Ibn Al Haitham Journal for Pure and Applied Science

Journal homepage: http://jih.uobaghdad.edu.iq/index.php/j/index

Using Travelling Salesman Principle to Evaluate the Minimum Total Cost of the Iraqi Cities

Sajjad Majeed Jasim **Faez Hassan Ali**

Department of Mathematics, Collage of Sciences, University of Mustansiriyah, Baghdad, Iraq.

sajadm49@gmail.com faezhassan@uomustansiriyah.edu.iq

Article history: Received 17 February 2019, Accepted 24 March 2019, Publish September 2019

Doi:10.30526/32.3.2286

Abstarct

The traveling salesman problem (TSP) is a well-known and important combinatorial optimization problem. The goal is to find the shortest tour that visits each city in a given list exactly once and then returns to the starting city. In this paper we exploit the TSP to evaluate the minimum total cost (distance or time) for Iraqi cities. So two main methods are investigated to solve this problem; these methods are; Dynamic Programming (DP) and Branch and Bound Technique (BABT). For the BABT, more than one lower and upper bounds are be derived to gain the best one. The results of BABT are completely identical to DP, with less time for number of cities (n), $5 \leq n \leq 25$. These results proof the efficiency of BABT compared with some good heuristic methods. We are suggesting some additional techniques to improve the computation time of BABT for $n \leq 80$.

Keywords: Travelling Salesman Problem, Dynamic Programming, Branch and Bound, Greedy method and Minimizing Distance Method.

1. Introduction

In the field of combinatorial optimization, the Traveling Salesman Problem (TSP) is probably the most famous and extensively studied problem, which aims to find the shortest Hamiltonian Cycle in a graph. This problem is NP-hard: that is to say, no polynomial time algorithm is known for solving this problem at present [1]. The algorithms for solving the TSP can be categorized into two main paradigms: exact algorithms and heuristic algorithms. The exact algorithms are guaranteed to find the optimal solution in an exponential number of steps. The major limitation of these algorithms is that they are quite complex and have heavy requirement of computing time [1]. For such reason, it is very difficult to find optimal solution for the TSP, especially for problems with a very large number of cities. Heuristic algorithms attempt to solve the Traveling Salesman Problem focusing on tour construction methods and tour improvement methods. Tour construction methods build up a solution step by step, while tour improvement methods start with an initial tour then try to transform it into a shortest tour [1].

2. TSP Background and Formulation

The first researcher, in 1932, considered the traveling salesman problem [2]. Menger gives interesting ways of solving TSP. He lays bare the first approaches which were considered during the evolution of TSP solutions. An exposition on TSP history is available in [2]. The mathematical formulation of TSP is as follows:
The distance between the towns i and j is marked with d_{ij}.

$$Minimize\ C = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij}$$

s. t.

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, \dots, n$$

x_{ij}=0 or 1.[2].

$$x_{ij} = \begin{cases} 1, \text{if city } j \text{ is reached from city } i \\ 0, \text{otherewise} \end{cases}$$

C is the total cost of travel.

3. Some Exact Methods to Solve TSP

3.1 Dynamic Programming [3].

Dynamic programming (DP) is a method of solving problems by breaking the solution into a set of steps or stages so that the solution of the problem can be viewed from a series of interrelated decisions. The inventor and the person responsible for the popularity of DP is Richard Bellman. In DP, a series of optimal decisions are made by using the principle of optimality. The principle of optimality: if the optimal total solution, then the solution to the kth stage is also optimal. With the principle of optimality is guaranteed that at some stage of decision making is the right decision for the later stages. The essence of DP is to remove a small part of a problem at every step, and then solve the smaller problems and use the results of the settlement to remedy the solution is added back to the issue in the next step. The algorithm of the DP is as follows:

Dynamic Programming Algorithm [3].

Initialization steps

- Complete directed graph (G).
- Non-empty finite set of vertices on a graph (*V*), *V* = {1, 2, 3, ... n}.
- The set of edges in a graph (E).
- The distance from *i* to *j* (the distance between cities) d_{ij}, *where* $d_{ij} \neq d_{ji}$.
- The series lines (*S*), $S \subseteq \{2, 3, ..., n\}$.
- The weight of the shortest path that starts at vertex *i* that through all vertices in *S* and ends at vertex 1 $(f(i, S))$, $i \notin S$ and $S \neq \emptyset$.

Therefore the steps in solving the TSP with DP are as follows:

Step 1: Determining the basis of the graph Hamiltonian that has been represented to be an adjacency matrix to the equation:

$$f(i,\emptyset) = d_{i,1},\ 2 \le i \le n$$

Step 2: Calculate *f (i, S)* for |*S*| = 1, then we can obtain *f (i, S)* for |*S*| = 2, until |*S*| = *n*-1. With the equation:

$$f(i,s) = \min_{j\in S}\{d_{ij} + f(j, S-\{j\})\}$$

Step 3: Having obtained the results from **step 2**, then calculate the equation of a recursive relationship by following equation:

$$f(1, v-\{1\}) = \min_{2\le k\le n}\{d_{1k} + f(k, V-\{1,k\})\}$$

Step 4: After calculating a recursive equation in **step 3**, will be obtained by weighting the shortest path, then to obtain an optimal solution or length of the shortest path for a graph is to calculate *f (1, {2, 3, 4, ..., n})* which means long lines from the initial vertex (1) to vertex 1 after passing through the vertices 2, 3, 4, .., n with any order (for the minimum) then locate the forming of the minimum optimal solution which has been obtained.

