Determine the Optimal Sequence-Dependent Completion Times for Multiple Demand with Multi-Products Using Genetic Algorithm

Watheq H. Laith[1*] Swsan S.Abed Ali[2] Mahmoud A. Mahmoud [2]

1. Dep. of statistical, College of Adm. and Econ., University of sumer, Al-Refaee , Thi-qar , Iraq.
2. School of Production and Metrology Eng .Dep. of Industrial Engineering, University of Technology , Baghdad , Iraq.

Abstract
Sequencing is the most impact factor on the total completion time , the products sequences inside demands that consist from muti-product and for multiple demands . It is very important in assembly line and batch production . The most important drawback of existing methods used to solve the sequencing problems is the sequence must has a few products and dependent completion time for single demand . In this paper we used genetic algorithm –based Travelling Salesman Problem with Precedence Constraints Approach (TSPPCA) to minimize completion time . The main advantage of this new method , it is used to solve the sequencing problems for multiple demand with multi-product

In this paper , we compare between modify the assignment method (MAM) and genetic algorithm depend on least completion time , the results discern that GA has minimum completion time

Keywords: *products sequences , completion time , travel salesman problem (TSP) ,* TSPPCA , genetic algorithm. .

1. Introduction

The TSP was documented as early as 1759 by Euler, whose interest was in solving the knight's tour problem (Lawler 1986). A correct solution would have a knight visit each of the 64 squares of a chess-board exactly once on its tour.

George Danzig, Ray Fulkerson and Selmer Johnson in their paper "Solution of a large-scale travelling salesman problem" proposed a novel method for solving instances of the TSP using linear programming (Lawler 1986 ; 2- Dantzig 1954) . They used this technique to solve a problem containing 49 cities in the USA. Danzig, while working at the RAND Corporation, developed a technique to optimize solutions for combinatorial problems called the Simplex Algorithm. This algorithm was refined and later named the cutting-plane method. The cutting-plane method has been successfully applied to a wide range of problems in the combinatorial field (Applegate 2003) . During the 1960's the cutting plane method was adapted by Land and Doig to form the branch-and-bound searching technique. The branch-and-bound technique was applied to the TSP by Little *et al.* in 1963 (Lawler 1986). The RAND Corporation's reputation helped to make the TSP a well known and popular problem. The TSP also became popular at that time due to the new subject of linear programming and attempts to solve combinatorial problems (Schrijver 1998).

Since the late 1980's the Centre for Research on Parallel Computation (CRPC) at Rice University has examined the travelling salesman problem. David Applegate, Robert Bixby and William Cook have examined a number of very large scale TSP problems. The problems evaluated were TSP problems in the region of 3000 – 15000 cities and were evaluated on super computers and large parallel computer systems. The technique that was implemented was the cutting-plane method (Applegate 2003).

The study on TSPPC is interesting as their concept can be applied to solve many scheduling and routing problems both in manufacturing and service industry. In the manufacturing industry, the problems are mostly dealing with products sequencing which arises as a subproblem in scheduling , routing and process planning. The products sequencing problem may be regarded as a generalization of the TSP with precedence constraints. TSPPC is harder to solve than the general TSP because the model formulations are complex and the algorithm for solving these models are difficult to implement . Since the TSPPC belongs to the class of NP-hard problem, the optimal solution to the problem cannot be obtained within a reasonable computational time for large size instance (Moon 2000). For an algorithm that cannot execute in polynomial time the term Non-deterministic Polynomial time is used, meaning that the execution time needed by the algorithm is not a polynomial function of the problem size (Garey & Johnson 2003 *).* There are many manufacturing optimization problems that are NP-hard, including vehicle routing problems, bin packing problems and scheduling problems (Stutzle, 1998) .

TSPPC is difficult to solve efficiently by conventional optimization techniques when its scale is very large. The earliest research on TSPPC problem was solved using exact methods such as branch-and-bound and dynamic programming. However the exact methods that guarantee to find the optimal solution of the problem are only capable of handling small and medium size of instances [Farber & Coves 2005). In addition, the size of the instances that are practically solvable is rather limited and the computational time increases rapidly with the instance size. The memory consumption of exact algorithms can also be very large and lead to the early termination of a program. Therefore it is necessary to develop more efficient algorithms for solving TSPPC problems.

