A network flow procedure for solving the multi-traveling salesman problem

2 authors:

Alberto Garcia-Diaz
University of Tennessee

Abdulaziz Alkabaa
King Abdulaziz University

Proceedings of the 2014 Industrial and Systems Engineering Research Conference
Y. Guan and H. Liao, eds.

A Network Flow Procedure for Solving the Multi-Traveling Salesman Problem

Alberto Garcia-Diaz
Abdulaziz Saud Alkabaa
Department of Industrial and Systems Engineering
The University of Tennessee
Knoxville, TN 37996-2315, USA

Abstract

In the multi-traveling salesman problem each city is visited exactly once by exactly one salesman. In addition to travel costs between cities there is a cost for each salesman. The solution is a set of cities forming a sub-tour for each salesman, such that each sub-tour starts and ends at a city referred to as the *base city*, and the resulting total cost is minimized. In the proposed procedure the original network of cities is transformed into a second network. In this network each city is represented by two directly connected nodes (referred to as the beginning and end nodes) forming a *branch*. Also each trip from one city to another is represented by a directed arc by connecting the end node of the first city to the beginning node of the second city. This network can be further modified into a *circulation network* by adding a *return arc* connecting the *beginning* node of the base city to its *end* node. The solution procedure consists of three steps: (a) representing the original network as a *circulation network*; (b) finding minimum-cost flows in the circulation network; (c) braking infeasible sub-tours by removing arcs and rerouting flows. Computational requirements are discussed.

Keywords
Traveling salesman problem, multi-traveling salesman problem, circulation network, sub-tours elimination.

1. Introduction
The Traveling Salesman Problem (TSP) and the Multi-Traveling Salesman Problem (MTSP) are combinatorial in nature. Their extremely simple description has attracted a significant number of analysts encouraged by the wide range of potential applications. However, through the years these problems have remained virtually unsolved in the case of large-scale applications, leaving most researchers in the field with the belief that it is indeed a very difficult problem to solve, as perhaps best summarized by the quotation *"Why, it's as hard as the traveling salesman problem!"* [1, 2].

The traveling salesman problem can be stated as follows. There are *n* cities with known distances for trips between these cities. A salesman, starting at a given city, referred to as the *base* city, intends to visit each of the other *n-1* cities exactly once and return to the starting city. It is desired to determine the *order* in which the cities must be visited to minimize the total distance traveled. Instead of distances, any other measure of effectiveness that is additive along circuits of the city network can be used, such as costs and times. The structure of the problem shows that there are (*n-1*)*! Tours* to be considered in order to select one or more that should be optimal.

The multi-traveling salesman model allows any city to be visited by exactly one salesman, given that there are multiple salesmen visiting the cities. In this case, in addition to the travel cost associated with the trips between cities, there is a fixed cost for each salesman. The solution corresponds to a set of sub-tours, each starting and ending at the *base* city, resulting in minimal total distance, time or cost. Each sub-tour is assigned to exactly one salesman.

If some of the trips are not allowed, the corresponding arc *lengths* are set equal to infinity. In many cases one can assume that the distance between two cities is the same, regardless of the direction of the trip. However, in the algorithms that will be presented and discussed in this chapter this feature is not a requirement.

2. Proposed Methodology

Some of the theoretical background related to this specific approach has been previously addressed by Garcia-Diaz [3]. There are three cases of the problem that are of particular interest:

- **Case 1.** *M = 1*; this is the well-known Traveling Salesman Problem [4, 5, 6], where it is desired to identify the order in which the cities should be visited to minimize total travel cost.

- **Case 2.** *M >1* and all salesmen are used; in this case, it is desired to identify *M* tours, each one including the base city, such that the combined travel and fixed costs are minimized.

- **Case 3.** $M \geq 1$ and at most *M* salesmen are used; in this case, the number of tours minimizing the travel cost plus the fixed charges is to be determined, in addition to the cities contained in each tour. It is noted that for travel cost data satisfying the triangular inequality the optimal solution consists of exactly one tour; hence, Case 3 would reduce to Case 1.

An optimal solution to the multi-traveling salesman problem can be obtained by transforming it into a single-salesman problem, following the procedure briefly illustrated in Section 3 [4], and solving it as a TSP model. Also genetic algorithms have been used to solve this problem [7].

Alternatively, in cases where an optimal solution is computationally challenging or impossible due to the large size of the problem, a heuristic methodology can be used. This article proposes one of such procedures based on a *circulation-network* representation and a labeling procedure based on the Out-of-Kilter algorithm [8]. A circulation network is one where all nodes are *intermediate* nodes, i.e. their *net flows* (flow out minus flow in) are equal to zero. An overview of other approaches to the problem being considered are summarized in survey papers such as the one by Bektas [9].

