On the strength of approximate linear programming relaxations for the traveling salesman problem

Ricardo Fukasawa
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, ON, Canada
rfukasawa@uwaterloo.ca

Allan Sapucaia Barboza
Institute of Computing
University of Campinas
Campinas, Brazil
allansapucaia@gmail.com

Alejandro Toriello
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA, USA
atoriello@isye.gatech.edu

August 24, 2016

Abstract

We study the strength of a recently proposed *approximate linear programming* (ALP) family of lower bounds for the *traveling salesman problem* (TSP). We show that the initial lower bound is equivalent to the Held-Karp bound (settling an open question from [28]), and then show that the final lower bound from the family is tight. We then compare it to another family of TSP lower bounds, the *branch-cut-and-price* (BCP) family. We show that the two bound families are in general incomparable in terms of bound strength, and then compare their performance in a computational study. Our empirical conclusion is that the ALP family produces bounds of similar quality to BCP.

1 Introduction

The *traveling salesman problem* (TSP) is one of the most well-known and studied combinatorial optimization problems [8]. Its general description is as follows: Given a set of cities and travel costs between every pair of cities, a "salesman" wishes to visit every city exactly once and then return to the starting city, incurring the cheapest possible total cost. The problem has been the focus of heavy study, with current heuristics and exact algorithms allowing instances of many thousands of cities to be solved to optimality.

In this paper, we consider the asymmetric TSP; let $G = (V, A)$ be the complete directed graph on vertices $V := \{0\} \cup N$, where $N := \{1, 2, ..., n\}$. Without loss of generality, we let 0 be the salesman's starting point, which we also call the *depot*. Each arc (i, j) has an associated cost, c_{ij}. The asymmetric TSP seeks the minimum-cost directed Hamiltonian cycle (also called a *tour*), starting and ending at the depot. For notation simplicity, we represent arc (i, j) by ij. Moreover, for any $U \subseteq V$, we use $\delta^+(U) := \{ij \in A : i \in U, j \notin U\}$ and $\delta^-(U) := \delta^+(V \setminus U)$. We also denote $\delta^+(i) := \delta^+(\{i\})$ and $\delta^-(i) := \delta^-(\{i\})$, and generally identify singleton sets with their unique elements when there is no danger of confusion.

The TSP has been modeled using *integer programming* (IP) since as far back as the 1950's [12]. One of the most common formulations is the following:

$$\min \sum_{a \in A} c_a x_a \tag{1a}$$

$$\text{s.t.} \sum_{a \in \delta^+(i)} x_a = \sum_{a \in \delta^-(i)} x_a = 1 \qquad i \in V \tag{1b}$$

$$\sum_{a \in \delta^+(U)} x_a \geq 1 \qquad \emptyset \subsetneq U \subseteq N \tag{1c}$$

$$x \in \mathbb{Z}_+^A. \tag{1d}$$

Here, variable x_{ij} represents the salesman using arc ij in the tour. The lower bound obtained by the *linear programming* (LP) relaxation of (1) is sometimes called the Held-Karp bound [21]; we denote this value as z_{HK}.

Recently, [28] proposed a family of bounds for the TSP based on *approximate linear programming* (ALP). Introduced in the mid 1980's and early 1990's [27, 30, 31], ALP is a technique that obtains bounds and policies 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 [9, 18, 20], 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 [13, 14, 15], and has been applied in commodity valuation [25], economic lot scheduling [5], inventory routing [2, 3], joint replenishment [6, 7, 24], knapsack problems [10], revenue management [4], stochastic games [16] and stochastic vehicle routing [29].

The family of ALP bounds for the TSP begins with a bound at least as strong as z_{HK} and successively tightens [28]; the purpose of this paper is to examine the strength of this bound family. Our main contributions involve theoretically studying the bound hierarchy and establishing its relation to various other bounds, including z_{HK}, the optimal TSP value, and another family of bounds stemming from *branch-cut-and-price* (BCP) techniques. BCP has seen empirical success when applied to the *capacitated vehicle routing problem* (CVRP) [17] and the *time-dependent TSP* (TDTSP) [1], and the BCP family of TSP bounds also includes successively tighter relaxations, making it an ideal comparison for ALP. In summary, our theoretical comparison shows three main results:

i) The initial ALP bound is actually equal to z_{HK}; this result settles an open question [28].

ii) We show that roughly $n/2$ steps in the hierarchy suffice to provide a tight bound.

iii) We show that the ALP and BCP bounds are incomparable in general.

We also perform an empirical comparison which shows that an ALP bound is as tight as its BCP analogue, or at worst within about 1%.

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 proof of equivalence between Held-Karp and the basic ALP bound as well as the tightness proof for the final member of the ALP family. In Section 4, we present examples that show that the ALP and BCP bounds are incomparable. Section 5 presents the results of our computational study, and Section 6 concludes outlining future avenues of research.

2 Preliminaries

2.1 Approximate Linear Programming Bounds

The ALP family of lower bounds [28] is based on the DP formulation of the TSP; we present this exact formulation first. The DP uses the observation that when the salesman is at a city i, the only information required to make the decision of which city to visit next is the subset $U \subseteq N$ of cities that has not yet been visited [9, 20]. This pair (i, U) is a DP state with cost-to-go equal to the minimum cost of an i-0 Hamiltonian path through U. The set $\mathcal{S} := \{(i, U) : i \in N,\ U \subseteq N \setminus i\} \cup \{(0, N), (0, \emptyset)\}$ denotes every possible state, and the action set is

$$\begin{aligned}\mathcal{A} := \{((0,N),(i,N\setminus i)) : i \in N\} \cup \{((i,\emptyset),(0,\emptyset)) : i \in N\}\\ \cup \{((i, U \cup j),(j,U)) : i \in N, j \in N \setminus i, U \subseteq N \setminus \{i,j\}\},\end{aligned}$$

where the cost of an action is given by the cost of the corresponding arc. The TSP is then the shortest path problem between $(0, N)$ and $(0, \emptyset)$ in the graph $(\mathcal{S}, \mathcal{A})$, which can be formulated as the linear program

