

利用几何结构求解欧氏平面 TSP 的改进遗传算法*

潘 亮,朱华勇,沈林成,常文森

(国防科技大学机电工程与自动化学院,湖南长沙 410073)

摘要 :TSP 是经典的组合优化问题。根据欧氏平面 TSP 最优环路的性质提出了子路径及相关的概念,利用点集凸壳设计了环路构造算法,并以点集 Delaunay 三角剖分图为启发信息设计了改进的遗传算法,通过中国 144 城市 TSP 等验证了算法的有效性。

关键词 欧氏平面 ;TSP ;凸壳 ;Delaunay 三角剖分 ;遗传算法

中图分类号 :TP301 **文献标识码** :A

An Improved Genetic Algorithm to Solve the Euclidean Plane TSP by Using Geometry Structure

PAN Liang, ZHU Hua-yong, SHEN Lin-cheng, CHANG Wen-sen

(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract :The TSP is a classic combinatorial optimization problem. According to the character of the optimal tour of Euclidean plane TSP problem, the sub-path and related notions are presented. A tour construction algorithm is designed by using convex hull, and a genetic algorithm is improved to solve the problem by using Delaunay triangulation diagram as heuristic information. The experimental results in the 144 cities in China and other TSP instances show that the algorithm is effective.

Key words :Euclidean plane ; traveling salesman problem ; convex hull ; Delaunay triangulation ; genetic algorithm

TSP (traveling salesman problem) 是经典的组合优化问题之一,已经证明其属于 NP 完全问题,其时间复杂性在最坏情况下随问题规模呈指数方式增长,即找不到一个算法能保证在多项式时间内得到最优解。现实世界中许多问题都可归结为求解 TSP,例如计算机布线、车辆选路、机器人控制、PCB 板打孔等。TSP 可描述为:

给定费用矩阵 $C = (c_{ij}), i, j = (1, \dots, n)$, 其中 c_{ij} 代表从城市 i 到城市 j 的费用。确定从 1 到 n 的一个排列 (i_1, i_2, \dots, i_n) , 使得 $c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_n i_1}$ 最小。

根据费用矩阵的性质可以对 TSP 分类^[1]:

- 若对所有的 i, j 有 $c_{ij} = c_{ji}$, 则称问题是对称的, 否则称为非对称的;
- 若对所有的 i, j, k 有三角不等式 $c_{ik} \leq c_{ij} + c_{jk}$ 成立, 则称问题是可度量的;
- 若 c_{ij} 是平面上点之间的欧氏距离, 则称问题是欧氏的或几何的。

显然欧氏 TSP 是对称的和可度量的。Garey 等^[2]及文献^[3]证明欧氏 TSP 也是 NP-hard 问题。求解 TSP 的算法可以分为两类:(1)精确算法。它能保证在有限步内给出最优解,但是运行时间呈指数复杂度。切平面法(cutting-plane)是目前最有效的精确算法。(2)近似算法(也是启发式的搜索算法)。它在相对短的短时间内给出较好解,但不能保证是最优解。近似算法分为三类^[1]。环路构造算法通过某种策略在每一步增加新的城市,直到得到一个合法解为止,这类算法包括最邻近算法^[4]、Christofides 算法^[5]等;环路改进算法则通过各种交换操作进行局部寻优,改进已有回路,这类算法包括 λ -opt 算法^[6]、Lin-Kernighan^[7]及改进算法等;复合算法是将环路构造算法和环路改进算法结合起来使用。

* 收稿日期:2004-04-06

基金项目:国家部委资助项目(2003-5130801-1-3)

作者简介:潘亮(1973—),男,博士生。

1 基本思想及分析

主要研究欧氏平面上的 TSP 问题,简记为 E^2 -TSP,距离采用 L^2 范数定义。由 E^2 的性质,可以将问题的几何结构信息引入求解过程。利用计算几何求解 TSP 可参看文献 [8~10]。通过分析 E^2 -TSP 的特点,发现最优环路有一些较好的性质。记 S 是 E^2 -TSP 所给的点集,是 $|S|$ 的点数, $T_{opt}(S)$ 是问题的最优环路, $\mathcal{T}(S)$ 是问题的可行环路, $BCH(S)$ 是点集 S 的凸壳边界多边形, $|BCH(S)|$ 是其顶点数。