3.2 Branch and Bound Technique for Solving TSP

BAB technique (BABT) is most widely used in TSP by constructing a state space tree to find the optimal solution among all feasible solutions by taking the value of the objective function. Branch and bound was initially studies by Dantzig and a more description was provided by him in the applications of TSP. The BABT gives all feasible solutions by solving the problem, by trying the practical solution ad starting the value in the upper bound for finding the optimal solutions [4]. The algorithm of the BABT is as follows:

Branch and Bound Technique (BABT) Algorithm [5].

Step 1: Choose a starting point.
Step 2: Choose one of the routes for that point.
Step 3: After choosing that route between the current point and unvisited point add the distance. After doing that choose a new destination without choosing the same point.
Step 4: Keep doing this until we have gone through each point.
Step 5: Add up each distance of each subgroup.

4. Some Heuristic Methods to Solve TSP [6].

In this section we will discuss two heuristic methods; Greedy method, Branch and Bound Method and Improved Minimum distance method.

The Greedy method (GRM) starts by sorting the edges by length, and always adding the shortest remaining available edge to the tour. The shortest edge is available if it is not yet added to the tour and if adding it would not create a 3-degree vertex or a cycle with edges less than n. This heuristics can be applied to run in O (n^2log(n)) time.

The terms “Branch and Bound” represent all the state space search methods such that all the children of E-node are generated any now nodes called line node when it became E-node. E-node is the node, which can be expended. The live-node is node generated all of whose children are not yet been expanded. A node which cannot be expanded called dead node, but this node can be useful for backtracking concept. If there are no more children to expand then we have to reach its parent and expand its children and we do so until we obtain the solution or complete tree path, this method is different in technique from BABT mentioned in section (3-2). The Minimizing Distance Method (MDM) is an efficient method for finding a good solution, but it has a weak point. This weak point has been manipulated by improved minimum distance method (IMDM) which is suggested in [6]. The IMDM has good achievement with high efficiency for solving TSP.

5. Using DP to Solve TSP

In this section the method of DP was applied to solve TSP. The obtained results were better than the method of Complete Enumeration Method (CEM) in terms of time and number of cities.

In this paper set of practical examples are be used with different choices for n such that $5 \leq n \leq 80$ with integer distance such that $d_{ij} \in [1,30]$ for $5 \leq n \leq 30$ and $d_{ij} \in [1,100]$ for $30 < n \leq 80$. Before we discuss the results we have to define the following notations:

- **C:** The cost of travel.
- **CT:** Complete time in seconds.
- **R:** $R \in [0,1]$.
- **E:** The difference between set of methods with first method in table.

Table 1. shows the comparison results between DP from one side with CEM and the best Heuristic method [6]. IMDM from another side for n=5,…,15 for average of (3) examples.

Table 1. Comparison results between DP with CEM and IMDM for n=5,…,15.

n	DP		CEM		IMDM	
	C	CT	C	CT	C	CT
5	51	R	51	R	51.7	R
6	46.3	R	46.3	R	50.3	R
7	53	R	53	R	53	R
8	56.7	R	56.7	R	56.7	R
9	43.3	R	43.3	R	43.3	R
10	52.7	1.7	52.7	3.8	52.7	R
11	61.7	7.4	61.7	37.1	63.7	R
12	42.3	33.7	42.3	409.1	42.3	R
Av-all	50.9	14.3	50.9	150	51.7	R
13	60.3	156.5	-	-	71	R
14	55	696.1	-	-	62	R
15	46.7	3114.6	-	-	50.7	R
Av-all	51.7	364.8	-	-	54.3	R

6. Using BABT to Solve TSP

In this section we will discuss and the applying of BABT to solve TSP. It is very well known that BABT is one of the most important methods of the exact solution for combinatorial optimization problem. This method can act with different upper and lower bound to get very good results within a good time. The choosing process of upper bound (UB) and lower bound (LB) is figured as (UB-LB) this symbol of UB and LB we called it a model for BABT with notation BABT: (UB-LB).

6.1. Suggesting an Upper Bounds for TSP

In order to use BABT to solve TSP, we apply three methods for finding three different UB; and these methods are IMDM, GRM and BABM.

6.2. Derivation of Different Lower Bounds for TSP

The LB is one of the most important parts of this method. The LB consists of two main parts such that LB= Sequenced nodes plus the unsequenced nodes, the sequence nodes: is the basic rout until the currant node. While the unsequenced nodes: it's the subsequence obtained from all the cities after eliminate the sequence nodes which are obtained from applying deterministic method. Several methods have been proposed to calculate the LB; these methods are GRM, BABM and IMDM. But based on the results obtained from these methods it was concluded that the best method is the IMDM because it gives accurate results with reasonable time.

