

A $\frac{7}{8}$ -approximation algorithm for metric Max TSP

Refael Hassin and Shlomi Rubinstein¹

Abstract

We present a randomized approximation algorithm for the metric version of undirected Max TSP. Its expected performance guarantee approaches $\frac{7}{8}$ as $n \rightarrow \infty$, where n is the number of vertices in the graph.

Keywords: Analysis of algorithms, maximum traveling salesman problem.

1 Introduction

Let $G = (V, E)$ be a complete (undirected) graph with vertex set V , $|V| = n$, and edge set E . For $e \in E$ let $w(e) \geq 0$ be its weight. For $E' \subseteq E$ we denote $w(E') = \sum_{e \in E'} w(e)$. For a random subset $E' \subseteq E$, $w(E')$ denotes the expected value. The MAXIMUM TRAVELING SALESMAN PROBLEM (Max TSP) is to compute a Hamiltonian circuit (a *tour*) with maximum total edge weight. If the weights $w(e)$ satisfy the triangle inequality, we call the problem METRIC MAXIMUM TRAVELING SALESMAN PROBLEM (metric Max TSP). The problem is max-SNP-hard [1] and therefore there exists some constant $\beta < 1$ such that obtaining a solution with performance guarantee better than β is NP-hard. A survey by Barvinok, Gimadi, and Serdyukov on Max TSP appears in a forthcoming book [2].

We denote the weight of an optimal tour by opt . In [4] a randomized polynomial algorithm is given for Max TSP that guarantees for any $r < \frac{25}{33}$ a solution of expected weight at least $r \cdot opt$. A paper by Kostochka and Serdyukov [5] contains an algorithm with a performance guarantee of $\frac{5}{6}$ for the metric Max TSP. It also contains a $\frac{3}{4}$ -approximation for the more general metric directed case.

This paper contains a randomized algorithm which builds on ideas from the $\frac{3}{4}$ approximation for the general case developed by Serdyukov in [7] and from the $\frac{5}{6}$ -approximation for the metric case developed by Kostochka and Serdyukov in [5]. We start by describing these two algorithms

¹Department of Statistics and Operations Research, School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. {hassin,shlomiru}@post.tau.ac.il

and then show how we use these ideas to construct an approximation algorithm for the metric case with expected performance guarantee which approaches $\frac{7}{8}$ as $n \rightarrow \infty$.

A *binary 2-matching* (also called *2-factor* or *cycle cover*) is a subgraph in which each vertex in V has a degree of exactly 2. A *maximum binary 2-matching* is one with maximum total edge weight. Hartvigsen [3] has shown how to compute a maximum binary 2-matching in $O(n^3)$ time (see [6] for another $O(n^2|E|)$ algorithm). The problem of computing a maximum binary 2-matching is a relaxation of Max TSP and therefore the weight of a maximum binary 2-matching is an upper bound on opt . All the algorithms mentioned in this paper start by constructing a maximum binary 2-matching. A *subtour* in this paper is a subgraph with no non-Hamiltonian cycles or vertices of degree greater than 2. Thus by adding edges to a subtour it can be completed to a tour.

2 Serdyukov's algorithm

Serdyukov's algorithm for the general Max TSP is given in Figure 1. This presentation assumes that n is even.

```

Serdyukov's Algorithm
input A complete undirected graph  $G = (V, E)$  with weights  $w_e$   $e \in E$ .
returns A tour.
begin
  Compute a maximum binary 2-matching  $\mathcal{C} = \{C_1, \dots, C_s\}$ .
  Compute a maximum perfect matching  $M$ .
  for  $i = 1, \dots, r$ :
    Transfer from  $C_i$  to  $M$  an edge so that  $M$  remains a subtour.
  end for
  Complete  $\mathcal{C}$  into a tour  $T_1$ .
  Complete  $M$  into a tour  $T_2$ .
return The tour with maximum weight between  $T_1$  and  $T_2$ .
end Serdyukov's Algorithm

```

Figure 1: Serdyukov's algorithm

Note that it is always possible to transfer an edge from C_i to M as required. The performance guarantee follows easily using the assumption that n is even. $w(\mathcal{C}) \geq opt$ while $w(M) \geq \frac{1}{2}opt$. Thus, $w(T_1) + w(T_2) \geq \frac{3}{2}opt$ and $\max\{w(T_1), w(T_2)\} \geq \frac{3}{4}opt$.

