A Semidefinite Optimization Approach to the Target Visitation Problem

P. Hungerländer
Alpen-Adria Universität Klagenfurt
Institut für Mathematik
9020 Klagenfurt, Austria
philipp.hungerlaender@uni-klu.ac.at

August 17, 2014

Abstract

We propose an exact algorithm for the Target Visitation Problem (TVP). The (TVP) is a composition of the Linear Ordering Problem and the Traveling Salesman Problem. It has several military and non-military applications, where two important, often competing factors are the overall distance traveled (e.g. by an unmanned aerial vehicle) and the visiting sequence of the various targets or points of interest. Hence our algorithm can be used to find the optimal visiting sequence of various pre-determined targets.

First we show that the (TVP) is a special Quadratic Position Problem. Building on this finding we propose an exact semidefinite optimization approach to tackle the (TVP) and finally demonstrate its efficiency on a variety of benchmark instances with up to 50 targets.

Keywords: Target Visitation Problem; Linear Ordering Problem; Traveling Salesman Problem; Semidefinite Programming; Global Optimization

1 Introduction

The Target Visitation Problem (TVP) was suggested by Grundel and Jeffcoat [20] in 2004. It is a composition of the Linear Ordering Problem (LOP) and the Traveling Salesman Problem (TSP). Hence let us briefly review the most important theoretical and practical aspects of the (LOP), the (TSP) and the (TVP).

The Linear Ordering Problem (LOP)**.** Ordering problems associate to each ordering (or permutation) of the set $[n] := \{1, 2, \dots, n\}$ a profit and the goal is to find an ordering of maximum profit. In the simplest case of the Linear Ordering Problem (LOP), this profit is determined by those pairs $(u, v) \in [n] \times [n]$, where u comes before v in the ordering. Thus in its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that $\sum_{\substack{i,j \in [n] \\ i<j}} w_{\pi(i),\pi(j)}$, is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of nodes i and j either arc (i, j) or arc (j, i), but not both. Then the (LOP) consists of finding an acyclic tournament, i.e. a tournament without directed cycles, of G of maximum total edge weight. We refer to the book by Martí and Reinelt [36] and the references therein for further material on the (LOP), its variants and various applications and details on many heuristic and exact methods.

The (LOP) is equivalent to the Acyclic Subdigraph Problem (ASP) and the Feedback Arc Set Problem (FASP). It is well known to be NP-hard [18] and it is even NP-hard to approximate (LOP) within the factor $\frac{65}{66}$ [40]. Surprisingly there is not much known about heuristics with approximation guarantees. If all entries of W are nonnegative, a $\frac{1}{2}$-approximation is trivial, but no better polynomial time approximation is known. To narrow this quite large gap $[\frac{1}{2}, \frac{65}{66}]$ is a challenging open problem. Some worthwhile steps have already been taken into that direction: Newman and Vempala [40] showed that widely-studied polyhedral relaxations for the (LOP) cannot be used to approximate the problem within a factor better than $\frac{1}{2}$. Furthermore Newman [39] analyzed a semidefinite programming (SDP) relaxation based on position variables that has an integrality gap of 1.64 (hence smaller than 2) on certain random graphs. Note that this SDP relaxation is in fact a rudimentary version (not suited for reasonable practical bound computations) of the SDP relaxations that we will propose to obtain optimal tours and strong upper bounds for large (TVP) instances in this paper.

SDP is the extension of linear programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\inf_X \{ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \qquad \texttt{(SDP)}$$
$$\mathcal{P} := \{ X \mid \langle A_i, X \rangle = b_i, \ i \in \{1, \dots, m\}, \ X \succcurlyeq 0 \},$$

where the data matrices $A_i, \ i \in \{1, \dots, m\}$ and C are symmetric. For further information on SDP we refer to the handbooks [2, 46] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

The (LOP) arises in a large number of applications in such diverse fields as economy (ranking and voting problems [31, 44] and input-output analysis [8, 25, 33]), sociology (determination of ancestry relationships [19]), graph drawing (one sided crossing minimization [29]), archaeology, scheduling (with precedences [7]), assessment of corruption perception [1] and ranking in sports tournaments. Additionally problems in the context of mathematical psychology and the theory of social choice can be formulated as linear ordering problems, see [17] for a survey.

There are also several problems that are closely related to the (LOP) like the (LOP) with cumulative costs [6, 15] that has a very interesting application in the area of mobile phone telecommunication and the Coupled Task Problem [5] that is concerned with scheduling n jobs each of which consists of two subtasks with associated required delays.

The Coupled Task Problem is concerned with scheduling n jobs each of which consists of two subtasks. Furthermore there is a requirement that between the execution of these subtasks a delay is required. We refer to [5] for an optimization model for this problem that successfully uses linear ordering variables together with additional constraints for modelling the processing times and delays properly.

The current state-of-the-art exact algorithm for the (LOP) is a Integer Linear Programming (ILP) Branch-and-Cut approach that was developed by the working group of Reinelt in Heidelberg and is based on sophisticated cut generation procedures (for details see [36]). It can solve large instances from specific instance classes with up to 150 objects, while it fails on other much smaller instances with only 50 objects. Hungerländer and Rendl [27] proposed an SDP approach (based on products of ordering variables) that proved to be a valuable alternative to the ILP approach for larger and/or notoriously difficult instances. There also exist many heuristics and methaheuristics for the (LOP) and some of them are quite good in finding the optimal solution for large instances in reasonable time. For a recent survey and comparison see [37].

Traveling Salesman Problem (TSP). The (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible tour that visits each city exactly once and returns to the origin city? The NP-hard (TSP) is doubtless the most famous of all (combinatorial) optimization problems with high importance in both operations research and theoretical computer science. We refer to the books [11, 21, 42] and the references therein for extensive material on the (TSP), its variants and various applications, details on many heuristic and exact methods and relevant theoretical results.

The Christofidess algorithm approximates the cost of an optimal symmetric (TSP) tour within the factor 1.5 [10]. In the asymmetric case paths may not exist in both directions or the distances might be different, forming a directed graph. This may be e.g. due to traffic collisions, one-way streets and motorways. In the asymmetric, metric case, only logarithmic performance guarantees are known. The best current algorithm achieves performance ratio $0.814 \log(n)$ [30]. It is an open question if a constant factor approximation exists. Results on the even more difficult non Euclidean (TSP) are e.g. discussed in [41].

The (TSP) has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, e.g. in DNA sequencing. In many further applications the (TSP) with additional constraints, such as limited resources or time windows, is of relevance.

Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved completely[1] and even problems with millions of cities

[1] The Branch-and-Cut algorithm by Applegate et al. [3] holds the current record, solving an instance with 85,900 cities.

can be approximated within a small fraction of 1%.

Target Visitation Problem. The (TVP) asks for a permutation $(p_1, p_2, \ldots, p_n)$ of n targets with given pairwise weights $w_{ij},\ i,j \in [n], i \neq j$, and pairwise distances $d_{ij},\ i,j \in [n], i \neq j$, maximizing the objective function

$$\sum_{\substack{i,j\in[n]\\ i<j}} w_{p_i,p_j} - \left(\sum_{i=1}^{n-1} d_{p_i,p_{i+1}} + d_{p_n,,p_1} \right).$$

As the NP-hard (LOP) and (TSP) are special cases of the (TVP), the (TVP) is also NP-hard.

The formulation of the (TVP) was inspired by the use of single unmanned aerial vehicles (UAVs) that have been used increasingly (especially for military purposes) over the last decades. Civilian applications of the (TVP) include environmental assessment, combat search and rescue and disaster relief [20]. In all military and non-military applications two important, often competing factors are the overall distance traveled by the UAV and the visiting sequence of the various targets or points of interest. E.g. in military applications we aim to visit high chance waypoints quickly, such that the coalition force may act on the intelligence they receive. Furthermore optimal (TVP) solutions could be of use for cooperative systems [38], where multiple dynamic entities (e.g. UAVs) share information or tasks to accomplish a common, though perhaps not singular, objective.

Despite there exist several relevant applications of the (TVP), only few methods have been suggested for obtaining good or optimal tours. There exist two heuristics for the (TVP): a very simple one proposed in [20] and a genetic algorithm by Arulselvan et al. [4] that was tested on (TVP) instances with up to 16 targets. Hildenbrandt et al. [24] conducted the first polyhedral study of the (TVP) polytope. They present several possible IP-models for the (TVP) and compare them to their usability for branch-and-cut approaches. Based on their findings Hildenbrandt et al. [23] are currently developing an exact IP approach for the (TVP) that has a strong potential to solve large-scale (TVP) instances to optimality.

