

The Review of Backbone Algorithm about TSP*

Jinbiao Wang, Famin Ma

School of Computer Science and Technology, Civil Aviation University of China, Tianjin
Email: jinbiaow@vip.sina.com

Received: Nov. 6th, 2013; revised: Nov. 16th, 2013; accepted: Nov. 22nd, 2013

Copyright © 2013 Jinbiao Wang, Famin Ma. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: When the Hamiltonian circuit calculation algorithms of TSP stopped at local optimum trap, professor Boese discovered the hole phenomenon in 1995, and made backbone algorithm into TSP research filed. The backbone algorithm in TSP edge recognition is making progress. The paper predicts that the fusion of backbone algorithm and fat algorithm is inevitable trend.

Keywords: TSP; Backbone Algorithm; The Hole Phenomenon; TSP Edge Recognition; Fusion

关于 TSP 的骨架算法综述*

王锦彪, 马发民

中国民航大学计算机科学与技术学院, 天津
Email: jinbiaow@vip.sina.com

收稿日期: 2013 年 11 月 6 日; 修回日期: 2013 年 11 月 16 日; 录用日期: 2013 年 11 月 22 日

摘要: 当 TSP 的哈密顿回路计算算法研究止步于局部最优陷阱时, 1995 年 Boese 教授发现了大坑现象, 使骨架算法悄然进入了 TSP 研究领域。骨架算法在 TSP 边识别方面正在取得进展。预言了骨架算法与脂肪算法相融合的必然趋势。

关键词: TSP; 骨架算法; 大坑现象; TSP 边识别; 融合

1. 引言

1967 年, blum^[1]教授在深入研究图形学有关算法的基础上提出了骨架的概念。他假设图形边界点同时着火, 火源向图形内部各个方向等速燃烧直至熄灭, 所有熄灭点就构成了该图形的骨架, 这是骨架的最早定义。经过将近半个世纪的发展, 逐步形成了模拟烧草模型^[2-6]、基于距离变换^[7-12]以及 voronoi 图^[13,14]等用于图形检索、路径导航等图形学难题的有效算法。

骨架算法在图形学上的成功, 引起学术界的广泛关注。1995 年 Boese^[15]教授将骨架概念引入 TSP 研究

领域, 1998 年 Monasson 等^[16-19]讨论了可满足性问题 SAT 的骨架算法; 2005 年 Zou 等^[20]提出了求解 QAP 问题的近似骨架导向蚁群算法 ABFANT(approximate backbone-guided fant)。其中, Boese 教授的研究最为引人关注。他用随机 2opt、快速 2opt、快速 3opt、LK、LSMC 等五种局部最优算法对 532 点的 TSP 反复进行实验, 发现这些算法求得的局部最优解与公布的最优解竟有高达 80%以上的共边, Boese 称这一现象为大坑现象。

大坑现象引起了许多学者的兴趣和深入思考。

2003 年 Schneider^[21]和邹鹏^[22]等利用 TSP 问题局部最优解交集作为近似骨架, 得到了骨架导向并行系

*本文受国家自然科学基金项目(60472121)资助。

统算法和求解 TSP 问题的多级规约算法；2004 年 Zhang^[23] 分析了骨架规模对于非对称旅行商问题 ATSP(asymmetric traveling salesman problem) 求解难度的影响；2005 年 Zhang 等^[24] 给出了求解 TSP 问题的骨架导向 LK 算法。

本文对骨架算法在 TSP 领域的研究进展进行综述。第 2 节简要的介绍 TSP 问题和骨架算法的特点；第 3 节介绍骨架算法系列中具有代表性的几个算法，着重阐述相关算法的实现思想及主要创新点；最后，对全文进行了总结并给出了对骨架算法的展望。

2. TSP 问题和骨架算法的特点

2.1. TSP 的定义和两种基本思路

TSP 也称为货郎担问题或骑士旅行问题，最早可以追溯到 1759 年，是典型的 NP 完全问题。其目标是在 n 个节点的图中找到边权之和最小的哈密顿回路。二维欧几里德平面上边权直接表述为欧几里德距离或欧几里德距离的线性函数的对称 TSP 问题也称为狭义 TSP。限于篇幅本文主要讨论狭义 TSP。TSP 算法设计的基本思路有两种：