定理 1 最优解是简单多边形。

定理 2 最优解 $T_{opt}(S)$ 顺序经历 $BCH(S)$ 的顶点。

定理 1、2 的观点最先由 Flood^[11] 于 1956 年阐明,Quintas 和 Supnick 于 1965 年给予了证明。这反映了人类解决此类问题的过程,也符合“全局—局部”的观点^[12]。本文以此为出发点,根据凸壳和环路的关系定义子路径及相关概念。

定义 1 $\mathcal{T}(S)$ 被 $BCH(S)$ 顶点分割为 n 个部分,其中, $n = |BCH(S)|$ 。从 $BCH(S)$ 任一顶点 p 沿 $\mathcal{T}(S)$ 割下次碰到 $BCH(S)$ 的顶点 p' 时所经历的路径称为 $\mathcal{T}(S)$ 的子路径 ($sub(\mathcal{T}(S))$),其中, p 、 p' 称为 $sub(\mathcal{T}(S))$ 的端点。

定义 2 $\mathcal{T}(S)$ 的子路径 $sub_i(\mathcal{T}(S))$ 和 $sub_j(\mathcal{T}(S))$ ($i \neq j, i, j = 1, \dots, |BCH(S)|$) 若有一个端点相同,则称 $sub_i(\mathcal{T}(S))$ 和 $sub_j(\mathcal{T}(S))$ 是相邻的,否则称为是不相邻的。

定义 3 若 $sub_i(\mathcal{T}(S))$ 的端点是 $BCH(S)$ 上相邻的顶点,则称 $sub_i(\mathcal{T}(S))$ 是可行的;否则称 $sub_i(\mathcal{T}(S))$ 是不可行的。

定义 4 若 $sub_i(\mathcal{T}(S))$ 是可行的,且与 $sub_j(\mathcal{T}(S))$ ($i \neq j$) 除端点外没有交点,则称 $sub_i(\mathcal{T}(S))$ 是完全可行的。

定理 3 $T_{opt}(S)$ 的子路径 $sub_i(\mathcal{T}(S))$ ($i = 1, \dots, |BCH(S)|$) 是完全可行的。

证明 根据定义,结合定理 1、2 容易证明。

推论 1 $T_{opt}(S)$ 中不相邻的子路径 $sub_i(\mathcal{T}(S))$ ($i = 1, \dots, |BCH(S)|$) 不相交。

引理 1 若 $S = |BCH(S)|$ 则 $T_{opt}(S) = BCH(S)$ 。

证明 用反证法容易得证。

定理 4 若 $\mathcal{T}(S)$ 的子路径 $sub_i(\mathcal{T}(S))$ ($i = 1, \dots, n, n = |BCH(S)|$) 是可行的,则按照子路径相邻关系依次连接 $sub_i(\mathcal{T}(S))$ 并删除重复出现的端,可以构成可行环路 $T'(S)$,且 $T'(S)$ 是由 $sub_i(\mathcal{T}(S))$ ($i = 1, \dots, n, n = |BCH(S)|$) 所决定的 $(n-1)/2$ 条可能环路中最优的。

证明 从子路径的定义可知, $T'(S)$ 遍历 S ,且每个点只经过一次,故 $T'(S)$ 是可行环路。 n 个子路径有 $(n-1)/2$ 种连接方式,但不同的连接方式不会改变 $sub_i(\mathcal{T}(S))$ 的长度,按照题设条件可将有相邻关系的子路径映射成 $BCH(S)$ 上相邻的顶点而不影响不同环路之间的长度差异。由引理可知 $BCH(S)$ 是连接其顶点的最优环路,其顶点连接方式与 $T'(S)$ 的连接方式相同,故 $T'(S)$ 是 $(n-1)/2$ 条可能环路中的最优环路。

