1000-9825/2003/14(01)0035 ©2003 Journal of Software 软 件 学 报 Vol.14, No.1

求解 TSP 问题的多级归约算法*

邹 鹏[1,2+], 周 智[1,2], 陈国良[1,2], 顾 钧[1,2,3]

[1](中国科学技术大学 计算机科学技术系,安徽 合肥 230026)
[2](国家高性能计算中心(合肥),安徽 合肥 230026)
[3](香港科技大学 计算机科学与技术系,香港)

A Multilevel Reduction Algorithm to TSP

ZOU Peng[1,2+], ZHOU Zhi[1,2], CHEN Guo-Liang[1,2], GU Jun[1,2,3]

[1](Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China)
[2](Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China)
[3](Department of Computer Science and Technology, Hong Kong University of Science and Technology, Hong Kong, China)
+ Corresponding author: Phn: 86-551-3603747, Fax: 86-551-3601013, E-mail: zouchar@mail.ustc.edu.cn
http://csrv.hpac.dhs.org/~zouchar
Received 2001-07-17; Accepted 2001-12-27

Zou P, Zhou Z, Chen GL, Gu J. A multilevel reduction algorithm to TSP. *Journal of Software*, 2003,14(1): 35~42.

Abstract: The TSP (traveling salesman problem) is one of the typical NP-hard problems in combinatorial optimization problem. The fast and effective approximate algorithms are needed to solve the large-scale problem in reasonable computing time. The known approximate algorithm can not give a good enough tour for the larger instance in reasonable time. So an algorithm called multilevel reduction algorithm is proposed, which is based on the analysis of the relation of local optimal tour and global optimal tour of the TSP. The partial tour of the global optimal tour is obtained by a very high probability through simply intersecting the local optimal tours. Using these partial tours it could contract the searching space of the original problem, at the same time it did not cut the searching capability down, this is the so-called reduction theory. And then the scale of the instance could be reduced small enough by multi-reduction. Finally it could solve the small-scaled instance using the known best algorithm and get a good enough tour by putting the partial tours together. The experimental results on TSPLIB (traveling salesman problem library) show that the algorithm could almost get optimal tour every time for instance in reasonable time and thus outperformed the known best ones in the quality of solutions and the running time.
Key words: TSP (traveling salesman problem); NP-hard; partial tour; reduction; multilevel reduction algorithm

摘 要: TSP(traveling salesman problem)问题是最经典的 NP-hard 组合优化问题之一.长期以来,人们一直在寻求快速、高效的近似算法,以便在合理的计算时间内解决大规模问题.由于对较大规模的问题,目前的近似算法

---

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030403 (国家重点基础研究发展规划(973))

第一作者简介: 邹鹏(1979－),男,内蒙古包头人,硕士生,主要研究领域为复杂性理论,NP 问题的启发算法.

尚不能在较短的时间内给出高质量的解,因此提出了多重归约算法.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的分析,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的性能.这就是所谓的归约原理.再通过多次归约使问题的规模降到足够小,然后对这个较小规模的实例直接用已有的算法求解,最后通过相反的次序拼接部分解,最终得到一个合法的解.在 TSPLIB(traveling salesman problem library)中,典型实例上的实验结果表明,此算法在求解质量和求解速度上与目前已知的算法相比有较大的改进.

关键词: TSP(traveling salesman problem);NP-hard;部分解;归约;多级归约算法

中图法分类号: TP301 **文献标识码**: A

TSP(traveling salesman problem)问题是为人们所广泛研究的典型 NP-Hard 组合优化问题[1],广泛应用于 VLSI 芯片设计、电路版布局、机器人控制、车辆选路等领域.对该问题的最新综述参见文献[2~4],其形式化描述为:给定加权图 $G=(V,E,w)$,V 为顶点集,$|V|=n$,E 为边集,$w:E\rightarrow R^{+}$ 为边权函数,G 中一个 TSP 环路就是一条不重复地访问 V 中所有顶点的 Hamilton 环路,记为 $T=\langle v_1,v_2,\dots,v_n\rangle$,其中 $\forall 1\le i\ne j\le n, v_i\ne v_j$ 且对 $1\le i\le n$, $\langle v_i,v_{(i\bmod n)+1}\rangle\in E$,记 $TSP(G)$ 为 G 中所有 TSP 环路的集合.定义 $w(T)=w(\langle v_1,v_n\rangle)+\sum_{1\le i\le n-1}w(\langle v_i,v_{i+1}\rangle)$,要求权最小的 TSP 环路 T^*,使得 $w(T^*)=\min_{T\in TSP(G)}w(T)$.

