

Solving Travelling Salesman Problem (TSP) by Hybrid Genetic Algorithm (HGA)

Ali Mohammad Hussein Al-Ibrahim

Faculty of Information Technology

Department of Computer Science

The World Islamic Sciences and Education University (WISE)

P.O. Box 1101, Amman 11947, Jordan

Abstract—The Traveling Salesman Problem (TSP) is easy to qualify and describe but difficult and very hard to be solved. There is known algorithm that can solve it and find the ideal outcome in polynomial time, so it is NP-Complete problem. The Traveling Salesman Problem (TSP) is related to many others problems because the techniques used to solve it can be easily used to solve other hard Optimization problems, which allows of circulating it results on many other optimization problems. Many techniques were proposed and developed to solve such problems, including Genetic Algorithms. The aim of the paper is to improve and enhance the performance of genetic algorithms to solve the Traveling Salesman Problem (TSP) by proposing and developing a new Crossover mechanism and a local search algorithm called the Search for Neighboring Solution Algorithm, with the goal of producing a better solution in a shorter period of time and fewer generations. The results of this study for a number of different size standard benchmarks of TSP show that the proposed algorithms that use Crossover proposed mechanism can find the optimum solution for many of these TSP benchmarks by (100%) ,and within the rate (96%-99%) of the optimal solution to some for others. The comparison between the proposed Crossover mechanism and other known Crossover mechanisms show that it improves the quality of the solutions. The proposed Local Search algorithm and Crossover mechanism produce superior results compared to previously propose local search algorithms and Crossover mechanisms. They produce near optimum solutions in less time and fewer generations.

Keywords—Traveling Salesman Problem (TSP); NP-Complete problem; genetic algorithms; local Search algorithm; crossover mechanism; neighboring solution algorithm

I. THE PROBLEM OF TRAVELLING SALESMAN

A. Introduction

The issues of improvement to find the best solution optimization problems of the most important problems worthy of study, and characterized these issues with difficulty to reach solutions that meet all the goals, and examples of the problem of scheduling (timetabling problem) and the problem of finding the shortest route for a (travelling salesman problem) and the problem of coloring schemes (graph coloring problem) and the problem of transport (transportation problem), given the differences in the nature of each problem and the degree of difficulty cannot be to work with each algorithm with the same efficiency with all kinds of problems of finding the best solution (Newall, 1998) [18] and it has been many studies to compare the mechanism of the different algorithms and see

which ones more suitable to solve the problem (Newall,1998) [18], (Burke,1995) [2].

This study is concerned the problem of travelling salesman, find the shortest route which passes a range of cities without repetition.

B. Reasons for Choosing a Sales Trip Problem

The problem of travelling salesman old, it was the first to predicate the same problem at 1759 by Lawler, who was his attention focused on solving the problem of round knight on the chessboard so that visit all the boxes, patch once without any recurrence (Michalewicz, 1999) [14] and was the first use of the term travelling salesman in 1932 by a travelling salesman call Veteran has focused on a Rand Corporation in 1948 to resolve the problem of travelling salesman and helped the company's reputation by making this problem of problems known (Michalewicz, 1999) [14]. In addition, it has helped the emergence of liner programming in that time, solving the problem of travelling salesman. In spite of oldest of this problem its simplicity, but that did not make it easy to resolve, which, the authors should be widely available to many areas of research in mathematics and linear algebra to can understand the theory behind the solution to this problem. And the researches increased in this problem by the increased development in computers in terms of processing speed and storage capacity and memory, And has this problem drew the attention of many mathematicians and computer experts (Michalewicz, 1995) [15] for several reasons including:

1) Easy description, but its solution is very difficult because it belongs to a class of optimization problems called NP-complete which mean that there are no known algorithm that can find the ideal time of the result can be represented by multiple dimensions equation (Johnson polynomial).

2) Problem of travelling salesman can be generalized to many of the problems, such as:

- The problem of scheduling the tasks set to the CPU machine.
- Network routing problem.
- Path planning in mobile robot problem.
- Airline schedule problem
- As the solution to many of the problems of artificial intelligence will be finding the order of the number of

data in certain ways as a problem of circuit boards, electrical circuit board drilling, a method of solving the problem of a sales trip

3) We have a lot of information available and the standard questions about this problem, which it considered the mother problem (Michalewicz, 1995) [15] for many of the problems, so you can make many experiments on this problem and then present the solution methods to other problems.