Now let us give an easy integer programming (IP) formulation of the (TVP) appropriately combining the standard IP formulations of the (LOP) and the (TSP): First we introduce traveling salesman and linear ordering variables:

$$x_{i,j} := \begin{cases} 1, & \text{if target } j \text{ is visited immediately after target } i, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_{i,j} := \begin{cases} 1 & \text{if target } i \text{ is visited (for the first time)}^2 \text{ before target } j \text{ is visited for the first time,} \\ 0, & \text{otherwise.} \end{cases}$$

And now we can already formulate the (TVP) as the following IP:

$$\max \sum_{\substack{i,j\in[n]\\ i\neq j}} w_{ij} y_{ij} - \sum_{\substack{i,j\in[n]\\ i\neq j}} d_{ij} x_{i,j} \tag{1}$$

subject to:

$$\sum_{\substack{j=1\\ j\neq i}}^{n} x_{j,i} = 1, \qquad \sum_{\substack{j=1\\ j\neq i}}^{n} x_{i,j} = 1, \quad i \in [n], \tag{2}$$

$$\sum_{\substack{i,j\in S\\ i\neq j}} x_{i,j} \leq |S| - 1, \quad \forall S \subset V,\ 2 \leq |S| \leq n, \tag{3}$$

$$y_{j,i} = 1 - y_{i,j}, \quad i,j \in [n],\ i < j \tag{4}$$

$$0 \leq y_{i,j} + y_{j,k} - y_{i,k} \leq 1,\ i,j,k \in [n],\ i < j < k, \tag{5}$$

$$x_{i,j} - y_{i,j} - \frac{1}{n-1} \sum_{\substack{k=1\\ k\neq i}}^{n} y_{k,i} \leq 0, \quad i,j \in [n],\ i \neq j, \tag{6}$$

$$x_{i,j} \in \{0,1\}, \quad y_{i,j} \in \{0,1\}, \quad i,j \in [n],\ i \neq j.$$

[2] The first target is visited twice such that the tour is closed.

Constraints (2) and (3) are the standard constraints for the asymmetric (TSP) and (4) and (5) are typically used to model the (LOP). Inequalities (6) connect the two problems, where the additional term $\frac{1}{n-1}\sum_{\substack{k=1\\k\neq i}}^{n} y_{k,i}$ makes sure that possible edges from location n to location 1 are allowed. Finally the integrality conditions ensure that the variables used are binary. Hence the above optimization problem is obviously an IP formulation of the (TVP). If we replace the integrality conditions by the bound constraints

$$0 \leq x_{i,j} \leq 1, \quad 0 \leq y_{i,j} \leq 1, \quad i,j \in [n],\ i \neq j, \tag{7}$$

we obtain a basic linear programming relaxation for the (TVP) that we denote by $(\mathrm{LP}_{\mathrm{TVP}})$.

Note that in the previous papers on the (TVP) [4, 20] the starting (and concurrently final) point of the tour was fixed and denoted as base. In this paper we suggest a more general version of the (TVP) where the starting point of the tour is not fixed in general. Nonetheless we can easily introduce a base by special preference settings, i.e. we can fix one target implicitly (through the input data) to be the first "target" (base) in the ordering by choosing the linear ordering weights in the row corresponding to the base high enough.

Toy Example. Next we want to further clarify the workings of the (TVP) with the help of a toy example. We consider 5 targets, where we set target 1 to be the base and hence the first in the ordering. We are given the following (LOP) weights W for the remaining 4 targets and the (TSP) distances D between all five targets:

$$W = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 3 & 5 & 5 & 3 \\ 3 & 0 & 3 & 5 & 5 \\ 5 & 3 & 0 & 3 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 3 & 5 & 5 & 3 & 0 \end{bmatrix} \tag{8}$$

Figure 1 illustrates the optimal (TSP) tours a.) and b.), the optimal (LOP) solution c.) and the optimal (TVP) tour d.) (the tours are displayed by grey edges) together with their corresponding (LOP) benefits (red edges and numbers) and (TSP) costs (grey numbers). Hence considering travel distances and target preferences simultaneously leads to optimal tours that can be quite different from the optimal (TSP) and (LOP) solutions.

The above toy example can also be enhanced by a motivational story that we partly adopt from [4]. Suppose that the targets represent a collection of villages in which a sought after terrorist is suspected of hiding out. The operation base is the location of the coalition force. Moreover assume the available intelligence data determining target preferences is summarized in W from (8). Such data can e.g. be based on hiding probabilities of the terrorist or the size of communication networks between the villages. In the application described the person of interest moves frequently and hence the intelligence date becomes less accurate quite fast. Now suppose the coalition force has the ability to launch a UAV that visits the targets in a pre-determined tour and returns to the base. During its flight, the UAV is capable of telemetering data back to the coalition force helping to establish the known location of the terrorist they seek. In summary already this simple motivational story demonstrates the importance of considering both distance and visitation sequence when looking for an optimal tour in various (military) applications.

Outline. The main contributions of this paper are the following:

- We define the (TVP) as a combination of the asymmetric, non-Euclidean (TSP) and the (LOP).
- Then we show that the (TVP) is a special Quadratic Position Problem (QPP).
- Building on this finding we propose an exact semidefinite optimization approach to tackle the (TVP).
- We showcase in a computational study that our algorithmic approach yields very promising results on a large variety of benchmark instances with up to 50 targets.

We refer to the companion paper [26] for a more detailed analysis of the (QPP), its polytope and some theoretical properties of our semidefinite relaxations for the (LOP) and the (TSP) respectively.[3] Herein, we are mainly interested in applying our SDP approach to the (TVP) and its associated military applications.

[3] We note that the content of the companion paper is fairly disjoint from this paper: it deals with the (QPP) from a polyhedral point of view and applies the resulting SDP relaxations to facet defining inequalities of small (TSP) and (LOP) polytopes to facilitate the theoretical analysis of the relaxations proposed. In the companion paper we do not consider the (TVP) nor conduct any large-scale computations. Due to space restrictions we omit the proofs concerning the (QPP) and refer to our companion paper for details.