Approximate or heuristic methods such as genetic algorithms, tabu search and simulated annealing do not guarantee the optimal solution [Rekiek & Delchambre 2006) , but empirically they have often been shown to return high quality solutions in short computation time. The genetic algorithm (GA) has powerful performance for solving combinatorial optimization problems, especially for sequencing problems such as TSP and flow shop scheduling (Chatterjee*, et al*1996 *;* Mitchell *2007 ;*Bryant 2000 *).* However, the use of conventional GA procedure for TSP with order-based representation might generate invalid candidate solution when applying to TSPPC problem. The infeasible sequences might be produced after crossover and mutation operations. A method to handle precedence constraints should be integrated in the GA procedure in order to generate only feasible solution during the evolutionary process. Hence, a study to develop an efficient genetic algorithm to obtain feasible and optimal solution of TSPPC is needed.

2. Solving multiple Demand Multi-products problem

In this paper ,we solving multiple demand multi-products problem with modified assignment method (MAM) to solve *travel salesman problem (TSP)* and multi-stages where all demand start with last product from previous demand .

The second method genetic algorithm (GA) to solve Travelling Salesman Problem with Precedence Constraints Approach (TSPPCA) .

2.1 Assignment method and travelling salesman problem (TSP)

An assignment problem is a particular case of transportation problem where the objective is to assign a number of resources to an equal number of activities so as to minimize total cost or maximize total profit of allocation.

The problem of assignment arises because available resources such as men, machines, etc. have varying degrees of efficiency for performing different activities. Therefore, cost, profit or time of performing the different activities is different. Thus, the problem is how the assignments should be made so as to optimize the given objective. The assignment problem is one of the fundamental combinatorial optimization problems in the branch of optimization or operations research in Mathematics (Gaglani 2011) .

Then, the mathematical model of the assignment problem can be stated as:

The objective function is to,

$$Minimize = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}X_{ij} \qquad (1)$$

Subject to the constraints

$$\sum_{j=1}^{N} X_{ij} = 1 \ , \quad for\ all\ i\ (\ resourxe\ availability\ ) \qquad (2)$$

$$\sum_{i=1}^{N} X_{ij} = 1 \ , \quad for\ all\ j\ (\ activity\ requirement\ ) \qquad (3)$$

Where:

X_{ij}: assignment variable $= \begin{cases} 1 & when \text{ assignment} \\ 0 & Otherwise \end{cases}$

cij : represents the cost of assignment of resource i to activity j.

The assignment problem is a variation of the transportation problem with two characteristics: (i) the cost matrix is a square matrix, and (ii) the optimal solution for the problem would always be such that there would be only one assignment in a given row or column of the cost matrix (Lawler *, et al.* 1986) .

In this paper, we have applied new method of an assignment problem for solving Travelling salesman problem where it is shown that this method also gives optimal solution. However the technique for solving Travelling Salesman problem using our method is more simple and easy for the optimal solution.

The travelling salesman problem is one of the problems considered as puzzles by the mathematicians.

Suppose a salesman wants to visit a certain number of cities allotted to him.

He knows the distances (or cost or time) of journey between every pair of cities, usually denoted by cij i.e. city i to city j.

His problem is to select such a route that stars from his home city. Passes through each city once and only once and returns to his home city in the shortest possible distance (or at the least cost or in least time). (Keller 2004).

Figure 1 shows an example of the simple TSP diagram , corresponds to the cost(or time) of the edge joining node *i* to node *j* . The integer programming formulation for TSP is given as follows (Ji & Ho 2005) ;

$$Minimize = \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} C_{ij}X_{ij} \qquad (4)$$

Subject to :

$\sum_{i=1}^{N} X_{ij} \leq 1\ ,\ for\ j = 1,2, \dots \dots, N \quad i \neq j$ (5)

$$\sum_{j=1}^{N} X_{ij} \leq 1\ ,\ for\ i = 1,2, \dots \dots, N \quad i \neq j \qquad (6)$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} X_{ij} = N - 1\ ,\ for\ j = 1,2, \dots \dots, N\ ,\ i = 1,2, \dots \dots, N \qquad (7)$$

Where:

X_{ij}: variable $= \begin{cases} 1 & when \text{ Passes salesman travels } from\ city\ i\ to\ city\ j \\ 0 & Otherwise \end{cases}$

cij : denoted to The distance (or cost or time).

The objective function (4) is simply to minimize the total cost or distance travelled in a tour. Constraint set (5), (6) and (7) ensures, respectively that the salesman enter and leaves each city exactly once (Ji & Ho, 2005) .

