

TSP 问题的脂肪计算复杂性与启发式算法设计*

江贺^{1,2+}, 胡燕¹, 李强¹, 于红¹

¹(大连理工大学 软件学院, 辽宁 大连 116621)

²(中国科学院 软件研究所 计算机科学国家重点实验室, 北京 100190)

Fat Computational Complexity and Heuristic Design for the TSP

JIANG He^{1,2+}, HU Yan¹, LI Qiang¹, YU Hong¹

¹(School of Software, Dalian University of Technology, Dalian 116621, China)

²(State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: jianghe@dlut.edu.cn

Jiang H, Hu Y, Li Q, Yu H. Fat computational complexity and heuristic design for the TSP. *Journal of Software*, 2009,20(9):2344–2351. <http://www.jos.org.cn/1000-9825/3418.htm>

Abstract: The traveling salesman problem (TSP) is one of the classic NP-Hard combinatorial optimization problems. Efficient heuristics for the TSP have been the focus in research of computer science at all times. As a new tool to describe the structure of the TSP, the fat is essential for heuristic algorithm design. Since research on the fat is still at the beginning stage, It still lacks theoretical results and related heuristic algorithms. In this paper Firstly, the fat computational complexity for the TSP is investigated. It's proved that it is NP-Hard to obtain the fat of the TSP through construction of biased instances, i.e., there is no algorithm to get the full fat of the TSP in polynomial time on the assumption $P \neq NP$. Furthermore, a meta-heuristic—dynamic candidate set search (DCSS) was proposed by analysis of the relationship between local optimal solutions and the fat. And the DCSS was applied to the LKH, one of the best existing algorithms for the TSP to improve it. Extensively wide experimental results on typical instances from the benchmark—TSPLIB indicate that the improved algorithm has a better performance than the LKH in terms of solution quality. This new fat-based heuristic shows a new way for other NP-hard problems.

Key words: traveling salesman problem; NP-hard; fat; heuristic algorithm

摘要: 旅行商问题(traveling salesman problem,简称 TSP)是经典的 NP-难解组合优化问题之一,求解它的高效启发式算法一直是计算机科学研究的热点.脂肪作为描述 TSP 结构特征的新工具,对启发式算法设计具有重要意义.目前,TSP 问题的脂肪研究还处于初始阶段,缺乏理论分析结果及相关的启发式算法.首先分析了 TSP 问题的脂肪计算复杂性,通过构造偏移实例的技巧,证明了获取 TSP 的脂肪是 NP-难解的,即在 $P \neq NP$ 的假设下,不存在算法可以在多项式时间内获得完整脂肪.在此基础上,通过分析 TSP 问题局部最优解与脂肪的关系,给出了求解 TSP 问题的元启发式算法——动态候选集搜索(dynamic candidate set search,简称 DCSS)算法.利用该算法,改进了目前求解 TSP 问题

* Supported by the National Natural Science Foundation of China under Grant No.60805024 (国家自然科学基金); the National Research Foundation for the Doctoral Program of the Ministry of Education of China under Grant No.20070141020 (国家教育部博士点基金)

Received 2008-04-08; Accepted 2008-07-02

最好算法之一的 LKH.TSPLIB 典型实例的实验结果表明,改进后的算法在解的质量上有较为明显的提高.新的基于脂肪的启发式算法对于求解其他 NP-难解问题具有一定的参考价值.

关键词: 旅行商问题;NP-难解;脂肪;启发式

中图法分类号: TP18 文献标识码: A

旅行商问题 TSP(traveling salesman problem)是经典的 NP-难解组合优化问题之一,在交通、网络、电路设计等方面有着广泛的应用背景.根据计算复杂性理论可知,对于 NP-难解问题,除非 $P=NP$,否则不存在多项式时间内得到最优解的精确算法.因而对大规模实例,人们寻求在可接受的计算时间内能得到(在解的性能上)可接受近似解的启发式算法(heuristic).现有 TSP 问题的启发式算法可以分为两类:简单启发式算法(simple-heuristic),包括贪心算法、局部搜索算法(如 2-OPT^[1],3-OPT^[2],LK^[3],LKH^[4])等;元启发式算法(meta-heuristic),包括模拟退火(如 CLK^[5])、遗传算法^[6]、蚁群算法^[7]、多级归约^[8,9]、人工免疫^[10]等.大部分启发式算法以局部搜索为基础,而目前被认为最简单有效的局部搜索算法是 Helsgaun 在经典 LK 算法的基础上发展出来的 LKH 算法^[4].

