USING TABU SEARCH ALGORITHM FOR SOLVING TWO-STAGE TRANSPORTATION PROBLEM BASED ON DECODING PROCEDURE

M. S. Salman
Technical Institute- Suwaira, Middle Technical University, Suwaira, Iraq
Contact: SUWIRATECH@MTU.EDU.IQ
lover.young99@gmail.com

ABSTRACT: *Supply Chain Management (SCM) is the process that optimizes the flow of goods, services, and information from the source to the end-user. In SCM, one of the most critical areas is the transportation structure, which provides many opportunities for cost savings and the enhancement of service delivery. This paper examines the two-stage transport problem (TsTP) to determine ways of lowering the overall logistics cost, namely the start-up costs of distribution centers (Distribution centers) and shipping costs from plants to distributing centers to clients. Toward this end, we designed a Priority Tabu search Algorithm (TS), in which decoding procedures are used to adapt to theTsTP, and conducted a two-stage experimental study. We compared tabu search with genetic algorithm and simulated annealing algorithm in different problems. The results showed effective tabu search for solving the problem compared with the two algorithms.*

Keywords:Two-stage transportation problem (tsTP), Tabu Search algorithms (TS), Priority-based decoding.

1. INTRODUCTION

The Transport Problem (TP) involves moving products from various originating sources to various destinations. The purpose of the TP is to establish the shipping schedule that incurs the lowest overall shipping cost while meeting the supply and demand requirements. The TP is applied in industry, planning, communication networks, to schedule transportation and attribution, etc. From the [1, 2, 3, 9], there is abundant information to provide an ideal and balanced solution for TP. In some cases, for reasons of storage limitations, destinations are able to accept only their minimum requirements. As such, delivery of quantities beyond the minimum storage capacity cannot be made in one shipment and so items are delivered to original destinations in two stages. In the initial stage, the minimum destination requests are sent from the source to destinations. Following consumption of this initial shipment, the original destinations are ready to accept the additional quantity in the second step. Such transport issues are referred to as two-stage Transport problems. The primary purpose of two-stage transport is to move the product from the source of origin to the destinations in a two-stage operation. In this way, the total transport costs in both stages are minimal. In both steps shipping of the product from the origin to the destination is carried out in parallel.

Sonia and Rita Malhotra [16] recommended a polynomial-bound algorithm for two minimizations of stage time to achieve ideal timetables for the first and second stages in the shortest possible total transport times, Sonia, Rita Malhotra and Puri [17] offered a polynomial time algorithm for two times minimization of the problem to achieve an ideal overall solution. Pandian and Natarajan [5] suggested a novel approach known as a zero point approach to find the best solution to the transport problem by applying a basic approach, a procedure for solution testing and optimality. Nagoor Gani and Abdul Razak [4] developed a two-step cost problem reducing hazy transport whereby supplies and demands are trapezoidal fuzzy numbers with a parametric approach. Ritha and Merline Vinotha [22] suggested a way to find a better acceptable solution to a multi-objective with two stages of the blurry transport problem employing a geometric programming approach.

SCM is the process that optimizes the flow of goods, services and information from the source to the end-user. Typical SCM objectives encompass designing the transmission system, plant, locations of distribution centers, perfecting the production process and, enhancing the response time to customer orders. Designing a proper transportation structure is a crucial aspect of the SCM as the area of transportation can potentially minimize costs and at the same time enhance the quality of customer service. The Transport Problem (TP) is an essential networking challenge first highlighted by Hitchcock [14]. The very concept of TP is based on the objective of transporting homogeneous products from various sources to different destinations such the overall cost can be reduced. In practice, the transportation exercise has to meet several demands in a multiplicity of steps. Essentially, the logistics of the transportation design can be described as the capability to do everything right: ship the right quantity of the right goods to the right location at the right moment (and preferably at the right cost) (Tilanus, [25]). The effectiveness of the logistics method is impacted by several variables: the number of distribution centers to be established and where to locate them in order to accommodate customer demand at lowest possible cost and lowest possible shipping cost.

This paper takes into consideration an expansion of the TsTP. The challenge is to decide on the ideal structure that can meet customer demand at least cost based on plant capacity and the distribution centers and also the highest viable number of distribution centers to establish . It must be noted that the majority of companies do not have unlimited resources and therefore have to plan carefully what they can afford. As such, it is critical for a manager to determine just how many distribution centers the company can afford establish and run and where the locations of these distribution centers should be. In this study, the limitation to the number of distribution centers to be established is a significant constraint.