2.1 A Heuristic Procedure

The objective of this section is to describe a *heuristic* circulation-network algorithm to solve the Multi-Traveling Salesman Problem without transforming it into an equivalent single TSP. When the number of salesmen is equal to 1 this heuristic procedure can be used to solve the TSP, as well. The proposed heuristic procedure consists of three steps.

Step 1. Representation of the MTSP as A Capacitated Circulation Network.

The original network of cities is transformed into a circulation network as follows. First, each city *i* is represented by two directly connected nodes i_a (referred to as the beginning node) and i_b (referred to as the end node) forming a *branch* (i_a, j_b) that must be traversed by one unit of flow by setting the lower and upper bound equal to 1. Furthermore, each trip from city i to city j is represented by a directed arc (i_b, j_a) by connecting the end node of the first city to the beginning node of the second city. The cost or length of this arc is equal to c_{ij} Figure 1 shows the basic transformation of nodes and arcs of the original network into arcs of the circulation network. In this figure, each arc of the circulation network has a triplet of values corresponding to the upper bound, the lower bound, and the per-unit cost of flow on the arc.

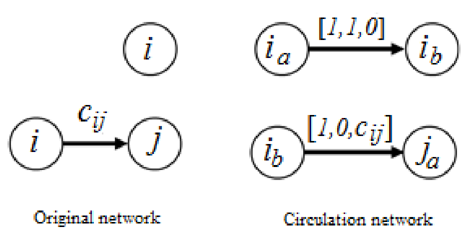


Figure 1: Transformation procedures from original network to capacitated

The network is further modified into a *circulation network* by adding a *return arc* $(1_a, 1_b)$ connecting the *beginning* node of the base city (assumed to be city 1) to its *end* node. The circulation network will be referred to as G'(**N',A'**). As an illustration, we will consider the 6-city network shown in Figure 2. Some trips in this figure are one-way and others are two-way trips. The corresponding transformed circulation network for Cases 1 and 2 is shown in Figure 3 and for Case 3 in Figure 4. In this illustration we assumed that there are only 2 salesmen. In Figure 3 all salesmen are used, while in Figure 4 the number of available salesmen becomes an upper bound on the actual number of salesman used.

In Figure 3 all salesmen must be used. The network in Figure 4 has four additional arcs to control the number of salesmen used. Specifically, arc $(1_b,1)$ has a lower bound equal to zero, an upper bound equal to one, and a per-unit cost equal to F_1, which is the cost of the first salesman. Furthermore, if the flow on this arc is equal to one then the first salesman will be used. A similar analysis can be done for the second salesman, which has a cost $F_2 \geq F_1$. If a unit of flow arrives at node D through arc $(1,D)$ which has a triplet of values equal to [1,0,0], it will be routed as described in Figure 3 for Case 1 or 2.

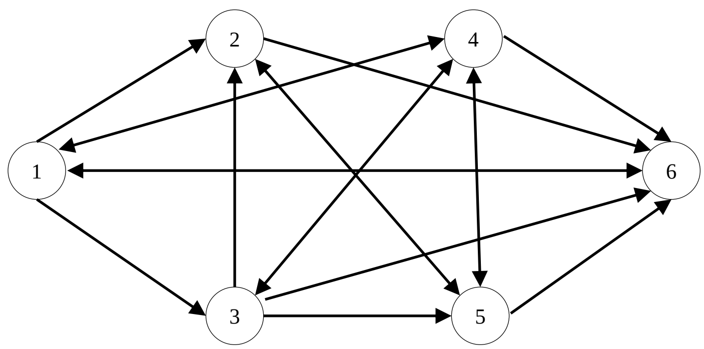
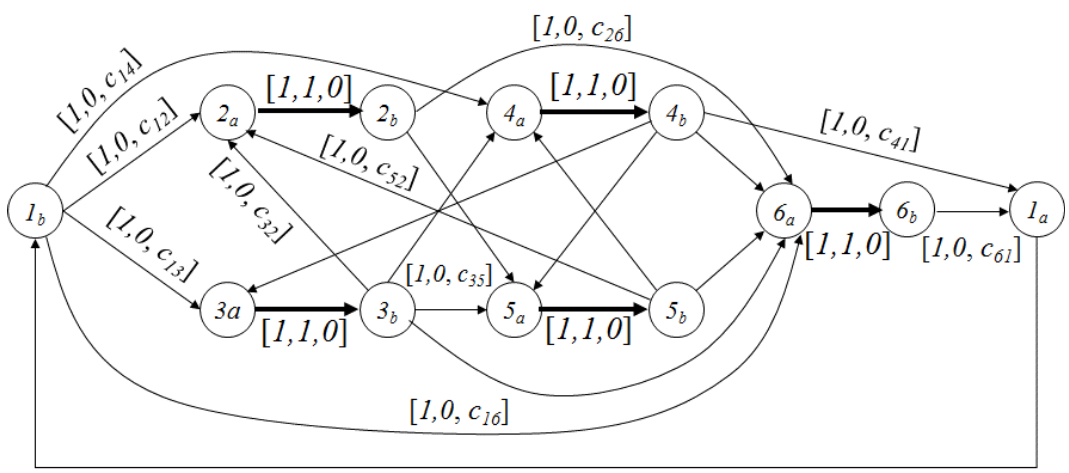