思路 1 n 个节点的完全图中存在 $\frac{1}{2}(n-1)!$ 个哈密顿回路。计算每一个哈密顿回路的长度，通过比较找出其中最短的 TSP。

这个思路是基于哈密顿回路计算的。因此，基于该思路的算法也可称为哈密顿回路计算算法。其中比较优秀的是通过增加一些策略，达到减少比较次数的目的，例如 3OPT 等。

思路 2 n 个节点的完全图中存在 $\frac{1}{2}n(n-1)$ 条边。从 $\frac{1}{2}n(n-1)$ 条边中识别出属于 TSP 的 n 条边。

这个思路是基于 TSP 边识别的，因此，基于该思路的算法也可称为 TSP 边识别算法。

当前关于 TSP 的算法研究正处于哈密顿回路计算算法向 TSP 边识别算法转变的关键时期。骨架算法的产生和发展正是这个转变的一个标志。

2.2. 骨架算法的特点

1) Boese 发现的大坑现象是 TSP 骨架算法的坚实基础。根据 Boese 大坑理论，3OPT 的局部最优解中含有 80% 以上的 TSP 边，因此若干个 3OPT 的局部最

优解的交集中包含 TSP 边的可能性就很大，而含非 TSP 边的可能性就极小。在一个 n 节点的完全图中若定义 TSP 为骨架。那么若干个 3OPT 的局部最优解的交集就必然使部分骨架显现出来，这意味着识别出部分 TSP 边。显然，骨架算法的交集迭代不会大于 n 次。

2) 骨架算法发现了部分骨架后，则把这部分已经确认的骨架空间压缩掉，使下一轮的骨架空间变小了，因此，骨架算法的收敛具有显著的方向性。

3) 具有鲜明的并行计算特征。

4) 以 3OPT 等局部最优解为初始解，是对经典算法的继承和发展。

3. 几个典型的骨架算法

3.1. 基于并行系统的骨架算法^[21]

Johannes Schneider^[21] 教授研究几个特殊的 TSP 算例后发现这些算例的局部解存在大量的交边，依据此他提出了一种寻找骨架降低 TSP 规模的高效并行算法。

首先由从处理器在短时间内产生合理的初始迭代解集合，所有从处理器产生的解路径发送到主处理器上，主处理器依据此产生重叠向量集，这个集合中包含所有路径中共同包含的边。

重叠向量集中包含的边叫做骨架，它们是全局最优解的一部分，因为它们包含在所有的局部最优解中。这种解释仅仅是基于统计学来说的。在接下来的迭代中用一对点来表示这些长度从 1 到点的规模最大值不等的骨架，例如对于 10 个点的对称性 TSP 问题产生如下四个解路径：

0 9 1 2 3 4 5 6 7 8

0 3 2 1 4 5 6 7 8 9

0 9 7 6 5 4 8 1 2 3

0 4 5 6 7 8 3 2 1 9

根据这些初始解可以产生如下重叠向量集：

0 9

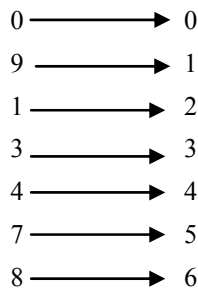
1 2 3

4 5 6 7

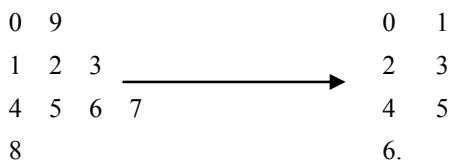
8

在接下来的迭代中对这些骨架进行重新编码，目的是尽可能的减少城市和距离矩阵的规模，从而降低从处理器的计算量，所有骨架的边界点得到新的编

号, 中间的点将会被忽略掉,



因此重叠向量得到如下编码:



经过这样的处理, 原来 10×10 的距离矩阵 d , 就变成现在的 7×7 的距离矩阵 D , 二者的关系如下:

$$D(0,1) = d(0,9)$$

$$D(2,3) = d(1,2) + d(2,3)$$

$$D(4,5) = d(4,5) + d(5,6) + d(6,7)$$

$$D(0,6) = d(0,8)$$

$$D(1,6) = d(9,8)$$

下列新路径和距离矩阵 D 一起返回给从处理器:

0-----1 2-----3 4-----5 6-----6

这些路径就是初始迭代解中产生的最优路径, “-----”表示该路径不允许被打破。这样就产生了较小规模的距离矩阵, 直到产生一条和原始城市规模相等长度的骨架运行终止。然而一个路径也许会包含更多的点; 假如用来产生重叠向量集的解完全不同, 那么重叠向量集中所有向量长度为 1, 这些向量由两个点表示, 这样会产生是原始规模两倍的点集。这种对重叠向量集中长度为 1 的向量用两个点表示的编码方式, 对于反复的去查看骨架是否被打破了是明智的。这种编码对于完全不同的初始解来说会增加额外的运算时间, 但是对于有很多相同部分的初始解来说它又会在很大程度上减少运算时间。最为重要的是这种编码方式对于确保骨架不被打破是必要的。

在接下来的迭代过程中, 从处理器产生新的路径时要采用略微不同的编程。它们只允许打破第二个、第四个、第六个……最后一个城市之间的连接方式,

这样才能确保骨架不被打破。每个从处理器独自用不同的初始路径做着各自的优化, 最后将它们优化结果发给主处理器。主处理器用先前的重叠向量集对这些解进行解码, 对这些解的解码程序是一致的。例如从处理器发来的解路径如下:

0---1 2---3 6---6 4---5
1---0 4---5 2---3 6---6
0---1 6---6 3---2 5---4
1---0 4---5 6---6 3---2

主处理器解码后的路径为:

0 9 1 2 3 8 4 5 6 7
9 0 4 5 6 7 1 2 3 8
0 9 8 3 2 1 7 6 5 4
9 0 4 5 6 7 8 3 2 1

这个结果优于之前的那个, 对之前已经存在的最优路径有了更好的表示。之前存在的骨架集合就会被舍弃掉, 程序根据解码的路径重新计算重叠向量集。虽然重叠向量集是根据新的路径得到的, 但是它也保证了旧的骨架被继承下来, 确保了骨架不被打破这一原则, 新产生的重叠向量集如下:

0 9
1 2 3 8
4 5 6 7

随着迭代次数的增加, 得到的骨架越来越少但是越来越长, 系统优化的规模越来越少, 继而运行时间也越来越少, 现在点的规模降低到了 6, 距离矩阵也变成了 6×6 , 在最后一次迭代中从处理器会产生相同的路径, 该路径便是最优解。

这个算法的依据是: 骨架是全局最优解的一部分, 当然不可能事先知道骨架一定是全局最优解的一部分。算法中假设一定数量的解会产生骨架, 因此需要初始化一个合适的解的数目。如果这个初始解的数目过少会造成产生的骨架过长, 也许这样的骨架不是最优解的一部分; 如果初始解的数目过多有可能不会产生一个骨架这也导致算法失效, 因此该算法的关键是给出一个合适的初始化解的数目。

3.2. 基于频繁度的骨架算法^[24]

由于骨架一被发现就被确定下来不再发生变化, 这容易使算法陷入局部最优解中, 针对这个问题

Zhang Weixiong^[24]提出了基于频繁度的骨架算法。

首先根据某一条边在局部样本解中出现的次数定义伪骨架频繁度。假如定义 S 是样本解集合, 给出已知边 e 定义集合 S_e , 其中 S_e 是 S 的子集, S_e 中包含的解是 S 解路径中有边 e 的解, 直观的将 $|S_e|/|S|$ 叫做伪骨架频繁度。