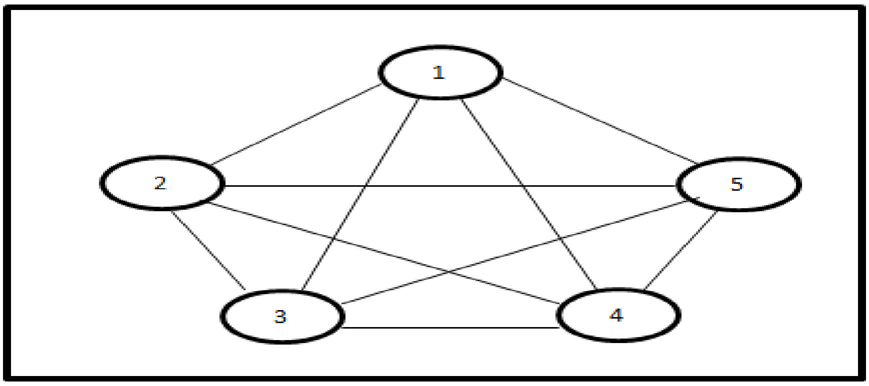


Figure 1. Graphic for Five City for TSP .

2.2. Modified Assignment Method (MAM) :

The main idea of the proposed algorithm (modified assignment method) is to convert TSP from closed loop (return to the original point) to open loop (no need to return to the original point) to agree sequence products problem with single demand multiple products depend on completion time .

The terms of the application of (MAM)algorithm as following :

1. Number products manufactured within a production line are known and specified.
2. Setup time and/or setup cost between products an known .
3- Demands consist from one or more products .
4 - products within the same Demand not to be repeated .
5- A product is manufactured within single batch .

The table 1 represents the assignment matrix **.**

Table 1. Modified Assignment Method

From product i	To product j								
	Products / Products	A		B		…….		N	
		Cost	Time	Cost	Time	Cost	Time	Cost	Time
	A	C_{11}	t_{11}	C_{12}	t_{12}	C_{1j}	t_{1j}	C_{1N}	t_{1N}
	B	C_{21}	t_{21}	C_{22}	t_{22}	C_{2j}	t_{2j}	C_{2N}	t_{2N}
	….	C_{i1}	t_{i1}	C_{i2}	t_{i2}	C_{ij}	t_{ij}	C_{iN}	t_{iN}
	N	C_{N1}	t_{N1}	C_{N2}	t_{N2}	C_{Nj}	t_{N}	C_{NN}	t_{NN}

2.2.1.The modified assignment method algorithm :

The algorithm can be described as following :

1. Configure assignment table, which rows and columns represent products manufactured within the production line , as shown in table 1 .

2 . Set of intersection point of the same products an infinite amount , so as to be ruled out in the assignment

process and the rest are placed according to setup time and/or setup cost of products , as shown in table 2 .

Table 2. Modified Assignment Method to TSP

From product i	To product j								
	Products \ Products	A		B		....		N	
		Cost	Time	Cost	Time	Cost	Time	Cost	Time
	A	∞	∞	C_{12}	t_{12}	C_{1j}	t_{1j}	C_{1N}	t_{1N}
	B	C_{21}	t_{21}	∞	∞	C_{2j}	t_{2j}	C_{2N}	t_{2N}
	....	C_{i1}	t_{i1}	C_{i2}	t_{i2}	∞	∞	C_{iN}	t_{iN}
	N	C_{N1}	t_{N1}	C_{N2}	t_{N2}	C_{Nj}	t_N	∞	∞

3. Each row and each column in the modified assignment matrix is smaller than or equal to one as described equations (8 and 9) ; except mentioned in step 9 .

$$\sum_{j=1}^{N} X_{ij} \leq 1\ ,\ \ for\ i = 1,2,\dots\dots,N \quad , \qquad (8)$$

$$\sum_{i=1}^{N} X_{ij} \leq 1\ ,\ \ for\ j = 1,2,\dots\dots,N \quad , \qquad (9)$$

4. If there is a transaction between two products such that $A \rightarrow B$, $B \rightarrow A$; then select one transaction ,i.e $X_{ij} + X_{ji} \leq 1$.

5. Number of transaction constrain as described equation (10) .

$$\sum_{i=1}^{N}\sum_{j=1}^{N} X_{ij} = N \qquad (10)$$

6.Intersection point between the same products clown and row which ended with previous demand set to be zero if this product found within the demand .Else set to be infinity (or large amount) . assume product A which ended with previous demand will be change table 2 to table 3 .

7.The column contains the intersection point in step 7 set to be infinity for other products , as shown in table 3 .

