Title

The Traveling Salesman Problem with Flexible Coloring

Authors
Roemer, T. A.
Ahmadi, R.
Dasu, S.

Publication Date
2011-10-01

The Traveling Salesman Problem with Flexible Coloring

Thomas A. Roemer
The Rady School of Management
University of California, San Diego
troemer@rady.ucsd.edu

Reza Ahmadi
Anderson School of Management
University of California at Los Angeles
rahmadi@anderson.ucla.edu

Sriram Dasu
Marshall School of Business
University of Southern California
dasu@marshall.usc.edu

October 2011

Abstract

This paper introduces a new generalized version of the Traveling Salesman Problem (TSP) in which nodes belong to various color classes and each color class must be visited as an entity. We distinguish the cases of the problem for which the colors are either pre-assigned or can be selected from a given subset of colors. We establish computational complexity and provide concise formulations for the problems that lend themselves to derive tight lower bounds. Exact solutions for special cases and a two-phase heuristic for the general case are provided. Worst case performance and asymptotic performance of the heuristic are analyzed and the effectiveness of the proposed heuristic in solving large industrial size problems is empirically demonstrated.

.
.
.

KEYWORDS: Traveling Salesman Problem, Complexity, Heuristics, Probabilistic Analysis, Error Bounds

1 Introduction

We consider a weighted, complete graph, $G(V, E)$, in which the edge weights satisfy the triangular inequality. Each vertex or node v_i $(i = 1, ..., N)$ must be assigned one "color" $k \in K_i$ from its allowable color set K_i. There are a total of K colors, that is $\left|\bigcup_{i=1}^{N} K_i\right| = K$. A problem of some practical interest is to find the shortest tour that visits all vertices, "paints" each vertex with one of its allowable colors, and returns to the starting position, while visiting all vertices of the same color consecutively. Note that if all nodes are completely flexible, that is if they can be colored with any of the K colors (i.e., $|K_i| = K\ \forall i$), then the problem reduces to the standard *Traveling Salesman Problem* (TSP). On the other hand, if none of the nodes are flexible, such that each node must be painted with a specific color (i.e., $|K_i| = 1\ \forall i$), then the problem becomes the *Clustered Traveling Salesman Problem* ($CTSP$), first introduced by Chisman (1975).

Ahmadi and Mamer (1999) provide a manufacturing example for $CTSP$, in which "pick and place" robots mount components on circuit boards. Due to the high switch-over costs between component types, each type has to be mounted consecutively and thus constitutes a cluster or color. This paper is motivated by the latter example, however here we consider the case with additional flexibility. In particular, when some components can be mounted by a variety of different nozzles, then additional routing flexibility arises. The same problem structure also arises during functional testing in the printed circuit card (PCB) assembly environment. In functional testing of PCBs, test probes are brought in contact with the test pads for input/output signal measurement. Two widely used technologies are pin-through-hole and surface-mounted. Usually the set of pads in the PCB are partitioned into subsets of electronically equivalent pads referred to as nets. Typical PCBs have more than 1000 nets and 10,000 pads. The problem that the test engineers face is the assignment of different size probes to a set of pads to avoid creation of short circuits. Each pad could be tested with a subset of probes and given the economics of testing, pads with similar test probes are performed contiguously. In CNC-machining for example, some activities can be performed by several different tools so that two distinct operations may be performed consecutively without switching tools and incurring setup times. Many testing operations in electronic circuitry require specialized equipment for some areas and multipurpose tools for other regions. (Sequential) dispatching of customer service representatives falls into this class of problems as well, if the representatives have different skill sets and the customers differing needs.

Because it is convenient to think of the node properties as colors, we refer to this problem as the *Traveling Salesman Problem with Flexible Colors* ($FCTSP$). We note, however, that the problem is not limited to this realm, but arises frequently in the presence of switching costs between specialized and general equipment.

Applications for the special case of the $CTSP$ are abundant and include order filling in warehouses (Chisman, 1975) or emergency vehicle dispatching

(Weintraub, 1999). In addition to a number of applications in manufacturing and vehicle routing, Laporte and Palekar (2002) discuss applications in areas as varied as computer disk defragmentation, computer programming, examination timetabling, and cytology.

In this paper, we will first formulate the $FCTSP$, and then, based on this, the $CTSP$ as well. The formulations are developed with the subsequent derivation of lower bounds in mind. Section 3 establishes the computational complexity of the $FCTSP$ and Section 4 discusses special cases that can be solved efficiently. In Section 5, we develop lower bounds on the optimal solutions based on Lagrangian relaxation of the formulations from Section 2. Section 6 presents heuristic procedures to solve the $FCTSP$ and the $CTSP$ and Sections 7 and 8, respectively, evaluate the quality of the heuristics and the lower bounds analytically and empirically.

2 Problem Formulation

The *Traveling Salesman Problem with Flexible Colors* ($FCTSP$) can formally be expressed as follows:

$$Min \sum_i \sum_j c_{ij} x_{ij} \tag{1}$$

s.t.

$$\sum_{j \neq i} x_{ij} = 1 \quad \forall i \in V \tag{2}$$

$$\sum_{i \neq j} x_{ij} = 1 \quad \forall j \in V \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, \;\; S \neq V, \;\; S \neq \{\} \tag{4}$$

$$\sum_{k \in K_i} w_{ik} = 1 \quad \forall i \tag{5}$$

$$\alpha_{ij} \geq w_{ik} - w_{jk} \quad \forall k, i, j \tag{6}$$

$$x_{ij} \leq 1 - \alpha_{ij} + y_{ij} \quad \forall i \neq j \tag{7}$$

$$\sum_{i,j \in N} y_{ij} \leq \widehat{K} \tag{8}$$

$$x_{ij} \in \{0, 1\} \tag{9}$$

$$y_{ij} \in \{0, 1\} \tag{10}$$

$$w_{ik} \in \{0, 1\} \tag{11}$$

$$\alpha_{ij} \in \{0, 1\} \tag{12}$$

Explanation of Variables:

$$x_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are adjacent and of different colors} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ik} = \begin{cases} 1 & \text{if node } i \text{ is of color } k \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are of different colors} \\ 0 & \text{otherwise} \end{cases}$$

Objective function (1) minimizes the total cost of the tour, in which x_{ij} indicates that the tour includes edge (i, j) and c_{ij} is the weight of this edge, (i.e. the Euclidean distance, between the two corresponding nodes). Constraints (2) and (3) warrant that each node "precedes" and "succeeds" exactly one other node.[1] The loop breaking constraints (4) warrant that the solution is a connected graph, and therefore a tour. Constraints (5) assign exactly one color to each node and Constraints (6) force indicator variable $\alpha_{ij} = 1$ if nodes i and j are assigned a different color. The logical constraints (7) force $y_{ij} = 1$ if two nodes of a different color are joined by an edge in the solution. Constraint (8) limits the number of color changes. Ideally the number of allowable color changes, $\widehat{K}$, is minimized, giving rise to the APX-hard set covering problem (Arora and Lund, 1997). However, many heuristics exist that have been shown to work well in practice (Chen et al., 2010). For the remainder of the paper, we therefore assume that $\widehat{K}$ is given. Notice that for the $CTSP$, all w_{ik} are given and that, by Equation (6) , all α_{ij} are given *apriori* and are thus no longer decision variables. Given the values for the α_{ij}, a formulation for $CTSP$ can then be obtained by removing all constraints containing w_{ik}:

$$Min \sum_i \sum_j c_{ij} x_{ij}$$

s.t. $(2) - (4)$ and $(7) - (10)$.

3 Computational Complexity

Since the TSP is strongly NP-hard and a special case of both, the $FCTSP$ and the $CTSP$, it follows immediately that both problems are strongly NP-hard. Unfortunately, as will be shown in this section, for the $FCTSP$ matters are still worse. In contrast to the (Euclidean) TSP, which can be polynomially approximated (Arora, 1998), no such approximation can exist for the $FCTSP$ as it will be shown that it belongs to the class of $APX-$complete problems (Papadimitriou and Yannakakis, 1991). Moreover as the proof will show, the

[1] If $x_{ij} = 1$, then we say that node i *precedes* node j, or equivalently, that node j *succeeds* node i, even though the actual direction in the final tour might be in the reverse order.

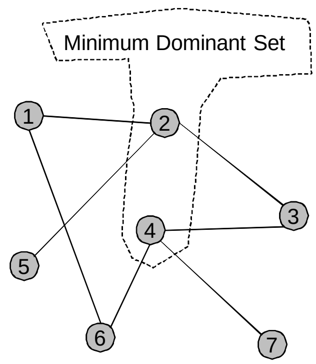

Figure 1: A Graph And Its Minimum Dominant Set

problem remains $APX-$complete, even if all nodes are on a line and if $|K_i| \leq 4$ $\forall i$.

Theorem 1 *The FCTSP is APX-complete.*

The proof will be by an L-reduction to the *Minimum Dominating Set-3 Problem*, which was shown to be APX-hard by Alimonti and Kann (1997). Figure 1 shows a graph and its minimum set, defined as follows:

Definition 2 (Minimum Dominant Set-3 Problem) *Given a graph,* $G = (V, E)$ *, with maximum vertex degree 3, find a subset* $V' \in V$ *of smallest cardinality* $|V'|$*, such that every vertex in* V *is also in* V' *or adjacent to a vertex in* V'*.*

Notice that the (unique) minimum dominating set (MDS) in Figure 1 is set $S = \{2, 4\}$, as all nodes in the graph are either in S or are adjacent to a node in S. Before we formally prove Theorem 1, we will explain the intuition and (the simplified) reduction. Consider again the graph in Figure 1. To create our instance of the $FCTSP$, we create two collinear line segments of length 1 and place $N = 7$ nodes (the number of nodes in Figure 1) on each segment as in Figure 2. Each of the nodes on the left-hand side can only be colored with one color and all colors are distinct. For these nodes, denote the color assigned to them by the number of the node.

