Equivalence of an Approximate Linear Programming Bound with the Held-Karp Bound for the Traveling Salesman Problem

Alejandro Toriello
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
atoriello *at* isye *dot* gatech *dot* edu

February 11, 2014

Abstract

We consider two linear relaxations of the asymmetric *traveling salesman problem* (TSP), the Held-Karp relaxation of the TSP's arc-based formulation, and a particular *approximate linear programming* (ALP) relaxation obtained by restricting the dual of the TSP's shortest path formulation. We show that the two formulations produce equal lower bounds for the TSP's optimal cost regardless of cost structure; i.e. costs need not be non-negative, symmetric or metric. We then show how the ALP formulation can be modified to yield a relaxation for several TSP variants, and discuss how the formulations differ from arc-based relaxations.

1 Introduction

One of the most natural ways to tractably bound difficult combinatorial optimization problems is to study the *linear programming* (LP) relaxation of their *integer programming* (IP) formulation. But how close is the LP relaxation optimum to the true IP optimum? The quality of this bound is an important question, both for exact and approximation algorithms. The *traveling salesman problem* (TSP) is one of the most widely researched models in integer programming and combinatorial optimization [10, 36], and its relaxations have received sustained attention from the community. The IP formulation of the TSP dates at least as far back as the 1950's and the work of Dantzig, Fulkerson and Johnson [25], but the bound given by its LP relaxation is commonly known as the *Held-Karp* bound because of the two authors' seminal work on the subject [38, 39].

One way to study a relaxation's quality is to study the ratio between its optimum and the original problem's optimum. The supremum of the ratio between the integer optimum and the relaxation optimum over all or some problem instances is known as the relaxation's *integrality gap* with respect to these instances. For the TSP, the first result of this kind was obtained by Wolsey [53], who used Christofides' heuristic [22] to show that the Held-Karp bound has an integrality gap of at most 3/2 when costs are non-negative, symmetric and satisfy the triangle inequality. The study of the Held-Karp bound and other TSP relaxations continued in the subsequent years [33, 46, 52] and beyond the turn of the century [20, 21], including recent results, e.g. [9, 11, 43, 45]. The survey [51] thoroughly covers much of this work, focusing on advances of the last few years.

Despite the sustained attention, Wolsey's 3/2 integrality gap bound for the symmetric TSP with non-negative costs satisfying the triangle inequality has not been improved in this general case, even though the worst known example is an instance family achieving a gap of 4/3. Similarly, in the asymmetric case with non-negative costs satisfying the triangle inequality, the worst known example achieves a gap of 2 [21], but only recently has an integrality gap proof broken the logarithmic barrier [11].

Our main goal in this paper is to compare the Held-Karp bound to an *approximate linear programming* (ALP) relaxation recently proposed in [47] and shown to produce a bound at least as good as Held-Karp. Introduced in the mid 1980's and early 1990's [44, 49, 50], ALP is an approximate dynamic programming technique that obtains relaxations and bounds for *dynamic programs* (DP) by restricting the feasible region of their LP formulation. In the TSP's case, the DP is a shortest path formulation originally studied in [18, 35, 37], where states track the salesman's current location and the cities remaining to visit. ALP has gained popularity within optimization and operations research in the last decade [26, 27, 28], and has been applied in commodity valuation [42], economic lot scheduling [6], inventory routing [3, 4], joint replenishment [7, 8, 40], revenue management [5], stochastic games [32] and stochastic vehicle routing [48].

Specifically, our paper has two main contributions:

i) We prove the ALP bound's equivalence to the Held-Karp bound under arbitrary arc costs. This entirely new reformulation of the Held-Karp bound may reveal additional structure and help researchers design improved algorithms. The equivalence is striking because the Held-Karp bound uses quadratically many variables and exponentially many constraints in its primal relaxation, whereas the ALP uses quadratically many variables and exponentially many constraints in the *dual* space.

ii) We show how the ALP bound can be extended for modified versions of the TSP, and discuss differences between the ALP extensions and the arc-based LP relaxations used for these problems.

The remainder of the paper is organized as follows. Section 2 formulates the TSP and its bounds, introduces notation and has other preliminaries. Section 3 has the main proof of the two bounds' equivalence. Section 4 discusses extensions to some TSP variants, and Section 5 concludes outlining future avenues of research.

2 Formulation and Preliminaries

The TSP is specified by a set of *cities* $N := \{1, \dots, n\}$ and a distinguished city 0, sometimes called the *home city* or *depot*, and by arc costs $c_{ij} \in \mathbb{R}$ for every ordered pair $i, j \in N \cup 0$. (Here and elsewhere we identify singleton sets with their unique elements.) We do not assume non-negativity, symmetry or any other restriction on c. The TSP's objective is to find a minimum-cost order to visit the cities in N exactly once starting at and ultimately returning to 0; that is, we seek a permutation $\sigma : N \to N$ that minimizes $c_{0,\sigma(1)} + \sum_{i=1}^{n-1} c_{\sigma(i),\sigma(i+1)} + c_{\sigma(n),0}$.