Serdyukov also shows how to modify the algorithm so that the bound holds when n is odd but this part is more involved and we are interested here only in asymptotic bounds so that the parity of n is not important. For example, if n is odd we can randomly choose a vertex and delete it from

the graph. Then apply the algorithm, and finally insert the vertex into the tour in an arbitrary location. The expected loss caused by this procedure is at most a fraction of $1/n$ of the solution's value.

3 The algorithm of Kostochka and Serdyukov

Kostochka and Serdyukov [5] proved the following result (assuming metric weights): Given a binary 2-matching \mathcal{C} with weight $w(\mathcal{C})$ and cycles C_1, \dots, C_s , let $K = \min\{|C_1|, \dots, |C_s|\}$. There exists a tour T with weight $w(T) \geq \frac{2K-1}{2K}w(\mathcal{C})$. To realize this bound they generate from \mathcal{C} a set of $2K$ tours and pick up the longest one. Algorithm `Kostochka_Serdyukov` given in Figure 2 is a randomized version of this algorithm:

```

Kostochka_Serdyukov
  input A complete undirected graph  $G = (V, E)$  with weights  $w_e$   $e \in E$ 
  satisfying the triangle inequality.
  returns A tour.
  begin
    Compute a maximum binary 2-matching  $\mathcal{C} = \{C_1, \dots, C_s\}$ .
    Delete from each cycle  $C_1, \dots, C_s$  a random edge.
    [Alternatively, delete from each cycle a minimum weight edge.]
    Let  $u_i$  and  $v_i$  be the ends of the path  $P_i$  that results from  $C_i$ .
    Give each path a random orientation and form a tour  $T$  by adding
    connecting edges between the head of  $P_i$  and the tail of  $P_{i+1}$  ( $P_{s+1} \equiv P_1$ ).
  return The tour  $T$ .
end Kostochka_Serdyukov

```

Figure 2: `Kostochka_Serdyukov` algorithm

Since $|C_i| \geq K$:

$$w(u_i, v_i) \leq \frac{1}{K}w(C_i).$$

By the triangle inequality,

$$w(u_i, u_{i+1}) + w(v_i, v_{i+1}) + w(u_i, v_{i+1}) + w(v_i, u_{i+1}) \geq 2w(u_i, v_i).$$

$$\begin{aligned} w(T) &= \sum_{i=1}^s w(P_i) + \frac{1}{4} \sum_{i=1}^s (w(u_i, u_{i+1}) + w(v_i, v_{i+1}) + w(u_i, v_{i+1}) + w(v_i, u_{i+1})) \\ &\geq \sum_{i=1}^s w(P_i) + \frac{1}{2} \sum_{i=1}^s w(u_i, v_i) \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^s w(C_i) - \frac{1}{2} \sum_{i=1}^s w(u_i, v_i) \\
&\geq \left(1 - \frac{1}{2K}\right) \sum_{i=1}^s w(C_i) = \left(1 - \frac{1}{2K}\right) w(\mathcal{C}).
\end{aligned}$$

This result leads to a $\frac{5}{6}$ approximation by using a maximum binary 2-matching for \mathcal{C} , noting that $K \geq 3$, and that $w(\mathcal{C}) \geq \text{opt}$.

4 Improving the bound

We will show now how to compute in polynomial time a random solution of expected weight at least $\frac{7}{8}$ times the optimal. Like in Serdyukov's algorithm, we move one edge from each cycle of the maximum binary 2-matching to the maximum matching. By the method of Kostochka and Serdyukov, half of the weights lost from the binary 2-matching can be reclaimed when combining the cycles into a tour. We try to reclaim more when combining the matching (with the added edges) into a cycle.

The proposed algorithm is given in Figure 3.

