SHORT NOTES

MSC 05C85 DOI: 10.14529/jcem170306

AN APPROXIMATION ALGORITHM FOR THE MAXIMUM TRAVELING SALESMAN PROBLEM

A. Yu. Evnin[1]**,** graph98@yandex.ru**,**
N. I. Yusova[1]**,** galericulata@mail.ru**.**
[1] South Ural State University, Chelyabinsk, Russian Federation

The question considered in the travelling salesman problem is the following. For a given list of cities and the distances between each pair of cities, to construct the shortest possible path that visits each city exactly once and returns to the origin city. The problem is an NP-hard problem in combinatorial optimization and is important in operations research. The traveling salesman problem is one of the most famous and heavily researched problems in theoretical computer science. We consider the version, which is the Symmetric Maximum Traveling Salesman Problem. The article describes an approximation algorithm for the maximum traveling salesman problem, based on two known polynomial time approximation algorithms. The accuracy of this algorithm is 25/33.

Keywords: Hamiltonian cycle, traveling salesman problem, approximation algorithm, cycle cover, matching, accuracy of solution.

Introduction

Let $G = (V, E)$ be a complete undirected graph with vertex set V and edge set E. For $e \in E$ let $w(e) \geq 0$ be its weight. For $E' \subseteq E$ we denote $w(E') = \sum\limits_{e \in E'} w(e)$. The maximum traveling salesman problem (Max TSP) is the following: to search for a Hamiltonian cycle (a tour) with maximum total edge weight. The problem is max-SNP-hard [1] and therefore there exists a constant $\beta < 1$ such that a probability to obtain a solution which is better than β is NP-hard.

Let μ be a weight of the optimal tour. A polynomial algorithm guaranteeing an accuracy of the solution 5/7 is described in [2]. Anatoly Serdyukov describes an approximation algorithm with better probability to obtain a solution and an accuracy 3/4 in [3] (a metric case of the maximum traveling salesman problem is considered in [4]).

Serdyukov's algorithm is given in the section 2. Then, we combine ideas from [3] and [2] to form a polynomial algorithm that solving the maximum traveling salesman problem with an estimate of accuracy of the solution 25/33. The improvement of the estimate is small, but we at least demonstrate that the estimate 3/4 can be improved and, therefore, further research in this direction is encouraged. This algorithm is described in section 3.

1. Serdyukov's Algorithm

A cycle cover, or binary 2-matching, is a subgraph in which each vertex in V has degree 2 exactly. A subset of edges of an undirected graph is called matching (perfect matching) if each vertex of the graph is incident at most (exactly, respectively) than to one edge in the given subset [5]. A subtour is a subset of edges of the graph that can be completed to a Hamiltonian cycle (i.e., the subset contains no Hamiltonian cycles and no vertex of degree more than 2). A maximum cycle cover is a cycle cover with maximum total edge weight. A maximum matching is a set of vertex-disjoint edges having maximum total weight. Serdyukov's algorithm is given in Fig. 1.

Serdyukov's Algorithm

input: a complete undirected graph $G = (V, E)$ with weights $w_e, e \in E$.
output: a Hamiltonian cycle.
1. Compute a maximum cycle cover $C = \{C_1, \dots, C_r\}$.
2. Compute a maximum matching W.
3. for i =1… r: Transfer from C_i to W an edge so that W remains a subtour.
4. Complete C into a tour T_1.
5. Complete W into a tour T_2.
Return the tour with maximum weight between T_1 and T_2.
End Serdyukov's Algorithm.

Fig. 1. Serdyukov's algorithm

A weight of the cycle cover is an upper bound for μ, and a weight of the matching is at least $\frac{1}{2}\mu$. Therefore, $w(T_1) + w(T_2) \geq \frac{3}{2}\mu$ and $\max\{w(T_1), w(T_2)\} \geq \frac{3}{4}\mu$. The article [3] also shows how to modify the algorithm such that the estimate is fulfilled regardless the parity of the graph vertices.

2. A New Algorithm

Algorithm Max_TSP is given in Fig. 2. The algorithm constructs three tours and selects a tour with greater weight.