The arc-based IP formulation for the TSP [25] is

$$\min_x \sum_{i \in N} \left(c_{0i} x_{0i} + \sum_{j \in N \setminus i} c_{ij} x_{ij} + c_{i0} x_{i0} \right) \tag{1a}$$

$$\text{s.t.} \sum_{i \in N} x_{0i} = 1 \tag{1b}$$

$$\sum_{i \in N} x_{i0} = 1 \tag{1c}$$

$$x_{0i} + \sum_{j \in N \setminus i} x_{ji} = 1, \qquad i \in N \tag{1d}$$

$$\sum_{j \in N \setminus i} x_{ij} + x_{i0} = 1, \qquad i \in N \tag{1e}$$

$$\sum_{i \in U} \Big(x_{0i} + \sum_{j \in N \setminus U} x_{ji} \Big) \geq 1, \qquad \varnothing \neq U \subseteq N \tag{1f}$$

$$\sum_{i \in U} \Big(\sum_{j \in N \setminus U} x_{ij} + x_{i0} \Big) \geq 1, \qquad \varnothing \neq U \subseteq N \tag{1g}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in N \cup 0, j \in (N \cup 0) \setminus i; \tag{1h}$$

Variables x_{ij} indicate when i immediately precedes j in the tour; we differentiate variables for arcs incident to 0 to highlight similarities to a subsequent formulation. The objective (1a) minimizes the tour's cost. Constraints (1b) through (1e) enforce unit flow balance through the depot and other cities, while the *subtour elimination constraints* (1f) and (1g) ensure the solution is connected. These constraints can be separated over in polynomial time using a min-cut algorithm as separation routine. The LP obtained by relaxing the integrality constraint in (1h) yields the Held-Karp bound; let z_{HK} denote this optimal value.

The DP formulation of the TSP [18, 35, 37] uses the state space

$$\mathcal{S} := \{(0, N), (0, \varnothing)\} \cup \{(i, U) : i \in N, U \subseteq N \setminus i\},$$

where the first two elements represent the initial and terminal states, while the intermediate states (i, U) indicate that the salesman's current location is i and he must still visit the cities in U before returning to 0. This state space yields the backwards Bellman recursion

$$y^*_{i,\varnothing} = c_{i0}, \qquad i \in N$$
$$y^*_{i,U} = \min_{j \in U} \{c_{ij} + y^*_{j,U \setminus j}\}, \qquad i \in N \cup 0, \varnothing \neq U \subseteq N \setminus i,$$

which can be alternately captured as the optimal solution of the LP

$$\max_y \; y_{0,N} \tag{2a}$$

$$\text{s.t. } y_{0,N} - y_{i,N \setminus i} \leq c_{0i}, \qquad i \in N \tag{2b}$$

$$y_{i,U \cup j} - y_{j,U} \leq c_{ij}, \qquad i \in N, j \in N \setminus i, U \subseteq N \setminus \{i, j\} \tag{2c}$$

$$y_{i,\varnothing} \leq c_{i0}, \qquad i \in N \tag{2d}$$

$$y_{i,U} \in \mathbb{R}, \qquad i \in N \cup 0, U \subseteq N \setminus i. \tag{2e}$$

The objective (2a) maximizes the value of the initial state $(0, N)$, and the optimal value $y^*_{0,N}$ is the optimal tour cost. Constraints (2b) through (2d) enforce shortest path label feasibility at all states in the DP; given an optimal y^* we can recover an optimal tour by starting at 0 and always choosing a minimizing city in the Bellman recursion. The dual of (2) is the TSP shortest path formulation.

The LP (2) is a strong dual for the TSP; in particular, any feasible solution provides a lower bound on the optimal cost. ALP attempts to provide a good bound by restricting the feasible region to a lower-dimensional subspace, and optimizing within this restriction. Many such approximations are possible for the TSP [47], but here we focus on the following. When the salesman is at city $i \in N$, suppose we assign him a nominal cost π_{i0}. For every remaining city $k \in U$ he must still visit, we add additional nominal costs π_{ik} that depend on the current location i. This yields an approximation

$$y_{i,U} \approx \pi_{i0} + \sum_{k \in U} \pi_{ik}, \qquad i \in N, U \subseteq N \setminus i.$$

Substituting into (2), we get the ALP

$$\max_{y,\pi} \; y \tag{3a}$$

$$\text{s.t. } y - \pi_{i0} - \sum_{k \in N \setminus i} \pi_{ik} \leq c_{0i}, \qquad i \in N \tag{3b}$$

$$\pi_{i0} - \pi_{j0} + \pi_{ij} + \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \leq c_{ij}, \qquad i \in N, j \in N \setminus i, U \subseteq N \setminus \{i, j\} \tag{3c}$$

$$\pi_{i0} \leq c_{i0}, \qquad i \in N \tag{3d}$$

$$y \in \mathbb{R}, \quad \pi \in \mathbb{R}^{n^2}. \tag{3e}$$

Though (3) still has an exponential number of constraints in (3c), we can separate over this class in $O(n^3)$ time: For every ordered pair (i, j), greedily add $k \in U$ when $(\pi_{ik} - \pi_{jk}) > 0$. Let z_{ALP} denote the optimal value.

Lemma 1 ([47])**.** $z_{HK} \leq z_{ALP}$.

3 Bound Equivalence

Our main result is the equality of the two bounds.

Theorem 2. $z_{HK} = z_{ALP}$.

