# Combinatorial relaxations of the $k$-traveling salesman problem

Andrei Horbach

Christian-Albrechts-Universität zu Kiel,
Institut für Betriebswirtschaftslehre,
Olshausenstr 40, 24098 Kiel, Germany,
horbach@bwl.uni-kiel.de

**Abstract**

The $k$-traveling salesman problem, or $k$-TSP is: given a graph with edge weights and an integer $k$, find a simple cycle of minimum weight visiting exactly $k$ nodes. To obtain lower bounds for the traveling salesman problem the $2$-matching relaxation and the $1$-tree relaxation can be used. We generalize these two relaxations for the $k$-TSP.

**Keywords:** $k$-traveling salesman problem, $k$-TSP, matching, 2-matching, 1-tree.

# 1 Introduction

Let $G = (V, E)$ be a graph with node set $V$ and edge set $E$. A *chain* $L$ in $G$ between nodes $u$ and $v$ is a subgraph with nodes $u = v_0, v_1, \ldots, v_k = v$ and edges $\{v_{i-1}, v_i\}$, $i \in \{1, \ldots, k\}$, where all the edges are distinct. A chain without repetition of nodes is called a *path*. If in a chain $v_0 = v_k$, then the chain is called a *cycle*. If all the nodes in a cycle $C$ except $v_0$ and $v_k$ are distinct, then $C$ is a *simple cycle*, or *circuit*. Let $k$ be an arbitrary integer from $3$ to $n$. A simple cycle in $G$ consisting of exactly $k$ edges is a *k-cycle*.
Henceforth we will identify a chain
$L = \big\{\{v_0, v_1, \ldots, v_k\}, \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}\}\big\}$ with its edge set, i.e. we will write $L = \big\{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}\big\}$.
For a set $F \subseteq E$ we define $x^F \in \mathbf{B}^E$ to be the incidence vector of $F$ such that

$$x_e^F = \begin{cases} 1 & if\ e \in F \\ 0 & otherwise. \end{cases}$$

In the following we will identify subsets of $E$ with their incidence vectors.
The *$k$-traveling salesman problem* ($k$-TSP) is: given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, an integer $k \leq n$ and a cost vector $c \in \mathbf{R}^E$, find a $k$-cycle of minimum length.
The *(symmetric) traveling salesman problem* (TSP) is to find in $G$ an $n$-cycle of minimum length.
Some polyhedral results for the $k$-TSP have been obtained by Maurras and Nguyen [12, 11], Maurras, Kovalev, and Vaxès [10] and by Girlich, Höding, Horbach and Kovalev [5]. Some polyhedral results for the asymmetric $p$-cycle problem (the problem to find in a directed graph a cycle on $p$ nodes of minimum length) has been obtained by Hartmann and Özlük [6].
A branch-and-cut algorithm has been implemented by Nguyen [13]. An upper bound for the diameter of the convex hull of the feasible solutions of the $k$-TSP on a complete graph has been given by Girlich, Höding, Horbach and Kovalev [4].

Let $x \in \mathbf{R}^E$, $S, T \subseteq V$, $F \subseteq E$, $v \in V$. We use the following notation:

$$\begin{aligned} x(F) &:= \sum_{e \in F} x_e, \\ E(S,T) &:= \{\{s,t\} \mid s \in S, t \in T\}, \\ \delta(S) &:= E(S, V \setminus S), \\ \delta(v) &:= \delta(\{v\}). \end{aligned} \tag{1}$$

# 2 $1$-trees and Lagrangean relaxation

When solving the traveling salesman problem by branch and bound we need a good bounding technique. Polyhedral bounds, i.e. bounds obtained by solving some linear programming relaxations of TSP can be used. Consider the following linear programming relaxation of the traveling salesman problem.

$$\begin{array}{lllr} \min & c^T x & & \\ s.t. & 0 \le x_e \le 1, & \forall e \in E, & (i) \\ & x(\delta(v)) = 2, & \forall v \in V, & (ii) \\ & x(E(U)) \le |U| - 1, & \forall U \subseteq V, \emptyset \ne U \ne V. & (iii) \end{array} \tag{2}$$