骨架(backbone)和脂肪(fat)作为设计启发式算法的有效工具,是近年来算法研究的热点领域.骨架是指一个问题实例的所有全局最优解变量的交集,它对于衡量 NP-难解问题的难度和启发式算法设计具有重要意义.Zhang^[11]发现,骨架规模的变化显著地影响非对称旅行商问题 ATSP(asymmetric tsp)的求解难度.Schneider^[12]和邹鹏^[9]几乎同时发现,TSP 的局部最优解交集与骨架非常相似,因此可将局部最优解交集作为近似骨架,并利用近似骨架对实例进行归约,得到了求解 TSP 问题的多级归约算法.Zhang 等人^[13]则利用局部最优解交集来引导局部搜索,给出了骨架导向 LK 算法.与骨架不同,脂肪的研究还远不充分.脂肪是由 Sharlee 等人^[14]首先提出的,它是指一个问题实例中不属于任何一个全局最优解的变量集合.由于脂肪的规模通常要远大于骨架,若能有效获取脂肪并将其从实例中剔除,则可以显著缩小搜索空间,从而提高各种基于局部搜索的算法的求解性能.然而,脂肪作为 TSP 问题的新特征,其研究尚处于起步阶段,仅有的成果是 Sharlee 等人^[14]给出的用于获取 ATSP 部分脂肪的限界交叉(limit-crossing)算法(时间复杂度为 $O(n^4)$).脂肪研究还存在以下亟待解决的问题:(1) 缺乏系统的理论分析结果,是否存在多项式算法可以获取完整脂肪?(2) 缺乏利用脂肪进行启发式算法设计的结果.

针对上述问题,本文做了两方面的工作:(1)理论上证明了获取 TSP 的脂肪是属于 NP-难解的.在证明过程中,首先给出了 TSP 实例的偏移实例的定义,并证明了其恰好只有唯一全局最优解,从而由脂肪可以在多项式时间内构造其全局最优解.随后,通过分析一般 TSP 实例与偏移实例的关系发现,任意 TSP 实例均可以构造出相应的偏移实例,而偏移实例的全局最优解恰好是原实例的一个全局最优解,并由此完成脂肪的计算复杂性分析.(2)利用脂肪进行启发式算法设计.由于边在局部最优解中出现的频率与脂肪关系密切,本文给出了新的元启发式算法——动态候选集搜索算法 DCSS(dynamic candidate set search).DCSS 根据边在局部最优解中出现的频率动态调整候选集中边的位序,将出现频率高的边调整到候选集的前面,并在调整后的候选集中调用局部搜索算子继续求解.本文利用 DCSS 算法对目前 TSP 最好算法之一的 LKH 进行改进,并在 TSPLIB 的多组不同规模的典型实例上进行实验比较.实验结果表明,改进后的算法在全局最优解数量、解的平均质量等方面与原 LKH 算法相比有较为明显的提高.

1 相关定义及性质

本节给出文中所使用的一些定义及性质.

定义 1. 给定无向加权图 $G=(V,E,w)$, V 为顶点集, $|V|=n$ 表示顶点数目, E 为边集, $|E|=m$ 表示边的数目, $w:E \rightarrow R^+$ 为边的权值函数. TSP 实例(记为 $TSP(V,E,w)$)的可行解为 $t=(i_1, i_2, \dots, i_n)$, 其中, $\forall 1 \leq j \neq k \leq n, i_j \neq i_k$ 且对 $1 \leq j \leq n$, 边 $(i_j, i_{(j \bmod n)+1}) \in E$. 可行解 t 的目标函数值定义为 $w(t) = \sum_{j=1}^n w(i_j, i_{(j \bmod n)+1})$, TSP 问题的目标是寻求目标函数值最小的解 t^* , 即满足 $w(t^*) = \min_{t \in \Pi(G)} (w(t))$, 其中, $\Pi(G)$ 为 G 中所有 TSP 环路的集合.

定义 2. 给定无向加权图 $G=(V,E,w)$,对于任意边 $(j,k) \in E$ (不妨假设 $w(j,k) \in \mathbb{Z}^{**}$),定义边的偏移权值为 $\hat{w}(j,k) = w(j,k) + 1/2^{j+kn+|j-k|}$,本文称 $TSP(V,E,\hat{w})$ 为 $TSP(V,E,w)$ 的偏移 TSP 实例.对于 $TSP(V,E,\hat{w})$ 的可行解 $t=\langle i_1,i_2,\dots,i_n \rangle$,它的目标函数值记为 $\hat{w}(t) = \sum_{j=1}^n \hat{w}(i_j, i_{(j \bmod n)+1})$.

容易验证,对于任意的两条边 $(j,k),(p,q)$,除非 (1) $j=p,k=q$ 或者 (2) $j=q,k=p$,否则 $jn+kn+|j-k| \neq pn+qn+|p-q|$.显然,给定一个 TSP 实例,构造其偏移实例的时间复杂度为 $O(m)$.根据定义 2,易知性质 1 成立:

性质 1. 给定无向加权图 $G=(V,E,w)$,偏移 TSP 实例 $TSP(V,E,\hat{w})$ 的任意可行解也是 $TSP(V,E,w)$ 的可行解.