Proof. We assume $n \geq 3$; the result is otherwise trivial. Given Lemma 1, it suffices to prove the reverse inequality. To do this, we show that any feasible solution of the LP relaxation of (1) has an objective that can be achieved by some feasible solution of the dual of (3). This dual is

$$\min_x \; \sum_{i \in N} \left(c_{0i} x_{0i} + \sum_{j \in N \setminus i} \sum_{U \subseteq N \setminus \{i,j\}} c_{ij} x_{ij}^U + c_{i0} x_{i0} \right) \tag{4a}$$

$$\text{s.t. } \sum_{i \in N} x_{0i} = 1 \tag{4b}$$

$$- x_{0i} + \sum_{j \in N \setminus i} \sum_{U \subseteq N \setminus \{i,j\}} (x_{ij}^U - x_{ji}^U) + x_{i0} = 0, \qquad i \in N \tag{4c}$$

$$-x_{0i} + \sum_{U \subseteq N \setminus \{i,j\}} x_{ij}^U + \sum_{k \in N \setminus \{i,j\}} \sum_{U \subseteq N \setminus \{i,j,k\}} (x_{ik}^{U \cup j} - x_{ki}^{U \cup j}) = 0, \qquad i \in N, j \in N \setminus i \tag{4d}$$

$$x \geq 0. \tag{4e}$$

As in (1), variables x_{0i} and x_{i0} represent the choice of visiting i at the start or end of the tour respectively. Meanwhile, the variables x_{ij}^U represent the choice of visiting j immediately after i when the other remaining cities are U. The objective (4a) minimizes the tour's cost, and constraint (4b) forces a unit of flow out of the depot. Constraint (4c) requires flow balance at each city $i \in N$. Constraints (4d) are "i-j" flow balance requirements: The flow into i when j must still be visited, either from the depot or other cities, must equal the flow out of i when j still remains, either to j itself or another city in $N \setminus \{i, j\}$. The formulation is indeed a relaxation, because this type of flow balance must hold not only for single remaining cities, but also for subsets of remaining cities; see [48] for a similar relaxation in a stochastic setting.

Given a feasible solution $\hat{x}$ to the LP relaxation of (1), in order to match $\hat{x}$'s objective value we need the following equations to be satisfied in terms of the variables in (4):

$$x_{0i} = \hat{x}_{0i}, \quad x_{i0} = \hat{x}_{i0}, \quad i \in N; \qquad \sum_{U \subseteq N \setminus \{i,j\}} x_{ij}^U = \hat{x}_{ij}, \quad i \in N, j \in N \setminus i. \tag{5}$$

Enforcing these equalities automatically satisfies (4b) and (4c) because $\hat{x}$ satisfies (1b) through (1e). After making the relevant substitutions, it remains to check the feasibility of the system

$$\begin{aligned} &\sum_{k \in N \setminus \{i,j\}} \sum_{U \subseteq N \setminus \{i,j,k\}} (x_{ik}^{U \cup j} - x_{ki}^{U \cup j}) = \hat{x}_{0i} - \hat{x}_{ij}, && i \in N, j \in N \setminus i \\ &\sum_{U \subseteq N \setminus \{i,j\}} x_{ij}^U = \hat{x}_{ij}, && i \in N, j \in N \setminus i \\ &x \geq 0. \end{aligned}$$

Applying Farkas' Lemma, this system is feasible if

$$\sum_{i \in N} \sum_{j \in N \setminus i} ((\hat{x}_{0i} - \hat{x}_{ij})\mu_{ij} + \hat{x}_{ij}\rho_{ij}) > 0 \tag{6a}$$

$$\sum_{k \in U} (\mu_{ik} - \mu_{jk}) + \rho_{ij} \leq 0, \qquad i \in N, j \in N \setminus i, U \subseteq N \setminus \{i,j\} \tag{6b}$$

$$\mu_{ij}, \rho_{ij} \in \mathbb{R}, \qquad i \in N, j \in N \setminus i \tag{6c}$$

is infeasible. By taking $U = \varnothing$ in (6b) it follows that $\rho \leq 0$; furthermore, given any μ, we can always satisfy all constraints (6b) by choosing ρ values that are negative enough. To make the left-hand side of (6a) as large as possible, we can therefore choose $-\rho_{ij} = \max_{U \subseteq N \setminus \{i,j\}} \sum_{k \in U} (\mu_{ik} - \mu_{jk})$; this is the largest (least negative) value of ρ_{ij} guaranteed to satisfy (6b) for this ordered i, j pair and all $U \subseteq N \setminus \{i, j\}$. Projecting out the ρ variables with this identity, we need to check whether any μ can satisfy

$$\sum_{i \in N} \sum_{j \in N \setminus i} \left(\mu_{ij}(\hat{x}_{ij} - \hat{x}_{0i}) + \hat{x}_{ij} \max_{U \subseteq N \setminus \{i,j\}} \sum_{k \in U} (\mu_{ik} - \mu_{jk}) \right) < 0.$$

