A Novel Reduction Algorithm for the Generalized Traveling Salesman Problem

Mehdi El Krari
Computer Science Laboratory, Faculty of Science
Mohammed V University in Rabat
mehdi@elkrari.com

Belaïd Ahiod
LRIT URAC 29, Faculty of Science
Mohammed V University in Rabat
ahiod@fsr.ac.ma

Bouazza El Benani
Computer Science Laboratory, Faculty of Science
Mohammed V University in Rabat
elbenani@hotmail.com

ABSTRACT

The Generalized Traveling Salesman Problem (GTSP) is a Combinatorial Optimization Problem considered as a generalization of the well known Traveling Salesman Problem. The GTSP, which is an NP-Hard problem, consists of visiting only one city/node from each region/cluster from all given clusters of an instance. This paper introduces a new reduction method that removes "farthest" cities from other clusters and keeps the nearest ones. Experimental tests done on 81 instances from the TSPLIB benchmark present a reduction rate from 9 to 73%. The runtime is almost less than *one* second for most instances and doesn't exceed 7 for the larger ones.

CCS CONCEPTS

•Theory of computation → Problems, reductions and completeness; •Mathematics of computing → *Combinatorial optimization;* **•Computing methodologies →** Discrete space search;

KEYWORDS

Generalized Traveling Salesman Problem; Reduction Algorithm; Nearest Nodes

ACM Reference format:
Mehdi El Krari, Belaïd Ahiod, and Bouazza El Benani. 2017. A Novel Reduction Algorithm for the Generalized Traveling Salesman Problem. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076028

1 MOTIVATION

The Traveling Salesman Problem (TSP) is an NP-Hard problem [4] that has been approached several times in Combinatorial Optimization. The problem is popular for its severity and the diverse applications it may resolve. The main purpose of the problem is to find the shortest cyclic path passing by all the given N nodes. Srivastava et al. [10] and Henry-Labordere [5] proposed a generalization of this problem called the Generalized Traveling Salesman Problem (GTSP). n nodes are partitioned over m clusters $(C_1, C_2, ..., C_m)$ and only one node from each cluster C_i must be visited and make a cyclic tour by getting back to the starting node/cluster.

Several real-world applications can be modeled as a GTSP. Mail delivery [6], airplane routing [1], computer file sequencing [5], vehicle routing [7], post-box selection and welfare agency [9] routing are some examples of various combinatorial optimization problems that can be interpreted as well.

NP-Hard problems can be solved via exact methods, approximative algorithms or metaheuristics which are different strategies. The first one guarantees to reach the optimal solution while the two others give a fair solution in an acceptable time. Runtime of all of these algorithms depends on the instance size. The larger the instance is, the more significant the runtime becomes, until it becomes impossible to run the algorithm, especially for exact methods.

Finding a near-optimal solution for a large instance in a relatively short time is always a proof of the algorithm efficiency since it is harder to obtain a satisfying result on both sides. Some approaches can be reduced by removing some data following a strategy with a specific criterion [3] since a feasible solution can be made without these elements.

We propose in this work a new reduction method for the GTSP. which keeps the nodes that are close to each others and are from different clusters. Then removes all the other nodes, since they are considered far from the clusters and probably will not ensure good solutions. Obtaining a GTSP solution consists of finding one node from each cluster, the proposed approach will not impact the feasibility of any instance of the problem.

2 NEAREST NODES TO CLUSTERS

"Nearest Nodes to Clusters (NN2C)" is a new method we present in this work. The aim of this algorithm is to select from each cluster nodes that are close to other clusters then remove the other nodes considering them too much far. Indeed, selected nodes are the favorites ones for constructing the best solutions.

For each couple of clusters (C_i, C_j), NN2C picks a pair of nodes $(v_x, v_y) \in (C_i, C_j)$ that provide the smallest distance between these clusters. v_x and v_y are then added to the set S of selected nodes. This selection procedure is redone while there are still couples of clusters not yet browsed. Observe that a node can be picked more than once, but will of course appear one time in the set S and won't be replaced by another one. The detailed process of NN2C reduction method is described in Algorithm 1.

Figure 1 shows an example of selected and removed nodes for a small instance of four clusters. Green nodes are those (v_x, v_y) selected in each iteration. Once all cluster pairs are browsed, unselected vertices (in red) can be removed from the instance to be reduced.

GECCO '17 Companion, Berlin, Germany

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076028

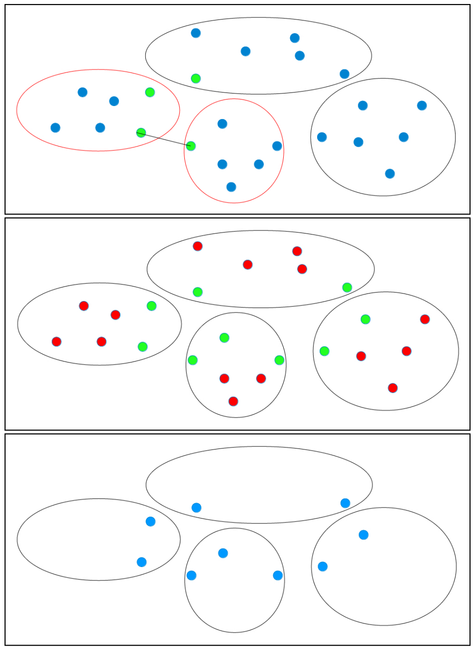

Figure 1: Running NN2C algorithm on an intance of 4 clusters: (1) selecting pair of nodes from each couple of clusters then (2) removing non-selected nodes and finally (3) return the new instance