Developing models and approaches to solve the problem of installation locations and the number of installations ([1], ReVelle & Laporte, 1996, [21]) depends on the respective limited resources of individual companies. The problem is the location of the p-medial installation and Problem NP-Hard [6]. The tsTP is studied in this paper and includes the problem of localization of the p-median plant in capacity, as well as the NP-Hard, discusses this issue from several viewpoints. Geoffrion and Graves [10] were the pioneers in studying the two-stage distribution issue and used the Benders partitioning method to develop a solution. Pirkul and Jayaraman [20] proposed a novel mathematical formula known as PLANWAR for the locating of multiple

plants and storehouses and for designing a network to distribute the products in such a manner that the overall operational expenses could be significantly lowered. They solution was the development of a method on the basis of the Lagrangian relaxation. Heragu [12], on the other hand, looked at the tsTP and proposed a mathematical model as the solution. The model takes into consideration costs of inputs, outputs and transport and with the aim of reducing overall costs to the minimum. Heragu maintained that with this model there was no requirement to know or modify the facility capacity. Hindi *et al.* [13] focused on a two-step distribution planning problem, taking into consideration two extra criteria: first, each client must be served by a single CD; second, it is necessary to be able to determine the source plant of each product as well as the quantity delivered. The authors provided a mathematical model to address the problem and proposed a branch and a related algorithm as a solution for the problem. Tragantalerngsak *et al*. [28] dealt with the issue of localization of the two scales in which the first-level facilities were not captured while facilities at the second were. The objective was to confirm how many installation locations were in both steps to meet customer demand for the product. They proposed a branch-based Lagrangian derivation and an algorithm in relation to the solution of the problem. Others [2] examined a problem in the design of the distribution structure in the supply chain that involved the location of plants and warehouses in relation to meeting customer demand. In contrast to the findings in the literature, the author made allowance for a multiplicity of plants and storehouses, and proposed a model based on mathematics to address the issue.

In recent times, scalable techniques have been effectively used to address transportation issues. Michalewicz *et al.* [18[and Vignaux and Michalewicz [26] first mentioned using GA for the resolution if linear and non-linear transportation problems. In their work, whereas the matrix representation was employed to construct a chromosome, crossing and developing a matrix mutation. Another approach to the GA to solve the solid TP was proposed by Li *et al.* [15] who utilized the 3-D matrix to address the issue (mGA). Syarif and Gen [23] studied the production / distribution issue arising from tsTP and proposed a hybrid genetics algorithm. Even as a solution at this juncture was offered by the Prüfer number, a hybrid method was adopted to improve GA efficiency.

This current study used the tabu search algorithm to solve the tsTP on the basis of decoding procedure and in comparison with the genetic algorithm and simulated annealing algorithm.

2. Mathematical formulation

The purpose of the study was to establish the network for the purpose of distributing goods and services as requested by clients at the lowest possible cost relative to the capacity of the plant and the distribution centers as well as the least required number of distribution centers to open. We made the assumption that awareness of the clients, their locations and their requests were already available. Information on the possible number of distribution centers, where they are located and also their maximum capacities was also available. The mathematical model of the problem was::

$$\min z = \sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij}x_{ij} + \sum_{j=1}^{M}\sum_{k=1}^{K} c_{jk}y_{jk} + \sum_{j=1}^{M} g_j z_j \ldots (1)$$

$$s.t. \sum_{j=1}^{M} x_{ij} \le a_i, \quad \forall i \qquad \ldots (2)$$

$$\sum_{k=1}^{K} y_{jk} \le b_j z_j, \quad \forall j \qquad \ldots (3)$$

$$\sum_{j=1}^{M} z_j \le W, \qquad \ldots (4)$$

$$\sum_{j=1}^{M} y_{jk} \ge d_k, \quad \forall k \qquad \ldots (5)$$

$$\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} = \sum_{j=1}^{M}\sum_{k=1}^{K} y_{jk} \quad \ldots (6)$$

$$x_{ij}, y_{jk} \ge 0 \quad \forall\, i,j,k \quad \ldots (7)$$

$$z_j = \{0,1\} \quad \forall j \qquad \ldots (8)$$

where: N = the number of plants (i = 1,2, ..., N), M = the number of distribution centers, (j = 1,2, ..., M) = the number of customers (k = 1,2, ..., K), ai = the plant capacity i, bj = the capacity of DC j, d_k = the customer request k, t_{ij} = the unit transport cost of plant i at DC j, c_{jk}= the unit cost of distribution transport Center j to client k, g_j = the fixed cost for the DC of exploitation j, W = a high limit on the maximum distribution center number that can be set up, x_{ij} = the quantum of the factory i at DC r j, y_{jk} = the quantum of dispatch from DC j to the client variable k, z_j = 0-1, which takes the value 1 if DC j is open. Although limitations (2) and (3) guarantee the capacity constraints, of the DC, respectively, stress (4) and ensure the established distribution centers do not go beyond the maximum specified. This limitation is very significant in the case of a manager with modest capital resources. Limitation (5) ensures that all customer requests are met by established distribution centers; the limits (6) and (7) impose the non-satisfaction restriction of decision variables and binary nature of decision variables employed in this approach. By maintaining generality, we can make the assumption that this approach fulfills the balanced condition, because modification can be made to the unbalanced problem by including fictitious suppliers or fictitious customers.