The nodes on the right-hand side are clones of the former, and can be colored in the node's own color and that of all its neighbors. For example, node 2 in Figure 1 requires color 2 and is adjacent to nodes 1, 3, and 5. Consequently, for node 2', the set of allowable colors is $K_{2'} = \{2, 1, 3, 5\}$. Our claim is that the optimal tour passes the chasm of width $M - 4$ between the two line segments exactly once for each member of the dominating set. Moreover, the elements of the dominating set are the left-hand side nodes incident to the edges crossing the chasm. Figure 2 shows the optimal path and identifies the dominating set $S =$

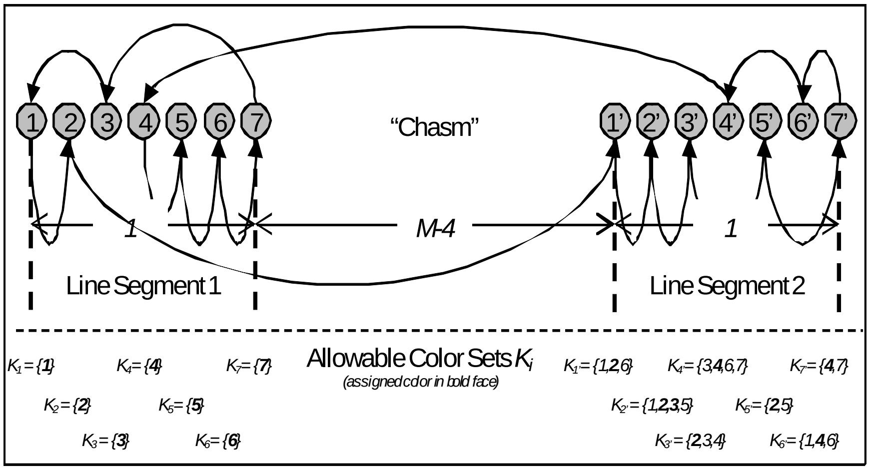

Figure 2: Reduction Instance

$\{2, 4\}$. Now, since M can be chosen to be arbitrarily large, any approximation of the optimal solution would also have to yield the dominating set. Matters are a bit more complicated if the dominating set contains an odd number of nodes because the chasm must be crossed an even number of times to yield a tour. Therefore we will assume throughout the proof that the cardinality, $|D|$, of the minimum dominating set D is even. There is no loss of generality in this assumption as we can simply extend our search to two disconnected isomorphic graphs, thus forcing the cardinality to be even.

Proof of Theorem 1. For a given instance of the $FCTSP$, create two collinear line segments of length 1 and let the distance between those line segments be $M-4$ as depicted in Figure 2. On each of these line segments, align N nodes equidistantly in increasing order of their indices. Let the color set for the nodes on line segment 1 (i.e., nodes $v_1, ..., v_N$) be $K_i = \{i\}$. Let $N(v_i)$ be the set of neighbors of node v_i in graph G. The color sets for nodes $v_{N+1}, ..., v_{2N}$ on line segment 2 are $K_i = \{i - N\} \cup N(v_{i-N})$.

Denote the optimal solution value to this instance of the $FCTSP$ by Z^* and let $|D| = m$. Suppose (for now) that D is given. We can then construct a feasible tour as follows and as shown in Figure 2. Starting for instance with node $v_k \in D$, we can visit all nodes on line segment 2 that can be painted with color k. Next, we can visit all nodes on line segment 2, that can be colored with one of the colors of a node in D that has not yet been visited, say color g. Then only one node of color g remains to be visited, i.e. node v_g on line segment 1. It is easily seen that by visiting all nodes in D in this manner, all nodes on line segment 2 are visited. Therefore only nodes on line segment 1 not in D remain to be visited. Since these nodes are all of different colors, this can be done by simply moving once up and down line segment 1. The large "chasm" in Figure 2 is therefore crossed exactly m times and, after each such crossing, line segments 1 and 2 must at most be traveled twice. It follows that

$$Z^* \leq mM. \tag{13}$$

Following the procedure outlined in Papadimitriou and Yannakakis (1991) we show next that given a tour π, with cost Z we can find, in polynomial time, a solution to the MDS problem with cost at most $m+Z-Z^*$, with m being the optimal solution to the MDS problem, i.e. every r-approximation algorithm for $FCTSP$ yields a $c(r)$-approximation algorithm for $3-DS$. Let S_h denote the nodes on line segment $h \in \{1,2\}$ and let S denote those nodes in S_1 that are adjacent to a node from set S_2 on path π. Notice that set S is a dominating set of graph G. Otherwise, there would have to be a node $j \notin S$ not adjacent to any node in S. But, by the construction of line segment 2, this implies that node $N+j$ cannot be painted with any color $i \in S$, contradicting that π is a feasible tour. Therefore, for a given tour π, we can easily compute a solution to the MDS problem with objective function value $\widetilde{m} = |S|$. It remains to show that

$$\widetilde{m} \leq m + Z - Z^*. \tag{14}$$

This is obviously true for $\widetilde{m} = m$ since $Z \geq Z^*$. Thus suppose $\widetilde{m} = m + n$, in which n is any strictly positive integer. Clearly

$$m + Z - Z^* \geq m + (m+n)(M-4) - Z^* \tag{15}$$

since path π covers at least $\widetilde{m}$ times distance $M-4$. Since $Z^* \leq mM$, we also have that

$$m + Z - Z^* \geq m + nM - 4m - 4n, \tag{16}$$

which for $M = 4N + 5$ yields

$$m + Z - Z^* \geq m + 4nN - 4m + n > m + n = \widetilde{m} \tag{17}$$

since the cardinality of a dominating set cannot exceed the number of nodes N in a graph. ■

Notice that the reduction instance has a very simple structure, in which all nodes are on a line. Moreover, the instance only requires limited flexibility; that is, no node requires more than four color choices. The corollary follows immediately.

Corollary 3 *The $FCTSP$ remains $APX-$complete, even if all nodes are on a line and even if the color choices for each node are limited to at most four, that is if $|K_i| \leq 4$ $\forall i$.*

4 The Single Line Problem

Corollary 3 gives rise to the question of whether there are any single line versions of the $FCTSP$ that can be solved efficiently, and in particular the single line version of the $CTSP$ (henceforth referred to as $L - CTSP$). Discussion of this problem is also interesting as the solution we will introduce shortly, motivates the general solution procedure presented later. Finally, two-dimensional Euclidean spaces are often mapped onto one-dimensional spaces using space filling curves and related procedures to facilitate simpler solution procedures (Bartholdi and Platzman, 1988). In this section, we therefore investigate several single line variations of the $FCTSP$ and present efficient solutions. To do this, we first establish a general property motivated by the TSP that all optimal solutions must satisfy. In particular, one of the fundamental results from the TSP for planar graphs shows that any tour with edge crossings is suboptimal (Johnson and Papadimitriou, 1986). Unfortunately, as the example in Figure 3 shows, this result no longer holds in the $FCTSP$. However, we will show that no two crossing edges can "share a color" and that at least one of the two edges must be "monochromatic".

Definition 4 *Let $C(u)$ denote the color assigned to node u. Set $C(u,v) \equiv C(u) \cup C(v)$ is said to be the colors of edge (u,v). Edge (u,v) is said to be monochromatic if $|C(u,v)| = 1$, that is if $C(u) = C(v)$.*

Lemma 5 *If in any optimal solution edges (u,v) and (u',v') cross, then $C(u,v) \cap C(u',v') = \{\}$ and at least one of the two edges is monochromatic.*

Proof. Let there be an optimal tour $(u, ...u', v', ...v, u)$ such that edges (v,u) and edges (u',v') cross. Notice that $(u, ...u', v, ...v', u)$ is also a tour as it merely traverses path $(v', ..., v)$ in the opposite direction, while it replaces edges (v,u) and (u',v') with edges (v',u), and (u',v). It is a well known result that this tour is no longer than the initial tour (Johnson and Papadimitriou, 1986). It remains to show that this exchange does not split up a path through one color. First, suppose that neither edge is monochromatic. Clearly in that case, no path through a color is interrupted and the resulting tour remains feasible. Without loss of generality assume that $C(v) = C(u)$. Now suppose that $C(u,v) \cap C(u',v') \neq \{\}$ and, without loss of generality, let $C(u) = C(u')$. Unless $C(v) = C(v')$, there must be a path of color $C(u)$ from node u to u'. Thus, after replacing edges (v,u) and (u',v') with edges (v',u), and (u',v), the nodes of color $C(u)$, that is u, u', and v are still connected by a path of color $C(u)$. If $C(v) = C(v')$, then all four nodes have the same color and must also still be connected by a path of color $C(u)$. ■

4.1 The Single Line Problem for Deterministic Nodes $(L - CTSP)$

We first note that the $L - CTSP$ is similar to the Rural Postman Problem on a Line, which has been solved in polynomial time by van den Berg (1996) and to the problem order execution of queries in linear storage proposed and solved by Kollias et al. (1990). Here, we provide a shorter and perhaps more intuitive alternative solution procedure. The proposed procedure serves also as the intuitive building block for the general heuristic procedure developed in Section 6.