定义 3. 给定 TSP 实例 $TSP(V,E,w)$ 的可行解 $t=\langle i_1,i_2,\dots,i_n \rangle$,记 t 所包含的边的集合为 $E(t)=\{i_j,i_{(j \bmod n)+1} | 1 \leq j \leq n\}$.实例 $TSP(V,E,w)$ 存在有限多个全局最优解,记为 t_1^*,t_2^*,\dots,t_s^* ,实例 $TSP(V,E,w)$ 的脂肪定义为 $F(V,E,w) = E - E(t_1^*) \cup E(t_2^*) \cup \dots \cup E(t_s^*)$,其规模记为 $|F(V,E,w)|$.

定义 4. 给定 TSP 实例 $TSP(V,E,w)$,若它有且仅有一个全局最优解,则称该实例是唯一全局最优解实例.

根据定义 3、定义 4,易知以下性质成立:

性质 2. 给定 TSP 实例 $TSP(V,E,w)$,它的脂肪 $F(V,E,w)$ 满足 $0 \leq |F(V,E,w)| \leq m$.当 $|F(V,E,w)|=m-n$ 时,该实例为唯一全局最优解实例.

定义 5. TSP 问题局部搜索算法的基本规则之一是,将与顶点关联的可能包含在合法环路上的边限制为有限的若干条,这些边的集合称为顶点候选集(vertex candidate set),对于顶点 i ,记其候选集为 $C(i)$.给定 TSP 实例 $TSP(V,E,w)$,通常将所有这样的边的集合称为实例候选集(instance candidate set),记为 $IC(V,E,w) = \bigcup_{i \in V} C(i)$,顶点 i 的候选集 $C(i)$ 中边的位置称为边的位序(rank).

2 脂肪的计算复杂性

本节给出 TSP 问题脂肪的计算复杂性分析.首先证明偏移 TSP 实例是唯一全局最优解实例,然后在分析偏移 TSP 实例与一般 TSP 实例之间关系的基础上证明获取 TSP 问题的脂肪是 NP-难解的.

定理 1. 给定 TSP 实例 $TSP(V,E,w)$,若 $\forall (i_j,i_k) \in E$ 满足 $w(i_j,i_k) \in \mathbb{Z}$,则它的偏移 TSP 实例 $TSP(V,E,\hat{w})$ 是唯一全局最优解实例.

证明:对于实例 $TSP(V,E,\hat{w})$ 的任意两个不同的解 $t_1 = \langle i_1^1, i_2^1, \dots, i_n^1 \rangle$ 和 $t_2 = \langle i_1^2, i_2^2, \dots, i_n^2 \rangle$,由 $t_1 \neq t_2$ 可知, $E(t_1) \neq E(t_2)$,故存在 $(j^*, k^*) \in t_1$ 且 $(j^*, k^*) \notin t_2$.在使用二进制表示权值的情况下, $\hat{w}(t_1)$ 在小数位的第 $j^*n + k^*n + |j^* - k^*|$ 位为 1,而 $\hat{w}(t_2)$ 在该位为 0.故 $\hat{w}(t_1) \neq \hat{w}(t_2)$,原命题得证. □

引理 1. 给定 TSP 实例 $TSP(V,E,w)$,若 $\forall (i_j,i_k) \in E$ 满足 $w(i_j,i_k) \in \mathbb{Z}$,则给定任意两个不同的可行解 $t_1 = \langle i_1^1, i_2^1, \dots, i_n^1 \rangle$ 和 $t_2 = \langle i_1^2, i_2^2, \dots, i_n^2 \rangle$,若 $w(t_1) < w(t_2)$,则偏移 TSP 实例 $TSP(V,E,\hat{w})$ 满足 $\hat{w}(t_1) < \hat{w}(t_2)$.

证明:对于 TSP 实例 $TSP(V,E,w)$ 的任意两个不同的可行解 t_1 和 t_2 ,若 $w(t_1) < w(t_2)$,则根据 $w(i_j,i_k) \in \mathbb{Z} (\forall (i_j,i_k) \in E)$ 可知, $w(t_2) - w(t_1) \geq 1$.而根据偏移 TSP 实例的定义可知:

$$0 < \hat{w}(t_1) - w(t_1) = \sum_{j=1}^n 1/2^{i_j^{n+1} + i_{(j \bmod n)+1}^{n+1} + |i_j^1 - i_{(j \bmod n)+1}^1|} < \sum_{j=1}^n 1/2^{3n} = n/2^{3n} < 1.$$

同理可知, $0 < \hat{w}(t_2) - w(t_2) < 1$,故有下式成立:

$$\hat{w}(t_2) - \hat{w}(t_1) = w(t_2) - w(t_1) + (\hat{w}(t_2) - w(t_2)) - (\hat{w}(t_1) - w(t_1)) > w(t_2) - w(t_1) - 1 \geq 0.$$

原命题得证. □

引理 2. 给定 TSP 实例 $TSP(V,E,w)$,若 $\forall (i_j,i_k) \in E$ 满足 $w(i_j,i_k) \in \mathbb{Z}$,则它的偏移 TSP 实例 $TSP(V,E,\hat{w})$ 的全局最优解也是 $TSP(V,E,w)$ 的全局最优解.