TSP 问题是典型的 NP-Hard 问题,即在 $P\ne NP$ 的假设下,找不到一个算法能保证在多项式时间内得到最优解.目前,对于该类问题的研究有两个方向:完全算法能保证得到最优解,但是运行时间是呈指数复杂度的,因而难以适应大规模的实例;近似算法则只要求找到近似解,而在多项式时间内结束.在 TSP 问题上的近似算法分为两类[2]:环路构造算法和环路改进算法.前者从某个非法解开始,通过某种增广策略逐步改变该解,直到得到一个合法解为止.这类算法包括最近邻算法、贪心算法、Clarke-Wright 算法[5]、Christofides 算法[6]等.环路改进算法则在给定初始的合法解之后,使用某种策略来改进初始解.这些策略包括局部搜索、模拟退火[7]、遗传算法[8]等,其中最为简单和有效的方法为局部搜索,如 2-OPT[9]、3-OPT[10]、LK[11] 、循环 LK[12]等.通常这两类算法结合起来使用,用环路构造算法来构造初始解,而用环路改进算法改进这个初始解.Arora[13]对欧氏空间的 TSP 问题提出了多项式时间的近似策略,这也是近年来在近似算法的研究中所取得的一项重要进展.但是,当问题规模很大时,现有的算法都不甚理想.

本文在对 TSP 问题进行分析的基础上,通过对全局最优解和局部最优解中的边之间关系的分析,发现通过简单的对局部最优解进行相交操作,能以较大概率得到全局最优解中的边.称这些边为全局最优解中的部分解,通过这些部分解可以极大地降低原 TSP 问题的规模,而并不降低搜索的性能.我们称此方法为归约方法.但在实验中发现,只经过单步归约后,较大的TSP实例仍然很大,对其直接用已知的较好的启发式算法求解,仍然不能在较短的时间内得到高质量的解.因此,我们想到了用多次归约的方法,并提出了一种新的多级归约算法,称为 multilevel reduction 算法.该算法分为 3 个阶段:实例多级归约阶段、实例求解阶段和实例恢复阶段.我们用该算法测试了 TSPLIB(traveling salesman problem library)[14]中的典型实例,与已知最好的求解 TSP 问题的局部搜索算法作了比较,发现这种多级归约算法无论在求解质量还是在求解时间上均较大地改进了已知算法.

本文第 1 节是多级归约算法的基本思想及其理论分析.第 2 节是算法描述及其实验细节.第 3 节给出实验结果.第 4 节得出结论.

1 多级归约算法的基本思想及理论分析

1.1 用FDC方法分析TSP问题

FDC(fitness distance correlation)分析方法是分析组合优化问题的一种重要方法.FDC 分析的目标在于发现局部最优解的适应度与其到最优解的距离之间的相关性.其基本原理如下:对于一个给定的组合优化问题(不妨设为一个最小化问题,如 TSP 问题)的实例 I,定义一个三元组 $L=(S,f,d)$ 来描述与该实例相关的解及其特性.其中,S 为对应于该实例的解集合;适应度函数 $f:S\rightarrow R$ 为解空间到实数空间上的映射(常取为目标函数);距离

函数 $d:S\times S\to R$ 满足以下条件: $\forall s,u,t\in S$, 有 $d(s,t)\ge 0$, $d(s,t)=0\Leftrightarrow s=t$, $d(s,t)\le d(s,u)+d(u,t)$, $d_{\min}\le d(s,t)\le d_{\max}$ 成立.对于局部搜索算法,一般定义两个解之间的距离为从一个解通过局部搜索到达另一个解所需要的最少步数.由于特定的局部搜索方法的距离难以得到,一般用解的编码之间的 Hamming 距离定义:设问题的解可以表示为向量 $s_j=\overrightarrow{x_{j1}\ldots x_{jn}}$,则定义 $d_H(s_1,s_2)=\sum_{1\le i\le n}[x_{1i}\ne x_{2i}]$.在 TSP 问题中,Hamming 距离为一条环路上与另一条环路的不同边的数量.