6.3. Two Proposed BAB Techniques

In this subsection we proposed two techniques for BAB, the first technique is the classical BABT with notation BABT1, which is mentioned in subsection 3.2. The BABT1 algorithm is as follows:

BABT1 Algorithm

Step 1: Read number of cities (n); Read Distance table.

Step 2: Calculate UB=Cost (N) using GRM, BABM or IMDM where N= {1, 2, … , n }; i=0.

Step 3: For each node in the search tree compute the LB= cost of sequencing nodes + Cost of unsequenced nodes; where cost of unsequenced nodes is obtained by GRM, BABM or IMDM, i=i+1.

Step 4: Branch each node with LB ≤ UB for level i.

Step 5: If i < n then goto **step 3**.

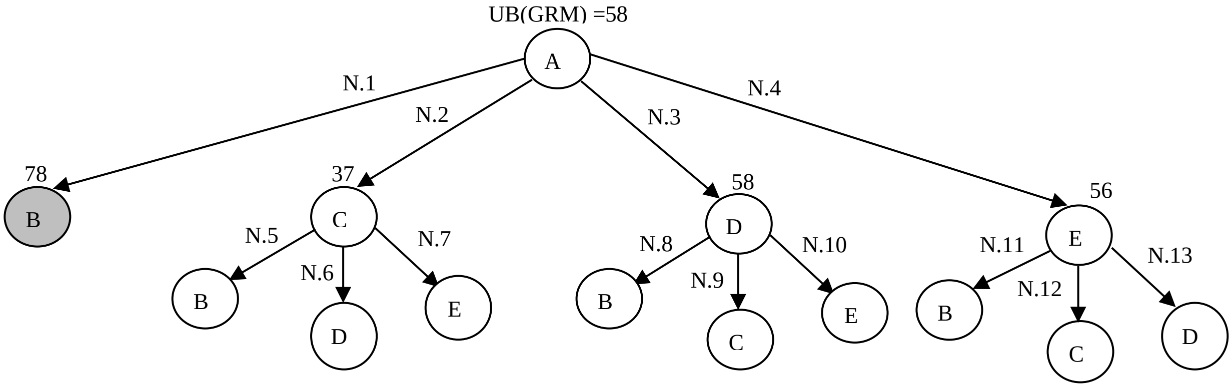
Step 6: If the last level (i=n-2) of BAB algorithm we obtain the optimal solution.

Step 7: Stop**.**

Example 1. Let's have the following TSP:

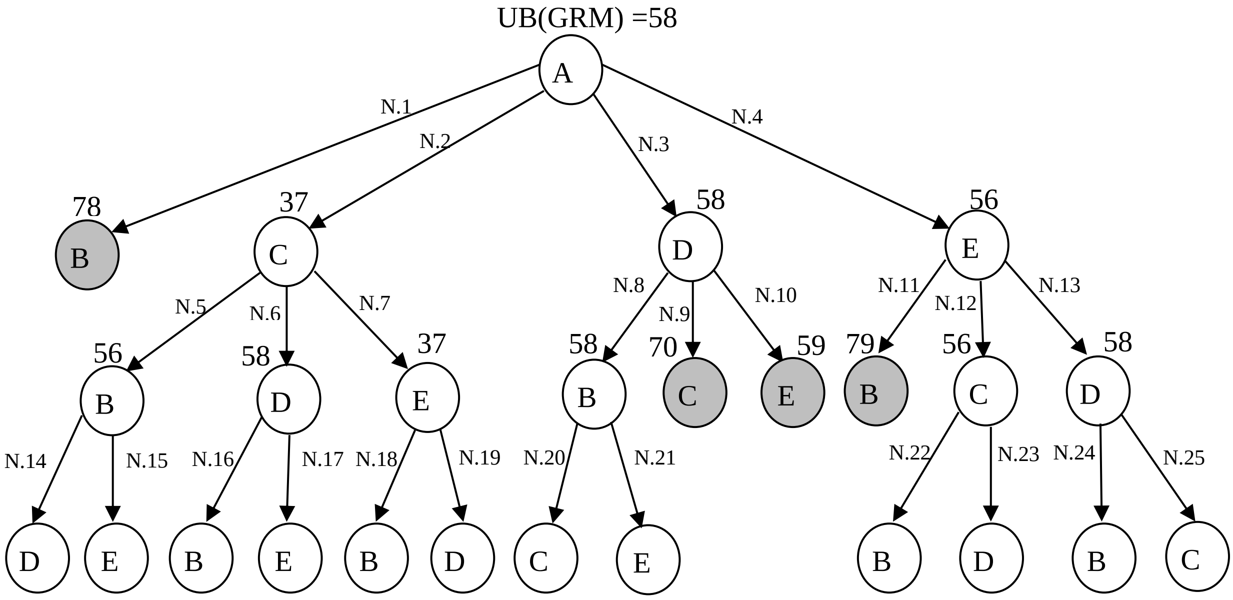
	A	B	C	D	E
A	−	30	8	10	17
B	11	−	13	9	10
C	19	12	−	9	9
D	18	1	18	−	8
E	30	22	18	8	−

To calculate UB, we use GRM so UB=58, for unsequnced nodes we use IMDM. For the 2nd level we have the following tree:

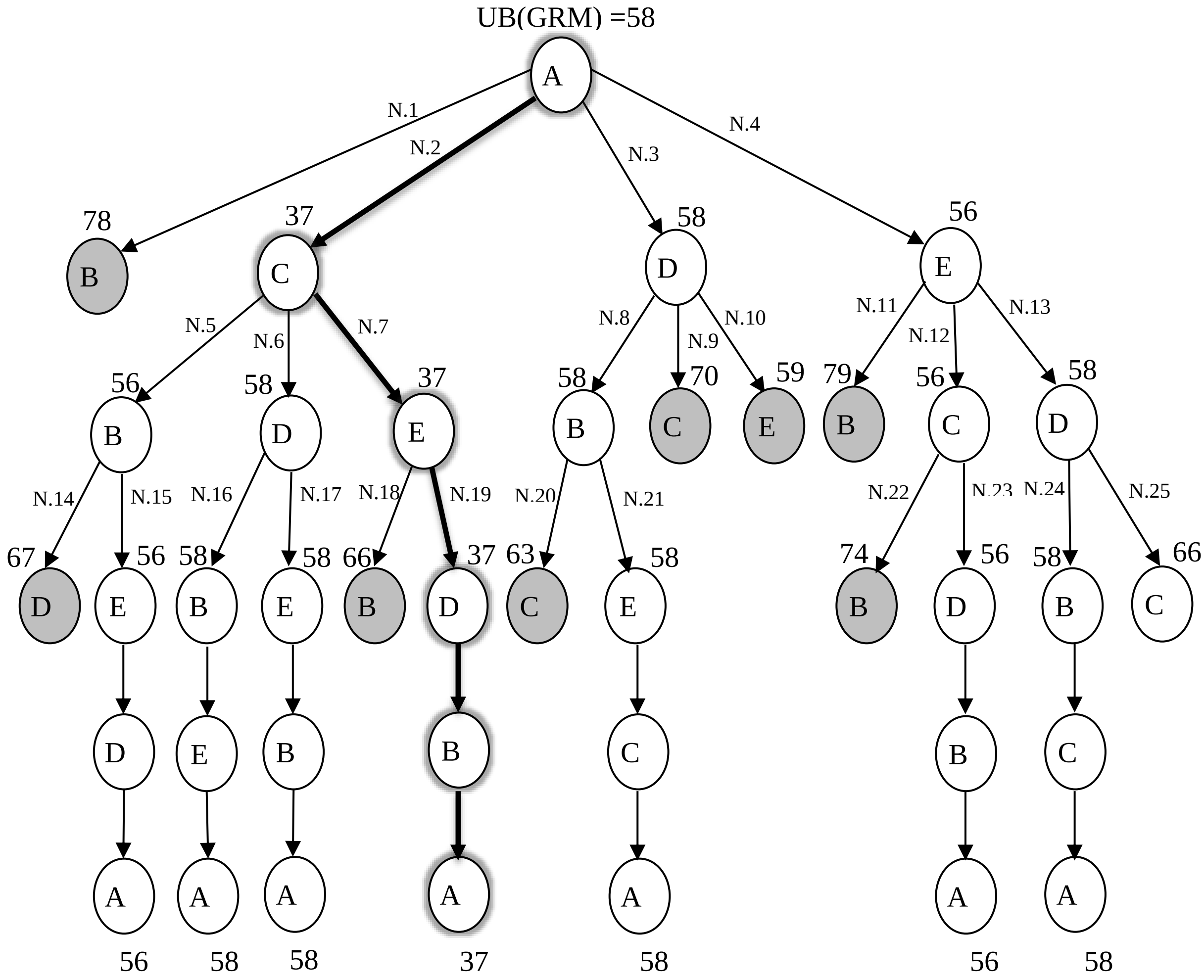


The shaded node is ignored.

It's clear that we ignore N.1, while for the 3rd level we obtain the following tree:



So we ignore the nodes N.9, N.10 and N.11. In the last level we have the following tree:



So the least cost C=37 with shaded path: **A→C→E→D→B→A**.

The second technique is similar for BABT1 but with modification. This modification includes finding a LB by branch form the least cost node and continue until get the root node and calculate the LB, then we update the initial UB by the new LB, and then apply the same steps of BABT1. The BABT2 algorithm is as follows:

BABT2 Algorithm

Step 1: Read number of cities (n); Read Distance table.

Step 2: Calculate UB=Cost (N) using GRM or BABM where N= {1,2,…,n}, UB1=UB.

Step 3: Compute the New_LB= cost of sequencing nodes + Cost of unsequenced nodes; where cost of unsequenced nodes is obtained by IMDM, if New_LB ≤ UB1 branch from this node and set UB1=New_LB, repeat until reach the root node set UB=UB1, if all New_LB > UB1 then UB=UB1.

Step 4: i=0, with upper bound UB.

Step 5: For each node in the search tree compute the LB is obtained by IMDM, i=i+1.

Step 6: Branch each node with LB ≤ UB for level i.

Step 7: If i < n then goto **step 5**.

Step 8: If the last level (i=n-2) of BAB algorithm we obtain the optimal solution.

