1000-9825/2005/16(10)1691 ©2005 Journal of Software 软 件 学 报 Vol.16, No.10

求解 QAP 问题的近似骨架导向快速蚁群算法*

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

[1](中国科学技术大学 计算机科学技术系 国家高性能计算中心(合肥),安徽 合肥 230027)
[2](香港科学技术大学 计算机科学系,香港)

Approximate-Backbone Guided Fast Ant Algorithms to QAP

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

[1](National High Performance Computing Center at Hefei, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)
[2](Department of Computer Science, 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

Received 2003-07-28; Accepted 2004-01-06

Zou P, Zhou Z, Chen GL, Jiang H, Gu J. Approximate-Backbone guided fast ant algorithms to QAP. *Journal of Software*, 2005,16(10):1691–1698. DOI: 10.1360/jos161691

Abstract: Quadratic Assignment Problem (QAP) is one of the classical combinatorial optimization problems and is known for its diverse applications. This paper presents a new fast ant heuristic for the QAP, the approximate-backbone guided fast ant colony algorithm (ABFANT). The main idea is to fix the approximate-backbone which is the intersection of several local optimal permutations to the QAP. After fixing it, the authors can smooth the search space of the QAP instance without losing the search capability, and then solve the instance using the known fast ant colony algorithm (FANT) which is one of the best heuristics to the QAP in the much smoother search space. Comparisons of ABFANT and FANT within a given iteration number are performed on the publicly available QAP instances from QAPLIB. The result demonstrates that ABFANT significantly outperforms FANT. Furthermore, this idea is general and applicable to other heuristics of the QAP.

Key words: QAP; approximate-backbone; ABFANT; QAPLIB

---

* 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))

ZOU Peng was born in 1979. He received his BS degree in Computer Software from University of Science and Technology of China (USTC) in 2000, and his Ph.D. degree in computer science also from USTC in 2005. His current research areas are algorithms for NP-hard problems and parallel computing. **ZHOU Zhi** was born in 1976. He received his BS degree in Computer Software from University of Science and Technology of China (USTC) in 1995. Now he is a Ph.D. candidate in computer science of USTC. His current research areas are combinatorial problem and parallel computing. **CHEN Guo-Liang** was born in 1938. Now He is a fellow of Chinese Academy of Sciences, a Ph.D. advisor in Department of Computer Science at USTC, and the Director of the National High Performance Computing Center at Hefei. His research areas are parallel computing, computer architecture and combinatorial optimization. **JIANG He** was born in 1980. He received his Ph.D. degree in USTC in 2004. His current research areas are parallel computing and combinatorial problem. **GU Jun** was born in 1956. He received his BS degree in Electronic Engineering from USTC in 1982, and his Ph.D. degree in Computer Science from University of Utah. Now he is a professor and a Ph.D. advisor in Computer Science at USTC and Hong Kong University of Science and Technology. His main research areas are algorithms for NP-hard problems.

摘 要: QAP(quadratic assignment problem)问题是经典的组合优化问题之一,广泛应用于许多领域中.针对QAP 问题,提出了一种新的蚁群算法——近似骨架导向的快速蚁群算法(ABFANT).该算法的基本原理是通过对局部最优解的简单相交操作得到 QAP 问题实例的近似骨架(approximate-backbone),利用这些近似骨架可以极大地缩小 QAP 问题的搜索空间,而同时不降低搜索的性能,最后对这个缩小后的搜索空间,直接用当前求解 QAP问题最好的启发式算法之一——快速蚁群算法(FANT)求解得到问题的解.在 QAPLIB 中的典型实例上的实验结果表明,近似骨架导向的快速蚁群算法明显优于快速蚁群算法.此外,指出基于近似骨架的算法思想可以很容易地被移植到其他求解 QAP 问题的启发式算法中.