定义该实例的适应度地貌为图 $G_L=(V,E)$,其中 $V=S$, $E=\{\langle s,s'\rangle\in S\times S\,|\,d(s,s')=d_{\min}\}$.对于解 s ,若 $\forall\langle s,s'\rangle\in G_L, f(s)\le f(s')$,则 s 为该距离定义(或局部搜索方法)下的局部最优解,简称为局部最优解,记其集合为 $LOCAL_d(I)$,简记为 $LOCAL(I)$.称适应度函数值最小的解为 I 上的全局最优解,记其集合为 $OPT(I)$,显然有 $\forall d, OPT(I)\subseteq LOCAL(I)$.

下面以 TSP 问题为例.我们取 TSPLIB 中的典型实例 att532 和 pr2392(数字代表实例大小,字母代表实例名称)为实验样本,用常用的 D.S.Johnson 的 Concode 代码[15]中的 LK 算法取 3 000 次随机实验得到的结果如图 1 所示.

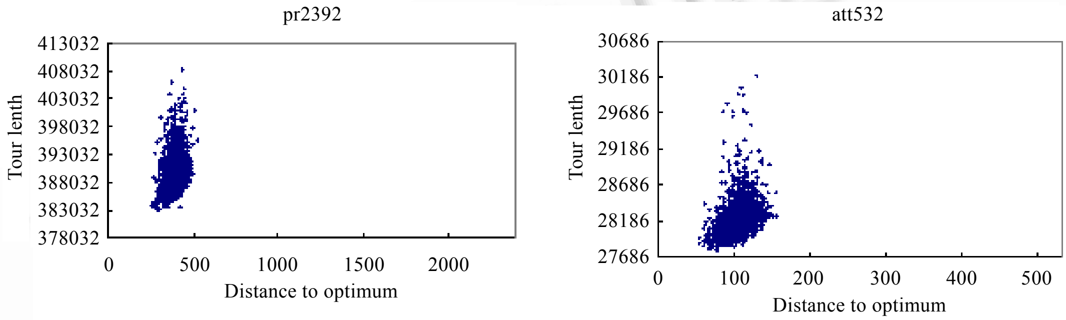


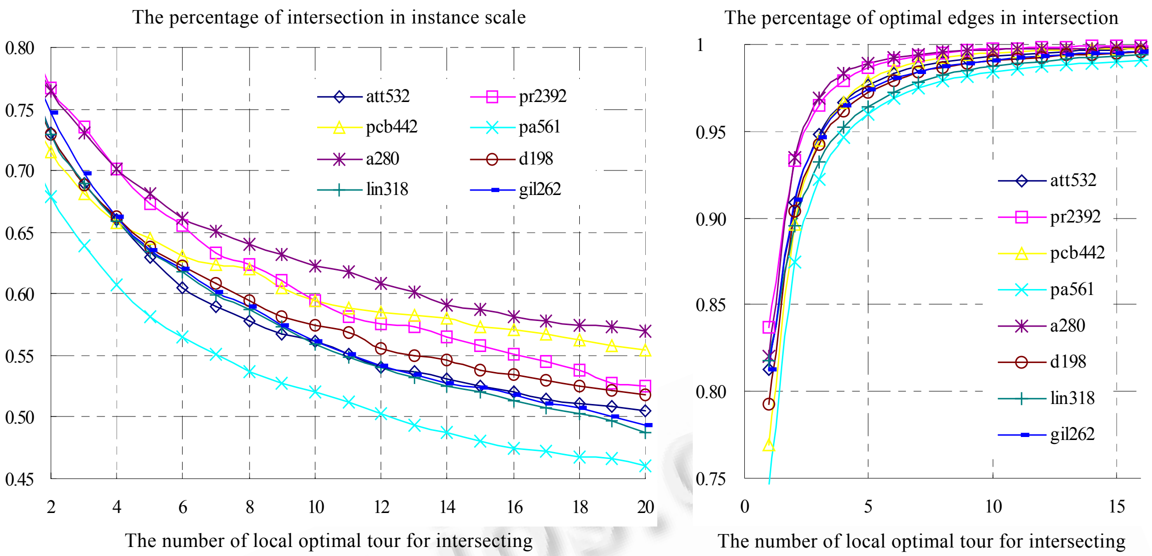
Fig.1 FDC analysis figure of typical TSP instances att532 and pr2392 in TSPLIB
图 1 TSPLIB 中的典型实例 att532 和 pr2392 的 FDC 分析图

