一种改进的求解 TSP 问题的演化算法

蔡之华[1] 彭锦国[1] 高 伟[1] 魏 巍[1] 康立山[1),2)]

[1)](中国地质大学计算机学院 武汉 430074)

[2)](武汉大学软件工程国家重点实验室 武汉 430072)

摘 要 演化算法是解决组合优化问题的高效搜索算法.该文在现有求解 TSP 问题的演化算法的基础上,通过引入映射算子、优化算子以及增加一些控制策略,提出了一种高效的演化搜索算法.实验表明,该算法是有效的,通过对 CHN144 以及国际通用的 TSPLIB 中不同城市规模的数据进行测试表明,其中实例 CHN144 得到的最短路径为 30353.860997,优于吴斌等运用分段算法得到的最短路径 30354.3,亦优于朱文兴等人的结果,实例st70和 kroB150 得到的最短路径分别与运用分段算法得到的最短路径值相同,实例 pr136 得到的最短路径值为 96770.924122,优于 TSPLIB 中提供的最短路径 96772,对于其它实例也均能快速地得到和 TSPLIB 中提供的最优路径相同或更优的路径,该算法不仅很容易收敛到问题的最优解,而且求解速度极快.

关键词 旅行商问题;演化算法;算子

中图法分类号 TP18

An Improved Evolutionary Algorithm for the Traveling Salesman Problem

CAI Zhi-Hua[1)] PENG Jin-Guo[1)] GAO Wei[1)] WEI Wei[1)] KANG Li-Shan[1),2)]

[1)](*School of Computer Science and Technology, China University of Geosciences, Wuhan* 430074)

[2)](*State Key Laboratory of Software Engineering, Wuhan University, Wuhan* 430072)

Abstract TSP(Traveling Salesman Problem) is one of the typical NP-hard problems in combinational optimization, and evolutionary algorithm is a more efficient and more effective for solving combinational optimization problem. This paper gives two operators, mapping operator and optimal operator, then proposes some control strategies, finally proposes an improved evolutionary algorithm based on Guotao algorithm for solving TSP. First four TSP instances are used to test the algorithm, then some standard test-instances are tested. Especially, authors test the CHN144 (Chinese 144 cities). The result, 30353.860997, outperforms those from existing literatures, 30354.3(from Wu), 30355.01(from Zhu).

Keywords traveling salesman problem; evolutionary algorithm; operator

1 引 言

对 TSP(Traveling Salesman Problem)的研究由来已久,它可简单描述为:给出一条遍历给定的若干个城市中所有城市的最短路径.用数学语言表述为:设有 N 个城市 $C=\{C_1,C_2,C_3,\cdots,C_n\}$,其中任意两个城市的距离记为 $d(C_i,C_j)$,求一条经过 C 中所有

收稿日期:2003-04-28;修改稿收到日期:2005-03-08.本研究得到国家自然科学基金(60473081)和湖北省自然科学基金(2001ABB006,2003ABA043)资助.蔡之华,男,1964 年生,博士,教授,博士生导师,主要研究方向为演化计算、数据挖掘. E-mail:zhcai@cug.edu.cn.
彭锦国,高 伟,魏 巍,均为大学本科生.康立山,1934 年生,教授,博士生导师,主要研究方向为计算机理论、并行与分布计算、演化计算.

城市一次且仅一次的一条路径($C_{n(1)}$,$C_{n(2)}$,$C_{n(3)}$,…,$C_{n(N)}$),使得闭合路径$\sum_{i=1}^{N} d[C_{n(i)}, C_{n(i+1)\bmod N}]$为最小.