Table 3. Modified Assignment Method to TSP after Applied Step 6 and 7

From product i	To product j								
	Products \ Products	A		B		....		N	
		Cost	Time	Cost	Time	Cost	Time	Cost	Time
	A	0	0	C_{12}	t_{12}	C_{1j}	t_{1j}	C_{1N}	t_{1N}
	B	∞	∞	∞	∞	C_{2j}	t_{2j}	C_{2N}	t_{2N}
	....	∞	∞	C_{i2}	t_{i2}	∞	∞	C_{iN}	t_{iN}
	N	∞	∞	C_{N2}	t_{N2}	C_{Nj}	t_N	∞	∞

8. Row constraint for the product which is ended the previous demand will be equal to :

a) 2 If the product within the demand .

b) 1 If not within the demand . As described in equation (11) .

$$\sum_{j=1}^{N} X_{ij} = \begin{cases} 1 \text{ if product which it ended the previous demand not within demand} \\ 2 \text{ if product which it ended the previous demand within demand} \end{cases} \qquad (11)$$

9. If the Product in the assignment matrix has no demand then its constrain equal to be zero except for the product which ended with previous demand .

10. The mathematical model of the modified assignment method can be solved by linear programming method as:

$$X_{ij}\text{: assignment variable} = \begin{cases} 1 \ when \text{ Passes salesman travels } from\ city\ i\ to\ city\ j \\ 0 \qquad Otherwise \end{cases}$$

tij : represents the time of passes salesman travels $from\ city\ i\ to\ city\ j$.

2.3. Travelling Salesman Problem with Precedence Constraints (TSPPC) :

Basic TSP has neither constraint nor priority given to any cities. The TSP with precedence constraints (TSPPC) is one in which a set of *n* nodes and distances for each pair of nodes are given, the problem is to find a tour from node 1 to node *n* of minimal length which takes given precedence constraints into account. In TSPPC some order of cities is given and we ought to visit cities in that order only. TSPPC differs from traditional TSP whereby in TSPPC, there is no need to return to the original city. TSPPC becomes more important because in real life problems we always have to follow some orders. An example of TSPPC is shown in figure 2 . Each precedence constraint requires that some node *i* have to be visited before some other node *j* (Kotecha & Gambhava 2003). In a directed graph, the vertices (circles) represent activities or tasks and the edges represent the precedence relations between activities (Moon 2000). The task dependencies deal with the relationships between giving tasks and how they affect each other. The four types of task dependencies are Finish-to-start in which predecessor task must be finished before the successor can start, Start-to-finish in which successor task can finish only after the predecessor task has started, Start-to-start in which two tasks can start simultaneously and Finish-to-finish in which two tasks must finish at the same time (Wysocki 2009). The TSPPC in this study is classified as Finish-to-start types of task dependencies .

The term of the travelling salesman problem with precedence constraints (TSPPC) was formerly used by Kusiak and Finke in 1987 to solve machine scheduling problem using the exact method. In 2002, Moon *et al.* introduced more efficient method to solve TSPPC instances. In some researches, other terms are also used to represent TSPPC problem such as precedence constraint routing problem (Psaraftis 1983) precedence constraint travelling salesman problem (PCTSP) (Kotecha & Gambhava 2003) , the TSPPC can be stated as the problem of finding a job sequence subject to the precedence constraints which minimize the total makespan . There is therefore equivalent to the problem of finding a feasible Hamiltonian path with minimal cost under precedence constraints given by set *R* (Gambardella & Dorigo1997) .

Moon has used the two-commodity network flow model to formulate TSPPC. In this formulation, c_{ij} is the travel distance from vertex v_i to v_j and *s* is the first selected vertex in the graph. Commodity *p* is supplied by (n -1) units at a selected starting node and used by one unit at each node that is not the starting node while commodity *q* is consumed by (n -1) units at the starting node and supplied by one unit at the other nodes. Here *n* is the number of nodes or cities (Moon 2000) .

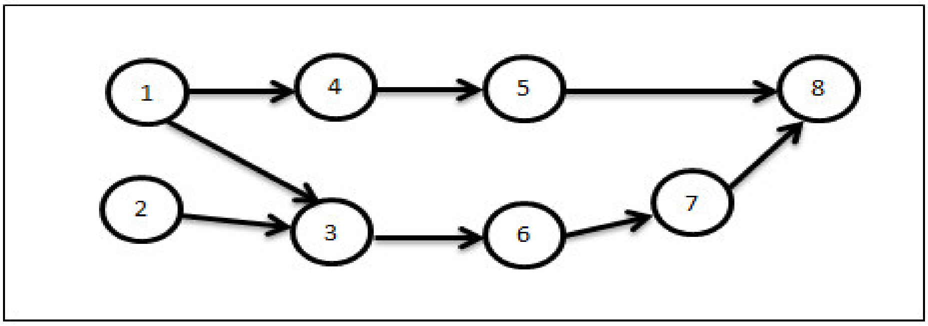


Figure 2. Example of TSP With Precedence Relationships

2.4. Genetic Algorithms

The GA operation is based on the Darwinian principle of survival of the fittest and it implies that the 'fitter' individuals are more likely to survive and have a greater chance of passing their 'good' genetic features to the next generation (Holland 1975 ; Katayama & Sakamoto 2000). Figure (3) illustrates the basic operation of GA while the general procedure is given in figure 4.

