Algorithmic Operations Research Vol.5 (2010) 13–20

Approximating the metric 2-Peripatetic Salesman Problem

Federico Della Croce

DAI - Politecnico di Torino, Italy

Vangelis Th. Paschos

LAMSADE - CNRS UMR 7024 and Université Paris Dauphine, France

Roberto Wolfler Calvo

LIPN - Université PARIS 13, France

Abstract

This paper deals with the 2*-Peripatetic Salesman Problem for the case where costs respect the triangle inequality. The aim is to determine* 2 *edge disjoint Hamiltonian cycles of minimum total cost on a graph. We first present a straightforward* $9/4$ *approximation algorithm based on the well known Christofides algorithm for the travelling salesman problem. Then we propose a* $2(n-1)/n$*-approximation polynomial time algorithm based on the solution of the minimum cost two-edge-disjoint spanning trees problem. Finally, we show that by partially combining these two algorithms, a* $15/8$ *approximation ratio can be reached if a* $5/4$ *approximation algorithm can be found for the related problem of finding two edge disjoint subgraphs consisting of a spanning tree and a Hamiltonian cycle of minimum total cost.*

Key words: Peripatetic Salesman Problem, Hamiltonian Circuit, Approximation

1. Introduction

The m-Peripatetic Salesman Problem ($m-PSP$) is defined on a complete n-vertex (undirected) graph $G=(V,E)$ where $V=\{1,\dots,n\}$ is a vertex set and $E=\{(i,j):i,j\in V, i<j\}$ is an edge set. A cost $c_e\in\mathbb{R}$ is assigned to each edge $e\in E$. The problem consists of determining m edge disjoint Hamiltonian cycles $H_1,\dots,H_m\subset E$, of minimum total cost $\sum_{k=1}^{m}C(H_k)$ (where $C(H_k)=\sum_{e\in H_k}c_e$) on G. When $m=1$ the $m-PSP$ reduces to the *Travelling Salesman Problem* (TSP). Let indicate the solution cost obtained by applying a generic algorithm A as $C(A)$. Let OPT represent the optimal m-PSP solution (where OPT_i is the optimal i-th cycle) and $C(OPT)=\sum_{i=1}^{m}C(OPT_i)$ its value. The optimal TSP solution is indicated with TSP_1 and $C(TSP_1)$ is its cost. Finally, $\rho(A)$ is the approximation ratio for the algorithm A, (i.e. $\frac{C(A)}{C(OPT)}$). In the sequel we consider the case with $m=2$ ($2-PSP$), where we assume $n\geq 5$ to avoid infeasible cases.

The $m-PSP$ was introduced by Krarup [4]. Applications include the design of watchman tours [5] where it is often important to assign a set of edge-disjoint rounds to the watchman in order to avoid always repeating the same tour and thus enhance security. In the same spirit, De Kort [7] cites a network design application where, in order to protect the network from link failure, several edge-disjoint cycles must be determined. This author also mentions a scheduling application of the $2-PSP$ where each job must be processed twice by the same machine but technological constraints prevent the repetition of identical job sequences.

In [8] $2-PSP$ was shown to be $NP-hard$ by reduction from the Hamiltonian path problem. In [6] some polynomially solvable cases of $2-PSP$ are considered. In [8], [9] and [10] upper and lower bounds calculation are presented and a branch and bound method is proposed. Polyhedral approaches for the m-PSP are presented in [2,3]. For the metric $2-PSP$ (when the cost function satisfies the triangular inequality), an approximation algorithm with performance ratio $9/4$ is

Email: Federico Della Croce [federico.dellacroce@polito.it], Vangelis Th. Paschos [paschos@lamsade.dauphine.fr], Roberto Wolfler Calvo [roberto.wolfler@lipn-paris13.fr].

proposed in [12]. Recently, Ageev et al. [1] proposed a 2-approximation polynomial-time algorithm starting from the solution of the minimum cost 2 edge-disjoint spanning trees problem. The proof of this result is however rather technical, non-intuitive and pretty hard to follow.