TSP 问题是著名的 NP 难问题.由于该问题的描述简单,而其实际模型在路径、网络、分配等优化问题中又有着广泛的应用[1],故长期以来一直吸引着许多领域的研究人员对其算法改进的关注.传统的穷举搜索法、贪心法、动态规划法等都面临着这样一个共同的问题,即当问题的规模 N 大到一定程度时,会产生所谓的"组合爆炸"问题,即问题的计算量极大地超出了机器所能允许的极限.例如,若用穷举搜索法对 $N=50$ 的 TSP 问题进行求解,即使采用每秒钟计算 1 亿次的超级计算机所需的时间要 5×1048 年[2].1975 年,Michigan 大学 Holland 教授的开创性的著作《Adaptation in Natural and Artificial Systems》,提出了基于生物进化论思想的遗传算法 GA (Genetic Algorithm).该算法是受达尔文的生物进化论的启发,通过将搜索对象(种群)进行繁殖(reproduction)、杂交(crossover)、变异(mutation)等遗传操作,并在一定的控制参数(演化代数、执行不同的遗传操作的概率等)的作用下,在解空间中迅速搜索满足停机条件的最优解.Holland 的著作中提出的算法通常被称作简单遗传算法(SGA).在此之后,各种在 SGA 基础上改进的演化算法相继涌现.文献[3]中提出的算法被证明是一种效率较高的演化算法.

本文在文献[3]的基础上,通过引入映射算子、优化算子以及增加一些控制策略,在原有演化算法的基础上,提出一种高效的演化搜索算法,本文选用通用的 TSPLIB 中的几个实例以及文献[4]中的中国 CHN144 城市问题的数据进行了测试,实验证明该算法是有效的.其中实例 1 对 CHN144 进行了测试,得到的最短路径值为 30353.860997,优于文献[5]中运用分段算法得到的最优路径 30354.3,也优于文献[6]的结果.实例 pr136 和实例 pr144 用 IGT 算法得到的最短路径值分别为 96770.924122 和 58535.221761,均优于 TSPLIB 中提供的最短路径值 96772 和 58537.对于实例 a280, TSPLIB 中提供的最短路径为 2579,而 TSPLIB 中最优路径计算结果为 2586.769648,IGT 算法得到的最短路径值也为 2586.769648,与 TSPLIB 中提供的最优路径长度相同.对于其它实例也均得到和 TSPLIB 中提供的最优路径相同或更优的路径.

2 演化算法求解 TSP

2.1 演化算法

2.1.1 演化算法基本原理

演化算法是用计算机模拟大自然的演化过程,特别是生物进化过程,以求解复杂问题的一类计算模型,其基本思想是 Darwin 的进化论和 Mendel 的遗传学说.这种算法是把各种复杂的问题采用编码技术来表示,并通过将一组编码表示置于问题的环境中进行遗传操作和优胜劣汰的自然选择来指导学习和确定搜索的方向,由于演化计算具有自组织、自适应、自学习等特征,因此优胜劣汰的自然选择和遗传操作具有不受其搜索空间限制性条件的约束以及不需要其他辅助信息的特点.但是,对于具体的问题而言(如 TSP),通常由于解空间的庞大以及当前计算机计算能力的局限,使得我们需要寻找更有效的杂交的变异算子以及更强有力的控制策略.

2.1.2 TSP 问题中演化算法的相关定义

染色体编码:本文采用 TSP 问题的最自然、最直接的表示方式,即路径表示.它直接将一条路中的城市按城市编号顺次记录,例如,路径 4—2—3—1—4可直接表示为(4,2,3,1)或(2,3,1,4),一条染色体表示一条合法的路径.本文所提到的路径均为环路,染色体为环链.

适应度:它是衡量某一个体优良与否的唯一标准,本文算法的适应度采用路径长度的倒数表示,适应度越大,则路径越短,个体越优.本文计算适应度的评价函数定义为 $f(S_i)=1\Big/\sum_{i=1}^{N} d[C_{n(i)}, C_{n(i+1)\bmod N}]$.

演化算子:演化算子即演化策略,本文提到的演化策略有杂交算子、变异算子、映射算子.

选择策略:进行杂交或是变异以后,产生新的个体,为了保持样本总数一定,需要进行淘汰,即去掉不良个体,保留优良个体.