于是有如下简单的结论:局部最优解到全局最优解的距离基本上为定值,而基本上与其环路长度无关.对实例 pr2392 来说,局部最优解到全局最优解的距离在 500 左右,问题规模为 2 392.对实例 att532 来说,局部最优解到全局最优解的距离在 100 左右,而问题规模为 532.经过归一化之后,我们发现不同实例的共同特性是,局部最优解到全局最优解的距离均在问题规模的 1/5 左右,即局部最优解中有 80%左右的边是全局最优解中的边,并提出了相应的“大坑原理”.Merz 等人在此原理的基础上设计了 DPX 交叉算子[16],其性能很好.该方法在遗传局部搜索算法 GLS 算法中应用得较为成功.

1.2 多级归约算法的基本思想

在 TSP 问题中,解是特殊的边集合.注意到集合的两种基本操作:交和并都只需线性时间来完成.根据上面的分析(一般 LK 算法得到的局部最优解到全局最优解的距离在问题规模的 1/5 左右),我们面临的问题是如何用较小的代价来保存易于在局部最优解中得到的全局最优解中的边.这自然可以用简单的并和交的操作来完成.我们对 TSPLIB 中的几个典型实例的局部最优解进行求交集的操作,所得到的实验结果如图 2 所示.显然,多个边集合的交可以增加全局最优解的边的比例.但是,减小这些边的数量,可以看到,随着局部最优解数量的增加,交集的大小缓慢地减小,同时,全局最优解的边所占的比例迅速增长,并很快趋近于 1.

我们从与以上思路相反的角度出发,考虑能不能得到一个部分解,使得其中的所有边以极大的概率在全局最优解中出现.一个最简便的方法就是考虑局部最优解的交集,从而可以得到一个所有边都在全局最优解中出现的部分解.于是,我们没有必要在整个状态空间中进行搜索,而只在这个部分解被固定的子空间中搜索.这样也就降低了原始空间的规模,即降低了 TSP 实例的大小.一次性的归约对于一般的较小实例可能是很有效的,而对于大规模的实例,我们取 TSPLIB 中的几个较大的实例进行归约分析(数字代表实例大小,字母代表实例名称),如图 3 所示(图中纵坐标为归约后的实例规模与原始实例规模的比值).