We have reversed the expression's sign for convenience. Using $(\cdot)_+ := \max\{0, \cdot\}$, we can rearrange terms and rewrite this left-hand side expression as

$$\sum_{i\in N}\sum_{j\in N\setminus i}\Big(\mu_{ij}(\hat{x}_{ij} - \hat{x}_{0i}) + \hat{x}_{ij}\sum_{k\in N\setminus\{i,j\}}(\mu_{ik} - \mu_{jk})_+\Big)$$

$$= \sum_{j\in N}\sum_{i\in N\setminus j}\Big(\mu_{ij}(\hat{x}_{ij} - \hat{x}_{0i}) + \sum_{k\in N\setminus\{i,j\}}\hat{x}_{ik}(\mu_{ij} - \mu_{kj})_+\Big).$$

The terms are now grouped so that within the first summation all μ variables share the same second city index, j. We focus on an arbitrary summand given by a fixed j; assume without loss of generality that $j = n$, and also assume (possibly by permuting the remaining indices) that $\mu_{1n} \geq \cdots \geq \mu_{n-1,n}$. The summand can then be written as

$$\sum_{i=1}^{n-1}\Big(\mu_{in}(\hat{x}_{in} - \hat{x}_{0i}) + \sum_{k=i+1}^{n-1}\hat{x}_{ik}(\mu_{in} - \mu_{kn})\Big)$$

$$= \sum_{i=1}^{n-1}\mu_{in}\Big(-\sum_{k=0}^{i-1}\hat{x}_{ki} + \sum_{k=i+1}^{n}\hat{x}_{ik}\Big)$$

$$= \mu_{1n}\Big(-\hat{x}_{01} + \sum_{k=2}^{n}\hat{x}_{1k}\Big) + \sum_{i=2}^{n-2}\mu_{in}\Big(-\sum_{k=0}^{i-1}\hat{x}_{ki} + \sum_{k=i+1}^{n}\hat{x}_{ik}\Big) + \mu_{n-1,n}\Big(-\sum_{k=0}^{n-2}\hat{x}_{k,n-1} + \hat{x}_{n-1,n}\Big)$$

$$= \mu_{1n}\Big(\sum_{k=2}^{n}(\hat{x}_{0k} + \hat{x}_{1k}) - 1\Big) + \sum_{i=2}^{n-2}\mu_{in}\Big(-\sum_{k=0}^{i-1}\hat{x}_{ki} + \sum_{k=i+1}^{n}\hat{x}_{ik}\Big) - \mu_{n-1,n}\Big(\sum_{k=0}^{n-2}(\hat{x}_{k,n-1} + \hat{x}_{kn}) - 1\Big),$$

where for the last equality we use (1b) in the leftmost term and (1d) (for $i = n$) in the rightmost term. Finally, using the identity

$$-\sum_{k=0}^{i-1}\sum_{\ell=i}^{n}\hat{x}_{k\ell} + \sum_{k=0}^{i}\sum_{\ell=i+1}^{n}\hat{x}_{k\ell} = -\sum_{k=0}^{i-1}\hat{x}_{ki} + \sum_{k=i+1}^{n}\hat{x}_{ik}$$

for each $i = 2, \ldots, n-2$, we arrive at

$$\sum_{i=1}^{n-2}(\mu_{in} - \mu_{i+1,n})\Big(\sum_{k=0}^{i}\sum_{\ell=i+1}^{n}\hat{x}_{k\ell} - 1\Big) \geq 0.$$

This quantity must be non-negative because of the non-increasing assumption on the μ_{in} values and the subtour elimination constraints (1f) for $U = \{i+1, \ldots, n\}$ and $i = 1, \ldots, n-2$. □

Theorem 2 also confirms that the equivalence between the Held-Karp and ALP bounds is constructive in the following sense: The proof of Lemma 1 from [47] gives a construction whereby any feasible solution of the Held-Karp dual formulation can be transformed into a feasible solution of (3) with equal objective. The preceding proof shows that a feasible solution of the Held-Karp formulation (1) has some transformation to a feasible solution of (4) with equal objective. But when solving (4) to optimality, Theorem 2 ensures that the transformation (5) in the reverse direction (which is unique) also yields an optimal solution to the Held-Karp formulation (1). We can therefore solve the Held-Karp bound to optimality and use its dual to obtain an optimal solution for the ALP, or solve the ALP bound and use its dual (4) to obtain an optimal solution for Held-Karp. Table 1 summarizes this discussion.

	Held-Karp		ALP
primal	(1)	$\xleftarrow{(5)}$	(4)
	$\Updownarrow$		$\Updownarrow$
dual	dual of (1)	$\xrightarrow{[47]}$	(3)

Table 1: Summary of optimal solution transformations between Held-Karp and ALP formulations.

4 Application to Other TSP Models

The TSP's applicability in a wide range of research areas [10] has motivated the study of many modified versions of the problem [36]. In some cases, the modification implies significant changes to the formulation (1). In this section we highlight how the ALP bound can be readily extended to several TSP variants, and compare these extensions to the corresponding change in the arc-based formulation. Table 2 summarizes the variants we focus on and the necessary change in the ALP formulation.

TSP Modification	Separation Problem for Each City Pair
None	Greedy
Time-Dependent	Sort
Average Cost	Greedy
Precedence Constraints	Optimal Closure
Time Slots	$O(n)$ Weighted Assignments
Prize-Collecting	Greedy (for every pair and every city)

Table 2: Summary of ALP separation problem for various TSP models.