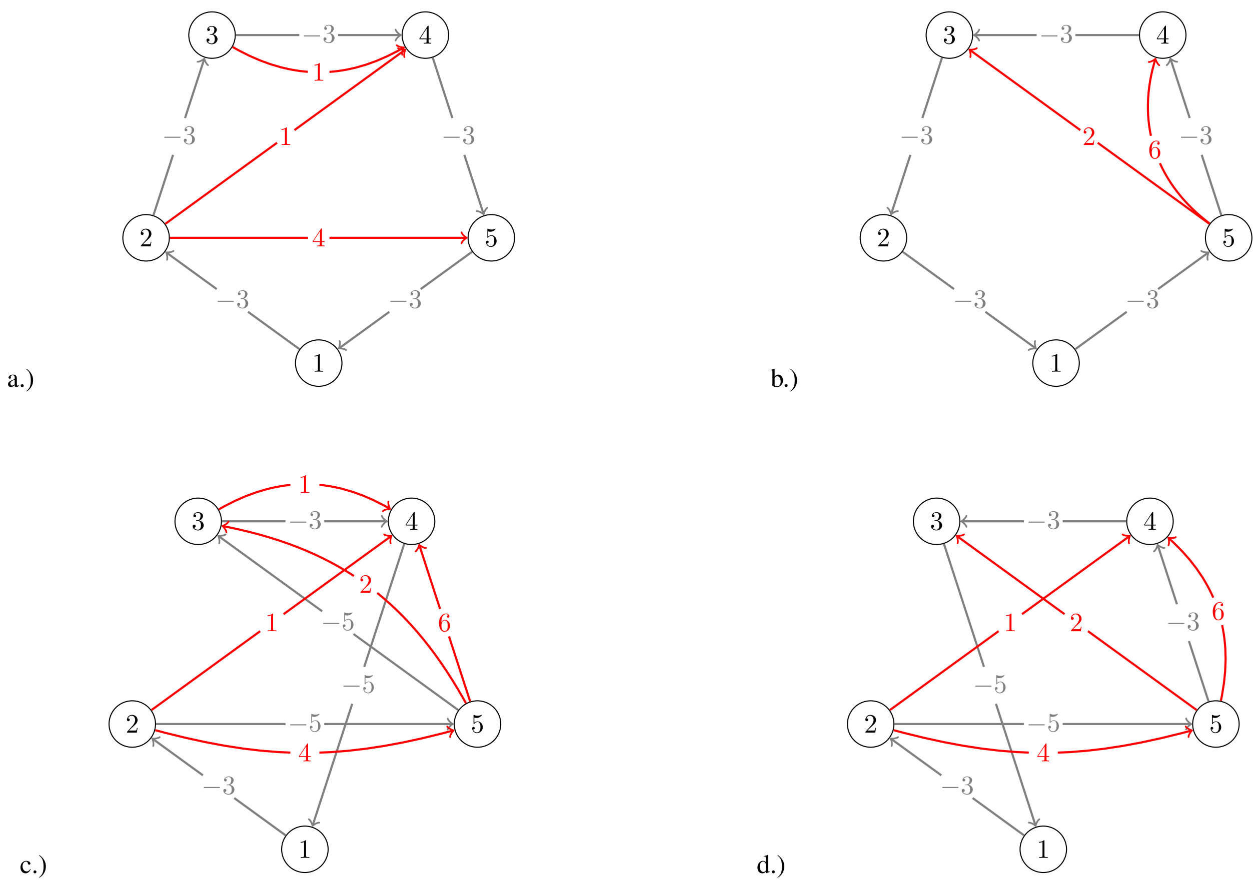

Figure 1: We are given 5 targets, where target 1 is the operation base. The input data is provided in (8). In a.) and b.) we display the optimal (TSP) tours with objective value -15 and corresponding (TVP) objective values -9 and -7 respectively. In c.) we depict the optimal (LOP) solution with objective value 14 and associated (TVP) objective value -7. Finally in d.) we display the optimal (TVP) tour with corresponding objective value -6.

The paper is structured as follows. In Section 2 we define and discuss the (QPP) and then propose a matrix-based formulation and several semidefinite relaxations for the (TVP). In Section 3 we explain how to solve our SDP relaxations for (TVP) instances and describe a simple two-opt improvement heuristic for obtaining feasible tours with a high (TVP) objective value. In Section 4 we show first results for our SDP relaxations on small instances. Based on this comparison, we choose the (SDP) relaxation best suited for tackling large scale instances and present associated, promising results in Section 5. Section 6 concludes the paper.

2 A Matrix-Based Formulation and Semidefinite Relaxations of the Target Visitation Problem

SDP methods proved to be a serious alternative to Branch & Cut approaches for ordering problems [9, 27, 28] and also yield interesting theoretical results for the (TSP) [12, 13]. As the (TSP) variables cannot be expressed as linear-quadratic terms in (LOP) variables and vice versa, we are looking for an alternative matrix-based formulation of the (TVP) to avoid lifting a vector with all (LOP) and (TSP) variables. Such a formulation can be obtained by encoding tours with the help of position variables. We will introduce these variables and the corresponding Quadratic Position Problem (QPP) in the following subsection. In Subsection 2.2 we propose several semidefinite relaxations of the (QPP). Finally in Subsection 2.3 we show that the (TVP) is a special (QPP) and hence we can use the SDP relaxations of the (QPP) to obtain upper bounds to the optimal solution value of (TVPs).

2.1 The Quadratic Position Problem (QPP)

Let a vector v of size $n(n-1)$ that contains bivalent position variables be given. We can write the components $v_{(i-1)n+j},\ i \in [n],\ j \in [n-1]$ of v more compactly as

$$v_i^j \in \{-1, 1\}, \quad i \in [n],\ j \in [n-1]. \tag{9}$$

With the help of these variables we model the position of object i in an ordering of n objects:

$$v_i^j - v_i^{j-1} = \begin{cases} 2, & \text{if object } i \text{ is assigned to location } j, \\ 0, & \text{if object } i \text{ is not assigned to location } j, \end{cases} \tag{10}$$

where we additionally use the parameters

$$v_i^0 := -1, \quad v_i^n := 1, \quad i \in [n]. \tag{11}$$

To further clarify this definition, we encode the tours displayed in our toy example (see Figure 1) in v, where we separate variables associated to different objects by a vertical dash $|$:

$$\begin{array}{ll}
a.) & v = \big(-1 \ 1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ 1 \ 1 \mid \\
b.) & v = \big(-1 \ 1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ -1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ 1 \ 1 \ 1 \mid \\
c.) & v = \big(-1 \ 1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ -1 \ 1 \mid \\
d.) & v = \big(-1 \ 1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ 1 \ 1 \ 1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ -1 \ 1 \mid -1 \ -1 \ -1 \ -1 \ 1 \ 1 \mid
\end{array} \tag{12}$$

Let us now propose a (new) combinatorial optimization problem corresponding to the position variables above. An instance of the (QPP) consists of n objects, n consecutive locations and individual and pairwise integer benefits $b_i^k,\ i, k \in [n]$, and $b_{i,j}^{k,l},\ i, j, k, l \in [n],\ i < j,\ k \neq l$. The optimization problem can be written down as

$$\max_{\pi \in \Pi_n} \sum_{\substack{i,j,k,l \in [n] \\ i<j,\ k \neq l}} \left(b_i^k w_i^k(\pi) + b_{i,j}^{k,l} w_i^k(\pi) w_j^l(\pi) \right), \tag{13}$$

where Π_n is the set of permutations of the indices $[n]$, defining an assignment of the n objects to the n consecutive locations and $w_i^k(\pi),\ i, k \in [n]$ is 1, iff object i is assigned to location k in the particular assignment $\pi \in \Pi_n$.[4] Note that the Quadratic Assignment Problem (QAP) [32, 34] can be formulated as a (QPP) as the quadratic assignment variables $q_i^j,\ i, j \in [n]$ can be written as a difference of position variables: $2q_i^j = v_i^j - v_i^{j-1}$.

To model the (QPP) with the help of the bivalent position variables v introduced above, we ask for monotonicity of variables belonging to the same object:

$$v_i^j \leq v_i^k, \quad i \in [n],\ j, k \in [n-1],\ j < k. \tag{14}$$

Additionally we have to assure that a switch from -1 to 1 occurs for different objects at different positions:

$$\sum_{i=1}^{n} v_i^j = 2j - n, \quad j \in [n-1]. \tag{15}$$

These two types of constraints already suffice to ensure that all objects are assigned to different locations:

Lemma 1. *The constraints* (14) *and* (15) *form, together with the integrality conditions* (9)*, a minimal constraint system for modeling the* (QPP) *using* $n(n-1)$ *position variables.*

In summary we are able to rewrite (13) as a a mathematical model in v with linear constraints and a quadratic objective function, where we summarize the individual benefits in a vector b and the pairwise benefits in a matrix B:

[4] If the benefits are negative, the associated optimization problem in fact minimizes the total costs over all assignments.

Theorem 2. *Maximizing $v^\top Bv + v^\top b$ over $v \in \{-1,1\}^{n(n-1)}$,* (14) *and* (15) *solves the* (QPP).

By replacing the integrality conditions above with $[-1,1]$ bounds we obtain a quadratic programming relaxation of the (QPP). Next we rewrite the objective function in terms of matrices to obtain a matrix-based formulation of the (QPP)

$$\max\left\{\langle C,Z\rangle \;:\; v \in \{-1,1\}^{n(n-1)},\; v \text{ satisfies (14) and (15)} \right\}, \tag{16}$$

where all position variables and their products are contained in the $(n^2-n+1)\times(n^2-n+1)$ variable matrix $Z := \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix}$ with $V = vv^\top$ and the cost matrix C is given by $C := \begin{pmatrix} 0 & b^\top \\ b & B \end{pmatrix}$.

Finally we can further rewrite the above matrix-based formulation as an SDP, where we denote by e the vector of all ones and by $\mathcal{E}$ the elliptope

$$\mathcal{E} := \{\, Z \;:\; \operatorname{diag}(Z) = e,\; Z \succcurlyeq 0 \,\}.$$

Theorem 3. *The semidefinite optimization problem*

$$\max\;\left\{\; \langle C,Z\rangle : Z \textit{ satisfies } (14) \textit{ and } (15)\;,\; Z \in \mathcal{E},\; v \in \{-1,1\}^{n(n-1)} \;\right\}, \tag{SDP−QPP}$$

is equivalent to the (QPP).