变异概率:为了保持样本的多样性,避免陷入局部极小值,同时又要保证算法收敛够快,要求变异概率一定要适当.

停机条件:即算法的中止条件.

2.2 求解 TSP 的郭涛算法(GT)

郭涛算法[3]被证明是一种高效率的遗传算法.该算法的创新之处在于遗传算子实施的倒位操作,通过该操作,可以有效地增加群体的多样性,扩大搜索域,防止过早地产生局部收敛,并迅速地找出满足条件的最优解.

郭涛算法中的演化算子：

变异算子：在父体 S 中随机地选取两城市 C、C'，对 C 的下一个城市与 C' 间的所有城市（包括 C'）进行倒位操作. 例如：在父体 $S(1,2,3,4,5,6,7,8,9,10,11,12)$ 中随机地选取 $C=3$，$C'=11$，则进行变异操作后所得的个体为 $S(1,2,11,10,9,8,7,6,5,4,3,12)$.

杂交算子：在一父体 S 中随机选取一城市 C，另取一父体 S'，在 S' 中指定 C 的下一个城市为 C'. 对 S 中 C 的下一个城市与 C' 间的城市实施倒位操作，若 S 中 C' 是与 C 相邻的城市，则不进行倒位操作. 例如：选取一父体 $S(1,2,3,4,5,6,7,8,9,10,11,12)$，令 $C=2$，随机地选取另一父体 $S'(1,3,5,6,7,9,2,8,10,11,4,12)$，则 $C'=8$，在 S 中对 3 至 8 间的基因实施倒位操作得到子代为 $(1,2,8,7,6,5,4,3,9,10,11,12)$.

2.3　改进的郭涛算法（IGT）

IGT 是在郭涛算法（GT）[3] 的基础上通过引入新的遗传算子，并对算法作了部分改进，以提高算法的收敛速度.

2.3.1　IGT 算法对 GT 算法的主要改进之处：

（1）引入新的演化算子映射算子.

映射算子：随机从种群 $\{P\}$ 中挑出两条染色体 S_1 和 S_2，不妨设 $f(S_2)>f(S_1)$（即 S_2 较优，其中 f 为评价函数），从 S_1 中随机选择一段基因片断 ΔS_1，判断 S_2 中是否存在这样一段基因 ΔS_2：它与 ΔS_1 长度相同且第一位城市编号相同，如果存在 ΔS_2，则将染色体 S_1 中的基因片断 ΔS_1 置换为 ΔS_2，剩余基因按部分映射[4] 进行调整.

优化算子：随机选择两条染色体 S_1，S_2，不妨设 $f(S_2)>f(S_1)$（即 S_2 较优），将 S_1 作为父体，随机选择 S_1 中一段路径 ΔS_1，然后判断 S_2 中是否存在这样一段路径 ΔS_2，它与 ΔS_1 中的基因长度相同且有相同的城市编号，只是基因城市的排列顺序不同，判断 ΔS_1 和 ΔS_2 的基因片路径长度，如果 ΔS_2 的路径较短，则用 ΔS_2 置换掉 S_1 的 ΔS_1 片段.

（2）动态地改变算子的选择概率. 由于该算法的演化主要以杂交为主，但是杂交算子对现有群体的依赖性较强很容易陷入局部最优，因此辅以变异算子可以改变这一局限，使得群体覆盖面大为增加，但是变异本身是盲目的，它只在算法执行的初期有效，当算法已经收敛到近似解时，通过变异成功优化染色体的概率显得极低，这就需要动态地改变变异算子的选择概率，以尽可能提高算法的收敛速度. IGT 算法通过以下公式动态改变变异概率：

$$p = p \times (1 - GenNum \times 0.01/maxGen),$$

其中 p 为变异概率，$GenNum$ 为当前代数，$maxGen$ 为算法终止的最大代数.