$$\min \sum_{i \in N} \Big(c_{0i} x_{0i} + \sum_{j \in N \setminus i} \sum_{U \subseteq N \setminus \{i,j\}} c_{ij} x_{ij}^U + c_{i0} x_{i0} \Big) \tag{2a}$$

$$\text{s.t.} \sum_{i \in N} x_{0i} = 1 \tag{2b}$$

$$x_{0i} - \sum_{j \in N \setminus i} x_{ij}^{N \setminus \{i,j\}} = 0,\ \forall\, i \in N \tag{2c}$$

$$\sum_{k \in N \setminus (U \cup i)} x_{ki}^U - \sum_{j \in U} x_{ij}^{U \setminus j} = 0,\ \forall i \in N,\ \emptyset \neq U \subsetneq N \setminus i \tag{2d}$$

$$\sum_{k \in N \setminus i} x_{ki}^{\emptyset} - x_{i0} = 0,\ \forall\, i \in N \tag{2e}$$

$$\sum_{i \in N} x_{i0} = 1 \tag{2f}$$

$$x_a \geq 0,\ \forall\, a \in \mathcal{A}.$$

We use abbreviated notation in which x_{0i} and x_{i0} respectively denote moves to and from the depot (i.e. using arcs $((0, N), (i, N \setminus i))$ and $((i, \emptyset), (0, \emptyset))$ in the DP respectively), while x_{ij}^U indicates a move from i to j when the other remaining cities are U; i.e. a move from state $(i, U \cup j)$ to (j, U). The dual is

$$\max\ y_{0,N} - y_{0,\emptyset} \tag{3a}$$

$$\text{s.t.}\ y_{0,N} - y_{i,N \setminus i} \leq c_{0i},\ \forall\, i \in N \tag{3b}$$

$$y_{i,U \cup j} - y_{j,U} \leq c_{ij},\ \forall\, i \in N,\ j \in N \setminus i,\ U \subseteq N \setminus \{i,j\} \tag{3c}$$

$$y_{i,\emptyset} - y_{0,\emptyset} \leq c_{i0},\ \forall\, i \in N \tag{3d}$$

This feasible region contains a line, so without loss of generality we can set $y_{0,\emptyset} = 0$, and then the remaining variables $y_{i,U}$ can be viewed as costs-to-go.

The approach proposed in [28] is to approximate the variables $y_{i,U}$ and solve an "approximate" LP, or ALP, in order to obtain a dual feasible solution (and thus a bound on the optimal TSP value). Given a parameter $t \in \mathbb{Z}$ with $0 \le t \le \frac{n+1}{2}$, we impose the restriction

$$y_{i,U} = \pi_{i,\emptyset} + \sum_{k \in U} \pi_{i,k} + \sum_{\substack{W \subseteq U \\ |W| \ge n-t}} \lambda_{i,W} + \sum_{\substack{W \subseteq N \setminus (U \cup i) \\ |W| \ge n-t}} \mu_{i,W}.$$

The variable $\pi_{i,\emptyset}$ is a cost associated with being at the vertex i. The variable $\pi_{i,k}$ represents an additional cost of being at i and still having to visit city k. Variable $\lambda_{i,W}$ is an analogous cost on certain city subsets W contained in the remaining set U; $\mu_{i,W}$ is a similar variable for certain sets of cities W already visited. The parameter t determines the granularity of our approximation; when t is 0, we do not consider any λ or μ variables, and as t increases we add successively more.

After this approximation, the formulation (3) becomes the weak dual

$$\max \; y_{0,N} \tag{4a}$$

$$\text{s.t. } y_{0,N} - \pi_{i,\emptyset} - \sum_{k \in N \setminus i} \pi_{i,k} - \sum_{\substack{U \subseteq N \setminus i \\ |U| \ge n-t}} \lambda_{i,U} \le c_{0i}, \quad \forall\, i \in N \tag{4b}$$

$$\begin{aligned} &\pi_{i,\emptyset} - \pi_{j,\emptyset} + \pi_{i,j} + \sum_{k \in U} (\pi_{i,k} - \pi_{j,k}) + \sum_{\substack{W \subseteq U \cup j \\ |W| \ge n-t}} \lambda_{i,W} + \sum_{\substack{W \subseteq N \setminus (U \cup \{i,j\}) \\ |W| \ge n-t}} \mu_{i,W} \\ &- \sum_{\substack{W \subseteq U \\ |W| \ge n-t}} \lambda_{j,W} - \sum_{\substack{W \subseteq N \setminus (U \cup j) \\ |W| \ge n-t}} \mu_{j,W} \le c_{ij}, \quad \forall\, i,j \in N,\; U \subseteq N \setminus \{i,j\} \end{aligned} \tag{4c}$$

$$\pi_{i,\emptyset} + \sum_{\substack{U \subseteq N \setminus i \\ |U| \ge n-t}} \mu_{i,U} \le c_{i0}, \quad \forall\, i \in N, \tag{4d}$$

which we denote ALP_t; let z^t_{ALP} be its optimal value. The separation problem for ALP_t requires $O(n^{t+2} + n^3)$ arithmetic operations [28], and therefore ALP_t is solvable in polynomial time for fixed t. In addition, the following result relates the base bound ALP_0 to Held-Karp.

Lemma 1 ([28]). *$z_{HK} \le z^0_{ALP}$.*

To finish this subsection, we present the relaxed primal given by ALP_t, since it will be useful later on. The constraints of the primal can be considered in two classes, those associated with the π variables (5b–5d), and those associated with the λ and μ variables (5e–5f). The primal is

$$\min \sum_{i \in N} \left(c_{0i} x_{0i} + \sum_{j \in N \setminus i} \sum_{U \subseteq N \setminus \{i,j\}} c_{ij} x^U_{ij} + c_{i0} x_{i0} \right) \tag{5a}$$

$$\text{s.t. } \sum_{i \in N} x_{0i} = 1 \tag{5b}$$

$$- x_{0i} + \sum_{j \in N \setminus i} \sum_{U \subseteq N \setminus \{i,j\}} (x^U_{ij} - x^U_{ji}) + x_{i0} = 0, \quad \forall\, i \in N \tag{5c}$$

