Approximation algorithms for the general and the asymmetric Stacker-Crane problems
Righini, G.; Trubian, M.

Published: 01/01/1994

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Approximation algorithms for the General and the Asymmetric Stacker-Crane Problems

Giovanni Righini
Marco Trubian

Eindhoven, June 1994
The Netherlands
Approximation algorithms for the General and the Asymmetric Stacker-Crane Problems *

Giovanni Righini
Technische Universiteit Eindhoven, Faculteit Wiskunde en Informatica
5600 MB Eindhoven, Nederland

Marco Trubian
Politecnico di Milano, Dipartimento di Elettronica e Informazione
piazza Leonardo da Vinci 32, 20133 Milano, Italia

20 May 1994

Abstract

The Stacker-Crane Problem (SCP) consists of finding the minimum length hamiltonian cycle on a mixed graph with oriented arcs and unoriented edges: feasible solutions must traverse all the arcs. Approximation algorithms are known for the case in which the triangle inequality holds. We consider the case in which the triangle inequality is violated (General SCP) and the case in which the problem is formulated on a complete digraph (Asymmetric SCP). We show how data-dependent worst-case bounds for the GSCP and the ASCP can be obtained by known approximation algorithms for the SCP with triangle inequality and by a new one. We also discuss the relationship with the Asymmetric Traveling Salesman Problem (ATSP) and we derive sufficient (and meaningful) conditions for ATSP instances to be solvable within constant worst-case bounds equal to 3, \( \frac{15}{7} \) and \( \frac{9}{5} \).

*The first author was supported by the Human Capital and Mobility Program of the European Community. The second author was partially supported by research contracts MURST 40% and 60% of the Italian Ministry of University and Scientific Research.
1 Introduction

The Stacker-Crane Problem (SCP) was defined by Frederickson, Hecht and Kim [4], who described two algorithms whose combination gives constant worst-case bound. We analyze two different generalizations, namely the General SCP and the Asymmetric SCP and we show that the same algorithms provide data-dependent worst-case bounds for both. The ATSP is one of the most widely studied NP-hard combinatorial optimization problems: in [3] and [8] approximation algorithms are given that provide data-dependent worst-case bounds; we use one of the algorithms described in [8] to obtain data-dependent worst-case bounds for the GSCP and the ASCP. Finally we study under which conditions an instance of the ATSP can be solved within constant worst-case bound by the algorithms of [4] and [8].

1.1 Problems

Stacker-Crane Problem

Instance: a complete weighted mixed graph $G(V, E, A)$, where edges in $E$ are unoriented and arcs in $A$ are oriented, satisfying the following two conditions:

$\alpha$) all nodes in $V$ are endpoint of exactly one arc;
$\beta$) all the edges satisfy the triangle inequality.

Question: find a minimum cost cycle on $G$, traversing all the arcs.

Remark 1. Since the TI holds, there exist an optimal cycle that is hamiltonian.

Proof. Any cycle traversing some arc $i$ more than once can be made hamiltonian without worsening its value by the following substitution: indicate by $h_i$ and $t_i$ the two endpoints (head and tail) of arc $i$; consider one of the edges incident to $h_i$ and call $u$ its endpoint different from $h_i$; consider one of the edges incident to $t_i$ and not included in the oriented path from $u$ to $t_i$ and call $v$ its endpoint different from $t_i$. Delete edges $[h_i, u]$ and $[t_i, v]$ and add edge $[u, v]$ (fig.1). Any cycle including pairs of adjacent edges can be made hamiltonian without worsening its value by the substitution of the two adjacent edges with a single edge between their non-common endpoints. Every cycle traversing the arcs only once and without pairs of adjacent edges is hamiltonian.

Remark 2. Since in every hamiltonian solution each arc is traversed exactly once, the sum of the costs of the arcs is a fixed cost. Therefore the value of every solution can be defined in two different ways, depending on taking into account total costs or only variable costs (defined as the difference between total costs and fixed costs). The choice does not affect the ranking of the solutions, but it affects the value of the worst-case bounds.
Asymmetric Traveling Salesman Problem

Instance: a complete weighted digraph $G(V, A)$.

Question: find a minimum cost hamiltonian cycle on $G$.

**Notation.** In this paragraph we establish the notation we use throughout the paper. $N$ is the number of arcs of a SCP: we assume arcs to be arbitrarily numbered from 1 to $N$. $l_i$ is the cost of the $i$-th arc and $L = \sum_{i=1}^{N} l_i$ is the fixed cost. We call tail and head the endpoints of every arc, such that the orientation is from the tail to the head. We indicate the tail and the head of the $i$-th arc with $t_i$ and $h_i$.

$EC$ and $HC$ stand for eulerian spanning cycle and hamiltonian cycle.

Let $S$ be any set of arcs: then $S' \Rightarrow \{S \text{ traversed in the opposite direction}\}$ or “reversed”; $S^*$ is the optimal value of $S$; $S_t$ and $S_v$ indicate the total and the variable costs of arc set $S$. Since we define auxiliary graphs and we number them $G_1, G_2, ..., G_k$, we indicate with $S_k$ a set $S$ of arcs on graph $G_k$.

We use the name of a set of arcs or edges to represent also its associated cost, since this introduces no ambiguity and enhances readability and intuition.

TI stands for the triangle inequality and ATI for the asymmetric triangle inequality.

TI is defined as follows: for each pair of vertices $i$ and $j$, the cost of edge $[i, j]$ is not greater than the cost of any (oriented or unoriented) path from $i$ to $j$ or from $j$ to $i$ (fig.2).

The ATI is defined on triplets of arcs forming non-coherent triangles; a non-coherent triangle is a triangle that is not a cycle (fig.3). Indicating with $d_{uv}$ the cost of an arc $(u, v)$ and with $V$ the ground set of a graph, the ATI is:

$$d_{ij} \leq d_{ik} + d_{kj} \quad \forall i, j, k \in V$$
Complexity. The SCP is NP-hard; this is proved by showing that every instance of the TSP is equivalent to an instance of a SCP with arcs of zero cost (see [4]). The following polynomial time procedure performs the transformation.

**Procedure Transform-TSP-into-SCP** (fig. 4).

*Input:* a TSP instance, defined by a graph $G(V_G, E_G)$ with weights $d$ on edges.

*Output:* an SCP instance, defined by a mixed graph $H(V_H, E_H, A_H)$, such that the optimal solution and its value are preserved.

*Step 1:* for each node $i \in V_G$, define an arc in $A_H$, with zero cost. Define $V_H$ to be the set of the $2N$ endpoints of the $N$ arcs.

*Step 2:* for each edge $[i, j] \in E_G$, define 4 edges $[h_i, t_j], [h_j, t_i], [h_i, h_j]$ and $[t_i, t_j]$ in $E_H$, and give them a weight equal to $d_{ij}$.

The ATSP is also NP-hard [5]. It is possible to show that every instance of the SCP with $N$ arcs and $2N$ nodes is equivalent to an instance of the ATSP with $N$ nodes: the
following polynomial time procedure performs the transformation.

**Procedure Transform-SCP-into-ATSP** (fig. 5).

*Input*: an SCP instance, defined by a complete mixed graph $G(V_G, E_G, A_G)$ with weights $l_i$ on each arc $i$ and weights $d_{uv}$ on each edge $[u,v]$.

*Output*: an ATSP instance, defined by a complete digraph $H(V_H, A_H)$, such that the optimal solution and its value are preserved.

*Step 1*: for each arc of $A_G$, define a node in $V_H$.

*Step 2*: for each head-to-tail edge $[h_i, t_j] \in E_G$ define an arc $(i,j) \in A_H$ with weight equal to $d_{h_it_j} + \frac{1}{2}(l_i + l_j)$.
Existing algorithms. Frederikson, Hecht and Kim [4] give 2 algorithms, SmallArcs (SA) and LargeArcs (LA), whose combination gives a worst-case bound equal to $9/5$ when total costs are considered. When only variable costs are considered the best constant worst-case bound is 3 and is given by algorithm LA alone. This observation was not made in [4].

Frieze, Galbiati and Maffioli [3] examine many algorithms for the ATSP with ATI, producing worst-case bounds depending on the number of vertices or on other parameters of the instance.

Righini and Trubian [8] present two algorithms with data-dependent worst-case bounds for the ATSP. One of them (RT2) improves the worst-case bound of one of the algorithms in [3]. The description of algorithms LA, SA and RT2 is recalled in the appendix.

Purpose of the research and paper outline. The purpose of this paper is:
1) to obtain data-dependent bounds for the SCP problem when the triangular inequality does not hold; we call this problem the General Stacker Crane Problem, GSCP (section 2);
2) to obtain data-dependent bounds for the SCP when $G$ is a digraph: we call this problem the Asymmetric Stacker Crane Problem, ASCP (section 3);
3) to obtain sufficient conditions for an ATSP instance to be solvable within constant worst-case bound; the way to obtain such result is to formulate the ATSP instance as a SCP instance for which constant worst-case bounds equal to $3, 15/7$ and $9/5$ are provided by algorithms LA, LA+RT2 and LA+SA (section 4).