（3）提出临界速度的概念，对算法进行优化控制以使收敛加快. 临界速度主要在以下两个地方对算法进行控制：

（a）当演化速度大于临界速度，对染色体运用杂交算子时，在整条染色体运算完毕时才进行测试是否替换父体染色体；当演化速度小于临界速度，对染色体运用杂交算子时，每进行一次逆转操作均进行测试，一旦测试到有所优化，便马上替换掉父代染色体. 这一策略使得算法运行的前期收敛很快，后期很容易收敛到最优解.

（b）当演化速度大于临界速度时，禁止映射算子作用于染色体，因为映射算子只有在算法已经群体较优的前提下，才开始发挥其高效性.

（4）满足优化概率时对最优个体进行集中优化，使其集中所得群体的较优性能，这一措施在算法执行的尾期非常有效.

改进的郭涛算法（IGT）.

Begin

1. 随机生成种群 $P=\{S_0,S_1,\cdots,S_{N-1}\}$，其每一条染色体是一条遍历所有城市的随机路径，计算每条染色体适应度 F_i，初始化各参数：变异算子、映射算子、演化速度；

2. 初始化 $i=0$；

3. 将种群 P 中的染色体 S_i 赋值给空染色体 S 从 S' 中随机选择一基因城市 C；

4. 变异算子. 如果 0～1 之间的随机数小于变异概率 $P1$，从 S' 中任意选择一不同于 C 的基因 C'，并逆转 C 和 C' 之间的基因片断，转至步 6；

5. 杂交算子. 从种群 P 中随机选择不同于 S_i 的另一条染色体 S''，在 S'' 中指定与 C 相邻的下一个基因位 C'. 判断：

5.1. 如果 S' 中 C' 的位置与 C 相邻，转至步 7；

5.2. 将 S' 中从 C 到 C' 之间的基因片断进行倒位操作；

5.3. 计算逆转前后路径改变值 d，如果 $d<0$ 且演化速度小于临界速度，转至步 7；

6. 将 C' 赋值给 C，重复执行步 4 开始的操作，直到 C 已经取过 S' 的每一个基因位；

7. 计算 S' 的适应值，判断：如果 $f(S')>F_i$，将 S' 替换掉 S_i，F_i 赋新值 $f(S')$（评价函数 $f(S)$ 为个体的路径长度的倒数，长度短的适应值高）；

8. $i+1$. 重复执行步 3 至步 8 直到 $i=N$；

9. 重新计算演化速度，更新变异概率；

10. 映射算子. 如果演化速度小于临界速度且 0～1 之间的随机数小于映射概率 $P2$：

10.1. 随机从 $\{P\}$ 中挑出两条染色体 S_1 和 S_2，不妨设 $f(S_2)>f(S_1)$；

10.2. 从 S_1 中随机选择一段基因片断 S_1，判断 S_2 中

是否存在这样一段基因 S_2：它与 S_1 长度相同且第一位城市编号相同.

10.3. 如果存在 S_2，则将染色体 S_1 中的基因片断 S_1 置换为 S_2，剩余基因按部分映射[4]进行调整.

11. 优化算子：选择目前最优个体中的若干个基因片断，分别判断其余个体中是否有城市编号相同、基因长度相同、片断适应值更高的片断，如果有，则替换掉最优个体中相应的基因片断.

12. 重复执行步 2～步 11，直到满足停机条件，则输出最优染色体.

End

2.3.2 改进的郭涛算法(IGT)的特点

(1)保留了 GT 算法中一个体只与自己的后代竞争的选择机制，使得算法在运行中能够保持群体的多样性，以避免传统遗传算法中的早熟现象.

(2)以杂交算子为主，其它算子为辅，同时增加一些控制机制，使得算法能够更为快速地收敛.

(3)该算法的演化过程是一个并行爬山的过程，因此很容易得到问题的全局最优解，避免了过早陷入局部最优解.

3 实验结果

本文的所有实验数据均在赛扬Ⅱ 1.3G 的 PC 机上运行得到，操作系统为 Windows XP，以下是部分实验数据.

