CO 754: Approximation Algorithms **(Univ. Waterloo, Winter 2012)**

The School Bus and the Orienteering problem

by Bock, Grant, Könemann, Sanità, Chen, Har-Peled

Author: Sahil Singla

1 Introduction

Vehicle routing problems (VRPs) are very popular mainly because of their interesting combinatorial properties and applications to various industries. Informally, the problem is to provide a service to a set of customers using a fleet of vehicles so as to minimize the 'transportation costs' without loosing 'customer satisfaction'. Many variations of the problem have been formulated corresponding to different definitions of 'transportation costs' and 'customer satisfaction'. These variations have been well studied in the fields of theoretical computer science and operations research, and many of them have been shown to be strongly NP hard. Two such variations are defined below:

(a) **The School Bus Problem (SBP):** Given an undirected graph $G(V, E)$ on a metric space, a node $s \in V$ representing the school, a set $W \subseteq V$ representing the houses of children, a bus capacity $C \in \mathbb{Z}_+$, and a regret bound $R \in \mathbb{Z}_+$. The aim is to find the minimum number of buses and their corresponding walks ending at s such that each child is allocated to a bus and the following constraints are satisfied:

   (i) No bus exceeds the capacity C

   (ii) For every child the regret bound is respected i.e. $d^P(w, s) \leq d(w, s) + R$. Here $d^P(w, s)$ is the distance from w to s along path P taken by the allocated bus, and $d(w, s)$ is the shortest distance from w to s in G.

(b) **The Distance constrained vehicle routing problem (DVRP):** Given a set of vertices in a metric space, a specified depot, and a distance bound D, find a minimum cardinality set of tours originating at the depot that covers all vertices, such that each tour has length at most D.

The above two problems are not only known to be APX hard but also no constant approximation factor algorithms are known. The problems can be studied as set cover problems, and hence a $O(log(n))$ approximation factor algorithms can be obtained using a greedy approach (see section 2). The integrality gap of the set-cover formulation of the above problems on a general metric is also known to be logarithmic. Still, progress has been made on special metrics for the problems, and there is a 2 approximation factor algorithm known for DVRP [8] and a 4

Problem	References	Approximation factor
SBP on tree metric	[3]	4
SBP with regret minimization on trees	[3]	12.5
DVRP on tree metric	[8]	2
Orienteering problem	[4]	$2 + \epsilon$
Euclidian rooted orienteering problem	[5]	PTAS
k-TSP	[6]	2
k-TSP on euclidian metric	[7],[1]	PTAS

Table 1: Summary of important results

approximation factor algorithm known for SBP [3], both on tree metric. The first half (section 3) of the report will be devoted to detailed discussion on the result by Bock et al. for SBP [3].

A very famous combinatorial optimization problem is the metric Traveling Salesman Problem (TSP). Given an undirected edge weighted graph on a metric, the goal is to find the shortest tour that covers all the nodes of the graph. TSP is a very old and a very well studied problem. A 1.5 approximation algorithm was known for the problem since many decades, but only recently better approximation algorithms have been obtained. The second half (section 4) of the report discusses a very similar problem called the Orienteering problem. Here instead of covering all the points input points with a minimum length tour, we are given a bound on the length of the tour/ path, and we need to maximize the number of distinct points covered by the tour/path.

The Orienteering Problem: Given an edge-weighted graph $G = (V, E)$, two nodes $s, t \in V$ and a budget B, find an $s - t$ walk in G of total length at most B that maximizes the number of distinct points visited by the walk.

The first constant approximation algorithm for the Orienteering problem was obtained by Blum et al. [2], but the current best known is a $2 + \epsilon$ approximation factor algorithm by Chekuri et al. [4]. For the euclidian rooted version of the problem, a PTAS by Chen et al. [5] was obtained. All these algorithms are based on some approximation algorithm for a related k-TSP problem (defined in section 4). Using Mitchell's PTAS algorithm [7] for euclidian k-TSP problem, we will explain Chen et al.'s proof for obtaining a PTAS for euclidian rooted orienteering problem in section 4.

Caution: *The proofs and results included in this report have been greatly simplified to enhance readability. Hence some of the important details might be missing. Interested readers should refer to the original references for rigorous proofs and statements of the results.*

2 Connections between SBP & Orienteering problem

The SBP can be formulated as a set-cover problem where 'the sets' are all possible subsets of the children node set W that can be covered by a single bus without violating capacity and regret constraints. All subsets have a unit weight as in the original problem we only wish to minimize the number of buses. To obtain a $O(log(n))$ approximation factor algorithm for the SBP using the greedy approach for set cover problem, we need to solve the greedy step in polynomial time. Even if one can obtain a solution to the greedy step problem within a constant factor approximation, the old analysis as that of the set-cover problem goes through. Here the greedy step problem is as follows:

Greedy step problem: Find a walk ending at s and covering the maximum number of distinct children nodes in W such that the capacity constraint for the bus and the regret bound constraint for every children on the bus is satisfied.

As shown in lemma 3.2, one can forget about the capacity constraint without loosing much in the approximation factor. Also, the following simple lemma 2.1 shows that if regret bound constraint is satisfied for the child node at which the bus starts, regret bound constraints are also satisfied for all the other nodes allocated to the bus.

Lemma 2.1. *For a walk P starting at vertex v, ending at s on a tree metric, and covering a subset S of nodes, the regret bound R is respected for all nodes in S if and only if it is respected for v.*

Proof. A simple proof by contradiction. Suppose there is a node $w \in S$, with $d^P(w,s) > d(w,s) + R$ while $d^P(v,s) \leq d(v,s) + R$. From the triangle inequality we get the following contradiction

$$d^P(v,s) \geq d^P(w,s) + d(v,w) > R + d(w,s) + d(v,w) \geq R + d(v,s)$$

□

As one can try every possible node as the starting node t for the optimal bus (there are only $|W|$ possible starting nodes), the modified problem becomes:

Modified greedy step problem: Find a s–t walk of length at most $d(s,t) + R$ such that it covers the maximum number of distinct children nodes in W.