(a) The size of intersections
(a) 交集的尺寸

(b) The percentage of optimal edges in intersection
(b) 交集的最优解中的边所占的比例

Fig.2 The property of intersections of local optimal tours
图 2 多个局部最优解的交的特性

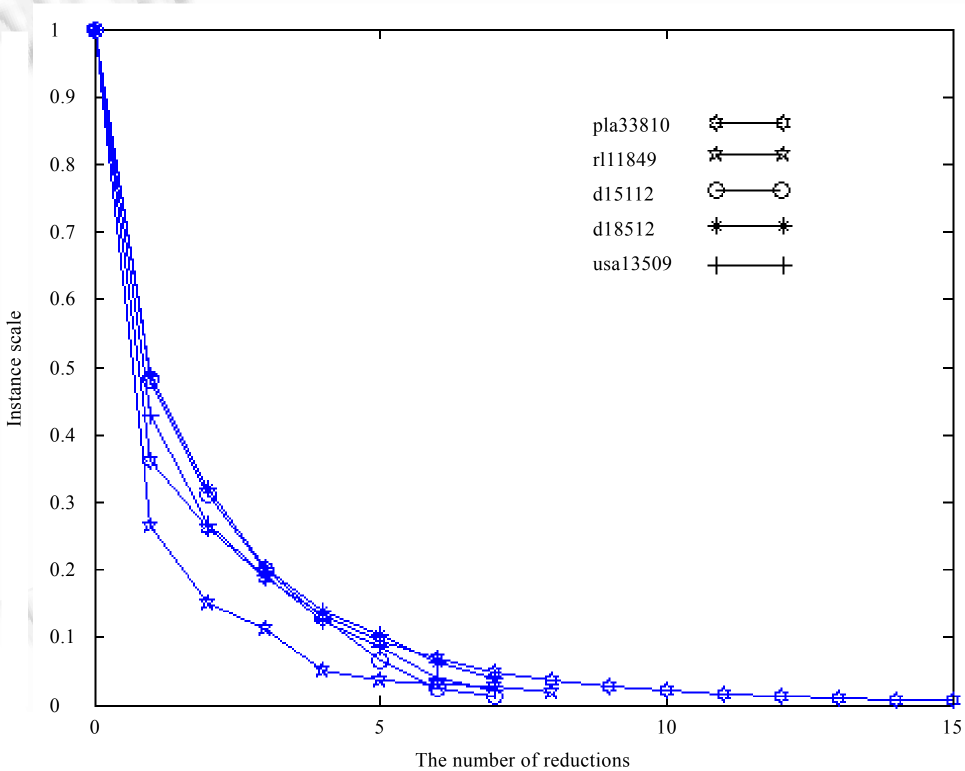


Fig.3 TSP instance scale analysis of multilevel reduction
图 3 多级归约的实例规模分析

我们可以看到,一次的归约不能把大规模实例约简到足够小的规模,对 pla33810 来说,只能将实例规模降到原来规模的 36%左右,对 rl11849 来说,也只能降到原来规模的 30%左右,这时的规模仍然很大.由于已知的局部搜索算法对小规模的实例很有效,但对大规模的实例的运行时间及求解质量不十分理想,在这种情况下对经过

一次归约后的实例直接用已知的算法求解,不一定能得到高质量的解.为此,我们设计了不断地约简问题规模的多级归约算法.其主体思想是,通过稳定的部分解进行多级归约,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部分解,最终得到一个合法的解.图 4 描绘了多级归约算法的主要思想.