4.1 Time-Dependent TSP

In the *time-dependent TSP* (TDTSP) [1], the cost of traveling from i to j depends also on the place in the tour when it occurs. Specifically, arc costs become c_{ij}^t for $t = 0, \dots, n$, where time indices $t \in \{0, n\}$ represent the arc out of and into 0 and other indices represent intermediate arcs in the tour. The transition between states $(i, U \cup j)$ and (j, U) thus has cost $c_{ij}^{n-|U|-1}$, and the ALP (3) becomes

$$\begin{aligned}
\max_{y,\pi}\ & y \\
\text{s.t. } & y - \pi_{i0} - \sum_{k \in N \setminus i} \pi_{ik} \le c_{0i}^0, && i \in N \\
& \pi_{i0} - \pi_{j0} + \pi_{ij} + \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \le c_{ij}^{n-|U|-1}, && i \in N, j \in N \setminus i, U \subseteq N \setminus \{i, j\} \\
& \pi_{i0} \le c_{i0}^n, && i \in N \\
& y \in \mathbb{R}, \quad \pi \in \mathbb{R}^{n^2}.
\end{aligned}$$

In contrast, an arc-based formulation for the TDTSP uses a cubic number of variables over a time-expanded network [1]. The ALP is polynomially solvable provided the exponential constraint class can be separated over efficiently. For an ordered pair i, j, this problem is

$$\max_{U \subseteq N \setminus \{i,j\}} \left\{ \sum_{k \in U} (\pi_{ik} - \pi_{jk}) - c_{ij}^{n-|U|-1} \right\} = \max_{t=1,\dots,n-1} \left\{ -c_{ij}^t + \max_{\substack{U \subseteq N \setminus \{i,j\} \\ |U| = n-t-1}} \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \right\}.$$

To solve, it suffices to sort the values $(\pi_{ik} - \pi_{jk})$ in non-increasing order, then add them one by one in this order while checking each successive sum against the corresponding c_{ij}^t. The complexity of this procedure is dominated by the sort, and thus the separation problem can be solved in $O(n^3 \log n)$ time.

One important version of the TDTSP with a simpler separation problem is the *average-cost TSP*, in which the objective is to minimize the average cost of the tour's induced path from 0 to each city in N; for instance, this objective is useful when arc costs represent time, cities represent customers and the objective is to minimize the customers' average waiting time. In this case, the costs are given by $c_{ij}^t = (n-t)c_{ij}$, where c_{ij} are the underlying arc costs, because an arc traversed at step t affects the average cost of all cities after it. The separation problem is then equivalent to

$$\max_{U \subseteq N \setminus \{i,j\}} \sum_{k \in U} (\pi_{ik} - \pi_{jk} - c_{ij}),$$

and this problem can be solved greedily as in the time-independent TSP, by putting $k \in U$ if $(\pi_{ik} - \pi_{jk} - c_{ij}) > 0$.

4.2 Precedence-Constrained TSP

In certain applications, the salesman may be required to visit some cities before others. One example is pickup and delivery problems (e.g. [31]), where a vehicle must transport products from some locations to others. In the *precedence-constrained TSP* [12, 16, 41], in addition to the other TSP parameters we are given a directed, acyclic and transitive precedence graph $G = (N, A)$: If $(i, j) \in A$ then the tour must visit i before j. In order to formulate a valid arc-based IP for this problem, a new exponential class of inequalities must be added to (1), and the resulting LP relaxation can be optimized in polynomial time; see [16].

For the ALP, the addition of precedence constraints renders some states (i, U) infeasible. Specifically, the set U must contain i's out-neighbors (successors), and must be closed in G; no precedence arcs may go from cities in U to cities outside U, since this would indicate a precedence violation. Let $\delta^+(U) := \{(i,j) \in A : i \in U, j \notin U\}$, $\Gamma^+(U) := \{j \in N \setminus U : (i,j) \in \delta^+(U)\}$, and define $\delta^-(U), \Gamma^-(U)$ analogously. The ALP (3) becomes

$$\begin{aligned}
\max_{y,\pi} \ & y \\
\text{s.t. } & y - \pi_{i0} - \sum_{k \in N \setminus i} \pi_{ik} \le c_{0i}, \qquad i \in N, \delta^-(i) = \varnothing \\
& \pi_{i0} - \pi_{j0} + \pi_{ij} + \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \le c_{ij}, \\
& \qquad i, j \in N, (j,i) \notin A, U \subseteq N \setminus (\{i,j\} \cup \Gamma^-(\{i,j\})),
\end{aligned}$$

$$U \supseteq \Gamma^+(\{i,j\}) \setminus j, \delta^+(U) = \varnothing$$
$$\pi_{i0} \le c_{i0}, \qquad i \in N, \delta^+(i) = \varnothing$$
$$y \in \mathbb{R}, \quad \pi \in \mathbb{R}^{n^2}.$$

Similarly, to separate over the exponential constraint class, for every ordered pair i,j with $(j,i) \notin A$ we solve

$$\max\left\{\sum_{k \in U} (\pi_{ik} - \pi_{jk}) : \Gamma^+(\{i,j\}) \setminus j \subseteq U \subseteq N \setminus (\{i,j\} \cup \Gamma^-(\{i,j\})), \delta^+(U) = \varnothing\right\}.$$

This is an optimal closure problem, in which we seek the maximum-weight set $U \subseteq N \setminus \{i,j\}$ that contains i's and j's successors (except j if it is a successor of i), does not contain i's and j's predecessors, and is closed in A. The problem can be recast as a minimum cut problem and solved in polynomial time [23].

