Network Design Foundations
Fall 2011
Results concerning Deadline Traveling Salesman Problem

Instructor: Mohammad T. Hajiaghayi
Students: Anshul Sawant, Catalin-Stefan Tiseanu

December 7, 2011

1 Introduction

Deadline-TSP (DTSP) and Vehicle routing with time-windows (VRTW) are extensions of the much studied TSP problem where each vertex to be visited has a time-window associated with it and server is rewarded only if it visits a node during this time window. For the deadline-TSP problem, all release times are identically zero. These problems have been studied extensively and find applications in areas such as vehicle routing, robot motion planning and task scheduling. In the above problems, only the starting vertex is specified. If both end points of walk are given, then the problem is called point to point orienteering with time-windows. The algorithms discussed in here apply to this problem as well.

2 Notation and Preliminaries

Let $G = (V, E)$ be a weighted graph, with a start node r. We use OPT to denote the optimal algorithm and the optimal path. When it is clear from the context we also use OPT to denote total reward collected by the optimal path. $D : V \mapsto \mathbf{Z}^+$ is the deadline function. Let $\Pi : V \mapsto \mathbf{Z}^+$ be the reward function. Let Π_P be the reward collected by path P. Let D_{max} denote maximum deadline. Let $R : V \mapsto \mathbf{Z}_{>=0}$ be the release time function.

3 Summary of Results

Here we present summary of current best know approximation for VRTW and related problems on undirected graphs. P2P is defined in the next section. Other problems listed in the table are: DTSP with constant number of different deadlines (DTSP-CD), VRTW with time-windows of size 1 (VRTW-1TW). Let $L_{max} = \max_{v \in V}(R(v) - D(v))$. Let $L_{min} = \min_{v \in V}(R(v) - D(v))$.

Table 1:

Problem	**Metric**	**Approximation**
DTSP[1]	Graph	$O(\log(n))$
VRTW[1]	Graph	$O(\log(D_{max}))$
VRTW[1]	Graph	$O(\log^2(n))$
VRTW[3]	Graph	$O(\log(\frac{L_{max}}{L_{min}}))$
DTSP-CD[4]	Graph	12
VRTW-1TW[5]	Graph	$6 + \epsilon$
VRTW-1TW[5]	Tree	Exact
VRTW[2]	Path	O(1)

4 Orienteering

In point-to-point orienteering (P2P), we are required to go from a node s to node t within time D. Objective is to maximize reward withing the above constraints. The paper gives a 3-approximation from point-to-point orienteering problem, which is an improvement over an earlier 4-approximation. The algorithms uses a $2 + \epsilon$- approximation to the min-excess path problem. In min-excess path problem, we are required to collect a reward of atleast k, starting at s and ending at t. The objective is to minimize $D - d(s, t)$, where D is the total length of path and $d(s, t)$ is the length of the shortest path between s and t. We assume that all distances, rewards and deadlines are polynomial in n.

4.1 Algorithm

Input: Metric G, s, t, D, rewards.
Output: A path that is a 3-approximation for p2p orienteering problem.

- For each pair of nodes (x, y) and value k, we compute a minimum excess path from x to y which visits k nodes.
- We then select the triple (x, y, k) with the maximum k, such that the computed path has excess $D - d(s, x) - d(x, y) - d(y, t)$ or smaller.
- We return the path which travels from s to x via shortest path, then x to y via the computed path, then y to t via shortest path.

4.2 Proof Sketch

Let reward be evenly spread among segments $s-x$, $x-y$ and $y-t$. Let segment $x-y$ have the least excess path among all segments. Then it will have atmost a third of the total excess. If we have a 3-approximation for min-excess problem we can get all the reward from segment $x - y$. Since, we can guess all triplets (x, y, k) in polynomial time, we get a 3-approximation to this problem.

5 Routing with Time-windows

The main intuituion behind this algorithm is that if we start collecting reward for a set of vertices after their release time, then the problem reduces to deadline-TSP. If, for a subset of vertices, OPT collects reward from some vertices before time d and deadlines of vertices are greater than or equal to d, then we can get a constant factor of reward from these vertices using the point-to-point orienteering problem.