Step 9: Stop.

7. Comparative Results of Applying BABT1 for different UB and LB

In this section all proposed types of BAB, UB and LB models will be presented, these models are (UB-LB): GRM-GRM, GRM-BABM, GRM-IMDM, BABM-BABM, BABM-IMDM and IMDM-IMDM.

All the above models have been tested and it has been shown that the best models (cost and time) in terms of results are GRM-IMDM, IMDM-IMDM and BABM-IMDM.

7.1 Results of Applying BABT1

Table 2 shows the comparison results of BABT1 for different models one side and with DP from the other side for n=5,…,15 for average of (3) examples.

Table 2. Comparison results of DP with BABT1 for n=5,…,15.

n	DP	UB			IMDM-IMDM			GRM-IMDM			BABM-IMDM		
		IMDM	GRM	BABM	C	E	CT	C	E	CT	C	E	CT
5	51	51.7	58.7	51.7	51	0	R	51	0	R	51	0	R
6	46.3	50.3	59.7	55	46.3	0	R	46.3	0	R	46.3	0	R
7	53	53	67	53	53	0	R	53	0	R	53	0	R
8	56.7	56.7	83	68.3	56.7	0	R	56.7	0	1.9	56.7	0	R
9	43.3	43.3	69.3	43.3	43.3	0	R	43.3	0	2.8	43.3	0	R
10	52.7	52.7	61.7	53.3	52.7	0	R	52.7	0	R	52.7	0	R
11	61.7	63.7	77.7	67.3	61.7	0	R	61.7	0	7.4	61.7	0	5.8
12	42.3	42.3	70.7	48.3	42.3	0	R	42.3	0	102.8	42.3	0	1.4
13	60.3	71	82	68.7	60.3	0	30.7	60.3	0	76.8	60.3	0	8.1
14	55	62	78.7	64.7	55	0	46.0	55	0	148.8	55	0	50.8
15	46.7	50.7	61	57.7	46.7	0	4.7	46.7	0	18.7	46.7	0	43.0
Av-all	51.7	54.3	69.95	57.4	51.7	0	7.8	51.7	0	32.8	51.7	0	10.2

From the results of **Table 2.** we notice that all models are identical to DP but IMDM-IMDM is the best in the time.

Table 3. shows the comparison results of BABT1 for the three models with each other for n=16,…,20,25.

Table 3. Comparison results of BABT1 for the three models with each other for n=16,…,20,25.

n	UB			IMDM-IMDM		GRM-IMDM			BABM-IMDM		
	IMDM	GRM	BABM	C	CT	C	E	CT	C	E	CT
16	50.3	67.3	54.7	45	11.3	45	0	534.3	45	0	22
17	53.7	68.3	60.7	53.7	1.6	53.7	0	179.3	53.7	0	170.3
18	53	67.3	60.7	49.3	25.5	49.3	0	354	49.3	0	230.6
19	48.7	61	52.3	48.7	2	48.7	0	84.1	48.7	0	8
Av-all	51.4	66	57.1	49.2	10.1	49.2	0	287.9	49.2	0	107.7
20	60	68.7	79	58.7	6.8	58.7	0	496.5	-	-	-
25	79.3	120.7	96	71.3	721	-		-	-	-	-

7.2 Results of Applying BABT2

For BABT2 we notice that all the results for the three models are identical with each other's in cost and time so we choose one of the models to describe and compare its results with others methods.

Table 4. Shows the comparison results between BABT1: IMDM-IMDM (or DP since they are identical) and BABT2: GRM-IMDM model for n=5,…,20,25 for average of (3) examples.

Table 4. Comparison results between BABT1 and BABT2 for n=5,…,20,25.

n	BABT1: IMDM-IMDM		BABT2: GRM-IMDM		
	C	CT	C	E	CT
5	51	R	51	0	R
6	46.3	R	48	1.7	R
7	53	R	53	0	R
8	56.7	R	56.7	0	R
9	43.3	R	43.3	0	R
10	52.7	R	52.7	0	R
11	61.7	R	61.7	0	R
12	42.3	R	42.3	0	1.2
13	60.3	30.7	62	1.7	1.5
14	55	45.97	57	2	1.6
15	46.7	4.7	46.7	0	2.3
16	45	11.3	45	0	2.3
17	53.7	1.6	53.7	0	4.3
18	49.3	25.5	50	0.7	4.8
19	48.7	2.03	48.7	0	6.1
20	58.7	6.8	58.7	0	5.9
25	71.3	721	74.3	3	16.8
Av-all	52.7	50.2	53.2	0.5	3.0