4.3 TSP with Time Slots

Another important constraint in some TSP applications restricts the timing of visits to cities. For the *TSP with time windows* [17, 30, 41], in addition to costs of travel between cities there is a travel time. We assume the tour begins at time 0, and the arrival time at each city i must be between a window specified by earliest and latest possible arrival times; the salesman may wait if he arrives before the time window.

Typical DP formulations of routing problems with time windows require an additional state space dimension tracking time [41], and are beyond the scope of this paper. Similarly, arc-based formulations usually require auxiliary variables indicating arrival and/or departure times at each city, e.g. [29]. However, there is a special case that can be modeled directly using our current formulation. If travel times are universally equal to one, the time windows become restrictions that cities be visited on particular positions, or slots, in the tour. For example, the salesman may be traveling by plane once a day between cities, but he can travel to some cities only on weekdays, or only at the start or end of the tour, etc. The example also illustrates that we can allow multiple windows (i.e. given by the union of disjoint intervals).

Let $M := \{1, \dots, n\}$ be the set of slots or positions on the tour, and let $B = (N \cup M, E)$ be a bipartite graph with bipartition (N, M) indicating each city's allowed slots on the tour: $(i,t) \in E$ if i may be in position t. A state (i,U) is feasible for the DP if and only if $(i, n-|U|) \in E$, U can be perfectly matched to $\{n-|U|+1, \dots, n\}$ in B, and $N \setminus (U \cup i)$ can be perfectly matched to $\{1, \dots, n-|U|-1\}$ in B. The ALP then becomes

$$\max_{y,\pi} \; y$$
$$\text{s.t. } y - \pi_{i0} - \sum_{k \in N \setminus i} \pi_{ik} \le c_{0i}, \qquad i \in N, (i,1) \in E$$
$$\pi_{i0} - \pi_{j0} + \pi_{ij} + \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \le c_{ij},$$
$$i \in N, j \in N \setminus i, U \subseteq N \setminus \{i,j\}, (i, n-|U|-1) \in E, (j, n-|U|) \in E,$$
$$U \text{ can be perfectly matched to } \{n-|U|+1, \dots, n\},$$
$$N \setminus (U \cup \{i,j\}) \text{ can be perfectly matched to } \{1, \dots, n-|U|-2\}$$

$$\pi_{i0} \le c_{i0}, \qquad i \in N, (i, n) \in E$$
$$y \in \mathbb{R}, \quad \pi \in \mathbb{R}^{n^2}.$$

To separate over the exponential constraint class, we consider an ordered pair i, j and every possible slot $t \in M$ with $(i, t), (j, t + 1) \in E$. For each such triple i, j, t, we must perfectly match the remaining cities $N \setminus \{i, j\}$ to remaining slots $M \setminus \{t, t + 1\}$ in G. Cities $k \in N \setminus \{i, j\}$ matched to slots $t + 2, \ldots, n$ add weight $(\pi_{ik} - \pi_{jk})$, while cities matched to slots $1, \ldots, t - 1$ add no weight. The separation problem for the triple i, j, t is therefore a weighted assignment problem, and there are $O(n)$ slots t for every pair i, j, requiring the solution of $O(n^3)$ assignment problems to separate.

4.4 Prize-Collecting TSP

In the *prize-collecting TSP* [13, 14, 15], the salesman can select which cities in N to visit. In the version we consider [19, 34], the salesman pays a penalty p_i for every city i left out of the tour. The IP formulation then includes additional city variables indicating which cities form part of the tour, and these variables appear in the right-hand side of the flow balance and subtour elimination constraints.

In the DP, every state (i, U) includes the additional feasible action of returning to the depot 0 without visiting the cities in U, incurring cost $c_{i0} + \sum_{k \in U} p_k$. The ALP is

$$\max \; y \tag{7a}$$
$$\text{s.t. } y \le \sum_{i \in N} p_i \tag{7b}$$
$$y - \pi_{i0} - \sum_{k \in N \setminus i} \pi_{ik} \le c_{0i}, \qquad i \in N \tag{7c}$$
$$\pi_{i0} - \pi_{j0} + \pi_{ij} + \sum_{k \in U} (\pi_{ik} - \pi_{jk}) \le c_{ij}, \qquad i \in N, j \in N \setminus i, U \subseteq N \setminus \{i, j\} \tag{7d}$$
$$\pi_{i0} + \sum_{k \in U} \pi_{ik} \le c_{i0} + \sum_{k \in U} p_k, \qquad i \in N, U \subseteq N \setminus i \tag{7e}$$
$$y \in \mathbb{R}, \quad \pi \in \mathbb{R}^{n^2}, \tag{7f}$$

where constraint (7b) indicates the salesman may choose not to visit any cities at all, and we replace constraint (3d) with the more general (7e) because the salesman is allowed to return to 0 at any point in the tour. The first exponential constraint class (7d) can be separated over exactly as in the basic TSP case, with a greedy algorithm for every pair of cities i, j. Similarly, the constraints (7e) can be separated over greedily: For each i, add $k \in U$ only if $(\pi_{ik} - p_{ik}) > 0$.