Since both algorithms SA and RT2 use the algorithm of Christofides as a subroutine, we first study how the $3/2$ bound provided by it changes when the TI does not hold.
1.2 The algorithm of Christofides when TI is violated

Consider a complete symmetric weighted graph \( H(V_H, E_H) \) with edge weights \( w_{ij} \) (\([i,j] \in E_H\) ). Define \( \Delta_H = \max_{i,j,k \in V_H} \{ w_{ij} - (w_{ik} + w_{jk}), 0 \} \), that is the maximum violation of the TI.

**Theorem 1.1** The algorithm of Christofides produces an eulerian spanning cycle \( EC \) and a hamiltonian cycle \( HC \), such that:

\[
EC \leq \frac{3}{2} HC^* + \frac{e}{2} \Delta_H
\]

\[
HC \leq \frac{3}{2} HC^* + \frac{N}{2} \Delta_H
\]

where \( N = \|V_H\|, 0 \leq e \leq N - 2 \) is the number of even degree nodes of the minimum spanning tree of \( H \) and \( HC^* \) is the minimum hamiltonian cycle on \( H \).

**Proof.** The algorithm initially builds the minimum spanning tree on graph \( H \). Call it \( ST^* \) and let \( O \) and \( E \) be the set of nodes of odd and even degree of \( ST^* \) and \( o \) and \( e \) their cardinalities. Since \( ST^* \) has \( N - 1 \) edges and no cycles, its cost is a lower bound to \( HC^* \), that is \( ST^* \leq HC^* \).

In a second step the algorithm finds the minimum cost matching \( M_0 \) on the nodes of odd degree. Since every hamiltonian cycle is the union of two edge-disjoint matchings, \( M_0 \leq \frac{1}{2} HC_0 \), where \( HC_0 \) is the minimum hamiltonian cycle on the nodes of \( O \).
It is possible to obtain a particular hamiltonian cycle \( HC_0 \) on the nodes of \( O \) from \( HC* \) by the following procedure: for every node \( u \) in \( E \), delete from \( HC* \) the two edges \([u, r]\) and \([u, s]\) connecting \( u \) to its adjacent nodes \( r \) and \( s \) and replace them by edge \([r, s]\) (fig.6). Perform the substitution sequentially for \( \varepsilon \) times. Every time the substitution is performed two edges of a triangle are replaced with the third edge; therefore each substitution costs at most \( \Delta_H \) and then \( HC_0 \leq HC* + \varepsilon \Delta_H \). Since by definition \( HC_0 \leq HC_0 \),

\[
EC = ST* + M_0 \leq HC* + \frac{1}{2} HC_0 \leq HC* + \frac{1}{2}(HC* + \varepsilon \Delta_H) = \frac{3}{2} HC* + \frac{\varepsilon}{2} \Delta_H
\]

For obtaining a hamiltonian cycle \( HC \) from \( EC \) the algorithm replaces pairs of adjacent edges by single edges, until every node is visited only once. Since \( ST* \) has \( N - 1 \) edges and \( M_0 \) has \( \frac{\varepsilon}{2} \) edges, the number of substitutions is \((N - 1) + \frac{\varepsilon}{2} - N\), that is \( \frac{\varepsilon}{2} - 1\). Each substitution costs at most \( \Delta_H \) and therefore

\[
HC \leq EC + \frac{\varepsilon}{2} \Delta_H \leq \frac{3}{2} HC* + (\frac{\varepsilon}{2} + \frac{\varepsilon}{2}) \Delta_H = \frac{3}{2} HC* + \frac{N}{2} \Delta_H
\]
2 The General Stacker Crane Problem

The question in the GSCP is to find a hamiltonian cycle on $G$ when the TI does not hold; the interest in this problem comes from the observation that the constant bounds provided by algorithms LA and SA of [4] are valid for eulerian cycles only, if the TI is violated. In particular both LA and SA can output a cycle traversing all arcs once but visiting some node more than once (fig.7a) as well as a cycle traversing some arc more than once (fig.7b).

General Stacker Crane Problem: GSCP

*Instance:* a (complete) weighted mixed graph $G(V, E, A)$, where edges in $E$ are unoriented and arcs in $A$ are oriented.

*Question:* find a minimum cost hamiltonian cycle $HC^*$, traversing all arcs.

Conditions $\alpha$ and $\beta$ of the formulation of the SCP can be violated. In particular we show that it is possible to enforce condition $\alpha$, but not condition $\beta$ in general. Therefore the preprocessing procedure used in [4] to enforce the triangle inequality cannot be executed since it makes loosing the hamiltonicity of solutions. First we reformulate the GSCP in an equivalent way and we give polynomial time preprocessing procedures to do the transformation. Then we apply algorithms LA, SA and RT2 in order to obtain data-dependent worst-case bounds for the GSCP.

If condition $\alpha$ is violated, apply the following preprocessing procedures, equal to those presented in [4].

**Procedure Preprocessing 1a.** (fig.8)

*Input:* an instance of the GSCP defined on a weighted mixed graph $G(V, E, A)$.

*Output:* an instance of the GSCP defined on a weighted mixed graph $H(V_H, E_H, A_H)$ such that all its nodes are endpoints of at least one of its arcs.

For each node $u \in V$ that is endpoint of no arc of $A$: split it into two nodes $u_1$ and $u_2$. 

![Figure 7](image-url)
in \( V_H \); split each edge \([u, v] \in E \) incident to the original node \( u \) into two edges \([u_1, v]\) and \([u_2, v]\) in \( E_H \) with the same weight of \([u, v]\); connect \( u_1 \) and \( u_2 \) with an arc \((u_1, u_2)\) of zero cost.

![Figure 8](image)

**Remark.** If some node is entered (or left) by more than one arc, no feasible solution can exist. Anyway in this case one could ask for a cycle traversing all arcs once and visiting each node the smallest possible number of times. This problem is again equivalent to a GSCP on a graph \( G_1 \) obtained by the following preprocessing procedure.

**Procedure Preprocessing 1b.** (fig.9)

*Input:* a GSCP instance defined on a weighted mixed graph \( H(V_H, E_H, A_H) \), such that all its nodes are endpoints of at least one of its arcs.

*Output:* a GSCP instance defined on a weighted mixed graph \( G_1(V_1, E_1, A_1) \), such that all its nodes are endpoints of one of its arcs.

For each node \( u \in V_H \) that is head of \( k_1 \) arcs of \( A_H \) and tail of \( k_2 \) arcs of \( A_H \), with \( k_1 + k_2 > 1 \), split \( u \) into a complete bipartite subgraph with \( k_1 \) nodes on one side, such that each of them is the head of one of the arcs entering \( u \), and \( k_2 \) nodes on the other side, such that each of them is the tail of one of the arcs leaving \( u \); give zero cost to all the edges of the bipartite subgraph; replace each edge \([u, v] \in E_H \) by \( k_1 + k_2 \) copies of it with the same weight of \([u, v]\).

![Figure 9](image)
The preprocessing procedures affect neither the hamiltonicity nor the cost of the feasible solutions. Therefore the GSCP on $G_1$ is equivalent to that on $G$. Condition $\alpha$ is satisfied on graph $G_1$.

Since the orientation of arcs is known and a hamiltonian cycle is required, the orientation in which the edges must be traversed in every feasible solution can be derived. Moreover all tail-to-tail and head-to-head edges can never belong to any feasible solution and then they can be discarded. Consider the following polynomial time preprocessing procedure, which transforms an instance of the GSCP on graph $G_1$ into an equivalent instance of the ATSP on graph $G_2$.

**Preprocessing 2. (fig.10)**

*Input:* a weighted mixed graph $G_1(V_1, E_1, A_1)$, such that all its nodes are endpoints of one of its arcs.

*Output:* a weighted digraph $G_2(V_2, A_2)$.

Replace head-to-tail edges $[h_i, t_j] \in E_1$ with arcs $(h_i, t_j)$ with the same weight; delete all other edges; shrink each arc $i \in A_1$ merging its endpoints into a single node and associate to the node a weight equal to $l_i$.

![Figure 10](image)

After the execution of Preprocessing 2, the weights on the nodes of $G_2$ correspond to the fixed costs of the GSCP, whereas the weights on the arcs of $G_2$ correspond to the variable costs of the GSCP. All feasible solutions of the ATSP on $G_2$ correspond to feasible solutions of the GSCP on $G_1$ and vice versa. Therefore $HC_{2*} = HC_v^*$.

Let us indicate by $d_{ij}$ the weight of arc $(i, j) \in A_2$. Define the maximum violation of the ATI on $G_2$, $\Delta_2 = \max_{i, j, k \in V_2} \{d_{ij} - (d_{ik} + d_{kj}), 0\}$; define also the maximum asymmetry of graph $G_2$, $\delta_2 = \max_{i, j \in V_2} \{d_{ij} - d_{ji}\}$. 
2.1 Algorithm LA for the GSCP

**Input:** a complete digraph $G_2(V_2, A_2)$ with $N$ nodes and with weights $d_{ij}$ for all $(i, j) \in A_2$ and $l_i$ for all nodes $v_i \in V_2$.