C. Description of the Travelling Salesman Problem

On the assumption that there are a number of cities, (c1, c2, ..., cn) represented as points scheme, and there are lines between points is called edges and each edge length is the distance between cities, d(cj, ci) and the problem of travelling salesman is to find a path going through all the points so that the edges along the shortest possible (Johnson and Mcgeoch 1998) [8]. In other words, the salesman must visit all these cities and return to the city of departure under the following conditions:

- Salesman can visit only one city at any time.
- The salesman must visit all the cities and then return to its point of departure.
- Should not visit any city more than once in one round.
- The total distance should be the least that can be.

N-1

$$d(C\pi(i), C\pi(j)) + d(C\pi(N), C\pi(1)) \sum_{i=1}^{N-1} \text{Minimize}$$

The equation above represents the length of the tour, which the salesman went through during a trip to visit cities in the tour route in the order and then return to starting point. The travelling salesman problem has two main types:

- Asymmetric travelling salesman problem TSP, it mean that the distance between the city(i) and the city(j), is not equal to the distance between the city(j) and the city(i)

$$d(C_i, C_j) \neq d(C_j, C_i)$$

1 <= i,j <= N, where (N is the number round cities)

- Symmetric TSP, it mean that the distance between the city(i) and the city(j), are the same as the distance between the city(j) and the city(i).

$$d(C_i, C_j) = d(C_j, C_i) \quad \text{For each } 1 <= i,j <= N$$

It is clear that the symmetric travelling salesman problem is a special case of Asymmetric travelling salesman problem, but the practical experiences declare that it cannot disseminate the results Asymmetric travelling salesman problem to symmetric travelling salesman problem (Hoffman, 1985, 1991) [7], therefore this type need special research. The absence of any known algorithm to solve this problem in a time can be represented in terms of the number of NP-complete, make sure and only way to find the optimal solution is to calculate the lengths of all possible ways, and then choose the shortest route of them to be the best. But if we suppose it is possible to hold

one billion a calculation every second, we will need several years to calculate the lengths of possible routes for 25 cities and numbering (2,65 × 1032) route by (Lalena, 1998-2003)[10]

So must look for alternative routes and approximate algorithms to find a solution in a reasonable time relatively (katayama and Narihisa, 1999) [21], to be any sacrifice to reach the ideal solution to dissolve and sufficiency close to which is obtained in less time.

The most important approximate algorithms is genetic algorithms, including those that use local search, genetic algorithms, and the genetic algorithms have formed the subject of numerous studies (Michalewicz, 1996) [39,40] (Mitchell, 1996) [17] (Obitko, 1998) [19], but suffer from high time for implementation to get a good result, this study aims to improve the performance of genetic algorithms to reduce the number of generations and reduce the time needed to implement them.

II. GENETIC ALGORITHMS

A. Introduction

Genetic algorithms were inspired by Darwin's theory of evolution. The first began in the description of the idea, (John Holland's, 1992) [9]owner of theory of adaptive systems in the sixties (Man, 1999) [13] at the University of Michigan. The program is adaptive to study how the program can establish that the procedures that will allow itself to be altered depending on the efficiency of the existing environment (Goldberg, 1989) [5].

The main idea of the Genetic Algorithms is to do what nature does, it simulates what happens in living nature of the selection processes and the evolution and mutations in order to produce improved generations of parents specification involved in the process of mating, the more selected good qualities from the parents led to the emergence in children (David, 1989)[3]. And genetic algorithms is search and optimization based on the mechanics and the foundations of the survival of the best as it is in the processes of natural selection, chromosomes compete with each other and the environment, in which only the best remains to be in the new gene

New algorithms start encrypts set of solutions to a problem, and the way in which the chromosome encoding or representation that would reduce or increase the efficiency of genetic algorithms, which is the key to the solution, which expands approved work (Man, 1999) [13], these encrypts called Genes, that yields chromosome which is form a solution for a problem.

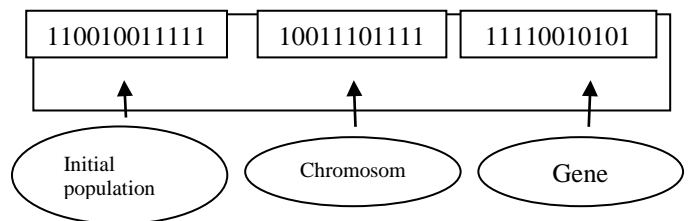


Fig 1. Genetic Algorithms Content.