Figure 1. show the comparison results of **Table 4.**

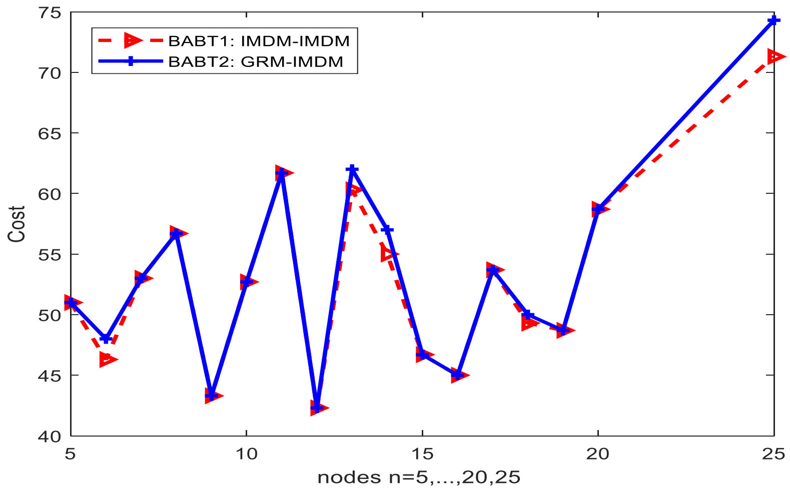


Figure 1. Comparison results between BABT1: IMDM-IMDM with BABT2: GRM-IMDM for n=5,…,20,25.

Table 5. shows the comparison results between best of heuristic methods: IMDM with BABT2 for n=30,…,80 for average of (3) examples.

Table 5. Comparison results between IMDM with BABT2 for n=30,…,80.

n	IMDM		BABT2: GRM-IMDM		
	C	CT	C	E	CT
30	58	R	56	-2	25
40	182	R	172.3	-9.7	651.5
50	201.3	R	180.3	-21	431.4
60	227	R	199.7	-27.3	888.1
70	233.3	R	218.7	-14.6	390.9
80	236.7	1.4	207.7	-29	839.0
Av-all	189.7	0.7	172.5	-17.3	537.7

Figure 2. show the comparison results of **Table 5**.

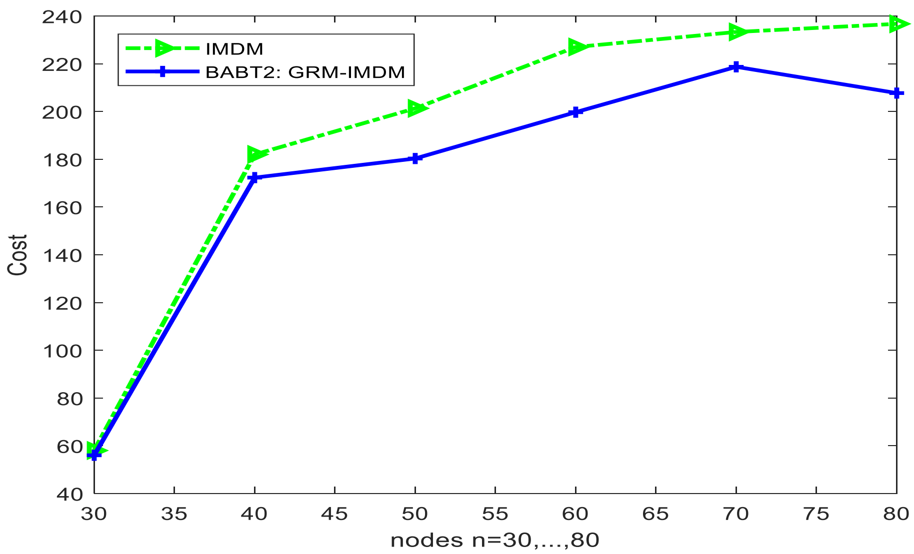


Figure 2. Comparison results between BABT2: GRMM-IMDM with IMDM for n=30,…,80.

8. Solving the Minimum Total Cost for the Iraqi's Cites as a TSP

In this section we will use the TSP as an application to compute the minimum total cost for n=18 Iraqi's cites. This done by applying the best method; the BABT1: (IMDM-IMDM) (see **Table 4**. The symbol of each city is as in **Table 6**.

Table 6. The symbol of Iraqi's Cites.

City	**Baghdad**	**Baqubah**	**Diwaniyah**	**Hillah**	**Ramadi**	**Karbala**	**Najaf**	**Kut**	**Tikrit**
Symbol	Bg	Bq	Dw	Hl	Rm	Kb	Nj	Ku	Tk
City	Kirkuk	Samawah	Sulaymaniyah	Nasiriyah	Erbil	Amarah	Mosul	Duhok	Basrah
Symbol	Kr	Sm	Sl	Ns	Er	Am	Ms	Dh	Bs

First we demonstrate the distance in km in **Table 6.** for the 18-cites which are represent the governorates centers [7].

Table 7. The distance (Km) between the Iraqi's cities.