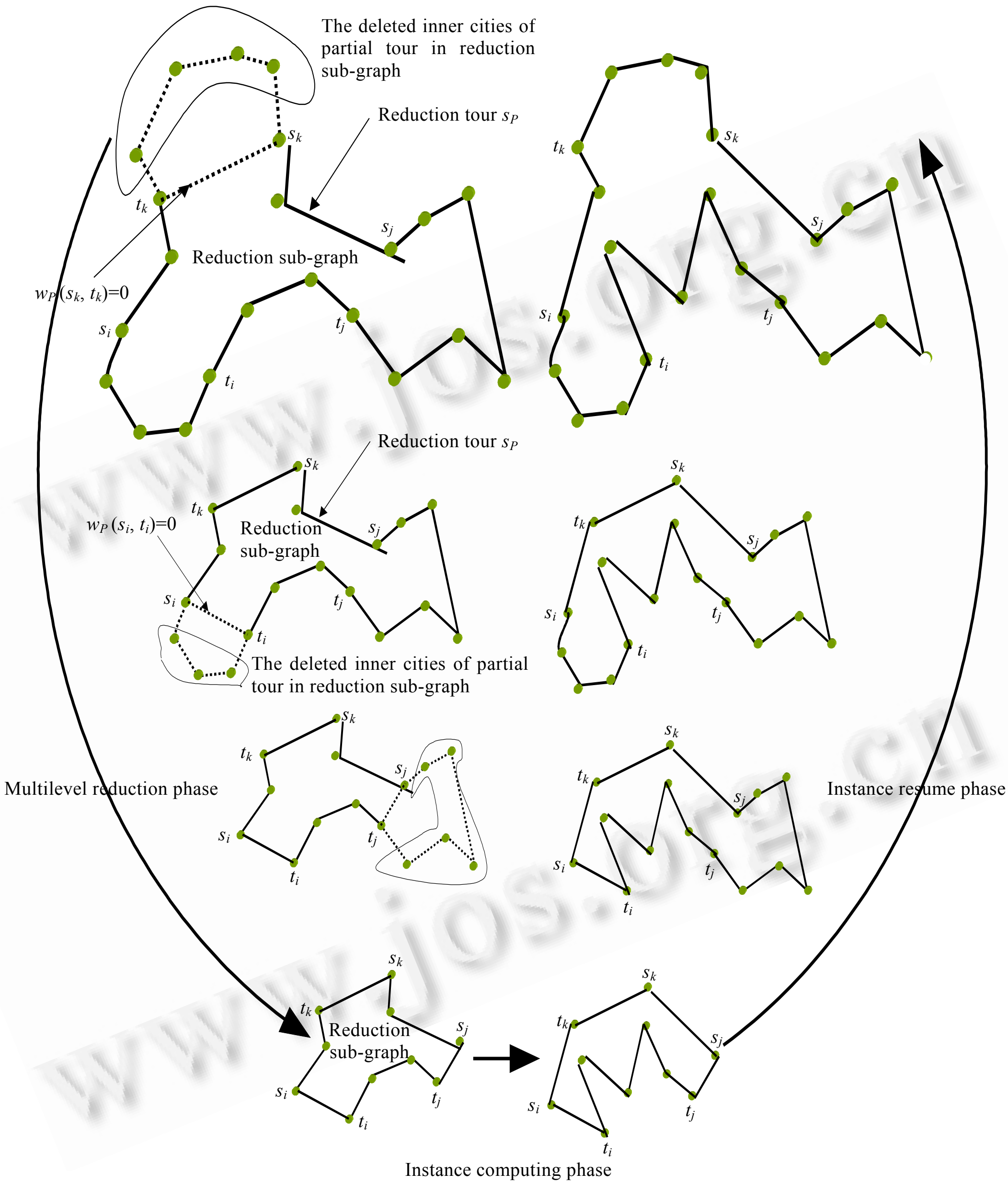


Fig.4 Main idea of multilevel reduction algorithm
图 4 多级归约算法主体思想描述图

2 多级归约算法描述及其合法性证明

2.1 多级归约算法描述

算法. Multilevel Reduction Algorithm.

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 $B_i(i=1,2,\ldots,R_s)$;

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≠原实例规模) do

$I=I+P$;

endwhile

End.

2.2 算法合法性的证明

在 TSP 问题中,给定图 $G=(V,E,W)$,其合法的解是一条不重复地访问 V 中所有顶点的 Hamilton 环路,记为 $T=\langle v_1,v_2,\ldots,v_n\rangle$,其中 $\forall 1\le i\ne j\le n, v_i\ne v_j$ 且对 $1\le i\le n,\langle v_i,v_{(i \bmod n)+1}\rangle\in E$,其中 $|T|=|V|=n$.而一个合法的部分解是 k 条没有公共顶点的简单路径 $p_i=\langle s_i=v_{i,1},v_{i,2},\ldots,v_{i,li}=t_i\rangle$ 的集合 $P=\{p_i\,|\,1\le i\le k\}$.显然,任何一个合法解都可以被划分为多个部分解之和.在TSP问题中,多个合法解的交集构成一个合法的部分解.为了描述上的方便,我们将单个的顶点描述为起点和终点一致的简单路径,令 $G'=(V',E',W')$,使得 G' 为 G 去掉部分解 P 的子图,在 G' 上定义简单的映射为 $O(s_i)=i$, $O(t_i)=i$.此时,重新定义 G 上的距离函数为

$$w'(\langle v,u\rangle)=\begin{cases} w_I(u,v), & \text{if } O(v)\ne O(u) \\ 0, & \text{otherwise} \end{cases},$$

其中 w_I 为原来定义两点在 G 上的距离(对所有的 $e\in E$,有 $w_I>0$).显然,如果在 G' 中, $O(u)=O(v)$,则 $w'(\langle v,u\rangle)=0$,则边 $e=\langle u,v\rangle$ 将出现在 G' 的局部最优解 s' 中.对 $s'=\langle e_{\pi(1)},\ldots,e_{\pi(k)}e^*_{\partial(1)},\ldots,e^*_{\partial(m)}\rangle$,其中 $e_i=\langle s_i,t_i\rangle$ 或 $e_i=\langle t_i,s_i\rangle$, $e^*_{\partial(j)}\,|\,(j=1,\ldots,m)$ 为长度大于 0 的局部最优解中的边.π为$[1,k]$上的一个置换.按 s'中 e 的次序重排 p_i ,显然, $T=\langle p_{\pi(1)},\ldots,p_{\pi(k)}\cup e^*_{\partial(1)},\ldots,e^*_{\partial(m)}\rangle$ (此处为了方便描述,用边的集合来表示一个 TSP 问题的解)为原问题 G 的一个合法解.