In his seminal paper, Krarup [4] discussed the following simple heuristic (denoted hereafter KH) for $2-PSP$: solve a first TSP on the initial graph G, remove from the graph the edges used in the TSP solution, solve a second TSP on the remaining graph and merge the two Hamiltonian cycles found to obtain the $2-PSP$ solution. If a complete graph with general cost function is considered, KH is shown in [4] to give unbounded error, even if the TSPs are solved optimally (we denote hereafter by KH_{OPT} such version of KH). Indeed, this is the case also for a complete bipartite graph: consider the following example with ten vertices $(1, \dots, 10)$ with the odd nodes in the left partition and the even nodes in the right partition. Edges costs are defined as follows: $c_{1,2} = c_{2,3} = c_{3,4} = c_{4,5} = c_{5,6} = c_{6,7} = c_{7,8} = c_{8,9} = c_{9,10} = c_{10,1} = \epsilon$; $c_{1,8} = c_{3,10} = c_{4,9} = M$; all the other edges have cost equal to 2ϵ. The unique optimal solution of the TSP is the cycle $\alpha = 1-2-3-4-5-6-7-8-9-10$, whose value is equal 10ϵ. Removing those edges all remaining Hamiltonian cycles contain either edge $e_{1,8}$ and/or edge $e_{3,10}$ and/or $e_{4,9}$ inducing a cost function value greater than or equal to M. The optimal solution of $2-PSP$ has value $\leq 31\epsilon$ obtained with the cycles $\lambda = 1-4-3-2-9-6-7-8-5-10-1$ whose value is 15ϵ and $\mu = 1-2-5-4-7-10-9-8-3-6-1$ whose value is 16ϵ. Thus the ratio $\frac{C(A)}{C(OPT)} \geq \frac{M}{31\epsilon} \to +\infty$ for M big and ϵ small suitably chosen.

In this paper we investigate the metric case. We first show that from the Hamiltonian cycle HC_1 computed by applying Christofides' algorithm (CA) [11] it is possible to derive a second disjoint cycle HC_2 such that the cost of HC_2 is at most twice the cost of HC_1. Correspondingly we deduce a straightforward $9/4$ approximation algorithm. A second approximation algorithm is then reported in Section 2.2.. It is based on the idea of solving first (similarly to [1]) the minimum cost two-edge-disjoint spanning trees problem which is known to be polynomially solvable. Then, by duplicating the edges, each tree is transformed into an Eulerian cycle and by means of a patching procedure similar to the one proposed in [11], two edge disjoint Hamiltonian cycles are obtained. We show that this second algorithm reaches a $2\frac{n-1}{n}$-approximation ratio.

Incidentally, we notice that this result was obtained somewhat contemporarily and independently from [1]) (a preliminary version of this work was already presented at the AGaPe - Algorithmique à Garanties de Performance / Algorithms with Performance Guarantees - workshop held at the University Paris-Dauphine, France, in December 2006) and its proof is much simpler and more intuitive than the one in [1].

Finally, we show that by partially combining the proposed two algorithms, a $15/8$ approximation ratio can be reached if a $5/4$ approximation algorithm can be found for the related problem of finding two edge disjoint subgraphs consisting in a spanning tree and a Hamiltonian cycle of minimum total cost.

2. Approximation results

2.1. *A simple $\frac{9}{4}$-approximation algorithm*

In this paragraph we present a first simple approximation algorithm which gives a $9/4$ approximation ratio. Let $G(V, E)$ be a complete graph with a cost c_e associated to each edge such that the triangular inequality holds, i.e. $c_{e_{ij}} + c_{e_{jk}} \geq c_{e_{ik}}$. Consider the following algorithm and the related figure (Figure 1).

Algorithm 1 The $9/4$-approximation ratio

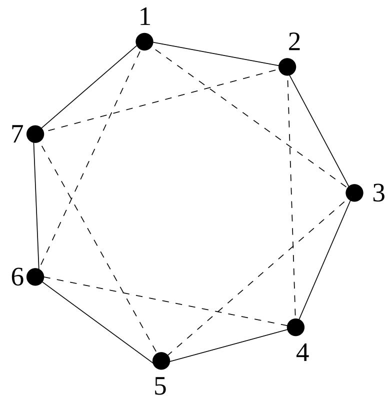
Phase 1) Apply Christofides' Algorithm (CA) for finding a TSP solution and build the Hamiltonian tour $H_1 = (1/2/\dots/n/1)$ (let w.l.o.g. e_{12} be the smallest cost edge among the ones used in H_1);
Phase 2) If n is odd **then** the second disjoint tour H_2 is $1/3/\dots/n-2/n/2/4/\dots/n-1/1$,
else the second disjoint tour H_2 is $1/3/\dots/n-1/2/n/n-2/n-4/\dots/4/1$.
return The edge disjoint Hamiltonian cycles H_1 and H_2.

It is well known that CA for the TSP [11] has approximation ratio $\leq \frac{3(n-1)}{2n}$. Then, the following theorem holds.

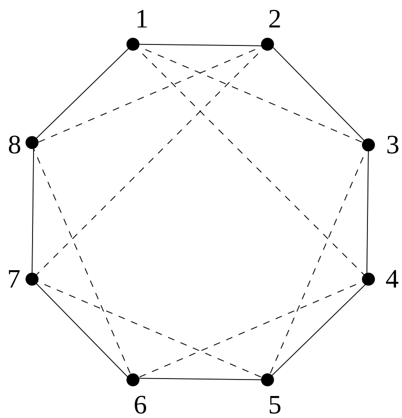
Theorem 1 *Algorithm 1 is polynomial and has a $\frac{9}{4}$-approximation ratio.*