The above problem is nothing but the Orienteering problem, and one can use constant factor approximation algorithms to solve it on a general metric [4],[2]. There is one catch though, the way we have defined orienteering problem in section 1 is that the walk is allowed to select any node in the graph. On the other hand for the modified greedy step problem, we only wish to pick nodes that belong to W. Fortunately this is not a concern as we can restrict our attention to just the subgraph induced by the nodes in $W \cup \{s\}$ (after taking metric closure). The reason being that steiner nodes have no benefit for the optimal walk and one can always take a shortcut (we are on a metric space) without increasing the length. Thus, using a constant factor approximation algorithms for the Orienteering problem as a subroutine, one can easily obtain a $O(log(n))$ approximation algorithm for SBP on general metric.

3 The School Bus Problem

The School Bus Problem (SBP): Given an undirected graph $G(V, E)$ with distances on the edges $d : E \to \mathbb{Z}_+$, a node $s \in V$ representing the school, and a set $W \subseteq V$ representing the houses of children. Additionally, there is a bus capacity $C \in \mathbb{Z}_+$, and a regret bound $R \in \mathbb{Z}_+$. The aim is to construct a minimum cardinality set $\mathcal{P}$ of walks ending at s, and assign each child $w \in W$ to be the responsibility of some bus $p(w) \in \mathcal{P}$ such that:

(i) The number of allocated house nodes to any bus does not exceed the capacity C

(ii) For every child the regret bound is respected i.e. $d^P(w, s) \leq d(w, s) + R$. Here $d^P(w, s)$ is the distance from w to s along path P, and $d(w, s)$ is the shortest distance from w to s in G.

As shown in section 2, there exists a $O(log(|W|))$ approximation factor on general metric using a greedy approach for the problem. Bock et al. [3] gave a 4 approximation algorithm for SBP on tree metrics. The general idea of their algorithm is to first show that removing the capacity constraints only leads to a small loss in the approximation factor, and to then divide the problem into the following two cases

1. When the tree is short, i.e. height of T is at most $R/2$: Here one can obtain a $2-$ *approximation* for SBP by cutting the Euler tour of the tree into shorter pieces and assigning each piece to a bus.

2. Long trees: Partition the general tree into smaller pieces such that each piece requires at least one bus, but each piece can be solved 'almost optimally' via a Euler tour method similar to short trees.

Using lemma 2.1, without loss of generality we can assume $W = V$ for SBP on tree metric. Also, we can assume the input tree to be a binary tree as we can always add zero cost edges to reduce the degree of nodes.

3.1 Notation

- $lca(a_i, a_j)$**:** Lowest common ancestor of vertices a_i and a_j in T.

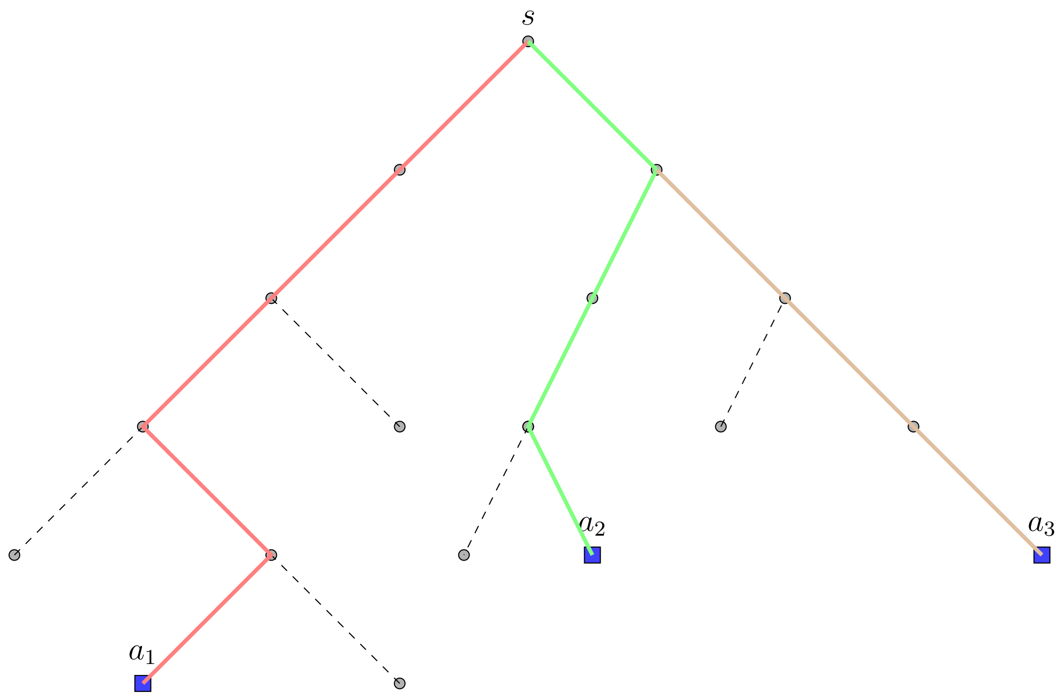

Figure 1: A tree metric showing the anchor set A and the skeleton $\mathcal{Q}$

- **R-independent set:** A set of vertices $\{a_1, a_2, \ldots, a_m\} \subseteq V$ is R-independent if for all $a_i \neq a_j$, we have $d(a_i, lca(a_i, a_j)) > R/2$.
- **Anchor set (A):** Inclusion wise maximal R-independent set of leaves A such that all vertices in T are within a distance $R/2$ from a path $P(s, a)$ for some $a \in A$. The elements of the anchor set are called anchors.

  One can always obtain an anchor set for any tree by starting with an empty set and then iteratively including the lowest leaf in the tree such that the set of selected nodes continue to remain R-independent. The maximal R-independent set (A) thus obtained is an Anchor set as if any node is not within a distance $R/2$ from a path $P(s, a)$ for some $a \in A$, then it can be also included but that gives a contradiction by violating maximality.
- **Skeleton of T:** $\mathcal{Q} = \bigcup_{a \in A} P(s, a)$ is the skeleton for tree T with respect to anchor A.

As no two distinct anchors can both be covered by the same walk without violating the regret constraint, the following claim holds true trivially.

Claim 3.1. *The size $|A|$ of the set of anchors is a lower bound on the number of buses that are needed in any feasible solution.*

Since visiting all anchors also covers the skeleton, we only consider covering the anchor set and the non-skeletal edges of T.

Algorithm:

(1) Obtain an anchor set $A = \{a_1, a_2, \ldots, a_m\}$ by iteratively marking the lowest leaf a_i in T such that R-independence is maintained among the marked leaves.