下列结果中各参数分别表示如下意义：

变异概率 $P1$，映射概率 $P2$，种群大小 N，临界速度 Es.

实例 1. 对 CHN144. CHN144 实例的测试结果(10 次运算)，所使用的参数 $P1=0.02$，$P2=0.05$，$Es=5000$，$N=80$：

30353.860997	30353.860997	30353.860997
30353.860997	30386.366881	30353.860997
30356.370158	30353.860997	
30356.370158	30353.860997	

通过对文献[7]中提供的 144 城市问题的数据进行测试，IGT 算法得到了比较满意的结果，以下是 10 次测试的数据结果.

最好路径长度：30353.860997，

最差路径长度：30386.366881，

平均路径长度：30357.613411，

最大偏差：32.505813，

最短运算时间：5.36s，

平均运算时间：7.938s.

对于实例 CHN144，GGT 算法得到的最短路径值为 30353.860997，优于文献[5]中运用分段算法得到的最优路径 30354.3，也优于文献[6]的结果.

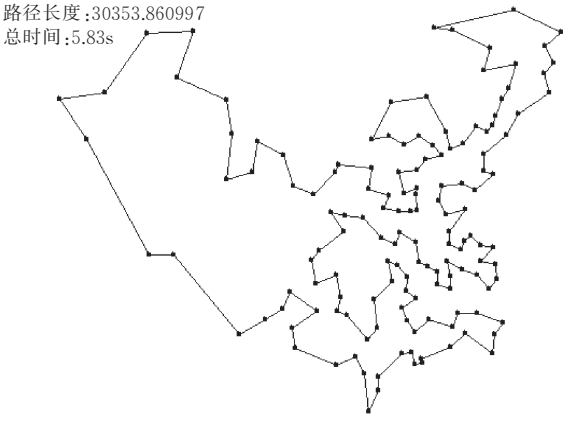

图 1 IGT 算法得到的 CHN144 最优路径图

为了更好地说明该算法的有效性，我们选用了国际上最通用的 TSP 测试库 TSPLIB① 中的多个实例进行测试.

实例 2. pr136.

pr136 实例测试结果(10 次运算)，所使网络参数为 $P1=0.02$，$P2=0.05$，$Es=5000$，$N=100$：

96770.924122	96770.924122	96770.924122
96770.924122	96770.924122	96770.924122
96770.924122	96770.924122	
96770.924122	96770.924122	

最好路径长度：96770.924122，

最差路径长度：96770.924122，

平均路径长度：96770.924122，

最大偏差：0.0，

最短运算时间：4.28s，

平均运算时间：7.105s.

对于实例 pr136，IGT 算法得到的最短路径值为 96770.924122，优于 TSPLIB 中提供的最短路径 96772.

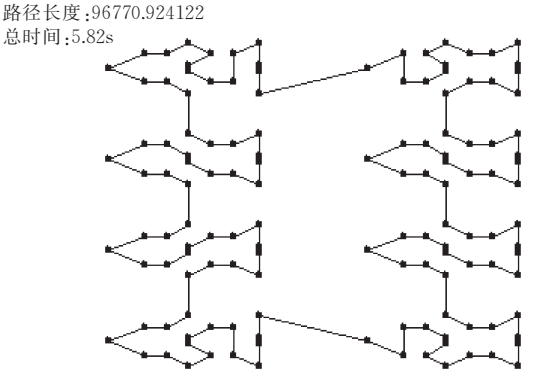

图 2 pr136 最优路径

① http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/

实例 3.　pr144.

pr144 **实例测试结果**(10 **次运算**),**所使用的参数为** $P1=0.02, P2=0.05, Es=10000, N=80$:

58535.221761	58535.221761	58535.221761
58586.536331	58535.221761	58535.221761
58535.221761	58535.221761	
58535.221761	58552.984948	

最好路径长度:58535.221761,

最差路径长度:58586.536331,

平均路径长度:58542.129537,

最大偏差: 51.31457,

最短运算时间:4.33s,

平均运算时间:7.973s.