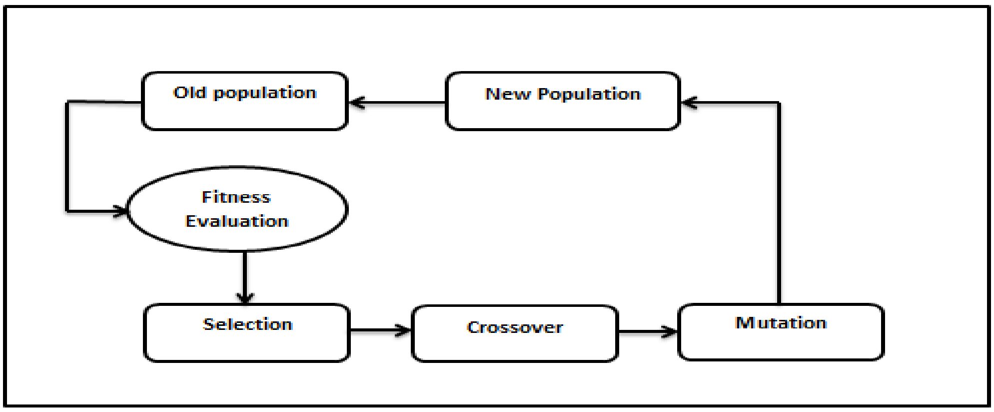


Figure 3. Basic Operation of Genetic Algorithm

In the standard or basic procedure of GA [Katayama & Sakamoto 2000 ; Larranaga & Kuijpers 1999) , an initial population is created containing a predefined number of individuals (i.e. solutions). Each individual has an associated fitness measure, typically representing an objective value. The concept that fittest (or best) individuals in a population will produce fitter offspring is then implemented in order to reproduce the next population. Selected individuals are chosen for reproduction (by crossover and mutation) at each generation, with an appropriate crossover and mutation factor to randomly modify the genes of an individual. The algorithm identifies the individuals with the optimising fitness values, and those with lower fitness will naturally get discarded from the population. Once crossover and mutation is done, a new generation is formed and the process is repeated until some stopping criteria have been reached (Youssef, *et al.* 2001) . A comprehensive explanation on the genetic algorithm operation can also be referred to Netnevitsky (Netnevitsky 2002) .

```
Begin GA
  Initialize population, P
  For generation_count < k do
   Evaluate P (i)
   Begin
         Select parents from the population
         Produce children by crossover from selected parents
         Mutate the individuals
         Increment generation_count
    End
    Return the best solution
End GA
```

Figure 4. General Procedure of Genetic Algorithm

3. Numerical Example

The proposed algorithm applied in numerical example where line production can be manufactured five products (A , B , C , D , E) and product (D) ended the previous demand , this example contain nine of demands as table 4, all demand contain on different number from products and every product has different process time as table 5 , and different quantity in all demand as table 3 , Setup time as show in table 6 and goals of company are optimum sequence of all demand products with minimum completion time that calculated as :

completion time (t_c)= Process time (t_p) +Setup time (t_s) .(13).

Table 4. Sequence of Demands Depend on EDD

No.of demands	Demand products	Quantity of product
1	A , B , C	100, 60 , 50
2	D	200
3	A , B , C , D , E	100 , 200 , 300 , 250 , 150
4	E	50
5	D , B	25 , 100
6	D , E , B	50 , 100 , 150
7	A , B , D	200 , 300 , 75
8	D	200
9	A , E	100 , 300

Table 5. Data of Products

products	Process time / unit
A	2
B	1
C	3
D	4
E	5

Table 6. Setup Time Matrix

	To product j					
From product i	Products \ Products	A	B	C	D	E
	A	0	4	6	20	20
	B	5	0	5	5	30
	C	7	6	0	10	5
	D	14	10	6	0	10
	E	15	2	12	25	0

The research include two methods to solve this problem to compare results and select the best method that give optimum sequence with minimum completion time as following :

1- modified assignment method (MAM) to convert TSP of closed-loop (return to the original point) to open loop (no need to return to the point of origin) to approve the problem of infiltration of products sequence to single demand with several products depending on the time of completion and repeat this several times approach to solve the problem multiple demands .