No combinatorial polynomial algorithm is known to solve (2). Held and Karp proposed a method to solve (2) using 1-trees and Lagrangean relaxation, see [7] and [8]. Let $1$ be a node in $G = (V, E)$. A set $F \subseteq E$ is a $1$-*tree* if $|F \cap \delta(1)| = 2$ and $E \setminus \delta(1)$ forms a spanning tree on $V \setminus \{1\}$.
The idea is to consider the following equivalent formulation of (2)

$$\begin{array}{lllr} \min & c^T x & & \\ s.t. & 0 \le x_e \le 1, & \forall e \in E, & (i) \\ & x(\delta(v)) = 2, & \forall v \in V \setminus \{1\}, & (ii') \\ & x(\delta(1)) = 2, & & (ii'') \\ & x(E(U)) \le |U| - 1, & \forall U \subseteq V \setminus \{1\}, \emptyset \ne U \ne V \setminus \{1\}, & (iii) \\ & x(E) = |V|, & & (iv) \end{array} \tag{3}$$

and to introduce $(ii')$ into the objective function with Lagrangean multipliers:

$$\begin{array}{lllr} \max\limits_{\gamma \in \mathbf{R}^{V \setminus \{1\}}} \min & c^T x + \sum\limits_{v \in V \setminus \{1\}} \gamma_v (2 - x(\delta(v))) & & \\ s.t. & 0 \le x_e \le 1, & \forall e \in E, & (i) \\ & x(\delta(1)) = 2, & & (ii'') \\ & x(E(U)) \le |U| - 1, & \forall U \subseteq V \setminus \{1\}, \emptyset \ne U \ne V \setminus \{1\}, & (iii) \\ & x(E) = |V|. & & (iv) \end{array} \tag{4}$$

The extreme solutions of $(i)$, $(ii'')$, $(iii)$, $(iv)$ are 1-trees, hence the minimum of a linear function over $(i)$, $(ii'')$, $(iii)$, $(iv)$ can be found by applying any shortest spanning tree algorithm on graph $G \setminus \{1\}$.
Iteratively modifying the Lagrangean multipliers we achieve the optimum of (2).

## 3 $(k,1)$-forests and Lagrangean relaxation

The idea of Held and Karp can be extended for the $k$-TSP.

**Definition 1.** *A set $F \subseteq E$ is a $(k,1)$-forest if $|F| = k$ and $F$ contains at most one simple cycle.*

Consider the following IP-formulation of the $k$-TSP.

$$\begin{array}{lll r}
\min & c^T x & & \\
s.t. & 0 \le x_e \le 1, & \forall e \in E, & (i) \\
& x\big(\delta(v)\big) \le 2, & \forall v \in V, & (ii) \\
& x\big(\delta(v)\big) - 2x_e \ge 0, & \forall v \in V,\ \forall e \in \delta(v), & (iii) \\
& x_{(u,v)} + x(u,T) + x(v,S) - x(S,T) \le 2, & \forall u,v,S,T \ partition\ of\ V, & (iv) \\
& x(E) = k, & & (v) \\
& x_e \in \{0,1\}, & \forall e \in E. & (vi)
\end{array} \qquad (5)$$

Using the fact that each $k$-cycle is a $(k,1)$-forest with the degree of each node equal to $2$ or $0$ we obtain the following formulation of the $k$-TSP.

$$\begin{array}{lll r}
\min & c^T x & & \\
s.t. & x\big(\delta(v)\big) \le 2, & \forall v \in V, & (ii) \\
& x\big(\delta(v)\big) - 2x_e \ge 0, & \forall v \in V,\ \forall e \in \delta(v), & (iii) \\
& x \text{ is an incidence vector of a (k,1)-forest.} & & (vi')
\end{array} \qquad (6)$$

We denote the family of all $(l,1)$-forests, $l = 0,\dots,k$, by $\mathcal{F}_{\le k}$.

**Lemma 1.** *The pair $(E, \mathcal{F}_{\le k})$ is a matroid with the ground set $E$ and the family of independent sets $\mathcal{F}_{\le k}$.*

*Proof.* Obviously if $F \in \mathcal{F}_{\le k}$ then any subset of $F$ is also in $\mathcal{F}_{\le k}$.
We must proof: for given $F_1, F_2 \in \mathcal{F}_{\le k}$, $|F_1| < |F_2|$, there is $e \in F_2 \setminus F_1$ such that $F_1 \cup \{e\} \in \mathcal{F}_{\le k}$. Two cases are possible:
1. Set $F_1$ contains no cycle. Then for any $e \in F_2 \setminus F_1$ we have $F_1 \cup \{e\} \in \mathcal{F}_{\le k}$.
2. The set $F_1$ contains a cycle. Consider the connected components of $F_1$, say $K_1 = (V_1, E_1)$, $K_2 = (V_2, E_2)$, ..., $K_s = (V_s, E_s)$. Since $|F_1| < |F_2|$ and $F_2$ contains at most one cycle, there is an edge $e \in F_2$ such that $e \notin \bigcup_{i=1}^{s} E(V_i, V_i)$ and therefore $F_1 \cup \{e\} \in \mathcal{F}_{\le k}$.
This completes the proof. □