(2) Obtain paths of edges $\{P_1, P_2, \ldots, P_m\}$ where $P_1 = P(s, a_1)$, and $P_i = P(s, a_i) \backslash (\bigcup_{j=1}^{i-1} \{P_j\})$ for $2 \leq i \leq m$. Let T_i be the set of edges in both the path P_i and the set of all short subtrees attached to P_i (solve ties arbitrarily).

(3) For all i, obtain directed walk W_i starting at anchor a_i such that

   (a) W_i contains each edge of P_i exactly once.

   (b) W_i contains each edge of $T_i \backslash P_i$ exactly twice: once going towards and once going away from s.

(4) Greedily assign edges to buses in the order they are visited by W_i such that a bus covers edges of total length exactly $R/2$ in the downward direction i.e. away from root s.

Note that from the definition of anchor set, $T \backslash \mathcal{Q}$ is a collection of short subtrees, i.e. trees of height at most $R/2$. Also note that it is important to obtain an anchor set A. Any maximal R-independent set might not satisfy that $T \backslash \mathcal{Q}$ forms a collection of trees with height at most $R/2$. For example, a set consisting of only the root node s is also a maximal R-independent set as adding any other node v to the set will give $lca(v, s) = s$, and $d(s, lca(v, s)) = 0 < R/2$.

3.2 Proof

Lemma 3.2. *Given an α-approximation to the SBP with unlimited capacity for each bus, there is an $\alpha + 1$- approximation to the SBP that respects a capacity bound C on each bus.*

Proof. A simple way of converting a solution with unlimited capacity to a solution with capacity C on each bus is to use the same routes and whenever a bus gets full, use a new bus starting at the same point. Since the new bus is following the old route (the one followed by the corresponding bus with unlimited capacity), regret bound for all the children on the route is respected. Let SOL_∞ and SOL_C be the solutions for infinite capacity and capacity C respectively. Similarly, let OPT_∞ and OPT_C be the optimal solutions for the two cases. Let Bus load$_i$ be the number of children nodes allocated to bus i with infinite capacity. We can thus write

$$SOL_C = \sum_{i=1}^{SOL_\infty} \left\lceil \frac{\text{Bus load}_i}{C} \right\rceil \leq \sum_{i=1}^{SOL_\infty} \left(\frac{\text{Bus load}_i}{C} + 1 \right) = SOL_\infty + \frac{|W|}{C}$$

Note we also have a lower bound on OPT_C

$$OPT_C \geq \left\lceil \frac{|W|}{C} \right\rceil \geq \frac{|W|}{C}$$

Thus we get

$$SOL_C \leq SOL_\infty + \frac{|W|}{C} \leq SOL_\infty + OPT_C \leq (\alpha+1)OPT_C$$

□

Lemma 3.3. *The number $\frac{1}{R}\sum_{e\in T\setminus\mathcal{Q}} d(e)$ is a lower bound on the number of buses needed for any feasible solution.*

Proof. Consider any feasible walk P starting at a vertex v. It will cover all the edges in $P(s,v)$, and possibly some additional detour edges. Since $T\setminus\mathcal{Q}$ is a collection of short trees, P can cover at max R non-skeletal edges:

- at most $R/2$ in length along the path from starting vertex v to the root s
- edges of length at most $R/2$ as detours.

Thus we get a lower bound of $\frac{1}{R}\sum_{e\in T\setminus\mathcal{Q}} d(e)$ on the number of buses needed by any feasible solution. □

Theorem 3.4. ***(Bock et al. [3])*** *The above polynomial time algorithm gives an approximation factor* 3 *solution to the school bus problem on trees with unlimited capacity.*

Proof. The number of buses used by the above algorithm is exactly $\sum_{i=1}^{|A|}\left\lceil\frac{2\sum_{e\in T_i\setminus P_i} d(e)}{R}\right\rceil$ because each bus other than the last bus consumes exactly $R/2$ of the downward direction edges of W_i, and W_i proceeds downward along each edge in $T_i\setminus P_i$ exactly once. This is a feasible solution as if a bus is travelling at most $R/2$ in the downward direction, it must make a detour no greater than R. Hence we can write

$$\sum_{i=1}^{|A|}\left\lceil\frac{2\sum_{e\in T_i\setminus P_i} d(e)}{R}\right\rceil \leq |A| + \frac{2}{R}\sum_{i=1}^{|A|}\sum_{e\in T_i\setminus P_i} d(e) = |A| + 2\frac{\sum_{e\in T\setminus\mathcal{Q}} d(e)}{R} \leq 3OPT$$

Here the last inequality uses claim 3.1 and lemma 3.3. □

Now using theorem 3.4 and lemma 3.2, we get a polynomial time approximation factor 4 algorithm for the school bus problem on trees as a corollary.

4 PTAS for the Euclidian Orienteering Problem

The current best known approximation factor for the orienteering problem on a general metric is $2 + \epsilon$ by Chekuri et al. [4]. There is a rooted variation of the orienteering problem as well where instead of fixing both the starting and the end vertices of the path, you are only given a starting vertex (root vertex) r and the other end is free (could be any vertex in V). The rooted version is of course harder than an unrooted one, i.e. where both the ends are free, as one try every possible node as a root. Chen et al. [5] gave a (ϵ, u) PTAS for the euclidian rooted path orienteering problem, where $0 < \epsilon < 1$ is a sufficiently small constant and $u \geq (2/\epsilon)$ is a sufficiently large constant. We present Chen et al.'s result in this section of the report.

Euclidian rooted path orienteering problem (EROP): Given an edge-weighted graph $G = (V, E)$ on a euclidian metric, a root node $r \in V$ and a budget B, find a path starting at r of total length at most B that maximizes the number of distinct points visited by the path.

4.1 Relationship between orienteering problem and k-TSP

The orienteering path problem is to obtain a path of length at most B that covers the maximum number of distinct points in V. Consider a 'dual' problem where we are instead given the number of points that the path needs to cover and the goal is to minimize the length of the path. This problem is known as the k-TSP problem.

Rooted k-TSP: Given an edge-weighted graph $G = (V, E)$, an integer $k > 0$, and a root node r, find the minimum length path in G starting at r and covering at least k points.

The rooted path orienteering problem is very related to the rooted k-TSP problem. Suppose you know the optimal number of points k^* covered by the optimal path to the rooted orienteering problem (for the algorithm try every possible value of k^* between 1 and n). Now we just need to find a path starting at r, covering k^* points and of length at most B. One way to solve this problem is to try finding the shortest path starting at r and covering k^* points, which is the rooted k-TSP problem. This shortest length has to be at most B as the optimal solution to rooted path orienteering problem is a feasible solution to rooted k^*-TSP problem.