Figure 2: Network for MTSP



Case 1: [*1, 1, 0*] Case 2: [*2, 2, 0*]

Figure 3: Circulation network for Cases 1 and 2

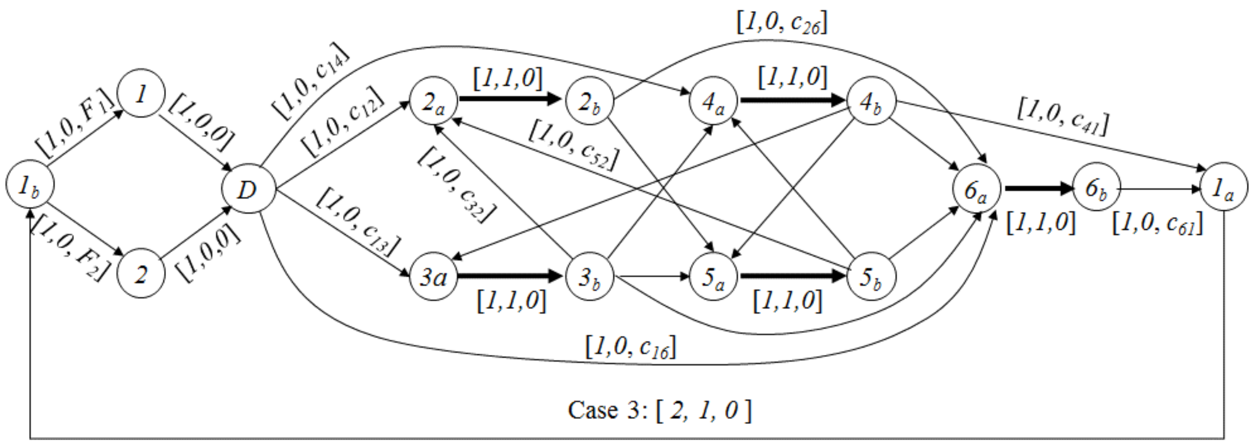


Figure 4: Circulation network for Case 3, $1 \leq M \leq 2$

Step 2. Identify Minimum-Cost Flows in the Circulation Network.

In the circulation network G'(**N'**,**A'**), the flow along arc (i_a, i_b) will be represented by f_{ab}^{i}, and that along arc (i_b, j_a) by f_{ba}^{ij}. The minimum-cost flows on the capacitated circulation network can be found using the Out-of-Kilter algorithm [8]. In the solution each arc satisfies the in-kilter (optimality) conditions given in (a), (b) and (c) below, and each node satisfies the flow-conservation condition given in (d):

(a) $\bar{c}_{ba}^{ij} < 0 \Rightarrow f_{ba}^{ij} = U_{ba}^{ij}$

(b) $\bar{c}_{ba}^{ij} > 0 \Rightarrow f_{ba}^{ij} = L_{ba}^{ij}$

(c) $\bar{c}_{ba}^{ij} = 0 \Rightarrow L_{ba}^{ij} \leq f_{ba}^{ij} \leq U_{ba}^{ij}$

(d) $\sum_{a \in \grave{N}} f_{ba}^{ij} - \sum_{a \in \grave{N}} f_{ab}^{ij} = 0$, for all $b \in \grave{N}$

where $\bar{c}_{ba}^{ij} = c_{ba}^{ij} + \pi_b - \pi_a$. Here π_a and π_b are the dual values associated with the flow conservation constraint for node *a* and *b*, respectively.

