AN EXACT ALGORITHM FOR THE CLOSE ENOUGH TRAVELING SALESMAN PROBLEM WITH ARC COVERING CONSTRAINTS

Minh Hoang Ha[1,3], Nathalie Bostel[2], André Langevin[1] and Louis-Martin Rousseau[1]

[1]*Department of Mathematics and Industrial Engineering and CIRRELT, École Polytechnique de Montréal, C.P. 6079, Succursale Centre-ville, Montréal, Qué., H3C 3A7, Canada*

[2]*IUT de Saint-Nazaire, IRCCyN, 58 rue Michel Ange, B.P. 420, 44606 Saint-Nazaire Cedex, France*

[3]*École des Mines de Nantes, IRCCyN, 4 rue Alfred Kastler, 44307 Nantes Cedex 3, France*

Keywords: Close enough traveling salesman problem, Automated meter reading, Radio frequency identification, Cutting planes.

Abstract: In this paper, we consider a problem still seldom studied in the literature, the Close Enough Traveling Salesman problem with covering constraints on the arcs. This problem arises in the context of utility companies that use automated meter reading (AMR) with radio frequency identification (RFID) technology to read meters from a distance. The contribution of this paper is to introduce a new mathematical formulation and to propose a first exact algorithm for this problem. Computational results show that our algorithm is capable of solving to optimality instances of realistic size, such as those introduced in (Golden et al., 2008), with 1000 arcs and 9000 customers in less than 2 hours.

1 INTRODUCTION

Recent technological advances in the combined usage of automated meter reading and radio frequency identification are allowing an increasing number of utility companies to retrieve customer information remotely. With this technology, the exact consumption of a resource (electricity, gas or water) by customers is coded in a unique radio frequency signal and transmitted from metering devices, situated at the customer location, to portable receivers, which can be mounted in moving vehicles. The effective radius of automated meter reading (AMR), also called read range, is normally between 150 and 300 meters, but may be as high as 381 meters (see (Golden et al., 2008)). Although AMR is based on various technologies, we consider the most common version (radio frequency based) to study a variant of an arc routing problem dealing with mobile or "drive-by" meter reading. In this variant, a reading device is installed in a vehicle, which drives around to collect remotely the data sent by metering devices. Thus, the reader does not have to reach each customer to collect the data, but only needs to pass within its read range. Each vehicle must traverse a service area and pass close enough to each meter so that they can all be read.

More formally, let $G = (V, A)$ be a directed graph, where V is the vertex set, $V = \{v_0, ..., v_{n-1}\}$, vertex v_0 is the depot, and $A = \{(v_i, v_j) : v_i, v_j \in V\}$ is the arc set of size m. Numbering the arcs enables A to be expressed as $\{a_1, a_2, ..., a_m\}$ and a cost c_a to be associated with each arc of A. Let $W = \{w_1, w_2, ..., w_l\}$ be a set of customers that must be covered. These customers can be located everywhere on the area covered by the network, and have a different vertex. The *Close Enough Traveling Salesman Problem with arc covering constraints* (CETSP) consists in finding a minimum cost tour, which begins and ends at the depot, such that every customer of W is covered by the tour, i.e. lies within a distance r from an *arc* of the tour. The CETSP is NP-hard as it reduces to a *Rural Postman Problem* (RPP) when $r = 0$, i.e. every customer of W coincides with a point on an arc of A, and the number of arcs containing the customers (called required arcs) is less than m, the number of arcs in the graph. Note that, if the number of required arcs is equal to m, the problem becomes a *Chinese Postman Problem* (CPP).