证明:首先, $TSP(V,E,\hat{w})$ 的全局最优解是 $TSP(V,E,w)$ 的一个可行解.下面用反证法证明.假设实例 $TSP(V,E,\hat{w})$ 的全局最优解 t^* 不是 $TSP(V,E,w)$ 的全局最优解,则存在一个解 t ,使得 $w(t) < w(t^*)$ 成立.而根据引理 1,

** 对于边的权值不为整数的实例,可以通过同比例放大边权值,把原实例转化为一个边权值均为整数的新实例.新实例的全局最优解与原实例相同.

可知, $\hat{w}(t) < \hat{w}(t^*)$, 与命题假设矛盾, 原命题得证. \square

引理 3. 给定一个唯一全局最优解实例 $TSP(V, E, w)$ 及该实例的脂肪 $F(V, E, w)$, 可以在 $O(m)$ 内求出这个实例的最优解.

证明: 不妨记实例 $TSP(V, E, w)$ 的唯一全局最优解为 t^* , 则根据定义 3 有 $F(V, E, w) = E - E(t^*)$, 故 $E(t^*) = E - F(V, E, w)$. 因此, 可以如下构造算法: 从任意顶点出发, 判断该顶点邻接边是否属于脂肪, 若不属于则标记该边, 并取该边的另一端点作为新的当前顶点, 再判断该顶点的未被标记的邻接边是否属于脂肪, 依次进行, 直到每一个顶点均被遍历过一次. 每条边最多遍历一次, 因此算法的时间复杂度为 $O(m)$, 命题得证. \square

定理 2. 在 $P \neq NP$ 的假设下, 不存在多项式时间算法可以获得 TSP 问题的完整脂肪.

证明: 反证法. 假设原命题不成立, 则存在算法 Γ 可以在多项式时间内(记为 $p(n)$)得到 TSP 问题的完整脂肪.

下面将构造多项式时间算法获得其全局最优解, 从而说明矛盾的存在.

给定任意的 TSP 实例 $TSP(V, E, w)$ (不妨设 $w(j, k) \in \mathbb{Z} (\forall (j, k) \in E)$ (对于边的权值不为整数的实例, 可按照脚注**同样处理)), 构造 $TSP(V, E, w)$ 的求解算法:

- (1) 根据定义 2 可知, 由 $TSP(V, E, w)$ 可在 $O(m)$ 时间内构造偏移实例 $TSP(V, E, \hat{w})$;
- (2) 因 $TSP(V, E, \hat{w})$ 也属于 TSP 问题, 按照假设, 可利用算法 Γ 在 $P(n)$ 时间内得到其完整的脂肪 $F(V, E, \hat{w})$;
- (3) 因 $TSP(V, E, \hat{w})$ 属于唯一全局最优解实例, 且已获得 $F(V, E, \hat{w})$, 根据引理 3, 可以找到一个算法, 在 $O(m)$ 时间得到 $TSP(V, E, \hat{w})$ 的全局最优解 t^* ; 而根据引理 2, t^* 即为 $TSP(V, E, w)$ 的全局最优解.

经过步骤(1)~步骤(3)可以得到实例 $TSP(V, E, w)$ 的全局最优解, 总的时间复杂度为 $O(m) + p(n)$, 即可以在多项式时间内找到任意 TSP 问题实例的全局最优解. 与 TSP 问题属于 NP-难解矛盾, 原命题得证. \square

3 动态候选集搜索算法

3.1 局部最优解的特征与 DCSS 算法

本文将从候选集入手, 通过逼近脂肪的手段给出一种新的元启发式算法——动态候选集搜索算法 DCSS.

根据定理 2 可知, 在 $P \neq NP$ 的假设下, 不存在多项式时间内算法可以获得 TSP 问题的脂肪. 而 Boese 等人^[15]发现, TSP 问题的局部最优解与全局最优解之间存在大约 80% 的重合边. 这一现象被称为“大坑”猜想. 因此, 可以从局部最优解与全局最优解的关系入手, 获取逼近脂肪的途径. 本文以目前国际上求解 TSP 问题最常用的 LK 算法为例, 统计了 TSPLIB 上典型实例的局部最优解与脂肪的关系. 图 1 给出了实验统计结果, 其中, 图 1(a) 的横坐标表示边在局部最优解中出现的频率, 纵坐标表示边属于脂肪的比例. 比如, 坐标点 (x, y) 表示在 10 个局部最优解中出现的频率为 x 次的边, 属于脂肪的比例为 y . 图 1(b) 给出了边的数量(归一化后)随其在局部最优解中出现的频率的变化情况. 从图中可以发现, 随着边在局部最优解中出现频率的增加, 属于脂肪的边比例迅速降低. 同时, 局部最优解中出现 10 次边的数量接近实例规模的 60%~80%. 根据这一性质, 本文对启发式算法中的候选集进行动态调整, 给出了新的元启发式算法——DCSS.