Algorithm 1 **Nearest Nodes to Clusters (NN2C)**

Require: An instance **I** with a set of m clusters **C** and a set of n nodes **V** distributed over **C**

Ensure: **I** reduced with a set of m clusters **C** and a set of n' ($n' < n$) nodes **V'** $\subset$ **V**

for each couple (C_i, C_j) from C **do**
    Select node v_x from C_i and node v_y from C_j such that $c(v_x, v_y) \leq c(v_a, v_b) \forall (v_a \in C_i, v_b \in C_j)$
    if ($v_x \notin S$) **then**
        add v_x to V'
    end if
    if ($v_y \notin S$) **then**
        add v_y to V'
    end if
end for
replace V with V'
return I

3 EXPERIMENTAL RESULTS

The first test of the NN2C reduction algorithm was made on different instances delivered by the TSPLIB benchmark provided by Reinelt [8] and converted to a GTSP instance by the standard clustering procedure [2]. This test includes instances with various sizes (small, large and very large) and different coordinates type. Table 1 shows some of the 85 studied instances chosen randomly and including the five largest. We expose for each one the number of clusters and nodes, then the number of removed nodes by NN2C, reduction rate and finally the runtime in seconds. The average concerns all the studied instances, not only the listed ones.

lcccc

Table 1: NN2C detailed performances

Instance	Clusters	Nodes	Removed Nodes	Reduction Rate (%)	t(s)
3burma14	3	14	9	64,29	0
11berlin52	11	52	25	48,08	0
26bier127	26	127	53	41,73	0
26ch130	26	130	43	33,08	0
28pr136	28	136	40	29,41	0
28gr137	28	137	49	35,77	0,01
30kroA150	30	150	47	31,33	0,01
30kroB150	30	150	56	37,33	0,01
31pr152	31	152	58	38,16	0,01
40d198	40	198	67	33,84	0,01
40kroa200	40	200	67	33,5	0,01
41gr202	41	202	65	32,18	0,01
45ts225	45	225	105	46,67	0,02
46gr229	46	229	79	34,5	0,02
49usa1097	49	1097	800	72,93	0,08
280fl1400	280	1400	573	40,93	0,1
316fl1577	316	1577	333	21,12	0,11
331d1655	331	1655	262	15,83	0,12
350vm1748	350	1748	654	37,41	0,19
364u1817	364	1817	173	9,52	0,12
378rl1889	378	1889	415	21,97	0,14
421d2103	421	2103	205	9,75	0,13
431u2152	431	2152	213	9,9	0,14
608pcb3038	608	3038	539	17,74	0,25
759fl3795	759	3795	1008	26,56	0,29
893fnl4461	893	4461	814	18,25	0,43
2370rl11849	2370	11849	1761	14,86	2,98
2702usa13509	2702	13509	4263	31,56	2,47
2811brd14051	2811	14051	3014	21,45	2,86
3023d15112	3023	15112	3670	24,29	5,35
3703d18512	3703	18512	3851	20,8	7,04
		Average		**31,19**	**0,33**

NN2C succeeded to reduce instancesfi size with a rate of up to 73% and a runtime less than one second except for the five largest instances. Given the fast performance and the significant reduction rate, NN2C performances seem to be very interesting and deserve to be further investigated.

REFERENCES

[1] Noon C.E. 1988. *The generalized traveling salesman problem.* Ph.D. Dissertation. University of Michigan.

[2] Matteo Fischetti, Juan Jose Salazar Gonzalez, and Paolo Toth. 1997. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research* 45, 3 (1997), 378–394.

[3] Imola K Fodor. 2002. A survey of dimension reduction techniques. *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory* 9 (2002), 1–18.

[4] M. R. Garey, D. S. Johnson, and L. Stockmeyer. 1974. Some Simplified NP-complete Problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing (STOC '74).* ACM, New York, NY, USA, 47–63. DOI: http://dx.doi.org/10.1145/800119.803884

[5] A.L. Henry-Labordere. 1969. The record balancing problem: A dynamic programming solution of a generalized travelling salesman problem. *RIRO* B-2 (1969), 43–49.

[6] Gilbert Laporte, Ardavan Asef-Vaziri, and Chelliah Sriskandarajah. 1996. Some applications of the generalized travelling salesman problem. *Journal of the Operational Research Society* 47, 12 (1996), 1461–1467.

[7] Gilbert Laporte and Frédéric Semet. 1999. Computational evaluation of a transformation procedure for the symmetric generalized traveling salesman problem. *INFOR: Information Systems and Operational Research* 37, 2 (1999), 114–120.

[8] Gerhard Reinelt. 1991. TSPLIB—A Traveling Salesman Problem Library. *ORSA Journal on Computing* 3, 4 (nov 1991), 376–384. DOI: http://dx.doi.org/10.1287/ijoc.3.4.376

[9] J_ P Saksena. 1967. *Mathematical Model of Scheduling Clients Through Welfare Angencies: II.* Depts. of Electrical Engineering and Medicine, University of Southern California.

[10] SS Srivastava, S Kumar, R.C Garg, and P Sen. 1969. Generalized travelling salesman problem through n sets of nodes. *CORS J.* 7 (1969), 97–101.