Max_TSP

input: a complete undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ with weights $\mathrm{w_e}, \mathrm{e} \in \mathrm{E}$, a constant $\varepsilon > 0$.
output: a Hamiltonian cycle.
1. Compute a maximum cycle cover $\mathrm{C} = \{\mathrm{C_1}, \dots, \mathrm{C_r}\}$.
2. $\mathrm{T_1} \coloneqq \mathrm{A1}(\mathrm{G}, \mathrm{C}, \varepsilon)$.
3. $(\mathrm{T_2}, \mathrm{T_3}) \coloneqq \mathrm{A2}(\mathrm{G}, \mathrm{C})$.
Return the tour with the maximum weight among $\mathrm{T_1}, \mathrm{T_2}$ and $\mathrm{T_3}$.
End Max_TSP.

Fig. 2. Algorithm Max_TSP

The first tour is constructed, as in [2], by Algorithm A1 (see Fig. 3). The algorithm uses a parameter $\varepsilon > 0$ and considers short cycles, such that $|C_i| \leq \varepsilon^{-1}$, and long cycles in different ways. For each short cycle the algorithm computes a maximum Hamiltonian path on vertices of the short cycle. For each long cycle the algorithm deletes an edge of minimum length. A tour T_1 is formed from the resulting Hamiltonian paths.

The second algorithm (see Fig. 4) is a modified version of Serdyukov's algorithm. The algorithm transfers edges from C to W using a randomized selection step, and generates two subtours. A set formed from W with the transferred edges is expanded arbitrarily to a tour T_2. Another set, consisting of the remaining edges of C, is first expanded by

*A*1

input: a complete undirected graph $G = (V, E)$ with weights w_e, $e \in E$, a cycle cover C, a constant $\varepsilon > 0$.
output: a Hamiltonian cycle T_1.
1. for i =1… r:
    if $|C_i| \le \varepsilon^{-1}$ then
        Compute a maximum Hamiltonian path H_i in the subgraph induced by the vertices of C_i.
    else
        Let e_i be a minimum weight edge of C_i.
        $H_i := C_i \backslash \{e_i\}$
2. Connect $H_1, \dots, H_r$ in some arbitrary order to form a tour T_1.
Return T_1.
End A1.

Fig. 3. Algorithm A1

new edges whose two ends belong to different cycles of C, an then the set is arbitrarily expanded to a tour T_3.

*A*2

input: a complete undirected graph $G = (V, E)$ with weights w_e, $e \in E$.
output: a Hamiltonian cycle T .
1. Compute a maximum cycle cover $C = \{C_1, \dots, C_r\}$.
2. Let E' be the edges of G with two ends in different cycles of C.
3. Compute a maximum weight matching $M' \subseteq E'$.
4. Compute a maximum matching W in G.
5. for i =1… r: construct disjoint nonempty matchings, M_i and M'_i from edges of C_i so that $M_i \cup W$ and $M'_i \cup W$ are subtours and each vertex of C_i is an end of at least one edge from $M_i \cup M'_i$. Transfer either M_i or M'_i from C_i to W, each with probability $\frac{1}{2}$.
6. Complete W into a tour T_2.
7. Let P be the set of paths that were formed from $C_1, \dots, C_r$ after the transfer of edges.
8. $M := \{(i, j) \in M' : \text{i and j have degree 1 in P}\}$. $M \cup P$ consists of paths $P_1^*, \dots, P_s^*$ and cycles $C_1^*, \dots, C_t^*$ such that each cycle contains at least two edges from M.
9. $P^* := \{P_1^*, \dots, P_s^*\}$.
10. for i =1… t: Randomly select an edge $e \in C_i^* \cap M$. $P^* := P^* \cup (C_i^* \backslash e)$.
11. Complete P^* to a tour T_3 by arbitrary addition of edges.
Return T_2, T_3.
End A2.

Fig. 4. Algorithm A2

Lemma 1. *Let Algorithm A2 considers C_i. Then it is possible to construct the desired matchings M_i and M'_i such that both matchings are nonempty, $M_i \cup W$ and $M'_i \cup W$ are subtours, and each vertex of C_i is incident to at least one edge from $M_i \cup M'_i$.*

Proof. Denote the edges of C_i by $e_1, , e_k$ in cyclic order, starting from an arbitrary edge.

Go through C_i starting from e_1. Alternately add edges of C_i to M_i and M'_i. If such addition (for example, e_j to M_i) creates a cycle in $M_i \cup W$ (in particular, if this edge is already in W) then omit the edge e_j and assign instead the next edge, e_{j+1} to M_i. Note that one of the additions is always possible, therefore we never omit two successive edges.

A conflict can take place for the last edge of C_i in the following two cases.