4.2 Proof Overview

The proof of Chen et al. [5] is primarily based on the PTAS algorithm for k-TSP problem by Mitchell [7]. The old analysis of Mitchell's algorithm (theorem 4.2) is improved to obtain a 'tighter' bound on the solution returned by the algorithm (theorem 4.3). Then they use it as a sub-routine to obtain a PTAS (theorem 4.4) for the euclidian rooted path orienteering problem. The algorithm for EROP is to first guess the optimal number of points k^* covered by the optimal EROP solution (by trying every possible value of k^* from 1 to n), and then run

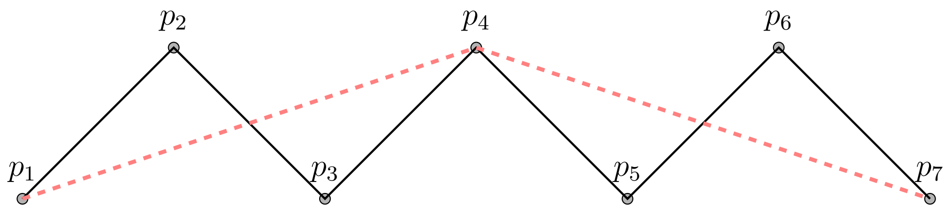

Figure 2: For path π on 7 vertices (shown in black), $\mathcal{S}^3_{opt}(\pi)$ is the path shown in red and 3-excess is the difference of lengths of paths in black and red

k-TSP algorithm for $k = k^*(1-\epsilon)$. Using Mitchell's k-TSP PTAS algorithm and the improved analysis, it can be shown (theorem 4.4) that a solution covering $k = k^*(1-\epsilon)$ points and of length at most B can be found in polynomial time. Thus we will obtain a PTAS for EROP.

4.3 Notation

Most of the notation used in this report is from Chen et al. [5] and Mitchell's paper [7]. For any path $\pi = \langle p_1, p_2, \ldots, p_k \rangle$, by $||\pi||$ we denote the path length $\sum_{i=1}^{k-1} ||p_i - p_{i+1}||$. A sub-path $\langle p_{\alpha_1}, p_{\alpha_2}, \ldots, p_{\alpha_u} \rangle$ is called $u-skeleton$ of π. Now let $\mathcal{S}^u_{opt}(\pi)$ be the maximum length of a $u-skeleton$ of π. The $u-excess$, $\xi_{\pi,u}$, of a path π is the difference of $||\pi||$ and $\mathcal{S}^u_{opt}(\pi)$.

Intuitively for a path π with k ($\geq u$) vertices, $\mathcal{S}^u_{opt}(\pi)$ is the best approximation to path π by a subpath on at most u vertices. $\xi_{\pi,u}$ is therefore the minimum loss that one incurs while trying to approximate π with a subpath on at most u vertices. See figure 2 for an example.

- $\mathcal{Q}$: Minimum axis parallel rectangle containing all the input points P.
- m: A fixed constant greater than 2.
- u: A fixed constant greater than 2 such that $m \geq 2\sqrt{2}u$, and $u \geq \frac{2}{\epsilon}$. Here ϵ is a small enough positive constant ($<< 1$) which is a parameter for the PTAS algorithm.
- Window($\overline{w}$): Axis parallel rectangle $\overline{w} \subseteq \mathcal{Q}$
- $\pi(\overline{w})$: Subset of π consisting of segments of π having at least one endpoint inside window $\overline{w}$.
- $\triangle_{\overline{w}}$: The larger of the width and height of window $\overline{w}$.
- m-perfect cut for π: A horizontal or vertical line l intersecting window $\overline{w}$, and intersecting segments of $\pi(\overline{w}) \cap \overline{w}$ at most m times.
- m-dense: A window $\overline{w}$ with no m-perfect cut for π.
- Minimal m-dense window: An m-dense window for π that contains no smaller window covering the same subset of points of P.

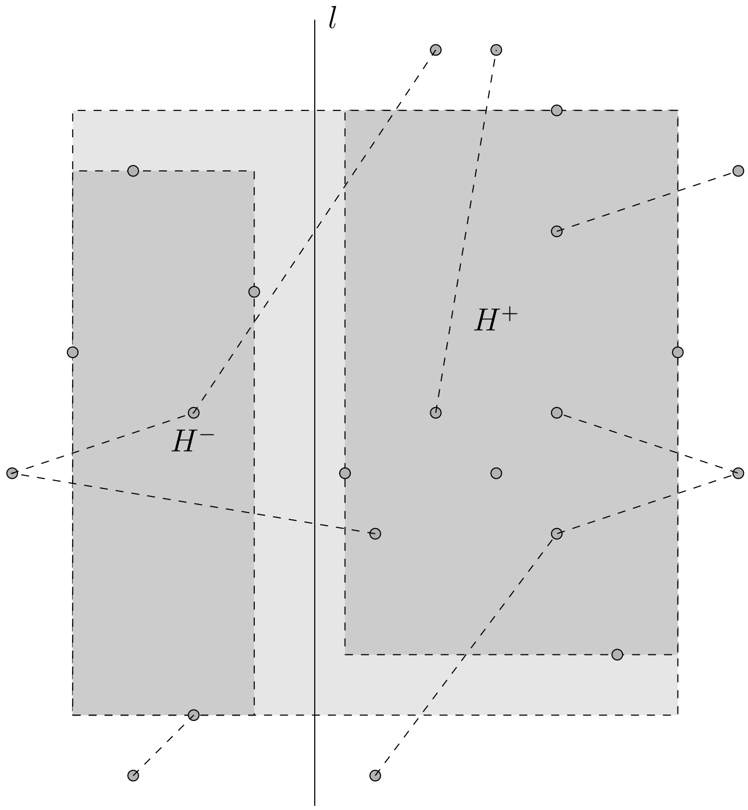

Figure 3: An m-perfect cut (line l) for the optimal path that divides window $\overline{w}$ into two minimal windows H^+ & H^- (figure redrawn from [5])

Definition 4.1. For any window $\overline{w}$ and a path π, the surplus ρ of π in $\overline{w}$ is

$$\rho(\overline{w}, \pi) = ||\pi \cap \overline{w}|| - \sqrt{(len_x(\pi \cap \overline{w}))^2 + (len_y(\pi \cap \overline{w}))^2}$$

The surplus is always non-negative as shown in lemma 4.6.