关键词: QAP;近似骨架;ABFANT;QAPLIB
中图法分类号: TP301 文献标识码: A

1 Introduction

The quadratic assignment problem (QAP) was first proposed by Koopmans and Beckman[1] in the context of the plant location problem. Given n facilities represented by the set $F=\{f_1,\ldots,f_n\}$, and n locations represented by the set $L=\{l_1,\ldots,l_n\}$, one must determine to which location each facility must be assigned. Let $B^{n\times n}=(b_{i,j})$ be a matrix where $b_{i,j}\in R^+$ represents the flow between facilities f_i and f_j. Let $A^{n\times n}=(a_{i,j})$ be a matrix where entry $a_{i,j}\in R^+$ represents the distance between locations l_i and l_j. Let $p:\{1,\ldots,n\}\rightarrow\{1,\ldots,n\}$ be an assignment and define the cost of this assignment to be $c(p)=\sum_{i=1}^{n}\sum_{j=1}^{n}a_{i,j}b_{p(i),p(j)}$. In the QAP, we want to find a permutation vector $p\in\Pi_n$ that minimizes the assignment cost, i.e. min $c(p)$, subject to $p\in\Pi_n$, where Π_n is the set of all permutations of $\{1,\ldots,n\}$. The QAP is well known to be strongly NP-hard[2].

Many practical problems from such areas as location science[3], architectural design[4], and hardware/chip design[5] can be formulated as instances of the QAP; other well known combinatorial optimization problems such as the traveling salesman problem and the graph partitioning problem are special cases of the QAP[6]. Since the QAP is a NP-hard problem, exact solution approaches are currently only effective for the instances of size $n<30$. Therefore several heuristics that attempt to find the near-optimum solutions to the large QAP instances in a reasonable time have been proposed. Such heuristic approaches include the ant colonies[7,8], evolution strategies[9], genetic algorithms[10], simulated annealing[11], neural networks[12], tabu search[3,13], threshold accepting[14], tree search heuristics[15], randomized greedy search (GRASP)[16], and hybrid approaches[17−19].

The *backbone* of a problem instance is referred to as the set of variables that are common to all global optimal solutions for the given instance. The concept first appeared in Ref.[20] when Kirkpatrick and Toulouse studied the Traveling Salesman problem, and attracted much attention recently[21−23]. An exact backbone, however, is generally hard to be obtained for many optimization problems such as QAP, MAXSAT, and TSP. Instead, an approximate-backbone, which is the intersection of different local optima of an instance, can be used to investigate the characteristic of the instance[24]. The original idea of local optima intersection can be found in the related study in TSP[25]. In this paper, we apply the similar idea to the QAP, where an approximate-backbone is the common variables of several local optimal permutations of instance (more details in Section 3).

In the past, ant system algorithms have been applied to a variety of combinatorial optimization problems[8,26]. In this paper we suggest a new fast ant algorithm for the QAP that incorporates the idea based on the approximate-backbone. We call this algorithm the approximate-backbone guided fast ant algorithm. The main idea is to fix the approximate-backbone of a QAP instance so that we can smooth the searching space of the instance without losing the searching capability. And then we can solve the QAP instance using the known fast ant colony algorithm[26] in the much smoother search space. A comparison of the ABFANT and FANT within the given iteration times is performed on the publicly available QAP instances from QAPLIB. The result indicates that our ABFANT obtains better solutions than FANT in about ninety percent of the performance.

The main contribution of this paper is an innovative method that exploits the solution structure of the QAP to improve the performance of a heuristic such as FANT (Section 2). We establish a connection between global optimal and local optimal by using the approximate –backbone, instead of the exact backbone, of the QAP. The idea developed here is general and applicable to other heuristics of the QAP. Due to the limited available source codes of heuristics, we only apply the new method to FANT (Section 2). We believe, however, the method can also achieve improvement on other heuristics.