A minimal weight $(k,1)$-forest in a given graph with $n$ nodes and $m$ edges can be found in $O(m \log n)$ operations using the greedy method which is an adaptation of Kruskal's algorithm, see [9].
We can solve (6) using subgradient optimization. The advantage of this approach is that we do not need to implement a linear programming algorithm and separation technique. However, the lower bounds may be worse than that obtained from the linear programming relaxation of (5).

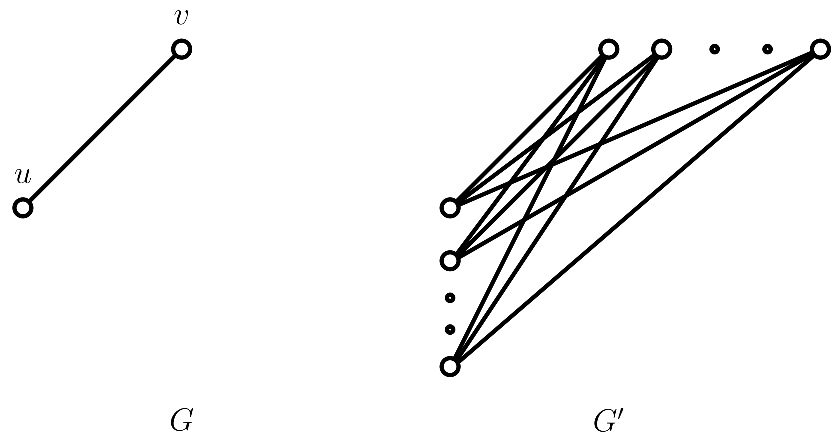


Figure 1: Reduction of the perfect $b$-matching problem to the perfect matching problem

# 4 Matchings

Let $G = (V, E)$ be a graph and $b$ be a vector in $\mathbf{Z}_+^V$. A vector $x \in \mathbf{Z}_+^E$ is a $b$-*matching* if

$$x\big(\delta(v)\big) \leq b_v \tag{7}$$

holds for each $v \in V$. A $b$-matching $x$ is called *perfect* if

$$x\big(\delta(v)\big) = b_v \tag{8}$$

holds for each $v \in V$. A $b$-matching $x$ is called *simple* if $x$ is a zero-one vector. If $b$ is the all-one vector, then $x$ is a $1$-*matching* or simply a *matching*. If all the components of $b$ equal $2$, then $x$ is a $2$-*matching*.
Let $c$ be a weight vector associated with $G$. The problem of finding a (simple, perfect) $b$-matching of minimum (maximum) weight is a minimum (maximum) weight (simple, perfect) $b$-matching problem. All such matching problems are polynomial time solvable, see e.g. Derigs [2] and Schrijver [14]. The matching and the $2$-matching problems are defined analogously.
Edmonds [3] gives a polynomial algorithm for the maximum weight matching problem. The minimum weight perfect matching problem can be easily reduced to the maximum weight matching problem by the following modification of the weight vector: for each edge $e$ set $c'_e := M - c_e$ where $M$ is a big number, and replace the weight vector $c$ with $c'$.
By the following transformation, which is due to Tutte [15], the perfect $b$-matching problem on a given weighted graph $G = (V, E)$ with the weight vector $c$ can be reduced to the perfect matching problem on a graph $G' = (V', E')$ with a weight vector $c'$. The graph $G'$ is constructed as follows: for each node $v \in V$ create $b_v$ nodes $v_1, v_2, \ldots, v_{b_v}$ in $V'$ and for each edge $\{u, v\} \in E$ create $b_v \cdot b_u$ edges $\{u_1, v_1\}$, $\{u_1, v_2\}$,..., $\{u_1, v_{b_v}\}$, ..., $\{u_{b_u}, v_1\}$, $\{u_{b_u}, v_2\}$, ..., $\{u_{b_u}, v_{b_v}\}$ with weights $c'_{u_i v_j} := c_{uv}$ for $i = 1, \ldots, b_u$, $j = 1, \ldots, b_v$, see figure 1.
Now if $x'$ is a feasible solution of the perfect matching problem on $G'$, then $x^* \in \mathbf{Z}_+^E$ with

$$x^*_{uv} = \sum_{i=1}^{b_u} \sum_{j=1}^{b_v} x'_{u_i v_j} \tag{9}$$

for each edge $\{u, v\} \in E$ is a perfect $b$-matching in $G$. Moreover, $c'x' = cx^*$ and for each perfect $b$-matching in $G$ there is at least one perfect matching in $G'$ with identical weight.