算法. DCSS.

输入: 实例 $TSP(V, E, w)$, 参数 κ , 现有求解 TSP 问题局部搜索算子 A ;

输出: 解 t^* .

begin

(1) 创建初始候选集 $IC(V, E, w)$

(2) for $j=1$ to κ do

(2.1) 调用算子 A , 利用候选集 $IC(V, E, w)$ 得到实例 $TSP(V, E, w)$ 的解 t_j

(2.2) 根据解 t_j , 修改候选集中各边在局部最优解中出现的频率信息

(3) 调整候选集 $IC(V, E, w)$ 中各边的位序, 得到新的候选集 $IC'(V, E, w)$

(4) 利用新候选集 $IC'(V, E, w)$, 调用算法 A 求解 $TSP(V, E, w)$, 得到解 t^*

end

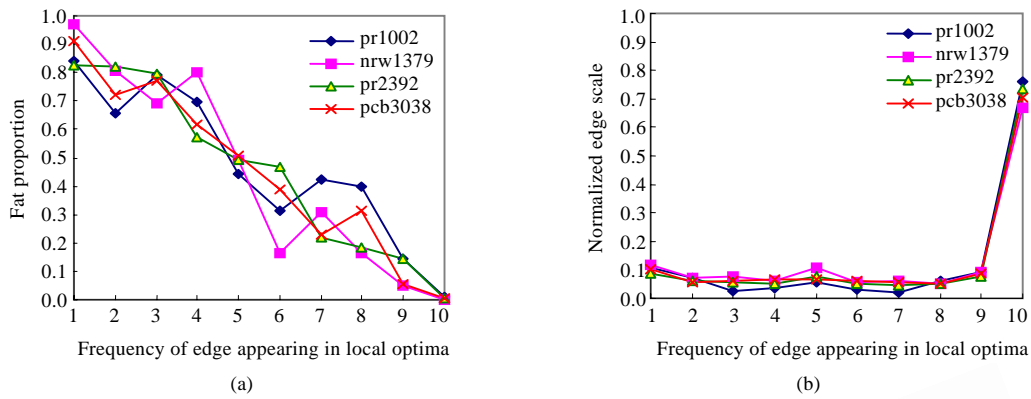


Fig.1 Relationship between fat and local optima in typical TSP instances

图 1 典型 TSP 实例的局部最优解的边与脂肪的关系

算法 DCSS 分为两部分:前一部分是在初始候选集上,利用现有的局部搜索算子 A 求得 TSP 实例的多个局部最优解;后一部分根据边在局部最优解中出现的频率调整候选集中边的位序,将出现频率高的边调整到候选集的前面,并在调整后的候选集中调用算子 A 继续求解.图 2 给出了算法 DCSS 的示意图.图 2(a)为初始候选集,其中,虚线部分为候选集中的边,在每一个顶点周围的虚线上标注的数字代表候选集中边的位序(需要注意的是,候选集中的边并非对称,即一条边可能只是该边的一个顶点的候选边,而非另外一个顶点的候选边).以图 2(a)中箭头所指向的顶点为例,它的候选集中包含 3 条边.图 2(b)~图 2(d)分别代表利用初始候选集所获得的不同局部最优解.图 2(e)表示根据 3 个局部最优解对于候选集调整后获得新候选集.同样以箭头所指向的顶点为例,它的候选集根据 3 个局部最优解中边出现的概率进行调整,将候选集中那些频繁在局部最优解中出现的边的位序提前,同时,最终的候选集中边的组成也发生了变化.图 2(f)为利用新候选集获得最终解.