The steps of the Out-of-Kilter procedure are briefly and conceptually described as follows:

1. Starting solution (flow conservation satisfied by all nodes).
2. Determine the state (in-kilter or out-of-kilter) of each arc. If all arcs are in-kilter, stop with an optimal solution.
3. If all arcs are not in-kilter, choose any arc out-of-kilter and determine the possible flow correction needed. In order to add or subtract the required flow quantity and preserve the flow conservation conditions, the labeling procedure is performed.
4. If the labeling procedure yields a breakthrough, adjust flows. Otherwise perform the node-number change procedure.
5. Find new node-numbers and go to Step 2. If all–finite node numbers are not possible, stop with an infeasible solution.

An optimal solution to the circulation problem corresponds to either a set of sub-tours starting and finishing at the base city or a set of sub-tours not all starting at the base city. Those sub-tours not starting the base city are referred to as infeasible sub-tours. If a minimum-cost solution exists for the circulation network, all arcs (i_a, i_b) will have $f_{ab}^{i} = 1$; in this case, Step 3 is performed next. Otherwise, the procedure is stopped with an infeasible solution for the MTSP.

Step 3. Link Elimination. An infeasible sub-tour is defined as a closed chain which does not contain both nodes 1_a and 1_b. The infeasible sub-tour elimination procedure consists of three phases:

A. Identification of sub-tours to determine the feasibility of the solution (set of tours).
B. Link elimination procedure to break unacceptable sub-tours.
C. Preservation of flow-conservation conditions for the nodes of the network.

Figure 5 will be used to illustrate the sub-tour elimination procedure. This figure only shows the arcs belonging to sub-tours. All other arcs of the network are not shown in the figure. In Figure 5(a) it is assumed that the solution obtained by the Out-of-Kilter algorithm [6] has two sub-tours (loops). All arcs of the circulation network belonging to the loops have flows equal to one, and all arcs not included in the loops, which are not shown in the figure, have flows equal to zero. The loop formed with nodes 5, 6, 7 and 8 is an infeasible sub-tour because it does not start and finish at the base city (city 1). Assuming that the arc having the largest cost in the infeasible sub-tour is removed, then its flow must be rerouted in order to preserve the flow conservation condition at each node.

In Figure 5(a) it is assumed that arc (8_b, 5_a) is removed from the infeasible sub-tour by setting its flow equal to zero. To accomplish this and preserve flow conservation, as shown in Figure 5(b), the closed path consisting of nodes $5a$, 8_b, 1_a, 1_b, 6_a, 5_b, and 5_a is used to *reroute* one unit of flow. This effectively reduces the flow from 1 to 0 on all *reverse* arcs and increases the flow from 0 to 1 on all forward arcs. Figure 5(c) shows the final result.

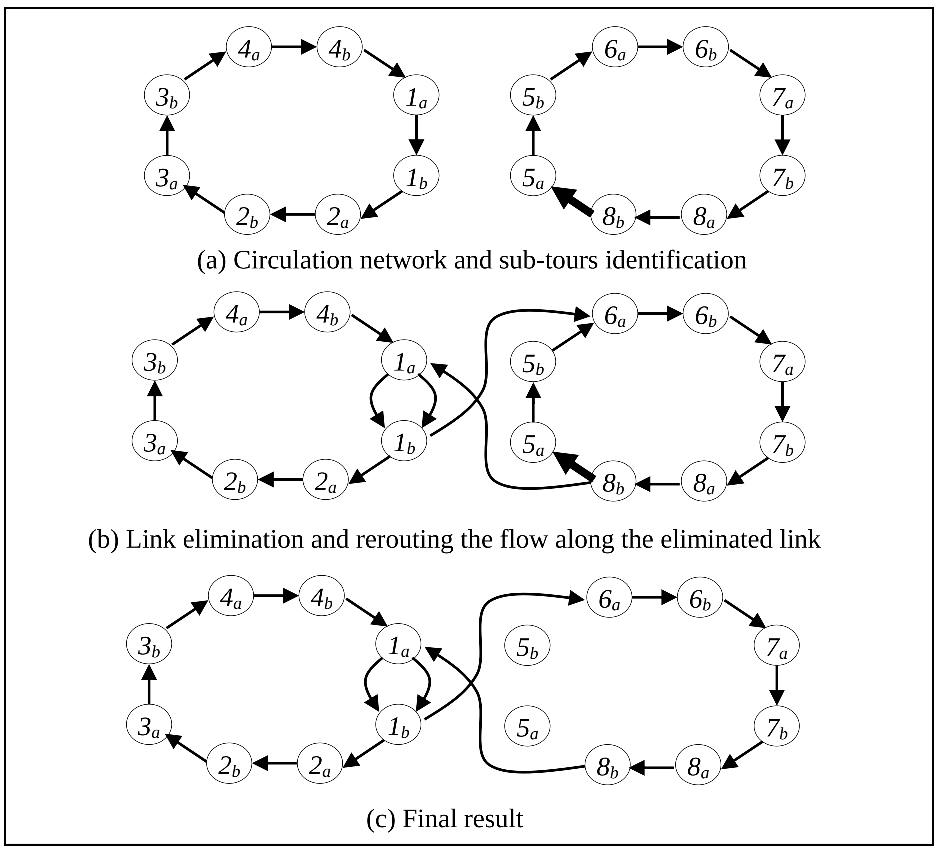
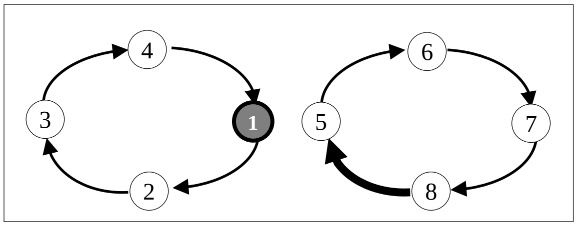
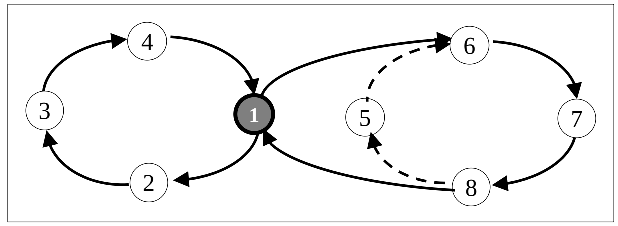