The paper is organized as follows. Section 2 presents the fast ant colony algorithm to the QAP. Section 3 introduces the backbone and our approximate-backbone to the QAP. Section 4 describes the ABFANT algorithm designed to solve the QAP. In Section 5, result for many QAP instances from QAPLIB produced by the ABFANT, as well as comparisons with FANT, is presented. Section 6 concludes the paper and outlines several future research directions.

2 Fast Ant Colony Algorithm to QAP

In this section, we introduce one of the best heuristics to the QAP, FANT[26,27] (Algorithm 3), which combines the local search with the ant colony algorithm for the QAP. FANT can be specified as four components: the memory structure, the constructing procedure, the improving procedure (local search procedure), and the way that the memory is updated.

The memory is principally constituted by a matrix T of size $n\times n$ whose entry τ_{ij} measures the preference of setting $p_i=j$, and from an ant system point of view, this matrix represents the pheromone trail left by the ants. The construction of a provisory solution is presented in Algorithm 1. The improvement procedure is a local search process described in Algorithm 2, and the procedure will be repeated twice (The evaluation of $\Delta(p,i,j)$ can be performed in $O(n)$ using Eq.(1)). The memory is updated as Eq.(2), where r and R represent the reinforcement of the matrix entries corresponding to p, the solution produced at the current iteration, and p^*, the best solution produced so far. More details of FANT can be found in Refs.[26,27].

$$\begin{aligned}\Delta(p,i,j)=&(a_{ii}-a_{jj})(b_{p_jp_j}-b_{p_ip_i})+(a_{ij}-a_{ji})(b_{p_jp_i}-b_{p_ip_j})+\\&\sum_{k\neq i,j}(a_{ki}-a_{kj})(b_{p_kp_j}-b_{p_kp_i})+(a_{ik}-a_{jk})(b_{p_jp_k}-b_{p_ip_k})\end{aligned}\tag{1}$$

$$\tau_{\mathrm{ip_i}}=\tau_{\mathrm{ip_i}}+\mathrm{r};\ \tau_{\mathrm{ip^*_i}}=\tau_{\mathrm{ip^*_i}}+\mathrm{R}\tag{2}$$

3 Backbone and the Approximate-Backbone

The *backbone* of a problem instance is referred to as the set of variables that are common to all global optimal solutions for the given instance. These variables are critically constrained as the elimination of any one of them will negate any possibility of finding any optimal solution. Currently, there are a significant amount of research activities in finding backbone variables[28], correlating the size of the backbone with problem hardness and phase transitions[21−23]. An exact backbone, however, is generally hard to be obtained for many optimization problems such as QAP, MAXSAT, and TSP. Instead, an approximate-backbone, which is the intersection of different local optima of an instance, can be used to investigate the characteristic of the instance

Algorithm 1. Constructing a provisory solution.
Input: The QAP instance from QAPLIB
Output: The provisory solution of the instance
Begin

1) $I=\varnothing$, $J=\varnothing$

2) While $|I|<n$ repeat:

2a) Choose i, randomly, uniformly, $1 \le i \le n, i \notin I$

2b) Choose j, randomly, uniformly, $1 \le j \le n, j \notin J$, with probability $\tau_{ij} \Big/ \sum_{1\le k\le n, k\notin J} \tau_{ik}$ and set $p_i = j$

2c) $I=I\cup\{i\}$, $J=J\cup\{j\}$

3) Return p

End

Algorithm 2. Local search procedure in FANT.

Input: The provisory solution of the instance

Output: The local optimal solution of the instance

Begin

1) $I=\varnothing$.

2) While $|I| < n$ repeat:

2a) Choose i, randomly, uniformly, $1 \le i \le n, i \notin I$.

2b) $J = \{i\}$

2c) While $|J| < n$ repeat:

2c1) Choose j, randomly, uniformly, $1 \le j \le n, j \notin J$.

2c2) If $\Delta(p,i,j) < 0$, exchange p_i and p_j in p.

2c3) $J=J\cup\{j\}$.