选择的局部解集合会影响伪骨架频繁度的质量。理想状态下, 期望局部解都是高质量近似解的无偏样本。能否达到这一理想状态, 其中一个影响因素就是所给出的初始化的开始路径: 开始路径差别越大, 一般的最后形成的最优解差别就越大。虽然由贪心算法初始化开始路径得到的局部最优解要优于随机初始化开始路径的解, 但是得到伪骨架的质量却是后者优于前者。

可以将伪骨架信息融合到 LS 算法中实现有偏搜索。LS 中边的移动取决于移入边与移出边的长度, 假如移入边的总体长度少于移出的, 那么这项移动将被执行。算法中将采用两种有偏移动策略, 一种是仅仅依靠伪骨架频繁度; 另外一种综合考虑伪骨架频繁度和城市之间的距离。定义集合 B 为骨架的边集合, 集合 T 为局部最优解集合, 可以根据它得到伪骨架频繁度。假如 X 和 Y 是被移出移入边的候选集, 分别的用 K-opt 对边进行移动。假如 Y 集合中的边有较大的概率比集合 X 中的边获得更多的骨架, 这时用集合 Y 中的边取代集合 X 中的边。假定边之间都是各自独立的, 假如

$$\sum_{x_i \in X} P(x_i \in B|T) < \sum_{y_i \in Y} P(y_i \in B|T) \quad (1)$$

其中 $P(x_i \in B|T)$ 就是边 x_i 的伪骨架频繁度, 该频繁度是由解集合 T 得到的, 我们就将 y_i 移入 x_i 移出。这种方法已经运用到最大可满足性问题^[17]并且取得了很好的效果。

但是, 单纯的用伪骨架频繁度诱导局部搜索算法解决 TSP 问题并不是很有效。造成这个结果的一个可能因素是二者的搜索空间不同, TSP 问题搜索空间存在的状态数量是 $2^E = O(2^{n^2})$, 其中 n 个城市存在边的数量是 $|E| = n(n-1)/2$ 。这些状态嵌入到可约束结构中, 在这个结构中一条边的移动会使得很多边不能移动。相比之下最大可满足问题的搜索空间存在的状态数是 2^n , 其中 n 是布尔变量。因此这种估计得到的伪骨架频繁度用在最大可满足问题上比 TSP 问题上更

可靠。这种仅仅依靠伪骨架频繁度诱导算法的缺陷表明: TSP 问题中实际城市之间的距离是不能被忽略的。这就是伪骨架频繁度诱导局部搜索算法在 TSP 问题上和在最大可满足问题上主要的不同点了。

LK 算法中边之间的替换是由替换边的长度决定的。同样的评价原则可以运用到骨架导向局部搜索算法中, 简称 BGLS。但是价值的计算方式不同, 依据不是两个城市之间的距离 w , 而是 $w(1 - P)$, 其中 P 是该边的伪骨架频繁度。因此一个边的长度是随着伪骨架频繁度成线性减少的。极端情况该边不在伪骨架中则该边距离就是实际两个城市间的距离, 如果该边在所有局部解中则该边的距离变成 0。

LK 搜索算法使用这种策略取代原始城市间距离后, 得到的局部解有了很大的提高。这种有偏搜索之所以有效的原因是: 伪骨架频繁度最终改变了 TSP 实例城市之间的距离。这种新得到的局部最优通常不是上一代中的局部最优, 因此对原始图形的持续搜索会提高解的质量, 即便有偏搜索已经找到了局部最优解。

此外伪骨架的信息同样可以用到初始化开始路径中。众所周知使用贪心算法初始化局部搜索算法的开始路径可以提高该算法的速度和效率。这种贪心路径的构造过程是, 首先随机选择一个城市点, 然后寻找距离该城市最近的城市, 依次找下去, 这样一条一条的边被加入到开始路径中, 直到所有的城市都被加入进来。我们可以修改此过程, 用伪骨架去重新定义最短边, 使用伪骨架频繁度计算各边的权值取代各边的实际距离。