Figure 5: Sub-tours identification and link elimination procedures (original network)

In the context of the original network, the flow alterations on Figure 5 correspond to the modifications shown in Figure 6. Specifically, arc (8, 5) is removed, and its flow of 1 unit is rerouted along a path that starts at node 8 and ends at node 5. In Figure 6(a), it is assumed that the rerouting path consists of arcs (8, 1), (1, 6), and (6, 5).



(a) Identification of sub-tours and the link to be eliminated



(b) Final result

Figure 6: Sub-tours identification and link elimination procedures (circulation network)

The two loops shown in Figure 6(b) correspond to feasible sub-tours since both start and end at the base city. Section 2.3 summarizes the basic steps followed in the labeling procedure used to reroute flows on arcs eliminated to remove infeasible sub-tours.

2.3 Labeling Procedure

The purpose of a labeling procedure is to mark the arcs and nodes selected in the flow-alteration path that reroutes the unit of flow on one of the arcs of an infeasible sub-tour. In the proposed heuristic procedure we use the labeling procedure of the Out-of-Kilter algorithm.

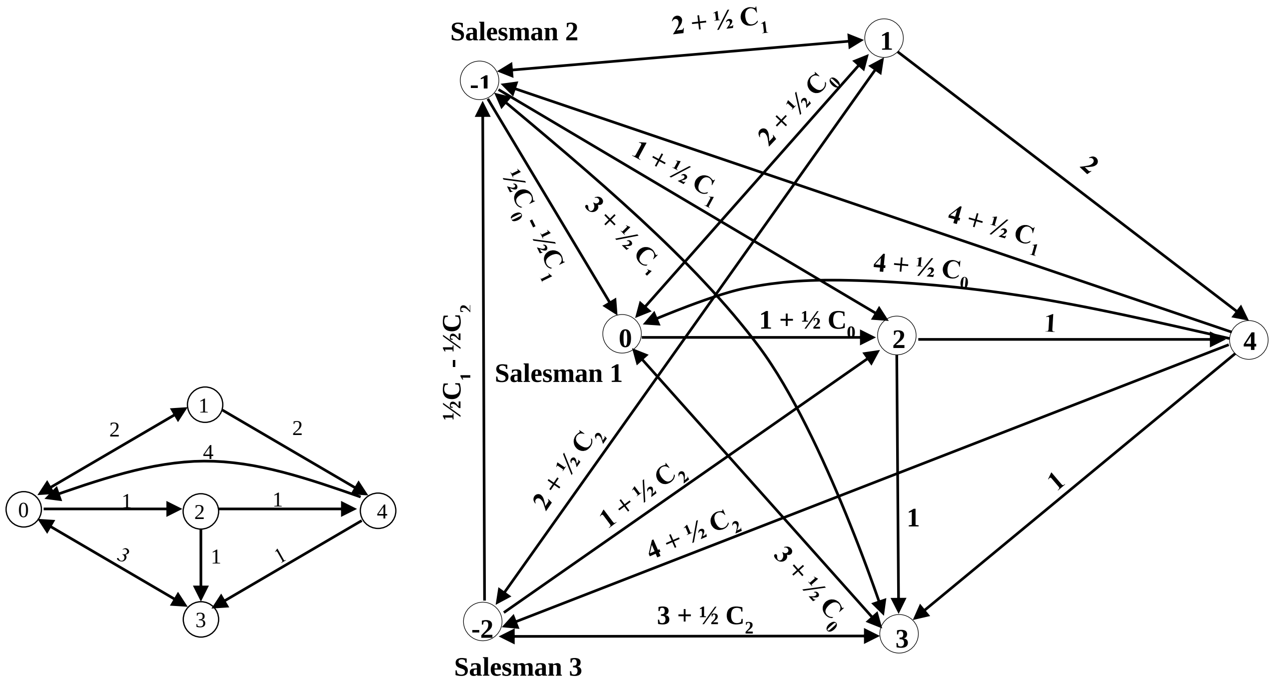
To simplify our presentation we will discuss the labeling procedure using the illustration given in Figure 5. As shown in Figure 5b, the flow-alteration path consists of nodes 8_b, 1_a, 1_b, 6_a, 5_b, and 5_a. Initially, node 8_b is labeled with $[1,5_a^-]$ to indicate that one unit of flow is sent from node 5_a in the direction opposite to the arc orientation, effectively resetting the flow on arc (8_b,5_a) to be equal to zero. Once the unit of low arrives at node 8_b, it is sent to node 1_a, assumed to exist although not shown in the figure because its flow is currently equal to zero. This corresponds to labeling node 1_a with $[1,8_b^+]$ since the unit of flow is sent in the same direction as the orientation of the arc. Now the unit is routed to node 1_b along arc (1_a, 1_b). Therefore, node 1_b is labeled with $[1,1_a^+]$. Assuming that there is an arc joining node 1_b to node 6_a, not shown in Figure 5b because its flow is currently zero, node 6_a is labeled with $[1,1_b^+]$. Similarly, we label node 5_b from node 6_a with $[1,6_a^-]$ and node 5_a from node 5_b with $[1,5_b^-]$.