To the best of our knowledge, the only work on the CETSP can be credited to (Golden et al., 2008). The authors call the problem a *CETSP over a street network* and propose four heuristics to solve 18 instances with an average of about 900 arcs and 9000 customers each. Among these instances, only two of them are described in more detail: a sparse instance that includes 3345 customers and 405 arcs and a dense instance with 10,230 customers and 1099 arcs. Basically, the heuristics of (Golden et al., 2008) are imple-

Hoang Ha M., Bostel N., Langevin A. and Rousseau L..
AN EXACT ALGORITHM FOR THE CLOSE ENOUGH TRAVELING SALESMAN PROBLEM WITH ARC COVERING CONSTRAINTS.
DOI: 10.5220/0003698702330238
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 233-238
ISBN: 978-989-8425-97-3

mented through a two-stage process. In stage one, the heuristic identifies a subset of arcs to be traversed, either with some simple greedy procedures or through solving an integer program. In stage two, the problem becomes the well-known *Rural Postman Problem* (RPP) and is solved by a sophisticated heuristic.

There is another problem which can be seen as a *CETSP with the vertex covering constraints*. It is named *the covering tour problem* by (Gendreau et al., 1997) and is similar to the CETSP with arc covering constraints except that a closed tour has to be determined so that every vertex of W lies within a distance r from a *vertex* of the tour. In (Gendreau et al., 1997), an exact algorithm and a heuristic are presented. Heuristics have also been proposed for problems belonging to the family of the covering tour problem, such as *the CETSP in the plane* in (Gulczynski et al., 2006) and (Dong et al., 2007).

In this paper, we address the CETSP, which we formulate as an integer program and solve through a cutting plane approach. Computational experiments show that our approach is efficient, being capable of solving instances in some cases with up to 1500 arcs in less than 2 hours.

The remainder of the paper is organized as follows: Section 2 presents in detail our exact algorithm to solve the CETSP. Computational results are reported and analysed in Section 3. Finally, Section 4 summarizes our conclusions.

2 EXACT ALGORITHM FOR CETSP

Given a node subset, $S \subseteq V$, let $\delta^+(S)$ denote the set of outgoing arcs of S and $\delta^-(S)$ denote the set of incoming arcs of S. If $S = \{v_k\}$, we simply write $\delta^+(k)$ (or $\delta^-(k)$) instead of $\delta^+(\{v_k\})$ (or $\delta^-(\{v_k\})$). $E(S)$ is the set of arcs with both end-points in S. Let x_a be the number of times arc a is traversed, and c_a the associated cost (distance or travel time). We define the binary coefficients λ_{lk} equal to 1 if and only if $w_l \in W$ can be covered by $a_k \in A$. Given $x \in N^{|A|}$ and $T \subset A$, $x(T)$ denotes $\sum_{e \in T} x_e$. Then the CETSP can be stated as:

$$\text{Minimize} \sum_{a \in A} c_a x_a \qquad (1)$$

$$\text{subject to } x\Big(\delta^+(0)\Big) \geq 1 \qquad (2)$$

$$x\Big(\delta^+(i)\Big) - x\Big(\delta^-(i)\Big) = 0 \ \forall i \in V \qquad (3)$$

$$\sum_{a \in A} x_a.\lambda_{wa} \geq 1 \ \forall w \in W \qquad (4)$$

$$Mx\Big(\delta^+(S)\Big) - x\Big(E(S)\Big) \geq 0 \ \forall S \subset V - \{v_0\} \text{ and } 2 \leq |S| \leq n-2 \qquad (5)$$

$$x_a \in Z^+ \ \forall a \in A \qquad (6)$$

where M is a large number. The objective (1) is to minimize the total travel cost. Constraint (2) ensures that the depot belongs to the tour, while (3) are the flow conservation constraints. Constraints (4) enforce that every customer of W is covered by the tour and constraints (5) are the disjoint subtour elimination constraints. (In arc routing, an optimal tour can contain a cycle. The disjoint "subtour elimination constraints" eliminate subtours disconnected from the tour containing the depot). These constraints force the presence of at least one outgoing arc of any set S, for every possible subset S of V containing an arc belonging to the tour. Constraints (6) define the variable domains.