随着局部最优解的数目 R_s 的增多,其中的边在全局最优解中出现的概率越大,而部分解的规模越小,归约后的搜索速度越慢.所以,对于一定的时间限制而言,对用于归约的局部最优解的数量的选择是需要调整的.这是一个参数调制问题,下一节将给出结果.

3 实验结果与比较分析

本文实现了用多级归约算法求解 TSP 问题,实验环境是 PIII550Mhz 微处理机,512MB 内存,操作系统为 Turbo_Linux4.0.我们对 TSPLIB 中的典型实例进行了测试(各运行到 20 次时的结果),对于基于并发多级归约的

算法, R_s 是个关键参数. R_s 越大,每次循环得到的解的质量越好,但循环一次的时间也越长.当 $R_s \ge 3$ 时,算法的运行时间是我们无法接受的.为了提高算法的速度,在实现中我们取 $R_s = 2$,即只保持两个用于归约的解.

作为比较,本文在相同的实验环境下运行了循环 LK 算法(当前最有效的基于局部搜索的算法之一),并在相同的实例上测试了我们的算法,将实验数据按照小规模的实例、中等规模的实例和大规模的实例分别与循环 LK 所得到的数据作了比较,其结果见表 1~表 3.

Table 1 Experiment results of multilevel reduction and iterated LK on small TSP instances over 20 runs

表1 多级归约算法和循环LK算法在小规模实例上运行20次的实验结果

TSP instance		Quality of the tour (mean)		(Distance to optimal)/optimal		Running time (s) (mean)	
Name	Global optimal tour	Multilevel reduction (MR)	Iterated LK (ILK)	MR	ILK	MR	ILK
D493	35 002	35 002	35 004	0.000 0	0.000 0	7.43	171.10
U574	36 905	36 905	36 905	0.000 0	0.000 0	7.67	9.30
Pcb442	50 778	50 778	50 778	0.000 0	0.000 0	2.65	8.47
Rat575	6 773	6 774	6 777	0.000 0	0.000 0	10.48	91.13
Ali535	202 310	202 339	202 339	0.000 1	0.000 1	21.14	30.53

Table 2 Experiment results of multilevel reduction and iterated LK on middle TSP instances over 20 runs

表2 多级归约算法和循环LK算法在中等规模实例上运行20次的实验结果

TSP instance		Quality of the tour (mean)		(Distance to optimal)/optimal		Running time(s) (mean)	
Name	Global optimal tour	Multilevel reduction (MR)	Iterated LK (ILK)	MR	ILK	MR	ILK
Pr1002	259 045	259 045	259 045	0.000 0	0.000 0	56.68	60.01
Fl1400	20 127	20 127	20 127	0.000 0	0.000 0	38.37	41.05
Fl1577	22 249	22 249	22 249	0.000 0	0.000 0	84.59	433.70
Vm1084	239 297	239 311	239 320	0.000 1	0.000 1	48.47	112.15
Vm1748	336 556	336 556	336 556	0.000 0	0.000 0	470.72	991.79
Rl1304	252 948	252 948	252 948	0.000 0	0.000 0	22.14	52.21
Rl1323	270 199	270 199	270 199	0.000 0	0.000 0	137.01	313.90
Pcb1173	56 892	56 893	56 892	0.000 0	0.000 0	115.79	120.44

Table 3 Experiment results of multilevel reduction and iterated LK on large TSP instances over 20 runs