First, a conflict can take place, if we assigned both edges e_1 and e_k to M_i. We solve this conflict as follows: if e_2 was added to M_i' then we just omit e_1. Else, if we omitted e_2 because it was not possible to add e_2 to M_i', then it is possible to add e_1 to M_i'. Therefore, in this case we add e_1 to M_i' rather than to M_i.

Second, a conflict can take place, if both edges e_1 and e_k were omitted. Therefore, we couldn't add e_1 to M_i and we couldn't add e_k to M_i'. In this case we add e_1 to M_i'.

Note that the property that each vertex of C_i is incident to at least one edge in $M_i \cup M_i'$ holds. Also, it is easy to see that M_i and M_i' contain at least one edge.

□

Note that both sets, M_i and M_i', are nonempty. Therefore, after the transfer of any of these matchings to W at least one edge from each cycle is transferred and the remaining edges form a subtour.

Lemma 2. *For each vertex of C_i, the probability that one of the edges, which is incident to the vertex in C_i, will be added to W by Algorithm A2 is at least 1/2.*

Lemma 3. *For every edge $e \in M'$, the probability that the edge is in M (i.e., both of its end vertices have degree 1 in P) is at least 1/4.*

Note that each cycle in $C_1^*, \dots, C_t^*$ contains at least two edges from M, therefore we obtain

Lemma 4. *For every edge $e \in M$, the probability that the edge is deleted by the deletion step of Algorithm A2 is at most 1/2.*

Theorem 1. $max\{w(T_1), w(T_2), w(T_3)\} \geq \dfrac{25(1-\varepsilon)}{33-32\varepsilon}\mu.$

Proof. Let T be an optimal tour, and $T_{int}(T_{ext})$ be the edges of T whose end vertices are in the same (respectively, in different) connectivity components of C. Suppose $w(T_{int}) = \alpha w(T) = \alpha\mu$. Consider the tour T_1. For each short cycle in C of Algorithm A1, a Hamiltonian path having maximum weight is computed. Therefore its contribution to the weight of T_1 is equal to at least the weight of T_{int} in the subgraph induced by its vertices. Since C is a maximum cycle cover, then $w(C_i)$ is at least the weight of T_{int} in the subgraph induced by the vertices of C_i. An edge having minimum weight is deleted in each long cycle, therefore at most a factor ε is subtracted from its weight. Therefore, $w(T_1) \geq (1-\varepsilon)w(T_{int}) \geq (1-\varepsilon)\alpha\mu$.

Consider T_2 and T_3. Let $\delta\mu$ be a total weight of the edges transferred from C to W. Since the original weight of W is at least $\frac{1}{2}\mu$, then $w(T_2) \geq (\frac{1}{2}+\delta)\mu$.

The weight of P, that is the set of paths formed from C after the transfer of edges, is at least $(1-\delta)\mu$. Then the edges are added as follows. First, compute a maximum matching M' over G'. $w(M') \geq \frac{1}{2}w(T_{ext})$, because T_{ext} can be covered by two disjoint matchings in G'. Then obtain M by deleting all edges of M' except such edges whose both ends have degree 1 in P. By Lemma 3, with probability $\frac{1}{4}$, each edge in G' has two ends that have degree 1 in P. Therefore, $w(M) \geq \frac{1}{4}w(M') \geq \frac{1}{8}w(T_{ext}) = \frac{1}{8}(1-\alpha)\mu$.

Next, consider the edges of M in set $M \cup P$ and delete eage $e \in M$ with probability at most $\frac{1}{2}$. The expected weight of the remaining edges is at least $\frac{1}{2}w(M) \geq \frac{1}{16}(1-\alpha)\mu$.

Finally, we obtain the tour T_3 by connecting the remaining edges with P. This step may only increase the weight of the solution. Therefore $w(T_3) \geq ((1-\delta) + \frac{1}{16}(1-\alpha))\mu$.

We conclude that

$$max\{w(T_1), w(T_2), w(T_3)\} \geq max\{(1-\varepsilon)\alpha, \frac{1}{2}+\delta, \frac{17}{16}-\delta-\frac{\alpha}{16}\}\mu.$$

The minimum value of the right hand side is obtained for $\alpha = \dfrac{25}{33-32\varepsilon}$, and then it equals $\dfrac{25(1-\varepsilon)}{33-32\varepsilon}\mu$.