3. Computational Results

3.1 Transformation of MTSP into Single-Traveling Salesman

For the analysis of the computational performance of the procedure several problems were generated and solved both with the proposed circulation network heuristic procedure, and then by transforming them into equivalent single-traveling salesman using the Bellman and Hong method [4] and subsequently solving them with Little's algorithm [6]. An *n*-city, *M*-salesman problem can be transformed into an equivalent TSP problems with $n+M-1$ cities by creating $M-1$ additional nodes and linking these nodes to all nodes of the original network. Figure 7 illustrates the transformation procedure for a five-city network and three salesmen. The base city in this example is represented by node 0. In this figure C_0, C_1 and C_2 are the fixed costs for salesmen 1, 2, and 3, respectively. It is assumed that the costs of the salesmen are such that $C_0 \leq C_1 \leq C_2$.

The fundamental idea for transforming the network of Figure 7(a) into the network of Figure 7(b) is to create two additional nodes -1 and -2 to represent the base city for Salesmen 2 and 3, respectively. All arcs out of or into city 0 are replicated for each of the additional nodes -1 and -2. Furthermore, an arc is added to connect node 0 to node -1 and another arc to connect node -1 to node -2. All other arcs of Figure 7(a) are then replicated in Figure 7(b) to finish the transformation process. The procedure for calculating arc costs is explained in reference [4].



(a) Original network (b) Transformed network

Figure 7: Illustration of network transformation procedure for Case 3

3.2 Sample run of computerized procedure (Case 3, M =2)

The purpose of this section is to show a sample run of the computerized procedure developed for the proposed heuristic method. Case 3 will be assumed. The number of cities is 8, the number of available salesmen is two, with costs equal to 250 and 278, and the trip cost matrix for this example is shown below:

$$\mathbf{C} = \begin{bmatrix} - & 131 & \infty & 19 & 40 & 28 & 32 & \infty \\ 35 & - & 70 & 33 & \infty & 86 & 29 & 39 \\ \infty & 95 & - & \infty & 70 & \infty & \infty & 20 \\ 26 & \infty & 34 & - & 46 & 23 & 5 & \infty \\ \infty & 20 & \infty & 86 & - & \infty & 40 & 100 \\ 13 & 11 & 30 & \infty & 125 & - & 37 & \infty \\ \infty & 45 & \infty & 33 & \infty & 117 & - & \\ 50 & \infty & 8 & \infty & 29 & 57 & \infty & - \end{bmatrix}$$

```
--------------------------------------------
     THE SOLUTION IS AS FOLLOWS:
 1 SALESMAN/SALESMEN OUT OF  2 IS/ARE USED
--------------------------------------------
     -------------------------
     SALESMAN    FIXED CHARGE
     -------------------------
           1            250
     -------------------------
  THE ROUTE FOR SALESMAN  1 IS AS FOLLOWS:
     ----------------------------
     FROM CITY    TO CITY   COST
     ----------------------------
           1          4     19
           4          6     23
           6          3     30
           3          8     20
           8          5     29
           5          7     40
           7          2     45
           2          1     35
     ----------------------------
     --------------------------
        SUMMARY OF SOLUTION
     --------------------------
     TOTAL FIXED CHARGE =  250
     TOTAL TRAVEL COST  =  241
     THE TOTAL COST IS  =  491
     --------------------------
```

Figure 8: Computer output for Case 3, *1≤M≤2*

Table 1: Optimal flows for circulation network, Case 3

Start node	End node	Meaning of arc in circulation network	Upper bound	Lower bound	Cost/unit	Arc flow
1b	2a	Trip from city 1 to city 2	1	0	131	0
1b	4a	Trip from city 1 to city 4	1	0	19	1
1b	5a	Trip from city 1 to city 5	1	0	40	0
1b	6a	Trip from city 1 to city 6	1	0	28	0
1b	7a	Trip from city 1 to city 7	1	0	32	0
2a	2b	City 2	1	1	0	0
2b	1a	Trip from city 2 to city 1	1	0	35	1
2b	3a	Trip from city 2 to city 3	1	0	70	0
2b	4a	Trip from city 2 to city 4	1	0	33	0
2b	6a	Trip from city 2 to city 4	1	0	86	0
2b	7a	Trip from city 2 to city 7	1	0	29	0
2b	8a	Trip from city 2 to city 8	1	0	39	0
3a	3b	City 3	1	1	0	1
3b	2a	Trip from city 3 to city 2	1	0	95	0
3b	5a	Trip from city 3 to City 5	1	0	70	0
3b	8a	Trip from city 3 to city8	2	1	20	1
4a	4b	City 4	1	1	0	1
4b	1a	Trip from city 4 to city 1	1	0	26	0
4b	3a	Trip from city 4 to city 3	1	0	34	0
4b	5a	Trip from city 4 to city 5	1	0	46	0
4b	6a	Trip from city 4 to city 6	1	0	23	1
4b	7a	Trip from city 4 to city 7	1	0	5	0
5a	5b	City 5	1	1	0	0
5b	2a	Trip from city 5 to city 2	1	0	20	0
5b	4a	Trip from city 5 to city 4	1	0	86	0
5b	7a	Trip from city 5 to city 7	1	0	40	1
5b	8a	Trip from city 5 to city 8	1	0	100	0
6a	6b	City 6	1	1	0	0
6b	1a	Trip from city 6 to city 1	1	0	13	0
6b	2a	Trip from city 6 to city 2	1	0	11	0
6b	3a	Trip from city 6 to city 3	1	0	30	1
6b	5a	Trip from city 6 to city 5	1	0	125	0
6b	7a	Trip from city 6 to city 7	1	0	37	0
7a	7b	City 7	1	1	0	0
7b	2a	Trip from city 7 to city 2	1	0	45	1
7b	4a	Trip from city 7 to city 4	1	0	33	0
7b	6a	Trip from city 7 to city 6	1	0	117	0
7b	8a	Trip from city 7 to city 8	1	0	21	0
8a	8b	City 8	1	1	0	0
8b	1a	Trip from city 8 to city 1	1	0	50	0
8b	3a	Trip from city 8 to city 3	1	0	8	0
8b	5a	Trip from city 8 to city 5	1	0	29	1
8b	6a	Trip from city 8 to city 6	1	0	57	0
1a	1b	Number of salesmen used	2	1	0	1
1b	1	Cost of salesman 1	1	0	250	1
1b	2	Cost of salesman 2	1	0	278	0
1	1b	Salesman 1	1	0	0	1
2	1b	Salesman 2	1	0	0	0

The output from the computer program is shown in Figure 8. As it can be seen, only one salesman is used. Furthermore, the cities are visited according to the sequence 1-4-6-3-8-5-7-2-1. The total cost is 491, including 241 for the trip costs plus 250 for the salesman. The minimal-cost flows for the circulation network with $1 \leq M \leq 2$ are summarized in Table 1. From this table, it can be seen that the optimal solution consists of one tour (one salesman) including the following trips: (1,4), (4,6), (6,3), (3,8), (8,5),(5,7),(7,2), and finally (2,1).