$$- x_{0i} + \sum_{U \subseteq N \setminus \{i,j\}} x^U_{ij} + \sum_{k \in N \setminus \{i,j\}} \sum_{U \subseteq N \setminus \{i,j,k\}} (x^{U \cup j}_{ik} - x^{U \cup j}_{ki}) = 0 \quad \forall\, i,j \in N \tag{5d}$$

$$
\begin{aligned}
& -x_{0i} + \sum_{j \in N \setminus i} \sum_{\substack{U \subseteq N \setminus \{i,j\} \\ U \cup j \supseteq W}} x_{ij}^U - \sum_{j \in N \setminus i} \sum_{\substack{U \subseteq N \setminus \{i,j\} \\ U \supseteq W}} x_{ji}^U = 0 && \text{(5e)} \\
& \qquad \forall i \in N, W \subseteq N \setminus i : |W| \geq n - t \\
& x_{i0} + \sum_{j \in N \setminus i} \sum_{\substack{U \subseteq N \setminus \{i,j\} \\ N \setminus (U \cup \{i,j\}) \supseteq W}} x_{ij}^U - \sum_{j \in N \setminus i} \sum_{\substack{U \subseteq N \setminus \{i,j\} \\ N \setminus (U \cup i) \supseteq W}} x_{ji}^U = 0 && \text{(5f)} \\
& \qquad \forall i \in N, W \subseteq N \setminus i : |W| \geq n - t \\
& x \geq 0.
\end{aligned}
$$

2.2 Branch-Cut-and-Price

In [17], a BCP algorithm was introduced for the CVRP, an NP-hard problem that generalizes the TSP. The approach was computationally successful and has been extended and applied to other closely related variants of the TSP, such as the TDTSP [1]. Here, we present the bounds in the context of the TSP. The idea of the BCP formulation is to combine the traditional arc formulation with a formulation involving an exponential number of variables. This is achieved by considering the set Q of *n-paths* [22].

Definition 2. Given the graph $G = (V = \{0\} \cup N, A)$, an *n-path* q of G is a sequence of vertices $v_q(0), v_q(1), ..., v_q(n+1)$ such that $v_q(0) = v_q(n+1) = 0$, $v_q(i) \in N, \ \forall \ i = 1, \ldots, n$ and $v_q(i) \neq v_q(i+1), \forall i = 1, \ldots, n-1$.

Informally, an n-path is a directed walk with $n+1$ arcs that begins and ends at the depot 0, and does not otherwise visit it. The BCP formulation connects n-paths with the standard arc-based formulation (1) by introducing the coefficient d_{ij}^q:

$$
d_{ij}^q := \sum_{k=1}^{n+1} \mathbb{1}_{\{v_q(k-1)=i, v_q(k)=j\}} \qquad \forall \ i, j \in V, \ q \in Q
$$

In other words, d_{ij}^q is the number of times an n-path q uses the arc (i, j). We can now write the BCP formulation for the TSP as

$$
\begin{aligned}
\min \ & \sum_{a \in A} c_a x_a && && \text{(6a)} \\
\text{s.t.} \ & x_a = \sum_{q \in Q} d_a^q \xi_q && \forall \ a \in A && \text{(6b)} \\
& \sum_{a \in \delta^+(i)} x_a = \sum_{a \in \delta^-(i)} x_a = 1 && \forall i \in V && \text{(6c)} \\
& \sum_{a \in \delta^+(U)} x_a \geq 1 && \forall \ \emptyset \subsetneq U \subsetneq N && \text{(6d)} \\
& 0 \leq \xi_q \leq 1 && \forall \ q \in Q && \text{(6e)} \\
& x \in \mathbb{Z}_+^A.
\end{aligned}
$$

This formulation imposes the Held-Karp constraints plus the requirement that x must be written as a combination of n-paths using the ξ_q variables. We call the resulting formulation BCP.

The pricing problem for the ξ variables can be solved in $O(n^3)$ time via DP, and thus the LP relaxation can be solved in polynomial time as well. (Though the polyhedron has exponentially many variables and constraints, it can be shown [19] that the projection to the x variables can be separated over in polynomial time.) Moreover, instead of pure n-paths, one can strengthen the formulation by considering only a subset of Q through cycle elimination. Since eliminating cycles entirely from the n-paths is NP-Hard, one can instead use *t-cycle-free* n-paths, for fixed values of t, where a walk is t-cycle-free if it contains no cycles of vertices in N of length t or less, including 2-cycles (directed walks of the form $i - j - i$). Formally, we say that an n-path q is t-cycle free if the vertices $\{v_q(k), v_q(k+1), ..., v_q(k+t)\}$ are all distinct for every $0 \le k \le n-t+1$. The pricing algorithm for t-cycle-free n-paths adds some extra states in the DP, but can be done in $O(t \cdot t!^2 \cdot n^3)$ time, which is polynomial for fixed t [23]. We can thus replace the set Q of n-paths with the set Q^t of t-cycle-free n-paths in (6), for $t \ge 1$, noting that $Q^1 = Q$. We call the associated LP relaxation the BCP$_t$ formulation, yielding the z^t_{BCP} lower bound for the TSP.

The definition of t-cycle-free n-paths still allows for the existence of one cycle of length t or less involving the depot. This formulation was derived from the CVRP context, where such cycles are actually allowed in feasible solutions; by allowing them, the pricing problem for the LP relaxation of (6) becomes simpler.

3 On the strength of ALP$_t$

The focus of this section is to show that the ALP$_t$ family of bounds starts with a bound that is equal to z_{HK} for $t = 0$ and ends with a bound that is equal to the convex hull for $t = \lfloor \frac{n+1}{2} \rfloor$. We start by showing that the ALP$_0$ bound proposed in [28] is actually equal to the Held-Karp bound.