2- Genetic algorithm (GA) to solve Travelling Salesman Problem with Precedence Constraints Approach (TSPPCA) , where consist from two matrices as :

A) The first matrix is the matrix represent the constraints that the constraint relationship between the products demands rely on earliest due date (EDD) or any others rule and the first column described a number of products to be finding sequencing with added to the product which ended with previous demand as figure 5 .

```
prec_data1 =[ 1 0 0 0 0 0 0 0
              2 0 1 0 0 0 0 0
              3 0 1 0 0 0 0 0
              4 0 1 0 0 0 0 0
              5 0 2 3 4 0 0 0
              6 0 5 0 0 0 0 0
              7 0 5 0 0 0 0 0
              8 0 5 0 0 0 0 0
              9 0 5 0 0 0 0 0
             10 0 5 0 0 0 0 0
             11 0 6 7 8 9 10 0
             12 0 11 0 0 0 0 0
             13 0 11 0 0 0 0 0
             14 0 12 13 0 0 0 0
             15 0 12 13 0 0 0 0
             16 0 12 13 0 0 0 0
             17 0 14 15 16 0 0 0
             18 0 14 15 16 0 0 0
             19 0 14 15 16 0 0 0
             20 0 17 18 19 0 0 0
             21 0 20 0 0 0 0 0
             22 0 20 0 0 0 0 0 ];
```

Figure 5. Constraints matrix

B- Second matrix is completion time matrix as show in figure 6 , that contain rows and columns are describe demands products . This matrix contain three groups from numbers as:

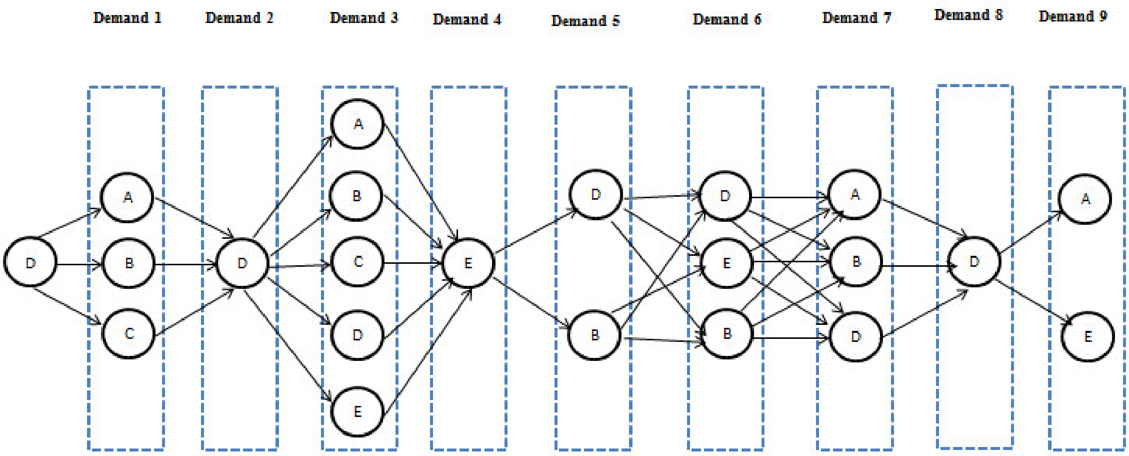
- A set of numbers all zeros, which offers no relationship between products based on the constraints matrix (as shown in the first matrix).

Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.5, No.8, 2015

- A set of numbers shown in the colored area , where is From –to matrix of all the demands that depend on the completion time, that present correlation between products within all demand depend on constraints matrix and put on configuration rectangles in figure 7 .
- A set of numbers in the region is not colored, which represents the possibility of a link between the demands put on configuration shares in figure7 .

```
trans_time =[ 100000 214 70 156 800 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
              100000 200 64 156 820 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
              100000 205 60 155 805 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
              100000 207 66 150 810 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 800 214 210 906 1000 765 0 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 200 204 906 1015 770 270 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 205 200 905 1020 780 280 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 207 206 900 1010 755 255 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 214 210 906 1000 760 260 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 215 202 912 1025 750 250 0 0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 0 0 250 125 102 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 100 110 220 520 164 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 105 100 205 530 150 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 200 510 160 414 310 300 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 225 500 152 415 302 325 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 205 530 150 408 300 305 0 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 400 304 320 820 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 405 300 305 805 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 414 310 300 810 0 0
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 800 214 1510
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 200 1520
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 215 1500];
```

Figure 6. Completion Time Matrix



Method	Optimum Sequence For multi Demands Multiple Products	Total Completion Time
MAM	D-C-A-B-D-D-C-E-B-A-E-B-D-D-E-B-B-D-A-D-E-A	9203
GA	D-C-A-B-D-D-B-A-C-E-E-B-D-E-B-D-D-A-B-D-E-A	9190

Figure 7. Sequencing Demands for Example .