Next we show how to construct several SDP relaxations from the SDP formulation of the (QPP) proposed above.

2.2 Semidefinite Relaxations of the Quadratic Position Problem

The above formulation of the (QPP) contains $n-1$ equalities stated in (15). We can use these equations to eliminate $n-1$ position variables. Such a reduction is irrelevant for the formulation but could matter for the values of relaxations of the (QPP). Hence we want to clarify, if we maybe get stronger semidefinite relaxations by working with matrices of order $n(n-1)+1$ or $(n-1)^2+1$? To answer this question, let us recall a result from [27]:

Theorem 4. *[27, Remark 2] Let m linear equality constraints $Ay = c$ be given. If there exists some invertible $m \times m$ matrix B such that we can partition the linear system in the following way*

$$Ay = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} t \\ u \end{bmatrix} = c. \tag{17}$$

Then we do not weaken the relaxation by first moving into the subspace given by the equations, and then lifting the problem to matrix space.

In other words, it is equivalent in terms of tightness to eliminate m variables or to lift the m equality constraints in all possible ways to quadratic space. Hence we decide to eliminate $n-1$ variables through (15) to reduce the number of variables to $(n-1)^2$ and to avoid additional constraint classes. Of course, this decision has also a disadvantage: We have to consider two versions for several constraint types, namely the cases where object n is considered and not considered.

Dropping the integrality condition on v in (SDP−QPP) and reducing the problem dimension with the help of (15), we obtain the following basic semidefinite relaxation of the (QPP)

$$\max\left\{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (14)},\; Z_s \in \mathcal{E}\;\right\}, \tag{SDP$_0$}$$

where the cost and variable matrices with index s consist of the first $(n-1)^2+1$ rows and columns of their larger counterparts from the previous subsection: $C_s = C_{1:(n-1)^2+1},\; Z_s = Z_{1:(n-1)^2+1}$. Note that the equality constraints (15) are implicitly assured by the above semidefinite relaxation. In general (SDP$_0$) gives quite weak upper bounds to the optimal solution value of the (QPP). Hence we will suggest several ways to improve on the tightness of (SDP$_0$).

First we propose $n(n-1)(n-2)$ valid equalities for the (QPP) polytope

$$\mathcal{P}_{\mathtt{QPP}} := \operatorname{conv}\left\{\; Z_s \;:\; Z_s = \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix},\; v \in \{-1,1\}^{(n-1)^2} \text{ satisfies (14)},\; V = vv^\top \;\right\},$$

and show that their rank is $n(n-1)(n-2)-1$, i.e. their number minus 1. The following equalities, consisting of 5 different types, can be easily deduced by exploiting the structure of v induced by (14) and (15):

$$v_i^j - v_i^k - v_{i,i}^{j,k} = -1, \quad i,j,k \in [n-1],\ j<k, \tag{18}$$

$$m_k \sum_{i=1}^{n-1} v_i^j + m_j \sum_{i=1}^{n-1} v_i^k - \sum_{h=1}^{n-1}\sum_{i=1}^{n-1} v_{h,i}^{j,k} = m_k m_j, \quad m_k = 2k-n-1,\ m_j = 2j-n+1,\ j,k \in [n-1], j<k, \tag{19}$$

$$v_i^1 + v_j^1 + v_{i,j}^{1,1} = -1, \quad i,j \in [n-1],\ i<j, \tag{20}$$

$$v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} = 1, \quad i,j \in [n-1],\ i<j, \tag{21}$$

$$v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k} = 0, \quad i,j \in [n-1],\ i<j,\ k \in [n-1],\ k \neq 1. \tag{22}$$

Lemma 5. *The* $n(n-1)(n-2)$ *equalities* (18) – (22) *are valid for* $\mathcal{P}_{\mathtt{QPP}}$ *and have rank* $n(n-1)(n-2)-1$.

We can also show that the the equalities (18) suffice together with the implicitly assured linear constraints (15) to ensure monotonicity in the variable vector for all objects.

Lemma 6. *The monotonicity constraints* (14) *are assured by the equalities* (18) *and* (15) *together with* $Z_s \in \mathcal{E}$.

In summary $(\mathtt{SDP}_0)$ can be tightened by adding the equalities analyzed above that implicitly ensure (14):

$$\max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (18) – (22)},\ Z_s \in \mathcal{E}\ \}, \tag{$\mathtt{SDP}_1$}$$

Now we can further improve the relaxation strength of $(\mathtt{SDP}_1)$ by adding several types of inequalities valid for $\mathcal{P}_{\mathtt{QPP}}$. First we discuss inequalities obtained by exploiting the structure of v induced by (14) and (15). Secondly we suggest valid inequalities associated to the integrality conditions $v \in \{-1,1\}^{(n-1)^2}$.

Lemma 7. *The following inequality constraints hold for* $Z_s \in \mathcal{P}_{\mathtt{QPP}}$*:*

$$-v_{i,j}^{h,m} - v_{i,j}^{g,l} + v_{i,j}^{g,m} + v_{i,j}^{h,l} \leq 0, \quad i,j \in [n-1],\ g,h,l,m \in [n],\ g<h,\ l<m, \tag{23}$$

$$2(m-l)\left(v_i^g - v_i^h\right) + \sum_{j=1}^{n-1}\left(v_{i,j}^{h,m} + v_{i,j}^{g,l} - v_{i,j}^{g,m} - v_{i,j}^{h,l}\right) \leq 0,\ \ i \in [n-1],\ g,h,l,m \in [n],\ g<h,\ l<m. \tag{24}$$

In fact the inequalities (23) and (24) are linear combinations of the following smaller set of constraints.

Lemma 8. *Inequalities* (23) *and* (24) *are assured by*

$$-v_{i,j}^{h,l} - v_{i,j}^{h-1,l-1} + v_{i,j}^{h-1,l} + v_{i,j}^{h,l-1} \leq 0, \quad i,j \in [n-1],\ i<j,\ h,l \in [n]. \tag{25}$$

$$-2v_i^h + 2v_i^{h-1} + \sum_{j=1}^{n-1}\left(v_{i,j}^{h,l} + v_{i,j}^{h-1,l-1} - v_{i,j}^{h-1,l} - v_{i,j}^{h,l-1}\right) \leq 0, \quad i \in [n-1],\ h,l \in [n]. \tag{26}$$

Adding the inequalities discussed above to $(\mathtt{SDP}_1)$ yields the following stronger semidefinite relaxation:

$$\max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (18) – (22), (25) and (26)},\ Z_s \in \mathcal{E}\ \}, \tag{$\mathtt{SDP}_2$}$$

There exists another obvious way to tighten our semidefinite relaxations. As $Z_s \in \mathcal{P}_{\mathtt{QPP}}$ is actually a matrix with $\{-1,1\}$ entries, it satisfies the triangle inequalities defining the metric polytope that are known to be facet-defining for the cut polytope, see e.g. [14]:

$$\mathcal{M} = \left\{ Z_s\ :\ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Z_{i,j} \\ Z_{j,k} \\ Z_{i,k} \end{pmatrix} \leq e, \quad 1 \leq i<j<k \leq (n-1)^2+1 \right\}. \tag{27}$$

Hence we can improve (SDP$_1$) and (SDP$_2$) by asking in addition that $Z_s \in \mathcal{M}$, which yields the following two tractable relaxation of $\mathcal{P}_{\texttt{QPP}}$:

$$\max \left\{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (18)} - \text{(22)},\ Z_s \in (\mathcal{E} \cap \mathcal{M}) \right\}, \qquad (\texttt{SDP}_3)$$

$$\max \left\{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (18)} - \text{(22), (25) and (26)},\ Z_s \in (\mathcal{E} \cap \mathcal{M}) \right\}. \qquad (\texttt{SDP}_4)$$

Already the semidefinite relaxation (SDP$_3$) implicitly implies the monotonicity constraints lifted to quadratic space with the help of an approach suggested by Lovász and Schrijver [35]: Applying their approach to our problem suggests to multiply the monotonicity constraints (14) by the nonnegative expressions $(1 - v_h^l)$ and $(1 + v_h^l)$ which gives

$$v_i^j - v_{i,h}^{j,l} - v_i^k + v_{i,h}^{k,l} \leq 0, \qquad v_i^j + v_{i,h}^{j,l} - v_i^k - v_{i,h}^{k,l} \leq 0, \quad i,j,k,h,l \in [n-1],\ j < k, \tag{28}$$

and correspondent inequalities if object n is involved.