Theorem 3. *$z_{HK} = z^0_{ALP}$.*

Proof. Consider the formulation for ALP$_0$ given by (4) without λ or μ variables. For each ordered triple $i \in N$, $j \in N \setminus i$, $k \in N \setminus \{i, j\}$, define an auxiliary variable $\eta^k_{ij} \ge 0$. We can rewrite (4) as

$$
\begin{aligned}
\max\ & y_{0,N} \\
\text{s.t.}\ & y_{0,N} - \pi_{i,\emptyset} - \sum_{k \in N \setminus i} \pi_{i,k} \le c_{0i}, \quad \forall\ i \in N \\
& \pi_{i,\emptyset} - \pi_{j,\emptyset} + \pi_{i,j} + \sum_{k \in N \setminus \{i,j\}} \eta^k_{ij} \le c_{ij}, \quad \forall\ i, j \in N, \\
& \pi_{i,\emptyset} \le c_{i0}, \quad \forall\ i \in N \\
& \pi_{i,k} - \pi_{j,k} - \eta^k_{ij} \le 0, \forall\ i, j, k \in N \\
& \eta^k_{ij} \ge 0, \forall\ i, j, k \in N,
\end{aligned}
$$

where at optimality we can take $\eta^k_{ij} = \max\{0, \pi_{i,k} - \pi_{j,k}\}$. This formulation's dual is equivalent to the multi-commodity flow relaxation of the TSP originally proposed in [11]; when projected to the space of x_{ij} variables, the multi-commodity flow relaxation's feasible region is known to be (1)'s polyhedron, see e.g. [26]. □

Theorem 3 confirms that the equivalence between the Held-Karp and ALP bounds is constructive in the following sense: The proof of Lemma 1 [28] gives a construction whereby any feasible solution of the Held-Karp dual formulation can be transformed into a feasible solution of (4) 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

	Held-Karp		ALP
primal	(1)	$\xleftarrow{(7)}$	(5)
	$\Updownarrow$		$\Updownarrow$
dual	dual of (1)	$\xrightarrow{[28]}$	(4)

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

(4) to optimality, Theorem 3 ensures that the unique transformation in the reverse direction,

$$\sum_{U \subseteq N \setminus \{i,j\}} x_{ij}^U = x_{ij}, \tag{7}$$

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.

We also note that the equivalence between Held-Karp and the base ALP bound is striking because the former uses quadratically many variables and exponentially many constraints in its primal formulation, whereas the latter uses quadratically many variables and exponentially many constraints in the *dual* space.

We end this section by focusing on the other end of the spectrum for the ALP_t relaxation; that is, the case when $t = \lfloor \frac{n+1}{2} \rfloor$.

Theorem 4. *For $t = \lfloor \frac{n+1}{2} \rfloor$, ALP_t is an exact formulation.*

Proof. Replacing the value of t, we get the state representation

$$y_{i,U} = \pi_{i,\emptyset} + \sum_{k \in U} \pi_{i,k} + \sum_{\substack{W \subseteq U \\ |W| \geq \frac{n-1}{2}}} \lambda_{i,W} + \sum_{\substack{W \subseteq N \setminus (U \cup i) \\ |W| \geq \frac{n-1}{2}}} \mu_{i,W}. \tag{8}$$

We can consider this set of equations as

$$y = B \begin{pmatrix} \pi \\ \lambda \\ \mu \end{pmatrix} \tag{9}$$

for some matrix B. If B has full row rank, then given any y that is feasible for the original dual (3), there exists (π, λ, μ) that is feasible for (4) and satisfying (9), thus showing that the optimal value of (4) is greater than or equal to the optimal value of (3), which gives us the desired result. Thus, it suffices to show that B has full row rank.

For two vertices i and j, if i is not equal to j, they are represented by two disjoint sets of variables, i.e. (9) can be rewritten as

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} B_0 & 0 & 0 & \dots & 0 \\ 0 & B_1 & 0 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & B_n \end{pmatrix} \begin{pmatrix} \pi_0 \\ \lambda_0 \\ \mu_0 \\ \hline \pi_1 \\ \lambda_1 \\ \mu_1 \\ \hline \vdots \\ \hline \pi_n \\ \lambda_n \\ \mu_n \end{pmatrix} \tag{10}$$

where $y_i, \pi_i, \lambda_i, \mu_i$ are the blocks of variables corresponding to vertex i. Thus, if all blocks in B corresponding to the states for each vertex have full row rank, B itself has full row rank and the result follows.

In matrix B_i, there is one row per variable $y_{i,U}$, and thus one row for every subset $U \subseteq N \setminus i$. Moreover, B has one column per variable λ and μ (we will not focus on the π variables here), that is, for every $W \subseteq N \setminus i$ such that $|W| \geq \frac{n-1}{2}$ there is a column for $\lambda_{i,W}$ and one for $\mu_{i,W}$. Finally note that variable $\lambda_{i,W}$ appears in row $y_{i,U}$ if and only if $W \subseteq U$ and variable $\mu_{i,W}$ appears in row $y_{i,U}$ if and only if $W \subseteq N \setminus (U \cup i)$.

Suppose for the sake of contradiction that there is a state $y_{i,U'}$ that can be written as a linear combinations of other states; i.e. suppose that

$$b_{i,U'} = \sum_{U' \neq W \subseteq (N \setminus i)} \alpha_W b_{i,W} \tag{11}$$

for some $i \in N$, $U' \subseteq N \setminus i$ with at least one α_W nonzero, where $b_{i,U}$ is the row of the submatrix B_i corresponding to variable $y_{i,U}$. If $|U'| \geq \frac{n-1}{2}$, the coefficient of $\lambda_{i,U'}$ is one in $b_{i,U'}$ and thus equation (11) holds only if there is a set $W \supsetneq U'$ such that $\alpha_W \neq 0$. Let W be the largest set such that $\alpha_W \neq 0$ and $W \supsetneq U'$. However, the coefficient of $\lambda_{i,W}$ in $b_{i,W}$ is 1, but it is 0 in $b_{i,U'}$, requiring yet another $W' \supsetneq W$ such that $\alpha_{W'}$ is nonzero, which contradicts the choice of W.

On the other hand, if $|U'| < \frac{n-1}{2}$, then $|N \setminus (U' \cup i)| > \frac{n-1}{2}$ and the coefficient of $\mu_{i,N \setminus (U' \cup i)}$ must be one in $b_{i,U'}$. This implies that α_W is nonzero for some $W \subsetneq U'$, and the proof follows as in the previous case. □

4 Incomparability of BCP and ALP

Having established the behavior of the two extreme members of the ALP hierarchy, we next study the hierarchy's relationship to the BCP bound family. We start giving a series of examples where BCP_1 is strictly better than ALP_t for any $t < \lfloor \frac{n+1}{2} \rfloor$, which shows that BCP_{t+k} can be much better than ALP_t. On the other hand, we show afterwards that there are also instances for which ALP_t can be better than BCP_{t+1} or even BCP_{t+k} for larger values of k.