2d) $I=I\cup\{i\}$

3) Return p

End

Algorithm 3. FANT.

Input: The QAP instance I from QAPLIB

Output: The solution of the QAP instance I

Begin

1. Initialize the memory structure T and the optimal solution p^*
2. Do the following for some given number M of iterations
   - 2.1 Construct a provisory solution p of instance I (Algorithm 1)
   - 2.2 Improve the solution p by the local search procedure to get p' (Algorithm 2)
   - 2.3 If p' is better than p^* then
     - 2.3.1 Set $p^*= p'$
     - 2.3.2 Initialize the memory structure T
   - 2.4 Update the memory structure T
3. Return p^*

End

In this paper, we define and investigate the properties of approximate-backbone of the QAP. Basically, an approximate-backbone of the QAP is the common permutation of k different local optima of a QAP instance, which can be obtained by any QAP heuristics. More specifically, given an instance I of the QAP and its k different local minima, $p_0,p_1,\ldots,p_{k-1}$, we have the following definition.

Definition 1. The approximate-backbone of I, $AB(p_0,p_1,\ldots,p_{k-1})=p_0\cap p_1\cap\ldots\cap p_{k-1}$, $(p_i\in\Pi_n, i\in[0,k-1])$ is the intersection of all these k local minima.

We test several QAP instances with different sizes for finding their approximate-backbones, and find out that segments in an approximate-backbone generally appear in the global optimum with high probability. For an instance, Table 1 gives the size of its approximate-backbone and the corresponding probability of appearing in global optimum averaged over 20 independent runs. All the local minima of the QAP instances are obtained by FANT within the given iteration times. We only report data of k=2 in Table 1, for the reason that the size of the approximate-backbone is generally very small (for almost all of the instances the size is zero) when $k\geq 3$.

Table 1 Statistical results of the approximate-backbone (AB)

QAP instance name	The size of QAP instance	The average size of AB	The average probability of AB appearing in the global minima (%)	Iteration times
Chr25a	25	5	57.0	100
Tai30b	30	14	77.8	100
Tai40b	40	21	90.0	100
Lipa40a	40	8	69.1	100
Tai50b	50	12	56.3	250
Chr22b	22	5	81.9	250
Tai80b	80	8	84.3	10000

Colligating all the above results, we conclude that, when the number of the local optima is 2, the size of an approximate-backbone is generally moderate to the size of instance, and the approximate-backbone appears in global optimum with high probability.

4 Approximate-Backbone Guided Fast Ant Colony Algorithm

Motivated by the conclusions obtained from the experimental results in the above section, we design a new ant colony heuristic: approximate-backbone guided fast ant colony algorithm (ABFANT for abbreviation). In the scheme of ABFANT, an approximate-backbone is obtained by FANT. Then we smooth the search space by fixing segments in the approximate-backbone. After that, FANT is used again to identify the remaining segments in the permutation. A formal description of ABFANT is shown below.

Algorithm 4. The Approximate-Backbone Guided FANT.

Input: The QAP instance I from QAPLIB.

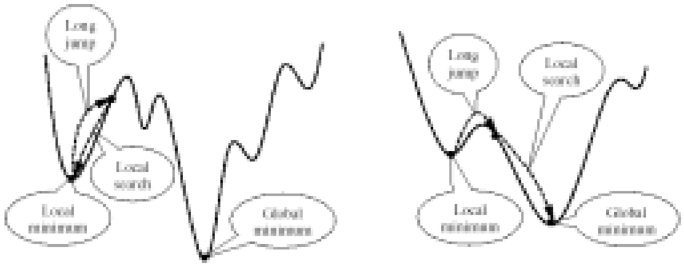
Output: The solution of the QAP instance I