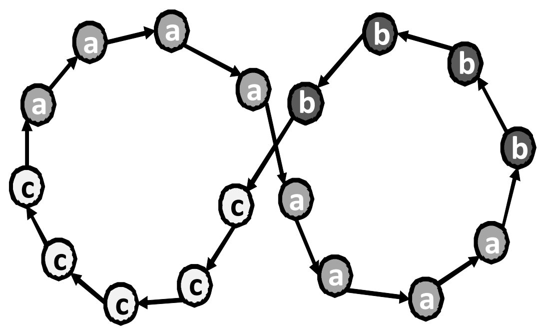

Figure 3: Example for Optimal Paths with a Crossing

Algorithm 6 **(**$L - CTSP$**)** *Step 1: For each color determine the leftmost and rightmost nodes. Delete all other nodes.*

Step 2: Join each pair of nodes of the same color by an edge. Re-index the nodes from left to right.

Step 3: Add edge (v_{2k-1}, v_{2k}) *for* $k = 1$ *to* K*.*

Step 4: Connect the independent components (if any) by a set (CES) *of edges with minimum total weight.*

Step 5: Double the edges added in Step 4.

Step 6: Construct a Eulerian tour.

Step 7: Construct the optimal tour on the original graph with all N *nodes: Whenever the Eulerian tour visits two nodes of the same color consecutively, visit all nodes of the same color in between. Otherwise, follow the route of the Eulerian tour.*

Figure 4 illustrates the procedure. Step 1 simplifies the problem by simply ignoring all nodes of the same color, except for the two nodes on the extreme sides of the line. As, per Lemma 5, each color will be visited in a straight line it is sufficient to know the endpoints. Step 2 joins these pairs by an edge, the solid lines in the upper half of Figure 4, and re-indexes the nodes from left to right. Central to the algorithm is Step 3; it recognizes that in order to visit all nodes in the graph and to return to the starting node, (i.e., to complete a tour) each segment of the line has to be traversed an even number of times. Since, in Step 2, exactly one edge terminates or originates at each node, every other segment between two adjacent nodes is crossed by an odd number of edges. Thus, the algorithm adds one edge to all "odd segments". Step 4 adds an edge set with minimum total weight that connects all components of the graph obtained in Steps 1 to 3, and Step 5 simply doubles the edges just obtained. Step 6 constructs a Eulerian Path on the *auxiliary graph* constructed in Steps 1 through 5 and, based on this Eulerian Path, Step 7 creates the actual tour.

Proposition 7 *The* $L-CTSP$ *Algorithm solves* $L-CTSP$ *optimally in* $O(N \log N)$ *time.*

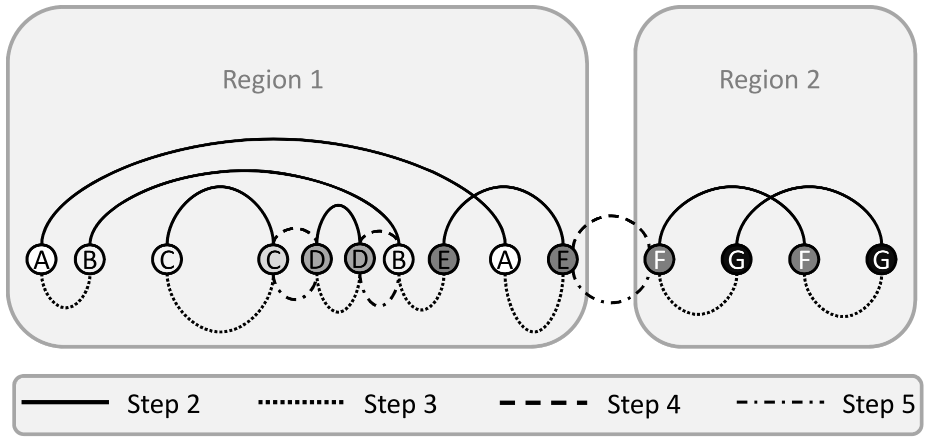

Figure 4: Illustration of the $L-CTSP$ Algorithm

Proof. Clearly, any feasible tour must contain the paths through the colors, that is all the edges added in Step 2, say set E_2. Since the optimal path must be a tour, each segment between two consecutive nodes must be traversed an even number of times. As Step 2 always has an odd number of edges traversing segment v_{2k-1}, v_{2k}, all edges added in Step 3, say set E_3, must also be contained in the optimal solution. If the resulting graph is not connected, then the shortest way of traveling back and forth between the components must be at least twice the length of the CES, that is the length of the edges added in Steps 4 and 5, say set E_4. It is easily seen that any edge in $E_3 \cup E_4$ joins two adjacent nodes. Therefore if $E_3 \cap E_4 = \{\}$, then the length of the tour constructed in $L-CTSP$ is a lower bound for the optimal solution. But if $(v_i, v_j) \in E_3$, then nodes v_i and v_j must be in the same component and hence $(v_i, v_j) \notin E_4$. Conversely, if $(v_i, v_j) \in E_4$, then nodes v_i and v_j must be in independent components and hence $(v_i, v_j) \notin E_3$.

It remains to show that the algorithm yields a feasible tour. A tour is feasible if it visits all colors without interruption and returns to the starting position. Clearly this is the case for any tour based on a Eulerian Path and it remains to show that the auxiliary graph created in Steps 1 to 4 of the $L-CTSP$ Solution Algorithm is a Eulerian graph. After Step 2, each node has degree 1 as pairs of nodes are connected. Step 3 adds one edge to each node, so that all vertex degrees are even. Finally, Steps 4 and 5 add two edges to select pairs of nodes, so that all vertex degrees remain even. Steps 1 and 7 can both be performed in $O(N)$ time, whereas Steps 3, 5, and 6 require only $O(K)$ time. In Appendix I, we provide a procedure to generate the edge set in Step 4 in $O(K \log K)$ time, such that the running time is $O(\max\{N, K \log K\})$. However as we need $O(N \log N)$ time to order the nodes on the line, the entire procedure will take $O(N \log N)$ time. ■

4.2 The Single Line Problem for Fully Flexible Nodes $(L - FFCTSP)$

The above algorithm extends in simple fashion to the special case $(FFCTSP)$ in which any node is either entirely flexible, (i.e., can be colored with any color) or can only be colored with one specific color. For this case, the $L - CTSP$ algorithm can be extended to the following $O(\max\{N, K \log K\})$ algorithm.

Algorithm 8 *Step 1: Let v_ℓ, and v_r, respectively be the leftmost and rightmost fixed nodes. Delete all flexible nodes between v_ℓ, and v_r.*

Step 2: Run Steps 1 to 6 of the $L - CTSP$ for the fixed nodes.

Step 3: Construct the optimal tour: If $v_1 \neq v_\ell$ then start at node v_1 coloring all nodes up to v_ℓ with $C(v_\ell)$. Follow the Eulerian tour π generated in Step 6 of the $L - CTSP$ Algorithm. For each edge $(i, j) \in \pi$, visit all not yet visited nodes on edge (i, j) that are of color $C(i)$ or that are flexible. Assign color $C(i)$ to the latter. If $v_N \neq v_r$, then, upon reaching v_r, visit all nodes up to v_N and color them with $C(v_r)$. Return to v_r and keep following the Eulerian tour π. Upon reaching v_ℓ, return to v_1.

Step 1 deletes all flexible nodes between the leftmost and rightmost fixed nodes. Step 2 determines an Eulerian Path through all fixed nodes by applying $L - CTSP$. Step 3 constructs the optimal tour by painting all flexible nodes in the beginning with $C(v_\ell)$, all flexible nodes at the end with $C(v_r)$, while applying the $L - CTSP$ Heuristic to all nodes in between, picking up flexible nodes whenever it encounters them.