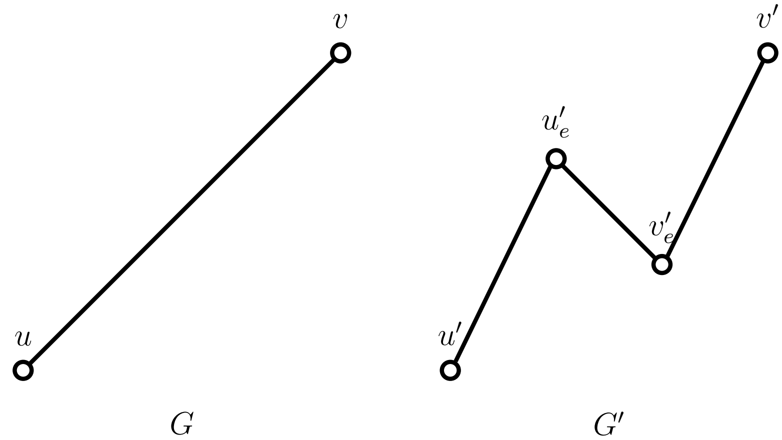


Figure 2: Reduction of the simple perfect $b$-matching problem to the perfect $b$-matching problem

Hence, if $x'$ is an optimal solution of the perfect matching problem on $G'$, then $x^*$ is an optimal solution of the perfect $b$-matching problem on $G$.
The perfect matching problem on $G'$ can be solved in time polynomial in $|G'|$, so the perfect $b$-matching problem on $G$ can be solved in time bounded by polynomial in $b(V)$ and $|E|$, which gives a pseudo-polynomial $\big($polynomial in the size of the problem and $b(V)\big)$ algorithm for the simple perfect $b$-matching problem. The simple perfect $b$-matching problem on a given graph $G = (V, E)$ can be reduced to the perfect $b'$-matching problem on some graph $G' = (V', E')$ by the following transformation.
For each node $v$ in $V$ create a node $v'$ in $V'$ and set $b'_{v'} = b_v$. For each edge $e \in E$, $e = \{u, v\}$, introduce two nodes $u_e$ and $v_e$ in $V'$ and three edges, $\{v', v_e\}$, $\{v_e, u_e\}$, and $\{u_e, u'\}$ in $E'$. Set $b'_{u_e} := b'_{v_e} := 1$ and set $c'_{v'v_e} := c'_{u_eu'} := c_{uv}$, $c_{v_eu_e} := 0$, see figure 2.
Now the optimal solution $x^*$ of the simple perfect $b$-matching problem can be derived from the optimal solution $x'$ of the perfect $b'$-matching problem by setting $x_e := x'_{uu_e}$ for each $e \in E$, $e = \{u, v\}$.
Note that for the simple perfect $b$-matching problem we can assume for each node $v$ the parameter $b_v$ to be at most the degree of $v$, otherwise the problem is not feasible. Therefore, the transformation described above reduces the simple perfect $b$-matching problem on $G = (V, E)$ to the perfect matching problem on a graph $G' = (V', E')$ with $O(|V|^2)$ nodes. It gives a polynomial algorithm for the simple perfect $b$-matching problem.
Another reduction of a (simple) $b$-matching problem is due to Berge, see [1]. For a deeper survey of matchings see Derigs [2] and Schrijver [14].