3. Tabu search algorithm

The search algorithm was developed by researcher Glover in 1986 after thinking about rationalizing the search to a more comprehensive direction and concentration in a simple and effective way. The prohibited search algorithm was a development of the local search algorithm and was used to avoid the local optimization with the local search algorithm (LS). The idea of the Tabu search algorithm is to create a short-term memory that maps the solutions that have passed in the near future and not to resort to them in the coming stages and record the solutions that passed

through a special list called Tabu list. Designed in the size of the problem and the desire of the programmer, inputs are zero values at the beginning of the solution. We then register each solution to generate for each stage and after recording all the solutions that have passed us, we generate a new solution compared with previous solutions if it is more efficient than the bad solution recorded in the list of prohibitions. We are replacing it. The search for solutions continues until the number of duplicates has been completed. This technique has given higher possibility in the search for new solutions that are highly efficient outside the prohibition list. It should be noted that the process of generating new solutions is done in different formats according to the algorithm designer's desire to solve the problem. This is done according to the search feature about the neighborhood used in the local search algorithm or by the mutation used in the genetic algorithm.

Despite the effectiveness of this algorithm, there remains a problem. The verification of the solution in the block list is a big calculation because the block list registers complete solutions and comparing the new solutions with the previous solutions takes a long time, especially if the size of the problem is large. And the conversion of the block list from the registration of complete solutions to the registration of features or characteristics of the solutions passed by previously and this is certainly much better and the process is done with the introduction of more memory to block each memory which recorded a recipe for the solution that we generate if we assume that we want to solve the seller roaming problem (TSP) in the Tabu list

The first is that we visited the second city, for example, and in the other memory, the second type, verification is better and faster. The following are the basic steps to implement the Tabu search algorithm

.

Procedures of the basic Tabu search algorithm

- *Generate a specific solution*
- *Calculate the value of the target function of the current solution Z (S)*
- *Make S X and Z (S) Z (X)*
- *TL (Tabu list) → Ø*
- *Repeat the process*
- *Generate a new X 'solution by moving a certain neighborhood N (S) from the current solution that does not e*
- *Record the best solution found in TL and delete the old solution*
- *Calculate the value of the target function of the new solution Z (X ')*
- *If Z (S) Z (X ') > Make X' → S terminate the condition*
- *Repeat the process until the condition of stop and print the best solution was found*

End algorithm

Many components have been proposed to design a taboo search algorithm (TS) to make them more effective and efficient than these suggestions:

a) The memory allocated to the algorithm (possession of the block list) is of equal size, i.e., the memory entered to block an attribute is equal to the other memory
b) Blocking memory has variable dimensions, that is, it can be enlarged or minimized within the algorithm according to the direction of the solution
c) When intensifying the solution and access to good areas it is possible to minimize memory to reach fast solutions
d) This memory enables us to recall a large number of solutions that have been generated. This enables us to extract from all the solutions that have been generated in the past and to make certain of their characteristics in generating a new solution. This reminder is made according to four different modes, namely, quality and finally the effect, the incident is a memory-based registration for each solution or property of the last frequency which has been resolved, and frequency is a memory which records the number of times the same solution or grade and quality record the best solutions recorded in memory according to the efficiency of the objective function, the effect of the intensity of the solution, and the effect of the solution generated in the improvement.

It is clear that there are many ways and advantages in this algorithm which often approaches the optimal solution in a systematic manner but needs more time than some of the other algorithms, and also bans some solutions which are not use again, which reduces the value of diversification despite the modifications made by the introduction of long-term memory, Finally, the programmer can design this algorithm in a way that is appropriate to the data of the issue to be solved and the elimination of some properties that do not correspond to the nature of the problem .

Tabu search algorithm with multiple memory

- *Generate new solution*
- *Calculate the value of the objective function of the current solution Z (S)*
- *Make S → X and Z(S) → Z(X)*
- *TL1…TLn → Ø tabu list*
- *Repeat the process*
- *Generate a new X 'solution by moving a certain neighborhood N (S) from the current solution that is not pre*
- *Recording the best solution found in TL1…TLn Delete the old solution*
- *Calculate the value of the target function of the new solution Z (X ')*
- *If Z (S) Z (X ') > Make X' → S terminate the condition*
- *Repeat the process until the condition of stop and print the best solution was found*

End algorithm

4. decoding procedure

This section provides a description of the algorithm responsible for decoding a Predict and the number to a spanning tree, examine the locality, and the heritability of the Prüfer coding under traditional operators, and notes that area of research space where a preferred EA could work well is too small.

Let$a_1, a_2, \ldots a_{n-2} \in \{1, \ldots, n\}^{n-2}$ is a Prüfer number. In the corresponding spanning tree, the degree of each model exceeds the frequency of the node label's appearance. For the purpose of identifying the edges of the spanning tree:

1. Scan the Prüfer number to identify the nodes of each node degree. Initialize an i variable in 1.
2. Locate the node v of degree 1 with the smallest label.$(v;\ a_i)$ is an edge of the edge.
3. Describe the degrees of v and a_i; Increase i.
4. Repeat steps (2) and (3) until all nodes have a diploma 0, with the exception of two with degree 1.