Proposition 9 *The $L - FFCTSP$ Algorithm solves the $L - FFTSP$ problem optimally in $O(N \log N)$ time.*

Proof. Notice that as before, each segment of the line needs to be traversed at least twice. Thus segments v_1, v_ℓ or v_r, v_N, to the extent that they exist, each have to be traversed at least twice. Moreover, the $L - CTSP$ solution value, say Z_{CTSP} is a lower bound for the nodes in interval v_ℓ, v_r. Thus, the $FFCTSP$ solution value, say Z_{FFCTSP} is bounded from below by

$$Z_{FFCTSP} \geq Z_{CTSP} + 2\left(c_{v_1, v_\ell} + c_{v_r, v_N}\right)$$

but is easy to see that the RHS is the objective function value obtained in the $L - FFCTSP$ Algorithm. It remains to show that the solution is feasible. Clearly, as node v_ℓ is colored with color $C(v_\ell)$, it is feasible to paint all flexible nodes to its left with $C(v_\ell)$ as well. The same logic holds for v_r. All other flexible nodes can be painted with any color and as the entire line from v_1 to v_N has been traversed, no flexible node is left uncolored, whereas the application of the $L - CTSP$ Heuristic warrants that all fixed nodes are properly colored as well. ■

5 Lower Bounds

Before we introduce heuristic procedures to solve the $FCTSP$ and the $CTSP$, we will first provide two Lagrangian relaxations to generate lower bounds on the optimal solutions for these two. Subsequently we will employ these bounds to evaluate the quality of our proposed heuristics. To strengthen these bounds, we add two additional redundant constraints to the $FCTSP$ and the $CTSP$:

$$\sum_i y_{ij} \leq 1 \quad \forall j \in N \tag{18}$$

$$\sum_j y_{ij} \leq 1 \quad \forall i \in N \tag{19}$$

5.1 Lagrangian Relaxation for the $FCTSP$

In order to get a lower bound, we remove the subcycle elimination constraints (4) and dualize the coupling constraints (6) and (7) by multiplying by γ_{ijk} and λ_{ij}, respectively.

$$\begin{aligned} L(\lambda) = \; & \min \sum_i \sum_j (c_{ij} + \lambda_{ij}) x_{ij} - \sum_i \sum_j \lambda_{ij} y_{ij} \\ & + \sum_k \sum_i \sum_j \gamma_{ijk} (w_{ik} - w_{jk}) \\ & + \sum_i \sum_j \alpha_{ij} \left(\lambda_{ij} - \sum_k \gamma_{ijk} \right) - \sum_i \sum_j \lambda_{ij} \end{aligned} \tag{20}$$

s.t. (2), (3), (5), (8), (9), (10), (11), (18), and (19). The problem separates into four subproblems as follows:

FCTSP − 1 (Lower Bound Problem 1):

$$L_1(\lambda) = \min \sum_i \sum_j (c_{ij} + \lambda_{ij}) x_{ij} \tag{21}$$

s.t. (2), (3), and (9), which is a simple assignment problem that can be solved by the Hungarian method in $O(N^3)$ time (Kuhn, 1955; Lawler, 1976).

FCTSP − 2 (Lower Bound Problem 2):

$$L_2(\lambda) = \max \sum_i \sum_j \lambda_{ij} \cdot y_{ij} \tag{22}$$

s.t. (8), (10), (18), and (19), which can be solved as a maximum weighted flow problem. This flow problem consists of three layers: N supply nodes with supply 1, N transshipment nodes, and one demand node with demand $\widehat{K}$. All supply nodes are connected to all transshipment nodes, which in turn are all

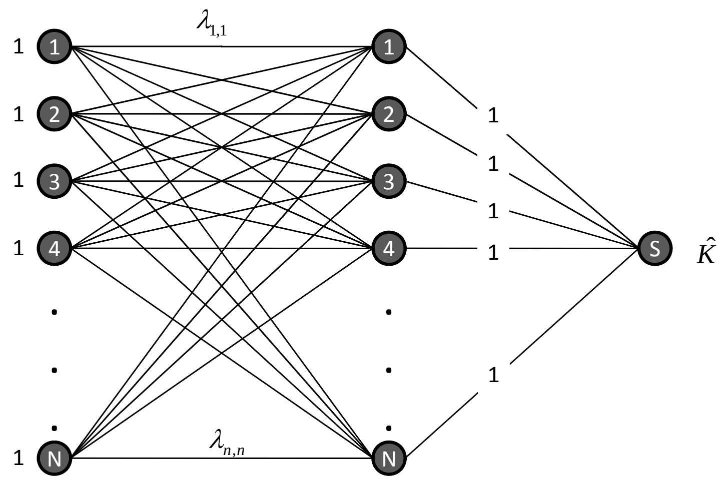

Figure 5: Maximum Weighted Flow Problem

connected to the demand node as depicted in Figure 5. Edge (i, j) between supply node i and transshipment node j has cost λ_{ij}. Edges incident to the demand node have capacity constraint 1.

FCTSP − 3 (Lower Bound Problem 3):

$$L_3(\gamma) = \max \sum_i \sum_k \sum_j \left(\gamma_{jik} - \gamma_{ijk}\right) w_{ik} \tag{23}$$

s.t. (5) and (11) , whose optimal solution is

$$w_{ik} = \begin{cases} 1 & \begin{array}{l} \text{for one (arbitrarily chosen) } k' \in K_i \\ \text{s.t. } \sum_j \left(\gamma_{jik'} - \gamma_{ijk'}\right) \geq \sum_j \left(\gamma_{jik} - \gamma_{ijk}\right) \quad \forall k \in K_i \end{array} \\ 0 & \text{otherwise.} \end{cases} .$$

FCTSP − 4 (Lower Bound Problem 4):

$$L_4(\lambda, \gamma) = \min \sum_i \sum_j \alpha_{ij} \left(\lambda_{ij} - \sum_k \gamma_{ijk}\right), \tag{24}$$

which is unconstrained and easily solved by inspection. The Lagrangian dual problem (D_{FCTSP}) can be represented as

$$D_{FCTSP} = L_1(\lambda) + L_2(\lambda) + L_3(\gamma) + L_4(\lambda, \gamma) - \sum_i \sum_j \lambda_{ij} \tag{25}$$

$$s.t. \;\; \lambda_{ij}, \gamma_{ijk} \geq 0,$$

which can be solved by subgradient optimization (Geoffrion, 1974; Fisher 1981).

5.2 Lagrangian Relaxation for the $CTSP$

In complete analogy, we can develop the following Lagrangian relaxation for $CTSP$

$$L(\lambda) = \min \sum_i \sum_j (c_{ij} + \lambda_{ij}) \cdot x_{ij} - \sum_i \sum_j \lambda_{ij} y_{ij} + \sum_i \sum_j (\alpha_{ij}\lambda_{ij} - \lambda_{ij}) \quad (26a)$$

s.t. (2), (3), (8) , (9), (10), (18), and (19), which decomposes into subproblems $FCTSP-1$ and $FCTSP-2$ as above with the minor modification, that for the $CTSP$, α_{ij} are no longer decision variables, but mere parameters known *apriori*. The resulting Lagrangian dual problem (D_{CTSP})

$$D_{CTSP} = L_1(\lambda) + L_2(\lambda) + \sum_i \sum_j (\alpha_{ij}\lambda_{ij} - \lambda_{ij}) \quad (27)$$

$$s.t. \;\; \lambda_i \geq 0$$

can again be solved by subgradient optimization (Geoffrion, 1974, Fisher 1981).

6 Heuristics

Now that we have developed procedures to obtain lower bounds, we have the means to evaluate the performance of heuristics to solve the $FCTSP$ and the $CTSP$. Common to both problems is that two principal challenges must be tackled: sequencing the colors and finding a path through each color. Since the latter problem is the $NP-$hard problem of finding a Hamiltonian Path (Garey and Johnson, 1979), we utilize well known results for this problem. In addition, to solve the overall TSP, we will borrow from Christofides (1976), who utilizes Minimum Matchings and Minimum Spanning Trees to derive good solutions to the TSP.

6.1 $CTSP$ Heuristic

In particular, we will first employ Hoogeveen's (1991) heuristic to determine an open-ended Hamiltonian path for each color. Next, we determine a Minimum Spanning Tree (MST) for the end nodes, match the *even* nodes in the MST, and then find a Eulerian path on the resulting graph:

Algorithm 10 ($CTSP-$Heuristic) *Step 1: Solve the open ended Hamiltonian path problem for each color using Hoogeveen's* (1991) *procedure. Define the two starting and ending nodes for each color. Call this set* $2K$*-Nodes.*

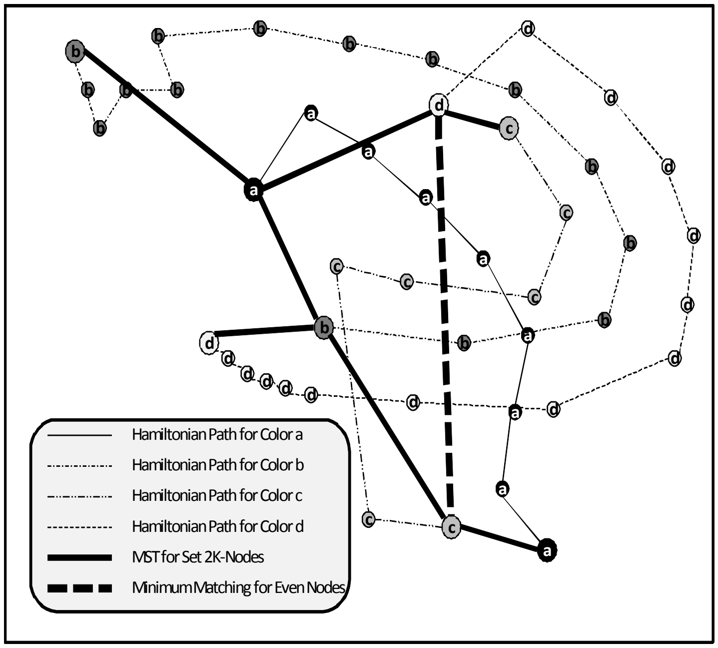

Figure 6: Illustration of the $FCTSP$ Heuristic

Step 2: *Find the Minimum Spanning Tree* (MST) *for the complete graph spanned by the* $2K$*-Nodes set.*

Step 3: *Find a Minimum Matching for the even degree nodes in the* MST *obtained in Step 2.*

Step 4: *Construct the graph,* G_E, *consisting of all* N *nodes and the edges determined in Steps 1 to 3.*

Step 5: *Determine a Eulerian Path through graph* G_E.