4.4 Algorithm for k-TSP

The algorithm in 2-dimesnsions is same as Mitchell's algorithm, but for higher dimensions it has to use Arora's algorithm as a subroutine. Conceptually it runs recursively (DP is used actually) starting with window $\overline{w} = \mathcal{Q}$. The basic idea is to find an m-perfect cut (if it exists) for the optimal path in the current window $\overline{w}$ (see figure 3). As there are only $O(n^2)$ possible cuts (details not important but that it is polynomial), even though the algorithm does not know if a picked cut is m-perfect or not (as it does not know the optimal path), it can still recursively solve for every possible cut, and select the minimum feasible solution amongst all of them. Also, as the optimal path crosses an m-perfect cut at most m times, there are only $O(n^{O(m)})$ possible ways in which a cut divides the optimal path (note m is a constant and therefore the function is still polynomial). Note that there are only $O(n^4)$ possible minimal windows (as 4 degrees of freedom for a window) and hence we can solve everything in polynomial time using Dynamic Programming.

In case there does not exist any m-perfect cut for the optimal path in the current window $\overline{w}$, the algorithm conceptually runs the PTAS algorithm for k-TSP problem by Mitchell (in 2-d), or by Arora (for any constant dimensions d). As shown in the proof, for m-dense windows the old analysis given by Mitchell or Arora suffices. Here again as we do not know if the current window $\overline{w}$ is m-dense or not, we run both the old algorithm (Arora or Mitchell) and the recursive algorithm for $\overline{w}$, and then take the minimum of the two feasible solutions.

Outline of algorithm: Start with window $\overline{w} = \mathcal{Q}$

(1) If $\overline{w}$ has at most m points of P then solve the problem optimally using brute force.

(2) Else the algorithm chooses the smaller value returned by the following two options

   (a) Solve the problem recursively, optimizing over all choices associated with an m-perfect cut of window $\overline{w}$ (there are $O(n^2)$ possible choices of cut and $O(n^{O(m)})$ boundary conditions for a cut).

   (b) Solve the problem for window $\overline{w}$ using Mitchell or Arora's algorithm (theorem 4.2), and use the old analysis i.e. a solution of length at most $(1 + 1/m)L(\overline{w})$ where $L(\overline{w})$ is the length of optimal solution inside $\overline{w}$.

4.5 Main Results

Mitchell and Arora independently gave algorithm for obtaining PTAS for rooted k-TSP problem. This report will not be covering the proof of Mitchell's/ Arora's PTAS algorithm and is going to assume theorem 4.2. Interested readers are referred to [7], [1].

Theorem 4.2. ***(Mitchell [7], Arora [1])*** *For any window $\overline{w} \in \overline{D}$, there is a polynomial time algorithm that outputs a set of paths (with pre-specified end-points) of total length $\leq L(\overline{w})(1+\frac{1}{m})$ that collectively visits a pre-specified number of points inside $\overline{w}$. Here $L(\overline{w})$ is the length of the optimal solution inside the window $\overline{w}$ satisfying the given conditions.*

The following theorem by Chen et al. basically provides a better analysis for Mitchell's algorithm for rooted k-TSP problem.

Theorem 4.3. ***(Chen et al. [5])*** *For any path $\pi = \langle p_1, p_2, \ldots, p_k\rangle$ and a constant $u > 0$ in a graph G on a euclidian metric space, it is possible to find in polynomial time a path starting at p_1, covering at least k points, and of maximum length $||\pi|| + \frac{\xi_{\pi,u}}{u}$.*

Assuming the above theorem (proved in section 4.6), one can find a PTAS for the EROP as follows.

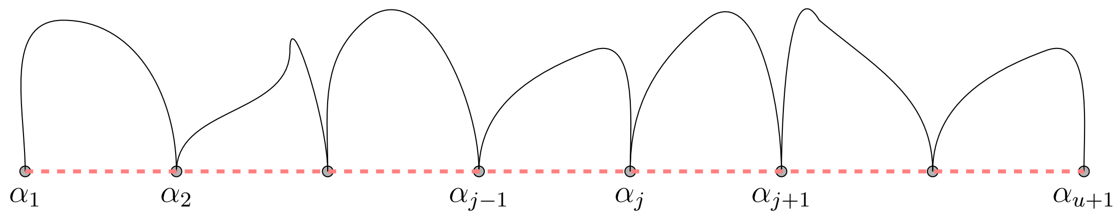

Figure 4: Optimal path π shown in black and path π_{u+1} shown in red

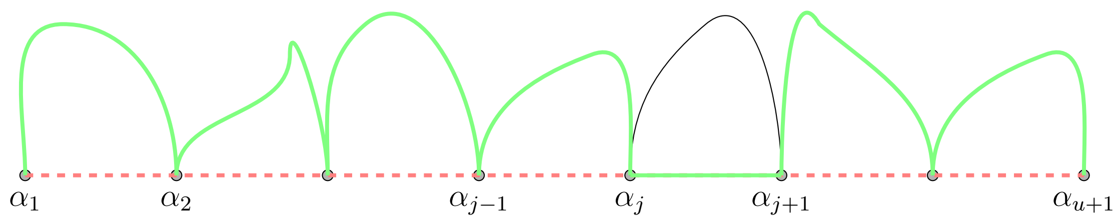

Figure 5: Path π' shown in green

Theorem 4.4. ***(Chen et al. [5])*** *Given an edge weighted graph $G = (V, E)$ in a euclidian metric space, a root node $r \in V$, and a budget B, let $\pi = \langle p_1, p_2, \ldots, p_k \rangle$, where $p_1 = r$, be an optimal solution to the rooted path orienteering problem. Then there is a polynomial time algorithm that finds a path starting at r of length not greater than B and covering $k(1-\epsilon)$ distinct points in V.*

The main idea of the proof of theorem 4.4 is to first guess the number of points k covered by the optimal solution to EROP. Then they show that there exists a path starting at r of length slightly less than $||\pi||$, say $||\pi'||$, and covering $k(1-\epsilon)$ distinct points. Now using the improved analysis for rooted euclidian k-TSP problem from theorem 4.3, it can be shown that in polynomial time we can find a path such that even after an increase in length from $||\pi'||$, its length remains less than B, and the path covers at least $k(1-\epsilon)$ distinct points. To simplify the proof a little we assume that $\frac{k}{u}$ is an integer, else we will have to work with $\left\lceil \frac{k}{u} \right\rceil$.