The final edge of the tree. To efficiently implement this algorithm there is a need to uses a Priority queue placed in a heap to bind the nodes. The degree of complexity of the time of the algorithm is then O (n log n). Figure 1 presents a Prüfer number of length Four and the tree on six nodes to which it decodes. Numerous features of Prüfer numbers indicate that they could promote an effective evolutionary research of spaces

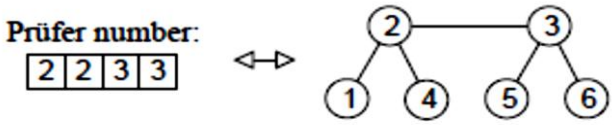

Figure 1: A spanning tree and its Prüfer number

A spanning tree GA (st-GA) is used to solve some network issues proposed by Gen and Cheng [who utilized the Prüfer number to present a solution possibility to the problems and developed the feasibility criteria for the Prüfer number for decoding into a spanning tree. They reported the use of a genetic algorithm for the tsTP utilizing a priority-based coding. Using the Prüfer number was very appropriate for coding a tree and certain areas of research, including TPs, the minimum spanning tree problems, etc. Moreover, it is demonstrated that there is only need for | K | + | J |-2 digits number for a unique representation of a distribution network with | K | sources and | J | deposits where each digit is an integer between 1 and | K | + | J |. Despite the Prüfer number, the decoding of the tree has been effectively used to address the TP (Gen & Li, 1998), but it requires a number of repair features.. As the transport shaft is a distinctive form of spanning shaft, the Prüfer number can be an infallible solution, that is, it does not represent a transport tree. From this perspective, it is necessary to develop a criterion to check the viability of chromosomes. However, if the chromosome fails in this check, it will be rectified and it is the first repair for a chromosome. A second repair may be necessary following the decoding of the chromosomes. After decoding the Prüfer number, certain links on its spanning tree can become zero flow, in which case these links are deleted from the tree and new trees are included to get a spanning tree. Following this settlement, the current Prüfer number ceases to represent the tree and it is necessary to code a Prüfer Number of the new spanning tree.

This present study avoids these rectification systems in the TS, using the decoding method created by Gen and Cheng [7],Despite this decoding being effectively used to short-path problems and project planning problems [8], the uniqueness of our approach lies in the special decoding and decoding method used for transport shafts.

.