Step 6: *Apply shortcuts.*

Figure 6 illustrates the procedure for a problem with four distinct colors. The Hamiltonian paths, as determined per Step 1 of the $CTSP-$Heuristic, are represented by thin lines, the MST from Step 2 by the fat solid line, and the matching in Step 3 of the even nodes is represented by the fat dashed line. Notice that in Step 3, contrary to Christofides algorithm, *even* nodes are matched. The reason is simple, as in the underlying graph G_E, which contains the edges from the MST *and* from the Hamiltonian paths, each node in $2K-$Nodes has exactly one additional edge incident to it from the Hamiltonian path as seen in Figure 6. The resulting graph is then a Eulerian graph and it is easy to see that any Eulerian tour is a solution to the $CTSP$.

As the following proposition demonstrates, the worst case performance of the solution cannot exceed 300% of the optimal solution.

Proposition 11 *The performance ratio ρ of the CTSP Heuristic (i.e., the ratio between the heuristic solution Z_{CTSP} and the optimal solution Z^*) is bounded by $\rho \leq 3$.*

Proof. By definition, the shortest path through any given color is the optimal Hamiltonian path and any optimal solution must at least have the total costs of the Hamiltonian paths. Since Hoogeveen's heuristic finds paths no worse than $\frac{3}{2}$ the optimal paths the total costs of the edges determined in Step 1 must necessarily be less than $\frac{3}{2}$ the optimal solution. Step 2 finds a minimum spanning tree for a subset of the nodes whose cost cannot exceed the cost of the optimal tour. Finally, the matching in Step 4 cannot exceed the half the length of the optimal tour, such that the total cost of the edges in G_E cannot exceed the total cost of the optimal solution by a factor of 3. It remains to show that G_E is Eulerian. Clearly, all nodes not in set $2K$-Nodes have degree 2. Those nodes in set $2K$-Nodes that, after Step 2, are of even degree are matched and will be of odd degree after Step 3. Step 4 adds one edge from the Hamiltonian path to each node in set $2K$-Nodes so that all nodes are of even degree. ■

Clearly the tightness of this bound depends on Hoogeveen's bound on Hamiltonian paths, which, to the best of our knowledge has not yet been shown to be tight.

6.2 $FCTSP$ Heuristic

In addition to the challenges it shares with the $CTSP$, any solution to the $FCTSP$ must assign colors to individual nodes. We expect this step to be critical to the performance of the heuristic, as it turns the simple structure of $L-CTSP$ into the $APX-$hard $L-FCTSP$. In our proposed heuristic, we therefore try to pregroup nearby nodes into color classes and then apply the $CTSP-$Heuristic to the resulting problem.

Algorithm 12 ($FCTSP-$Heuristic) *Step 1: Solve the TSP for all nodes irrespective of color. Re-index the nodes according to the solution obtained.*

Step 2: Let graph $G_C = (V, E)$ be a graph such that $v_{i,k} \in V$ for $i = 1, ..., N$, if $k \in K_i$, and $(v_{i,k}, v_{i+1,m}) \in E \; \forall i, k, m$. Let the cost $C(v_{i,k}, v_{i+1,m}) = \begin{cases} 0 & \text{if } k=m \\ 1 & \text{otherwise} \end{cases}$. In addition, let $S \in V$, $(S, v_{1,k}) \in E$, and $C(S, v_{1,k}) = 1 \; \forall k \in K_1$, as well as $T \in V$, $(v_{N,k}, T) \in E \; \forall k \in K_N$, and $C(v_{N,k}, T) = 0$. Solve a shortest path problem from S to T on G_C.

Step 3: If $v_{i,k}$ is in the solution to the shortest path from Step 2 then assign color k to node v_i.

Step 4: Apply the $CTSP-$Heuristic to the resulting instance.

Step 1 performs any of the many heuristics that solve the TSP well in practice. Step 2 creates a shortest path problem as depicted in Figure 7. For example in the instance depicted, node 1 can be colored with color B or C. Similarly, node 5 can be colored with A, D, F, or G. Edges exist between two consecutive nodes and their costs are 0 if the nodes are of the same color, otherwise the costs are 1. Notice that any path from S to T will visit exactly one color of each node in order of the TSP solution. The shortest path through this network switches colors as few times as possible. The solution is used to assign colors to nodes so that the structure of the $CTSP$ obtains. Due to the special structure of graph G_C, the shortest path problem can be solved in linear time by the simple greedy algorithm presented in Appendix II.

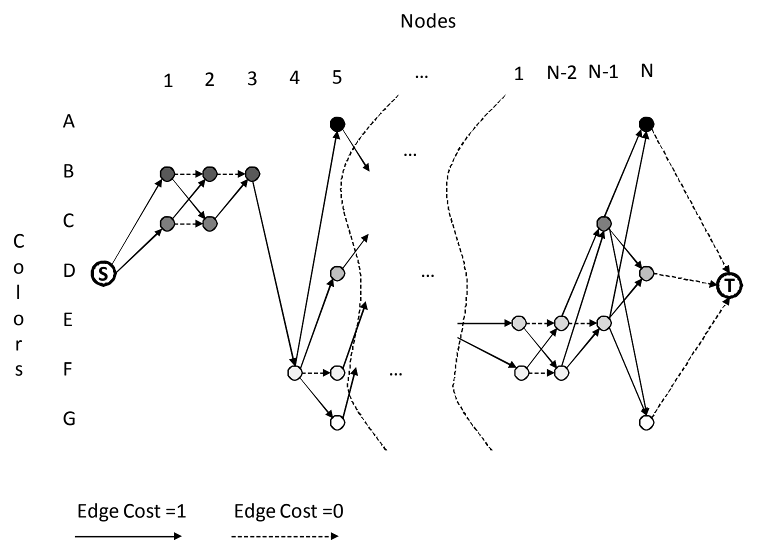

Figure 7: Color Assignment Network

Proposition 13 *When employing any TSP heuristic in the $FCTSP$-Heuristic with performance ratio 1.5, the performance ratio ρ of the $FCTSP$ Heuristic (i.e., the ratio between the heuristic solution Z_{FCTSP} and the optimal solution Z^*) is bounded by $\rho \leq 2K$.*

Proof. The heuristic solution consists of two parts: The Hamiltonian paths H_k for color k and the links between colors k and g, L_{kg}. With $|H_k|$ and $|L_{kg}|$ as the length of the Hamiltonian paths and links, respectively, the heuristic solution Z_{FCTSP} can be expressed as

$$Z_{FCTSP} = \sum_{k=1}^{K} |H_k| + \sum_{k=1}^{K} \sum_{g=1}^{K} |L_{kg}| \,. \tag{28}$$

Notice that finding a Hamiltonian path for a color is a subproblem to finding a Hamiltonian path through all nodes, so that $|H_k| \leq 1.5Z^* \; \forall k$ and that there are at most K Hamiltonian paths. Also, because of the triangle inequality, the distance between any two nodes cannot exceed the length of one half of any

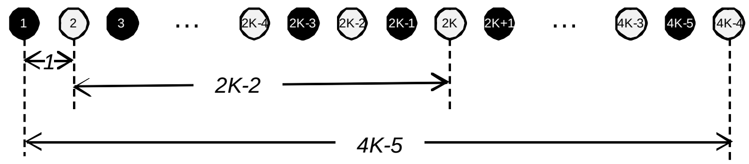

Figure 8: Instance with Arbitrarily Bad Performance of the $FCTSP$ Heuristic

tour, and $|L_{kg}| \leq 0.5Z^* \; \forall k, g$ follows. As there are at most K links, Equation (28) yields

$$Z_{FCTSP} \leq K \cdot (1.5Z^*) + K \cdot (0.5Z^*) = 2KZ^* \tag{29}$$

and the proposition follows. ■

Just as with Proposition 11, the tightness of the bound depends on the tightness of the bound on the Hamiltonian paths, which is still an open problem. However, Proposition 13 allows for the possibility that the performance becomes arbitrarily bad as the number of colors, K, becomes large. The following proposition shows that this is indeed the case.

Proposition 14 *The worst case performance of the $FCTSP-$Heuristic becomes arbitrarily bad as the number of colors becomes large.*

Proof. Consider the following instance with K colors and $N = 4K - 4$ collinear nodes with distance 1 between two adjacent nodes. As depicted in Figure 8 let the nodes be numbered from left to right and let color set K_i of node v_i be

$$K_i = \begin{cases} \{1\} & \text{if } i \text{ is odd} \\ \{2, ..., K\} & \text{if } i \text{ is even} \end{cases}. \tag{30}$$

Notice that none of the odd indexed nodes are flexible (i.e., they all require color 1) and that all even indexed nodes are almost completely flexible, (i.e., can be painted with any color except with color 1). Notice further that assigning color 2 to all nodes with even indices allows a solution of length $8K - 10$, in which all nodes of color 1 are visited first in increasing order of indices and then all nodes of color 2 in decreasing order of indices. It is easy to see, but not necessary for our proof, that this is indeed the optimal solution to this problem. A possible, if not likely, outcome of Step 1 of the heuristic is that the nodes will keep their initial indices. The resulting network structure, G_c, from Step 2 in the heuristic is depicted in Figure 9 and it is easy to see that any solution to this network has length $4K - 3$. In particular, the solution that assigns color 1 to all odd nodes and color $k + 1$ to nodes v_{2k} and $v_{2(K+k-1)}$ $(k = 1, ...K - 1)$ (indicated by bold arcs in Figure 9) is a shortest path through the network.