子路径可以看作点集 S 的分类。若能够保证 $\mathcal{T}(S)$ 所有子路径是可行的,则定理 4 给出了构造可行解的方法,并且构造的过程就是优化的过程。从以上定理和结论中可知,若在子路径构造过程中对点集 S 做了错误的划分,即使 $\mathcal{T}(S)$ 的所有子路径都是完全可行的和最优的,并且按照定理 4 的方法构造 $\mathcal{T}(S)$ 也不是最优的。因此一种近似算法若没有在子路径之间交换点的机制,则很难找到最优解。最近邹鹏等提出的多级归约算法^[13],其主体思想是对解进行多级归约,不断地降低问题规模,在小规模的实例上用成熟的算法得到较好的解,然后按归约的相反顺序逐步拼接,最终得到一个合法解。按照本文提出的概念,这种方法可以等效为在多个子路径之间交换点。对于随机性的近似算法,如遗传算法,文献 [14~17] 中提出的各种交叉、变异和其它操作由于不能保证子路径可行等性质,因此其搜索是随机盲目的。虽然 E^2 -TSP 的约束只有一项,即限制可行解遍历 S 且每个点只经过一次,但若将 $T_{opt}(S)$ 的必要条件作为一项约束,则在搜索过程中可以排除大量非优解,从而提高求解效率,而且对问题的最优解

没有影响。

2 算法设计

采用改进的遗传算法求解 E^2 -TSP 问题。首先基于 $BCH(S)$ 设计环路构造算法。计算凸壳 $BCH(S)$ 可以采用格雷厄姆算法、分治算法(其时间复杂度为 $O(n \log(n))$ ^[18]),作为初始化在构造算法之前完成。本文在计算凸壳时采用了开源的计算几何算法库 CGAL^[19](Computational Geometry Algorithm Library)。环路构造算法如下:

step 1 以 $BCH(S)$ 为初始环路 $T(S)$;

step 2 对 $T(S)$ 所有的边 $\overline{m_i n_i}$, 以及所有不在 $T(S)$ 上的点 r , 查找满足 $(c_{m_r} + c_{n_r} - c_{m_i n_i})$ 最小的 m_i, n_i, r ;

step 3 将 r 插入 $\overline{m_i n_i}$ 之间以更新环路 $T(S)$;

step 4 若 $T(S)$ 没有遍历所有的点, 则重复执行第 2 步。

构造算法可以理解为以 $BCH(S)$ 为起点, 按照距离增量最小的原则向内收缩直至遍历所有的点。算法复杂度大致为 $O(n^2)$ 。

任何一种有效的 TSP 近似算法都采用一定的启发信息或规则。本文以点集的 Delaunay 三角剖分图 $DT(S)$ 作为启发信息。虽然 Kantabutra^[20] 已经证实 E^2 -TSP 的最优解不一定是 $DT(S)$ 的子图, 此外 Michael Dillencourt 也提出了反例的构造方法, 但通过 TSPLIB^[21] 中的大量实例, 我们观察到对于 E^2 -TSP, 其最优解在 $DT(S)$ 上的概率很大, 表 1 是对 TSPLIB 中给出最优解的 E^2 -TSP 实例的分析结果。

表 1 TSPLIB 实例分析结果

Tab.1 The analysis result of some TSPLIB examples

实例名	规模	最优解在 DT 上的边数	%	实例名	规模	最优解在 DT 上的边数	%
eil51	51	51	100.0	eil101	101	99	98.01
berlin52	52	52	100.0	lin105	105	103	98.09
st70	70	69	98.57	ch130	130	129	99.23
eil76	76	76	100.0	ch150	150	150	100.0
pr76	76	74	97.37	tsp225	225	225	100.0
kroC100	100	100	100.0	a280	280	280	100.0
kroA100	100	99	99.0	pr1002	1002	999	99.7
kroD100	100	99	99.0				

算法初始化: 首先消除点集 S 中的重复点, 此项操作不会影响最优环路。例如 TSPLIB 中的 Djibouti-89 问题中就存在多组重复的点, 其次计算 $BCH(S)$ 和 $DT(S)$, 为环路构造算法和操作算子提供几何结构信息。

编码方式: 按照子路径的定义将可行环路编码为子路径集合, 集合规模根据 $BCH(S)$ 的顶点数确定。子路径采用动态数组编码, 其首尾元素为 $BCH(S)$ 上相邻的顶点。如图 1 所示。

群体初始化: 由于构造算法是确定性的, 为保证个体的多样性, 可先构造一条可行环路 $T(S)$, 然后对 $T(S)$ 的子路径按照不自交和沿 $DT(S)$ 经历的点尽量多的原则生成指定规模的种群。