We propose to solve the CETSP optimally through a cutting plane approach. The main strategy of this algorithm is to solve iteratively an integer program including the constraints (2), (3), (4), and (6). At each iteration, the disjoint subtour elimination constraints (5) violated by the optimal solution are added to the model. A disjoint subtour can be identified through a depth-first search which, starting from a given vertex, traverses all the solution arcs to reach all other vertices. The encountered vertices marked are placed on a stack in the order in which they are visited. After an arc is traversed, it is removed from the current graph. A disjoint subtour is created when, starting from some vertex, it is not possible to mark all the vertices present in the solution. Once a disjoint subtour is identified, the process can be repeated starting from any unmarked vertex of the solution.

Note that the constraints (5) can only be applied to the disjoint subtours that do not contain the depot. This is because, for the subtours S_0 containing the depot, the disjoint subtour elimination can be obtained by the use of arcs not already used in the solution but with head points belonging to S_0, the set of nodes of the subtour containing the depot. In order to improve the performance of the algorithm, we first check the covering of such subtours. If we can not cover all the customers with the arcs of S_0 (used and unused), the constraint (5) is then still applied; otherwise, we add the following constraint for S_0:

$$Mx\Big(\delta^*(S_0)\Big) - x\Big(E'(S_0)\Big) \geq 0 \qquad (7)$$

where $E'(S_0)$ denotes the set of arcs used in S_0 while $\delta^*(S_0)$ denotes the set of arcs that are not used and whose heads are in S_0.

3 COMPUTATIONAL EXPERIMENTS

In this section, we describe the CETSP instances and the computational evaluation of the proposed approach. Our algorithm is coded in C/C++ using Cplex 12.1 with Callable Library and was run on a 2.4 GHz CPU with 2GB RAM. The running time for each instance was limited to 2 hours. We have tested different values of M (5000, 10,000 and 20,000) and observed that its impact on the performance of the algorithm is negligible. We decided to predetermine M at a value of 10,000 in the implementation.

3.1 Data Instances

To build the CETSP instances, we randomly generate graphs that imitate real street networks by the following procedure:

- The coordinates of n vertices are randomly generated in a unitary square. Then a heuristic is used to find the shortest tour passing through all the nodes exactly once. This tour is a Hamiltonian cycle and is used as a framework to construct the full graph. The resulting graph is therefore strongly connected.
- In order to imitate real networks, random arcs are added to the current tour to reach a total number of arcs $m = nd$, where n denotes the number of vertices and d the ratio between the number of arcs and the number of nodes, in such a way that: (i) the arcs are not too long, and (ii) there is no intersection between any two arcs.

In our tests, we use the graphs with the number of vertices $n \in \{300, 400, 500\}$ and the ratio between the number of arcs and the number of nodes $d \in \{1.5, 2, 2.5, 3\}$. For each couple of n and d, we have generated 10 different graphs. Arc costs are defined as c_{ij} kilometres, where c_{ij} is the Euclidean distance between v_i and v_j multiplied by 5 to obtain an average length of arcs close to reality (from about 0.2 to 0.4 kilometres).

Once the graphs are created, the CETSP instances are generated by randomly positioning $q = mt$ customer nodes on the square containing the graph, where m denotes the number of arcs and t the ratio between the number of customers and the number of arcs, $t \in \{0.5, 1, 5, 10\}$. Thus, for each graph, four CETSP instances are created. The effective RFID radius r is set at a value of 150 meters. In order to ensure the existence of a solution, we delete all the customers that can not be covered by any arc. We also examine the impact of increasing the radius parameter from 150 meters to 200 meters. To do this, we still use the graph created with r = 150 meters but change the read range to 200 meters. In other words, the arcs and coordinates of the customers and vertices are kept constant while r is increased.

For each value of r, we thus generate 480 CETSP instances named ce-n-k-t, where n is the number of nodes, k indicates the number of arcs and t is the ratio between the number of customers and the number of arcs. For example, ce-300-450-10 stands for an instance with 300 nodes, 450 arcs, and t =10.