To visit all nodes of color 1, any tour must travel at least distance $4K - 6$ as that is the distance between node v_1 and node v_{4K-5}. Moreover, the distance between the $K-1$ pairs of color $k>1$ is $2K-2$, such that any tour must be larger than $K\cdot(2K-6)$. Consequently, we obtain for the performance ratio ρ of the heuristic

$$\rho > \frac{K\cdot(2K-6)}{8K-10} \tag{31}$$

which grows with K. ■

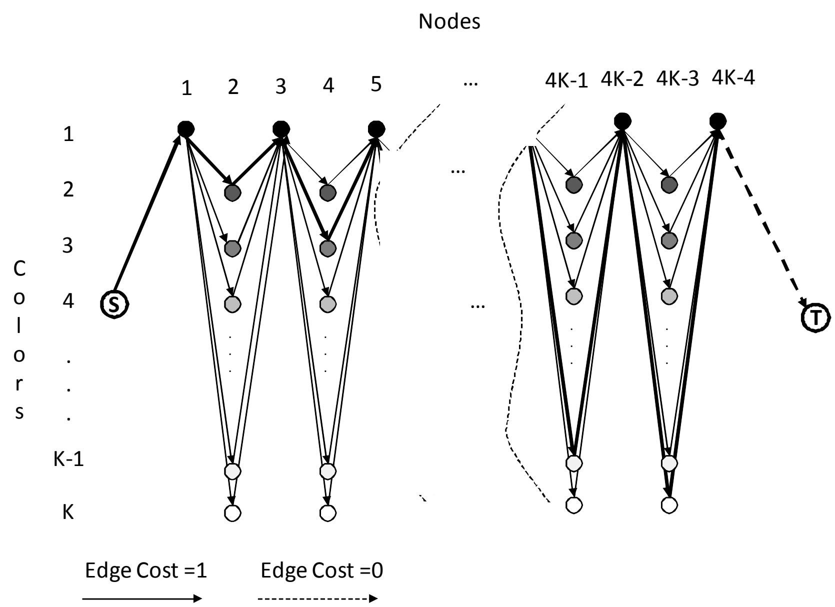

Figure 9: Network G_C for the Problem Instance

7 Asymptotic Analysis

Based on the seminal paper by Karp and Steel (1986), in this section we develop an asymptotic analysis of the heuristics when the location of the nodes is given by a set of independent identically distributed uniform random variables in $(0,1)\times(0,1)$. We derive the performance of the heuristic in the limit as the number of nodes grows incrementally to infinity. Let $V(N)$ and $H(N)$ denote the optimal and heuristic solutions, respectively.

7.1 Asymptotic Analysis of the $CTSP$ Heuristic

We consider two models for assigning colors to nodes. In model 1, colors are randomly assigned to nodes; that is, a node is of color k with probability β_k

and $\sum_{k=1}^{K} \beta_k = 1$. Depending on the application, we can expect different colors to have different spatial distributions. Accordingly, in model 2, we permit clustering of colors by geographic region. Nodes of color k lie in $J_k(N)$, non-overlapping sub-squares that lie inside $(0,1) \times (0,1)$. The same geography may be shared by two or more colors.

Proposition 15 *Both models for the CTSP yield*

$$\lim_{N\to\infty} \frac{H(N)}{V(N)} \leq 1 + O\left(\frac{1}{\sqrt{N}}\right). \tag{32}$$

Proof. Let $L_k(N)$ be the optimal length of the open ended Hamiltonian path for color k when the total number of nodes of all colors is N. By the Kolmogorov strong law of large numbers (Resnick, 2005) the number of nodes with color k converges almost surely to $\beta_k N$. In model 1, by Theorem 3 in Karp and Steele (1986)

$$\lim_{N\to\infty} \frac{L_k(N)}{c\sqrt{\beta_k N}} \to 1. \tag{33}$$

In model 2, for each color we first determine the Hamiltonian path in each sub-square and patch them together. If $\lim_{N\to\infty} \frac{J_k(N)}{N} \to 0\ \forall k$, then by Theorem 3 in Karp and Steele (1986) we get

$$\lim_{N\to\infty} \frac{L_k(N)}{c\sqrt{\beta_k N}} \to 1, \tag{34}$$

in which β_k is the expected proportion of nodes that have color k. The heuristic connects the Hamiltonian paths for each color, hence for large N

$$H(N) \leq \sum_{k=1}^{K} L_k(N) + K\sqrt{2} \tag{35}$$

almost surely, since the length of any arc is bounded by $\sqrt{2}$. Note also that

$$V(N) \geq \sum_{k=1}^{K} L_k(N), \tag{36}$$

almost surely and hence

$$\lim_{N\to\infty} \frac{H(N)}{V(N)} \leq \frac{\sum_{k=1}^{K} L_k(N) + K\sqrt{2}}{\sum_{k=1}^{K} L_k(N)} = 1 + \frac{K\sqrt{2}}{\sum_{k=1}^{K} L_k(N)} \tag{37}$$

such that

$$\lim_{N\to\infty} \frac{H(N)}{V(N)} \leq 1 + O\left(\frac{1}{\sqrt{N}}\right). \tag{38}$$

■

7.2 Asymptotic Analysis of the $FCTSP$ Heuristic

In the $FCTSP$ a set of colors is associated with each node . Let C be the set of distinct choices, then $K_i \in C$ for all i. Here too, we can either assume (model 1) that the nodes are randomly assigned to a color set, or (model 2) that each color set is restricted to a set of rectangular regions, similar to the structure in the $CTSP$. In either case, let β'_c correspond to the probability that a node is assigned a color set $c \in C$ and $\sum_{c \in C} \beta'_c = 1$. By assumption, the minimum number of distinct colors needed is $\widehat{K}$. Assign to each color set c a color $k \in C$, such that $\sum_{k=1}^{\widehat{K}} \sqrt{\beta_k}$ is minimized, where β_k is the probability that a node is assigned color k, with $\sum_{k=1}^{\widehat{K}} \beta_k = 1$. Clearly, β_k will depend on the color assigned to each set c.

Proposition 16 *Under both models, for FCTSP we get*

$$\lim_{N \to \infty} \frac{H(N)}{V(N)} \leq \frac{\sqrt{\widehat{K}}}{\sum_{k=1}^{\widehat{K}} \sqrt{\beta_k}} + O\left(\frac{1}{\sqrt{N}}\right). \tag{39}$$

Proof. By assumption, the minimum number of colors needed to cover all nodes is $\widehat{K}$. For large N, the optimal solution will include exactly $\widehat{K}$ Hamiltonian paths. By the Kolmogorov law of large numbers (Resnick, 2005) and theorem 3 in Karp and Steele (1986)

$$\lim_{N \to \infty} V(N) \geq \sum_{k=1}^{\widehat{K}} c\sqrt{N\beta_k},$$

almost surely.

The heuristic also consist of $\widehat{K}$ Hamiltonian paths. If the heuristic assigns color k to $\theta_k N$ nodes, then

$$H(N) \leq \sum_{k=1}^{\widehat{K}} c\sqrt{N\theta_k} + \widehat{K}\sqrt{2}.$$

Note that $1 \leq \sum_{k=1}^{\widehat{K}} \sqrt{\theta_k} \leq \sqrt{\widehat{K}}$ for θ_k such that $\sum_{k=1}^{\widehat{K}} \theta_k = 1$, yielding

$$\lim_{N \to \infty} \frac{H(N)}{V(N)} \leq \frac{\sum_{k=1}^{\widehat{K}} c\sqrt{N\theta_k} + \widehat{K}\sqrt{2}}{\sum_{k=1}^{\widehat{K}} c\sqrt{N\beta_k}} \leq \frac{\sqrt{\widehat{K}}}{\sum_{k=1}^{\widehat{K}} \sqrt{\beta_k}} + O\left(\frac{1}{\sqrt{N}}\right).$$

■

8 Computational Results

In this section, we evaluate the quality of the proposed heuristic and lower bounding procedures. Initially, we do so by solving problems of a small size

P	K	N	L(1)/OPT	L(2)/OPT	H(1)/OPT	H(2)/OPT	H(1)/L(1)	H(2)/L(2)
1	2	10	0.9915	0.9894	1.0098	1.0108	1.0184	1.0216
2		15	0.9677	0.9911	1.0386	1.0196	1.0733	1.0288
3		20	0.9661	0.9922	1.0075	1.0113	1.0429	1.0192
4	3	10	0.9941	0.9756	1.0271	1.0239	1.0332	1.0496
5		15	0.9688	0.9916	1.0415	1.0032	1.0750	1.0117
6		20	0.9922	0.9753	1.0404	1.0137	1.0486	1.0394
7	4	10	0.9951	0.9786	1.0169	1.0178	1.0219	1.0400
8		15	0.9677	0.9832	1.0019	1.0238	1.0353	1.0413
9		20	0.9734	0.9924	1.0200	1.0204	1.0478	1.0282
10	5	10	0.9802	0.9925	1.0174	1.0084	1.0379	1.0160
11		15	0.9721	0.9873	1.0212	1.0017	1.0505	1.0146
12		20	0.9867	0.9860	1.0380	1.0187	1.0520	1.0332
Average			**0.9796**	**0.9863**	**1.0233**	**1.0144**	**1.0447**	**1.0286**

Figure 10: Performance for Small Problem Instances

(up to 20 nodes and 5 colors) to optimality, using general purpose solution approaches. Afterwards, for large problem instances, we will show the quality of the proposed procedures by evaluating the ratio of the heuristic solutions to the lower bounds. Finally, we will compare our heuristic for $CTSP$ with some of the well known heuristics suggested in the extant literature.