Proof. The polynomiality trivially holds from the polynomiality of Cristofides' algorithm. For the ratio: if n is odd, then $C(H_1) = C(CA)$ and:

$$\begin{aligned} C(H_2) &= C(1/3/\dots/n-2/n/2/4/\dots/n-1/1) \\ &\leq C(1/2/3/\dots/n/1/2/3/\dots/n/1) \\ &= 2C(H_1) \end{aligned}$$



(a)



(b)

Figure 1. An example of Algorithm 1

Therefore:

$$\rho(H_1 \cup H_2) = \frac{C(H_1 \cup H_2)}{C(OPT)} \leq \frac{C(H_1 \cup H_2)}{2C(TSP_1)} \leq \frac{\frac{3(n-1)}{2n} + \frac{3(n-1)}{n}}{2} = \frac{\frac{9(n-1)}{2n}}{2} \leq \frac{9}{4}$$

Analogously, if n is even, then $C(H_1) = C(CA)$ and:

$$\begin{aligned} C(H_2) &= C(1/3/\dots/n-1/2/n/n-2/n-4/\dots/4/1) \\ &\leq C(1/2/3/4/\dots/n-2/n-1/n/1/2/1/n \\ &\quad /n-1/n-2/n-3/n-4/\dots/4/3/2/1) \\ &= 2C(H_1) + 2c_{e_{12}} \quad \leq \quad \frac{2n+1}{n} C(H_1) \end{aligned}$$

Therefore:

$$\begin{aligned} \rho(H_1 \cup H_2) &\leq \frac{C(H_1 \cup H_2)}{2C(TSP_1)} \\ &\leq \frac{\frac{3(n-1)}{2n} + \frac{2(n+1)}{n}\frac{3(n-1)}{2n}}{2} \\ &= \frac{\frac{3(n-1)}{2n} + \frac{6(n^2-1)}{2n^2}}{2} \leq \frac{9}{4} \end{aligned}$$

The proof of the theorem is now complete. ∎

Remark

Phase 2 of Algorithm 1 is actually applicable to any cycle $A = (1/2/\dots/n/1)$, namely, for any given A with cost $C(A)$, it is possible to derive a disjoint cycle B with cost $C(B) \leq \frac{2(n+1)}{n} C(A)$. This induces the following upper and lower bounds on the approximation ratio of Algorithm KH_{OPT}.

Corollary 2 $\frac{3n+2}{2n} \geq \frac{C(KH_{OPT})}{C(OPT)} \geq 7/6$

Proof. For the upper bound, by computing TSP_1 and then finding a second disjoint cycle with cost not superior to $\frac{2(n+1)}{n} C(TSP_1)$ as in Algorithm 1, we get $\frac{C(KH_{OPT})}{C(OPT)} \leq \frac{C(TSP_1) + \frac{2(n+1)}{n} C(TSP_1)}{C(OPT)} = \frac{\frac{3n+2}{n} C(TSP_1)}{C(OPT)} = \frac{\frac{3n+2}{n} C(TSP_1)}{C(OPT_1) + C(OPT_2)} \leq \frac{\frac{3n+2}{n} C(TSP_1)}{2C(TSP_1)} = \frac{3n+2}{2n}$. For the lower bound, consider the following example with six vertices $(1, \dots, 6)$ and edge costs defined as follows: $c_{1,2} = c_{1,6} = c_{2,3} = c_{3,4} = c_{4,5} = c_{5,6} = 1$; $c_{1,3} = c_{1,5} = c_{2,4} = c_{2,6} = c_{3,5} = c_{4,6} = 1 + \epsilon$; $c_{1,4} = c_{2,5} = c_{3,6} = 2$. The unique solution of the TSP is the cycle $\alpha = 1-2-3-4-5-6-1$, whose value is equal to 6. Removing those edges all remaining three Hamiltonian cycles contain two edges with cost 2 inducing a cost function value ≥ 8. Hence each of these three $2-PSP$ solutions has cost ≥ 14. However, the optimal solution of $2-PSP$ has value $12+6\epsilon$ obtained with the cycles $\lambda = 1-2-3-4-6-5-1$ and $\mu = 1-3-5-4-2-6-1$. Thus the ratio is $\frac{C(KH_{OPT})}{C(OPT)} \geq \frac{14}{12+6\epsilon} \to \frac{7}{6}$ for $\epsilon \to 0$ suitably chosen. ∎

2.2. *A $\frac{2(n-1)}{n}$-approximation algorithm*

Given a complete graph $G(V, A)$, we first determine two edge-disjoint spanning trees of minimum cost (it is well known that such problem is polynomially solvable, see for instance [13]) and such cost obviously constitutes a lower bound to $2-PSP$. Let define in the following these spanning trees as tree A and tree B.