□

Conclusion

The most time consuming parts of the algorithm are the computation of the maximum 2-matching and the computation of the maximum Hamiltonian paths on the subgraphs induced by the short cycles. The first can be computed during time $O(n^3)$, and the second can be computed during time $O(l^2 2^l)$ by applying dynamic programming method for subgraph induced by l vertices. Since for short cycles the inequality $l \leq \varepsilon^{-1}$ holds, then the time is $O(n^2 2^{1/\varepsilon})$. Therefore, the overall complexity of the algorithm is $O(n^2(n+2^{1/\varepsilon})$. If $\varepsilon > 0$ is fixed, then it is possible to find solution with the accuracy at least 25/33 and during time $O(n^3)$.

References

1. Barvinok A.I., Johnson D.S., Woeginger G.J., Woodroofe R. The Maximum Traveling Salesman Problem Under Polyhedral Norms. *IPCO VI LNCS*, 1998, vol. 1412, pp. 195–201. doi: 10.1007/3-540-69346-7_15.
2. Hassin R., Rubinstein S. An Approximation Algorithm for the Maximum Traveling Salesman Problem. *Information Processing Letters*, 1998, vol. 67, no. 3, pp. 125–130.
3. Serdyukov A. I. An Algorithm with an Estimate for the Traveling Salesman Problem of Maximum. *Upravlyaemye Sistemy*, 1984, vol. 25 pp. 80–86. (in Russian).
4. Kostochka A.V., Serdyukov A.I. Polynomial Algorithms with the Estimates 3/4 and 5/6 for the Traveling Salesman Problem of He Maximum. *Upravlyaemye Sistemy*, 1985, vol. 26, pp. 55–59. (in Russian).
5. Gutin G., Punnen A.P. *The Traveling Salesman Problem and Its Variations.* Boston/Dordrecht/London, Kluwer Academic Publishers, 2002.

Aleksandr Yu. Evnin, PhD (Pedagogics), Associate Professor, Department of Applied Mathematics and Programming, South Ural State University (Chelyabinsk, Russian Federation), graph98@yandex.ru.

Natalya I. Yusova, Undergraduate Student, Department of Applied Mathematics and Programming, South Ural State University (Chelyabinsk, Russian Federation), galericulata@mail.ru.

Received May 16, 2017

УДК 519.161 **DOI: 10.14529/jcem170306**

ПРИБЛИЖЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА НА МАКСИМУМ

Н. И. Юсова, А. Ю. Эвнин

Задача коммивояжера состоит в следующем: учитывая список городов и расстояние между каждой парой городов, необходимо составить самый короткий маршрут, по которому каждый город посещается ровно один раз и маршрут заканчивается в том городе, к котором начинался. Это NP-сложная проблема в комбинаторной оптимизации, важной в исследовании операций и теоретической информатике.
Задача коммивояжера – одна из самых известных и исследуемых проблем в информатике. В статье описывается приближенный алгоритм решения задачи коммивояжера на максимум, основанный на двух известных полиномиальных алгоритмах. Точность данного алгоритма составляет 25/33.

Ключевые слова: гамильтонов цикл, задача коммивояжера, приближенный алгоритм, 2-фактор, паросочетание, оценка точности.

Литература

1. Barvinok, A.I. The maximum traveling salesman problem under polyhedral norms / A.I. Barvinok, D.S. Johnson, G.J. Woeginger, R. Woodroofe // IPCO VI LNCS. – 1998. – V. 1412. – P. 195–201.

2. Hassin, R. An approximation algorithm for the maximum traveling salesman problem / R. Hassin, S. Rubinstein // Information Processing Letters. – 1998. – V. 67, № 3. – P. 125–130.

3. Сердюков, А.И. Алгоритм с оценкой для задачи коммивояжера на максимум / А.И. Сердюков // Управляемые системы: сб. науч. тр. – 1984. –Вып. 25 С. 80–86.

4. Косточка, А.В. Полиномиальные алгоритмы с оценками 3/4 и 5/6 для задачи коммивояжера на максимум / А.В. Косточка, А.И. Сердюков // Управляемые системы: сб. науч. тр. – 1985. – Вып. 26, С. 55–59.

5. Gutin, G. The Traveling Salesman Problem and Its Variations / G. Gutin, A.P. Punnen. – Boston/Dordrecht/London: Kluwer Academic Publishers, 2002.

Эвнин Александр Юрьевич, кандидат педагогических наук, доцент, кафедра прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), graph98@yandex.ru.

Юсова Наталья Игоревна, магистрант, кафедра прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), galericulata@mail.ru.

Поступила в редакцию 16 мая 2017 г.