Proof. Let $u = \frac{2}{\epsilon}$ and $\alpha_i = 1 + (i-1)\frac{k}{u}$, for $1 \le i \le u+1$. Note that for all i there are same number of integers between α_i and α_{i+1}. Now consider the path $\pi_{u+1} = \langle p_{\alpha_1}, p_{\alpha_2}, \ldots, p_{\alpha_{u+1}} \rangle$ of $u+1$ vertices (see the figure above). Certainly $||\pi_{u+1}|| \le \mathcal{S}_{opt}^{u+1}(\pi)$. Let ξ_i be $2-$*excess* of the path $\langle p_{\alpha_i}, p_{\alpha_i+1}, \ldots, p_{\alpha_{i+1}-1}, p_{\alpha_{i+1}} \rangle$. Hence $\sum_{i=1}^{u} \xi_i = ||\pi|| - ||\pi_{u+1}||$.

Now $\exists j$ such that $\xi_j \ge (\sum_{i=1}^{u} \xi_i)/u$. Let $\pi' = \langle p_1, p_2, \ldots, p_{\alpha_j - 1}, p_{\alpha_j}, p_{\alpha_{j+1}}, p_{\alpha_{j+1}+1}, \ldots, p_k \rangle$. Hence $||\pi'|| = ||\pi|| - \xi_j$, and it covers $k - \frac{k}{u} = k(1-\frac{1}{u})$ distinct points. Applying theorem 4.3 on π' we can find a path covering $k(1-\frac{1}{u})$ points and of length less than $||\pi'|| + \frac{\xi_{\pi',u+1}}{u+1}$.

We also have $\xi_{\pi',u+1} \le ||\pi'|| - ||\pi_{u+1}|| = (||\pi|| - \xi_j) - (||\pi|| - \sum_{i=1}^{u} \xi_i) = \sum_{i=1}^{u} \xi_i - \xi_j$. Thus the output path of the above algorithm has length less than

$$||\pi'|| + \frac{\xi_{\pi',u+1}}{u+1} \leq (||\pi|| - \xi_j) + \frac{\sum_{i=1}^{u} \xi_i - \xi_j}{u+1} = ||\pi|| + \frac{\sum_{i=1}^{u} \xi_i - (u+2)\xi_j}{u+1} \leq ||\pi|| \leq B$$

□

4.6 Proof of theorem 4.3

Theorem 4.3. (Chen et al. [5]) For any path $\pi = \langle p_1, p_2, \ldots, p_k \rangle$ and a constant $u > 0$ in a graph G on a euclidian metric space, it is possible to find in polynomial time a path starting at p_1, covering at least k points, and of maximum length $||\pi|| + \frac{\xi_{\pi,u}}{u}$.

To begin discussion on the proof of theorem 4.3, we will need a couple of simple lemmas. Then using the lemmas we will first give a proof for the special case when $\mathcal{Q}$ is m-dense for the optimal path π^* (in section 4.6.1). The ideas for proving this special case are quite similar to the final proof for the general case (in section 4.6.2).

Lemma 4.5. ***(Chen et al. [5])*** *For any m-dense window $\overline{w}$, $||\pi(\overline{w}) \cap \overline{w}|| \geq m.\triangle_{\overline{w}}$*

Proof. Without loss of generality assume that for $\overline{w}$, the width is $\geq$ the height. Let $[x_1, x_2]$ be the projection of $\overline{w}$ on the x-axis. As $\overline{w}$ is an m-dense window, it means that any vertical line l in $[x_1, x_2]$ intersects the segments of $\pi(\overline{w}) \cap \overline{w}$ at least $m+1$ times. Let $f(x)$ denote the number of intersections of a vertical line, with x-coordinate x, with the segments of $\pi(\overline{w}) \cap \overline{w}$. Now the integral of $f(x)$ in the interval $[x_1, x_2]$ is a lower bound on $||\pi(\overline{w}) \cap \overline{w}||$. As the integral is at least $m(x_2 - x_1) = m.\triangle_{\overline{w}}$, we have proved the above lemma. □

Lemma 4.6. *If $X, Y, X_1, X_2, \ldots X_n, Y_1, Y_2, \ldots, Y_n$ are non negative real numbers such that*

$$\sum_{i=1}^{n} X_i \geq X \text{ and } \sum_{i=1}^{n} Y_i \geq Y$$

then $\sum_{i=1}^{n} \sqrt{X_i^2 + Y_i^2} \geq \sqrt{X^2 + Y^2}$

Proof. For $1 \leq i \leq n$, let q_i be a point with co-ordinates $(\sum_{j=1}^{i} X_j, \sum_{j=1}^{i} Y_j)$, and $q_0 = (0, 0)$. Let π be the path $\langle q_0, q_1, \ldots, q_n \rangle$. Now triangle inequality gives that length of direct path between q_0 and q_1 is at least the length of path π, i.e.

$$||\pi|| = \sum_{i=1}^{n} ||q_i - q_{i-1}|| = \sum_{i=1}^{n} \sqrt{X_i^2 + Y_i^2} \geq ||q_n - q_0||$$

$$= \sqrt{\left(\sum_{i=1}^{n} X_i\right)^2 + \left(\sum_{i=1}^{n} Y_i\right)^2} = \sqrt{X^2 + Y^2}$$

□

4.6.1 Proof assuming $\mathcal{Q}$ is m-dense for π^*

If the window $\mathcal{Q}$ containing all the input points P is m-dense for the optimal path π^*, then our algorithm will basically just return the output of Mitchell's/ Arora's algorithm, and the cost of the solution will be given by the old analysis. This is because m-dense implies that there does not exist any m-perfect cut and hence the algorithm will just return the output as given in part (b) of our outline algorithm. Hence we need to prove that in this situation even the old analysis is as good as the new analysis, i.e. the returned solution $||\pi^*||(1+\frac{1}{m})$ is $\leq ||\pi^*|| + \frac{\xi_{\pi^*,u}}{u}$, for $m \geq 2\sqrt{2}u$. The idea is to first prove a lower bound on $\xi_{\pi,2}$ and then use to it to prove $\xi_{\pi,u} \geq \frac{||\pi||}{2}$, which is then used to prove $\frac{||\pi^*||}{m} \leq \frac{\xi_{\pi^*,u}}{u}$.