	Bg	Bq	Kb	Hl	Rm	Dw	Nj	Ku	Tk	Kr	Sm	Sl	Ns	Er	Am	Ms	Dh	Bs
Bg	–	69	107	115	120	162	172	182	242	267	272	335	345	364	373	404	477	548
Bq	69	–	180	187	181	234	244	226	172	213	345	272	417	310	467	362	435	618
Kb	107	180	–	47	148	132	77	211	293	373	234	442	315	470	369	505	578	502
Hl	115	187	47	–	194	86	58	165	306	387	171	455	269	484	323	519	592	457
Rm	120	181	148	194	–	279	223	301	185	288	371	400	443	384	491	398	471	631
Dw	162	234	132	86	279	–	83	130	346	463	100	531	192	560	246	595	668	379
Nj	172	244	77	58	223	83	–	225	370	447	164	515	257	543	311	578	651	450
Ku	182	226	211	165	301	130	225	–	368	433	242	432	182	530	192	580	653	373
Tk	242	172	293	306	185	346	370	368	–	121	457	234	529	218	554	231	304	717
Kr	267	213	373	387	288	463	447	433	121	–	540	112	612	98	616	174	246	799
Sm	272	345	234	171	371	100	164	242	457	540	–	640	107	668	255	703	776	295
Sl	335	272	442	455	400	531	515	432	234	112	640	–	613	180	624	284	336	799
Ns	345	417	315	269	443	192	257	182	529	612	107	613	–	713	146	753	826	200
Er	364	310	470	484	384	560	543	530	218	98	668	180	713	–	722	85	164	900
Am	373	467	369	323	491	246	311	192	554	616	255	624	146	722	–	760	833	179
Ms	404	362	505	519	398	595	578	580	231	174	703	284	753	85	760	–	75	946
Dh	477	435	578	592	471	668	651	653	304	246	776	336	826	164	833	75	–	1016
Bs	548	618	502	457	631	379	450	373	717	799	295	799	200	900	179	946	1016	–

In the same time we have to estimate another cost which is represented by time factor by using distance cost mentioned in **Table 6.** In order to estimate the time cost we have to use the following transformation:

$$T = D / V \qquad (4)$$

Where T is the time, D is the distance and V is the velocity factors respectively.

Table 8. describes the time cost in minutes depending on distance cost mentioned in **Table 7**. using constant velocity 70km/hour.

Table 8. The Time (minute) between the Iraqi's cities.

	Bg	Bq	Kb	Hl	Rm	Dw	Nj	Ku	Tk	Kr	Sm	Sl	Ns	Er	Am	Ms	Dh	Bs
Bg	–	59	92	99	103	139	147	156	207	229	233	287	296	312	320	346	409	470
Bq	59	–	154	160	155	201	209	194	147	183	296	233	357	266	400	310	373	530
Kb	92	154	–	40	127	113	66	181	251	320	201	379	270	403	316	433	495	430
Hl	99	160	40	–	166	74	50	141	262	332	147	390	231	415	277	445	507	392
Rm	103	155	127	166	–	239	191	258	159	247	318	343	380	329	421	341	404	541
Dw	139	201	113	74	239	–	71	111	297	397	86	455	165	480	211	510	573	325
Nj	147	209	66	50	191	71	–	193	317	383	141	441	220	465	267	495	558	386
Ku	156	194	181	141	258	111	193	–	315	371	207	370	156	454	165	497	560	320
Tk	207	147	251	262	159	297	317	315	–	104	392	201	453	187	475	198	261	615
Kr	229	183	320	332	247	397	383	371	104	–	463	96	525	84	528	149	211	685
Sm	233	296	201	147	318	86	141	207	392	463	–	549	92	573	219	603	665	253
Sl	287	233	379	390	343	455	441	370	201	96	549	–	525	154	535	243	288	685
Ns	29	35	27	23	38	16	22	15	45	52	92	52	–	61	12	64	70	17

	6	7	0	1	0	5	0	6	3	5		5		1	5	5	8	1
Er	312	266	403	415	329	480	465	454	187	84	573	154	611	−	619	73	141	771
Am	320	400	316	277	421	211	267	165	475	528	219	535	125	619	−	651	714	153
Ms	346	310	433	445	341	510	495	497	198	149	603	243	645	73	651	−	64	811
Dh	409	373	495	507	404	573	558	560	261	211	665	288	708	141	714	64	−	871
Bs	470	530	430	392	541	325	386	320	615	685	253	685	171	771	153	811	871	−

Now we suggest to apply more than one method like DP, BABT1: IMDM-IMDM and BABT2: GRM-IMDM for distance cost in **Table 6.** and time cost in **Table 7.** for n=18. Before we describe the results we suggest using successive rules (SR's) to be certain that what we obtain is optimal path. These SR's is as follows:

1. Since the subpath Ms-Dh-Er is the only and the minimum cost of available path this mean that we have n=16.

2. Let's add another certain subpath to the previous one, Am-Bs-Ns, this mean we have n=14.

Table 9. shows the results of the three suggested methods for n=18 (without SR) and for n=16 and 14 (with SR).

Table 9. The results of the three suggested methods for different n with and without SR.

Method	**n**								
	18			**16**			**14**		
	Distance	**Time**	**CT**	**Distance**	**Time**	**CT**	**Distance**	**Time**	**CT**
DP	-	-	-	2502	2145	13677	2502	2145	673
BABT1	2502	2145	8	2502	2145	2	2502	2145	1.6
BABT2	2502	2145	3	2502	2145	2	2502	2145	1.6

For the above optimal costs we have the following unique symmetric path.

Path is:

"Bg→Bq→Sl→Kr→Er→Dh→Ms→Tk→Rm→Kb→Hl→Nj→Dw→Sm→Ns→Bs→Am →Ku→Bg"

It's important to mentioned that we get better solution (for distance 2496 and for time 2140) than the obtained solutions mentioned above but the path doesn't satisfies the constraints of TSP. this mean we obtained the optimal solutions for the total costs of the Iraqis cities. **Figure 3.** shows the best path with optimal minimum total costs for the Iraqi's cities.