Theorem 1 *The expected weight of the tour T returned by Algorithm Metric satisfies $w(T) \geq (\frac{7}{8} - O(\frac{1}{\sqrt{n}}))\text{opt}$.*

Proof: Since the problem of computing a binary 2-matching of maximum weight is a relaxation of Max TSP, the binary 2-matching \mathcal{C} computed in Step 1 satisfies $w(\mathcal{C}) \geq \text{opt}$. Since any tour can be decomposed into two disjoint perfect matchings, $w(M) \geq \text{opt}/2$.

In Step 2, the algorithm selects (sequentially) a pair of edges, which we call *candidates*, from each cycle of \mathcal{C} and deletes one of them, where the selection of the edge to be deleted is with probability $1/2$. Denote by α the relative weight in \mathcal{C} of the edges that were candidates for deletion. The expected relative weight of the edges that were actually deleted is $\alpha/2$. However, as explained above with respect to Algorithm Kostochka_Serdyukov, half of this weight is regained when connecting the resulting paths to a tour T_1 . Hence,

$$w(T_1) \geq \left[1 - \frac{\alpha}{2} + \frac{\alpha}{4}\right] w(\mathcal{C}) \geq \left(1 - \frac{\alpha}{4}\right) \text{opt}. \quad (1)$$

We now consider Step 3. The algorithm adds to M one of the pair of candidates from each cycle of \mathcal{C} . The expected weight of the added edges is $(\alpha/2)w(\mathcal{C})$. Note that if a vertex v is incident

```

Metric
input A complete undirected graph  $G = (V, E)$  with weights  $w_e$   $e \in E$ 
satisfying the triangle inequality.
returns A tour.
begin
Step 1
    Compute a maximum binary 2-matching  $C = \{C_1, \dots, C_s\}$ .
    Compute a maximum perfect matching  $M$ .
end Step 1
Step 2
for  $i = 1, \dots, s$ :
    Identify  $e, f \in E \cap C_i$  such that both  $M \cup \{e\}$  and  $M \cup \{f\}$  are subtours.
    Randomly choose  $g \in \{e, f\}$  (each with probability  $1/2$ ).
     $P_i := C_i \setminus \{g\}$ .
     $M := M \cup \{g\}$ .
end for
Complete  $\cup_{i=1}^s P_i$  into a tour  $T_1$  as in Algorithm Kostochka_Serdyukov.
end Step 2
Step 3
Let  $S :=$  set of end nodes of paths in  $M$ .
Compute a random perfect matching  $M_S$  over  $S$ .
Delete an edge from each cycle in  $M \cup M_S$ .
Arbitrarily complete  $M \cup M_S$  into a tour  $T_2$ .
end Step 4
return The tour  $T$  with maximum weight between  $T_1$  and  $T_2$ .
end Metric

```

Figure 3: New algorithm for metric Max TSP

to two candidates then certainly $v \notin S$. If v is not incident to a candidate then certainly $v \in S$. Finally, if v is incident to one candidate then $v \in S$ with probability $1/2$ (which is the probability that this candidate is *not* chosen).

Let $|S| = k + 1$. For $i \in S$, exactly one edge from $\{(i, j) | j \in S \setminus i\}$ is chosen to M_S . Thus, for an edge $(i, j) \in E \cap (S \times S)$ the probability that this edge will be selected to M_S is $1/k$. If (i, j) is selected, charge its weight $w(i, j)$ in the following manner: Suppose that i is incident to edges $e', e'' \in \mathcal{C}$. If none of these edges was a candidate, charge $w(i, j)/4$ to each of e' and e'' . If one of them, say e' was a candidate, charge $w(i, j)/2$ to e'' (and nothing to e'). Note that it cannot be that both e' and e'' were candidates since in such a case $i \notin S$. The expected weight charged to an edge $(g, h) \in \mathcal{C}$ that was not a candidate is then $(1/k)[\sum_{r \in S \setminus g} w(r, g)/4 + \sum_{r \in S \setminus h} w(r, h)/4]$. Note that the $1/4$ factor arises also in the case that the vertex, say g , is incident with a candidate edge on \mathcal{C} , since in such a case $g \in S$ with probability $1/2$ and then it gets half of the weight of the edge