Set of solutions that begins with it is called initial population, which will be subject to genetic algorithms for the

formation of new solutions and to obtain or approach the final solution of the problem, as shown in Fig. 1.

One Chromosome Genes subject to simple changes in the site or the situation which leading to the emergence of recipes and new genetic. This is called mutation Change, either inherited qualities of the chromosomes of parental to sons using the mechanism of the Crossover, through the merger of parts of chromosomes have been selected a set of solutions called parents, to create new solutions are called sons (David, 1989) [3] (Michalewicz, 1996) [39, 40]. After many generations, begins the solutions proposed begins approach each other with the hope that this is closer to the ideal solution. The most important characteristic of genetic algorithms from other methods of solution, where they produce a lot of solutions to the problem of time and then to improve the Crossover and mutations in an attempt to reach the best solution to the problem under research (Michalewicz, 1996) [16].

This is done by choosing the number of solutions the primary either randomly or by using one of algorithms approximation against known, then the subject of these solutions to the operations of the mutation and the Crossover which leads to better solutions or degradation, which requires the evaluation process help to choose good solutions based on the degree of efficiency, in order to reduce the bad solutions and access to more appropriate chromosomes.

B. The Mechanism of Genetic Algorithms

Genetic algorithms depend in his work on the adaptive methods and the stochastic search method used and the possibilities in the process of local search and un local.

C. Main Components of the Genetic Algorithms

Identifying the components of the genetic algorithms of the basic things, and the main components of these algorithms are:

- ✓ development solutions tools (Operators)
- ✓ mechanism for the representation of chromosomes (Solution Representation)
- ✓ mechanism for the establishment of elementary solutions (Initialization Procedure)
- ✓ Assessment mechanism resulting solutions (Evaluation Function).

D. Importance of Genetic Algorithms

Any researcher can take advantage of genetic algorithms, as long as he can encrypt the problem solving being addressed and put that solution in the chromosome, creating a function to evaluate solutions generated, which is a key to success in genetic algorithms comes from the ease of dealing with it, the time of their ability to find good solutions and quick to the problems incurable. And genetic algorithms are effective when:

- ✓ Research in a large and complex problem, or when it is difficult to understand.
- ✓ Be difficulty in reducing the search domain to a certain extent can be dealt with by traditional methods.

- ✓ There are no ways of mathematical solving
- ✓ Traditional search methods fail.

Have used genetic algorithms to solve many problems such as:

- ✓ Improvement problems to design electrical circuits (Louis, 1997) [11] and job scheduling (Goldstein, 1991)[6] and travelling salesman problem.
- ✓ Automatic programming.
- ✓ Machine learning, genetic algorithms has been used in many of these areas such as classification, prediction and design of control circuits.
- ✓ Economic models, development models and strategies of supply demand and the integration of economic markets.

Interaction between evolution and learning. Genetic algorithms have been used to demonstrate the ability of people to learn and their ability to influence others.

III. PREVIOUS STUDIES

A. Nagata and Kobayashi Study

This study (Nagata and Kobayashi, 1997) [12] provided a solution to the problem of travelling salesman where the proposed Crossover mechanism for the genetic algorithms called (Edge Assembly Crossover EAX), What distinguishes this study, it was able, using an edge assembly crossover EAX and without any algorithm local search solving the problem of travelling salesman until (3038) city and access to the ideal solution, but with implementation time over of the two and a half (Watson et al, 1998) [20].

B. Brady Study

Bardy (Bardy, 1985) [1] adopted in his study on the local optimization algorithms, and use the algorithm (2-opt) for this purpose.

Where the following steps declare Bardy Algorithm:

- 1) Generate a population of K starting solutions $S = \{P_1 \dots P_K\}$.
- 2) Apply 2-opt algorithm to each solution P in S., Letting the resulting locally optimal solution replace Pin S.
- 3) While not yet converged do the following:
 - 3.1. Select k' distinct subsets of S of size 1 or 2 as parents (the mating strategy).
 - 3.2. For each 1-element subset, perform a randomized mutation operation to obtain a new solution.
 - 3.3. For each 2-element subset, perform a (possibly randomized) crossover operation to obtain a new solution that reflects aspects of bath parents.
 - 3.4. Apply 2-opt algorithm to each of the k' solutions produced in step 3.3, and let S' be the set of resulting solutions.
 - 3.5. Using a selection strategy, choose K survivors from PU S', and replace the contents of S by these survivors.
- 4) Return the best solution in S.