Begin

1. Find an initial solution p_0 by using FANT for inputting the QAP instance I
2. Do the following:
   2.1. Find $k-1$ different solutions $p_0,p_1,\ldots,p_{k-1}$ of I by FANT
   2.2. Obtain the approximate-backbone $AB(p_0,p_1,\ldots,p_{k-1})$ of these k solutions
   2.3. Fix $AB(p_0,p_1,\ldots,p_{k-1})$ to get a new search space $S^*(I)$
   2.6. Run FANT to get the solution p_k in the search space $S^*(I)$
   2.7. If p_k is better than p_0, set $p_0=p_k$
3. Return p_0

End

We fix $AB(p_0,p_1,\ldots,p_{k-1})$ from the QAP instance I to reduce the search space of I. As shown in Fig.1, when the search space of instance I, $S(I)$, is very large, it is also very rugged so that the chance of escaping from a local minimum by a long jump of the local search in FANT is small. After fixing the approximate-backbone of I, we get a new search space $S^*(I)$, which is much smoother than $S(I)$, and a long jump can escape from a local minimum with large probability. So the local search process in FANT will be more efficient in the search space $S^*(I)$ than in $S(I)$.



The search space $S(I)$ The new search space $S^*(I)$

Fig.1 Search space of the QAP

5 Experimental Results

Our ABFANT was implemented in C++ on a Pentium IV PC (1.4GHz) running Redhat 7.2. To evaluate its performance, we selected several QAP instances from QAPLIB[29], ranging from n=20 locations up to n=150. The QAPLIB contains different types of QAP instances, which may be distinguished by their flow dominance and distance dominance[30]. The flow dominance fd for the flow matrix B is defined as:

$$fd(B)=100\times\sigma/\mu,\quad \mu=1/n^2\sum_{i=1}^{n}\sum_{j=1}^{n}b_{ij},\quad \sigma=\sqrt{1/n^2-1\sum_{i=1}^{n}\sum_{j=1}^{n}(b_{ij}-\mu)^2}.$$

The flow dominance is high when few entries in the flow matrix have a high influence on the total cost, and if almost all entries are equally sized, the flow dominance is low. The distance dominance dd can be defined in a similar manner for the distance matrix A. QAP instances with randomly generated flows (distances) using a uniform distribution typically have a low flow (distance) dominance, whereas real-life instances and (non-uniformly) randomly generated instances close to real-life instances have considerably higher dominance values for at least one of the matrices. We ran ABFANT and FANT to solve the selected instances, including problems with high and low flow and/or distance dominance value, within the given iteration times.

In Figs.2 and 3, the detailed comparisons of one performance of ABFANT and FANT with QAP Instance wil50 and sko64 are given. The x-axis denotes the number of ABFANT and FANT's iterations; and the y-axis represents the quality of the solution to the QAP instance. The iteration number in ABFANT is the sum of the iterations used to find the approximate-backbone and the iterations to get the solution in the new search space $S^*(I)$.

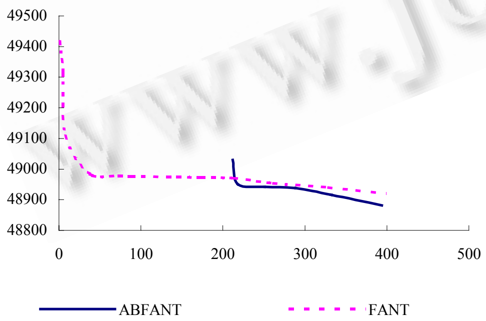


Fig.2 Comparisons of ABFANT and FANT on Instance wil50 within 400 iterations

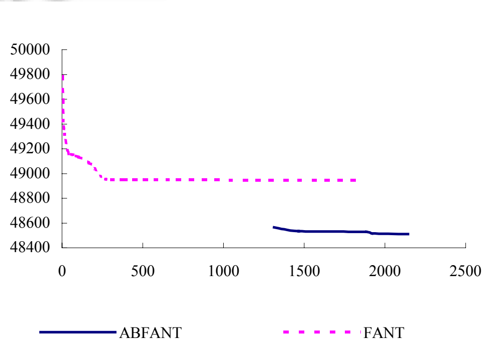


Fig.3 Comparisons of ABFANT and FANT on Instance sko64 within 2500 iterations

In Table 2, the instance name denotes the name of the QAP instance from the QAPLIB (the number indicates its size n). The average quality of solutions obtained by the ABFANT or FANT is the average value over 20 runs.