Lemma 9. *The lifted monotonicity constraints* (28) *are assured by the equalities* (18) *together with* $Z_s \in \mathcal{M}$.

2.3 The Target Visitation Problem as a Special Quadratic Position Problem

In the following we explain how to model the (TVP) as a special (QPP): First we show that the (LOP) and the (TSP) variables can be expressed as products of position variables and then we use this property to reformulate (LP$_{\texttt{TVP}}$) as a semidefinite optimization problem in position variables that yields provably stronger upper bounds than (LP$_{\texttt{TVP}}$).

Lemma 10. *We can express the traveling salesman variables and the linear ordering variables as linear-quadratic terms in position variables:*

$$x_{i,j} = \frac{1}{4}\Bigg[1 - v_{i,j}^{1,1} + v_{i,j}^{1,2} + v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k} + v_{i,j}^{k-1,k}\right)$$
$$-v_i^{n-2} + v_{i,j}^{n-2,n-1} - v_{i,j}^{n-1,n-1} - v_{i,j}^{n-1,1}\Bigg], \quad i,j \in [n-1],\ i \neq j, \tag{29}$$

$$x_{i,n} = \frac{1}{4}\Bigg[5 - n + \sum_{j=1}^{n-1}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2} - v_j^2\right) + \sum_{k=2}^{n-2}\sum_{j=1}^{n-1}\Big(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k} - v_{i,j}^{k,k+1}$$
$$+v_{i,j}^{k,k}\Big) + (n-1)v_i^{n-2} + \sum_{j=1}^{n-1}\left(-v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad i \in [n-1], \tag{30}$$

$$x_{n,j} = \frac{1}{4}\Bigg[5 - n + \sum_{i=1}^{n-1}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2}\right) - (n-1)v_j^2 + \sum_{k=2}^{n-2}\sum_{i=1}^{n-1}\Big(v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k}$$
$$+v_{i,j}^{k,k}\Big) + \sum_{i=1}^{n-1}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad j \in [n-1], \tag{31}$$

$$y_{i,j} = \frac{1}{4}\left[1 - v_j^1 + \sum_{k=2}^{n-1}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k-1}\right) + v_i^{n-1} - v_{i,j}^{n-1,n-1}\right], \quad i,j \in [n-1],\ i \neq j, \tag{32}$$

$$y_{i,n} = \frac{1}{4}\left[n - 1 + \sum_{j=1}^{n-1} v_j^1 + \sum_{k=2}^{n-1}\left(2v_i^{k-1} + \sum_{j=1}^{n-1}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\right) - (n-3)v_i^{n-1} + \sum_{j=1}^{n-1} v_{i,j}^{n-1,n-1}\right],\ i \in [n-1]. \tag{33}$$

$$y_{n,j} = \frac{1}{4}\left[n - 1 + (n-3)v_j^1 + \sum_{k=2}^{n-1}\left(-2v_j^k + \sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\right) - \sum_{i=1}^{n-1}\left(v_i^{n-1} - v_{i,j}^{n-1,n-1}\right)\right],\ j \in [n-1]. \tag{34}$$

Proof. Using $V = vv^\top$ and (11) in (29) yields

$$x_{i,j} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) + \left(1 - v_i^{n-1}\right)\left(v_j^1 + 1\right)\right], \quad i,j \in [n-1],\ i \neq j.$$

Due to (14) $v_i^k - v_i^{k-1} \geq 0,\ i \in [n-1]$, and $v_j^{k+1} - v_j^k \geq 0,\ j \in [n-1]$, hold and $\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) = 1$ iff target i is assigned to the precedent location of target j or target i is assigned to location n and target j to location 1.

Using $V = vv^\top$, (11) and (15) in (30) and (31) respectively gives

$$x_{i,n} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_i^k - v_i^{k-1}\right)\left(v_n^{k+1} - v_n^k\right) + \left(1 - v_i^{n-1}\right)\left(v_n^1 + 1\right)\right], \quad i \in [n-1],$$

$$x_{n,i} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_n^k - v_n^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) + \left(1 - v_n^{n-1}\right)\left(v_j^1 + 1\right)\right], \quad j \in [n-1].$$

Analogically to the reasoning above, (14) ensures that both $x_{i,n}$ and $x_{n,j}$ are equal to 1, iff target n is assigned to the successive (first type) or the precedent (second type) location of target i and target j respectively.

Finally applying $V = vv^\top$, (11) and (15) to (32) and (33) yields

$$y_{i,j} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad i,j \in [n-1],\ i \neq j$$

$$y_{i,n} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_i^{k-1}\left(v_n^k - v_n^{k-1}\right)\right], \quad i \in [n-1], \quad y_{n,j} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_n^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad j \in [n-1].$$

Due to (14) $v_j^k - v_j^{k-1} \geq 0,\ j,k \in [n]$, holds and $v_j^k - v_j^{k-1} = 2$, iff target j is assigned to location k. Hence the term $v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j,k \in [n]$, is equal to 2, iff target j is assigned to location k and additionally target i is assigned to some location in front of location k. In summary $\sum_{k=1}^{n} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j \in [n],\ i < j$ is equal to 2, iff target j is assigned to location k and additionally target i is assigned to some location in front of location k, and equal to 0 otherwise. □ □

Corollary 11. *The* `(TVP)` *objective function* (1) *and the* `(TVP)` *constraints* (2) – (7) *can be reformulated as linear-quadratic expressions in position variables.*

Proof. On the one hand the `(TVP)` objective function and `(TVP)` constraints are linear expressions in the `(TSP)` variables $x_{i,j},\ i,j \in [n],\ i \neq j$ and the `(LOP)` variables $y_{i,j},\ i,j \in [n], i < j$, and on the other hand the `(TSP)` and the `(LOP)` variables can be expressed as linear-quadratic expressions in position variables due to Lemma 10. □ □

Remark 1. *We refrain from restating the objective function* (1) *and the constraints* (2) – (7) *as (partly quite long) linear-quadratic terms in position variables but of course we encoded all of them in our algorithm. But let us point out that the a bit complicated condition* (6) *can be reformulated in position variables in a very elegant way: For* $i,j \in [n-1],\ i \neq j$, *we just have to subtract the term* $1 + v_j^1 - v_i^{n-1} - v_{i,j}^{n-1,1}$ *from* (29) *to rule out the edges from location* n *to location* 1. *In this way we get rid of the term* $\frac{1}{n-1}\sum_{\substack{k=1\\k\neq i}}^{n} y_{ki}$. *In summary* (6) *can be reformulated as follows in position variables:*

$$v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k+1}\right) + v_{i,j}^{n-2,n-1} - v_i^{n-1} \leq 1, \quad i,j \in [n-1],\ i \neq j.$$

Note that these differences of `(TSP)` *and* `(LOP)` *variables result in easier expressions in position variables than the* `(TSP)` *variables. Similar simplifications also occur when rewriting* (6) *for* $i = n$ *and* $j = n$:

$$-\sum_{j=1}^{n-1} v_j^2 - 2\sum_{k=2}^{n-1} v_i^{k-1} + \sum_{k=2}^{n-2}\sum_{j=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + (n-1)v_i^{n-2} - \sum_{j=1}^{n-1} v_{i,j}^{n-2,n-1} \leq n-3, \quad i \in [n-1],$$

$$-(n-1)v_j^2 + 2\sum_{k=2}^{n-1} v_j^k + \sum_{k=2}^{n-2}\sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + \sum_{i=1}^{n-1}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1}\right) \leq n-3, \quad j \in [n-1].$$

Furthermore we can show that the equality constraints for both the (TSP) and the (LOP), reformulated as linear-quadratic expressions in position variables, are already satisfied by (SDP_1).

Lemma 12. *The equality constraints* (4) *for the* (LOP) *are assured by* (20) – (22).

Proof. For two fixed targets i and j with $i, j \in [n-1],\ i < j$, we combine (20) – (22) to obtain

$$2 - v_i^1 - v_j^1 - v_{i,j}^{1,1} + v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} + \sum_{k=2}^{n-1} \left(v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k} \right) = 4 \left(y_{i,j} + y_{j,i} \right).$$

Hence the constraints (4) not considering target n are linear combinations of the constraints (20) – (22).