# 5 The simple $(k, 2)$-matching relaxation

Let $G = (V, E)$ be a graph with $n$ nodes and $k$ be an integer, $0 \le k \le n$. A 2-matching $x$ is a $(k, 2)$-matching if $x(E) = k$. A $(k, 2)$-matching $x$ is called *perfect* if

$$x\big(\delta(v)\big) \in \{0, 2\} \tag{10}$$

for each $v \in V$. A $(k, 2)$-matching $x$ is called *simple* if $x$ is a zero-one vector.
The problem to find a simple $(k, 2)$-matching of minimum weight $c^T x$ in a given graph $G$ is the *minimum weight simple $(k, 2)$-matching problem.*

**Theorem 1.** *The minimum weight simple $(k, 2)$-matching problem is polynomial time solvable.*

*Proof.* The simple $(k,2)$-matching problem on $G=(V,E)$ can be reduced to the perfect $b$-matching problem on the graph $G'=(V',E')$ with the weight vector $c' \in \mathbf{R}^{E'}$ and the $b$-vector $b' \in \mathbf{Z}_+^{V'}$ defined as follows. For each node $v \in V$ create a node $v' \in V'$ with $b(v')=2$. For each edge $e=\{u,v\} \in E$ create nodes $u_e, v_e \in V'$ with $b(u_e)=b(v_e)=1$ and three edges $\{u', u_e\}$, $\{u_e, v_e\}$ and $\{v_e, v'\}$ and set $c'_{u'u_e} := c'_{v'v_e} := c_{uv}$ and $c_{u_e w_e} := 0$. Create a new node $v_k \in V'$ with $b(v_k) = 2 \cdot (n-k)$. Create for each $v \in V$ an edge $\{v_k, v'\}$ with $c'_{v_k v'} = 0$.
The graph $G'$ has $O\big(|E|\big)$ nodes and $O\big(|E|+n-k\big)$ edges and can be built in $O\big(|E|\big)$ operations. Moreover, it holds that $b(V') \leq 2n$. The $b$-matching problem on $G'$ can be solved in polynomial in $|E'|$ and $b(V')$ number of operations, which yields a strongly polynomial time algorithm for the simple $(k,2)$-matching problem. □

However, the algorithm given in the proof above can be hardly used in practice. Indeed, for a complete graph $G$ we need to solve a perfect matching problem on a graph $G'$ with $O(n^2)$ nodes, which can be a challenge even for rather small values of $n$.
An integer vector $x \in {\mathbf{Z}_+}^E$ is the incidence vector of a $k$-cycle if the following three conditions are satisfied:

1. $x$ is a simple $(k,2)$-matching,
2. $x$ is a perfect $(k,2)$-matching,
3. the support graph of $x$ is connected.

Now we can see that for a given weighted graph $G$ an optimal solution of the simple $(k,2)$-matching problem on $G$ gives a lower bound for the optimal objective value of the $k$-TSP on $G$. The simple $(k,2)$-matching problem is polynomial time solvable. However, we leave open the question how to design an efficient combinatorial algorithm for the simple $(k,2)$-matching problem and the question whether the perfect $(k,2)$-matching problem is polynomially solvable.

# 6 Examples

Here we show some small examples of applying these two bounding techniques for the $k$-TSP.
**Example 1.** Consider a complete graph with $4$ nodes which are points in a plane with coordinates (0,3), (3,3), (3,6), (2,0), see figure 3. Here we have $k=3$, the weights of the edges are euclidian length. The minimum weight
(3,1)-forest a) has the weight of $6+\sqrt{10}$, the minimum weight simple (3,2)-matching b) has the weight of $6+\sqrt{13}$, and the minimum weight 3-cycle c) has the weight of $6+\sqrt{18}$.
**Example 2.** A complete graph with $5$ nodes, all the same as in the first example plus one point (5,5) and $k=4$, see figure 4. The minimum weight (4,1)-forest a) has the weight of $6+\sqrt{5}+2\sqrt{2}$, the minimum weight simple (4,2)-matching b) has the weight of $3+\sqrt{13}+\sqrt{5}+2\sqrt{2}$, and the minimum weight 4-cycle c) has the weight of $3+\sqrt{18}+\sqrt{5}+2\sqrt{2}$.