C. Freiseleben and Study

This study (Freiseleben and Merz, 1997) [4] proposed Genetic local search algorithm with high efficiency to solve travelling salesman Problem based on the integration of genetic algorithms with the algorithm of Lin - Kernighan also proposed a new Crossover mechanism (Distance Preserving Crossover, DPX), where the sons born between them and each of the parents is equal distances, i.e., that the distances between the Son and the first Father are the same as the distance between the Son and the second Father.

It is notes her that using of Lin – Kernighan algorithm increase the efficiency of the results, on the other hand increasing the execution time by 80% .

D. Results of Previous Studies

From previous studies we note the following: -

- ✓ Use of local search algorithms to solve the problem of travelling salesman will give an appropriate solution, but it is often located in a local ideal solution, then there is a noticeable increase in execution time.
- ✓ In most studies have been dispensed from the representation of chromosomes(cities) the way the simple binary numbers and replaced so that the round shape is the order of the actual cities, even though the operations of Crossover and the mutation is difficult.
- ✓ Been developed more than style of the Crossover but these methods did not find the ideal solution, especially when the number of cities increase, also need high space in the memory and more execution or implementation time.

IV. PROPOSED ALGORITHM FOR SOLVING TRAVELLING SALESMAN PROBLEM

A. Search for Neighbor Solution Algorithm

Search for neighbor solution Algorithm Been proposed search algorithm for solving travelling salesman problem through genetic algorithms, which rely on a simple modification of the round, so that the re-arrangement of the sites of some cities for new rounds and keep optimal of them, and repeat the switch mechanism by (the number of round cities- 3) which mean that this algorithm does not need a long time to implement them. And viewed this process as a search for solutions in the vicinity of the rounds the current round, which may be one of them better than the original round.

The mechanism of this algorithm as following:

- 1) Cancel the path between the second city and the third city in the tour.
- 2) Re-arranged the tour to reverse direction of the cities on the tour starting from the end of the tour and ended by the third city (zone of separation in the tour).
- 3) Round length is calculated by comparing the length of the canceled tour lengths with the new tour, in order to avoid re-collection of lengths of all tracks in order to reduce the execution time of the algorithm.

4) If the efficiency of the new round is best than the original round, is retained in the provider population otherwise they be ignored.

5) Repeat the previous mechanism to separate the path between the third city and the fourth city, and then between the fourth city and the fifth city, and so on until the separation between the city and the city before the recent.

B. Proposed Crossover Mechanism

Each chromosome in the travelling salesman represents a solution to the problem. In this mechanism is to keep as much as possible of the tracks between cities and create a new paths do not exist in the rounds of parents, or even all the chromosomes in the provider population. This is opposed to most of the mechanics of the previous Crossover, which is based on inheritance as much as possible paths of parents to children, and without relying on previous information.

To apply the Proposed Crossover mechanism, two chromosomes are selected from the provider population, such that the first chromosome is the owner of the best efficiency until know, through the implementation of genetic algorithm (the best local solution), and selection of second chromosome randomly, thus the efficiency of first chromosome always the owner of the efficiency of the best local solution. On the assumption that the efficiency of the first chromosome with the best local solution has some qualities in common with the general chromosome with the best efficiency, we can also assume that the second chromosome may exhibit some qualities in common with the general chromosome with the best efficiency. So both chromosomes can similar in many paths, and tracks with similar characteristics are dominant in the total of the best parts of the general solution.

Based on previous assumptions, in the proposed Crossover mechanism to keep as much as possible from the common paths between the cities in the chromosomes of parents and the transfer of these tracks as they are to each child chromosome, in order to benefit from as much in common information in this issue. The uncommon tracks are being introduced in new common paths to choose the city closest to the city starting from the current remaining cities, hoping to get a new round with the best qualities of efficient and higher efficiency of the parents to continue the process to reach the best general solution.