Figure 8 show the optimum products sequencing for multiple demands and figure 9 explain total completion time and number of generation when optimum solution.

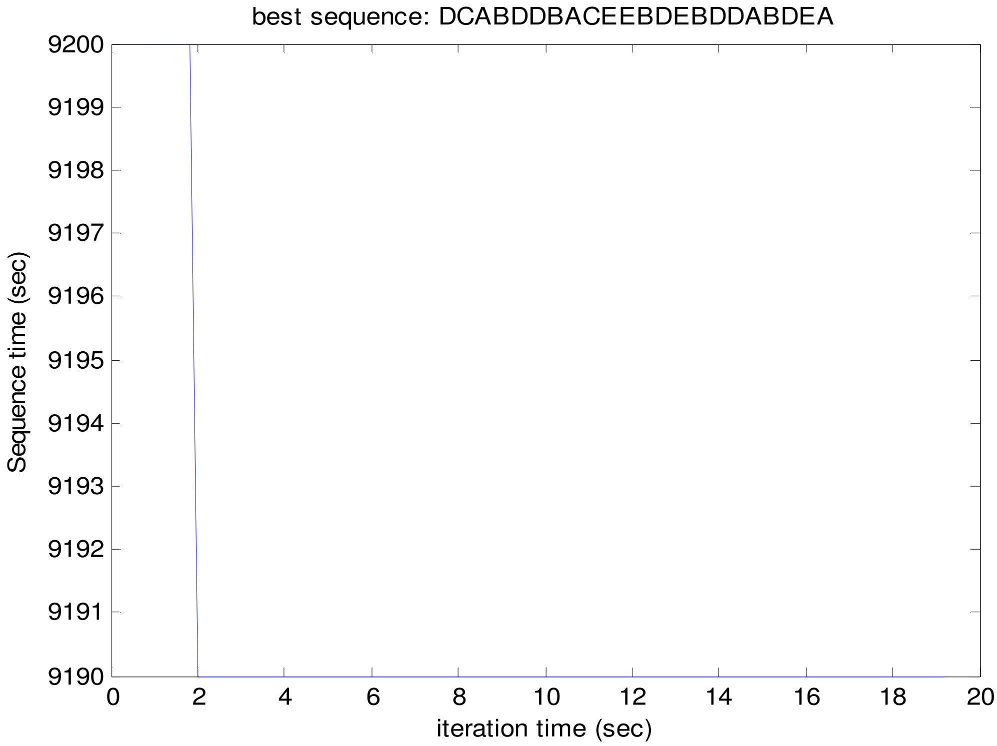


Figure 8. Optimum Products Sequencing for Multiple Demands

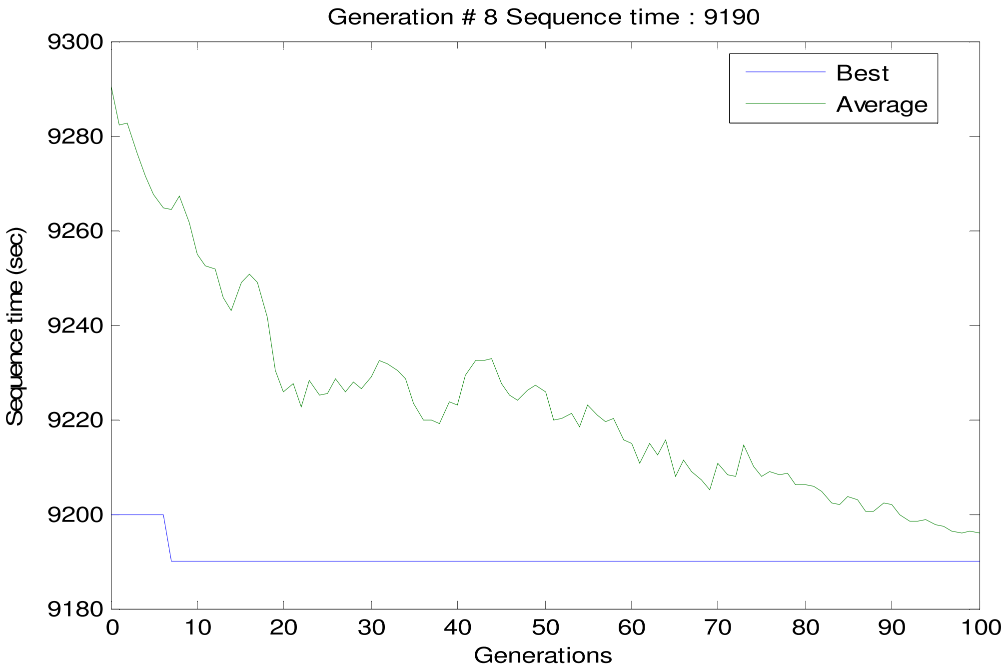


Figure 9. Total Completion Time and Number of Generation GA

4. Conclusion

We have described two method (MAM and GA) for determine the optimal products sequences in multiple demand , that all demand consist of multi-products dependent completion time.

