Computing Bounds and Solutions to the Asymmetric Traveling Salesman Problem with Approximate Linear Programming

Michael Poremba   Alejandro Toriello
Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California
{poremba, toriello} *at* usc *dot* edu

Background

Problem Statement

- We consider the asymmetric *traveling salesman problem* (TSP) over cities in $N \cup 0$, where the cost from city i to city j is $c_{i,j} \in \mathbb{R}$ for $i, j = 0, \ldots, |N|$ with $i \neq j$.
- In previous work[(1)], the second author:
  - applied *approximate linear programming* (ALP) techniques to the TSP's *dynamic program* (DP) formulation
  - found a nested family of polyhedral lower bounds, solvable in polynomial time
- In this poster, we present:
  - two methods for solving these ALP formulations to compute TSP lower bounds
  - a branch-and-bound heuristic which uses solutions to the ALP to generate TSP tours via a *price-directed* policy

Lower Bound Framework

Dual of the Dynamic Programming Formulation:

$$\begin{aligned} \max\ & y_{0,N} && \text{(1a)}\\ \text{s.t. } & y_{0,N} - y_{i,N\setminus i} \leq c_{0i}, \quad \forall\, i \in N && \text{(1b)}\\ & y_{i,U\cup j} - y_{j,U} \leq c_{ij}, \quad \forall\, i \in N,\ j \in N \setminus i,\ U \subseteq N \setminus \{i,j\} && \text{(1c)}\\ & y_{i,\emptyset} \leq c_{i0}, \quad \forall\, i \in N && \text{(1d)}\\ & y_{0,N} \in \mathbb{R}; \quad y_{i,U} \in \mathbb{R}, \forall\, i \in N, U \subseteq N \setminus i. && \text{(1e)} \end{aligned}$$

Approximate Cost-to-Go:

$$y_{i,U} = \pi_{i,\emptyset} + \sum_{k\in U} \pi_{i,k} + \sum_{\substack{W\subseteq U\\ |W|\geq n-t}} \lambda_{i,W} + \sum_{\substack{W\subseteq N\setminus(U\cup i)\\ |W|\geq n-t}} \mu_{i,W} \quad \text{for } t \geq 0 \qquad (2)$$

Approximate Linear Programming Formulation:

$$\begin{aligned} \max\ & y_{0,N} && \text{(3a)}\\ \text{s.t. } & y_{0,N} - \pi_{i,\emptyset} - \sum_{k\in N\setminus i} \pi_{i,k} \leq c_{0i}, \quad \forall\, i \in N && \text{(3b)}\\ & \pi_{i,\emptyset} - \pi_{j,\emptyset} + \pi_{i,j} + \sum_{k\in U} (\pi_{i,k} - \pi_{j,k}) \leq c_{ij}, \\ & \qquad \forall\, i \in N, j \in N \setminus i,\ U \subseteq N \setminus \{i,j\} && \text{(3c)}\\ & \pi_{i,\emptyset} \leq c_{i0}, \quad \forall\, i \in N && \text{(3d)}\\ & y_{0,N} \in \mathbb{R}; \quad \pi_{i,\emptyset} \in \mathbb{R}\ \forall\, i \in N,\ \pi_{i,j} \in \mathbb{R}\ \forall\, i \in N, j \in N \setminus i && \text{(3e)} \end{aligned}$$

Note: The above formulation is for the case where $t = 0$. It becomes slightly more complex if $t > 0$. This poster encompasses results where $t = 0$ or $t = 1$.

References

(1) Toriello, A. Optimal Toll Design: A Lower Bound Framework for the Asymmetric Traveling Salesman Problem. Preprint available at $www - bcf.usc.edu/\ toriello/tsp\_bound.pdf$, 2012.