Claim 4.7. $\xi_{\pi,2} \geq ||\pi|| - \sqrt{2}\triangle_{\mathcal{Q}}$

Proof. Let the path π start at node s and end at node t, where $s, t \in P$. Let X and Y be the projection $s - t$ path on the x- and y- axis respectively. Now,

$$\xi_{\pi,2} = ||\pi|| - \mathcal{S}_{\pi,2} = ||\pi|| - ||s-t|| \geq ||\pi|| - \sqrt{X^2+Y^2} \geq ||\pi|| - \sqrt{2}\triangle_{\mathcal{Q}}$$

□

Claim 4.8. $\xi_{\pi,u} \geq \frac{||\pi||}{2}$

Proof. Let $\pi_1, \pi_2, \ldots, \pi_{u-1}$ be the breakup of π into subpaths by the vertices of $\mathcal{S}^u_{opt}(\pi)$. We can now write

$$\begin{aligned}\xi_{\pi,u} = \sum_{i=1}^{u-1} \xi_{\pi_i,2} &\geq \sum_{i=1}^{u-1}(||\pi_i|| - \sqrt{2}\triangle_{\mathcal{Q}}), \text{ using claim 4.7} \\ &\geq \sum_{i=1}^{u-1}(||\pi|| - \sqrt{2}\frac{||\pi||}{m}), \text{ using lemma 4.5} \\ &\geq \sum_{i=1}^{u-1}(||\pi_i|| - \frac{||\pi||}{2u}) \geq \frac{||\pi||}{2}\end{aligned}$$

□

Now using π^* instead of π in the above two claims, we get the desired result

$$\frac{||\pi^*||}{m} \leq \frac{2\xi_{\pi^*,u}}{m} \leq \frac{\xi_{\pi^*,u}}{\sqrt{2}u} \leq \frac{\xi_{\pi^*,u}}{u}$$

4.6.2 General Proof

The general proof will follow the idea of proof of the special case that assumed $\mathcal{Q}$ to be m-dense for π^*. Here instead of claim 4.7 we have lemma 4.9, and instead of claim 4.8 we have lemma 4.10

Lemma 4.9. ***(Chen et al. [5])*** *Let $\overline{D}$ be a set of interior disjoint windows (inside $\mathcal{Q}$), and let π be a polygonal path inside $\mathcal{Q}$. We have $\xi_{\pi,2} \geq \sum_{\overline{w}\in\overline{D}}(||\pi \cap \overline{w}|| - \sqrt{2}\triangle_{\overline{w}})$*

The above lemma is proved by showing that the overall surplus is at least the sum of the surplus for disjoint windows.

Proof. Let Ψ be a decomposition of $\mathcal{Q}$ into interior disjoint axis-parallel rectangles such that Ψ contains all rectangles of $\overline{D}$. Let $X = len_x(\pi \cap \mathcal{Q})$ and $Y = len_y(\pi \cap \mathcal{Q})$. Similarly for a window $\overline{w}$, let $X_{\overline{w}} = len_x(\pi \cap \overline{w})$ and $Y_{\overline{w}} = len_y(\pi \cap \overline{w})$.

$$\xi_{\pi,2} = ||\pi|| - ||s-t|| = ||\pi|| - \sqrt{len_x(st)^2 + len_y(st)^2} \geq ||\pi|| - \sqrt{X^2+Y^2}$$

since $len_x(st) \leq X$ and $len_y(st) \leq Y$. Now using lemma 4.6, we get$\sqrt{X^2+Y^2} \leq \sum_{\overline{w}\in\Psi}\sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2}$ since $\sum_{\overline{w}\in\Psi} X_{\overline{w}} \geq X$ and $\sum_{\overline{w}\in\Psi} Y_{\overline{w}} \geq Y$. Therefore,

$$\xi_{\pi,2} \geq ||\pi|| - \sqrt{X^2+Y^2} \geq ||\pi|| - \sum_{\overline{w}\in\Psi}\sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2} = \sum_{\overline{w}\in\Psi}\left(||\pi \cap \overline{w}|| - \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2}\right)$$

$$= \sum_{\overline{w}\in\Psi}\rho(\overline{w},\pi) = \sum_{\overline{w}\in\overline{D}}\rho(\overline{w},\pi) + \sum_{\overline{w}\in\Psi\setminus\overline{D}}\rho(\overline{w},\pi) \geq \sum_{\overline{w}\in\overline{D}}\rho(\overline{w},\pi)$$

as surplus $\rho(\overline{w},\pi)$ is always non-negative. Now,

$$\rho(\overline{w},\pi) = ||\pi \cap \overline{w}|| - \sqrt{X_{\overline{w}}^2 + Y_{\overline{w}}^2} \geq ||\pi \cap \overline{w}|| - \sqrt{\triangle_{\overline{w}}^2 + \triangle_{\overline{w}}^2} = ||\pi \cap \overline{w}|| - \sqrt{2}\triangle_{\overline{w}}$$

Thus we get $\xi_{\pi,2} \geq \sum_{\overline{w}\in\overline{D}}\rho(\overline{w},\pi) \geq \sum_{\overline{w}\in\overline{D}}(||\pi \cap \overline{w}|| - \sqrt{2}\triangle_{\overline{w}})$. □

Lemma 4.10. *Let $\overline{D}$ be the set of minimal m-dense windows visited by the algorithm when applied to the optimal path π of the k-TSP problem. Also let $u \geq 2$ be an arbitrary fixed integer, then $\xi_{\pi,u} \geq \sum_{\overline{w}\in\overline{D}}\frac{||\pi(\overline{w})\cap\overline{w}||}{2}$*

Proof. Let σ be the output of Mitchell's algorithm. Let $\boldsymbol{S}^u_{opt}(\pi)$ be an optimal u-skeleton for π with $\pi_1, \pi_2, \ldots, \pi_{u-1}$ as the breakup of π into subpath by the vertices of $\boldsymbol{S}^u_{opt}(\pi)$. Now,

$$\xi_{\pi,u} = \sum_{j=1}^{u-1}\xi_{\pi_j,2} \geq \sum_{j=1}^{u-1}\sum_{\overline{w}\in\overline{D}}(||\pi_j \cap \overline{w}|| - \sqrt{2}\triangle_{\overline{w}}) \text{ , by lemma 4.9}$$

$$= \sum_{\overline{w}\in\overline{D}}\sum_{j=1}^{u-1}(||\pi_j \cap \overline{w}|| - \sqrt{2}\triangle_{\overline{w}}) = \sum_{\overline{w}\in\overline{D}}(||\pi \cap \overline{w}|| - \sqrt{2}(u-1)\triangle_{\overline{w}})$$