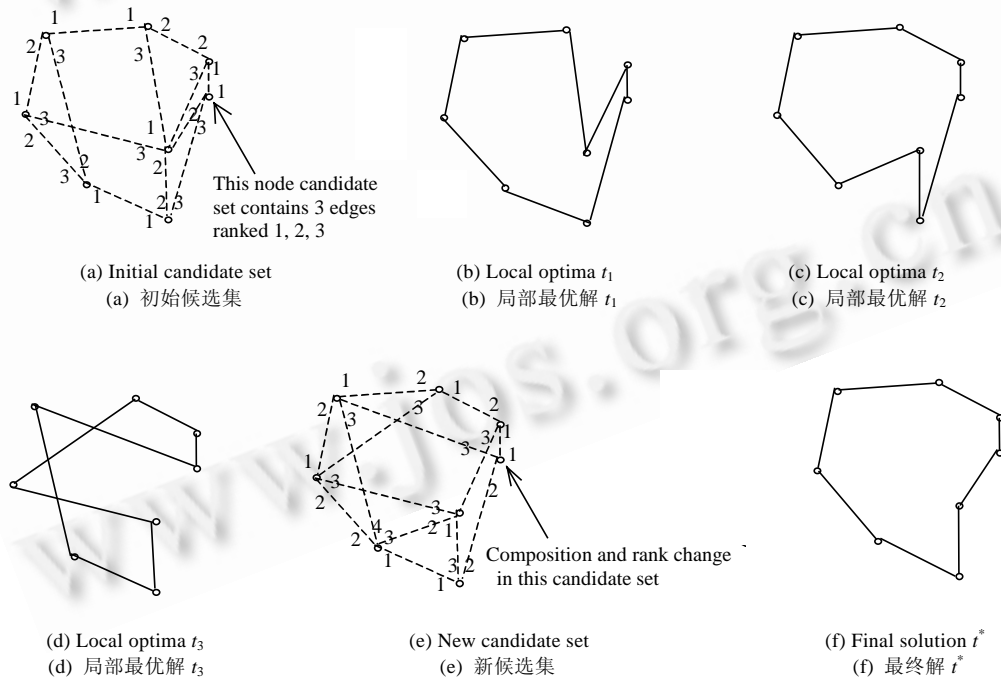


Fig.2 Illustration of algorithm DCSS

图 2 算法 DCSS 示意图

算法 DCSS 的运行时间复杂度分析如下:记步骤(1)的时间复杂度为 $p(n)$ (常见的候选集定义方法(如最近邻、 α -nearness 候选集)的时间复杂度均为 $O(n^2)$);记步骤(2.1)中局部搜索算子的时间复杂度为 $q(n)$,步骤(2.2)中修改候选集中边在局部最优解中出现的频率信息的时间复杂度为 $O(m)$,故此步骤(2)所需时间复杂度为 $\kappa q(n)+O(m)$.由于常见 TSP 局部搜索算子(如 2-opt,3-opt 和 LK)的时间复杂度均不低于 $O(n^2)$ (2-opt,3-opt 和 LK 算子的时间复杂度分别为 $O(n^2)$, $O(n^3)$ 和 $O(n^5)$),故步骤(2)所需时间复杂度可写为 $\kappa q(n)$.步骤(3)是对各个顶点候选集按照边在局部最优解中出现的频率高低重新排序,在使用快速排序的情况下,顶点 i 的候选集 $C(i)$ 所需时间为 $O(|C(i)|\log(|C(i)|))$,整个实例的候选集的调整所需时间则为 $\sum_{i \in V} O(|C(i)|\log(|C(i)|))$;步骤(4)中局部搜索的时间复杂度为 $q(n)$.故 DCSS 所需总的时间复杂度为 $p(n) + \sum_{i \in V} O(|C(i)|\log(|C(i)|)) + (\kappa+1)q(n)$.

3.2 DCSS-LKH算法

首先简要介绍算法 LKH 的流程,然后将它与 DCSS 相结合,给出 DCSS-LKH 算法.

LKH 算法是 Helsgaun 在经典 LK 算法的基础上提出的高效局部搜索算法,是目前已知的求解 TSP 问题的最好算法之一.LKH 算法对经典 LK 算法的改进主要是两个方面:

- (1) 用 α -nearness 候选集代替最近邻候选集;
- (2) 使用更加复杂、更大的搜索步数求解.

算法 LKH 的基本框架是:(1) 生成 TSP 问题的 α -nearness 候选集;(2) 根据当前最好的两个解及 α -nearness 候选集来选择初始解;(3) 在候选集上调用 LK 算子求得局部最优解 t_j ;(4) 若解 t_j 优于当前最优解,则令 t_j 为当前最优解;(5) 重复步骤(2)~步骤(4),直至算法结束.

算法. DCSS-LKH.

输入:实例 $TSP(V,E,w)$,参数 κ ,局部搜索算子 LK ;

输出:解 t^* .

begin

(1) 创建初始候选集 $IC(V,E,w),w(t^*)=+\infty$

(2) for $j=1$ to $\kappa/2$ do

(2.1) 调用 LK 算子,利用候选集 $IC(V,E,w)$ 得到实例 $TSP(V,E,w)$ 的解 t_j

(2.2) if $w(t_j) < w(t^*)$ then $t^* = t_j$

(2.3) 根据解 t_j ,修改候选集中各边在局部最优解中出现的频率信息

(3) 调整候选集 $IC(V,E,w)$ 中各边的位序,得到新的候选集 $IC'(V,E,w)$

(4) for $j=1$ to $\kappa/2$ do

(4.1) 调用 LK 算子,利用新候选集 $IC'(V,E,w)$ 得到 $TSP(V,E,w)$ 的解 t_j

(4.2) if $w(t_j) < w(t^*)$ then $t^* = t_j$

end

本文将 DCSS 与 LKH 相结合,给出了算法 DCSS-LKH.算法可以分成 3 部分:第 1 部分类似原局部搜索算法 LKH,所不同的是,此时的循环次数为 $\kappa/2$,另外还增加了修改候选集中各边在局部最优解出现的频率信息动作(这时候选集与原算法相同);第 2 部分是调整候选集中各边的位序;第 3 部分类似第 1 部分,所不同的是没有调整候选集中频率信息的操作.另外,局部搜索算子 LK 是在新候选集 $IC'(V,E,w)$ 上求解.算法中使用 α -nearness 候选集作为初始候选集.

