Transactions on Image Processing, 1998,7(1):62-81
- [11] 丁鑫. 全景视觉泊车辅助系统研究[D]. 杭州: 浙江大学, 2010
- [12] 鲁斌, 秦瑞, 李庆, 等. 车载环视拼接方法的研究[J]. 计算机科学, 2013,40(9):293-295
- [13] 赵逢. 基于单目视觉的后方障碍物检测算法的研究与实现[D]. 沈阳: 东北大学, 2008
- [14] 王红伟. 基于单目视觉三维重建的障碍物检测算法的设计与实现[D]. 沈阳: 东北大学, 2008

(上接第 71 页)

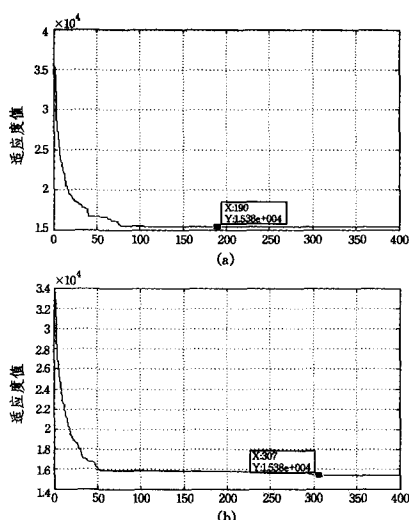


图 7 IPSO 算法适应度值变化曲线

同样将 IPSO 算法与 ACA 算法和传统 PSO 算法对大样本 TSP 问题求解结果进行比较, 比较结果如表 4 所列。表 4 表明: 在 10 次求解中, IPSO 算法都能获得全局最优解, 而 ACA 算法和 PSO 算法只能获得局部最优解, 这是由于 ACA 算法和 PSO 算法全局搜索能力差, 易陷入某一局部最优位置无法跳出, 且众多可调参数和初始位置的随机性, 导致每一次求解都可能获得不一样的最优值^[12]。针对 ACA 和 PSO 算法存在的缺陷, IPSO 算法提出相应的应对策略, 从而很好地解决了这类缺陷, 使其具有更高的应用价值。

表 4 实验结果对比表

计算方法	求解次数	最优解次数	最优解值	平均值
IPSO	10	10	15377.7113	15377.7113
PSO	10	1	16944.3547	18267.6424
ACA	10	6	15601.9195	15818.4354

结束语 本文提出利用 IPSO 算法求解不同样本 TSP 问题的最优遍历路径, 在小样本 TSP 问题求解中, 各算法求解性能差距不大, 基本能获得全局最优解。但在大样本 TSP 问题求解中, 传统算法易获得局部最优解, 且受众多可调参数和

初始位置随机设置的影响, 解具有很强的随机性, 因此 IPSO 算法通过采取粒子自适应更新来拓展搜索空间, 确保完成对整个解空间的搜索, 同时利用继承式判断机制来确保获得全局最优解。通过对比实验研究, 且与传统的 PSO 和 ACA 算法进行比较, 验证了 IPSO 算法在大样本 TSP 问题求解中的有效性和稳定性, 表明 IPSO 算法在解决大规模寻优问题中具有突出的全局寻优能力, 因此其具有更高的实际使用价值。

参考文献

- [1] Meer K. Simulated Annealing Versus Metropolis for a TSP Instance[J]. Information Processing Letters, 2007,12:216-219
- [2] 张辉, 赵正德, 杨立朝, 等. TSP 问题的算法与应用研究[J]. 计算机应用与软件, 2009,26(4):274-276
- [3] 王剑文, 戴光明, 谢柏桥, 等. 求解 TSP 问题算法综述[J]. 计算机工程与科学, 2008,30(2):72-74
- [4] 谷文祥, 李向涛, 王春颖, 等. 一种求解 TSP 问题的混合算法[J]. 东北师范大学: 自然科学版, 2011,43(3):60-64
- [5] Shi X H, Liang Y G, Lee H P, et al. Particle Swarm Optimization-based Algorithms for TSP and generalized TSP [J]. Information Processing Letters, 2007,8:169-176
- [6] Venter Sobieszczanski-sobieski J. Particle swarm optimization [J]. AIAA Journal, 2003,41(8):1583-1589
- [7] 郑波. 基于 PSO-SVM 的民航发动机送修等级决策研究[J]. 推进技术, 2013,34(5):68-69
- [8] Eberhart R, Shi Yu-hui. Particle swarm optimization: Developments applications and resources[C]// Proc IEEE Int Conf on Evolutionary Computation. Seoul 2001 :81-86
- [9] 陈文兰, 戴树贵. 旅行商问题算法研究综述[J]. 滁州学院学报, 2006,8(3):1-6
- [10] 史峰, 王辉, 郁磊, 等. MATLAB 智能算法——30 个案例分析 [M]. 北京: 北京航空航天大学出版社, 2011
- [11] Dorigo M, Birattari M, Stutzle T. Ant Colony Optimization[J]. Computational Intelligence Magazine, 200,61(4):28-39
- [12] Misevicius A. A tabu search algorithm for the quadratic assignment problem [J]. Computational Optimization and Applications, 2005(30):95-111