Computing Bounds

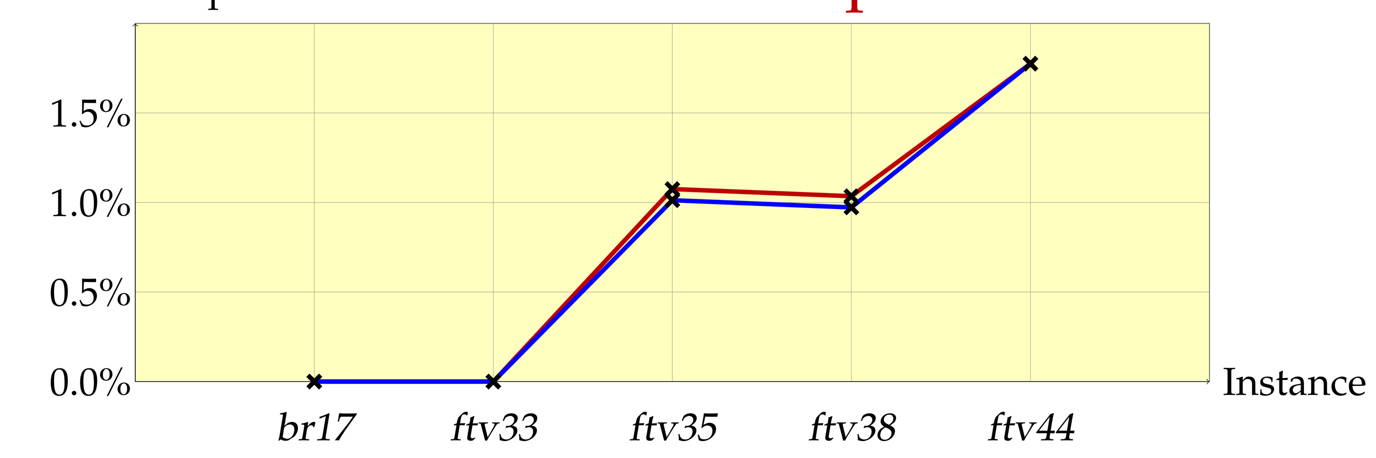
Constraint Generation Algorithm

Method:
This approach solves the formulation directly, using a constraint generation procedure to manage the exponential number of constraints.

Computational Challenges:
In previous work of the second author, he outlines a polynomial-time separation routine for our formulation. Computationally, the main challenge came down to managing the number of constraints to add/remove each iteration.

Results:

Lower Bound Comparison



RED PLOT = bound for the linear relaxation of the arc-based formulation of the TSP
BLUE PLOT = bound for our ALP where t = 1

Note: For the *ftv35* and *ftv38* instances, the bound improvement is roughly 0.06%.

Primal-Dual Algorithm

Method:
This method takes any given feasible solution and attempts to improve it by solving an auxiliary problem in polynomial time to determine a potential direction of improvement.

Computational Challenges:
In order to identify tight constraints in polynomial time, we perform the following for each $i \in N,\ j \in N\setminus i,$ and $k \in N\setminus\{i,j\}$:

1. If $\pi_{i,k} - \pi_{j,k} > 0$, then $k \in U^{+}_{i,j}$.
2. If $\pi_{i,k} - \pi_{j,k} < 0$, then $k \in U^{-}_{i,j}$.
3. If $\pi_{i,k} - \pi_{j,k} = 0$, then $k \in U^{=}_{i,j}$.

To construct the direction LP, include the following class of constraints:

$$\hat{\pi}_{i,\emptyset} - \hat{\pi}_{j,\emptyset} + \hat{\pi}_{i,j} + \textstyle\sum_{k\in U^{+}_{i,j}} (\hat{\pi}_{i,k} - \hat{\pi}_{j,k}) + \sum_{k\in W} (\hat{\pi}_{i,k} - \hat{\pi}_{j,k}) \leq 0,$$
$$\forall (i,j) \in A,\ W \subseteq U^{=}_{i,j},\ t - |U^{+}_{i,j}| \leq |W| \leq (n-t-2) - |U^{+}_{i,j}|$$

Note: If $|U^{+}_{i,j}| > n - t - 2$ or $|U^{+}_{i,j}| + |U^{=}_{i,j}| < t$, a minor change to $U^{+}_{i,j}$ and $U^{=}_{i,j}$ is needed.

Results:
We are currently working to obtain useful results utilizing this Primal-Dual Algorithm for instances of moderate size. Rounding errors have made it difficult to define appropriate tolerances for "tight" constraints.