Next we analyze the equalities (4) considering target n, i.e. our fixed targets are $i,\ i \in [n-1]$, and n. Adding up 4 times (33) and 4 times (34) yields

$$\begin{aligned} 4 \left(y_{i,n} + y_{n,i} \right) = 2 \left(n-1 \right) + (n-1) v_i^1 + \sum_{j=1}^{n-1} v_j^1 + \sum_{j=1}^{n-1} v_{i,j}^{1,1} + \sum_{k=2}^{n-1} \sum_{j=1}^{n-1} \Big(v_{i,j}^{k,k} - v_{i,j}^{k-1,k} \\ - v_{i,j}^{k,k-1} + v_{i,j}^{k-1,k-1} \Big) - (n-1) v_i^{n-1} - \sum_{j=1}^{n-1} v_j^{n-1} + \sum_{j=1}^{n-1} v_{i,j}^{n-1,n-1}, \quad i \in [n-1]. \end{aligned} \tag{35}$$

But the above linear-quadratic expressions in position variables are again linear combinations of the equalities (20) – (22): Summing up (20) for $j \in [n-1],\ j \neq i$, (21) for $j \in [n-1],\ j \neq i$, and (22) for $j, k \in [n-1],\ j \neq i,\ k \neq 1$, gives (35). □ □

Lemma 13. *The equality constraints* (2) *for the* (TSP) *are assured by* (18) – (22).

Proof. The equalities (18) ensure that $x_{i,i} = 0,\ i \in [n-1]$, holds and the equalities (19) – (22) ensure $x_{n,n} = 0$. Now fixing target $i,\ i \in [n-1]$, and summing up (29) for $j \in [n-1]$, gives minus the right hand side of (30) plus 1: 1 – (30). Hence finally adding (30) proves $\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{i,j} = 1,\ i \in [n-1]$.

The remaining (TSP) equalities, i.e. $\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{j,i} = 1,\ i \in [n-1]$, and $\sum_{\substack{j=1 \\ j \neq n}}^{n} x_{n,j} = \sum_{\substack{j=1 \\ j \neq n}}^{n} x_{j,n} = 1$, can be shown in an analogous fashion. □ □

Finally adding the (TVP) inequality constraints (3) and (5) – (7), reformulated as linear-quadratic terms in position variables, to (SDP_4) gives our strongest, still tractable relaxation (SDP_5):

$$\max \ \{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (3), (5) – (7), (18) – (22), (25) and (26), } Z_s \in (\mathcal{E} \cap \mathcal{M}) \ \}. \tag{SDP_5}$$

Corollary 14. *The semidefinite programming relaxation* (SDP_5) *is as least as strong as the linear programming relaxation* $(\mathrm{LP}_{\mathrm{TVP}})$.

Proof. As the SDP formulation (SDP_5) ensures all $(\mathrm{LP}_{\mathrm{TVP}})$ constraints (reformulated as linear-quadratic expressions in position variables), the claim follows immediately. □ □

3 Obtaining Upper Bounds and Feasible Tours

The core of our SDP approach is to solve our SDP relaxations. The resulting fractional solutions constitute upper bounds for the exact SDP formulation of the (TVP). By the use of an easy two-opt improvement heuristic, which inter alia exploits the fractional SDP solutions, we obtain quite strong lower bounds, i.e., integer feasible solutions that describe feasible tours through the targets. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum. Let us now give some details on our upper and lower bound computation.

Looking at the constraint classes and their sizes it should be clear that maintaining explicitly $O(n^2)$ or more constraints is not an attractive option, at least for large-scale instances with for $n \geq 15$ targets. We therefore consider an approach suggested in [16] that was applied to the max cut problem [43] and several ordering problems [27], and adapt it for our SDP relaxations of the (TVP). Initially, we only aim at explicitly ensuring $Z \in \mathcal{E}$, which can be achieved with standard interior point methods, see, e.g. [22].

All other constraints are handled through Lagrangian duality in the objective function f. Thus the objective function f becomes non-smooth. The bundle method [16] iteratively evaluates f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to solving an SDP with the constraint $Z \in \mathcal{E}$ that can be solved efficiently by using again interior point methods. Finally we obtain an approximate minimizer of f that is guaranteed to yield a upper bound to the optimal solution of the SDP relaxations. Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations that are responsible for more than 95% of the required running time to control the overall computational effort. This limitation of the number of function evaluations leaves some room for further incremental improvement.

Let us further mention that we also tried to apply quadratic programming approaches to our (TVP) formulations, which however proved to be inapplicable already for small instances with 12 targets. Similar results were observed for quadratic ordering problems in the area of graph drawing [9].

To obtain feasible tours, we apply a simple two-opt improvement heuristic that yields quite good practical results. We call the heuristic every five bundle iterations and conduct each time 100 runs to generate 100 locally optimal tours until the duality gap is closed. If the current best tour is improved or another tour with the best known objective value is found, we update the current best tour. We use three different types of starting tours as input to our heuristic:

1. In the first run we generate an input tour from the current SDP solution vector v as follows: We compute the sums $\sum_{j=1}^{n-1} v_i^j$ for all $i \in [n]$ and sort them downwards to obtain a feasible tour. If the SDP upper bound is equal to the optimal objective value, this strategy yields the optimal tour.

2. If the current best tour is updated, we use the reversed best tour as input tour in the following run of our heuristic.

3. For the remaining runs we use randomly generated input tours.

To improve the input tours to locally optimal tours, we switch the targets i and j and adapt the current tour (Figure 2 a.)) as depicted in Figure 2 b.), which results in the following change of the objective function

$$\Delta_{\text{obj}} = w_{ji} - w_{ij} - d_{(i-1)j} - d_{j(i+1)} - d_{(j-1)i} - d_{i(j+1)} + d_{(i-1)i} + d_{i(i+1)} + d_{(j-1)j} + d_{j(j+1)}.$$

We update the tour if $\Delta_{\text{obj}} > 0$ and rerun the heuristic until no improvement of the tour is found. As we execute each improving switch, we consider the targets in ascending and descending order, each with 50 runs, to add further randomness to the heuristic procedure.

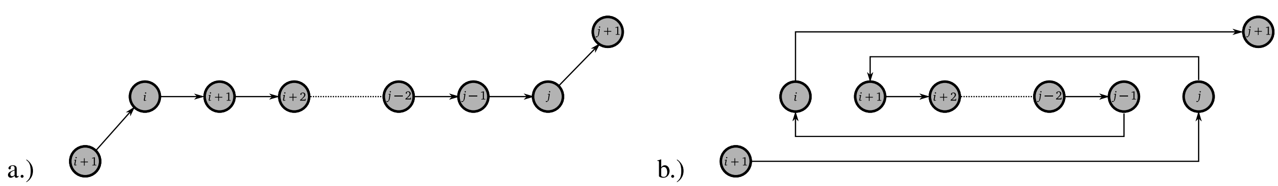

Figure 2: If we switch two targets i and j, we propose to adapt the original tour a.) as depicted in b.).

4 First Results and Comparisons of our Semidefinite Relaxations on Small and Medium Instances

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. We generate (TVP) benchmark instances in 2 different ways by

Instance (LOP)	Instance (TSP)	T	n	Best tour	$\texttt{SDP}_1$ (exact) ub	gap	# c	time	$\texttt{SDP}_2$ (exact) ub	gap	# c	time	$\texttt{SDP}_3$ (exact) ub	gap	# c	time	$\texttt{SDP}_4$ (exact) ub	gap	# c	time	$\texttt{SDP}_5$ (exact) ub	gap	# c	time
Toy example			5	-6	-5.39	0.03	64	0	-6.0	0.00	88	0	-6.0	0.00	107	0	-6.0	0.00	137	0	-6.0	0.00	242	0
PAL11	BR17	a	8	151	175.0	0.16	421	3	152.3	0.01	941	42	152.1	0.01	1734	6:17	151.0	0.00	1849	1:23	151.0	0.00	2238	2:17
PAL11	BR17	a	9	211	247.0	0.17	598	5	211.0	0.00	1502	1:37	211.0	0.00	2097	5:15	211.0	0.00	2456	3:37	211.0	0.00	3094	5:28
PAL11	BR17	a	10	253	313.9	0.24	1208	9	267.6	0.06	2496	26:36	279.8	0.11	3338	59:18	254.2	0.00	4956	4:28:08	253.0	0.00	4615	1:52:40
PAL11	BR17	a	11	304	385.9	0.27	1208	31	328.7	0.08	3760	1:33:21	348.8	0.15	4191	2:29:59	312.1	0.03	6587	11:24:50	305.8	0.01	7327	21:14:05
RAND8		r	8	76.4	80.3	0.05	376	12	76.4	0.00	820	11	76.4	0.00	1494	1:02	76.4	0.00	1887	1:00	76.4	0.00	2418	4:16
RAND9		r	9	93.6	107.6	0.15	566	22	95.6	0.02	1511	5:01	98.6	0.05	2323	28:09	93.6	0.00	2986	57:17	93.6	0.00	3571	1:11:14
RAND10		r	10	132.0	148.5	0.12	820	36	132.0	0.00	2422	36:28	136.0	0.03	3459	1:39:37	132.0	0.00	2986	1:08:49	132.0	0.00	3914	1:50:12
RAND11		r	11	179.4	197.4	0.10	1112	1:20	180.4	0.01	3310	1:26:26	184.4	0.03	5501	10:49:29	179.4	0.00	4720	7:14:59	179.4	0.00	6823	17:54:13

Table 1: (TVP) results obtained by our semidefinite relaxations in conjunction with our 2-opt heuristic. The SDP relaxations are solved exactly by Sedumi. The running times are given in sec or min:sec or in h:min:sec respectively.