表3 多级归约算法和循环LK算法在大规模实例上运行20次的实验结果

TSP instance		Quality of the tour (mean)		(Distance to optimal)/optimal		Running time (s) (mean)	
Name	Global optimal tour	Multilevel reduction (MR)	Iterated LK (ILK)	MR	ILK	MR	ILK
Rl11849	923 368	925 421.7	926 076.7	0.002 2	0.002 9	906.05	950.78
Rl5915	565 530	565 896.2	566 408.8	0.000 6	0.001 6	40.54	41.15
Fl3795	28 772	28 779.4	28 808.6	0.000 2	0.001 3	85.356	85.36
D2103	80 450	80 498.4	80 664.0	0.000 6	0.002 7	238.84	284.35
U2319	234 256	234 420.6	234 519.0	0.000 7	0.001 1	1216.33	1217.15
U2152	64 253	64 281.0	64 305.6	0.000 4	0.000 8	173.98	173.98

对于中小规模的实例(见表 1 和表 2),多级归约算法无论是在运行时间还是在求解质量上,均优于循环 LK 算法.对于大规模的实例,要想求得全局最优解,所需的时间很长且对内存的要求也很高.由实验环境所限,我们对多级归约算法和循环 LK 算法在运行时间相同且循环 LK 算法求解精度较好的情况下,比较它们的解.可以看出,多级归约算法的解其质量明显优于循环 LK 算法.

4 结 论

本文在对 TSP 问题进行分析的基础上,通过对全局最优解和局部最优解中的边之间关系的分析发现,通过简单的对局部最优解进行相交操作,能以较大的概率得到全局最优解中的边.通过归约这些边,可以极大地降低原 TSP 问题的规模,并不降低搜索的性能.同时,在实验中发现,经过单步归约后,较大的 TSP 实例仍然很大,直接用已知的较好的启发式算法对其求解,仍然不能在较短的时间内得到高质量的解.因此,我们给出了一种新的多级归约算法,并用该算法测试了 TSPLIB 中的典型实例.结果显示,我们的算法无论在求解时间上还是在求解质量上,与当前求解 TSP 问题的最好的局部搜索算法相比都有所改进.

References:

[1] Garey MR, Johnson DS. Computers and Intractability: a Guide to the Theory of NP-Completeness. San Francisco: W.H. Freeman, 1979.

[2] Johnson DS, McGeoch LA. The traveling salesman problem: a case study in local optimization. In: Aarts EH, Lenstra JK, eds. Local Search in Combinatorial Optimization. New York: John Wiley and Sons, 1996.

[3] Jünger M, Reinelt G, Rinaldi G. The traveling salesman problem. In: Ball M, Magnanti T, Monma CL, Nemhauser G, eds. Handbook on Operations Research and Management Science: Networks North-Holland. 1995. 225~330.

[4] Burkard RE, Deineko VG, Dal RV, *et al*. Well-Solvable special cases of the traveling salesman problem: a survey. SIAM Review, 1998,40(3):496~546.

[5] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 1964,12: 568~581.

[6] Christofides N. Worst-Case analysis of a new heuristic for the traveling salesman problem. Technical Report, No.388, Pittsburgh, PA: Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[7] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science, 1983,220(4598):671~680.

[8] Holland JH. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. 2nd ed., Cambridge, MA: MIT Press, 1992.

[9] Croes GA. A method for solving traveling salesman problems. Operations Research, 1958,6:791~812.

[10] Lin S. Computer solutions to the traveling salesman problem. Bell System Technical Journal, 1965,44(10):2245~2269.

[11] Lin S, Kernighan BW. An effective heuristic algorithm for the traveling salesman problem. Operations Research, 1973,21:498~516.

[12] Johnson DS. Local optimization and the traveling salesman problem. In: Proceedings of the 17th Colloquium on Automata, Language, and Programming. Lecture Notes in Computer Science 443, Berlin: Springer-Verlag, 1990. 446~461.

[13] Arora S. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In: Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science. 1996. 2~11. http://citeseer.nj.nec.com/arora96polynomial.html.

[14] Reinelt G. TSPLIB—a traveling salesman problem library. ORSA Journal on Computing, 1991,3(4):376~384.

[15] Johnson DS. Concode. 1997. ftp://ftp.caam.rice.edu/pub/people/bico/970827/cc970827.tgz.

[16] Boese K. Cost versus distance in the traveling salesman problem. Technical Report, TR-950018, CS Department, UCLA, 1995.