**Output:** a hamiltonian cycle on $G_2$.

**Step 1:** find the minimum weighted matching $M_v^*$ on an auxiliary complete bipartite graph $K(L, R, E_K)$, where $L = V_2$, $R = V_2$ and the following edge weights $w_{ij}$ for all $(i, j) \in L$, $j \in R$: if $i \neq j$ then $w_{ij} = d_{ij}$; if $i = j$ then $w_{ij} = M$, where $M$ is sufficiently large to forbid edges of weight $M$ to belong to $M_v^*$. $M_v^*$ can be computed in time polynomial in $N$ by a linear assignment algorithm. The matching provides a lower bound on the cost of the minimum hamiltonian cycle $HC_2^*$.

(i) $M_v^* \leq HC_2^*$

The arcs of $M_v^*$ form a number $s$ of subtours on $G_2$; if self-loops are not allowed (as in our version of the algorithm) $s \leq \frac{N}{2}$ (fig.11).

**Step 2:** find the minimum cost arc set $SD_v^*$ connecting the subtours of $M_v^*$ (fig.12). Since in every feasible solution all subsets of nodes must be connected,

(ii) $SD_v^* \leq HC_2^*$

**Step 3:** consider the reverse arc of each arc of $SD_v^*$ and define $S_v = M_v^* + SD_v^* + SD_v^*$. $S_v$ is a eulerian spanning cycle, but not hamiltonian in general (fig.13). For the definition of $\delta_2$,

(iii) $S_v \leq M_v^* + SD_v^* + (SD_v^* + (s - 1)\delta_2)$

**Step 4:** for every arc $(i, j)$ of $SD_v^*$, call $l$ one of the successors of $i$ in $S_v$ and $k$ one of the predecessors of $j$ in $S_v$ (fig.14). Delete from $S_v$ arc $(i, l)$, arc $(k, j)$ and arc $(j, i)$ and add arc $(k, l)$. The resulting arc set is a hamiltonian cycle, $HC_2^*$. The substitution must be done $s - 1$ times and every substitution costs at most $2\Delta_2$, since by the definition of $\Delta_2$, $d_{kl} \leq d_{kj} + d_{jl} + \Delta_2$ and $d_{ij} \leq d_{ji} + d_{li} + \Delta_2$. Therefore:

(iv) $HC_2^* \leq S_v + (s - 1)2\Delta_2$.

From inequalities (i), (ii), (iii) and (iv) and from the relation $s \leq \frac{N}{2}$, it follows

$$HC_2^* \leq 3HC_2^* + N\Delta_2 + \frac{1}{2}N\delta_2$$

and when total costs are considered

$$HC_2^* \leq 3HC_2^* - 2L + N\Delta_2 + \frac{1}{2}N\delta_2$$
2.2 Algorithm SA for the GSCP

Input: a complete digraph \( G_2(V_2, A_2) \) with \( N \) nodes and with weights \( d_{ij} \) for all \( (i, j) \in A_2 \) and \( l_i \) for all nodes \( v_i \in V_2 \).

Output: a hamiltonian cycle on \( G_2 \).

Step 1: define a complete unoriented graph \( G_3(V_3, E_3) \), with \( V_3 = V_2 \) and edge weights \( w_{ij} = \min\{d_{ij}, d_{ji}\} \forall (i, j) \in E_3 \). For the construction of \( G_3 \), it follows

(i) \( HC_3^* \leq HC_2^* \).

Define \( \Delta_3 = \max_{i,j,k \in V} \{w_{ij} - (w_{ik} + w_{jk}), 0\} \), that is the maximum violation of the TI on \( G_3 \).

Observation 2.1: \( \Delta_3 \leq \Delta_2 + \delta_2 \).

Proof. Consider any triple of nodes \( (i, j, k) \) and the three arcs of \( A_2 \) forming the minimum cost triangle with vertices \( i, j \) and \( k \). Two cases can happen: either the three arcs form a coherent triangle (case 1) or they form a non-coherent triangle (case 2).

Case 1 (fig.15): suppose wlog that the arc between \( i \) and \( j \) is directed from \( i \) to \( j \); the following inequalities hold:

\[
w_{ij} = d_{ij} \leq d_{ji} + \delta_2 \leq (d_{jk} + d_{ki} + \Delta_2) + \delta_2 = w_{jk} + w_{ik} + \Delta_2 + \delta_2.
\]

Case 2 (fig.16): suppose wlog that the arc between \( i \) and \( j \) is directed from \( i \) to \( j \). If the non-coherent arc is \( (i, j) \) then apply the definition of \( \Delta_2 \) (fig.16a); otherwise suppose wlog that the non coherent arc is arc \( (i, k) \) (fig.16b); the following inequalities hold:

\[
w_{ij} = d_{ij} \leq d_{ik} + d_{kj} + \Delta_2 \leq d_{ik} + (d_{jk} + \delta_2) + \Delta_2 = w_{ik} + w_{jk} + \delta_2 + \Delta_2.
\]

Remark. In [4], the construction of the auxiliary undirected graph is followed by the substitution of the edges weights by the cost of the shortest path between their endpoints. This is no longer possible, under our hypotheses, because it would cause the algorithm to output eulerian instead of hamiltonian cycles. Therefore in this case we cannot enforce the TI on graph \( G_3 \).

Step 2: apply the algorithm of Christofides to graph \( G_3 \), obtaining a eulerian spanning cycle \( EC_3 \). By theorem 1.1 and observation 2.1 it follows

(ii) \( EC_3 \leq \frac{3}{2} HC_3^* + \frac{1}{2} e \Delta_3 \),

where \( e \) is the number of nodes of even degree on the minimum spanning tree of \( G_3 \).

Step 3: consider on \( G_2 \) the arc set \( S_v \) made of the pairs of arcs \( ((i, j), (j, i)) \) correspondent to the edges of \( EC_3 \) (fig.17). By definition of \( \delta_2 \) and by the construction of \( G_3 \) it follows that \( S_v \leq 2EC_3 + q\delta_2 \), where \( q \) is the number of edges of \( EC_3 \). \( S_v \) can be traversed in two different and arc-disjoint ways; both of them are eulerian spanning cycles of \( G_2 \). Choose the best of the two and call it \( EC_{2*} \). Then

(iii) \( EC_{2*} \leq \frac{1}{2} S_v \leq EC_3 + \frac{1}{2} q \delta_2 \).
Step 4: in order to obtain a hamiltonian cycle $HC_2$, apply $q - N$ times the substitution of two consecutive arcs of $EC_2$ with one arc with the same endpoints. Since each substitution costs at most $\Delta_2$,

(iv) $HC_2 \leq EC_2 + \Delta_2(q - N)$.

Figure 15

Figure 16

Figure 17
From inequalities (i), (ii), (iii) and (iv) and from the relations $q = (N - 1) + \frac{e}{2}$ and $a = N - e$ it follows

$$HC_{2*} \leq \frac{3}{2} HC_{2*}^* + \frac{1}{2} N \Delta_2 + \frac{3}{2} N \delta_2$$

and when total costs are considered

$$HC_{2*} \leq \frac{3}{2} HC_{2*}^* + \frac{1}{2} L + \frac{1}{2} N \Delta_2 + \frac{3}{2} N \delta_2$$
2.3 Algorithm RT2 for the GSCP

Input: a complete digraph $G_2(V_2, A_2)$ with $N$ nodes and with weights $d_{ij}$ for all $(i, j) \in A_2$ and $l_i$ for all nodes $v_i \in V_2$.

Output: a hamiltonian cycle on $G_2$.

Step 1: define a complete unoriented graph $G_3(V_3, E_3)$, with $V_3 = V_2$ and edge weights $w_{ij} = d_{ij} + d_{ji} \forall [i, j] \in E_3$.

For the construction of $G_3$, it follows that $HC_3^* = (HC_{2*} + HC_{2*})^* \leq HC_{2*} + HC_{2*}$.

By definition of $\delta_2$, $HC_{2*}^* \leq HC_{2*}^* + N\delta_2$. Therefore

(i) $HC_3^* \leq 2HC_{2*}^* + N\delta_2$.

Define $\Delta_3 = \max_{i,j,k \in V_3} \{w_{ij} - (w_{ik} + w_{jk}).0\}$, that is the maximum violation of the TI on $G_3$.

Observation 2.2 $\Delta_3 \leq 2\Delta_2$.

Proof. Applying the ATI on the two triangles of $G_2$ defined by the same three vertices $i, j$ and $k$, $d_{ij} \leq d_{ik} + d_{kj} + \Delta_2$ and $d_{ji} \leq d_{ki} + d_{jk} + \Delta_2$; summing up the two inequalities, $w_{ij} \leq w_{ik} + w_{jk} + 2\Delta_2$ and then $\Delta_3 \leq 2\Delta_2$.

Step 2: apply the algorithm of Christofides, which provides a hamiltonian cycle $HC_3$.

For theorem 1.1 and observation 2.2,

(ii) $HC_3 \leq \frac{3}{2}HC_3^* + \frac{1}{2}N\Delta_3$.