Shown by the results of the two methods as follows:

1- GA is the best of MAM in determining the sequence of several demands with multi- products for being the less completion time .

2- GA take into account the product which ended the previous demand and which will begin in the later demand , either MAM depends on which ended the previous demand only.

3-TSPPCA helped to simplify the matrix of completion time as shown in Figure 8.

References

Applegate D. (2003) "Implementing the Dantzig Fulkerson-Johnson Algorithm for large traveling salesman problems," *Mathematical programming* vol. 97, pp. 91-153.

Bryant K., (2000)"Genetic Algorithms and the Traveling Salesman Problem," Dept of Maths, Harvey Mudd College.

Chatterjee S., *et al* (1996), "Genetic Algorithms and Traveling Salesman Problems," *European Journalof Operational Research,* vol. 93, pp. 490-510.

Dantzig G., (1954) "Solution of a large-scale traveling salesman problem," *Operations Research Letters,* vol. 2, pp. 393-410.

Farber G. and Coves A. (2005) "Overview on sequencing in mixed model flowshop production line with static and dynamic context," Universitat Politecnica De Catalunya.

Gaglani, Mansi S.(2011)"A study on Transportation Problem, Transshipment Problem, Assignment Problem and Supply Chain Management", thesis PhD, Saurashtra University**.**

Gambardella L. M. and Dorigo M. (1997) "HAS-SOP: Hybrid ant system for the sequential ordering problem,".

Garey M. R. and Johnson D. S. (2003), *Computers and Intractability: A guide to the Theory of NP-Completeness*. New York: W. H. Freeman and company.

Holland J. (1975), *Adaptation in Natural and Artificial Systems*: Ann Arbor: University of Michigan Press.

Ji P. and Ho W., (2005) "The traveling salesman and quadratic assignment problems: integration, modeling and genetic algorithm," in *Proceedings of International symposium on OR and its applications*.

Katayama K. and Sakamoto H. (2000) "The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem," *Mathematical and Computer Modelling,* vol. 31, pp. 197-203.

Keller M. T., (2004)"Knot theory, history and applications with a connection to graph theory," Thesis, Dept of Maths, North Dakota State University.

Kotecha K. and Gambhava N. (2003), "Solving Precedence Constraint Traveling Salesman Problem Using Genetic Algorithm," in *Proceedings of National Conference on Software Agents and embedded System*.

Larranaga P. and Kuijpers C. M. H. (1999) "Genetic algorithms for the traveling salesman problem: A review of representation and operators," in *Artificial Intelligence Review*. vol. 13, ed: Kluwer Academic Publishers, , pp. 129-170.

Lawler E. L. (1986) *The Traveling Salesman Problem*: John Wiley & Sons .

Mitchell G., (2007) "Evolutionary Computation Applied to Combinatorial Optimisation Problems," PhD Thesis, School of Electronic Engineering, Dublin City University.

Moon C. (2000), "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research,* vol. 140, pp. 606-617.

Netnevitsky M. (2002), *Artifial intelligence: A guide to intelligent systems*. England: Pearson Education Limited.

Psaraftis H. N (1983), "k-Interchange procedures for local search in a precedence-constrained routing problem," *European Journal of Operational Research,* vol. 13, pp. 391-402.

Reeves C. R.(1995) "A Genetic Algorithm for Flowshop Sequencing," *Computers & Operations Research,* vol. 22, pp. 5-13 .

Rekiek B. and Delchambre A. (2006), *Assembly line design: The balancing of mixed-model hybrid assembly lines with genetic algorithms*: Springer Series in Advanced Manufacturing, Springer-Verlag London Limited.

Schrijver A. (1998) " On the history of combinatorial optimization (till 1960),".

Stutzle,T. G. (1998) "Local search algorithms for combinatorial problems-analysis, improvements, and new applications," PhD Thesis, Technical University Darmstadz .

Wysocki R. K. (2009), *Effective Project Management: traditional, agile, extreme, 5th ed.* . Indianapolis: Wiley Publishing,.

Youssef H., *et al.*(2001) "Evolutionary algorithms, simulated annealing and tabu search: a comparative study," *Engineering Applications of Artificial Intelligence* vol. 14, pp. 167–181.