More precisely, if $C(A) + C(B)$ constitutes the cost function value of the minimum cost two-edge-disjoint spanning trees problem, we have $\frac{C(A)+C(B)}{C(OPT)} \leq \frac{n-1}{n}$. Indeed, this occurs because $C(A)+C(B) \leq C(HP_1)+C(HP_2)$ where HP_1 and HP_2 denote the Hamiltonian paths derived from the optimal solution by eliminating the two largest edges in OPT_1 and OPT_2 respectively. But, then, $C(HP_1) \leq \frac{n-1}{n}C(OPT_1)$ and $C(HP_2) \leq \frac{n-1}{n}C(OPT_2)$ and, hence, $C(A)+C(B) \leq \frac{n-1}{n}(C(OPT_1)+C(OPT_2)) = \frac{n-1}{n}C(OPT)$. The algorithm is divided into two main phases. We first transform A into a Hamiltonian cycle A^* while possibly modifying B but in such a way that B remains a tree. Then, also B is transformed into a Hamiltonian cycle while possibly modifying A^* but in such a way to keep A^* Hamiltonian.

In the first phase, the algorithm duplicates all edges of tree A which becomes Eulerian and is denoted in the following as A'. Given A', the algorithm iteratively selects the node with largest degree and substitutes a pair of its edges with their diagonal in a way similar (but not identical) to what is done in Christophides algorithm [11].

To apply these patching iterations, we need to take into account not only that substituting a pair of edges belonging to the same node with their diagonal may disconnect the Eulerian cycle, but also the possibility that the diagonal may belong to tree B. In this case an exchange of edges between A' and B is operated in order to keep a Eulerian cycle and a tree. In all cases, at each iteration the number of edges of A' is reduced until A' becomes Hamiltonian. At the end of this phase, we have then a Hamiltonian cycle A^* and a tree B.

The second phase starts by duplicating all edges of tree B which becomes then Eulerian and is denoted in the following as B'. Then, the algorithm transforms B' into a Hamiltonian cycle B^* while keeping A^* Hamiltonian. This is done in a way similar to what was described above for obtaining A^* except for those subcases requiring an exchange of edges that are handled differently.

In both phases, the algorithm always operates on the largest degree node of an Eulerian cycle: in the following we present the exhaustive cases that may occur.

(1) Phase 1: the considered Eulerian cycle is A', while B is a tree and the largest degree node of A' has at least six edges: then, either it has at least two double edges (case 1a), or it has at least four single edges (case 1b).

  (a) The node has at least two double edges, namely it has two double edges and at least one further edge which may be either double (then three double edges) or single (then two double edges and one single edge) as represented in Figure 2 where 1 is the highest degree node, 2 and 3 are the nodes adjacent to 1 by means of double edges, while 4 is the node adjacent to 1 by means of a single or double edge indicated with a dashed line: then, either we substitute e_{12} and e_{13} with e_{23} or e_{12} and e_{14} with e_{24} or e_{13} and e_{14} with e_{34}. Notice that since e_{23}, e_{24} and e_{34} form a three edges cycle, then at least one of these edges is not included in B and is therefore available for A'. Also, note that the graph remains Eulerian as e_{12} and e_{13} are double edges.

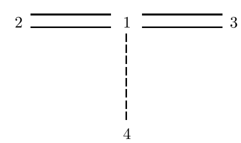


Figure 2. Case 1a

  (b) The node has at least four single edges (as represented in Figure 3.i where 1 is the highest degree node and 2, 3, 4 and 5 are the nodes adjacent to 1).

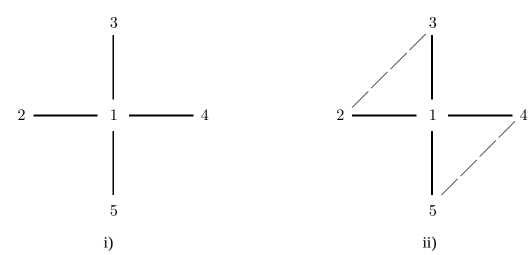


Figure 3. (i) Case 1b; (ii) Patching inducing disconnection of A' in case 1b.

There are six possible ways of substituting two edges with their diagonal, namely substitutions $s_1 = \{e_{12}$ and e_{13} with $e_{23}\}$, or $s_2 = \{e_{12}$ and e_{14} with $e_{24}\}$, or $s_3 = \{e_{12}$ and e_{15} with $e_{25}\}$, or $s_4 = \{e_{13}$ and e_{14} with $e_{34}\}$, or $s_5 = \{e_{13}$ and e_{15} with $e_{35}\}$, or $s_6 = \{e_{14}$ and e_{15} with $e_{45}\}$. Also, as edges e_{23}, e_{24}, e_{25}, e_{34}, e_{35} and e_{45} form several subcycles, at least some of them must be available. Assume w.l.o.g. that e_{23} is available (the other options work in a similar way) that is it does not belong to B: then, substituting e_{12} and e_{13} with e_{23} either keeps A' Eulerian or disconnects A' (see Figure 3.ii where the dashed lines indicate that there is a chain from 2 to 3 and from 4 to 5). In the first case, we are done. In the second case, as A' is Eulerian, then substitutions s_2, s_3, s_4 and s_5 do not disconnect A'. But then, as edges e_{24}, e_{25}, e_{34} and e_{35} form a subcycle, then either s_2, or s_3, or s_4 or s_5 are available and do not disconnect A'.