Step 3: define $S_v$ to be the set of pairs of opposite arcs on $G_2$, corresponding to the edges of $HC_3$. $S_v$ can be traversed in two arc-disjoint ways (fig.18); call $HC_{2*}$ the best of the two.

(iii) $HC_{2*} \leq \frac{1}{2}S = \frac{1}{2}HC_3$.

From inequalities (i), (ii) and (iii) and observation 2.2 it follows

$$HC_{2*} \leq \frac{3}{2}HC_{2*}^* + \frac{3}{4}N\delta_2 + \frac{1}{2}N\Delta_2$$

and when total costs are considered

$$HC_{2*} \leq \frac{3}{2}HC_{2*}^* - \frac{1}{2}L + \frac{3}{4}N\delta_2 + \frac{1}{2}N\Delta_2$$
2.4 Comparison

The data-dependent worst-case bounds provided by algorithms LA, SA and RT2 are of the form:

$$\eta \leq k + k_L \frac{L}{HC_2} + k_{\Delta_2} \frac{N\Delta_2}{HC_2} + k_{\delta_2} \frac{N\delta_2}{HC_2}$$

Tables in figure 19 show the comparison between the three bounds. The worst-case bound of algorithm RT2 dominates the one of algorithm SA.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(k)</th>
<th>(k_L)</th>
<th>(k_{\Delta_2})</th>
<th>(k_{\delta_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1/2</td>
</tr>
<tr>
<td>SA</td>
<td>3/2</td>
<td>0</td>
<td>1/2</td>
<td>3/2</td>
</tr>
<tr>
<td>RT2</td>
<td>3/2</td>
<td>0</td>
<td>1/2</td>
<td>3/4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(k)</th>
<th>(k_L)</th>
<th>(k_{\Delta_2})</th>
<th>(k_{\delta_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>3</td>
<td>-2</td>
<td>1</td>
<td>1/2</td>
</tr>
<tr>
<td>SA</td>
<td>3/2</td>
<td>-1/2</td>
<td>1/2</td>
<td>3/2</td>
</tr>
<tr>
<td>RT2</td>
<td>3/2</td>
<td>-1/2</td>
<td>1/2</td>
<td>3/4</td>
</tr>
</tbody>
</table>

Figure 19
3 The Asymmetric Stacker Crane Problem

Instance: a complete digraph \( G(V, E, A) \) where \( A \) is a subset of \( E \).

Question: find a minimum cost cycle \( EC^* \), traversing all arcs in \( A \).

In order to distinguish arcs in \( A \), we call them fixed arcs and we call variable arcs the others.

A solution of the ASCP is neither required to be hamiltonian nor to traverse fixed arcs only once. If a hamiltonian cycle is required, the problem becomes equivalent to that of section 2.

Consider the following conditions:

\( \alpha' \) All nodes of the graph are endpoints of one of the fixed arcs.

\( \beta' \) Variable arcs satisfy the ATI.

Both conditions \( \alpha' \) and \( \beta' \) can be violated by a general instance of the ASCP. So, first we reformulate the ASCP in an equivalent way and we give polynomial time preprocessing procedures to do the transformation. Then we apply algorithms LA, SA and RT2 in order to obtain data-dependent worst-case bounds.

If condition \( \alpha' \) is violated, execute the following preprocessing procedures.

Procedure Preprocessing \( 1 \alpha' \). (fig.20)

Input: an instance of the ASCP defined on a complete weighted digraph \( G(V, E, A) \).

Output: an instance of the ASCP defined on a complete weighted digraph \( H(V_H, E_H, A_H) \), such that all nodes are endpoints of at least one of the fixed arcs.

For each node \( u \in V \) that is endpoint of no fixed arc, split \( u \) into two nodes \( u_1 \) and \( u_2 \); split each variable arc incident to the original node \( u \) into two variable arcs, incident to \( u_1 \) and to \( u_2 \), with the same weight; connect \( u_1 \) and \( u_2 \) with a fixed arc of zero cost.

![Figure 20](image-url)
Procedure Preprocessing 1b'. (fig.21)

*Input:* an instance of the ASCP defined on a complete weighted digraph \( H(V_H, E_H, A_H) \), such that all nodes are endpoints of at least one of the fixed arcs.

*Output:* an instance of the ASCP defined on a complete weighted digraph \( G_1(V_1, E_1, A_1) \), such that all nodes are endpoints of one of the fixed arcs.

For each node \( u \in V_H \) that is endpoint of \( k > 1 \) fixed arcs of \( A_H \): split \( u \) into \( k \) nodes \( u_1, \ldots, u_k \), such that each of them is the endpoint of one of the fixed arcs incident to \( u \); replace each variable arc incident to \( u \) by \( k \) copies of it, incident to \( u_1, u_2, \ldots, u_k \), with the same weight of the original variable arc; connect all nodes \( u_1, u_2, \ldots, u_k \) with variable arcs of zero cost.

![Figure 21](image)

Neither of the preprocessing procedures affects the variable or the total cost of the feasible solutions. Therefore the ASCP on \( G_1 \) is equivalent to that on \( G \).

A second preprocessing step enforces condition \( \beta' \), transforming the ASCP instance on \( G_1 \) into an equivalent ASCP instance on \( G_2 \).

Procedure Preprocessing 2'.

*Input:* an instance of the ASCP defined on a complete weighted digraph \( G_1(V_1, E_1, A_1) \), such that \( \alpha' \) holds.

*Output:* an instance of the ASCP defined on a complete weighted digraph \( G_2(V_2, E_2, A_2) \), where \( A_2 \) is the subset of fixed arcs and such that both conditions \( \alpha' \) and \( \beta' \) hold.

For every pair of nodes, compute the minimum cost oriented path in \( G_1 \). This can be done in polynomial time with well known algorithms [2]. Define a graph \( G_2 \) with the same arc sets as \( G_1 \) but with weight of each variable arc equal to the cost of the shortest oriented path between its endpoints.

The original ASCP instance is equivalent to the ASCP instance on \( G_2 \): suitable post-processing procedures that undo the replacements done by preprocessing procedures 1a', 1b' and 2' can transform any feasible solution on \( G_2 \) into a feasible solution on \( G \).
The value of all feasible solutions is the same on $G_2$ and on $G$. Moreover, an optimal solution on $G_2$ is hamiltonian, since any cycle visiting a node more than once can be made hamiltonian without worsening its value, exploiting condition $\beta'$. Therefore $HC_2^* = EC^*$.

Let $N$ be the number of fixed arcs and $2N$ the number of nodes of $G_2$. Let us define the following parameters:

\[ \delta^A_2 = \max_{(i,j) \in A} \{d_{ji} - d_{ij}\}. \]
\[ \delta^E_2 = \max_{i,j \in (i,j) \in \mathcal{A} \setminus (j,i) \in \mathcal{A}} \{d_{ij} - d_{ji}\}. \]
\[ \delta_2 = \max\{\delta^A_2, \delta^E_2\}. \]

Remark. $\delta_2$ cannot be computed by applying its definition to $G$ directly, before the preprocessing. Anyway the time needed to compute $\delta_2$ is not more than that needed to do the preprocessing.
3.1 Algorithm LA for the ASCP

Input: a complete weighted digraph $G_2(V_2, E_2, A_2)$, where $A_2$ is the subset of fixed arcs and such that conditions $\alpha'$ and $\beta'$ hold.
Output: a hamiltonian cycle traversing all fixed arcs.

Step 1: Consider the auxiliary bipartite graph $K(T, H, E_K)$ such that $T$ is the set of nodes of $G_2$ that are tails of fixed arcs, $H$ is the set of nodes of $G_2$ that are heads of fixed arcs and $E_K$ is the set of variable arcs of $G_2$. Find the optimal matching $M_\ast$ on graph $K$; it corresponds to a set $M_\ast$ of variable arcs of $G_2$. The arc set $M_\ast = M_\ast \cup A_2$ forms $s$ subtours on $G_2$ (fig.22) and its cost is a lower bound of the cost of the minimum hamiltonian cycle.

(i) $M_\ast \leq HC_\ast$
Since self-loops are forbidden, $s \leq N/2$.

Step 2: Choose the smallest variable arcs connecting the subtours: call $SD_\ast$ the set of such arcs (fig.23). Since in all feasible solutions the subtours must be connected,

(ii) $SD_\ast \leq HC_\ast$

Step 3: Consider the set $SD_\ast$ of arcs opposite to those of $SD_\ast$. By definition of $E_2$, $SD_\ast \leq SD_\ast + (s - 1)E_2$. Consider the eulerian cycle $EC_\ast = M_\ast + SD_\ast + SD_\ast$ (fig.24).

(iii) $EC_\ast \leq M_\ast + 2SD_\ast + (s - 1)E_2$

Step 4: Substitute all pairs of adjacent variable arcs by single variable arcs with the same endpoints until a hamiltonian cycle $HC_2$ is left. By condition $\beta'$,

(iv) $HC_2 \leq EC_\ast$

From inequalities (i), (ii), (iii) and (iv) it follows

$$HC_2 \leq 3HC_\ast + \frac{1}{2}N\delta_2^E$$

and when total costs are considered

$$HC_2 \leq 3HC_2 + 2L + \frac{1}{2}N\delta_2^E$$
3.2 Algorithm SA for the ASCP

Input: a complete weighted digraph $G_2(V_2, E_2, A_2)$, where $A_2$ is the subset of variable arcs and such that conditions $\alpha'$ and $\beta'$ hold.