4.1 BCP_1 can be strictly better than ALP_t

We construct a family of instances where the bound obtained from BCP_1 is strictly greater than ALP_t for $t < \lfloor \frac{n+1}{2} \rfloor$. We start with an example for $t = 0$ based on the solution to the LP relaxation of the Held-Karp formulation (1) given in Figure 1. The example shows a solution that can be readily verified to be feasible for (1), with dashed arcs representing variables with value 1/2 and solid arcs representing variables with value 1. If we let the costs of arcs $(5, 2)$ and $(1, 3)$ be 0, the costs of all other drawn arcs be 1, and give all other arcs sufficiently large cost, the solution then has value 5, which by Theorem 3 implies $z^0_{ALP} \leq 5$. On the other hand, it can also be readily verified that the only possible n-path that exclusively uses the arcs in Figure 1 is 0,4,5,3,1,2,0, with a cost of 6. Therefore, the optimal solution to BCP_1 must be equal to 6, which is strictly greater than the optimal solution to ALP_0.

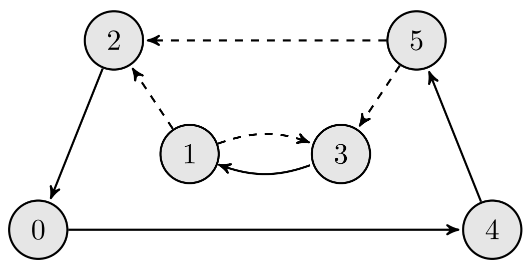

Figure 1: An example showing the strict dominance of BCP_1 to ALP_0. Solid arcs have value 1 in the Held-Karp solution, while dashed arcs have value $\frac{1}{2}$.

The previous example can be extended to show that BCP_1 can strictly dominate ALP_t. Consider the instance presented in Figure 2. As in the previous example, we let the costs of arcs $(2t+4, 2t+1)$ and $(2t + 2, 2t + 3)$ be 0, the costs of all other drawn arcs be 1, and give all other arcs sufficiently large cost. The solution depicted has variables equal to 1 for solid arcs and 1/2 for dashed arcs and has a cost of $2t + 5$. It can also be readily verified that the only possible n-path that exclusively uses the arcs in Figure 1 is $0, 1, \ldots, t, 2t + 5, 2t + 4, \ldots, t + 1, 0$, with a cost of $2t + 6$. Therefore, the optimal solution to BCP_1 (and hence to $\text{BCP}_{t'}$ for any t') must be equal to $2t + 6$.

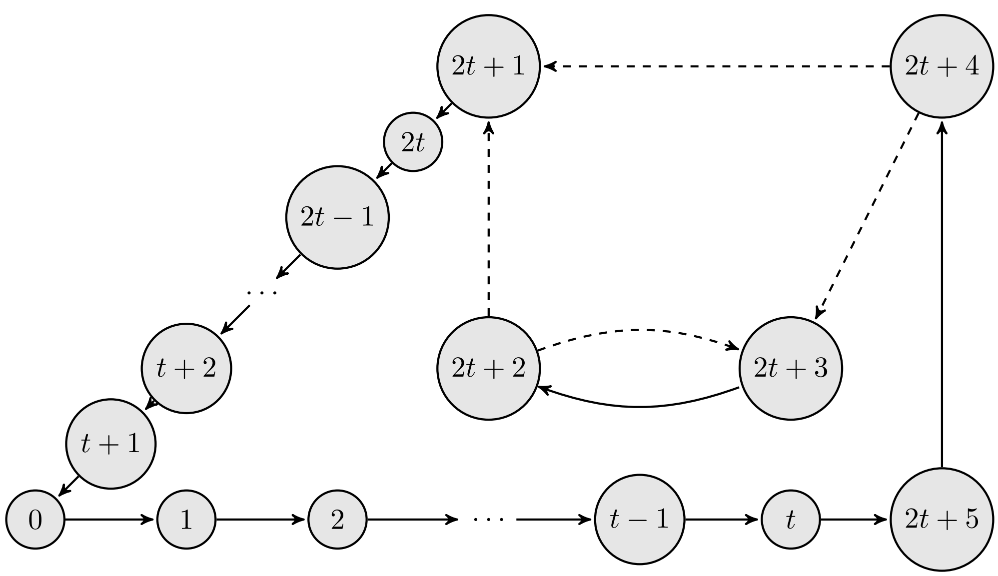

Figure 2: An example showing the strict dominance of BCP_1 to ALP_t. Solid arcs have value 1 in the solution, while dashed arcs have value $\frac{1}{2}$.

We also need to establish that the solution of Figure 2 is feasible for the ALP_t formulation. For that purpose, we can set the following variables in the ALP_t formulation to match the depicted solution, which implies that $z^t_{ALP} \leq 2t+5$:

$$\begin{aligned}
x_{01} &= 1 \\
x^{\{i+2,\dots,2t+5\}}_{i,i+1} &= 1, && \forall\ i = 1,\dots,t-1 \\
x^{\{t+1,\dots,2t+4\}}_{t,2t+5} &= 1 \\
x^{\{t+1,\dots,2t+3\}}_{2t+5,2t+4} &= 1 \\
x^{\{t+1,\dots,2t+2\}}_{2t+4,2t+3} &= 1/2 \\
x^{\{t+1,\dots,2t+1\}}_{2t+3,2t+2} &= 1 \\
x^{\{t+1,\dots,2t\}}_{2t+2,2t+1} &= 1/2 \\
x^{\{t+1,\dots,2t\}}_{2t+4,2t+1} &= 1/2 \\
x^{\{t+1,\dots,2t\}}_{2t+2,2t+3} &= 1/2 \\
x^{\{t+1,\dots,t+i-1\}}_{t+i+1,t+i} &= 1, && \forall\ i = 1,\dots,t \\
x_{t+1,0} &= 1.
\end{aligned}$$

It is not hard to verify then that the only constraints of (2) that are being violated by the solution are the ones corresponding to the constraints (2d) for the following values of (i,U) and with the corresponding violations (left-hand side minus right-hand side):

i) $i = 2t+4, U = \{t+1,\dots,2t+3\}$ (violation 0.5)

ii) $i = 2t+4, U = \{t+1,\dots,2t+1\}$ (violation -0.5)

iii) $i = 2t+2, U = \{t+1,\dots,2t+1,2t+3\}$ (violation -0.5)

iv) $i = 2t+2, U = \{t+1,\dots,2t+1\}$ (violation 0.5) .