(2) Phase 1 (cont.): the considered Eulerian cycle is A', while B is a tree and the largest degree node of A' has four edges: then, either it has four single edges (case 2a), or it has one double edge and two single edges (case 2b), or it has two double edges (case 2c).
   (a) The node has four single edges: see case 1b.
   (b) The node has one double edge and two single edges (as represented in Figure 4) where 1 is the highest degree node and 3, 4 are the nodes adjacent to 1 by means of single edges, while 2 is the node adjacent to 1 by means of a double edge.

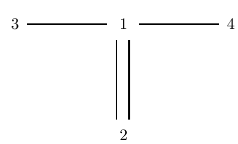


Figure 4. Case 2b

Then, either we substitute e_{12} and e_{13} with e_{23} or e_{12}, e_{14} with e_{24}. If any of e_{23} or e_{24} does not belong to tree B, then we simply apply the corresponding substitution, else the substitution is coupled to an edge exchange between A' and B. If the first substitution occurs, that is edge e_{23} is taken from B and assigned to A', then, correspondingly, edge e_{13} eliminated from A' is assigned to B. Alternatively, edge e_{24} is taken from B and assigned to A' and, correspondingly, edge e_{14} eliminated from A' is assigned to B. To see that the exchange guarantees that B remains a tree, notice that, by definition of tree, node 1 is connected to 2, 3, 4 in B through a path that reaches first either 2, or 3, or 4. If the path reaches first 3, then by substituting in B edge e_{24} with edge e_{14} we guarantee that B remains a tree. Similarly, if the path reaches first 4, then by substituting in B edge e_{23} with edge e_{13} we guarantee that B remains a tree. Finally, if the path reaches first 2, it is indifferent to substitute in B edge e_{23} with edge e_{13} or edge e_{24} with edge e_{14} and in all cases A' remains Eulerian while B remains a tree.

   (c) The node has two double edges and subcases 2a and 2b do not hold. Then, this configuration may only apply if all nodes (except two - the head and the tail) have two double edges, that is when the Eulerian cycle is actually a double path (as represented in Figure 5) $P = (1/2/\ldots/n)$ where 1 is the head node and n is the tail node with 1 and n having one double edge and all the other nodes having two double edges.

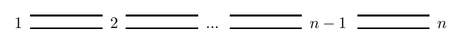


Figure 5. The double path of case 2c

Then, we obtain the Hamiltonian cycle $(1/2/\ldots/n/1)$ by keeping just one single edge for each pair of adjacent nodes and adding the edge e_{1n}. In the case where e_{1n} belongs to tree B, then e_{1n} is substituted in B by any available edge e_{ij} that makes B becoming again a spanning tree.

(3) Phase 2: the considered Eulerian cycle is B', while A^* is a Hamiltonian cycle and the largest degree node of B' has at least six edges: then, either it has at least two double edges (case 3a), or it has at least four single edges (case 3b).
   (a) The node has at least two double edges: see case 1a as no exchanges are involved.
   (b) The node has at least four single edges: see case 2b as no exchanges are involved.

(4) Phase 2 (cont.): the considered Eulerian cycle is B', while A^* is a Hamiltonian cycle and the largest degree node of B' has four edges: then, either it has four single edges (case 4a), or it has one double edge and two single edges (case 4b), or it has two double edges (case 4c).
   (a) The node has four single edges: see case 2a as no exchanges are involved.
   (b) The node has one double edge and two single edges where 1 is the highest degree node and 3, 4 are the nodes adjacent to 1 by means

of single edges, while 2 is the node adjacent to 2 by means of a double edge. Then, either we substitute e_{12} and e_{13} with e_{23} or e_{12}, e_{14} with e_{24}. If any of e_{23} or e_{24} does not belong to cycle A^*, then we apply the corresponding substitution. Alternatively, let 5 and 6 be the nodes adjacent to 2 in A^*. Then, if edge e_{25} (e_{26}) does not belong to B', we apply an edge exchange between B' and A^* by substituting in B' edges e_{12} and e_{13} with edge e_{23} taken from A^* and, in the same time, substituting in A^* edges e_{23} and e_{15} (e_{16}) with edges e_{13} and e_{25} (e_{26}). Finally, if both e_{25} and e_{26} belong to B' (see Figure 6), then no exchange is necessary, as only cycle B' is affected.