5 Conclusions

We have proved that the ALP bound (3) for the TSP is a reformulation of the Held-Karp bound, and shown how to extend the bound to several important TSP variants. We believe this opens several research questions for computational and theoretical work on the TSP.

Computationally, ALP directly implies heuristics known sometimes as *price-directed policies* [3, 4]. These heuristics can be embedded within an exact optimization branch-and-bound framework, where they become strong branching rules or a version of hybrid pseudo-cost branching strategies [2], to either enhance the heuristic's performance or exactly solve the instances. This topic is one of our current research avenues.

From a theoretical perspective, the design of many approximation algorithms based on an LP relaxation depends on the structure of the relaxation's polyhedron and its extreme points. It may be useful to characterize the extreme points of (3) and (4) with results similar to those in [20, 24, 43], possibly assuming here that arc costs have additional structure, such as non-negativity, the triangle inequality or symmetry.

References

[1] H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa, *The Time Dependent Traveling Salesman Problem: Polyhedra and Algorithm*, Mathematical Programming Computation **5** (2013), 27–55.

[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*, Operations Research Letters **33** (2005), 42–54.

[3] D. Adelman, *Price-Directed Replenishment of Subsets: Methodology and its Application to Inventory Routing*, Manufacturing and Service Operations Management **5** (2003), 348–371.

[4] ______, *A Price-Directed Approach to Stochastic Inventory/Routing*, Operations Research **52** (2004), 499–514.

[5] ______, *Dynamic Bid Prices in Revenue Management*, Operations Research **55** (2007), 647–661.

[6] D. Adelman and C. Barz, *A Unifying Approximate Dynamic Programming Model for the Economic Lot Scheduling Problem*, Mathematics of Operations Research (2013), Forthcoming.

[7] D. Adelman and D. Klabjan, *Duality and Existence of Optimal Policies in Generalized Joint Replenishment*, Mathematics of Operations Research **30** (2005), 28–50.

[8] ______, *Computing Near-Optimal Policies in Generalized Joint Replenishment*, INFORMS Journal on Computing **24** (2011), 148–164.

[9] H.-C. An, R. Kleinberg, and D.B. Shmoys, *Improving Christofides' algorithm for the s-t path TSP*, Proceedings of the 44th ACM Symposium on Theory of Computing (STOC), Association for Computing Machinery, 2012, pp. 875–886.

[10] D.L. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook, *The traveling salesman problem: A computational study*, Princeton University Press, Princeton, New Jersey, 2006.

[11] A. Asadpour, M.X. Goemans, A. Madry, S. Oveis Gharan, and A. Saberi, *An* $O(\log n/\log\log n)$*-approximation algorithm for the asymmetric traveling salesman problem*, Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2010, pp. 379–389.

[12] N. Ascheuer, M. Jünger, and G. Reinelt, *A Branch & Cut Algorithm for the Asymmetric Traveling Salesman Problem with Precedence Constraints*, Computational Optimization and Applications **17** (2000), 61–84.

[13] E. Balas, *The Prize Collecting Traveling Salesman Problem*, Networks **19** (1989), 621–636.

[14] ______, *The Prize Collecting Traveling Salesman Problem: II. Polyhedral Results*, Networks **25** (1995), 199–216.

[15] ______, *The Prize Collecting Traveling Salesman Problem and Its Applications*, in Gutin and Punnen [36], pp. 663–696.

[16] E. Balas, M. Fischetti, and W.R. Pulleyblank, *The Precedence-Constrained Asymmetric Traveling Salesman Polytope*, Mathematical Programming **68** (1995), 241–265.

[17] R. Baldacci, A. Mingozzi, and R. Roberti, *New State-Space Relaxations for Solving the Traveling Salesman Problem with Time Windows*, INFORMS Journal on Computing **24** (2012), 356–371.

[18] R. Bellman, *Dynamic Programming Treatment of the Travelling Salesman Problem*, Journal of the Association for Computing Machinery **9** (1962), 61–63.

[19] D. Bienstock, M.X. Goemans, D. Simchi-Levi, and D. Williamson, *A note on the prize collecting traveling salesman problem*, Mathematical Programming **59** (1993), 413–420.

[20] R. Carr and S. Vempala, *On the Held-Karp relaxation for the asymmetric and symmetric traveling salesman problems*, Mathematical Programming **100** (2004), 569–587.

[21] M. Charikar, M.X. Goemans, and H. Karloff, *On the Integrality Ratio for the Asymmetric Traveling Salesman Problem*, Mathematics of Operations Research **31** (2006), 245–252.

[22] N. Christofides, *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*, Tech. Report ADA025602, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[23] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, John Wiley & Sons, Inc., New York, 1998.

[24] G. Cornuéjols, J. Fonlupt, and D. Naddef, *The Traveling Salesman Problem on a Graph and some Related Integer Polyhedra*, Mathematical Programming **33** (1985), 1–27.