对于实例 pr144,IGT **算法得到的最短路径值为** 58535.221761,**优于** TSPLIB **中提供的最短路径** 58537.

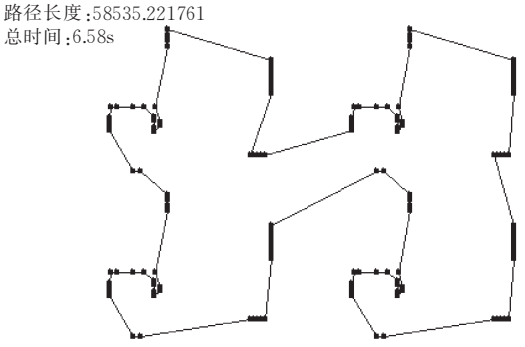

图 3　pr144 **最优路径图**

实例 4.　a280.

a280 **实例测试结果**(10 **次运算**),**所使用的参数为** $P1=0.02, P2=0.05, Es=500, N=100$:

2586.769648	2589.697334	2586.769648
2588.658191	2586.769648	2594.764832
2590.904403	2586.769648	
2586.769648	2587.808791	

最好路径长度:2586.769648,

最差路径长度:2594.764832,

平均路径长度:2588.568179,

最大偏差: 7.995184,

最短运算时间:26.11s,

平均运算时间:33.471s.

对于实例a280, TSPLIB**中提供的最短路径为** 2579,**而**TSPLIB**中最优路径计算结果为** 2586.769648, IGT **算法得到的最短路径也值为** 2586.769648,**与** TSPLIB **中提供的最优路径长度相同.**

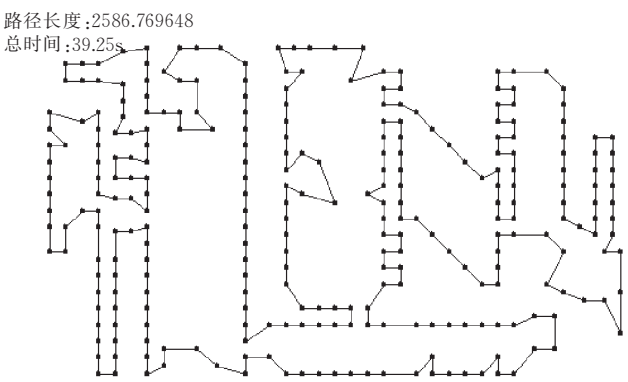

图 4　a280 **最优路径图**

其它实例的测试结果见表 1.

表 1　**其它实例测试结果**

实例名称	TSPLIB 提供最短路径值	TSPLIB 提供最优路径计算结果	实验所得最好路径值	计算时间(s)
st70	675	678.597452	677.109609	0.67
eil76	538	545.387552	544.369053	1.16
pr76	108159	108159.438274	108159.438274	0.75
rd100	7910	7910.396210	7910.396210	2.14
kroA100	21282	21285.443182	21285.443182	1.69
kroB100	22141		22139.074615	2.62
kroA150	26524		26524.863036	7.44
kroB150	26130		26127.357889	9.20

从表 1 **中可以看出,实验中,实例** st70, eil76 **所得结果均优于** TSPLIB **中所提供的最优路径计算结果;实例** kroB100, kroB150 **所得结果优于** TSPLIB **中所提供的最短路径值;实例** pr76, rd100, kroA100, kroA150 **所得运算结果与** TSPLIB **所提供的最优路径计算结果或最短路径值相同.**

4　结　语

本文通过对求解 TSP **问题的演化算法进行改进,通过引入新的演化算子:映射算子和优化算子,以及增加临界速度和动态地改变变异概率等控制策略,来加快算法的收敛速度,实验表明该算法是高效**

的,特别是在问题规模很大时,选择较小的群体规模进行演化,仍然能收敛到问题的最优解,而且收敛速度非常快,解决了传统遗传算法中运算时间过长和收敛较慢的问题.通过我们的实验,表明该算法在求解城市规模在 500 以下的 TSP 问题非常有效.