Instance (LOP)	Instance (TSP)	T	n	Best tour	$\texttt{SDP}_1$ (bundle) ub	gap	time	$\texttt{SDP}_2$ (bundle) ub	gap	time	$\texttt{SDP}_3$ (bundle) ub	gap	time	$\texttt{SDP}_4$ (bundle) ub	gap	time	$\texttt{SDP}_5$ (bundle) ub	gap	time
PAL19	BR17	a	10	253	332.5	0.31	11	309.5	0.22	12	304.4	0.20	13	294.9	0.17	12	337.3	0.33	13
PAL19	BR17	a	11	304	412.6	0.36	15	388.9	0.28	16	366.6	0.21	18	364.3	0.20	19	424.2	0.40	24
PAL19	BR17	a	12	426	506.6	0.19	23	478.4	0.12	24	474.4	0.11	25	457.3	0.07	28	553.3	0.30	25
PAL19	BR17	a	13	498	607.9	0.22	52	575.5	0.16	44	568.7	0.14	47	549.8	0.10	45	685.9	0.38	43
PAL19	BR17	a	14	555	736.0	0.33	1:21	678.6	0.22	1:04	712.1	0.28	1:08	667.4	0.20	1:06	854.2	0.72	1:03
PAL19	BR17	a	15	627	837.1	0.34	1:26	787.8	0.26	1:28	812.5	0.30	1:33	766.7	0.22	1:31	1144.9	0.83	1:28
PAL19	BR17	a	16	736	952.5	0.29	1:55	932.3	0.27	1:56	938.6	0.28	1:57	900.7	0.22	2:13	1395.5	0.90	2:01
PAL19	BR17	a	17	816	1124.7	0.38	2:40	1063.9	0.30	2:48	1109.5	0.36	2:49	1052.5	0.29	2:44	1301.3	0.59	2:49
PAL19	FTV33	a	18	8607	10132	0.18	3:33	9973	0.16	3:32	10006	0.16	3:43	9913	0.15	3:45	11688	0.36	3:46
PAL19	FTV33	a	19	9309	11368	0.22	4:52	11115	0.19	4:40	11264	0.21	5:17	11017	0.18	4:56	14629	0.57	4:56
RAND10		r	10	132.0	152.5	0.16	11	142.5	0.08	12	142.5	0.08	13	138.0	0.05	18	139.5	0.06	15
RAND11		r	11	179.4	203.8	0.14	16	191.4	0.07	18	195.4	0.09	19	188.4	0.05	26	192.4	0.07	21
RAND12		r	12	207.6	239.6	0.15	25	223.2	0.07	26	227.2	0.09	28	218.2	0.05	28	224.2	0.08	27
RAND13		r	13	251.4	296.8	0.18	45	273.8	0.09	46	280.4	0.12	50	286.4	0.07	48	278.4	0.11	49
RAND14		r	14	282.4	335.4	0.19	1:04	310.4	0.10	1:02	319.8	0.13	1:07	304.8	0.08	1:05	315.4	0.12	1:08
RAND15		r	15	350.2	414.6	0.18	1:32	385.6	0.10	1:26	401.2	0.15	1:35	380.6	0.09	1:32	392.6	0.12	1:33
RAND16		r	16	399.0	483.4	0.21	2:08	451.4	0.13	1:58	467.4	0.17	1:57	445.0	0.12	2:05	482.4	0.21	2:04
RAND17		r	17	448.0	535.0	0.19	2:47	505.6	0.13	2:46	520.0	0.16	2:46	499.0	0.11	2:57	548.4	0.22	2.52
RAND18		r	18	510.2	615.6	0.21	3:32	577.2	0.13	3:49	599.2	0.17	3:40	570.2	0.12	3:55	694.6	0.36	3:41
RAND19		r	19	582.8	709.8	0.22	4:36	667.2	0.15	4:53	592.2	0.19	4:56	659.2	0.13	5:14	840.2	0.44	5:02

Table 2: (TVP) results obtained by our semidefinite relaxations in conjunction with our 2-opt heuristic. The (SDP) relaxations are solved approximately by the bundle method, restricted to 250 function evaluations. The running times are given in sec or min:sec respectively.

1. combining well-known (LOP) and asymmetric (TSP) instances from the literature,
2. using random generated data, where all (LOP) weights and non (Euclidean distances) are uniformly distributed random integers between 1 and 10 and the matrix collecting the (LOP) weights has a density of 50 %.[5]

For variant 1 we use appropriate instances from the well-known benchmark libraries LOLIB and TSPLIB.[6] If no instance of appropriate size s is available, we take the first s rows and columns of the instance next in size.[7] For the random data we generate five instances for each number of targets and report the averages over our runs. Additionally we assign one of the following two types T to each instance: a(symmetric), r(andom).

In Table 1 we compare the exact solution values of the relaxations (SDP_1) – (SDP_5) for (TVP) instances with up to 17 targets. For reasons of efficiency we used 10 function evaluations of the bundle method to obtain an initial set of constraints violated for our relaxations. We then applied Sedumi [45]; added the 500 most violated constraints; solved again using Sedumi; and repeated this process until no more violations were found or the duality gap is closed down completely. We also tried to solve our relaxations directly but the running times were at least one order of magnitude slower. The best tours are provided by the heuristic described in Section 3. The total running time of our heuristic is always clearly less than one second for all instances with up to 20 targets. Additionally we state the average number of constraints $\#c$ that we considered when obtaining the optimal solution of our SDP relaxations. The results in Table 1 show that

1. all proposed constraint classes help to improve the relaxation strength considerably,
2. all relaxations yield quite tight upper bounds and especially (SDP_5) nearly always closes the duality gap entirely,
3. the proposed heuristic yields very strong (TVP) tours, at least for small instances,
4. computing the SDP upper bounds becomes time consuming for $n \geq 10$ targets due to the larger number of constraints considered.

Hence for large instances with $n \geq 12$ targets we apply only the bundle method (without Sedumi) to our SDP relaxations and summarize the results obtained in Table 2. We restrict the bundle method to 250 function evaluations because its convergence process mostly slows down before that point. With this restriction, the running times of the bundle method are about the same for all SDP relaxations but there is a quite clear ranking of the quality of the upper bounds provided: 1.) $(\mathrm{SDP}_4) \to$ 2.) $(\mathrm{SDP}_2) \to$ 3.) $(\mathrm{SDP}_3) \to$ 4.) $(\mathrm{SDP}_1) \to$ 5.) (SDP_5). Additionally we observe that the random instances are easier to solve than the ones combining difficult (LOP) and (TSP) instances from the literature and that instances containing (a submatrix of) "PAL11" are particularly hard to solve.

Note that we also tried to solve (SDP_1) exactly as an alternative to solving (SDP_4) with the bundle method. This was only possible for instances with up to $n \leq 17$ within one day of computing time and for all instances with $n \geq 12$ we got worse gaps at considerably higher computational costs.