Generating Tours

Price-Directed Branch-and-Bound Heuristic

Method:
The lower bound solutions can be used to generate TSP tours in a price-directed tour generation heuristic. The approximate costs-to-go derived from a dual solution serve as proxies for the actual costs as shown below:

If we are currently at city i and have cities U left to visit,
the approx. cost of next visiting city $j \in U$ is given by $\{\ c_{i,j} + y_{j,U\setminus j}\ \}$

Computational Challenges:
It is when two cities' approximate costs-to-go are tied that challenges arise. To help us make a decision in this event, we perform further analysis as follows:

1. For each candidate city $k \in \hat{U}$, solve a subproblem defined as the linear relaxation of the arc-based TSP formulation on the remaining cities, U.
   Note: Costs into city k are replaced by costs into the start city ($c_{i,0} \Rightarrow c_{i,k}$).
2. Update the approx. cost of next visiting each city $k \in \hat{U}$.
3. Choose to move to the city of minimum estimated cost.
4. Save the solution to the subproblem in order to calculate relevant costs-to-go later on.
5. In the event that two or more candidate cities are still tied, branch on each tied city.

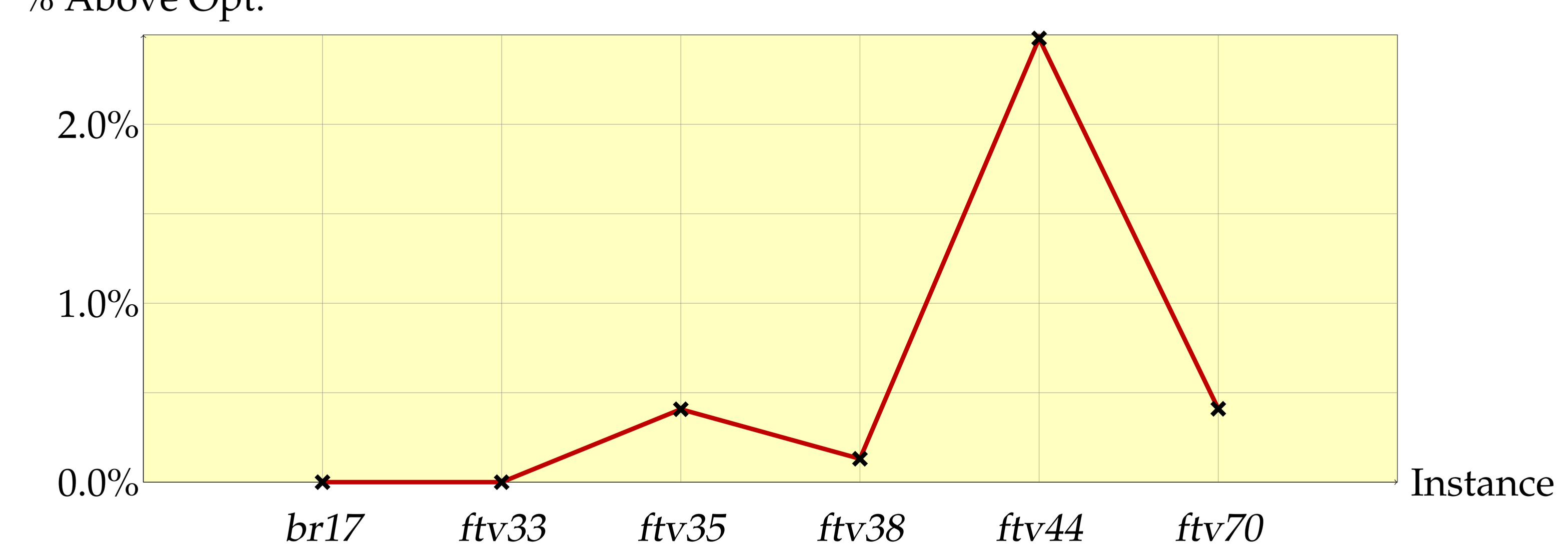
Note: We have experimented with performing Step (1) both more and less frequently.

To reduce the number of iterations, we fathom branches of the tree when:

- The bound on the cost of a partial tour exceeds the cost of a previously-found complete tour (eliminating the need to continue exploring this branch of the tree).
- A min cost subproblem yields an integral optimal solution (immediately giving us a min cost way to complete that partial tour).

Results:

Min Cost Tours Found



RED PLOT = min tour cost we found for the given test instance.

Note: Generally, we are able to find a tour of cost no more than 1-2% greater than the optimal tour cost.

Our heuristic terminated in under 10 min for all instances under 50 cities in size. Results for the 70 city instance took approximately 6 hours to obtain. We are currently running experiments on test instances of over 150 cities.