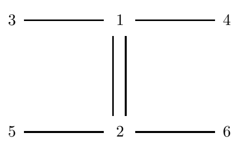


Figure 6. A configuration of case 4b

Indeed, either edges e_{12}, e_{13} and e_{25} are substituted by edge e_{35}, or edges e_{12}, e_{13} and e_{26} are substituted by edge e_{36}, or edges e_{12}, e_{14} and e_{25} are substituted by edge e_{45}, or edges e_{12}, e_{14} and e_{26} are substituted by edge e_{46} and at least three among the edges e_{35}, e_{36}, e_{45} and e_{46} are available. This occurs because these edges form a 4-edge subcycle and because A^* already uses edges e_{15}, e_{16}, e_{23} and e_{24}. But then, at least one of the substitutions related to the available edges does not disconnect the cycle (it can be shown by means of an analysis analogous to the one presented in case 1b) and can therefore be applied.

(c) The node has two double edges. We consider here the case where there are no nodes with four edges according to subcases 4a or 4b. But then, as for subcase 2c the Eulerian cycle is actually a path (as represented in Figure 5) $P = (1/2/\ldots/n)$ where 1 is the head node and n is the tail node with 1 and n having one double edge and all the other nodes having two double edges. Then, if e_{1n} does not belong to A^*, we obtain the Hamiltonian cycle $(1/2/\ldots/n/1)$ by keeping just one single edge for each pair of adjacent nodes and adding the edge e_{1n}. Alternatively, e_{1n} belongs to A^*. Then consider the three following Hamiltonian cycles:

$$H_A = (1/2/n/n-1/\ldots/6/5/4/3/1)$$
$$H_B = (1/2/3/n/n-1/\ldots/6/5/4/1)$$
$$H_C = (1/2/3/4/n/n-1/\ldots/6/5/1)$$

All these cycles contain $n-2$ edges of the path P plus two other edges, namely e_{2n} - e_{13} for H_A, e_{3n} - e_{14} for H_B and e_{4n} - e_{15} for H_C. But then at least one of these Hamiltonian cycles can be obtained as A^* already contains edge e_{1n} and therefore can contain at most one more edge connected to node 1 and another connected to node n.

The pseudo-code of the algorithm is given in Algorithm 2.

Theorem 3 *Algorithm 2 has a $\frac{2(n-1)}{n}$-approximation ratio and requires polynomial time.*

Proof. Consider the optimal solution value $C(OPT)$. As $C(A) + C(B) \leq \frac{n-1}{n}C(OPT)$, we know that duplicating the edges of both trees does not exceed $\frac{2(n-1)}{n}C(OPT)$, namely $C(A') + C(B') = 2C(A) + 2C(B) \leq \frac{2(n-1)}{n}C(OPT)$. We want to prove that $C(A^*) + C(B^*) \leq 2(C(A) + C(B)) \leq \frac{2(n-1)}{n}C(OPT)$. To this extent, as far as Phase 2 is considered, if we denote by A_1^* (A_2^*) and B_1' (B_2') the Hamiltonian cycle A^* and the Eulerian cycle B' before (after) applying any considered subcase, we need only that $C(A_2^*) + C(B_2') \leq C(A_1^*) + C(1_2')$. Instead, as far as Phase 1 is concerned, if we denote by A_1' (A_2') and B_1 (B_2) the Eulerian cycle A' and the spanning tree B before (after) applying any considered subcase, we need to have $C(A_2') + C(B_2) \leq C(A_1') + C(B_1)$ but also $C(A_2') + 2C(B_2) \leq C(A_1') + 2C(B_1)$ as at the end of phase 1 all edges of tree B are duplicated.

Now, with respect to cases 1a, 1b and 2a, as the triangular inequality holds and no exchange is applied, we have $C(A_2') \leq C(A_1')$ and $C(B_2) = C(B_1)$. Hence, $C(A_2') + C(B_2) \leq C(A_1') + C(B_1)$ and $C(A_2') + 2C(B_2) \leq C(A_1') + 2C(B_1)$.

For case 2b, if no exchange is necessary, then the same analysis of cases 1a, 1b and 2a holds. Besides, if an exchange occurs, let assume w.l.o.g. that edge e_{23} substitutes e_{12} and e_{13} in A' and that e_{13} substitutes e_{23} in B. Then, $C(A_2') = C(A_1') + c_{23} - c_{12} - c_{13}$ and $C(B_2) = C(B_1) + c_{13} - c_{23}$. Hence: $C(A_2') + C(B_2) = C(A_1') + C(B_1) - c_{12} \leq C(A_1') + C(B_1)$. Also: $C(A_2') + 2C(B_2) = C(A_1') + 2C(B_1) + c_{13} - c_{12} - c_{23} \leq C(A_1') + 2C(B_1)$.