6 Bicriterion Algorithm

Algorithm 1 Bicriteria algorithm for general case [1]

Input: Graph $G = (V, E)$ with deadline $d(v)$; parameter ϵ.
Output: Path P with $\Pi_P \geq \Omega(\frac{1}{\log(\frac{1}{\delta})}) * \Pi_{OPT}$ and $R(v) \leq t_r(v) \leq (1+\epsilon) * d(v)$.

Let $f = \frac{1}{\sqrt{1+\epsilon}}$ and s be the smallest such integer for which $f^{1.5^s} \leq \frac{1}{4}$.
Apply small margin algorithm with parameter f to the graph and let P_0be the path obtained.
Apply large margin algorithm to the graph, and let P_{s+1}be the path obtained.
for all $i \in \{1, .., s\}$ **do**
    For all $v \in V$,define $d'(v) = d(v)f^{1.5^{i-1}}$.
    Apply small margin algorithm with parameter $f^{0.5(1.5)^{i-1}}$to the graph with new deadline d', and let P_i the path obtained.
end for
Among the paths constructed above, return the one with the maximum reward.

6.1 Small Margin Algorithm

Let ϵ be a fixed constant. Consider the set of nodes $V_\epsilon = \{v : D(v)/(1+\epsilon) \leq t_{OPT}(v) \leq D(v)\}$. We can get a constant factor of reward from part of OPT that is in V_ϵ while exceeding the deadlines by a factor of $(1+\epsilon)^2$. Let $f = \frac{1}{(1+\epsilon)^{0.5}}$. Divide the nodes in the graph into segments where segment S_j has nodes with deadlines between $(f^j D_{max}, f^{j-1} D_{max}]$. Notice that vertices in S_j are visited before vertices in S_{j+2} by V_ϵ. To get a constant fraction of reward, one can compute a path that get optimal reward from every third of these segments and join them by taking shortcuts across the intermediate sets. To approximate the optimal path in some S_j, we guess the first and the last vertex that OPT visites in this set and corresponding times. Then we use p2p algorithm to construct a path of guessed length. Then we append these subpaths. Finally, we slow-down this path by a factor if f^3. We get a 9-approximation.

6.2 Large Margin Algorithm

Let $V_{1/4} = \{v : t_{OPT}(v) \leq \frac{D(v)}{4}\}$. To all nodes, assign new deadlines that are a fourth of original deadlines. Let $\alpha = 1.2$. Let S_i be the set of vertices with deadlines between $[\alpha^i, \alpha^{i+1})$. Let $S_{j \mod \beta} = \cup_{\geq 0} S_{\beta i+j}$. Let P_i be a path returned by the p2p algorithm with parameter $D = \alpha^{i+1}$ when applied to graph induced by $\{r\} \cup S_i$. Let T_i be the tour that starts at root, follows P_i and then returns to root. Append all paths of form $T_{\beta i+j}$. It can be proved, that this path does not exceed the original deadlines by a factor of two. We can slow this path down by a factor of two to ensure that release times are honored.

6.3 General Case

The algorithm for the general case is given above. It divides deadline-time space for the optimal path into $O(\log(D_{max}))$ segments. On one end we can get a constant factor reward from the segment that corresponds to large margin case. On the other end, we have a very thin segment and we are constrained to reap rewards in order of their deadlines. In between we have a bit of leeway and we can use the small margin algorithm above with appropriate scaling of deadlines.

References

[1] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 166–174. ACM, 2004.

[2] R. Bar-Yehuda, G. Even, and S.M. Shahar. On approximating a geometric prize-collecting traveling salesman problem with time windows. *Journal of Algorithms*, 55(1):76–92, 2005.

[3] C. Chekuri and N. Korula. Approximation algorithms for orienteering with time windows. *Arxiv preprint arXiv:0711.4825*, 2007.

[4] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 72–83, 2004.

[5] G.N. Frederickson and B. Wittman. Approximation algorithms for the traveling repairman and speeding deliveryman problems. *Algorithmica*, pages 1–24, 2009.