We next examine the constraints for (5). Constraint (5c) can be rewritten as

$$x_{0i} - \sum_{j\in N\setminus i} x^{N\setminus\{i,j\}}_{ij} - \sum_{j\in N\setminus i}\sum_{U\subsetneq N\setminus\{i,j\}} x^U_{ij} + \sum_{j\in N\setminus i}\sum_{\emptyset\subsetneq U\subseteq N\setminus\{i,j\}} x^U_{ji} + \sum_{j\in N\setminus i} x^{\emptyset}_{ji} - x_{i0} = 0$$

and equivalently as

$$x_{0i} - \sum_{j\in N\setminus i} x^{N\setminus\{i,j\}}_{ij} - \sum_{\emptyset\subsetneq U\subsetneq N\setminus i}\sum_{j\in U} x^{U\setminus j}_{ij} + \sum_{\emptyset\subsetneq U\subsetneq N\setminus i}\sum_{j\in N\setminus(U\cup i)} x^U_{ji} + \sum_{j\in N\setminus i} x^{\emptyset}_{ji} - x_{i0} = 0,$$

which implies that constraint (5c) is the sum of constraint (2c), (2e) and all constraints (2d) for a fixed i. Since the violations above offset each other for a fixed i, we have that (5c) is satisfied.

Similarly, constraint (5d) can be rewritten as

$$x_{0i} - \sum_{k\in N\setminus i} x^{N\setminus\{i,k\}}_{ik} + x^{N\setminus\{i,j\}}_{ij} - \sum_{k\in N\setminus\{i,j\}}\sum_{U\subsetneq N\setminus\{i,j,k\}} x^{U\cup j}_{ik} - \sum_{U\subseteq N\setminus\{i,j\}} x^U_{ij} + \sum_{k\in N\setminus\{i,j\}}\sum_{U\subseteq N\setminus\{i,j,k\}} x^{U\cup j}_{ki} = 0,$$

and the result follows.

The above discussion proves the following result.

Proposition 5. *There exist instances where* $z^1_{BCP} > z^t_{ALP}$ *for* $0 \leq t < \lfloor \frac{n+1}{2} \rfloor$.

4.2 ALP$_t$ can be strictly better than BCP$_{t+k}$

First, we show a family of graphs for which BCP has a weak bound unless t is large, which will give us the following theorem.

Theorem 6. *Let C be a positive real number and $n \geq 5$. There exists a graph with n vertices such that the integrality gap is at least C for BCP_t with $t \leq n-3$.*

Proof. Let $G(n, C) = C_n \cup \{(1, n-2), (n-1, 2)\}$, where C_n is a bi-directed cycle with n vertices appearing in increasing order. Let all arc costs be 1, except for $(n-1, n-2)$ and $(1, 2)$, which cost $nC - n + 1$ (see Figure 3). We complete the instance setting remaining edge costs to a sufficiently large number $M > nC$.

The optimal tour has cost nC, which can be derived as follows. First, note that the two directed cycles in C_n give cost nC, so the optimal tour has cost at most nC. Therefore, the optimal tour does not use any of the arcs with cost M. Since any feasible tour must enter and leave the depot, the optimal tour must then use either the pair of arcs $(n-1, 0)$ and $(0, 1)$ or the pair of arcs $(0, n-1)$ and $(1, 0)$.

Now if a feasible tour uses the arcs $(n-1, 0)$ and $(0, 1)$, and does not use arcs of cost M, then it must enter vertex $n-1$ through arc $(n-2, n-1)$. But then, it cannot use $(1, n-2)$, since that creates a subtour $(0, 1, n-2, n-1, 0)$. Thus, the only feasible tour that uses arcs $(n-1, 0)$ and $(0, 1)$ is one using one of the directed cycles in C_n. The case that uses arcs $(1, 0)$ and $(0, n-1)$ is symmetric and the argument is analogous.

To show the integrality gap, let $q_1 = (0, 1, n-2, n-3, \ldots, 3, 2, 1, 0)$ and $q_2 = (0, n-1, 2, 3, \ldots, n-2, n-1, 0)$ be two t-cycle-free n-paths (see Figure 4) and let $\overline{x}$ be a solution to (6) obtained by setting $\xi_{q_1} = \xi_{q_2} = 0.5$.

We first check that $\overline{x}$ is actually feasible for (6). It is clear that $\overline{x}$ satisfies the degree constraints. Now pick a set $\emptyset \subset U \subseteq N$. Let $supp(\overline{x}) := \{ij \in A : \overline{x}_{ij} > 0\}$. Note that every arc in $supp(\overline{x})$ has value 0.5. Also note that the arcs in $supp(\overline{x})$ are exactly the arcs appearing in either q_1 or q_2. Denote that set of arcs by A_n. If we can show that $G' := (V, A_n)$ is 2-arc-connected, then we have that $\overline{x}(\delta^+(U)) = 0.5|\delta^+(U) \cap A_n| \geq 1$, so subtour elimination is satisfied.

It is easy to argue that C_n is 2-arc-connected. So if $\delta^+(U) \cap A_n \supseteq \delta^+(U) \cap C_n$, then $|\delta^+(U) \cap A_n| \geq 2$. But then, the only arcs that are in C_n, but not in A_n are $(n-1, n-2)$ and $(1, 2)$. Therefore, for the inclusion above to be violated, we must have either

i) $n-1 \in U$ and $n-2 \notin U$, or

ii) $1 \in U$ and $2 \notin U$.