选择算子: 采用按适应度比例复制的最优个体保留策略。

交叉算子: 按照最优环路的性质, 存在不可行子路径的环路一定不是最优的。而采用常规单点或多点交叉、部分映射交叉 PMX、顺序交叉 OX、循环交叉 CX、边重组交叉 EX 都不能保证子路径是可行的, 可以认为这些交叉算子产生的结果是错误的。利用构造算法和编码方式的特点, 本文采用子路径交叉的方法, 即只在端点相同的子路径之间进行交叉操作。设两条环路 $T_a(S), T_b(S)$ 按 $BCH(S)$ 顶点顺序对子路径 $sub_i(T_a(S))$ 和 $sub_j(T_b(S))$ 交叉, 形成环路 $T'_a(S), T'_b(S)$ 其中 $sub_i(T'_a(S)) = sub_j(T_b(S)), sub_j(T'_b(S)) = sub_i(T_a(S))$, 若 $sub_i(T_a(S))$ 和 $sub_j(T_b(S))$ 经历相同的点, 则完成交叉操作。否则以 $T'_a(S)$ 为例按以下步骤操作:

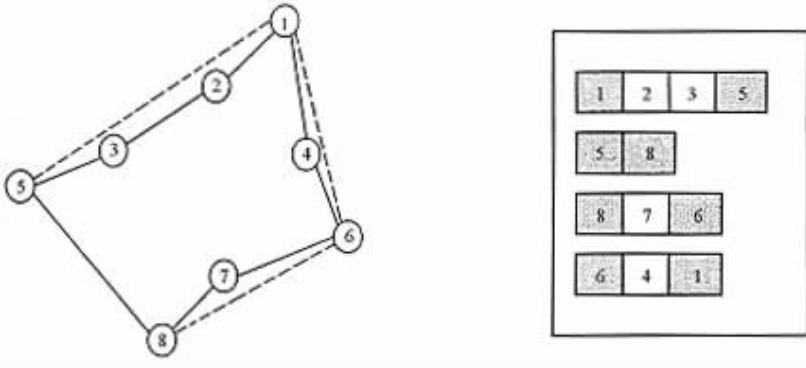


图1 环路编码方式

Fig.1 The encode method of tour

step 1 若 $T'_a(S)$ 中有未经历的点 p 则按照沿 $DT(S)$ 直接可达的原则将其加入相应子路径中。

step 2 若 $T'_a(S)$ 中有重复点 p 则按照 $sub_j(T'_a(S))$ 优先和沿 $DT(S)$ 直接可达的原则将其保留在所在子路径中,并删除其余重复的点 p 。

step 3 若 $T'_a(S)$ 是不可行的,则重复 step 1 和 step 2。

step 4 对被改变的子路径按照不自交的原则形成新的子路径。

交叉的目的是交换 $T_a(S), T_b(S)$ 的子路径,在保证环路可行的前提下尽量使其余子路径少受影响,同时使所有子路径是可行的。此算法可看作子路径单点交叉操作,也可将其修改为子路径多点交叉操作。

变异算子:根据最优化原理,最优环路其子路径也是最优的。由于子路径可认为是对 S 的分类,要找到最优解,首先要对 S 进行正确的分类,其次对分类结果(即子路径)进行优化。交叉算子更多的是对不同的分类结果进行组合,而变异算子则是改变分类并对子路径进行局部优化。

step 1 随机选择子路径 $sub_j(T(S))$

step 2 顺序选择 $sub_j(T(S))$ 中的非端点的路径点 p ,若不存在,则到 step 5。

step 3 若点 p 在 $DT(S)$ 上相邻的点都在 $sub_j(T(S))$ 上,则返回 step 2。

step 4 若点 p 有至少两个 $DT(S)$ 上相邻的点都在 $sub_j(T(S))$ ($j \neq i, i, j = 1, \dots, m$) 上,尝试将 p 从 $sub_j(T(S))$ 上删除,对 $sub_j(T(S))$ 所有的边 $\overline{m_i n_i}$,按照 $(c_{m_i} + c_{n_i} - c_{m_i n_i})$ 最小的原则,将 p 加入 $sub_j(T(S))$,从而形成可行环路 $T_j(S)$ 。计算 $\Delta_j = cost(T) - cost(T_j)$,若 $\Delta_j < 0, i, j = 1, \dots, m$,则返回 step 2,否则,令 $T(S) = T_k(S)$,其中 $\Delta_k = \max(\Delta_1, \dots, \Delta_m)$,返回 step 2。

step 5 对发生变化的子路径执行 2-opt 算法,直至子路径的长度不再减小。

可以看出变异算子满足子路径的可行性。变异算子第 3 步的目的是避免选择子路径凸壳内的点,以提高子路径完全可行的概率。第 4 步是按照相邻的原则在子路径之间交换点,以改变分类,并保证满足正向收益原则,也可看作是子路径之间的 2-opt 操作。第 5 步是对受影响的子路径进行局部优化。从实验中可以看出,由于 $DT(S)$ 的性质限制了点的自由度,交换时满足条件的子路径通常是很少的。如图 2 所示。