3.3 Summary of Computational Results

Table 2: Comparison of heuristic and optimal procedures

N	*M*	Case	Heuristic	Optimal
10	1	1	27	27
	1	1	17	17
	2	1	36	36
	2	2	38	38
	2	2	40	40
	3	2	55	55
	4	2	76	76
	5	2	99	99
	4	3	17	17
	5	3	27	27
20	1	1	37	37
	1	1	94	93
	2	2	52	52
	2	2	102	100
	3	2	121	118
	4	2	142	138
	5	2	165	160
	6	2	56	56
	4	3	37	37
	5	3	94	93
30	1	1	37	37
	1	1	44	44
	2	2	44	44
	2	2	50	50
	3	2	64	62
	4	2	81	81
	5	2	126	16
	4	3	37	37
	5	3	44	44
40	1	1	33	33
	1	1	61	61
	2	2	41	40
	2	2	77	77
	3	2	51	51
	4	2	65	56
	5	2	82	79
	4	3	61	61
	5	3	33	33
50	1	1	26	26
	1	1	88	88
	2	2	31	30
	2	2	96	96
	3	2	38	38
	4	2	50	48
	5	2	65	62
	4	3	88	88
	5	3	26	26

Forty-seven sets of non-symmetrical and symmetrical problems with the number of cities ranging from 10 to 50 were conducted to assess the accuracy of the heuristic solutions. In most of these problems the arc density was equal to 1 and for the remaining problems it was either 0.20 or 0.33. The density is defined as the number of actual arcs divided by the number of arcs in a totally connected network. The number of salesmen was varied between 1 and 5. The costs of going from one city to another were generated randomly according to a uniform discrete distribution between 1 and 10. The fixed costs were systematically set equal to 2, 4, 6…, etc. Data in Table2 were extracted from Reference [3]. In addition to being solved by the heuristic procedure, the MTSP problems were transformed into equivalent TSP problems and the corresponding optimal solutions were obtained by Little's algorithm. A summary of the results is given in Table 2.

4. Conclusions

It is noted that the heuristic method yields the optimal solution for 35 problems (approximately 74.5%) of the 47 problems. Moreover, in the remaining problems the average deviation from the optimal solution was within 3%. When the runs were separated for symmetric and non-symmetric problems it was noted that the heuristic algorithm performs faster in the case on non-symmetric data. The proposed heuristic procedure was used on additional sets of problems including up 100 cities and up to 10 salesmen. Based on all runs we concluded that this is an effective method to solve multi-traveling salesman problems for realistic applications. Moreover, this heuristic procedure incorporates optimality conditions for circulation problems and when all sub-tours are found to be feasible the solution is guaranteed to be optimal. Otherwise, infeasible sub-tours are eliminated and new sub-tours are determined using the labeling procedure of the Out-Kilter algorithm.

References

1. Garey, M. R. and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979.
2. Lawler, E. L., J. K. Lenstra, AH.G. Rinnooy Kan, and D. B. Shmoys, The Traveling Salesman Problem, John Wiley & Sons, New York, 1985.
3. Garcia-Diaz A., 1985, "A Heuristic Circulation-Network Approach to Solve the Multi-Traveling Salesman Problem," John Wiley & Sons, Networks, Vol. 15 (1985) 455-467.
4. Bellmore, M. and S. Hong, Transformation of Multi-salesman Problem to the Standard Traveling Salesman Problem. J. ACM. 21 (1974) 500-504
5. Christofides, N. and S. Eilon, Algorithm for Large-Scale Traveling Salesman Problems. Operational Res. Quart. 23 (1972) 511-518.
6. Little, J.D.C., K. G. Murty, D. W. Sweeney, and C. Karel, An Algorithm For The Traveling Salesman Problem, Operations Res., 11 (1963) 972-978.
7. Sedighpour, M., M. Yousefikhoshbakht, N. M. Darani, "An Effective Genetic Algorithm for Solving the Multiple Traveling Salesman Problem", *Journal of Optimization in Industrial Engineering 8 (2011) 73-79.*
8. Fulkerson, D. R., "The Out-of-Kilter Method for Minimal Cost Flow Problems," Journal of Applied Mathematics, 9 (1) (March 1961).
9. Bektas, T., The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega, Volume (Year): 34 (2006), Issue 3 (June), pp. 209-219.