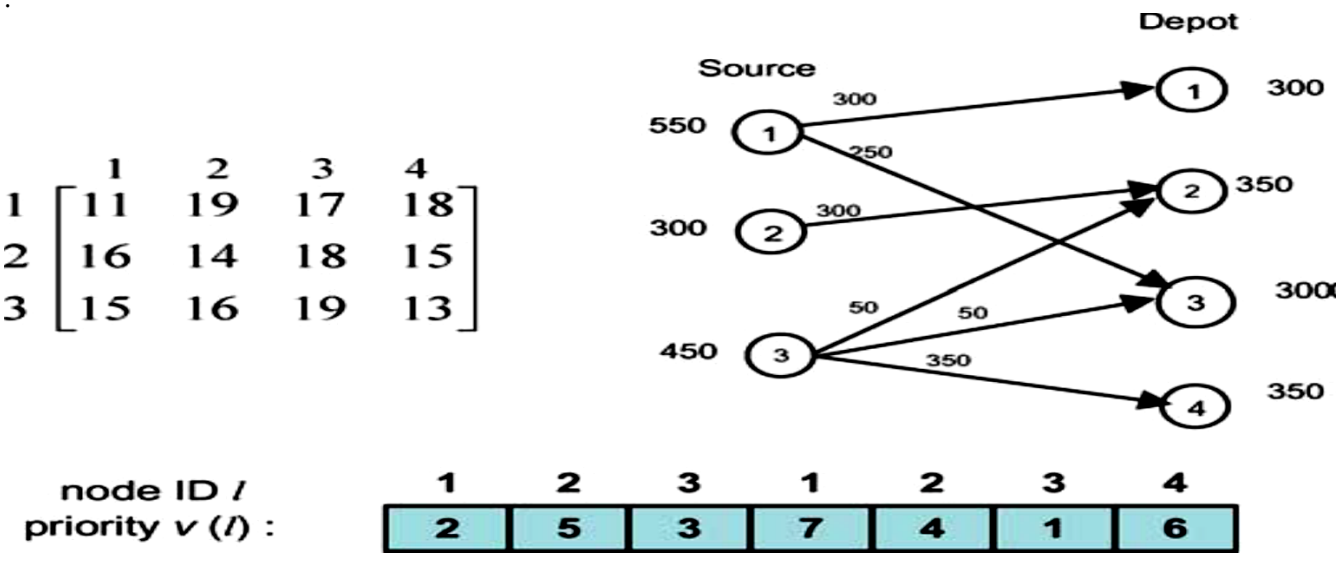

Fig. 2 Sample of transportation tree and its encoding

Decoding algorithm for transportation problem

Procedure : Decoding
Input
K: set of sources.
J: set of depots.
b_j: demand of depot i.
a_k: capacity of source i.
C_{kj} : Cost of transportation *for one unit of product from source k to depot j.*
$V(k+j)$: solution for transportation
Output
g_{kj} : The amount of shipment *of product from source k to depot j.*
Step 1:$g_{kj} \leftarrow 0 \ \forall \ k \in K \quad j \in J$
Step 2:$l \leftarrow$ argmax$\{V(t), t \in |\ K\ | + |\ J\ |\}$; select a $node$
Step 3:$if\ l \in K\ then\ k^* \leftarrow l\ ; select\ a\ source$
$j^* \leftarrow arg\ min\ \{C_{kj}|\ v(j), \neq 0\ , j \in J\}; select\ a\ depot\ with\ th\ lowst\ cost$
Else $j^* \leftarrow l\ ; select\ a\ depot$
$k^* \leftarrow arg\ min\ \{C_{kj}|\ v(j), \neq 0\ , k \in :\}; select\ a\ source\ with\ th\ lowst\ cost$
Step 4 :$g_{k^*j^*} \leftarrow min\{a_{k^*}, b_{j^*}\}; assign\ aviable\ amount\ of\ units$
Update availabilities on source (k^*) and depot (j^*)
$a_{k^*}=a_{k^*} - g_{k^*j^*} b_{j^*}=b_{j^*} - g_{k^*j^*}$
Step 5:$if\ a_{k^*} = 0\ then\ v(k^*) = 0$
$if\ b_{j^*} = 0\ then\ v(j^*) = 0$
Step 6:$if\ v(\quad |K\ | + j) = 0\ ,\ \forall j \in J$, then calculate transportation cost and return, else goto Step 1.

As is well known, a gene in a chromosome is typified by two factors: the locus (which is the position of the gene in the chromosome structure and the allele (which is the value taken by the gene). The position of a gene is utilized in representing a node (source / deposit) while the value is for the purpose of representing the priority of the node for the construction of a tree among the candidates. In case of a TP, a chromosome comprises source priorities and deposits to obtain a transport tree and its length matches the total number of sources (| K |) and deposits (| J |), that is | K | + | J |. The transport shaft in parallel with a given chromosome is produced by the addition of a sequential arc between the sources and Deposits. At every step, a single arc is introduced to the tree by the selection of a source (repository)

The top priority and the connection to a repository (source) taking into account the least cost shows a transport shaft comprising three sources and four deposits - cost matrix and priority-based coding. The priority decoding algorithm and its trace table are shown in Fig. 3 and Table 1, respectively. As mentioned in the initial step of the decoding procedure, an arc between deposition 1 and source 1 is added to the chromosome and the minimum cost is between source 1 and deposit 1. Following the determination of the quantity of the shipment that is g11 = min {550, 300} = 300, capacity, the source and the deposit requests are brought up to date as a1 = 550-300 = 250, b1 = 300-300 = 0, respectively. Since b1 = 0, the priority of Deposit 1 is set to 0 and Deposit 4 with the next. The top priority is chosen. Following the addition of the arc between Depot 4 and Source 3, the quantum of a shipment between them is established and their capacity brought up to date as mentioned above, and this process is self-repeating until the demands of all deposits are met.

Table 1 Trace table of decoding procedure

Iteration	$v(k+j)$	a	b	k	j	g_{kj}
0	[2 5 3 | 7 4 1 6]	(550, 300, 450)	(300, 350, 300, 350)	1	1	300
1	[2 5 3 | 0 4 1 6]	(250, 300, 450)	(0, 350, 300, 350)	3	4	350
2	[2 5 3 | 0 4 1 0]	(550, 300, 100)	(0, 350, 300, 0)	2	2	300
3	[2 0 3 | 0 4 1 0]	(550, 0, 100)	(0, 50, 300, 0)	3	2	50
4	[2 0 3 | 0 0 1 0]	(550, 0, 50)	(0, 0, 300, 0)	3	3	50
5	[2 0 0 | 0 0 1 0]	(550, 0, 0)	(0,0, 250, 0)	1	3	250
6	[0 0 0 | 0 0 0 0]	(0, 0, 0)	(0, 0, 0, 0)			

5. EXPERIMENTAL RESULTS

We generate different problems for tsTP in different sizes for N,M,K and solve each problem using the three algorithms in 10 runs for each problem and then calculate (min,max,mean) all algorithms programming by using MATLAB 2013a and for each problem size generate 10 problems, the results
of which are shown in Tables (2-1) to (2-9).