For (i), either $U = \{n-1\}$, in which case $(n-1, 2)$ and $(n-1, 0)$ in $\delta^+(U) \cap A_n$, or there exists $v \notin \{0, n-1\}$ such that $v \in U$. But then, there exists an arc in q_1 that is in $\delta^+(U)$ since there exists a $v-0$ path in q_1 and there exists an arc in q_2 that is in $\delta^+(U)$ since there exists a $(n-1)-0$ path in q_2; so either way $|\delta^+(U) \cap A_n| \geq 2$. The proof of (ii) is analogous.

Having established that $\overline{x}$ is feasible for (6), we note that the cost of that solution is n, which implies that $z^t_{BCP} \leq n$, which implies the integrality gap is at least C. □

The previous example coupled with Theorem 4 implies a reverse result to Proposition 5. More precisely, it cannot be the case that for a given fixed $k \geq 0$, BCP$_{t+k}$ dominates ALP$_t$ for every t.

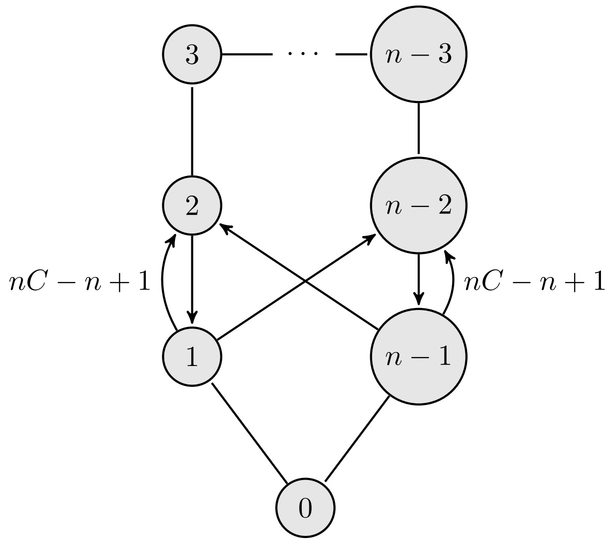

Figure 3: Generic construction for a graph with n vertices for which the BCP_{n-3} solution has integrality gap at least C.

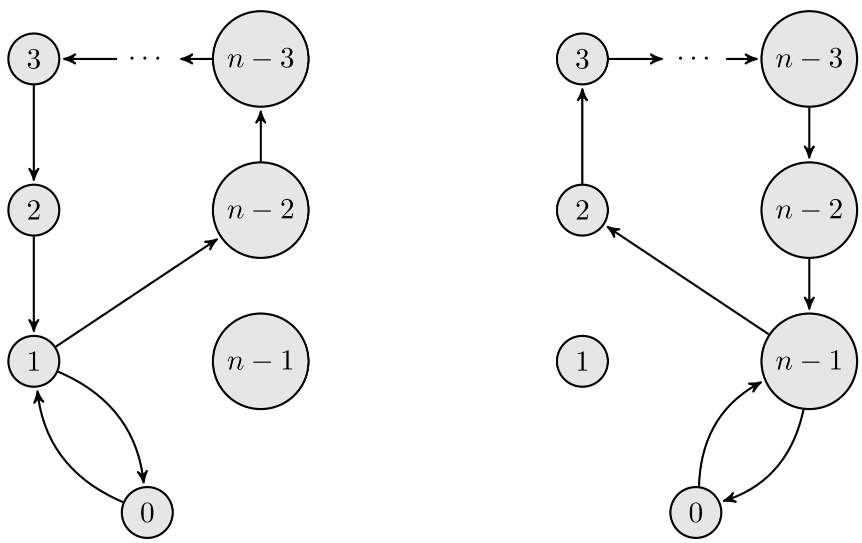

Figure 4: $(n-3)$-cycle-free n-paths q_1 and q_2 that yield a solution to (6) of value n

5 Empirical comparison to BCP

Having established in Section 4 that no dominance relationship exists between the ALP and BCP families of bounds, we compared the two empirically by evaluating the lower bounds obtained by each formulation for increasing values of t in a few benchmark instances. This section presents the description of these experiments as well as their results. All experiments were conducted in an AMD server, with 48 cores of 2.3GHz and 256 GB of RAM. The model was generated using C++ and solved with CPLEX 12.6.

One of our main difficulties was finding instances for which the Held-Karp bound is not tight, but the bounds z^t_{BCP} and z^t_{ALP} could also be obtained in a reasonable amount of time. Due to this problem, even though we have fully functional implementations, we were only able to test a few instances.

The first set of instances were from TSPLIB. There are very few of these instances with n small enough for which the formulations BCP_t and ALP_t can be tested with nontrivial values of t. From the ones we were able to test, only `bayg29` and `bays29` had a non-optimal Held-Karp bound. The instance `rand25` was generated using CONCORDE, by selecting random Euclidean points to be vertices and rounding the pairwise distances to the nearest integer. Finally, instance `f6` is composed of two triangles and a matching between them (6 vertices and 18 arcs), using shortest path distances between non-adjacent vertices. Table 2 shows the bounds for our first set of instances.

Since other TSPLIB instances took too long to solve, we performed the following experiment. We generated instances for a specific value of n by uniformly sampling n points from a 100 by 100 grid and rounding distances to the nearest integer, until we found one where the Held-Karp bound and the optimal value were different. These are the instances used to create Table 3. This was similar to the process used to generate `rand25`, but we used a smaller number of vertices since we wanted to test for larger values of t. The table is missing entries corresponding to bounds we were not able to compute.