Proposed Crossover mechanism actions are as follows:

- a) Choose two chromosomes from the provider population to configure the parents such that the first chromosome to be the owner of the local best efficiency, and the second chromosome are selected randomly.
- b) Choose a city tour of cities at random and placed in the output son chromosome and considered it the starting point.
- c) Find the location of this city in both rounds.
- d) Identify the neighboring towns of the city beginning in the first chromosome and determine if these cities had been previously visited or not?
- e) If you were to visit the neighboring towns of the city is already beginning to go a step (g), but neighboring cities are

selected for the city beginning in the second chromosome and compared these cities with cities adjacent to the city beginning in the first chromosome to determine the existence of any common paths.

f) If you find common paths between the cities of the first chromosome and the cities of the second chromosome, is the joint selection of the city and add to child chromosome and the mind as a new beginning.

g) In the absence of any tracks shared between the city of the beginning and nearby towns in the chromosome I and chromosome II or the neighboring towns of the city first had been previously visited, is selection of the city closest to the Town of the Beginning of the cities that have not been had yet to be added to the Son chromosome, and be the point a new beginning.

h) Repeat steps (c - g) to be completed the round.

Simply the Proposed Crossover mechanism as follows:

Choose two chromosomes, P1 is the local optima, P2 is chosen randomly Choose a starting city i randomly from the cities in the tour.

- Do the following until the stopping criteria matches. Find the position of the starting city in both p1 and p2.
- Make sure that the neighboring cities are not visited yet.
- If this is an unvisited city in common between the neighboring cities in p1 and p2, connect the starting city to it.
- Otherwise, find the nearest unvisited city to the starting city, and make it the new starting city.

C. Proposed Genetic Algorithm

The mechanism of the proposed genetic algorithm is as follows:

- 1) Create a primary provider population by the max limit.
- 2) Compute the efficiency of all chromosomes in the provider population.
- 3) Apply of the proposed local search algorithm on all chromosomes in the provider population.
- 4) Choose a chromosome with a better efficiency to put it in the provider population to cancel the worse the efficiency of the chromosome and replaced with a new chromosome with better efficiency, provided they do not repeat any chromosome in population provider.
- 5) Make sure the condition of the stay, the check expires.
- 6) Choose a chromosome with the best efficiency p1, and choose another chromosome randomly p2.
- 7) Apply the new Crossover mechanism to get a new chromosome.
- 8) Apply the local improvement algorithm on output chromosome.
- 9) Apply the mutation on the Son chromosome as the ratio of installed.
- 10) Compute the efficiency of the resulting chromosomes.

- 11) Put chromosomes that carry a better efficiency in the population provider.
- 12) Return the population provider to its original position.
- 13) Repeat the process from step 5 to step No. 12

Also see flowchart of the proposed Genetic Algorithm in Fig. 2 below.