[25] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson, *Solution of a large scale traveling-salesman problem*, Operations Research **2** (1954), 393–410.

[26] D.P. de Farias and B. van Roy, *The Linear Programming Approach to Approximate Dynamic Programming*, Operations Research **51** (2003), 850–865.

[27] ______, *On Constraint Sampling in the Linear Programming Approach to Approximate Dynamic Programming*, Mathematics of Operations Research **29** (2004), 462–478.

[28] V.V. Desai, V.F. Farias, and C.C. Moallemi, *Approximate Dynamic Programming via a Smoothed Linear Program*, Operations Research **60** (2012), 655–674.

[29] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh, and F. Soumis, *Vehicle Routing with Time Windows: Optimization and Approximation*, Vehicle Routing: Methods and Studies (B.L. Golden and A.A. Assad, eds.), Elsevier Science Publishers B.V., North-Holland, 1988, pp. 65–84.

[30] Y. Dumas, J. Desrosiers, E. Gelinas, and M. Solomon, *An Optimal Algorithm for the Traveling Salesman Problem with Time Windows*, Operations Research **43** (1995), 367–371.

[31] I. Dumitrescu, S. Ropke, J.-F. Cordeau, and G. Laporte, *The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm*, Mathematical Programming **121** (2010), 269–305.

[32] V.F. Farias, D. Sauré, and G.Y. Weintraub, *An Approximate Dynamic Programming Approach to Solving Dynamic Oligopoly Models*, RAND Journal of Economics **43** (2012), 253–282.

[33] M.X. Goemans, *Worst-case Comparison of Valid Inequalities for the TSP*, Mathematical Programming **69** (1995), 335–349.

[34] ———, *Combining Approximation Algorithms for the Prize-Collecting TSP*, Preprint available on-line at arXiv:0910.0553v1, 2009.

[35] R.H. Gonzales, *Solution to the traveling salesman problem by dynamic programming on the hypercube*, Tech. Report 18, Operations Research Center, Massachusetts Institute of Technology, 1962.

[36] G. Gutin and A.P. Punnen (eds.), *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[37] M. Held and R.M. Karp, *A Dynamic Programming Approach to Sequencing Problems*, Journal of the Society of Industrial and Applied Mathematics **10** (1962), 196–210.

[38] ———, *The traveling-salesman problem and minimum spanning trees*, Operations Research **18** (1970), 1138–1162.

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

[40] D. Klabjan and D. Adelman, *An Infinite-Dimensional Linear Programming Algorithm for Deterministic Semi-Markov Decision Processes on Borel Spaces*, Mathematics of Operations Research **32** (2007), 528–550.

[41] A. Mingozzi, L. Bianco, and S. Ricciardelli, *Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints*, Operations Research **45** (1997), 365–377.

[42] S. Nadarajah, F. Margot, and N. Secomandi, *Approximate Dynamic Programs for Natural Gas Storage Valuation Based on Approximate Linear Programming Relaxations*, Tech. Report 2011-E5, Tepper School of Business, Carnegie Mellon University, 2011.

[43] T. Nguyen, *A simple LP relaxation for the asymmetric traveling salesman problem*, Mathematical Programming **141** (2013), 549–559.

[44] P.J. Schweitzer and A. Seidmann, *Generalized Polynomial Approximations in Markovian Decision Processes*, Journal of Mathematical Analysis and Applications **110** (1985), 568–582.

[45] A. Sebő, *Eight-Fifth Approximation for TSP Paths*, Proceedings of the 16th Conference on Integer Programming and Combinatorial Optimization (Berlin-Heidelberg) (M. Goemans and J. Correa, eds.), Lecture Notes in Computer Science, vol. 7801, Springer, 2013, pp. 362–374.

[46] D.B. Shmoys and D.P. Williamson, *Analyzing the Held-Karp TSP Bound: A Monotonicity Property with Application*, Information Processing Letters **35** (1990), 281–285.

[47] A. Toriello, *Optimal Toll Design: A Lower Bound Framework for the Asymmetric Traveling Salesman Problem*, Mathematical Programming (2013), DOI 10.1007/s10107-013-0631-6. Forthcoming.

[48] A. Toriello, W.B. Haskell, and M. Poremba, *A Dynamic Traveling Salesman Problem with Stochastic Arc Costs*, Submitted. Preprint available at `http://www2.isye.gatech.edu/~atoriello3/stoch_tsp_alp_v2.pdf`, 2012.

[49] M.A. Trick and S.E. Zin, *A Linear Programming Approach to Solving Stochastic Dynamic Programs*, Unpublished manuscript available online at `http://mat.gsia.cmu.edu/trick/`, 1993.

[50] ______, *Spline Approximations to Value Functions: A Linear Programming Approach*, Macroeconomic Dynamics **1** (1997), 255–277.

[51] J. Vygen, *New approximation algorithms for the TSP*, Optima: Mathematical Optimization Society Newsletter **90** (2012), 1–12.

[52] D.P. Williamson, *Analysis of the Held-Karp Heuristic for the Traveling Salesman Problem*, Master's thesis, Massachusetts Institute of Technology, 1989.

[53] L.A. Wolsey, *Heuristic Analysis, Linear Programming and Branch and Bound*, Mathematical Programming Study **13** (1980), 121–134.