Table(2-1) Results of comparisons between the algorithms $N = 8$ $M = 10$ $K = 8$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	2486.8	2662	2401	2377	2377	2377	2624.2	3258	2403
2	5314.2	5722	4725	5343.2	5418	5314	5269.2	5753	4822
3	3422	3458	3375	3659.2	3831	3408	3482	3744	3371
4	3800.2	4236	3590	3588.8	3604	3573	3828.8	4303	3642
5	4562.4	4757	4448	4441.4	4451	4439	4727.8	4793	4578
6	4844.4	5280	4503	4646.8	4654	4618	4964.4	5347	4776
7	4197	4431	3816	4248.8	4368	4216	4085	4311	3741
8	3984.6	4156	3885	3889.6	3991	3856	3988.6	4254	3892
9	5380.2	5473	5314	5235	5318	5099	5447.8	5594	5293
10	6367.2	6446	6292	6349.2	6360	6306	6304	6517	6053
No. of. best	1	1	2	6	8	5	3	1	3

In the table (2-1) the TS algorithm shows the best results in six problems by using the min measure while the TS is outperformed by GA in problem 4 and outperforms SA in problems (2,3 and 10) and in max measure the TS algorithm shows the best result in 8 problems and outperforms GA in problem 3 and outperforms SA in 7, and in mean measure the TS algorithm shows the best result in 5 problems and outperforms GA and SA in 5 problems

.

Table(2-2) Result of comparison between the algorithms $N = 8$ $M = 12$ $K = 8$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	4591.6	4659	4459	4443.4	4459	4436	4558.6	4672	4468
2	2940.4	3088	2771	2785.4	2843	2762	2915	3071	2787
3	5777.2	5829	5648	5634	5658	5628	5771.8	6115	5552
4	4294.6	4352	4259	4173.4	4195	4127	4300.6	4385	4221
5	2816.6	3060	2699	2698	2698	2698	2992	3213	2765
6	5104.4	5288	4902	5086	5086	5086	5175.6	5281	5039
7	3755.2	3990	3653	3642.8	3654	3630	3804.2	4024	3652
8	5540.2	5643	5452	5468.2	5493	5369	5505.8	5643	5384
9	5151.8	5651	4956	4892.8	5273	4751	4949.8	4999	4800
10	5183	5341	4990	4870	4891	4861	5179.2	5315	4957
No. of. best	0	0	1	10	9	8	0	1	1

In Table (2-2) the TS algorithm shows the best result in all problems by using the min measure, in max measure the TS algorithm shows the best result in 9 problems and outperform SA in problem 9 ,and in mean measure the TS algorithm shown the best result in 8 problem and outperform GA in problem 6 and outperform SA in problem 3 .

In Table (2-3) the TS algorithm shows the best result in all problems by using the max outperforms GA in problem 5 and outperform SA in problem 6 ,and in mean measure TS.

Table(2-3) Result of comparison between the algorithms $N = 10$ $M = 14$ $K = 7$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	2185.2	2378	1863	2099.8	2229	1776	2162.4	2429	2075
2	3215	3407	3026	3058	3058	3058	3235.8	3494	3051
3	2931.8	3034	2824	2814.8	2883	2719	2834.6	3040	2394
4	3088.8	3267	2943	3065.6	3235	2856	3230.2	3315	3090
5	2840.8	2897	2788	2859	2887	2831	2883.8	2961	2811
6	3724.6	3911	3449	3807	3807	3807	3627.4	3831	3432
7	2033	2130	1931	1966.2	2046	1901	2058.4	2147	2005
8	2767.2	3138	2599	2550.6	2562	2543	2659.4	2840	2539
9	3485.2	3706	3153	3401	3401	3401	3577.4	3715	3472
10	2835.8	3034	2568	2784	2792	2772	2831	2932	2683
No. of. best	1	0	4	8	10	3	1	0	3

Table(2-4) Result of comparison between the algorithms $N = 12$ $M = 16$ $K = 10$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	4735	4966	4329	4390.4	4581	4275	4669.6	5128	4311
2	3599	3766	3500	3327.4	3353	3321	3871.6	4026	3691
3	3825.6	4119	3625	3677.4	3703	3669	3971	4080	3793
4	4650.8	4859	4519	4310.8	4342	4245	4619.8	4970	4194
5	3143.8	3219	3062	2913	2924	2901	3065	3123	2976
6	3198.8	3595	2965	2799.8	2815	2783	3123.8	3420	2847
7	4415	4593	4134	4124.4	4127	4123	4541.2	4649	4406
8	3755	3899	3567	3526.4	3537	3520	3978.4	4551	3639
9	3102.4	3756	2844	2693.8	2748	2541	3090.8	3209	2986
10	3703.6	3808	3518	3422.6	3480	3383	3624.2	4013	3265
No. of. best	0	0	1	10	10	7	0	0	2

.

Table(2-5) Result of comparison between the algorithms $N = 14$ $M = 12$ $K = 12$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	5080.8	5186	5014	4884.8	4908	4879	5057.8	5083	5012
2	6020.4	6161	5926	5636.6	5661	5617	6004.6	6206	5862
3	4101.8	4211	3864	3830.8	3833	3822	4068.2	4159	3991
4	4366.4	4439	4203	4123.4	4124	4121	4403.6	4517	4338
5	5247.6	5335	5140	5067	5069	5059	5267.2	5389	5183
6	5956.8	6091	5832	5608.2	5657	5590	5864.4	5913	5827
7	5876.6	6110	5701	5613.4	5634	5592	5756.2	5962	5670
8	4468.6	4517	4420	4246.6	4247	4246	4583.6	4785	4448
9	5249	5427	5046	4977	5049	4959	5291.2	5381	5213
10	5692.2	5841	5414	5564	5568	5558	5675.2	5884	5465
No. of. best	0	0	1	10	10	9	0	0	0