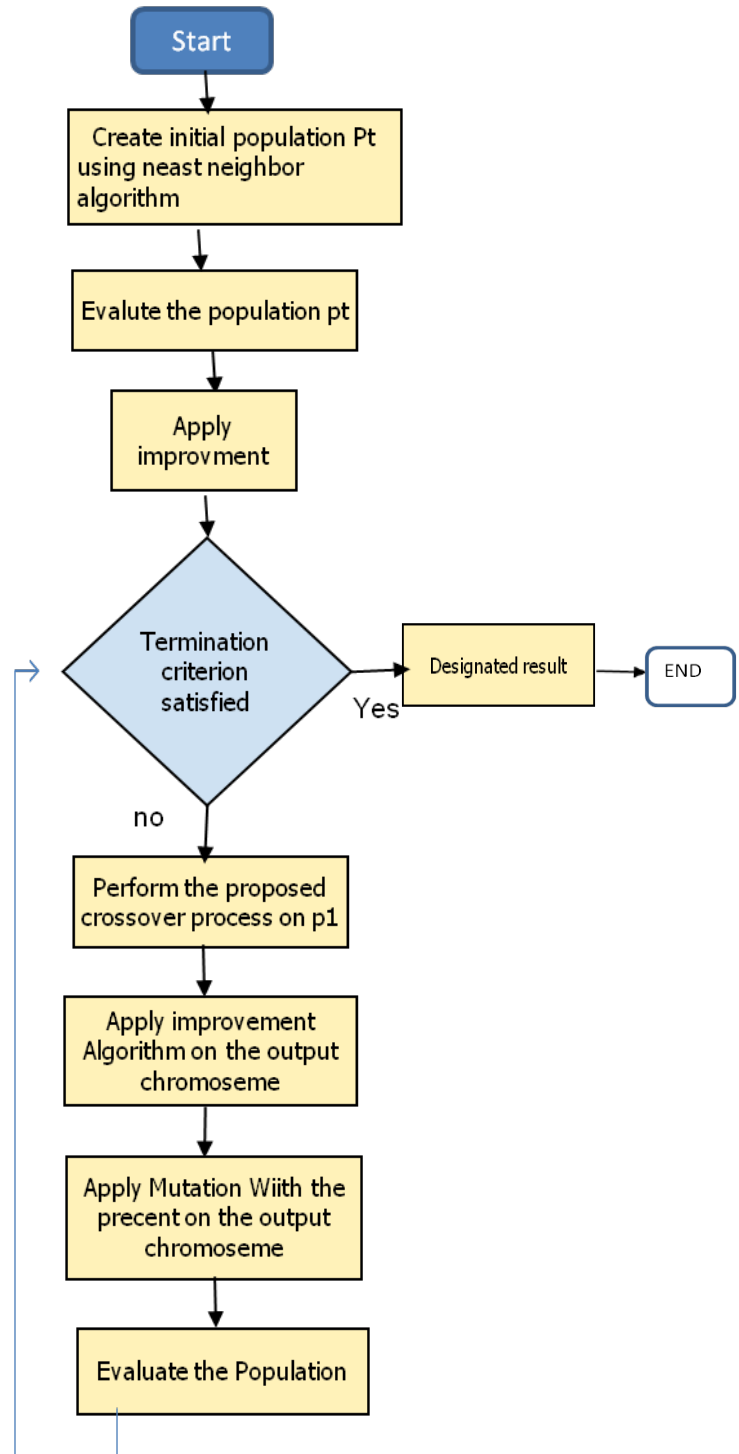


Fig 2. Proposed Genetic Algorithm.

V. RESULTS AND COMPARISONS

A. Results of Experiments

Genetic algorithm and local search algorithm to solve the travelling salesman problem implement in two forms, namely:

1) Individual form: Genetic algorithm works in single form without any local improvement local mechanism, and for comparing some of Crossover mechanism from previous studies are implemented.

2) Hybrid form: in which the local search algorithm works as a tool in the development of genetic algorithm to solve the research problem.

Individual and hybrid forms are been selected of the following objectives:

1) Define the ability of the proposed Crossover mechanism to solve the research problem.

2) show the importance of the introduce of the local improvement algorithm on the genetic algorithm

3) Note the difference between the results when the genetic algorithm works as Individual format and the hybrid

4) Determine the effect of the proposed Crossing on the performance of individual genetic algorithms by comparing the number of generations and time spent in twice.

5) Compare the different results obtained from the proposed Crossover mechanism with the results of previous studies.

these algorithms Implemented using a programming language (C++) and carried out all experiments on a Pentium 4 (Pentium IV), which operates 1000 MHz frequency and main memory (RAM) is equal to 128 kilobytes. The program was applied in each case ten times and taking the arithmetic average.

Genetic algorithm was implemented on some of the issues the proposed standard (Benchmark) as shown in Table I, obtained from the library of a sales trip (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB/tsp>)

TABLE I. BENCHMARK PROBLEMS

Benchmark	Cities Number	Optimal solution
A280	280	2579
Berlin52	52	7542
Ch130	130	6110
Ch150	150	6528
D1291	1291	50801
Gil262	262	2378
Rat195	195	2323
Rat575	575	6773
Tsp225	225	3916

B. Comparisons

Table II to Table VII show the practical results obtained for a sales trip in the following standard questions, Berlin52, Ch130, Ch150, Rat195, Tsp225, Gil262, A280, Rat575, D1291, using closest neighbor algorithm to the solution-building school. Known percentage of the solution resulting

from the proposed genetic algorithm optimization of the solution according to the following equation:

The proportion of approaching the solution of the resulting solution =

Solution resulting from the genetic algorithm / Standard solution (Benchmark)

Table II and Fig. 3 show the result of proposed genetic algorithm using nearest neighbor algorithm to formulate primary population provider and apply proposed crossover mechanism with research for neighbor solutions for local development algorithm.

Table III and Fig. 4 shows the result of proposed genetic algorithm using nearest neighbor algorithm to formulate primary population provider and apply proposed crossover mechanism without implementation of research for neighbor solutions for local development algorithm, which shows form the result the increasing of number of generations and faraway from optimal solution comparing with late table.