Output: a hamiltonian cycle traversing all fixed arcs.

Let us indicate with $d_{uv}$ the cost of the generic variable arc $(u, v)$ of $G_2$.

Step 1: define a complete unoriented graph $G_3(V_3, E_3)$ with $N$ nodes, one for each fixed arc of $G_2$; define the following edge weights on $G_3$ (fig.25):

$$w_{ij} = \min\{d_{ht_ij}, d_{t_jh_i}, d_{h_iht_j}, d_{h_jh_i}, d_{t_iht_j}, d_{t_jh_i}\} + \frac{l_i + l_j}{2} \quad \forall i, j \in V_3.$$  

Define $\Delta_3 = \max_{i,j,k \in V_3}\{w_{ij} - (w_{ik} + w_{jk})\}$, the maximum violation of the TI on $G_3$.

Observation 3.1 $\Delta_3 \leq \delta_2$

Proof. Consider the variable arcs $(r, s)$ and $(u, v)$ of $G_2$ such that $w_{ik} = d_{rs} + \frac{l_i}{2} + \frac{l_k}{2}$ and $w_{jk} = d_{uv} + \frac{l_j}{2} + \frac{l_k}{2}$. The following cases can happen: both $(r, s)$ and $(u, v)$ enter (or both leave) the $k$-th fixed arc (case 1) or one of them enters and the other leaves the $k$-th fixed arc (case 2).

Case 1) without loss of generality, suppose both $(r, s)$ and $(u, v)$ enter fixed arc $k$ (fig.26). If $s = v$ (fig.26a), the following inequalities hold: $w_{ij} \leq d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + d_{uv} + \frac{l_i}{2} + \frac{l_j}{2} \leq w_{ik} + w_{jk} - l_k + \frac{\delta_2}{2} \leq w_{ik} + w_{jk} + \frac{\delta_2}{2}$; otherwise suppose wlog that $s = t_k$ and $v = h_k$ (fig.26b); then the following inequalities hold: $w_{ij} \leq d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + l_k + d_{uv} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + l_k + (d_{uv} + \frac{\delta_2}{2}) + \frac{l_i}{2} + \frac{l_j}{2} = w_{ik} + w_{jk} + \frac{\delta_2}{2}$.

Case 2) without loss of generality suppose $(r, s)$ enters fixed arc $k$ and $(u, v)$ leaves it. If $s = u$ (fig.27a), then $w_{ij} \leq d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + d_{uv} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + d_{uv} + l_k + \frac{l_i}{2} + \frac{l_j}{2} = w_{ik} + w_{jk}$. If $s \neq u$ and the fixed arc is oriented from $s$ to $u$, then $w_{ij} \leq d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + l_k + d_{uv} + \frac{l_i}{2} + \frac{l_j}{2} = w_{ik} + w_{jk}$; if the fixed arc is oriented from $u$ to $s$ (fig.27b), then $w_{ij} \leq d_{ru} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + d_{uv} + \frac{l_i}{2} + \frac{l_j}{2} \leq d_{rs} + (l_k + \frac{\delta_2}{2}) + d_{uv} + l_k + \frac{l_i}{2} + \frac{l_j}{2} = w_{ik} + w_{jk} + \frac{\delta_2}{2}$.

By the definition of $\delta_2$, the statement of the observation follows.

By the definition of the weights on $G_3$.

(i) $HC_3^* \leq HC_{2*}^* + L = HC_{21}^*$.

Step 2: apply the algorithm of Christofides, obtaining a eulerian spanning cycle $EC_3$; for theorem 1.1 it follows

(ii) $EC_3 \leq \frac{3}{2} HC_3^* + \frac{1}{2} e \delta_2$,

where $e$ is the number of even degree nodes of the minimum spanning tree of $G_3$.

Step 3: consider the arc set $S_v$ made of the variable arcs of $G_2$ correspondent to the edges in $EC_3$ (fig.28). For the definition of weights on $G_3$, $EC_3 \geq S_v + L$ (if $EC_3$ is hamiltonian, the inequality holds as an equality). Define $S_i$ to be the set of arcs of $S_v$ plus two copies of the fixed arcs (fig.29a). Then $S_i \leq S_v + 2L$ and $S_i \leq EC_3 + L$.  

24
Consider \( \hat{S}_t \). For the definition of \( \delta^E_2 \) and \( \delta^A_2 \), \( \hat{S}_t \leq S_v + q \delta^E_2 + N \delta^A_2 \), where \( q \) is the number of variable arcs in \( S_v \). Then \( S_v + \hat{S}_t \leq 2S_v + q \delta^E_2 + 2L + N \delta^A_2 \) (fig. 29b).

The set of arcs \( S_t \cup \hat{S}_t \) can be traversed in two arc-disjoint ways, that are both eulerian spanning cycles of \( G_2 \) and feasible solutions (fig. 30). Choose the best of the two and call it \( EC_2 \).

(iii) \( EC_2 \leq \frac{1}{2}(S_t + \hat{S}_t) \leq S_v + L + \frac{1}{2}q \delta^E_2 + \frac{1}{2}N \delta^A_2 \)

Step 4: substitute all pairs of adjacent variable arcs by single variable arcs with the same endpoints until a hamiltonian cycle \( HC_2 \) is left. By condition \( \beta' \),

(iv) \( HC_2 \leq EC_2 \).

From inequalities (i), (ii), (iii) and (iv), from the relations \( q = (N - 1) + \frac{\beta}{2} \) and \( \sigma = N - e \) and from the definition of \( \delta_2 \) it follows

\[
HC_2 \leq \frac{3}{2} HC^{*}_{2*} + \frac{3}{2} L + 2N \delta_2
\]

and when total costs are considered

\[
HC_2 \leq \frac{3}{2} HC^{*}_{2*} + L + 2N \delta_2
\]
Figure 28

Figure 29

Figure 36
3.3 Algorithm RT2 for the ASCP

**Input:** a complete weighted digraph $G_2(V_2, E_2, A_2)$, where $A_2$ is the subset of fixed arcs and such that conditions $\alpha'$ and $\beta'$ hold.

**Output:** a Hamiltonian cycle traversing all fixed arcs.

Let us indicate with $d_{uv}$ the cost of the generic variable arc $(u, v)$ of $G_2$.

**Step 1:** define a complete unoriented graph $G_3(V_3, E_3)$ with $N$ nodes, one for each fixed arc of $G_2$; define the following edge weights: $w_{ij} = d_{h_i t_j} + d_{h_j t_i} + l_i + l_j \ \forall i, j \in V_3$.

**Observation 3.1** The TI holds on $G_3$.

**Proof.** From condition $\beta'$ it follows $d_{h_i t_j} \leq d_{h_i t_k} + l_k + d_{h_k t_j}$ and $d_{h_j t_i} \leq d_{h_j t_k} + l_k + d_{h_k t_i}$. Summing up the two inequalities, the statement of the observation follows from the definition of the weights on $G_3$.

From the definition of $G_3$, $HC_3 = (HC_{2*} + H\hat{C}_{2*}) + 2L \leq HC_{2*} + H\hat{C}_{2*} + 2L = 2HC_{2*} + 2L + (H\hat{C}_{2*} - HC_{2*})$ and by the definition of $\delta^E$, (i) $HC_3 \leq 2HC_{2*} + 2L + N\delta^E$.

**Step 2:** apply the algorithm of Christofides and obtain a Hamiltonian cycle $HC_3$ on $G_3$.

(ii) $HC_3 \leq \frac{3}{2}HC_{2*}$.

**Step 3:** transform $HC_3$ into a solution on $G_2$: consider the pairs of variable arcs $(h_i, t_j)$ and $(h_j, t_i)$ correspondent to edges of $HC_3$; add two copies of each fixed arc; call $S_t$ the resulting arc set (fig.31).

$S_t = (HC_3 - 2L) + 2L = HC_3$. $S_t$ can be traversed in two arc-disjoint ways, which are both feasible solutions (fig.32). Choose the best of the them and call it $HC_{2t}$.