Each run is guaranteed to be independent of others by starting with different random seeds (The only exception is the instance tho150, where run times is 5, due to its very large execution CPU time). The result in Table 2 indicates that ABFANT is superior to FANT for all but two instances, in terms of quality of solution within a given iteration limit. For the instances of tai30b and sko42, FANT shows a slightly better performance than ABFANT. However, the two instances are the smallest in similar instances such as tai50b, tai80, sko64, and sko72, so that their search spaces are smooth enough to obtain a good solution purely by FANT.

Table 2 Comparisons of ABFANT and FANT for the QAP

Instance name	Average quality of solutions obtained by ABFANT	Average quality of solutions obtained by FANT	Iteration times
Tai30b	637550702.3	**637141763.6**	350
Tai40b	**638030216.9**	638672951.1	350
Tai50b	**459994458.1**	460061936.0	850
Chr22b	**6296.9**	6310.9	750
Chr25a	**4160.5**	4220.7	350
Kra30a	**90246.3**	90309.0	300
Kra30b	**91723.6**	91782	350
Wil50	**48914.3**	48916.2	400
Esc32a	**134.7**	135.6	350
Ste36a	**9603.1**	9612.7	350
Lipa40a	**31827.8**	31831.4	400
Sko42	15867.0	**15857.2**	650
Sko64	**48564.3**	48576.4	2500
Sko72	**66411.0**	66436.0	3500
Tai80b	**821885368.0**	826026281.4	30000
Sko100a	**152135.2**	152214.4	30000
Tho150	**8142022.0**	8151250.0	50000

6 Conclusion and Future Work

In this paper, an approximate-backbone guided fast ant colony algorithm (ABFANT) for the quadratic assignment problem is presented. The main idea of the guided heuristic is to fix the approximate-backbone of QAP instances, which is the common variables of several local optimal permutations, and fixing it can smooth the search space so that local search in FANT will be more efficient in the space. The performance of the ABFANT algorithm was investigated on a set of QAP instances with high and low flow and/or distance dominance value, and compared to the performance of the FANT algorithm, which is one of the best heuristic approaches to the QAP. The ABFANT outperforms the FANT on almost all QAP instances within the given iteration number. Moreover, The idea developed in this paper is generic and applicable to other heuristics of the QAP.

There are two possible future research directions. Firstly, we can apply the idea of approximate-backbone to other combinatorial optimization problems. We believe that the new applications will also get good results. Second, a detailed analysis of the QAP search space will certainly be beneficial to understanding, as well as predicting, the behavior of the approximate-backbone guided fast ant colony algorithm.

References:

[1] Koopmans TC, Berkmann MJ. Assignment problems and the location of economic activities. Econometrica, 1957,25:53−76.

[2] Sahni S, Gonzalez T. P-Complete approximation problems. Journal of the Association of Computing Machinery, 1976,23(3):555−565.

[3] Taillard ED. Comparison of iterative searches for the quadratic assignment problem. Location Science, 1995,3:87−105.

[4] Elshafei AN. Hospital layout as a quadratic assignment problem. Operations Research Quarterly, 1977,28(1):167−179.

[5] Steinberg L. The backboard wiring problem: A placement algorithm. SIAM Review, 1961,3:37−50.

[6] Finke G, Burkard RE, Rendl F. Quadratic assignment problems. Annals of Discrete Mathematics, 1987,31:61−82.