In order to analyse further the impact of customer number on our algorithm, we use a customer reduction procedure. Given $w_l \in W$, let $Z(w_l)$ be the set of arcs that can cover w_l. Consider each pair of customers w_i and w_j, if $Z(w_i) \subseteq Z(w_j)$ then customer w_j can be eliminated. This is because when we service w_i, w_j is covered at the same time. Note that, the number of remaining customers is also the maximum number of arcs that have to be activated for covering purpose.

3.2 Computational Results

Tables 1 and 2 present the characteristics of the instances and the computational results obtained for two different values of r (radius of AMR). They include the name of the instance and the average number of remaining customers after the reduction procedure (in columns 1 and 2). $\#ofOpt$ indicates the number of optimal solutions obtained for each set and $OptVal$ the average optimal value (in km) for these solutions. $\#ofIPIter$ presents the average, minimum and maximum number of integer programming iterations. $Time$ shows the average, minimum and maximum running time (in seconds).

The results presented on Table 1 and 2 indicate that the ratio between the number of remaining customers and the number of arcs in graph is often between 0.2 and 0.5. Therefore, the number of arcs that must be activated for covering purpose is smaller than the total number of arcs and our instances are thus far from a CPP.

As can be observed in Table 1, for t = {5,10} our algorithm was able to solve all but one of the instances with 1250 arcs. It was also capable of solving some instances with up to 1500 arcs. But when t is smaller, the instances become more difficult. This counterintuitive behaviour can be explained as follows. When the number of customers decreases, the number of arcs that must be activated for covering purpose also decreases. Thus, there are more potential combinations to connect these ”covering” arcs. More MIP it-

Table 1: Computational results with r = 150 meters.