(iii) $HC_{2t} \leq \frac{1}{2}S_t$

From inequalities (i), (ii) and (iii) it follows

$$HC_{2t} \leq \frac{3}{2}HC_{2*} + \frac{1}{2}L + \frac{3}{4}N\delta^E$$

and when total costs are considered

$$HC_{2t} \leq \frac{3}{2}HC_{2*} + \frac{3}{4}N\delta^E$$
Figure 31

Figure 32
3.4 Comparison

The data-dependent worst-case bounds provided by algorithms LA, SA and RT2 are of the form:

\[ \eta \leq k + k_L \frac{L}{HC_2^*} + k_{\delta_2} \frac{N\delta_2}{HC_2^*} \]

Tables in figure 33 show the comparison between the three bounds. As for the GSCP, also for the ASCP the worst-case bound of algorithm RT2 dominates the one of algorithm SA.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>k</th>
<th>k_L</th>
<th>k_{\delta_2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>3</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>SA</td>
<td>3/2</td>
<td>3/2</td>
<td>3/2</td>
</tr>
<tr>
<td>RT2</td>
<td>3/2</td>
<td>1/2</td>
<td>3/4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>k</th>
<th>k_L</th>
<th>k_{\delta_2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>3</td>
<td>-2</td>
<td>1/2</td>
</tr>
<tr>
<td>SA</td>
<td>3/2</td>
<td>1</td>
<td>3/2</td>
</tr>
<tr>
<td>RT2</td>
<td>3/2</td>
<td>0</td>
<td>3/4</td>
</tr>
</tbody>
</table>

Figure 33
4 ATSP instances with constant worst-case bound

Computational complexity theory allows to classify combinatorial optimization problems: in particular the ATSP belongs to the class of NP-hard problems [5], that is no algorithm is known for finding the optimal solution in time polynomial in the size of the problem (conveniently measured by the number of nodes of the graph). Sahni and Gonzales [9] proved that even finding a solution within a prespecified constant ratio from the global optimum is also NP-hard, if the ATI can be violated. Anyway even for the ATSP with ATI no polynomial algorithm is known to provide a constant worst-case bound. In [3] Frieze, Galbiati and Maffioli consider the ATSP with ATI and describe some polynomial algorithms, whose worst-case performance is data-dependent. Since it is related to the complexity of the problem, the computational complexity analysis does not take into account the data of any particular instance. Anyway, when a particular instance has to be solved, it is possible to achieve more information about how difficult it is, just exploiting information contained in input data. Thus, instead of examining the worst-case performance of algorithms on all possible instances of a problem, we ask an inverse question: which are the instances that can be solved within a prespecified worst-case bound by some known polynomial time algorithm?

In this section we show how to analyze ATSP instances, exploiting the known algorithms LA, SA and RT2, which are polynomial and give constant worst-case bounds for the SCP. In order to do so, we study how an ATSP instance can be equivalently reformulated as a SCP instance without TI and under which conditions algorithms LA, SA and RT2 can be used to obtain constant bounds even when the TI does not hold. There are two ways to reformulate an ATSP instance as a SCP instance. In both of them each node of the ATSP instance corresponds to an arc of the SCP instance. Hereafter we outline two different procedures that do the transformation in polynomial time.

Procedure Transform-ATSP-into-SCP-1

**Input:** an ATSP instance, defined by a digraph $G(V_G, A_G)$ with weights $d$ on arcs.

**Output:** an SCP instance, defined by a mixed graph $H(V_H, E_H, A_H)$, such that there is a one-to-one correspondence between feasible solutions of the two instances and their value is preserved.

**Step 1:** for each node $i$ of $V_G$ define two nodes $h_i$ and $t_i$ in $V_H$ and an arc $(t_i, h_i)$ in $A_H$ and give the arc an arbitrary large cost $l \geq \max_{i,j \in V_G} \{d_{ij} - d_{ji}\}$.

**Step 2:** Insert in $E_H$ all head-to-tail edges $[h_i, t_j]$ and define their cost to be $w_{ij} = d_{ij}$ (fig.34).

**Step 3:** Add to $E_H$ all the missing edges and give them a weight $M \geq \max_{i,j \in V_G} \{d_{ij}\}$.

Procedure Transform-ATSP-into-SCP-2

**Input:** an ATSP instance, defined by a digraph $G(V_G, A_G)$ with weights $d$ on arcs.

**Output:** a SCP instance, defined by a mixed graph $H(V_H, E_H, A_H)$, such that there is a one-to-one correspondence between feasible solutions of the two instances and their
value is preserved.

Step 1: define suitable quantities $U_i \geq 0$ and $E_i \geq 0$ for each node $i$ of $V_G$.

Step 2: for each node $i$ of $V_G$ define two nodes $h_i$ and $t_i$ in $V_H$ and an arc $(t_i, h_i)$ in $A_H$ and give the arc a cost $l_i = U_i + E_i$.

Step 3: insert in $E_H$ all head-to-tail edges $[h_i, t_j]$ and define their cost to be $w_{ij} = d_{ij} - U_i - E_j$ (fig.35). Insert also edges $[h_i, t_i]$ with weight equal to $l_i$.

Step 4: Add to $E_H$ all the missing edges and give them a weight equal to $M \geq \max_{i,j \in V_a \{d_{ij}\}}$.

The technique used in Transform-ATSP-into-SCP-2 is similar to that discussed in [6] in order to obtain data-dependent bounds for the TSP when TI is violated. In that paper it is shown what is the best way to obtain data-dependent bounds for TSP instances in which the TI is violated, using algorithms that give constant bounds for instances in which the TI holds. Such data-dependent bounds are obtained by solving an auxiliary linear assignment problem whose dual variables are just the quantities $U_i$ and $E_i$ we have mentioned above.

Remark. The TI does not hold on the mixed graph obtained by either of the transformations.
After the first (resp. second) transformation every solution of the ATSP corresponds to a solution of the SCP whose value is given by the variable (resp. total) costs. When variable costs are considered, algorithm LA provides a constant worst-case bound equal to 3 (see appendix); when total costs are considered, the combination of algorithms LA and SA gives a constant worst-case bound equal to 9/5 and the combination of LA with RT2 gives a constant worst-case bound equal to 15/7 (see appendix). Since the TI is not satisfied in general, the above transformations are not sufficient to generate instances of the SCP as those defined in the introduction. In particular condition $\alpha$ is satisfied but condition $\beta$ is violated. In the next subsections we derive sufficient conditions for the algorithms to give the same constant bounds as for the SCP with TI. Obviously, more restrictive conditions correspond to a better bound. In order to test the meaningfulness of the conditions obtained we require (a) that some graph exists satisfying them (that is they are not too restrictive) and (b) on such graphs no better or equivalent worst-case bound can be immediately derived from the simple examination of input data: in particular we consider the trivial lower and upper bounds, $LB$ and $UB$, given by the sum of the $N$ smallest and the $N$ biggest arcs. Let us also define the following quantities:
\[
\delta_i^{in} = \max_{j \in V} \{d_{ji} - d_{ij}, 0\} \quad \forall i = 1, \ldots, N
\]
\[
\delta_i^{out} = \max_{j \in V} \{d_{ij} - d_{ji}, 0\} \quad \forall i = 1, \ldots, N
\]
Let us indicate the amount by which the ATI is satisfied in each triangle by $\overline{\Delta}_{ijk} = d_{ik} + d_{kj} - d_{ij} \quad \forall i, j, k = 1, \ldots, N$. 

33
4.1 Algorithm LA: worst-case bound = 3

A worst-case bound equal to 3 can be achieved by applying Transform-ATSP-into-SCP-1 and then algorithm LA. From the analysis of the algorithm it follows that the only conditions that must hold are those required by the postprocessing. For the definition of $M$, only head-to-tail edges belong to the eulerian cycle built by LA. Therefore in the postprocessing triples of adjacent head-to-tail edges are replaced by one head-to-tail edge (fig.36).

![Figure 36](image)

Therefore, necessary and sufficient condition for an ATSP instance to be solvable within worst-case bound equal to 3 by algorithm LA (plus postprocessing) is:

\[(A1) \quad d_{kl} \leq d_{il} + d_{ij} + d_{kj} \quad \forall i, j, k, l \in V_G\]

The example in figure 37 shows a graph satisfying condition (A1); in that example $\frac{UB}{LB} = \frac{14}{4} > 3$.

Testing condition (A1) requires $O(N^4)$, whereas the application of the algorithm requires $O(N^3)$. More restrictive conditions that are faster to test can be obtained, but we could not prove they are meaningful. For instance, it is possible to imply condition (A1) in the following way.

\[d_{kl} \leq d_{kj} + d_{ij} + d_{il} = d_{kj} + d_{jl} + (d_{ij} - d_{jl})\]

This is implied by the two conditions

\[d_{kl} \leq d_{kj} + d_{jl} + 1/2(d_{ij} - d_{jl}) \quad \text{and} \quad d_{jl} \leq d_{ji} + d_{il} + 1/2(d_{ij} - d_{jl}),\]

which are implied by

\[
\Delta_{ijk} \geq \frac{1}{2}(\delta_k^{in} + \delta_k^{out}) \quad \forall i, j, k \in V_G,
\]

A sufficient condition is then

\[
\overline{\Delta}_{ijkl} \geq \frac{1}{2}(\delta_k^{in} + \delta_k^{out}) \quad \forall i, j, k \in V_G.
\]

On the other side such condition is stronger than the ATI, since $\delta_k^{in}$ and $\delta_k^{out}$ are non-negative and we could not prove it is meaningful.
Remark. Even if the sufficient condition (41) does not hold, algorithm LA provides a data-dependent bound for the ATSP, like those of [3] and [8]. In particular, algorithm LA (plus postprocessing) produces a solution $HC$ of the ATSP such that

$$HC \leq 3HC^* + N\Delta + \frac{1}{2}N\delta$$

where $\Delta = \max_{i,j,k} (d_{ij} - (d_{ik} + d_{kj}))$ and $\delta = \max_{i,j} (d_{ij} - d_{ji})$. 