To verify our optimal result DP software obtained from [8]. implemented to solve Iraqi cities problem and obtained same result in compotation time 26007s.

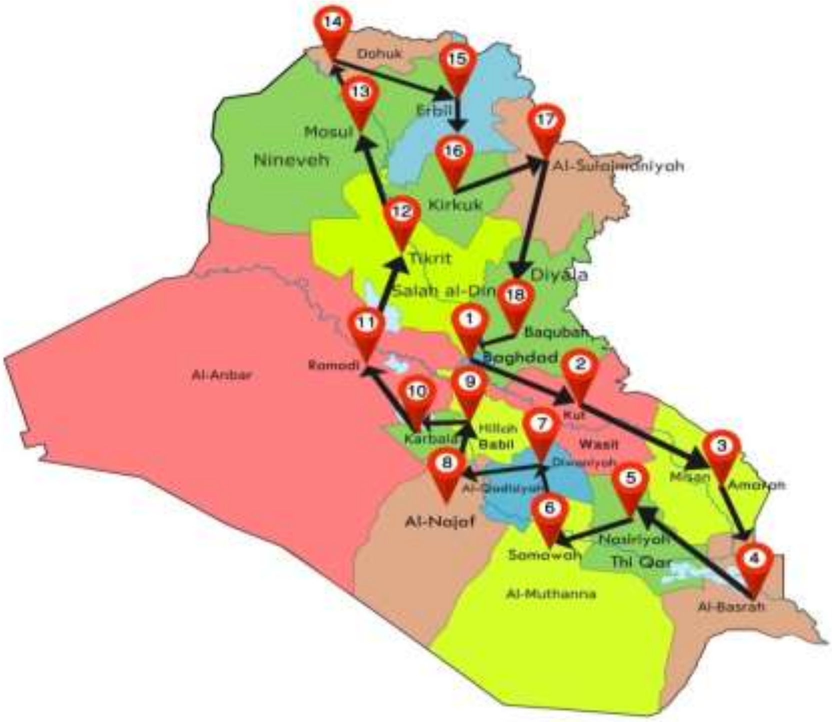


Figure 3. The best path with optimal minimum total costs for the Iraqi's cites.

9. Conclusions

1. From **Table 1**.we conclude that the DP is an exact and efficient method for solving TSP for n ≤ 15 in reasonable time.

2. The IMDM serves a good method as LB for BABT to solve the TSP for different n compared with other methods like GRM and BABM.

3. The results of the practical examples of TSP, proof that the BABT1 is better than BABT2 in cost for n ≤ 25, while BABT2 is better in time for different n.

4. Since TSP considered as NP-hard problem, we recommend to use some perfect local search methods like, Genetic Algorithm, Particle Swarm Optimization, Bees Algorithm, …,etc, to find an optimal or near optimal solution.

References

1. Xiang, Z.; Chen, Z.; Gao, X.; Wang, X.; Di, F.; Li, L.; Liu, G.; Zhang, Y. Solving Large-Scale TSP Using a Fast Wedging Insertion Partitioning Approach. Hindawi Publishing Corporation, *Mathematical Problems in Engineering*.**2015**, *2015*, 1-8, doi.org/10.1155/2015/854218.
2. Akandwanaho, S.M.; Adewumi, A.O.; Adebiyi, A.A. Solving Dynamic Traveling Salesman Problem Using Dynamic Gaussian Process Regression. Hindawi Publishing Corporation, *Journal of Applied Mathematics*.**2014**, *2014*, 1-10, doi.org/10.1155/2014/818529.
3. Nugroho, A. Solving Traveling Salesman Problem By Dynamic Programming Approach In Java Programming. CE 6001 Operations Management And Infrastructure System, Department Of Civil Engineering National University Of Singapore,**2010**.
4. Ranjana, P. Travelling salesman problem using reduced algorithmic Branch and bound approach. *International Journal of Pure and Applied Mathematics*.**2018**, *118*, 419-424, http://www.ijpam.eu.

5. Pineda, P. Implementation and solutions of the Traveling Salesman Problem (TSP) in R. In Partial Fulfillment of Stat 4395-Senior Project Department of Mathematics and Statistics Spring,**2017**.
6. Jassim, S.M.; Ali, F.H. New Improved Heuristic Method for Solving Travelling Salesman Problem. *Iraqi Journal of Science*.**2018**, *59*, 2289-2300.
7. Tawfiq, L.N.M.; Jabber, A.K. New Transform Fundamental Properties and its Applications.*Ibn Alhaitham Journal for Pure and Applied Science*.**2018**, *31*, *1*, 151-163. doi.org/10.30526/31.2.1954.
8. Jasim, K. A.; Abdulhmeed, E. O.; Tawfiq, L. N. M. Numerical model for Estimation the concentration of heavy metals in soil and its application in Iraq. *Global Journal of Engineering Science and Research*. **2016**, *3*, *3*, 75-81.