[7] Maniezzo V, Colorni A. The ant system applied to the quadratic assignment problem. IEEE Trans. on Data and Knowledge Engineering, 1999,11(5):769−778.
[8] Gambardella L, Taillard E, Dorigo M. Ant colonies for the QAP. Journal of the Operations Research Society, 1999, 50:167−176.
[9] Nissen V. Solving the Quadratic Assignment Problem with Clues from Nature. IEEE Trans. on Neural Networks, 1994,5(1): 66−72.
[10] Merz P, Freisleben B. A genetic local search approach to the quadratic assignment problem. In: Proc. of the 7th Int'l Conf. of Genetic Algorithms, Morgan Kauffman Publishers, 1997. 465−472.
[11] Misevičius A. A modified simulated annealing algorithm for the quadratic assignment problem. Informatica, 2003,14:497−514.
[12] Ishii S, Sato M. Constrained neural approaches to quadratic assignment problems. Neural Networks, 1998,11:1073−1082.
[13] Battiti R, Tecchiolli G. The reactive tabu search. ORSA Journal on Computing, 1994,2:126−140.
[14] Nissen V, Paul H. A modification of threshold accepting and its application to the quadratic assignment Problem. OR Spektrum, 1995,17:205−210.
[15] Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. Technical Report, CSR 98-1, Sede di Cesena: Universit'a di Bologna, 1998.
[16] Oliveira CAS, Pardalos PM, Resende MGC. GRASP with path-relinking for the QAP. In: MIC2003: the 5th Metaheuristics Int'l Conf. 2003.
[17] Mills P, Tsang E, Ford J. Applying an extended guided local search to the quadratic assignment problem. Submission to Annals of Operation Research 2002.
[18] Drezner Z. A new genetic algorithm for the quadratic assignment problem. INFORMS Journal on Computing, 2003,15(3):320−330.
[19] Drezner Z. Robust heuristic algorithms for the quadratic assignment problem, mathematical methods in manufacturing and logistics. Mathematisches Forschungsinstitut, Oberwolfach, Germany, December 2001.
[20] Kirkpatrick S, Toulouse G. Configuration space analysis of traveling salesman problems. Journal de Physique, 1985,46:1277−1292.
[21] Zhang W. Phase transitions and backbones of 3-sat and maximum 3-sat. In: Proc. of the 7th Int'l Conf. on Principles and Practice of Constraint Programming (CP2001). 2001. 153−167.
[22] Johnson DS, Gutin G, McGeoch LA, Yeo A, Zhang W, Zverovich A. Experimental analysis of heuristics for the ATSP. In: Gutin G, Punnen A, eds. The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, 2002. 445−488.
[23] Climer S, Zhang W. Searching for backbones and fat: A limit-crossing approach with applications. In: Proc. of the AAAI2002. 2002. 707−712.
[24] Zhang W, Rangan A, Looks M. Backbone guided local search for maximum satisfiability. In: Proc. of the 18th Int'l Joint Conf. on AI (IJCAI2003). Acapulco, Mexico, 2003. 1179−1184.
[25] Zou P, Zhou Z, Chen GL, Guo J. 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.pdf
[26] Taillard ED. FANT: Fast ant system. Technical Report, IDSIA-46-98, Lugano: IDSIA, 1998.
[27] Taillard ED, Gambardella L. Adaptive memories for the quadratic assignment problems. Technical Report, IDSIA-87-97, Lugano: IDSIA, 1997.
[28] Schneider J, Froschhammer C, Morgenstern I, Husslein T, Singer JM. Searching for backbones—An efficient parallel algorithm for the traveling salesman problem. Computer Physics Communications, 1996,96:173−188.
[29] Burkard RE, Karisch S, Rendl F. QAPLIB—A quadratic assignment problem library. European Journal of Operational Research, 1991,55:115−119. http://www.opt.math.tu-graz.ac.at/qaplib/
[30] Vollmann TE, Buffa ES. The facilities layout problem in perspective. Management Science, 1966,12(10):450−468.

附中文参考文献:

[25] 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.pdf