5 Computational Experience on Large Scale Instances

Based on the above comparisons, we decide to use (SDP_4) to tackle large scale instances with $n \geq 20$ targets. We double the number of function evaluations to 500 because the convergence process of the bundle method elongates for larger instances. We summarize the results obtained for instances with up to 50 targets in Table 3 and 4. We obtain reasonable gaps $\leq 55\%$ for all instances with up 50 targets. Again the random instances are easier to solve and our 2-opt improvement heuristic is very fast as its running time is always less than 5 seconds for all instances with up to 50 targets. In summary the results from both computational sections indicate that the size of the gaps obtained is dependent on the SDP relaxation applied, the difficulty of the (LOP) and (TSP) instances involved and the size of the (TVP) instance considered.

6 Conclusion

In this paper we proposed the first sophisticated SDP based algorithm for the Target Visitation Problem (TVP) that has several (military) applications. We showed that the (TVP) is a special Quadratic Position Problem (QPP) and exploited

[5] We do note count the diagonal entries as the have to be 0 by definition

[6] See https://www.iwr.uni-heidelberg.de/groups/comopt/software/index.html for details.

[7] When combining the "pal"-Instances with "br17" and "ftv33", we multiply the binary entries by 10 and 100 respectively to balance the (LOP) benefits with the (TSP) distances. Otherwise the optimal (TVP) tour would be very similar to the optimal (TSP) tour.

Instance		T	n	Best	$\texttt{SDP}_4$ (bundle)		
LOP	TSP			tour	ub	gap	time
PAL23	FTV33	a	20	11649	12377	0.06	14:25
PAL27	FTV33	a	25	19585	22134	0.13	57:37
PAL31	FTV33	a	30	24524	30737	0.25	2:08:26
P40	FTV35	a	35	18433	25465	0.38	9:13:42
P40	FTV44	a	40	24861	35302	0.42	21:51:08
P50	FTV44	a	45	31201	46139	0.48	41:48:57
P50	FTV55	a	50	39289	59769	0.52	65:39:28

Table 3: (TVP) results for instances from the literature, obtained by $(\texttt{SDP}_4)$ in conjunction with our 2-opt heuristic. $(\texttt{SDP}_4)$ solved approximately by the bundle method, restricted to 500 function evaluations. The running times are given in min:sec or h:min:sec respectively.

Instance		T	n	Best	$\texttt{SDP}_4$ (bundle)		
LOP	TSP			tour	ub	gap	time
RAND20		r	20	655	733	0.12	14:06
RAND25		r	25	990	1173	0.18	53:57
RAND30		r	30	1440	1760	0.22	2:08:01
RAND35		r	35	1985	2486	0.25	9:06:39
RAND40		r	40	2568	3317	0.29	21:43:24
RAND45		r	45	3216	4301	0.34	41:44:03
RAND50		r	50	4056	5608	0.38	65:35:46

Table 4: (TVP) results for random instances, obtained by $(\texttt{SDP}_4)$ in conjunction with our 2-opt heuristic. $(\texttt{SDP}_4)$ solved approximately by the bundle method, restricted to 500 function evaluations. The running times are given in sec or min:sec or h:min:sec respectively.

this connection to suggest a both theoretical sound and practicably competitive semidefinite optimization approach. Extending the proposed approach to further difficult combinatorial problems that are also special cases of the (QPP) (e.g. the Sequential Ordering Problem) is a worthwhile direction for future research.

References

[1] H. Achatz, P. Kleinschmidt, and J. Lambsdorff. Der corruption perceptions index und das linear ordering problem. *ORNews*, 26:10–12, 2006.

[2] M. F. Anjos and J. B. Lasserre, editors. *Handbook on Semidefinite, Conic and Polynomial Optimization Theory, Algorithms, Software and Applications*. International Series in Operations Research & Management Science. Springer-Verlag, New York, 2012.

[3] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem:A Computational Study*. Princeton University Press, 2006.

[4] A. Arulselvan, C. Commander, and P. Pardalos. A random keys based genetic algorithm for the target visitation problem. In *Advances in Cooperative Control and Optimization*, pages 389–397. Springer Berlin / Heidelberg, 2007.

[5] J. Békési, G. Galambos, M. Oswald, and G. Reinelt. Comparison of approaches for solving coupled task problems. Technical report, Ruprecht-Karls-Universität Heidelberg, 2008.

[6] L. Bertacco, L. Brunetta, and M. Fischetti. The linear ordering problem with cumulative costs. *European Journal of Operational Research*, 189(3):1345–1357, 2008.

[7] K. Boenchendorf. *Reihenfolgenprobleme/Mean-flow-time sequencing*. Mathematical Systems in Economics 74. Verlagsgruppe AthenÄaum, Hain, Scriptor, 1982.

[8] H. Chenery and T. Watanabe. International comparisons of the structure of production. *Econometrica*, 26:487–521, 1958.

[9] M. Chimani and P. Hungerländer. Exact approaches to multilevel vertical orderings. *INFORMS Journal on Computing*, 25(4):611–624, 2013.

[10] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, GSIA, Cranegie-Mellon University, 1976.

[11] W. J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2011.

[12] D. Cvetković, M. Cangalović, and V. Kovačevič-Vujčić. Semidefinite programming methods for the symmetric traveling salesman problem. In *Proceedings of the 7th International IPCO Conferenceo on Integer Programming and Combinatorial Optimization*, pages 126–136. Springer, 1999.

[13] E. de Klerk, D. V. Pasechnik, and R. Sotirov. On semidefinite programming relaxations of the traveling salesman problem. *SIAM Journal on Optimization*, 19(4):1559–1573, 2008.

[14] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer Verlag, Berlin, 1997.

[15] A. Duarte, R. Martí, A. Álvarez, and F. Ángel-Bello. Metaheuristics for the linear ordering problem with cumulative costs. *European Journal of Operational Research*, 216(2):270–277, 2012.

[16] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle method for semidefinite cutten plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105:451–469, 2006.

[17] P. C. Fishburn. Induced binary probabilities and the linear ordering polytope: a status report. *Mathematical Social Sciences*, 23(1):67–80, 1992.

[18] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, New York, 1974.

[19] F. Glover, T. Klastorin, and D. Klingman. Optimal weighted ancestry relationships. *Management Science*, 20: 1190–1193, 1974.

[20] D. Grundel and D. Jeffcoat. Formulation and solution of the target visitation problem. *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, 2004.

[21] G. Gutin and A. Punnen. *The Traveling Salesman Problem and Its Variations*. Springer, 2002.

[22] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[23] A. Hildenbrandt, O. Heismann, and G. Reinelt. The target visitation problem. Presentation held at the 18th Combinatorial Optimization Workshop in Aussois on January 09, 2014.

[24] A. Hildenbrandt, G. Reinelt, and O. Heismann. Integer programming models for the target visitation problem. In *Proceedings of the 16th International Multiconference of the Information Society*, volume A, pages 569–572, 2013.

[25] H.-W. Holub and H. Schnabl. *Input-Output-Rechnung: Input-Output-Tabellen*. Oldenbourg Wissenschaftsverlag, 1982.

[26] P. Hungerländer. New semidefinite programming relaxations for the linear ordering and the traveling salesman problem. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-03, 2014.

[27] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming*, 140(1): 77–97, 2013.

[28] P. Hungerländer and F. Rendl. A computational study and survey of methods for the single-row facility layout problem. *Computational Optimization and Applications*, 55(1):1–20, 2013.

[29] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1:1–25, 1997.

[30] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *Journal of the ACM*, 52(4):602–626, 2005.

[31] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[32] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.

[33] W. Leontief. Quantitative input-output relations in the economic system of the united states. *The Review of Economics and Statistics*, 18(3):105–125, 1936.

[34] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[35] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[36] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*. Applied Mathematical Sciences. Springer, 2011.

[37] R. Martí, G. Reinelt, and A. Duarte. A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, pages 1–21, 2011.

[38] R. Murphey and P. Pardalos, editors. *Cooperative Control and Optimization*. Springer, 2002.

[39] A. Newman. Cuts and orderings: On semidefinite relaxations for the linear ordering problem. In K. Jansen, S. Khanna, J. Rolim, and D. Ron, editors, *Lecture Notes in Computer Science*, volume 3122, pages 195–206. Springer, 2004.

[40] A. Newman and S. Vempala. Fences are futile: On relaxations for the linear ordering problem. In *Proceedings of the 8th International IPCO Conferenceo on Integer Programming and Combinatorial Optimization*, pages 333–347.

Springer, 2001.
[41] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.
[42] G. Reinelt. *The traveling salesman: computational solutions for TSP applications*. Springer, 1994.
[43] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.
[44] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrika*, 48(3-4):303–312, 1961.
[45] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.
[46] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.