For all problem instances created, the (x, y) positions of the nodes were generated from two independent uniform distributions. For each node i, the number of assigned colors $|K_i|$ was computed form a uniform distribution over 1 to K and the colors themselves were then assigned from a uniform distributions as well. Table 10 shows the performance of our procedures for small problem instances that we could solve to optimality. OPT, $L(i)$, and $H(i)$ for $i = 1, 2$ are, respectively, the values for the optimal solutions, lower bounds, and heuristic solutions obtained for problem i, where $i = 1$ indicates the $FCTSP$ and $i = 2$ indicates the $CTSP$. The reported ratios are the average results for ten problem instances. Table 10 indicates that the average lower bound was within 2.04% (i.e., $1 - 0.9796$) of the optimal solution for the $FCTSP$ and within only 1.37% for the $CTSP$. The error introduced by the heuristic for the $FCTSP$ is 2.33% (i.e., $1.0233 - 1$) and 1.44% for the $CTSP$. The cumulative errors introduced by the heuristics and lower bounding procedures are 4.47% for the $FCTSP$ and only 2.86% for the $CTSP$.

For the large problem instances we generated problem instances in likewise manner with up to 1000 Nodes and 100 Colors. Table 11, where each row represents the average of 20 experiments, shows performance results of the proposed heuristic for the large problem set. Overall, the $FCTSP-$Heuristic performs in average within 4.3% of the lower bound and the $CTS-$Heuristic within 4.2%, whereas the worst case performances never exceeded 9%.

Finally, we have compared the performance of the $CTSP-$Heuristic with some well known heuristics from literature. In particular, we used the algorithms proposed by Arkin et al. (1996), Guttmann et al. (2000), and Ahmadi

P	K	N	H(1)/L(1)	H(2)/L(2)	P	K	N	H(1)/L(1)	H(2)/L(2)
1	10	100	1.031	1.041	51	60	100	1.053	1.042
2		200	1.036	1.067	52		200	1.015	1.030
3		300	1.070	1.004	53		300	1.038	1.055
4		400	1.065	1.052	54		400	1.069	1.038
5		500	1.079	1.013	55		500	1.047	1.029
6		600	1.073	1.021	56		600	1.011	1.042
7		700	1.026	1.026	57		700	1.075	1.049
8		800	1.002	1.077	58		800	1.031	1.046
9		900	1.002	1.039	59		900	1.071	1.056
10		1000	1.034	1.047	60		1000	1.083	1.041
11	20	100	1.007	1.052	61	70	100	1.052	1.015
12		200	1.006	1.026	62		200	1.032	1.033
13		300	1.057	1.020	63		300	1.084	1.002
14		400	1.029	1.086	64		400	1.047	1.060
15		500	1.061	1.057	65		500	1.031	1.078
16		600	1.078	1.062	66		600	1.056	1.018
17		700	1.074	1.021	67		700	1.061	1.030
18		800	1.044	1.056	68		800	1.062	1.024
19		900	1.017	1.067	69		900	1.076	1.057
20		1000	1.059	1.089	70		1000	1.037	1.025
21	30	100	1.058	1.050	71	80	100	1.001	1.030
22		200	1.007	1.042	72		200	1.047	1.077
23		300	1.031	1.012	73		300	1.034	1.052
24		400	1.040	1.001	74		400	1.063	1.034
25		500	1.036	1.062	75		500	1.050	1.051
26		600	1.048	1.031	76		600	1.057	1.022
27		700	1.055	1.066	77		700	1.045	1.057
28		800	1.058	1.063	78		800	1.026	1.016
29		900	1.057	1.071	79		900	1.029	1.073
30		1000	1.016	1.067	80		1000	1.060	1.058
31	40	100	1.011	1.080	81	90	100	1.037	1.025
32		200	1.015	1.018	82		200	1.048	1.075
33		300	1.064	1.082	83		300	1.037	1.026
34		400	1.020	1.079	84		400	1.046	1.062
35		500	1.047	1.069	85		500	1.058	1.018
36		600	1.069	1.034	86		600	1.041	1.022
37		700	1.017	1.029	87		700	1.047	1.021
38		800	1.081	1.074	88		800	1.063	1.011
39		900	1.072	1.034	89		900	1.035	1.047
40		1000	1.004	1.085	90		1000	1.021	1.043
41	50	100	1.003	1.015	91	100	100	1.052	1.033
42		200	1.059	1.036	92		200	1.012	1.034
43		300	1.048	1.013	93		300	1.063	1.002
44		400	1.073	1.045	94		400	1.032	1.024
45		500	1.061	1.010	95		500	1.025	1.041
46		600	1.037	1.056	96		600	1.060	1.042
47		700	1.003	1.066	97		700	1.015	1.050
48		800	1.018	1.002	98		800	1.079	1.065
49		900	1.010	1.040	99		900	1.039	1.043
50		1000	1.048	1.033	100		1000	1.027	1.020
Average								**1.043**	**1.042**

Figure 11: Performance for Large Problem Instances

and Mamer (1999), respectively referred to as AHK, $GHKR$, and AM. In addition, by adding large penalties for switching between colors, we also investigate the performances of Christofides' algorithm ($CTFD$) and the Farthest Insertion algorithm (FI). Table 12 indicates that the $CTSP-$Heuristic tends to outperform the existing approaches. In particular, its average performance for this set of experiments is with in 3.3% of the lower bound, compared to 4.7%, 5.9%, 5.4%, 8.9%, and 10.3% for AHK, $GHKR$, AM, $CTFD$, and FI respectively. In 42% of all cases the $CTSP-$Heuristic found a solution superior to the other approaches, compared to the 20% for runner up AHK. Moreover, whenever the $CTSP-$Heuristic did not yield the best result of the 6 heuristics, it was on average within 1.4% of the best heuristic, compared to 2.4% for AHK. Finally, its worst performance never exceeded the best solution by 3.5%, compared to 6.2% for AHK. In summary, the $CTSP-$Heuristic tends to solidly outperform existing heuristics across a wide spectrum of problem instances across several performance measures.

9 Conclusion

In this paper, we introduced a new class of TSP-based' problems that arise in a variety of practical settings. Our initial interest was raised in the context of semiconductor manufacturing, but applications include chemistry, biology, scheduling, transportation, and other problems. In particular, we discussed the generalized version of the TSP, in which nodes need to be "colored" and all nodes of a color have to be visited consecutively without interruption. We distinguished two principal cases. In the more general case ($FCTSP$), each node must be assigned a color, whereas in the more restricted problem ($CTSP$), the color assignment is given. We provided a general formulation for both problems, which lends itself to deriving strong lower bounds based on Lagrangian relaxation. We established that, in contrast to the metric TSP, the $FCTSP$ is $APX-$hard. However, for the $CTSP$ we could only trivially establish $NP-$hardness, but were neither able to provide a polynomial time approximation, nor could we establish it to be $APX-$hard as well. We expect establishing a more definite result for the $CTSP$ to be quite a formidable task.

Finally, we proposed and evaluated heuristics to solve both problems. We provided analytical upper bounds and evaluated the heuristics and lower bounds empirically. Overall, bounds and heuristics perform very well and are usually well within 5% of each other, and within 3% of the optimal solution. Our results also indicate that the procedures are robust to scaling of the problem instances.

9.1 Appendix I - Determining a Connecting Edge Set (CES) of Minimum Total Weight

Algorithm 17 (CES) *Step 1: For each interval (v_{2i}, v_{2i+1}) determine if any edges traverse it. If not, interval (v_{2i}, v_{2i+1}) is said to be a "gap". The $m \geq 0$ gaps found in this manner give rise to $m+1$ "regions" of the graph.*