控制参数选择:TSP 问题的凸壳点数相对较少,为保持子路径的多样性,建议种群规模至少为 $|BCH(S)|$ 。本文采用的规模为 $\max(20, 2|BCH(S)|)$ 。交叉概率 $p_c = 0.6$,变异概率 $p_m = 0.2$ 。

停止准则:若当前最优个体在一定演化代数 n 内未得到改进,则算法停止。本文取 $n = \frac{|BCH(S)|}{p_m}$ 。

3 试验结果

在表 2 给出的环境中实现了上述算法。

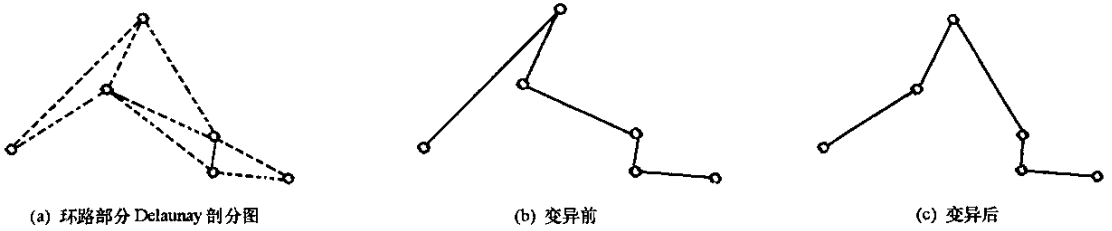


图 2 变异算子

Fig.2 The mutation operator

表 2 实验环境配置

Tab.2 The computing environment configure of algorithm

CPU	内存	操作系统	开发环境	编译器	软件包
Intel P4 1.4G	256M DDR	Win2000	cygwin	gcc 2.95	CGAL

计算 $BCH(S)$ 、 $DT(S)$ 以及几何查询等采用了 CGAL^[19]。针对表 1 中的部分实例及 TSPLIB^[21] 中的 Djibouti-89、Qatar-194 和 China-144 问题求解结果如表 3 所示。其中中国 144 城市 TSP 的结果如图 3 所示。

表 3 算例结果

Tab.3 The result of examples by algorithm

实例名	运行次数	找到最优解的次数	最优解长度	$BCH(S)$	平均运行时间(s)	平均演化代数	最大演化代数
eil51	20	20	426	8	1.9	158	192
berlin52	20	20	7541	8	2.1	177	252
st70	20	20	675	10	1.0	117	162
eil76	20	20	538	10	2.1	210	257
pr76	20	20	108159	4	1.7	147	173
kroC100	20	20	20749	11	1.6	121	178
kroA100	20	20	21282	12	1.6	146	189
kroD100	20	20	21294	14	1.1	110	162
eil101	20	20	629	10	2.6	236	421
lin105	20	20	14379	5	0.8	122	176
ch130	20	20	6110	10	1.8	172	360
ch150	20	20	6528	15	2.0	193	277
a280	20	20	2583	12	4.1	368	497
Djibouti-89	20	20	6656	8	0.8	83	112
Qatar-194	20	20	9352	15	2.2	328	388
China-144	20	20	30355	10	2.5	291	392

从试验结果可以看出问题求解时间与规模并不是一致的。构造算法产生的环路其所有的子路径都是完全可行的,对于一些实例(如 Djibouti-89)构造的初始环路与最优环路的差别就已经很小。 $T_{opt}(S)$ 中一些较短的子路径通常在构造时就已经出现,并且很少受变异算子的影响。