Instance	Formulation							
	Optimal	Held-Karp	ALP			BCP		
			$t=0$	$t=1$	$t=2$	$t=1$	$t=2$	$t=3$
`bayg29`	1610.00	1608.00	1608.00	1608.00	1608.00	1608.00	1610.00	1610.00
`bays29`	2020.00	2013.50	2013.50	2013.50	2013.50	2013.50	2019.64	2020.00
`f6`	8.00	6.00	6.00	6.67	8.00	6.00	8.00	8.00
`rand25`	1071.00	1062.50	1062.50	1062.50	1062.50	1062.5.0	1064.50	1067.6667

Table 2: Bounds given by Held-Karp, ALP and BCP formulations for small t

# vertices	**Optimal Value**	**Formulation**	**Bound**					
			t=0	t=1	t=2	t=3	t=4	t=5
12	36	ALP_t	35.50	35.50	35.50	35.57	36.00	36.00
		BCP_{t+1}	35.50	36.00	36.00	36.00	36.00	36.00
13	44	ALP_t	43.50	43.50	43.50	43.50	43.50	44.00
		BCP_{t+1}	43.50	44.00	44.00	44.00	44.00	44.00
14	52	ALP_t	51.50	51.50	51.50	51.50	51.50	51.50
		BCP_{t+1}	51.50	51.50	51.50	51.50	51.50	51.50
15	49	ALP_t	48.50	48.50	48.50	48.50	48.58	-
		BCP_{t+1}	48.50	49.00	49.00	49.00	49.00	49.00
16	48	ALP_t	47.50	47.50	47.50	47.50	47.50	-
		BCP_{t+1}	47.50	47.83	48.00	48.00	48.00	48.00
17	65	ALP_t	64.50	64.50	64.50	64.50	64.50	-
		BCP_{t+1}	64.50	65.00	65.00	65.00	65.00	65.00

Table 3: Bounds for ALP and BCP, and optimal values for random Euclidean instances as a function of number of vertices

Notice that we compare ALP_t and BCP_{t+1} (rather than comparing ALP_t with BCP_t) because the BCP_t bound keeps track of the last $t-1$ vertices visited, i.e. sets U such that $|U| \le t-1$.

Similarly, ALP_t takes into account sets U such that $|U| \leq t$. In addition, ALP_t is defined starting with $t = 0$, while BCP_t is defined starting with $t = 1$.

The results show that BCP_{t+1} is always at least as good as ALP_t, but the bounds' difference is quite small – around 1% in a few cases and usually smaller or zero. The one exception is `f6`, where in one of three cases the difference is larger. (This instance's pathological behavior is well studied, as it is used to construct an instance collection giving the worst known gap for Held-Karp.) The two bounds' closeness, coupled with BCP's empirical success in models related to the TSP, suggest that ALP may be useful to empirically bound the TSP and other routing models. However, it is worth remarking that the nature of the experiments limited our choice of instance to smaller problems, and the bound families' behavior may be different when the number of vertices grows larger.

6 Conclusions

In this work, we studied the strength of a new ALP family of relaxations for the TSP. We have shown that ALP_0 is equivalent to the Held-Karp bound, a question which was left open in [28]. We also showed that $\text{ALP}_{\lfloor (n+1)/2 \rfloor}$ is tight. We also showed that the ALP and BCP families are incomparable, i.e. neither family can dominate the other in terms of bound strength. Our empirical results show that each tested ALP bound is comparable to its BCP counterpart, though sometimes the ALP bound is slightly lower.

Our study suggests possible improved results or modifications for either family. For instance, it may be possible to prove a more strict relationship between the ALP and BCP families in some special cases. Specifically, the counterexamples from Section 4 have asymmetric costs. If we assume the instances have symmetric costs (which is indeed the case in the tested instances), it may be possible to establish dominance.

An interesting potential direction is to consider combining the ALP and BCP bounds. One way would be to provide an enhanced ALP approximation that can match the BCP bounds. Another would be to somehow include the n-path restriction directly into the ALP relaxed primal, which would involve two kinds of column generation procedures working in tandem. In either case, progress along these lines could suggest new efficient bounds for the TSP as well as related routing problems where BCP has seen success.

References

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

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

[3] D. Adelman. A Price-Directed Approach to Stochastic Inventory/Routing. *Operations Research*, 52:499–514, 2004.

[4] D. Adelman. Dynamic Bid Prices in Revenue Management. *Operations Research*, 55:647–661, 2007.

[5] D. Adelman and C. Barz. A Unifying Approximate Dynamic Programming Model for the Economic Lot Scheduling Problem. *Mathematics of Operations Research*, 39:374–402, 2014.

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

[7] D. Adelman and D. Klabjan. Computing Near-Optimal Policies in Generalized Joint Replenishment. *INFORMS Journal on Computing*, 24:148–164, 2011.

[8] 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.

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

[10] D. Blado, W. Hu, and A. Toriello. Semi-Infinite Relaxations for the Dynamic Knapsack Problem with Stochastic Item Sizes. *SIAM Journal on Optimization*, 2016. Forthcoming.

[11] A. Claus. A New Formulation for the Travelling Salesman Problem. *SIAM Journal on Algebraic Discrete Methods*, 5:21–25, 1984.

[12] G. Dantzig, D. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2:393–410, 1954.

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

[14] D.P. de Farias and B. van Roy. On Constraint Sampling in the Linear Programming Approach to Approximate Dynamic Programming. *Mathematics of Operations Research*, 29:462–478, 2004.

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

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

[17] R. Fukasawa, H. Longo, J. Lysgaard, M. de Aragão, M. Reis, E. Uchoa, and R. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106:491–511, 2006.

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

[19] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, second corrected edition, 1993.

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

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

[22] D. Houck, J.C. Picard, M. Queyranne, and R.R. Vemuganti. The Traveling Salesman Problem and Shortest n-paths. Technical report, University of Maryland, 1977.

[23] S. Irnich and D. Villeneuve. The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing*, 18:391–406, 2006.

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

[25] S. Nadarajah, F. Margot, and N. Secomandi. Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 2014. Forthcoming.

[26] M. Padberg and T.-Y. Sung. An analytical comparison of different formulations of the travelling salesman problem. *Mathematical Programming*, 52:315–357, 1991.

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

[28] A. Toriello. Optimal Toll Design: A Lower Bound Framework for the Asymmetric Traveling Salesman Problem. *Mathematical Programming*, 144:247–264, 2014.

[29] A. Toriello, W.B. Haskell, and M. Poremba. A Dynamic Traveling Salesman Problem with Stochastic Arc Costs. *Operations Research*, 62:1107–1125, 2014.

[30] 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.

[31] M.A. Trick and S.E. Zin. Spline Approximations to Value Functions: A Linear Programming Approach. *Macroeconomic Dynamics*, 1:255–277, 1997.