Data	# of Cus	# of Opt	Opt Val (km)	# of IP Iter Aver.	Min	Max	Time (sec) Aver.	Min	Max
ce300-450-0.5	102	10	94.5	6.4	2	24	0.52	0.05	2.5
ce300-450-1	142	10	118.8	2.8	1	8	0.08	0.02	0.31
ce300-450-5	202	10	175.6	1.6	1	3	0.08	0.05	0.16
ce300-450-10	212	10	195.4	1.6	1	3	0.15	0.09	0.36
ce300-600-0.5	126	10	100.6	11.5	2	31	3.98	0.17	14.61
ce300-600-1	166	10	123.5	8.5	2	33	3.01	0.14	17.78
ce300-600-5	224	10	182.1	4.9	2	27	1.07	0.16	8.88
ce300-600-10	236	10	195.6	2.1	1	4	0.27	0.16	0.45
ce300-750-0.5	147	9	95.6	36.2	5	112	308.52	1.98	1807.64
ce300-750-1	193	10	120.4	8.2	4	12	5.40	1.59	10.17
ce300-750-5	248	10	172.1	3.3	2	6	0.70	0.25	2.30
ce300-750-10	264	10	186.3	2.9	2	5	0.60	0.30	1.08
ce300-900-0.5	166	7	99.1	12.6	4	32	194.09	1.77	1034.83
ce300-900-1	213	10	122.0	10.9	2	26	45.03	0.56	258.08
ce300-900-5	276	10	163.5	3.9	2	9	3.24	0.56	11.52
ce300-900-10	292	10	175.0	3.1	1	5	2.18	0.61	3.78
ce400-600-0.5	144	10	100.3	11.2	2	49	1.94	0.05	12.45
ce400-600-1	194	10	129.4	3.3	2	7	0.16	0.06	0.41
ce400-600-5	260	10	185.1	1.8	1	3	0.13	0.08	0.22
ce400-600-10	274	10	207.1	1.5	1	3	0.19	0.14	0.30
ce400-800-0.5	174	9	102.1	35.7	4	88	88.25	1.76	356.92
ce400-800-1	226	10	126.5	17.2	3	42	11.49	0.75	33.45
ce400-800-5	296	10	178.8	3.8	2	12	1.09	0.22	7.03
ce400-800-10	309	10	199.0	2.6	2	5	0.52	0.30	1.64
ce400-1000-0.5	198	3	103.5	11.0	10	13	745.15	40.94	2113.89
ce400-1000-1	252	9	122.1	13.1	5	25	492.43	4.63	1992.3
ce400-1000-5	313	10	169.2	4.9	2	13	4.39	0.72	18.20
ce400-1000-10	327	10	183.5	4.2	2	9	4.94	0.59	24.59
ce400-1200-0.5	221	0							
ce400-1200-1	279	4	123.0	8.0	4	12	73.65	40.03	135.44
ce400-1200-5	357	10	157.1	5.7	4	9	29.07	6.43	129.79
ce400-1200-10	373	10	167.0	6.9	3	23	26.83	3.21	91.48
ce500-750-0.5	185	10	113.5	39.0	3	323	674.82	0.13	6731.81
ce500-750-1	249	10	141.3	10.7	2	70	6.71	0.11	63.13
ce500-750-5	319	10	201.3	2.6	2	4	0.21	0.17	0.28
ce500-750-10	328	10	227.1	2.2	1	4	0.30	0.23	0.43
ce500-1000-0.5	224	6	126.2	58.0	8	211	1271.11	7.09	6161.19
ce500-1000-1	285	10	131.0	14.3	3	65	30.72	0.95	102.42
ce500-1000-5	360	10	181.8	4.4	3	9	1.80	0.57	7.25
ce500-1000-10	375	10	203.0	4.1	2	6	1.47	0.58	3.00
ce500-1250-0.5	262	0							
ce500-1250-1	330	4	122.5	13.8	5	30	1385.46	235.72	2496.89
ce500-1250-5	408	9	172.0	9.0	3	31	62.24	2.17	201.83
ce500-1250-10	428	10	182.2	5.5	3	13	38.53	2.48	124.22
ce500-1500-0.5	284	0							
ce500-1500-1	354	0							
ce500-1500-5	458	5	163.9	8.0	6	12	402.75	32.19	1008.89
ce500-1500-10	477	8	172.0	5.8	4	8	353.63	15.22	1994.36

Table 2: Computational results with r = 200 meters.