of M_S which is incident to it. By the triangle inequality, $w(r, g) + w(r, h) \geq w(g, h)$ so that the above sum is at least $w(g, h)/4$. We conclude that $w(M_S) \geq w(\mathcal{C})(1 - \alpha)/4$ and consequently

$$w(M \cup M_S) \geq \left(0.5 + \frac{\alpha}{2} + \frac{1 - \alpha}{4}\right)w(\mathcal{C}) = \left(\frac{3 + \alpha}{4}\right)opt.$$

Finally, the algorithm deletes edges from cycles in $M \cup M_S$. We claim that $|S| \geq n/3$. The reason is that the perfect matching computed in Step 1 had all n vertices of V with degree 1. Then, one candidate from each cycle of \mathcal{C} was added to M . The number of added edges is equal to the number of cycles which is at most $n/3$. Therefore, after the addition of these edges the degrees of at most $2n/3$ vertices became 2, while at least $n/3$ vertices remained with degree 1. The latter vertices are precisely the set S , and this proves that $|S| \geq n/3$. Since M_S is a random matching, the probability that an edge of $M \cup M_S$ is contained in a cycle whose size is smaller than \sqrt{n} is at bounded from above by

$$\frac{1}{\frac{n}{3}} + \frac{1}{\frac{n}{3} - 1} + \cdots + \frac{1}{\frac{n}{3} - \sqrt{n}} \leq \frac{\sqrt{n}}{\frac{n}{3} - \sqrt{n}} = O\left(\frac{1}{\sqrt{n}}\right).$$

(The j -th term in the left-hand side of this expression bounds the probability that a cycle containing exactly j edge from M_S is created.) Therefore, the expected weight of edges deleted in this step is $O(1/\sqrt{n})w(M \cup M_S)$ and

$$w(T_2) \geq \left(\frac{3 + \alpha}{4}\right)\left(1 - O\left(\frac{1}{\sqrt{n}}\right)\right)opt. \quad (2)$$

Combining (1) and (2) we get that when $\alpha \leq 1/2$, $w(T_1) \geq (7/8)opt$ and when $\alpha \geq 1/2$ $w(T_2) \geq (7/8)(1 - O(1/\sqrt{n}))opt$. Thus,

$$w(T) = \max\{w(T_1), w(T_2)\} \geq \left(\frac{7}{8} - O\left(\frac{1}{\sqrt{n}}\right)\right)opt.$$

■

References

- [1] A. Barvinok, D. S. Johnson, G. J. Woeginger, and R. Woodroffe, "Finding maximum length tours under polyhedral norms," *Proceedings of IPCO VI, Lecture Notes in Computer Science* **1412** (1998) 195-201.
- [2] A. Barvinok, E. Kh. Gimadi, and A. I. Serdyukov, *The maximum traveling salesman problem*, in *The Traveling Salesman Problem and Its Variations*, G. Gutin and A. Punnen (eds.), Kluwer Academic Publishers, forthcoming.
- [3] D. Hartvigsen, *Extensions of Matching Theory*. Ph.D. Thesis, Carnegie-Mellon University (1984).
- [4] R. Hassin and S. Rubinstein, "Better approximations for Max TSP", *Information Processing Letters* **75** (2000), 181-186.
- [5] A. V. Kostochka and A. I. Serdyukov, "Polynomial algorithms with the estimates $\frac{3}{4}$ and $\frac{5}{6}$ for the traveling salesman problem of the maximum" (in Russian), *Upravlyaemye Sistemy* **26** (1985) 55-59.
- [6] J. F. Pekny and D. L. Miller, "A staged primal-dual algorithm for finding a minimum cost perfect two-matching in an undirected graph", *ORSA Journal on Computing* **6** (1994), 68-81.
- [7] A. I. Serdyukov, "An algorithm with an estimate for the traveling salesman problem of the maximum" (in Russian), *Upravlyaemye Sistemy* **25** (1984) 80-86.