算法 DCSS-LKH 与原算法相比,所增加的额外运行时间包括:步骤(2.3)中修改频率信息所需的 $O(m)$ 时间;步骤(3)调整候选集所需的 $\sum_{i \in V} O(|C(i)|\log(|C(i)|))$ 时间.由于 TSP 局部搜索算子 LK 的时间复杂度为 $O(n^5)$,故此 DCSS-LKH 总的时间复杂度与原算法相同.

4 实验结果及分析

目前,研究人员通常采用实验分析方法评价算法性能.在实验分析中,算法运行在一系列的实例上,通过衡量解与全局最优解之间的距离来评价算法的性能.本文实现了 DCSS-LKH 算法(取 $\kappa=n$),实验环境是 AMD Opteron 246 双核微处理器,2G 内存,Redhat Linux 9.0 操作系统.为了比较算法的性能,本文在相同的实验环境下运行了 LKH 算法,并在 TSPLIB 中相同的实例上测试了算法(分别运行 20 次的结果).表 1 给出了实验结果.其中, n_{opt} 为算法得到最优解的个数; τ_{aver} , τ_{min} , τ_{max} 分别为算法得到的解与最优解距离的均值、最小值、最大值; t_{aver} 表示运行时间的均值.本文用黑体将对比算法中较好的结果标注出来.从表 1 可以看出,在 a280 等 8 个实例上,算法 DCSS-LKH 与 LKH 均取得了 20 个全局最优解,这表明 LK 算子的性能非常好.在 vm1084,pcb3038 实例上,LKH 取得了比 DCSS-LKH 更多的全局最优解.在 si535,pla7397 等 8 个实例上,DCSS-LKH 取得了比 LKH 更多的全局最优解.从总体上看,算法 DCSS-LKH 在最优解个数、解的平均值、最大值等方面均明显优于算法 LKH.

Table 1 Experiment results of 20 runs for LKH and DCSS-LKH on typical TSPLIB instances

表 1 算法 LKH 及 DCSS-LKH 在 TSPLIB 典型实例上各运行 20 次的实验结果

Instance		LKH					DCSS-LKH				
Name	Optima	n_{opt}	τ_{aver} (%)	τ_{min} (%)	τ_{max} (%)	t_{aver} (s)	n_{opt}	τ_{aver} (%)	τ_{min} (%)	τ_{max} (%)	t_{aver} (s)
a280	2 579	20	0.000	0.000	0.000	0.1	20	0.000	0.000	0.000	0.1
d493	35 002	20	0.000	0.000	0.000	0.7	20	0.000	0.000	0.000	0.8
att532	27 686	20	0.000	0.000	0.000	0.7	20	0.000	0.000	0.000	1.0
si535	48 450	4	0.007	0.000	0.017	5.8	8	0.006	0.000	0.017	6.6
u574	36 905	20	0.000	0.000	0.000	0.4	20	0.000	0.000	0.000	0.5
d657	48 912	20	0.000	0.000	0.000	0.5	20	0.000	0.000	0.000	0.6
pr1002	259 045	20	0.000	0.000	0.000	0.8	20	0.000	0.000	0.000	0.8
u1060	224 094	17	0.000	0.000	0.003	12.3	20	0.000	0.000	0.000	3.4
vm1084	239 297	15	0.008	0.000	0.046	5.7	7	0.023	0.000	0.046	10.6
rl1304	252 948	12	0.070	0.000	0.193	5.5	12	0.047	0.000	0.193	5.8
fl1577	22 249	8	0.035	0.000	0.067	206.8	20	0.000	0.000	0.000	72.2
d1655	62 128	18	0.000	0.000	0.002	7.5	20	0.000	0.000	0.000	3.2
vm1748	336 556	15	0.008	0.000	0.053	14.5	18	0.002	0.000	0.024	11.8
d2103	80 450	0	0.036	0.014	0.098	197.8	1	0.020	0.000	0.068	321.3
u2319	234 256	20	0.000	0.000	0.000	1.0	20	0.000	0.000	0.000	0.9
pr2392	378 032	20	0.000	0.000	0.000	10.0	20	0.000	0.000	0.000	10.5
pcb3038	137 694	17	0.005	0.000	0.034	88.2	15	0.006	0.000	0.034	106.5
fl3795	28 772	5	0.100	0.000	0.379	1104.0	10	0.050	0.000	0.191	2801.2
fnl4461	182 566	12	0.001	0.000	0.005	204.6	12	0.001	0.000	0.004	252.9
rl5915	565 530	0	0.024	0.012	0.038	606.6	0	0.023	0.006	0.038	841.9
rl5934	556 045	0	0.062	0.030	0.097	797.4	0	0.054	0.006	0.093	1187.2
pla7397	23 260 728	6	0.005	0.000	0.014	4793.8	9	0.001	0.000	0.005	8751.4