Algorithm 2 The $\frac{2(n-1)}{n}$-approximation ratio

Compute two edge-disjoint minimum cost spanning trees A and B;
Make A Eulerian **and** $Stop = FALSE$;
PHASE 1
repeat
  Select the node j with largest degree d_j
  if $d_j \geq 6$ and j has at least 2 double edges **then** apply case 1a
  else if $d_j \geq 6$ and j has at least 4 single edges **then** apply case 1b
  else if $d_j = 4$ and j has 4 single edges **then** apply case 2a
  else if $d_j = 4$ and j has 1 double edge and 2 single edges **then** apply case 2b
  else if $d_j = 4$ and j has 2 double edges **then** apply case 2c
  else the cycle A is Hamiltonian and $Stop = TRUE$
until The condition Stop=FALSE
END OF PHASE 1
Make B Eulerian **and** $Stop = FALSE$;
PHASE 2
repeat
  Select the node j with largest degree d_j
  if $d_j \geq 6$ and j has at least 2 double edges **then** apply case 3a
  else if $d_j \geq 6$ and j has at least 4 single edges **then** apply case 3b
  else if $d_j = 4$ and j has 4 single edges **then** apply case 4a
  else if $d_j = 4$ and j has 1 double edge and 2 single edges **then** apply case 4b
  else if $d_j = 4$ and j has 2 double edges **then** apply case 4c
  else the cycle B is Hamiltonian and $Stop = TRUE$
until The condition Stop=FALSE
END OF PHASE 2
return two edge disjoint Hamiltonian cycles

For case 2c, if no exchange is necessary, then the same analysis of cases 1a, 1b and 2a holds. Besides, if an exchange occurs, then $C(A_2') = C(A_1') + c_{1n} - \sum_{k=1}^{n-1} c_{k,k+1}$ and $C(B_2) = C(B_1) + c_{ij} - c_{1n}$. Hence, due to the triangular inequality, $\forall i, j$:

$$\begin{aligned} C(A_2') + C(B_2) &= C(A_1') + C(B_1) - \sum_{k=1}^{n-1} c_{k,k+1} + c_{ij} \\ &\leq C(A_1') + C(B_1) - \sum_{k=i}^{j-1} c_{k,k+1} + c_{ij} \\ &\leq C(A_1') + C(B_1) \end{aligned}$$

Also, always due to the triangular inequality:

$$\begin{aligned} C(A_2') + 2C(B_2) &= C(A_1') + 2C(B_1) - c_{1n} - \\ &\qquad \sum_{k=1}^{n-1} c_{k,k+1} + 2c_{ij} \\ &\leq C(A_1') + 2C(B_1) - c_{1n} - \sum_{k=1}^{i-1} c_{k,k+1} \\ &\qquad - \sum_{k=j}^{n-1} c_{k,k+1} - \sum_{k=i}^{j-1} c_{k,k+1} + 2c_{ij} \\ &\leq C(A_1') + 2C(B_1) - c_{1n} - \sum_{k=1}^{i-1} c_{k,k+1} \\ &\qquad - \sum_{k=j}^{n-1} c_{k,k+1} + c_{ij} \\ &\leq C(A_1') + 2C(B_1) - c_{ni} - c_{j,n} + c_{ij} \\ &\leq C(A_1') + 2C(B_1) \end{aligned}$$

As far as phase 2 is concerned, for cases 3a, 3b and 4a, as no exchange is necessary, then the same analysis of cases 1a, 1b and 2a holds.

For case 4b, if no exchange is necessary, then the same analysis of cases 1a, 1b and 2a holds. Besides, if an exchange occurs, let assume w.l.o.g. that edge e_{23} substitutes e_{12} and e_{13} in B' and that e_{13} and e_{25} substitute e_{23} and e_{15} in B. Then: $C(B_2') = C(B_1') + c_{23} - c_{12} - c_{13}$ and $C(A_2^*) = C(A_1^*) + c_{13} + c_{25} - c_{23} - c_{15}$. Hence: $C(A_2') + C(B_2) = C(A_1') + C(B_1) - c_{12} - c_{15} + c_{25} \leq C(A_1') + C(B_1)$.

Finally, for case 4c, no exchange is necessary and for all three Hamiltonian cycles it is immediate to show by means of the triangular inequality that $C(B_2') \leq C(B_1')$. We prove it for cycle H_A, but a similar analysis holds for H_B and H_C. We have:

$$\begin{aligned} C(B_2') &= C(H_A) \\ &= C(1/2/n/n-1/\ldots/6/5/4/3/1) \\ &\leq C(1/2/3/\ldots/n/n-1/n-2/\ldots/3/2/1) \\ &= 2C(1/2/3/\ldots/n-2/n-1/n) \\ &= C(B_1'). \end{aligned}$$

For the computational complexity, we note that computing two edge-disjoint minimum cost spanning trees requires polynomial time. Also, the edges duplications to obtain the Eulerian circuits require linear time. Finally, the selection of the largest degree node and the application of any of the cases 1a...4c require at most linear time and can be applied at most $O(n)$ times. ∎

2.3. *Towards a $\frac{15}{8}$-approximation algorithm*

Let the $HC - ST$ problem denote the problem of finding in a given graph G a Hamiltonian circuit and a spanning tree edge disjoint inducing minimum total cost. Here we derive a $\frac{15}{8}$-approximation algorithm for the $2 - PSP$, provided that a $\frac{5}{4}$ approximation ratio is available for the related $HC - ST$ problem.