4 结论

根据最优环路以及子路径的概念和性质,以问题点集的凸壳设计环路构造算法,利用遗传算法全局

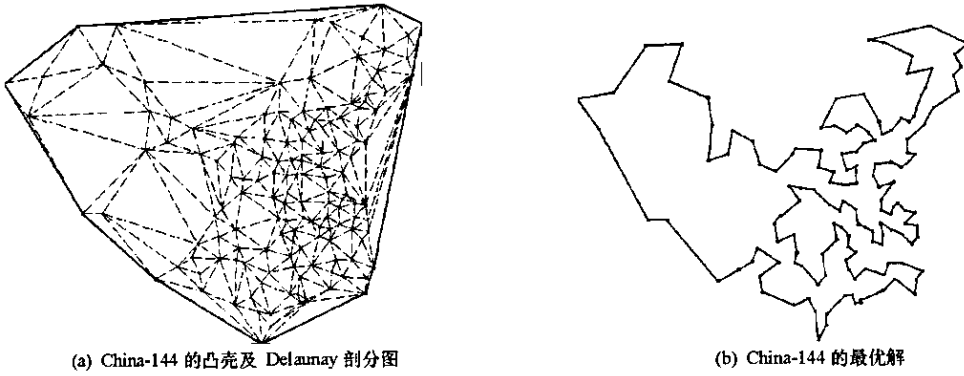


图3 China-144 算例结果

Fig.3 The result tour of China-144 TSP

寻优的能力并以 Delaunay 三角剖分图为启发信息可以有效解决欧氏平面 TSP。通过实验验证了利用问题特点和几何结构信息可以有效减小搜索空间、提高搜索的质量和效率。

参考文献：

- [1] Helsgaun K. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic[J]. European Journal of Operational Research ,2000 , 126(1).
- [2] Garey M R , Graham R L , Johnson D S. Some NP-Complete Geometric Problems[A]. In Proc. ACM Symp on Theory of Computing , 1976.
- [3] Papadimitriou C H. Euclidean TSP is NP-complete[R]. TCS , 4 :237 - 244 , 1977.
- [4] Rosenkrantz D E , Stearns R E , Lewis P M , II . An Analysis of Several Heuristics for the Traveling Salesman Problem[J]. SIAM J. Comput. , 1977 , 6.
- [5] Clarke G , Wright J W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points[J]. Operations Research , 1964 , 12.
- [6] Lin S. Computer Solutions of the Traveling Salesman Problem[J]. Bell System Tech. J. , 1965 , 44.
- [7] Lin S , Kernighan B W. An Effective Heuristic Algorithm for the Traveling-Salesman Problem[J]. Oper. Res. , 1973 , 21.
- [8] 周培德. 寻求中国货郎担问题最短回路的多项式时间算法[J]. 北京理工大学学报 , 2000 , 20(2).
- [9] Zhou Peide. Geometric Method for Solving TS Problem[J]. Journal of Software(in Chinese) , 1995 , 6(7).
- [10] Golden B , Bodin L , Doyle T , et al. Approximate Travelling Salesman Algorithms[J]. Operations Research , 1980 , 28 :694 - 711.
- [11] Flood M M. The travelling salesman problem[J]. Operations Research , 1996 , 44 :61 - 75.
- [12] MacGregor J N , Ormerod T C. Human Performance on the Traveling Salesman Problem[J]. Perception Psychophysics , 1996 , 58 :527 - 539.
- [13] 邹鹏,等. 求解 TSP 问题的多级归约算法[J]. 软件学报 , 2003 , 14(1).
- [14] 唐立新. 旅行商问题(TSP)的改进遗传算法[J]. 东北大学学报 , 1999 (1).
- [15] 陈国良,等. 遗传算法及其应用[M]. 北京 :人民邮电出版社 , 1996.
- [16] Goldberg D E , Lingle R. Alleles , Loci and Traveling Salesman Problem[A]. Proc. Int. Conf. On Genetic Algorithms and Their Applications , Carnegie-Mellon University , 1985.
- [17] 刘勇,康立山,陈毓屏. 非数值并行算法(第二册)[M]. 北京 :科学出版社 , 1995.
- [18] 周培德. 计算几何——算法分析与设计[M]. 北京 :清华大学出版社 , 2000.
- [19] <http://www.cgal.org>[EB].
- [20] Viti Kantabutra. Traveling Salesman Cycles are not Always Subgraphs of Voronoi Dual[J]. Information Processing Letters , 1983 , 16(1).
- [21] <http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/>[EB].