Figure 37
4.2 Algorithms LA + RT2: worst-case bound = 15/7

A constant worst-case bound equal to 15/7 can be obtained by applying Transform-ATSP-into-SCP-2 followed by the combination of algorithms LA and RT2. In order to obtain the constant bound the following conditions are necessary and sufficient.

(B1) \( l_i \geq 0 \quad \forall i = 1, \ldots, N \).
(B2) \( w_{ij} \geq 0 \quad \forall i, j = 1, \ldots, N \).
(B3) \( w_{ij} + w_{ji} \leq w_{ik} + w_{kj} + w_{jk} + w_{ki} \quad \forall i, j, k = 1, \ldots, N \).
(B4) \( w_{ij} \leq l_i + l_j + w_{kl} + w_{li} \quad \forall i, j = 1, \ldots, N \).
(B5) \( w_{kl} \leq w_{kj} + w_{ij} + w_{il} \quad \forall i, j, k, l = 1, \ldots, N \).

Conditions (B1) and (B2) are required for obtaining a mixed graph with non-negative costs on the arcs and edges after the transformation. Condition (B3) is required by algorithm RT2, that builds an auxiliary undirected graph on which the TI must hold. Condition (B4) is necessary in step 1 of algorithm RT2 (see appendix). Condition (B5) is required in the postprocessing of the output of algorithm LA (as outlined in the previous subsection).

We substitute condition (B3) with condition (B3'), that implies it.

(B3') \( w_{ij} \leq w_{ik} + w_{kj} \quad \forall i, j, k = 1, \ldots, N \).

We choose \( U_i = \frac{1}{4} \delta_{\text{out}} \) and \( E_i = \frac{1}{4} \delta_{\text{in}} \) for all \( i = 1, \ldots, N \). By this choice conditions (B1) and (B4) are always satisfied. The following set of conditions is now sufficient.

(B2) \( \delta_{\text{out}} + \delta_{\text{in}} \leq 4d_{ij} \quad \forall i, j = 1, \ldots, N \)

(B3') \( d_{ij} \leq d_{ik} + d_{kj} - \frac{1}{4}(\delta_{\text{in}} + \delta_{\text{out}}) \)

(B5') \( d_{kl} \leq d_{kj} + d_{ij} + d_{il} - \frac{1}{2}(\delta_{\text{in}} + \delta_{\text{out}} + \delta_{\text{in}} + \delta_{\text{out}}) \)

Condition (B5') is obtained from condition (B5), by the use of the inequality \( d_{ji} - d_{ij} \leq \frac{1}{2}(\delta_{\text{in}} + \delta_{\text{out}}) \) coming from the definition of parameters \( \delta_{\text{in}} \) and \( \delta_{\text{out}} \).

Figure 38 shows an example of a graph satisfying all the three sufficient conditions, in which \( \frac{UB}{LB} = \frac{12}{5} > \frac{15}{7} \).

Conditions (B3') and (B5') are both implied by the following condition (B6).

(B6) \( d_{ij} \leq d_{ik} + d_{kj} - \frac{1}{2}(\delta_{\text{in}} + \delta_{\text{out}}) \quad \forall i, j, k = 1, \ldots, N \).

By definition of \( \Delta_{ijk} \) a set of sufficient conditions is then:

(B2) \( \delta_{\text{out}} + \delta_{\text{in}} \leq 4d_{ij} \quad \forall i, j = 1, \ldots, N \)

(B6) \( \Delta_{ijk} \geq \frac{1}{2}(\delta_{\text{in}} + \delta_{\text{out}}) \quad \forall i, j, k = 1, \ldots, N \)
This set of sufficient conditions is checkable in $O(N^3)$; anyway we could not prove that it is meaningful. It can be remarked that it is stronger than the ATI.

Figure 38
4.3 Algorithms LA + SA: worst-case bound $= 9/5$

A constant worst-case bound equal to $9/5$ can be obtained by applying Transform-ATSP-into-SCP-2 followed by the combination of algorithms LA and SA (both followed by postprocessing). In order to obtain the constant bound the following conditions are necessary and sufficient.

(C1) $l_i \geq 0 \ \forall i = 1, \ldots, N$.
(C2) $w_{ij} \geq 0 \ \forall i, j = 1, \ldots, N$.
(C3) $z_{ij} \leq z_{ik} + z_{jk} \ \forall i, j, k = 1, \ldots, N$, where $z_{ij} = \min\{w_{ij}, w_{ji}\}$.
(C4) $w_{ij} \leq l_i + w_{ji} + l_j \ \forall i, j = 1, \ldots, N$.
(C5) $w_{ik} \leq w_{ij} + w_{kj} + l_k$
(C6) $w_{ik} \leq w_{jk} + w_{ji} + l_i$
(C7) $w_{kl} \leq w_{ii} + w_{ij} + w_{kj} \ \forall i, j, k, l = 1, \ldots, N$.

Conditions (C1) and (C2) are required for obtaining a mixed graph with non-negative costs after the transformation. Condition (C3) is required by algorithm SA, that builds an auxiliary undirected graph on which the TI must hold. Conditions (C4) (fig.39), (C5) (fig.40) and (C6) (fig.41) are necessary in the postprocessing for replacing triples of adjacent head-to-tail edges by single edges. Condition (C7) is required by the postprocessing of the output of algorithm LA as in the previous subsections.

![Figure 39](image-url)
By substitution from the definition of weights $l$ and $w$, we obtain the equivalent set of conditions.

(C1) $U_i + E_i \geq 0 \quad \forall i = 1, \ldots, N.$

(C2) $U_i + E_j \leq d_{ij} \quad \forall i, j = 1, \ldots, N.$

(C3) $z_{ij} \leq z_{ik} + z_{jk} \quad \forall i, j, k = 1, \ldots, N.$

(C4) $d_{ij} - d_{ji} \leq 2U_i + 2E_j \quad \forall i, j = 1, \ldots, N.$

(C5) $d_{ij} \leq d_{ik} + d_{jk} + 2E_j - 2E_k$

(C6) $d_{ij} \leq d_{kj} + d_{ki} + 2U_i - 2U_k$

(C7) $d_{kl} \leq d_{il} + d_{ij} + d_{kj} - 2U_i - 2E_j \quad \forall i, j, k, l = 1, \ldots, N.$
We choose again \( U_i = \frac{1}{4} \delta_i^{\text{out}} \) and \( E_i = \frac{1}{4} \delta_i^{\text{in}} \) for all nodes \( i = 1, \ldots, N \). Therefore conditions (C1) and (C4) are satisfied. We obtain the following sufficient conditions:

\[
\begin{align*}
\text{(C2)} & \quad \delta_i^{\text{out}} + \delta_j^{\text{in}} \leq 4d_{ij} \quad \forall i, j = 1, \ldots, N \\
\text{(C3)} & \quad z_{ij} \leq z_{ik} + z_{jk} \quad \forall i, j, k = 1, \ldots, N \\
\text{(C5)} & \quad d_{ij} \leq d_{ik} + d_{jk} + \frac{1}{2}(\delta_j^{\text{in}} - \delta_k^{\text{in}}) \quad \forall i, j, k = 1, \ldots, N \\
\text{(C6)} & \quad d_{ij} \leq d_{kj} + d_{ki} + \frac{1}{2} (\delta_i^{\text{out}} - \delta_k^{\text{out}}) \quad \forall i, j, k = 1, \ldots, N \\
\text{(C7)} & \quad d_{kl} \leq d_{il} + d_{ij} + d_{kj} - \frac{1}{2} (\delta_i^{\text{out}} - \delta_j^{\text{in}}) \quad \forall i, j, k, l = 1, \ldots, N
\end{align*}
\]

Figure 42 shows an example of a graph satisfying sufficient conditions above. The bound is meaningful since \( \frac{UB}{LB} = \frac{14}{\epsilon} > \frac{9}{\epsilon} \).

---

Due to condition (C7) the time required for the testing is \( O(N^4) \). Faster but more restrictive conditions can be obtained, but again we could not prove they are meaningful. For instance, the following system of two conditions

\[
\begin{align*}
\text{(C2)} & \quad \delta_i^{\text{out}} + \delta_j^{\text{in}} \leq 4d_{ij} \quad \forall i, j = 1, \ldots, N \\
\text{(C8)} & \quad \overline{\Delta}_{ijk} \geq \frac{3}{4} (\delta_k^{\text{in}} + \delta_k^{\text{out}}) + \frac{1}{4} (\delta_i^{\text{in}} + \delta_i^{\text{out}} + \delta_j^{\text{in}} + \delta_j^{\text{out}}) \quad \forall i, j, k = 1, \ldots, N
\end{align*}
\]

is sufficient and is checkable in \( O(N^3) \) but it seems to be largely too restrictive.
ACKNOWLEDGMENTS

Prof. Anton Volgenant pointed out the relationship between the reduction outlined in section 4 and those illustrated in [6] for the TSP.
The first author acknowledges the support of the Human Capital and Mobility Program of the European Community.
References

   *Computers and Intractability: a Guided Tour to the Theory of NP-completeness*
   Freeman, San Francisco, 1979

   *Algorithm 97: Shortest Path*
   Comm. ACM 5, 1962 [345]

   *On the worst-case performance of some algorithms for the Asymmetric Traveling Salesman Problem*
   Networks 12, 1982 [23-39]

   *Approximation algorithms for some routing problems*

   *Computers and Intractability: a Guide to the Theory of NP-completeness*
   Freeman, San Francisco, 1979

   *Data-dependent bounds for heuristics to find a minimum weight Hamiltonian circuit*

   *The traveling salesman problem*
   John Wiley, Chichester, 1985 [171-177]

   *Worst-case analysis of two approximate algorithms for the Asymmetric Traveling Salesman Problem*
   to appear in European Journal of Operational Research

   *P-complete problems and approximate solutions*
Stacker Crane Problem:
*Instance:* a weighted mixed graph $G(V, E, A)$, where edges in $E$ are unoriented and arcs in $A$ are oriented, satisfying the following two conditions:

$\alpha$) all nodes in $V$ are endpoint of exactly one arc;
$\beta$) all the edges satisfy the triangle inequality.