将上面的思想运用到 LK 算法中, 得到骨架导向 LK 算法, 简称 BGLK。和 LK 算法相似 BGLK 也需要运行很多代, 每一代都由新的初始路径计算得到局部最优解。与 LK 不同的是 BGLK 有两个阶段。第一个阶段是学习阶段, 运用原始的 LK 算法采用随机化的开始路径运行一定的迭代次数, 使用其求得的局部最优解计算得到伪骨架频繁度。第二个阶段是骨架导向阶段, 使用有偏 K-opt 算法。并且, 在第二个阶段使用有偏的开始路径同样也可以提高局部最优解的精度和局部搜索的速度。在实验中发现整个运行次数的 30%用于学习就已经有效了。

该算法提出的伪骨架频繁度概念, 一方面它可以

避免 LK 算法陷入局部最优解中；另一方面可以使用它初始化开始路径加速局部搜索和提高局部最优解精度。但是，此方法并没有在本质上降低每代城市的规模。

3.3. 求解 TSP 的多级规约算法^[22]

随着对骨架算法研究的深入，人们从不同的角度来看骨架的含义。人们最初对骨架的定义就是局部最优解的交集，把它看成全局最优解的一部分。既然它

已经是全局最优解的一部分，那么接下来计算中就不需要再去找它，而把它保留下来作为解的一部分即可。因此邹鹏、周智、陈国良提出了多级归约算法^[22]。

该算法的基本原理是通过对 TSP 问题的局部最优解与全局最优解之间关系分析，发现对局部最优解的简单的相交操作能以很高的概率得到全局最优解的部分解，利用这些部分解可以大大缩小原问题的搜索空间，同时也不会降低搜索算法的性能。