	# of	# of	Opt Val	# of IP Iter			Time (sec)		
Data	Cus	Opt	(km)	Aver.	Min	Max	Aver.	Min	Max
ce300-450-0.5	101	10	74.2	20.9	2	93	7.76	0.05	52.38
ce300-450-1	143	10	89.3	11.3	2	70	1.31	0.05	10.98
ce300-450-5	197	10	115.7	1.5	3	3	0.07	0.05	0.13
ce300-450-10	205	10	129.5	1.3	2	3	0.11	0.08	0.18
ce300-600-0.5	126	9	74.1	53.3	10	201	844.57	4.42	6959.88
ce300-600-1	168	10	83.5	23.2	4	65	16.74	0.72	60.38
ce300-600-5	217	10	115.1	6.4	2	25	1.91	0.13	8.88
ce300-600-10	220	10	125.7	5.6	1	24	1.72	0.22	11.94
ce300-750-0.5	149	4	71.4	29.3	13	61	2981.45	208.53	7088.2
ce300-750-1	192	7	87.5	29.0	8	109	1749.44	14.36	7198.27
ce300-750-5	238	10	113.9	11.2	5	37	31.05	2.41	145.95
ce300-750-10	240	10	122.9	7.0	4	12	7.83	1.88	22.86
ce300-900-0.5	169	1	81.4	87.0	87	87	4352.8	4352.8	4352.8
ce300-900-1	212	3	90.0	35.7	9	52	2895.48	127.5	7070.88
ce300-900-5	264	9	110.4	15.4	4	55	761.51	6.25	2542.66
ce300-900-10	269	10	117.2	8.1	3	25	148.26	2.80	1225.56
ce400-600-0.5	146	10	79.2	13.4	2	50	4.37	0.08	28.67
ce400-600-1	202	9	88.3	22.4	1	157	24.17	0.06	209.91
ce400-600-5	274	10	114.3	6.4	2	21	0.90	0.13	6.06
ce400-600-10	276	10	123.6	4.9	1	25	0.52	0.20	2.66
ce400-800-0.5	180	7	77.6	20.0	7	43	89.64	6.02	264.20
ce400-800-1	239	8	85.2	11.3	6	25	18.18	2.77	62.84
ce400-800-5	314	9	111.6	8.8	4	22	6.61	1.44	16.95
ce400-800-10	321	10	120.3	24.5	4	162	180.89	1.22	1723.02
ce400-1000-0.5	208	0							
ce400-1000-1	270	1	95.9	24.0	24	24	7171.13	7171.13	7171.13
ce400-1000-5	331	0							
ce400-1000-10	339	6	116.0	16.0	6	39	1874.72	18.91	7186.65
ce500-750-0.5	192	10	78.0	51.8	3	326	384.42	0.36	3534.50
ce500-750-1	263	10	96.3	28.1	3	181	54.15	0.25	505.33
ce500-750-5	363	10	118.6	5.0	2	24	1.44	0.19	11.27
ce500-750-10	377	10	128.8	3.0	2	4	0.54	0.34	0.78
ce500-1000-0.5	242	1	85.1	22.0	22	22	842.83	842.83	842.83
ce500-1000-1	321	5	90.8	19.8	9	52	436.34	13.56	814.22
ce500-1000-5	411	9	109.0	28.2	6	78	1025.59	6.23	5406.33
ce500-1000-10	429	8	117.8	17.3	5	28	483.14	2.81	3248.66

erations are thus needed to solve the problem.

We also observe that, for a given number of vertices, the greater the vertex degree is, the harder the instance is.

In Table 2, we see the results for the case where $r = 200$ meters. The performance of our algorithm degrades as it can only solve the instances with up to 1000 arcs. When the read range increases, each customer can be covered by more arcs, so the solution space increases and the optimal solution becomes more difficult to find. Obviously, increasing the read range leads to a considerable decrease in the traveling distance and cost. The column *OptVal* confirms this remark. Developing new technology to allow an increased read range could therefore be an effective means to reduce the distance driven to collect customer information.

4 CONCLUSIONS

In this paper, we have formulated and solved the CETSP with arc covering constraints. An integer linear programming formulation has been proposed and solved through a cutting plane algorithm. Computational results on a set of 960 instances have been reported and analysed. These results show that our algorithm is capable of solving to optimality instances of realistic size and works better when there are many customers to be covered. As we notice that RFID technology is rather used when the customer density is important, our algorithm is quite suitable to solve real problems of utility companies.

REFERENCES

Dong, J., Yang, N., and Chen, M. (2007). *Extending the Horizons: Advances in Computing, Optimisation, and Decision Technologies*, chapter Heuristic Approaches for a TSP Variant: The Automatic Meter Reading Shortest Tour Problem, pages 145–163. Springer Verlag.

Gendreau, M., Laporte, G., and Semet, F. (1997). The covering tour problem. *Operations Research*, 45:568–576.

Golden, B., Raghavan, S., and Wasil, E. (2008). *The Vehicle Routing Problem: Lastest Avances and New Challenges*, chapter Advances in Meter Reading: Heuristic Solution of the Close Enough Traveling Salesman Problem over a Street Network, pages 487–501. Springer.

Gulczynski, D., Heath, J., and Price, C. (2006). *Perspectives in Operations Research: Papers in Honor of Saul Gass 80th Birthday*, chapter The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics, pages 271–283. Springer Verlag.