Note that $||\pi \cap \overline{w}|| \geq ||\pi(\overline{w}) \cap \overline{w}||$. Also from lemma 4.5 we have $||\pi(\overline{w}) \cap \overline{w}|| \geq m.\triangle_{\overline{w}}$ for each $\overline{w} \in \overline{D}$. Taking $m \geq 2\sqrt{2}u$, we get

$$\xi_{\pi,u} \geq \sum_{\overline{w}\in\overline{D}}(||\pi \cap \overline{w}|| - \sqrt{2}(u-1)\triangle_{\overline{w}}) \geq \sum_{\overline{w}\in\overline{D}}\frac{||\pi(\overline{w}) \cap \overline{w}||}{2}$$

□

Here is the proof of theorem 4.3 assuming old analysis for k-TSP problem (theorem 4.2).

Theorem 4.3. For any path $\pi = \langle p_1, p_2, \ldots, p_k \rangle$ and a constant $u > 0$ in a graph G on a euclidian metric space, it is possible to find in polynomial time a path starting at p_1, covering at least k points, and of maximum length $||\pi|| + \frac{\xi_{\pi,u}}{u}$.

Proof. From theorem 4.2, for every m-dense window $\overline{w} \in \overline{D}$, the path σ output by the algorithm inside $\overline{w}$ is of length $\leq (1+1/m).||\pi(\overline{w}) \cap \overline{w}||$. Thus if we sum errors introduced in each window, we get

$$||\sigma|| - ||\pi|| \leq \sum_{\overline{w} \in \overline{D}} \frac{||\pi(\overline{w}) \cap \overline{w}||}{m} = \frac{2}{m} \sum_{\overline{w} \in \overline{D}} \frac{||\pi(\overline{w}) \cap \overline{w}||}{2} \leq \frac{2\xi_{\pi,u}}{m} \leq \frac{\xi_{\pi,u}}{u}$$

Here the second-last inequality is give by lemma 4.10 □

5 Why Arora's algorithm doesn't work?

Arora's PTAS algorithm (say $\mathcal{A}$) for euclidian k-TSP problem has the benefit over Mitchell's algorithm (say $\mathcal{M}$) that it can be generalized to more than two dimensions. However, the improved analysis for euclidian k-TSP problem is true only for $\mathcal{M}$ in two dimensions, or a generalization of $\mathcal{M}$ with $\mathcal{A}$ as a subroutine for higher dimensions (see section 4.4). The reason that $\mathcal{A}$ cannot be generalized is that it divides the current window into four equal regions with portals in between. The portals are cleverly placed such that a PTAS can be obtained for k-TSP problem but the higher level lines (with large interportal distance) always incur an error that is high enough to not obtain the improved tighter bound for k-TSP problem. On the other hand for $\mathcal{M}$, one does not incur any error as long as the grid has an m-perfect cut with respect to the optimal path for the current window. It is only when the window is m-dense that $\mathcal{M}$ incurs an error. But then as shown in lemma 4.10, u-excess is 'large' enough for us to ensure that the total error still remains less than $\frac{\xi_{\pi^*,u}}{u}$.

6 Conclusions

This report is primarily based on two papers. After the introduction, the first half of the report explains the 4-approximation algorithm to the school bus problem on tree metric by Bock et al. [3]. In the second half of the report we discuss (ϵ, u)-PTAS algorithm to the euclidian rooted orienteering path problem. Other than improving the approximation factors given in table 1 (not possible for PTAS), some of the other interesting 'open' (see note below) problems are:

- Obtaining/improving hardnes results for SBP (the authors of [3] claim to have obtained some hardness results that will be appearing in the journal version of their paper).
- Obtaining a constant factor approximation algorithm for SBP or DVRP on euclidian metric or planar graphs.

- Constant factor approximation algorithm for SBP on trees with a multiplicative regret.
- PTAS for s–t orienteering problem on euclidian metric.
- PTAS for orienteering problem on planar graphs (some breakthrough results have appeared on planar graphs for related problems, like TSP, in the past decade).

Note: *The above list of open problems is according to the knowledge of the authors of this report. It is possible that some of them have been already resolved!*

Acknowledgements

This report and a presentation were prepared as a course project for the course CO-754 at University of Waterloo. We are thankful to the instructor of the course Prof. Joseph Cheriyan for suggestions and feedback on the project. We are also thankful to the course TA Isaac Fung, and to our fellow classmates that were present during our presentation as their questions and comments were helpful in deciding the topics that need more elaborate discussion in the report.

References

[1] Sanjeev Arora: Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. J. ACM 45(5): 753-782 (1998)

[2] Avrim Blum, Shuchi Chawla, David R. Karger, Terran Lane, Adam Meyerson, Maria Minkoff: Approximation Algorithms for Orienteering and Discounted-Reward TSP. FOCS 2003: 46-55

[3] Adrian Bock, Elyot Grant, Jochen Könemann, and Laura Sanità: The School Bus Problem on Trees. Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC), 2011

[4] C. Chekuri, N. Korula, and M. Pal: Improved Algorithms for Orien- teering and Related Problems. in SODA, pp. 661670, 2008

[5] Ke Chen, and Sariel Har-Peled: The Euclidean Orienteering Problem Revisited. SIAM J. Comput., 2008: 385-397

[6] Naveen Garg: Saving an epsilon: a 2-approximation for the k-MST problem in graphs. STOC 2005: 396-402

[7] Joseph S. B. Mitchell: Guillotine Subdivisions Approximate Polygonal Subdivisions: A Simple Polynomial-Time Approximation Scheme for Geometric TSP, k-MST, and Related Problems. Journal SIAM Journal on Computing, Volume 28 Issue 4, Aug. 1999

[8] V. Nagarajan, and R. Ravi: (2008): Approximation Algorithms for Distance Constrained Vehicle Routing Problems. Tepper School of Business, Carnegie Mellon University, Pittsburgh 2008