Table (2-6) Result of comparison between the algorithms $N = 12$ $M = 16$ $K = 16$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	6112.2	6366	5463	5707.4	5804	5508	5979	6315	5384
2	5340.4	5510	5263	5001.4	5027	4989	5380	5676	5219
3	5783.8	5845	5753	5409.2	5446	5350	5896.4	6065	5757
4	5795.6	5961	5664	5549.2	5615	5521	5964.4	6103	5831
5	4958.4	5279	4788	4668.8	4690	4638	5022.6	5347	4928
6	6526.8	6672	6404	5929.2	5944	5922	6503.8	6695	6346
7	6159.6	6306	6018	5782.4	5784	5776	6154.6	6327	6015
8	6166.4	6424	5979	5735.2	5815	5694	6079	6295	5870
9	6226.4	6323	6125	5613.8	5677	5579	6164.4	6368	5840
10	5309.6	5372	5175	4911.8	4957	4879	5329	5440	5222
No. of. best	0	0	0	10	10	9	0	0	1

Table (2-7) Result of comparison between the algorithms $N = 12$ $M = 14$ $K = 14$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	6122	6282	5993	5732.2	5761	5717	6151.2	6509	5999
2	4946.8	5121	4789	4686	4690	4685	4879.4	4969	4759
3	5915	5949	5875	5712	5728	5703	5967.6	6019	5919
4	6055	6199	5855	5741.6	5897	5604	5980.4	6234	5789
5	5176.4	5299	4976	4858.8	4882	4842	5203.4	5338	5068
6	6469	6681	6351	6103	6198	5987	6350.6	6682	6223
7	5212.6	5370	5098	5055.6	5109	4989	5150.8	5305	5054
8	6163	6257	6086	5973.2	5990	5945	6252.2	6341	6104
9	4982.4	5045	4905	4825.6	4855	4806	5044.4	5092	5011
10	5546	5631	5385	5259.8	5355	5230	5639.2	5911	5495
No. of. best	0	0	0	10	10	10	0	0	0

Table(2-8) Result of comparison between the algorithms $N = 14$ $M = 18$ $K = 20$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	6122	6282	5993	5732.2	5761	5717	6151.2	6509	5999
2	4946.8	5121	4789	4686	4690	4685	4879.4	4969	4759
3	5915	5949	5875	5712	5728	5703	5967.6	6019	5919
4	6055	6199	5855	5741.6	5897	5604	5980.4	6234	5789
5	5176.4	5299	4976	4858.8	4882	4842	5203.4	5338	5068
6	6469	6681	6351	6103	6198	5987	6350.6	6682	6223
7	5212.6	5370	5098	5055.6	5109	4989	5150.8	5305	5054
8	6163	6257	6086	5973.2	5990	5945	6252.2	6341	6104
9	4982.4	5045	4905	4825.6	4855	4806	5044.4	5092	5011
10	5546	5631	5385	5259.8	5355	5230	5639.2	5911	5495
No. of. best	0	0	0	10	10	10	0	0	0

Table(2-9) Result of Comparative between the algorithms $N = 20$ $M = 20$ $K = 20$

Ex	Genetic algorithm			Tabu search algorithm			Simulated Annealing		
	min	max	mean	min	max	mean	min	max	mean
1	6657.8	6849	6523	5971.8	6030	5937	6744	6938	6459
2	8177.4	8533	7903	7290.6	7378	7237	8189.2	8305	7909
3	7775.4	8143	7416	7228.6	7300	7158	7706.4	8070	7420
4	7700.2	7940	7562	6911.2	6960	6864	7584.6	7773	7340
5	7299.2	7565	7102	6660.6	6718	6629	7443.6	7538	7290
6	6412.8	6532	6303	5654.6	5775	5579	6289	6452	6168
7	6145.8	6236	5988	5469.6	5529	5426	6211	6337	6145
8	6500.6	6596	6402	6018.4	6051	5919	6523.6	6710	6409
9	6302.4	6445	6170	5637	5676	5594	6296.4	6488	6053
10	8030.8	8314	7756	6994	7076	6885	7775	7980	7579
No. of. best	0	0	0	10	10	10	0	0	0

In Table (2-4) the TS algorithm shows the best result in all problems by using the min and max measures, and in mean measure TS algorithm shows the best result in 7 problems and outperforms in 3 problems

In Table (2-5) the TS algorithm shows the best result in all problems by using the min and max measures, and in mean measure TS algorithm shows the best result in 9 problems and outperforms in one problem.

In Table (2-6) the TS algorithm shows the best result in all problems by using the min and max measures, and in mean measure TS algorithm shows the best result in 9 problems and outperform in one problem.

In Table (2-7) the TS algorithm shows the best result in all problems by using the min, max and mean measures.

In Table (2-8) the TS algorithm shows the best result in all problems by using the min , max and mean measures.