参 考 文 献

1 Michalewicz Z. *et al*.. How to Solve It——Modern Heuristick. Berlin Heidelberg: Springer-Verlag, 2000

2 Wang Wei. Artificial Neural Network: Theory and Application. Beijing: Beijing University of Aeronautic and Astronautic Science and Technology Press, 1995(in Chinese)
(王 伟. 人工神经网络原理与应用. 北京:北京航空航天大学出版社, 1995)

3 Guo Tao, Michalewicz Z.. Inver-over operator for the TSP. In: Eiben A. E. *et al*. eds.. Proceedings of the 5th Parallel Problem Soving from Nature Conference. Lecture Notes in Computer Science 1498, Berlin: Springer, 1998, 803～812

4 Pan Zheng-Jun, Kang Li-Shan, Chen Yu-Ping. Evolutionary Computation. Beijing: Tsinghua University Press, Nanning: Guangxi Science and Technology Press, 1998(in Chinese)
(潘正君,康立山,陈毓屏. 演化计算. 北京:清华大学出版社,南宁:广西科学技术出版社, 1998)

5 Wu Bin, Shi Zhong-Zhi. A ant colony algorithm based partition algorithm for TSP. Chinese Journal of Computers, 2001, 24(12): 1328～1333(in Chinese)
(吴 斌,史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法. 计算机学报, 2001, 24(12): 1328～1333)

6 Zhu Wen-Xin, Fu Qing-Xiang. A filled function method for the traveling savelsman problems. Chinese Journal of Computers, 2002, 25(7): 701～707(in Chinese)
(朱文兴,傅清祥. 一个基于填充函数变换的对称 TSP 问题的局部搜索算法. 计算机学报, 2002, 25(7): 701～707)

7 Kang Li-Shan, Xie Yun, You Shi-Yong, Luo Zu-Hua. Non-Numeric Parallel Algorithm (Vol. 1). Beijing: Science Press, 1997 (in Chinese)
(康立山,谢 云,尤矢勇,罗祖华. 非数值并行算法(第一册):模拟退火算法. 北京:科学出版社, 1997)

8 Zhao Chun-Ying, Zhang-Ling. A good-point set genetic algorithm for TSP. Computer Engineering and Applications, 2001, 37(3):83～84(in Chinese)
(赵春英,张 铃. 求解货郎担问题(TSP)的佳点集遗传算法. 计算机工程与应用, 2001, 37(3): 83～84)

CAI Zhi-Hua, born in 1964, Ph. D., professor, Ph. D. supervisor. His main research interests include data mining and evolutionary computation.

PENG Jin-Guo, **GAO Wei**, **WEI Wei**, all of them are student.

KANG Li-Shan, born in 1933, professor, Ph. D. supervisor. His main research interests include theoretical computer sciences, parallel and distributed computing, evolutionary computation.

Background

TSP(Traveling Salesman Problem) is a famous NP-hard problem. There are many algorithms have been proposed for solving this problem, for example, ant colony algorithm based partition algorithm, filled function method, gene-based genetic algorithm, etc. A class powerful algorithm for solving TSP are evolutionary algorithms. They are stochastic search algorithms that are inspired by models of biological evolution, and Guotao's Inver-order algorithm for the TSP has been demonstrated that is the one of the more famous evolutionary algorithm based on GuoTao algorithm(GT). In this paper, authors first design mapping and optimal operators, and propose some control strategies. Then an improved GuoTao algorithm (IGT) for TSP is proposed. This algorithm is tested on some standard test instances. The results show that the algorithm presented in this paper is more efficient and more effective than existing algorithms for TSP, and it can be easily and directly generalized to solve other NP-hard combinatorial optimization problems. The research is supported by the National Natural Science Foundation of China under grant No. 60073043 and the National Natural Science Foundation of Hubei Province of China under grant No. 2001ABB006, 2003ABA043.