Consider the following algorithm.

Algorithm 3

---

a) Solve the $HC - ST$ problem computing correspondingly a Hamiltonian cycle A^* and a disjoint spanning tree B;
b) Apply to A^* step b) of Algorithm 1 computing a second tour B^* disjoint from A^* so that $A^* \cup B^*$ is a feasible solution of $2 - PSP$.
c) Apply to A^* and B phase b) of Algorithm 2 inducing two disjoint tours A' and B'. Let $A' \cup B'$ be the second feasible solution of $2 - PSP$.
return The minimum cost solution between $A^* \cup B^*$ and $A' \cup B'$.

Proposition 4 *If the $HC - ST$ problem is $5/4$-approximable, then Algorithm 3 induces a $\frac{15}{8}$-approximation ratio for 2-PSP.*

Proof. As a Hamiltonian path is also a tree, the optimal solution of the $HC - ST$ problem constitutes a lower bound on the optimal solution of $2 - PSP$. Hence, if the $HC - ST$ problem is $5/4$-approximable, then $C(A^* \cup B) \leq \frac{5}{4}C(OPT)$. Then either $C(A^*) \leq C(B)$ or $C(A^*) > C(B)$. If $C(A^*) \leq C(B)$, then $C(A^*) \leq \frac{5}{8}C(OPT)$ and, correspondingly, $C(A^* \cup B^*) \leq 3C(A^*) \leq \frac{15}{8}C(OPT)$. Besides, if $C(A^*) > C(B)$, then $C(B) \leq \frac{5}{8}C(OPT)$. But then, $C(A' \cup B') \leq C(A^*) + 2C(B) \leq \frac{5}{4}C(OPT) + \frac{5}{8}C(OPT) = \frac{15}{8}C(OPT)$. ■

Notice that, concerning the $HC - ST$ problem, by applying Phase 1 of algorithm 2, a straightforward $\frac{3n-1}{2n}$ approximation ratio holds. Should one improve this result down to $5/4$, then a $15/8$ approximation ratio would hold for $2 - PSP$ by means of Proposition 4.

Received 9-4-2009; revised 12-10-2009; accepted 25-11-2009

References

[1] A.A. Ageev and A.V. Pyatkin. A 2-Approximation Algorithm for the Metric 2-Peripatetic Salesman Problem. WAOA 2007 Conference Proceedings, LNCS 4927:103–115, 2008.

[2] E. Duchenne, G. Laporte and F. Semet. Branch and cut algorithms for the undirected m-peripatetic salesman problem. *European Journal of Operational Research*, 162:700–712, 2005.

[3] E. Duchenne, G. Laporte and F. Semet. The Undirected m-Peripatetic Salesman Problem: Polyhedral Results and New Algorithms. *Operations Research*, 55:949–965, 2007.

[4] J. Krarup. The peripatetic salesman and some related unsolved problems. in: Roy b (ed.) *Combinatorial Programming Methods and Applications*, 31:173 – 178, 1975.

[5] R. Wolfler Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computers and Operations Research*, 30:1269–1287, 2003.

[6] M. J. D. De Brey and A. Volgenant, Well-solved cases of the 2-peripatetic salesman problem, *Optimization*, 39 (3): 275-293, 1997.

[7] J.B.J.M. De Kort and A. Volgenant. On the generalized peripatetic salesman problem. *European Journal of Operational Research*, 73:175–180, 1994.

[8] J.B.J.M. De Kort. Lower bounds for symmetric k-peripatetic salesman problem. *Optimization*, 22(1):113–122, 1991.

[9] J.B.J.M. De Kort. Upper bounds for symmetric 2-peripatetic salesman problem. *Optimization*, 23(4):357–367, 1992.

[10] J.B.J.M. De Kort. A branch and bound algorithm for the symmetric 2-peripatetic salesman problem. *European Journal of Operational Research*, 70:229–243, 1993.

[11] N. Christofides. Worst-case analysis of a new heuristic for the Traveling Salesman Problem. Technical report, GSIA, Carnegie-Mellon University, Pittsburgh, 1976.

[12] A. Y. Baburin, E. Kh. Gimadi and N. M. Korkishko, Algorithms with performance guarantees for a metric problem of finding two edge-disjont Hamiltonian circuits of minimum total weight, Operations Research Proceedings. 2004. Berlin: Springer, 316-323, 2004.

[13] J. Roskind and R. Tarjan. A note on finding minimum-cost edge-disjoint spanning trees. *Mathematics of Operations Research*, 10: 701-708, 1985.