5 结论

本文对 TSP 问题的脂肪计算复杂性进行了理论分析,证明在 $P \neq NP$ 的假设下,不存在算法可以在多项式时间内获得完整脂肪.在此基础上,通过分析局部最优解与脂肪的关系,给出了新的元启发式算法——DCSS.利用 DCSS 改进后的 LKH 算法,在解的质量上有较为明显的提高.

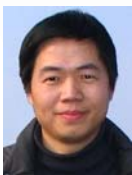
致谢 特别感谢本文评审专家为完善论文所付出的辛劳工作,他们给出的详细而中肯的意见显著地提高了本文的学术水平和可读性.

References:

- [1] Groes GA. A method for solving traveling salesman problems. *Operations Research*, 1958,6:791–812.
- [2] Bentley JL. Fast algorithm for geometric traveling salesman problems. *ORSA Journal on Computing*, 1992,4(4):387–411.
- [3] Lin S, Kernighan BW. An effective heuristic algorithm for traveling-salesman problem. *Operations Research*, 1973,21:498–516.
- [4] Helsgaun K. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 2000,126:106–130.
- [5] Applegate D, Cook WJ, Rohe A. Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 2003,15(1):82–92.
- [6] Merz P, Freisleben B. Fitness landscapes and memetic algorithm design. In: Corne D, Dorigo M, Glover F, eds. *Proc. of the New Ideas in Optimization*. London: McGraw-Hill, 1999. 245–260.
- [7] Dorigo M, Caro GD. The ant colony optimization meta-heuristic. In: Corne D, Dorigo M, Glover F, eds. *Proc. of the New Ideas in Optimization*. London: McGraw-Hill, 1999. 11–32.
- [8] Walshaw C. A multilevel lin-kernighan-helsgaun algorithm for the traveling salesman problem. Technical Report, 01/IM/80. London: School of Computing & Mathematical Sciences, University of Greenwich, 2001. 1–19.
- [9] Zou P, Zhou Z, Chen GL, Gu J. A multilevel reduction algorithm to TSP. *Journal of Software*, 2003,14(1):35–42 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/35.htm>
- [10] Qi YT, Liu F, Jiao LC. Immune algorithm with selfadaptive reduction for large-scale TSP. *Journal of Software*, 2008,19(6): 1265–1273 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1265.htm>
- [11] Zhang WX. Phase transitions and backbones of the asymmetric traveling salesman problems. *Journal of Artificial Intelligence Research*, 2004,21:471–497.
- [12] Schneider J. Searching for backbones—a high-performance parallel algorithm for solving combinatorial optimization problems. *Future Generation Computer Systems*, 2003,19(1):121–131.
- [13] Zhang WX, Look M. A novel local search algorithm for the traveling salesman problem that exploit backbones. In: *Proc. of the 19th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2005)*. San Francisco: Morgan Kaufman Publishers, 2005. 343–351.
- [14] Sharlee C, Zhang WX. Search for backbones and fat: A limit-crossing approach with application. In: *Proc. of the 18th National Conf. on Artificial Intelligence (AAAI 2002)*. AAAI Press, 2002. 707–712.
- [15] Boese KD. Cost versus distance in the traveling salesman problem. Technical Report, TR-950018. Los Angeles: Department of Computer Science, University of California at Los Angeles, 1995. 1–10.

附中文参考文献:

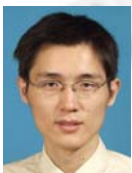
- [9] 邹鹏,周智,陈国良,顾钧.求解 TSP 问题的多级归约算法. *软件学报*,2003,14(1):35–42. <http://www.jos.org.cn/1000-9825/14/35.htm>
- [10] 戚玉涛,刘芳,焦李成.求解大规模 TSP 问题的自适应归约免疫算法. *软件学报*,2008,19(6):1265–1273. <http://www.jos.org.cn/1000-9825/19/1265.htm>



江贺(1980—),男,江西临川人,博士,副教授,CCF 会员,主要研究领域为智能计算,可信软件.



李强(1983—),男,硕士,主要研究领域为智能计算.



胡燕(1979—),男,博士,讲师,主要研究领域为智能计算,可信软件.



于红(1977—),女,讲师,CCF 高级会员,主要研究领域为 Web 信息处理.