P	K	N	H(2)/L(2)	AHK/L(2)	GHKR/L(2)	AM/L(2)	CTFD/L(2)	FI/L(2)
1	20	100	1.044	1.028	1.048	1.049	1.123	1.083
2		200	1.015	1.053	1.023	1.095	1.128	1.123
3		300	1.021	1.038	1.089	1.065	1.092	1.119
4		400	1.024	1.053	1.042	1.076	1.120	1.093
5		500	1.025	1.027	1.042	1.077	1.060	1.073
6		600	1.034	1.062	1.074	1.025	1.107	1.078
7		700	1.056	1.024	1.036	1.085	1.096	1.117
8		800	1.041	1.054	1.087	1.064	1.072	1.151
9		900	1.026	1.024	1.021	1.031	1.157	1.180
10		1000	1.050	1.042	1.033	1.032	1.147	1.163
11	40	100	1.045	1.054	1.097	1.034	1.102	1.073
12		200	1.014	1.041	1.035	1.065	1.136	1.166
13		300	1.051	1.038	1.047	1.049	1.089	1.108
14		400	1.071	1.035	1.068	1.042	1.056	1.098
15		500	1.032	1.021	1.037	1.042	1.106	1.178
16		600	1.030	1.028	1.039	1.052	1.082	1.037
17		700	1.055	1.073	1.036	1.048	1.091	1.170
18		800	1.045	1.028	1.037	1.042	1.060	1.020
19		900	1.015	1.038	1.074	1.042	1.075	1.128
20		1000	1.030	1.072	1.086	1.034	1.097	1.090
21	60	100	1.017	1.019	1.041	1.026	1.055	1.053
22		200	1.017	1.041	1.052	1.079	1.113	1.118
23		300	1.039	1.083	1.037	1.069	1.091	1.146
24		400	1.005	1.050	1.041	1.061	1.052	1.057
25		500	1.036	1.085	1.063	1.054	1.029	1.089
26		600	1.048	1.034	1.066	1.055	1.020	1.074
27		700	1.038	1.025	1.015	1.058	1.042	1.119
28		800	1.038	1.081	1.056	1.049	1.018	1.138
29		900	1.042	1.068	1.046	1.026	1.116	1.025
30		1000	1.039	1.060	1.040	1.079	1.122	1.076
31	80	100	1.058	1.054	1.065	1.074	1.094	1.082
32		200	1.023	1.054	1.085	1.068	1.059	1.067
33		300	1.039	1.061	1.030	1.033	1.120	1.115
34		400	1.021	1.049	1.073	1.071	1.036	1.052
35		500	1.054	1.042	1.049	1.078	1.047	1.076
36		600	1.031	1.074	1.037	1.024	1.128	1.158
37		700	1.031	1.006	1.046	1.050	1.115	1.072
38		800	1.062	1.082	1.079	1.058	1.057	1.129
39		900	1.035	1.035	1.041	1.037	1.049	1.111
40		1000	1.031	1.052	1.041	1.058	1.075	1.082
41	100	100	1.010	1.065	1.083	1.020	1.120	1.142
42		200	1.031	1.032	1.034	1.037	1.088	1.160
43		300	1.037	1.055	1.069	1.040	1.091	1.121
44		400	1.048	1.054	1.059	1.088	1.075	1.116
45		500	1.020	1.043	1.076	1.013	1.066	1.140
46		600	1.058	1.040	1.040	1.033	1.097	1.185
47		700	1.028	1.044	1.075	1.080	1.082	1.117
48		800	1.042	1.052	1.083	1.023	1.084	1.077
49		900	1.053	1.053	1.049	1.074	1.094	1.141
50		1000	1.039	1.050	1.058	1.036	1.119	1.125
Average			**1.033**	**1.047**	**1.059**	**1.054**	**1.089**	**1.103**
% Best Performance			**42%**	**20%**	**12%**	**18%**	**8%**	**0%**
% Outperformed by H(2)				**68%**	**74%**	**70%**	**88%**	**96%**
Average Ratio to Best			**1.014**	**1.024**	**1.029**	**1.029**	**1.063**	**1.079**
Worst Ratio to Best			**1.035**	**1.062**	**1.073**	**1.079**	**1.133**	**1.156**

Figure 12: The Performance of the $CTSP$-Heuristic Compared to Existing Heuristics

Step2: For each region with more than one component, find the shortest edge connecting each component to another component. Add these edges to the edge set of the region until the region has only one single component.

Step 3. Add edges across all gaps.

Proposition 18 *The edges added in Steps 2 and 3 of the $CES-$Algorithm connect the independent components from the $L-CTSP$ algorithm with edges of minimum total weight in $O(K \log K)$ time.*

Proof. Notice first that we only need to consider adding edges between adjacent nodes (because any other edge could be replaced by a series of such edges of equal length). As any CES must connect the regions, it follows directly that all edges added in Step 3 must be in the CES. Now suppose an edge added in Step 2 is not in the CES. Then, there is either more than one component, contradicting that the edges form an CES, or another edge in the CES warrants that the (formerly disconnected) components are still connected. As, by the choice of the edge in Step 2, this edge cannot be shorter than the original, we can replace it with the original.

Steps 1 and 3 are easily seen to run in $O(K)$ time. To determine the running time of Step 2, notice that it is sufficient to determine, for each of the $2K$ nodes, if it is the nearest neighbor to a node from a different component. As each component is joint to at least one other component, the number of components is reduced by at least one half in each step, such that there are at most $O(\log K)$ loops with running time $O(K)$ each. ■

10 Appendix II - Determining a Shortest Path

Algorithm 19 (Shortest Path) *Step 0: The current node is source S.*

Step 1: From the current node, change to the color that allows to move as far as possible towards sink T without a further color change (ties may be broken arbitrarily). Make that node the current node and repeat until level N of the network is reached.

The algorithm has a greedy structure, that starting with node S, sweeps forward on the longest continuous subpath of one color before switching to the next longest continuous subpath.

Proposition 20 *The Shortest Path algorithm finds the optimal solution for graph G_C in Algorithm 12.*

Proof. Denote the nodes on the heuristic solution by $v_{i,k'}$ and those of the optimal solution by $v_{i,k''}$. Suppose the heuristic solution is not optimal, then there must be a smallest i such that $v_{i,k'} \neq v_{i,k''}$. Let the next color change in the optimal solution occur after node $v_{j,k''}$. By choice of k'' in the heuristic, replacing nodes $v_{i,k''}, v_{i+1,k''}, ..., v_{j,k''}$ with nodes $v_{i,k'}, v_{i+1,k'}, ..., v_{j,k'}$ cannot increase the value of the optimal solution and the heuristic solution must also be optimal. ■

References

[1] Ahmadi,R.H. and J.W. Mamer. "Routing Heuristics for Automated Pick and Place Machines", European Journal of Operational Research, 117(3), 533-552, 1999.

[2] Alimonti, O. and V. Kann, "Hardness of approximating problems on cubic graphs", in "Lecture Notes in Computer Science", Springer, Berlin, Volume 1203, 1997.

[3] Arkin, E. M. and R. Hassin, "Approximation algorithms for the geometric covering salesman problem", Discrete Applied Mathematics, 55, 197-218, 1994.

[4] Arora, S. and C. Lund, "Hardness of Approximations", in "Approximation Algorithms for NP-Hard Problems" by D. Hochbaum, ITP, Boston, 1997.

[5] Arora, S., "Polynomial time approximation schemes for (Euclidean) traveling salesman and other geometric problems", Journal of the ACM, 45(5), 753–782, 1998.

[6] Bartholdi, J. and L.K. Platzman, "Heuristics Based on Spacefilling Curves for Combinatorial Problems in Euclidean Space", Management Science, 34, 3, 291-305, 1988.

[7] Chen, D.-S., R.G. Batson, and Y. Dang, "Applied Integer Programming", Wiley, Hoboken, NJ, 2010.

[8] Chisman, J., "The Clustered Traveling Salesman Problem", Computers and Operations Research 2, 115-119, 1975.

[9] Christofides, N. , "Worst-case analysis of a new heuristic for the travelling salesman problem", Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University,Pittsburgh, PA, 1976.

[10] Dumitrescu, A., and J.S.B. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane", Journal of Algorithms, 48, 135-159, 2003.

[11] Arkin, E.M., R. Hassin, and L. Klein, " Restricted Delivery Problems on a Network", Networks 29, 4, 205-216, 1997.

[12] Fisher, M.L., 1981, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," Management Science, 27, 1, 1-18.

[13] Garey, M.R. and D.S. Johnson, "Computers and intractability - A guide to the theory of NP-Completeness", Freeman, New York, NY, 1979.

[14] Geoffrion, A.M., "Lagrangian Relaxation for integer programming," Mathematical Programming Study, 2, 82-114, 1974.

[15] N. Guttmann-Beck, R. Hassin, and B. Raghavachan, "Approximation Algorithms with Bounded Performance Guarantees for the Clustered Traveling Salesman Problem," Algorithmica, 28, 422-437, 2000.

[16] Hoogeveen, J.A., "Analysis of Christofides' heuristic: Some paths are more difficult than cycles", Operations Research Letters, 10, 291-295, 1991.

[17] Johnson, D.S. and C.H. Papadimitriou, "Performance guarantees for heuristics", in "The Traveling Salesman Problem", by E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds.), John Wiley, New York, NY, 1986.

[18] Karp, R.M. and J.M. Steele, "Probabilistic Analysis of Heuristics" in "The Traveling Salesman Problem" by E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.), John Wiley, New York, NY, 1986.

[19] Kollias, J.G., Y. Manolopoulos, and C.H. Papadimitriou, "The Optimum Execution of Queries in Linear Storage", Information Processing Letters 36, 141-145, 1990.

[20] Kuhn, H.W., "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 2, 83-97, 1955.

[21] Laporte, G., A. Asef-Vaziri, and C. Sriskandarajah, "Some applications of the generalized travelling salesman problem", Journal of the Operational Research Society, 47, 1461-1467, 1996.

[22] Laporte G. and U. Palekar, "Some applications of the clustered traveling salesman problem", Journal of the Operational Research Society, 53, 972-976, 2002.

[23] Lawler, E.L., "Combinatorial Optimization: Networks and Matroids", Holt, Rinehart and Winston, New York, 1976.

[24] Papadimitriou C. and M. Yannakakis." Optimization, approximation, and complexity classes", Journal of Computer and System Sciences, 43, 425–440, 1991.

[25] Resnick, S.I., "A Probability Path," Birkhäuser, Boston, 2005.

[26] Van Den Berg, J.P., "Multiple Order Pick Sequencing in a Carousel System: A Solvable Case of the Rural Postman Problem", The Journal of the Operational Research Society, 47,12, 1504-1515,1996.

[27] Weintraub, A., J. Aboud, C. Fernandez, G. Laporte, and E. Ramirez, "An emergency vehicle dispatching system for an electric utility in Chile", Journal of the Operational Research Society 50, 690–696, 1999.