其算法可以由图 1 形象的描述。

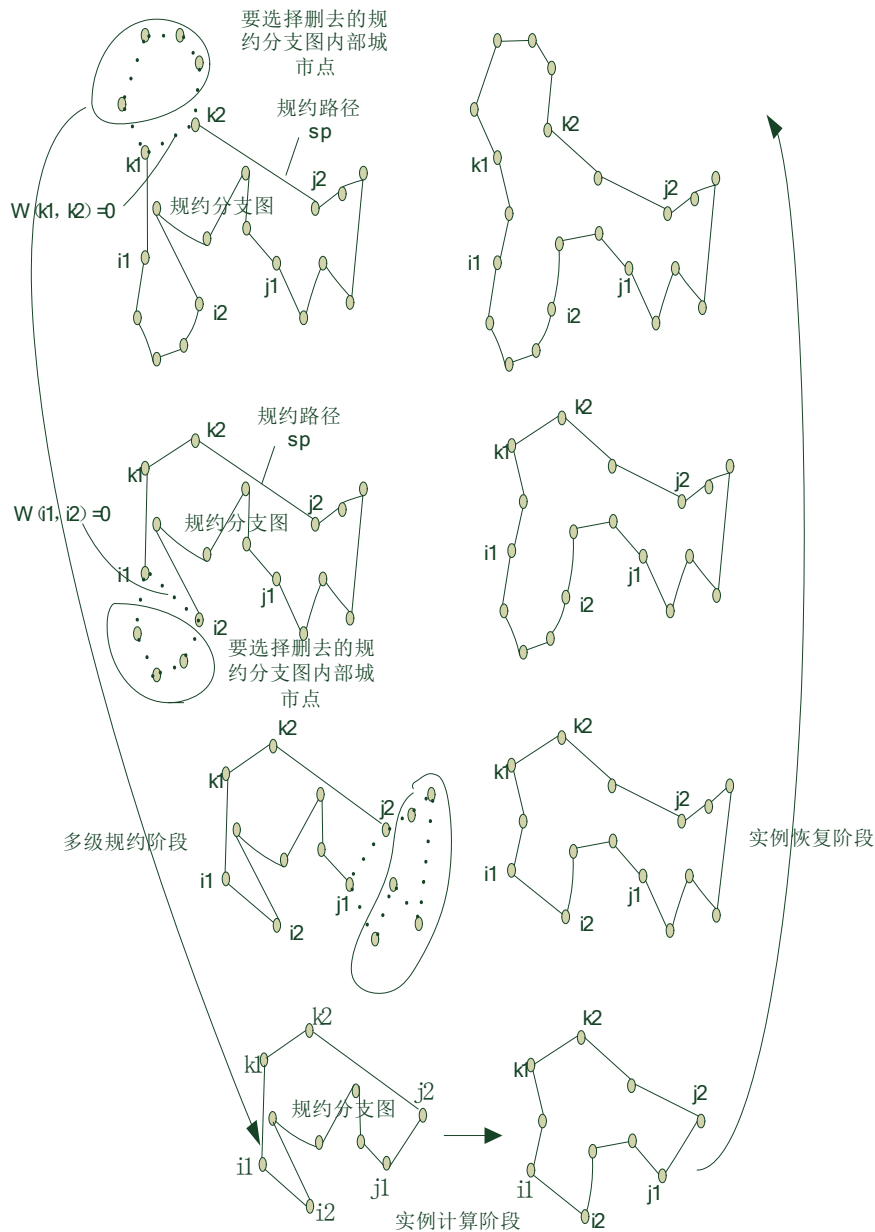


Figure 1. Main idea of multilevel reduction algorithm
图 1. 多级规约算法主体思想描述图

算法描述:

输入: TSPLIB 中的典型实例;

输出: TSPLIB 中的典型实例的解。

Begin

初始化(对当前实例求解一个用于被归约的局部最优解, 记为 A);

1) 实例的多级归约

while (当前实例规模不是足够小) do

对当前实例规模生成 R_s 个局部最优解记为 $B_i(I = 1, 2, \dots, R_s)$;

由 B_i 和 A 求交得到部分解 P;

$A = A - P$;

用部分解 P 对当前实例进行归约, 减小其规模从而生成新的实例;

Endwhile

2) 用已知算法直接求解当前实例(解记为 I)

3) 恢复当前实例的解到原实例的解

while (当前实例规模 \neq 原实例规模) do

$I = I + P$;

endwhile

End.

该算法中一个难点是参数 R_s 的取值, 从本质上来说就是选择几个个体相交产生骨架, 在介绍 3.1 算法中也提到了这个问题, 但是目前缺少理论的支撑。该算法中如果 R_s 取值过大会加大运算的时间, 但是结果的精度会有较大的提高, 在时间和精度上该算法折中了一下将 R_s 的值设定为 2, 即三个个体共同含有的边定为骨架。

4. 总结与展望

综上所述不难看出, 当 TSP 领域的哈密顿回路计算算法陷入局部最优陷阱不能自拔时, 骨架算法使 TSP 边识别算法的实现变为可能, 从而使 TSP 研究出现了新的生机, 开辟了新的思路。

如果我们把 TSP 边看做是骨架, 那么, 包裹着骨架的非 TSP 边就可以看做是脂肪了。脂肪算法是将若干局部最优解通过并集操作剥离部分脂肪, 通过多次迭代直至脂肪被剥净。近年来已有学者研究脂肪算法^[26,27], 并且取得了一定的成果。目前可以期待的是实现骨架算法与脂肪算法的融合, 也就是每一次迭代通

过交集操作获取部分骨架, 同时也通过并集操作剥离部分脂肪^[25]。不难看出, 这种融合将大大提高计算效率, 不仅是必要的, 也是可能的。这个方向的研究希望引起有关学者的关注。

当然, 由于骨架算法出现较晚, 尚未成熟, 还有许多改进空间。例如, 作为初始解的局部最优解数目现在还是经验参数, 对 TSP 边的识别率还有待提高。

参考文献 (References)