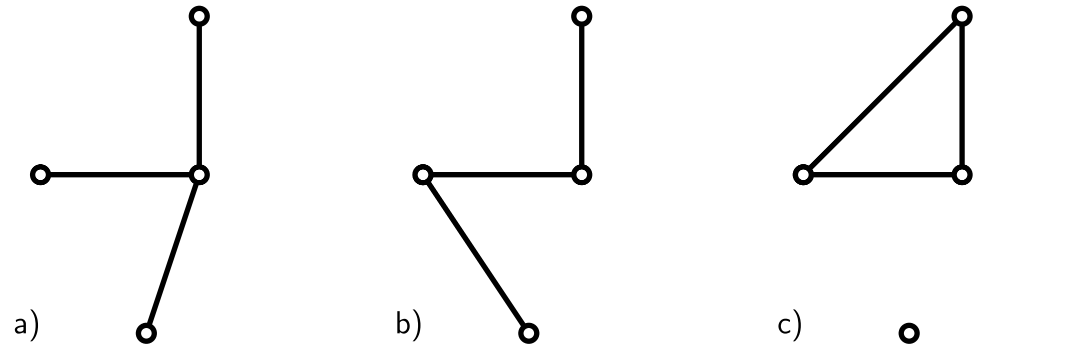


Figure 3: Minimum weight a) (3,1)-forest, b) simple (3,2)-matching, and c) 3-cycle

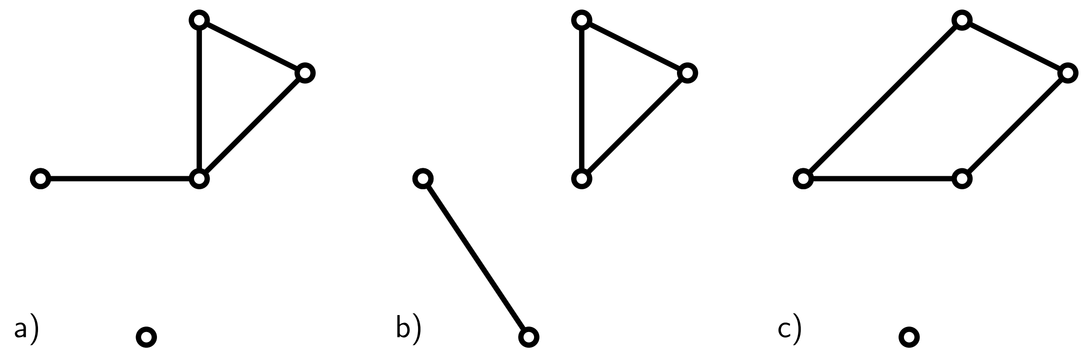


Figure 4: Minimum weight a) (4,1)-forest, b) simple (4,2)-matching, and c) 4-cycle

**Acknolegment.** The author is thankful to Andreas Drexl for his useful suggestions and comments.

# References


[1] C. Berge, *The Theory of Graphs and its Applications*, John Wiley, New York, 1962.

[2] U. Derigs, *Programming in Networks and Graphs*, Springer-Verlag, 1988.

[3] J. Edmonds, *Maximum matching and a polyhedron with 0,1-vertices*, Journal of Research of the National Bureau of Standards Section B **69** (1965), 67–72.

[4] E. Girlich, M. Höding, A. Horbach, and M. Kovalev, *On the diameter of the circuit and the $k$-cycle polytopes*, Technical Report 7, Faculty of Mathematics, Otto-von-Guericke University Magdeburg, 2002.

[5] ______, *Regular $\delta$-path inequalities for the $k$-cycle polytope*, Technical Report 15, Faculty of Mathematics, Otto-von-Guericke University Magdeburg, 2003.

[6] M. Hartmann and Ö. Özlük, *Facets of the $p$-cycle polytope*, Discrete Applied Mathematics **112** (2001), 147–178.

[7] M. Held and R.M. Karp, *The traveling-salesman problem and minimum spanning trees*, Operations Research **18** (1970), 1138–1162.

[8] ______, *The traveling-salesman problem and minimum spanning trees: part II*, Mathematical Programming **1** (1971), 6–25.

[9] J.B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical Society **7** (1956), 48–50.

[10] J.-F. Maurras, M.M. Kovalev, and Y. Vaxès, *On the convex hull of the 3-cycles of the complete graph*, Pesqui. Oper. **23** (2003), 99–109.

[11] J.-F. Maurras and V.H. Nguyen, *On the linear description of the $k$-cycle polytope*, International Transactions in Operational Research **8** (2001), 673–692.

[12] ______, *On the linear description of the 3-cycle polytope*, European Journal of Operational Research **137** (2002), 310–325.

[13] V. H. Nguyen, *Polyèdres de cycles: description, composition et lifting de facettes*, Ph.D. thesis, Université de la Méditerranée, 2000.

[14] A. Schrijver, *Combinatorial Optimization – Polyhedra and Efficiency*, Springer, 2003.

[15] W.T. Tutte, *A short proof of the factor theorem for finite graphs*, Canadian Journal of Mathematics **6** (1954), 347–352.