*Question:* find a minimum cost cycle on $G$ that traverses all arcs.

Algorithms LA and SA output eulerian spanning cycles. Such solutions can be made hamiltonian without worsening their value, by postprocessing procedures that repeatedly replace pairs of adjacent edges by a single edge, exploiting the TI.


### 5.1 Algorithm LA for the SCP

*Input:* a weighted mixed graph $G(V, E, A)$, satisfying conditions $\alpha$ and $\beta$.

*Output:* a hamiltonian cycle traversing all arcs.

**Step 1:** find the minimum weighted matching $M_v^*$ on an auxiliary bipartite graph $K(T, H, E_K)$, where $T$ is the set of nodes that are tails of arcs in $A$, $H$ is the set of nodes that are heads of arcs in $A$ and $E_K$ is made of all edges $[i, j]$ of $E$ such that $i \in T$ and $j \in H$. $M_v^*$ can be computed in time polynomial in $N$ by a linear assignment algorithm. The matching provides a lower bound on the variable cost of the minimum cycle $HC_v^*$:

(i) $M_v^* \leq HC_v^*$

The edges of $M_v^*$ and the arcs of $A$ form a number $s$ of subtours on $G$ (fig.43).

**Step 2:** find the minimum cost edge set $ST_v^*$ connecting the subtours (fig.44). Since in every feasible solution all subsets of nodes must be connected,

(ii) $ST_v^* \leq HC_v^*$

**Step 3:** duplicate the edges of $ST_v^*$. Call $S_v = M_v^* + 2ST_v^*$ the resulting edge set. The edges of $S_v$ and the arcs of $A$ form a spanning eulerian cycle (fig.45).

(iii) $EC_v = S_v = M_v^* + 2ST_v^*$

From inequalities (i), (ii) and (iii) it follows:

$$EC_v \leq 3HC_v^*$$

and if total costs are considered
Remark. In this paper we consider algorithm LA in a slightly improved version: self-loops on arcs are not allowed when the minimum matching is looked for. Neither the worst-case bound nor the computational complexity of the algorithm are altered by this assumption.
5.2 Algorithm SA for the SCP

**Input:** a weighted mixed graph $G(V, E, A)$, satisfying conditions $\alpha$ and $\beta$.

**Output:** a Hamiltonian cycle traversing all arcs.

Let us indicate by $d_{uv}$ the cost of the generic edge $[u, v] \in E$.

**Step 1:** define a complete unoriented graph $G_1(V_1, E_1)$, with $V_1 = A$ and edge weights $w_{ij} = \min\{d_{h_i t_i}, d_{h_j t_j}, d_{h_i b_i}, d_{h_j b_j}\} \forall i, j \in V_1$. Substitute the edge weights by the length of the shortest path between their endpoints, so that the TI holds on $G_1$. For the construction of $G_1$, it follows

(i) $HC^*_1 \leq HC^*_0$.

**Step 2:** apply the algorithm of Christofides to graph $G_1$, obtaining an Eulerian cycle $EC_1$.

(ii) $EC_1 \leq \frac{3}{2}HC^*_1$.

**Step 3:** consider the set $S$ made of the edges of $G$ correspondent to the edges of $EC_1$ plus all arcs of $A$ plus all edges $[h_i, t_i]$ (fig.46). Duplicate all the arcs and edges in $S$, obtaining the set $2S$ (fig.47). $2S \leq 2(EC_1 + 2L)$. The set $2S$ can be traversed in two different and disjoint ways, both of which are Eulerian spanning cycles of $G$ (fig.48). Choose the best of the two and call it $EC$. Then

(iii) $EC \leq \frac{3}{2}(2S) \leq EC_1 + 2L$

From inequalities (i), (ii) and (iii) it follows

$$EC\leq \frac{3}{2}HC^*_v + L$$

and when total costs are considered

$$EC_t \leq \frac{3}{2}HC^*_t + \frac{1}{2}L$$
5.3 Algorithm RT2 for the SCP

**Input:** a weighted mixed graph \(G(V, E, A)\), satisfying conditions \(\alpha\) and \(\beta\).

**Output:** a Hamiltonian cycle traversing all arcs.

Let us indicate by \(d_{uv}\) the cost of the generic head-to-tail edge \([u, v] \in E\).

**Step 1:** define a complete unoriented graph \(G_1(V_1, E_1)\), with \(V_1 = A\) and edge weights \(w_{ij} = d_{ij} + d_{ji} + l_i + l_j \forall i, j \in V_1\). On \(G_1\) the TI holds, since the relations \(d_{ij} \leq d_{ik} + l_k + d_{kj}\) and \(d_{ji} \leq d_{jk} + l_k + d_{ki}\) are implied by condition \(\beta\) (fig. 49). For the construction of \(G_1\), it follows \(HC_1^* = (HC_v + HC_{v'})^* + 2L \leq HC_v^* + HC_{v'}^* + 2L\). Since for condition \(\beta\), \(d_{ij} \leq l_i + d_{ji} + l_j\), it follows that \(HC_v^* \leq HC_{v'}^* + 2L\) (fig. 50). Therefore
(i) \(HC_1^* \leq 2HC_v^* + 4L\).

**Step 2:** apply the algorithm of Christofides, which provides an Hamiltonian cycle \(HC_1\).
(ii) \(HC_1 \leq \frac{3}{2}HC_1^*\).

**Step 3:** define \(S\) to be the set of pairs of edges of \(G\) corresponding to the edges of \(HC_1\), plus two copies of each arc of \(A\) (fig. 51). For the definition of weights \(w\), it follows that \(S = (HC_1 - 2L) + 2L = HC_1\). \(S\) can be traversed in two disjoint ways; call \(HC\) the best of the two. \(HC_t \leq \frac{1}{2}S = \frac{1}{2}HC_1\) and then
(iii) \(HC_t \leq \frac{1}{2}HC_1 - L\).

From inequalities (i), (ii) and (iii) it follows
\[
HC_t \leq \frac{3}{2}HC_v^* + 2L
\]

and if total costs are considered
\[
HC_t \leq \frac{3}{2}HC_t^* + \frac{3}{2}L
\]
5.4 Comparison and combination of algorithms

The worst-case bounds provided by algorithms LA, SA and RT2 are of the form:

\[ \eta = \frac{HC}{HC^*} \leq k + kL \frac{L}{HC^*} \]

In [4] it is shown that running algorithms SA and LA and taking the best of the two solutions found, gives a worst-case bound equal to 9/5, when total costs are considered. We add the further observation that the worst-case bound is 3 when only variable costs are considered and that such bound is given by algorithm LA alone. This can be seen from the graphical representation of the bounds in figures 53 and 54.

An analogous combination of algorithms LA and RT2 gives a constant worst-case bound equal to 15/7, when total costs are considered. In fact LA gives \( EC_{t_LA} \leq 3HC_i^* - 2L \) and RT2 gives \( HC_{t_RT2} \leq \frac{3}{2}HC_i^* + \frac{3}{2}L \). The bounds on the costs of the two solutions are equal when \( 3HC_i^* - 2L = \frac{3}{2}HC_i^* + \frac{3}{2}L \), that is when \( \frac{3}{2}HC_i^* = \frac{7}{2}L \), that is \( L = \frac{3}{7}HC_i^* \) and in such case the bound is given by \( 3HC_i^* - 2(\frac{3}{7}HC_i^*) = \frac{15}{7}HC_i^* \).

Tables in figure 52 show the comparison between the three bounds. The worst-case bound of algorithm SA dominates the one of algorithm RT2.

| SCP \( \eta = \frac{HC}{HC^*} \) |
|------------------|---|---|
| algorithm | \( k \) | \( kL \) |
| LA | 3 | 0 |
| SA | 3/2 | 1 |
| RT2 | 3/2 | 2 |

| SCP \( \eta = \frac{HC}{HC^*} \) |
|------------------|---|---|
| algorithm | \( k \) | \( kL \) |
| LA | 3 | -2 |
| SA | 3/2 | 1/2 |
| RT2 | 3/2 | 3/2 |

Figure 52