- [1] Blum, H. (1967) A transformation for extracting new descriptors of shape. In: Walthen-Dunn, W., ed., *Models for the Perception of Speech and Visual Form*, Cambridge: MIT Press, 362-380.
- [2] Ma, C.M. and Sonka, M. (1996) A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding*, **64**, 420-433.
- [3] Pudney, C. (1998) Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, **72**, 404-413.
- [4] 车武军, 杨勋年, 汪国昭 (2003) 动态骨架算法. *软件学报*, **14**, 818-823.
- [5] Leymarie, F. and Levin, M.D. (1992) Simulating the grass fire transform using an active. Contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 56-75.
- [6] Golland, P. and Grimson, W.E.L. (2000) Fixed topology skeletons. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 10-17.
- [7] Niblack, C.W., Gibbon, P.B. and Capson, D.W. (1992) Generating skeletons and center lines from the distance transform. *CVGIP: Graphical Models and Image Processing*, **54**, 420-437.
- [8] Ge, Y. and Fitzpatrick, J.M. (1996) On the generation of skeletons from discrete euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 1055-1066.
- [9] Bitter, I., Kaufman, A.E. and Sato, M. (2001) Penalized-distance volumetric skeleton algorithm. *IEEE Transactions on Visualization and Computer Graphics*, **7**, 195-206.
- [10] Zhou, Y. and Toga, A.W. (1999) Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, **5**, 196-209.
- [11] Choi, W.P., Lam, K.M. and Siu, W.C. (2003) Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition*, **36**, 721-729.
- [12] 张琳波, 肖柏华, 王枫等 (2013) 图像内容表示模型综述. *计算机科学*, **7**, 1-8.
- [13] Ogniewicz, R.L. and Kubler, O. (1995) Hierarchic Voronoi skeleton. *Pattern Recognition*, **28**, 343-359.
- [14] Reddy, J.M. and Turkiyyah, G.M. (1995) Computation of 3D skeletons using a generalized Delaunay triangulation technique. *Computer Aided Design*, **27**, 677-694.
- [15] Boese, K.D. (1995) Cost versus distance in the traveling salesman problem. Technical Report CSD-950018.
- [16] Monasson, R., Zecchina, R., Kirkpatrick, S., et al. (1998) Determining computational complexity for characteristic phase transition. *Nature*, **400**, 133-137.
- [17] Zhang, W.X. (2004) Configuration landscape analysis and backbone guided local search: Part I: Satisfiability and maximum satisfiability. *Artificial Intelligence*, **158**, 1-26.
- [18] Dubois, O. and Seymour, P.A. (2001) Backbone-search heuristic for efficient solving of hard 3-SAT formula. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, San Francisco: Morgan Kaufmann Publishers, 248-253.
- [19] Valnir, F.J. (2006) Backbone guided dynamic local search for

- propositional satisfiability. *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics (AI&Math-06)*. New York: Springer, 100-108.
- [20] Zou, P., Zhou, Z., Chen, G.L., et al. (2005) Approximate-backbone guided fast ant algorithms to QAP. *Journal of Statistical Software*, **16**, 1691-1698.
- [21] Schneider, J. (2003) Searching for backbones: A high-performance parallel algorithm for solving combinatorial optimization problems. *Future Generation Computer Systems*, **19**, 121-131.
- [22] 邹鹏, 周智, 陈国良等 (2003) 求解 TSP 问题的多级归约算法. *软件学报*, **1**, 35-42.
- [23] Zhang, W.X. (2004) Phase transition and backbones of the asymmetric traveling salesman problem. *Journal of Artificial Intelligence Research*, **21**, 471-497.
- [24] Zhang, W.X. and Looks, M. (2005) A novel local search algorithm for the traveling salesman problem that exploit backbones. *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 343-350
- [25] Wang, J.B. and Wang, K.C. (2010) Union-intersection any system. *5th International 2010 Conference on Bio-Inspired Computing: Theories and Applications*, 364-371.
- [26] Sharlee, C. and Zhang, W.X. (2002) Search for backbones and fat: A limit-crossing approach with application. *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, Menlo Park: AAAI Press, 707-712.
- [27] 江贺, 胡燕, 李强, 于红 (2009) TSP 问题的脂肪计算复杂性与启发式算法设计. *软件学报*, **9**, 2344-2351.