In Table (2-9) the TS algorithm shows the best result in all problems by using the min , max and mean measures.

6. CONCLUSION

In this research, the tabu search algorithm was used to solve the two-stage transportation problem by using a special method for coding and decoding and then generating problems of different sizes and then comparing the results with the simulated annealing algorithm and the genetic algorithm. The arithmetic result showed the superiority of the tabu search algorithm over the rest of the algorithms in almost all problems .

It was demonstrated that there is a need to use decoding algorithms with complex mathematical models with numerical algorithm to achieve good solutions in fast time.

REFERENCES

[1] Aiken. CH (1985) "Facility location models for distribution planning", Eur J OperResv22:263–279.

[2] Amiri A (2005) "Designing a distribution network in a supply chain system: formulation and efficient solution procedure", European Journal of Operational Research Vo. 171, Issue 2, 1 June 2006, Pages 567-576.

[3] Avealla P (1998) "Some personal views on the current state and the future of locational analysis" .Eur J Oper Res 104:269–287.

[4] Das. C, Heragu S (1988) " A transportation approach to locating plants in relation to potential markets and raw material sources " . Decis Sci 19(4):819–829

[5] Davis. L (1995) " Job-shop scheduling with genetic algorithms" Proceedings of the First international Conference on Genetic Algorithms. Lawrence Erlbaum Associates, pp 36–140.

[6] Garey MR, Johnson DS (1979) " Computers and intractability: a guide to the theory of NP – Completeness ". Freeman.

[7] Gen M, Cheng RW (1997)" Genetic algorithms and engineering design". Wiley, New York , ISBN: 978-0-471-12741-3 , 432 pages.

[8] Gen M, Cheng RW (2000) "Genetic algorithms and engineering optimization". Wiley, New York

[9] Gen M, Li YZ (1998) Solving multi-objective transportation problem by spanning tree-basegenetic algorithm. In: Adaptive computing in design and manufactured. Springer, BerlinHeidelberg New York, pp 98–108

[10] Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by bendersdecomposition. Manage Sci 20:822–844

[11] Goldberg D, Lingle R (1995) Alleles, Loci and the traveling salesman problem. Proceedings ofthe First International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, pp154–159

[12] Heragu S (1997) Facilities design. PSW

[13] Hindi KS, Basta T, Pienkosz K (1998) Efficient solution of a multi-commodity, two-stagedistribution problem with constraints on assignment of customers to distribution centers. IntTrans Oper Res 5(6):519–527

[14] Hitchcock FL (1941) The distribution of a product from several sources to numerous localities.J Math Phys 20:24–230

[15] Li YZ, Gen M, Ida K (1998) Improved genetic algorithm for solving multi-objective solidtransportation problem with fuzzy number. Jpn J Fuzzy Theory Syst 4(3):220 -229

[16] Malhotra, S., Malhotra, R., Puri, M.C.: Two Stage Interval Time Minimizing TransportationProblem. ASOR Bulletin. 23, 2–14 (2004)

[17] Malhotra, S., Malhotra, R.: A polynomial Algorithm for a Two – Stage Time MinimizingTransportation Problem. OPSEARCH 39, 251–266 (2002)

[18] Michalewicz Z, Vignaux GA, Hobbs M (1991) A non-standard genetic algorithm for thenonlinear transportation problem. ORSA J Comput 3(4):307–316

[19] NagoorGani, A., Abdul Razak, K.: Two Stage Fuzzy Transportation Problem. Journal ofPhysical Sciences 10, 63–69 (2006)

[20] Pirkul H, Jayaraman V (1998) A multi-commodity, multi-plant capacitated facility locationproblem: formulation and efficient heuristic solution. Computer Operations Research 25(10):869–878

[21] ReVelle C, Laporte G (1996) The plant location problem: new models and research prospects.Oper Res 44:864–874

[22] Ritha, W., MerlineVinotha, J.: Multiobjective Two Stage Fuzzy Transportation Problem.Journal of Physical Sciences 13, 107–120 (2009)

[23] Syarif A, Gen M (2003) Double spanning tree-based genetic algorithm for two stagetransportation problem. Int J Knowl-Based IntellEngSyst 7(4): October

[24] Syswerda G (1995) Uniform crossover in genetic algorithms. Proceedings of the ThirdInternational Conference on Genetic Algorithms. Lawrence Erlbaum Associates, pp 2–9

[25] Tilanus B (1997) Introduction to information system in logistics and transportation. In: Tilanus B(ed) Information systems in logistics and transportation. Elsevier, Amsterdam, pp 7–16

[26] Vignaux GA, Michalewicz Z (1991) ," A genetic algorithm for the linear transportation problem. IEEE Trans Syst Man Cybern " 21(2):445–452.

[27] Xie . Fanrong , Jiab . Renan " Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm" Volume 63, Issue 4, December 2012, Pages 763-778 .

[28] Tragantalerngsak, Suda, John Holt, and Mikael Rönnqvist. "An exact method for the two-echelon, single-source, capacitated facility location problem." *European Journal of Operational Research* 123.3 (2000): 473-489.