Table IV and Fig. 5 show the result of proposed genetic algorithm using nearest neighbor algorithm and using Operation Crossover (OX), which shows form the result the increasing of number of generations and faraway from optimal solution comparing with late Table II.

TABLE II. PROPOSED GA OUTPUT USE NEAREST GA WITH NEW CROSSOVER + NLSA

		Output solution	Percentage to	Execution time
Benchmark	Number of iteration	New Cross + NLSA	optimal solution	hh:mm:ss
A280	6500	2604	99%	00:00:48
Berlin52	50	7544	100%	< 1
Ch130	4800	6181	99%	00:00:11
Ch150	4920	6557	99%	00:00:12
D1291	25000	53055	96%	00:09:35
Gil262	9702	2416	98%	00:00:45
Rat195	2460	2346	99%	00:00:09
Rat575	18021	7013	97%	00:04:56
Tsp225	4750	3916	100%	00:00:23

TABLE III. PROPOSED GA OUTPUT USE NEAREST GA WITH NEW CROSSOVER + WITHOUT NLSA

		Output solution +	Percentage to	Execution time
Benchmark	Number of iteration	New Cross + without NLSA	optimal solution	hh:mm:ss
A280	86240	2781	96%	00:02:02
Berlin52	1900	7544	100%	00:00:02
Ch130	53940	6357	96%	00:00:24
Ch150	31970	6577	99%	00:00:21
D1291	90316	55427	92%	00:07:23
Gil262	90934	2472	96%	00:01:52
Rat195	23290	2371	98%	00:00:20
Rat575	90096	7481	90%	00:03:53
Tsp225	31250	3973	98%	00:00:30

TABLE IV. GA OUTPUT USE OPERATION CROSSOVER (OX) + NEAR LOCAL SEARCH ALGORITHM (NLSA)

Benchmark	Number of iteration	Output solution + OX+(NLSA)	Percentage to optimal solution	Execution time hh:mm:ss
A280	21390	2890	90%	00:02:12
Berlin52	960	7549	99%	00:00:11
Ch130	8955	6402	97%	00:00:36
Ch150	6616	6872	95%	00:00:30
D1291	15000	57277	89%	00:06:36
Gi1262	19790	2591	92%	00:01:62
Rat195	6035	2388	98%	00:00:29
Rat575	36891	7483	91%	00:03:05
Tsp225	9393	4143	95%	00:00:35

TABLE V. GA OUTPUT USE OX WITH + WITHOUT NEAR LOCAL SEARCH ALGORITHM (NLSA)

Benchmark	Number of Iteration	Output solution +	Percentage to optimal solution	Execution time hh:mm:ss
		OX +Without(NLSA)		
A280	89159	3066	84%	00:00:49
Berlin52	88375	7825	96%	00:00:13
Ch130	93083	6781	90%	00:00:16
Ch150	87747	6985	93%	00:00:16
D1291	100000	59108	86%	00:02:21
Gi1262	98734	2726	87%	00:00:35
Rat195	88195	2551	91%	00:00:17
Rat575	91257	8216	83%	00:00:58
Tsp225	90327	4673	83%	00:00:14

TABLE VI. GA OUTPUT USE MAPPED PARTIALLY CROSSOVER (MPX) WITH + NLSA

Benchmark	Number of iteration	Output solution MPX + NLSA	Percentage to optimal solution	Execution time hh:mm:ss
A280	23715	2881	90%	00:00:48
Berlin52	630	7544	100%	00:00:08
Ch130	9070	6217	98%	00:00:11
Ch150	8310	6565	99%	00:00:12
D1291	15000	56159	90%	00:04:41
Gi1262	21702	2531	94%	00:00:36
Rat195	12625	2367	98%	00:00:14
Rat575	28150	7403	91%	00:01:45
Tsp225	8447	4024	97%	00:00:18

Table V and Fig. 6 show the result of proposed genetic algorithm using nearest neighbor algorithm and using Operation Crossover (OX), without NLSA, for comparing results.

Table VI and Fig. 7 show the result of proposed genetic algorithm using Mapped Partially Crossover (MPX), with NLSA, which show Convergence of results.

TABLE VII. GA OUTPUT USE MPX WITH + WITHOUT NLSA

Benchmark	Number of iteration	Output solution MPX+NLSA	Percentage to optimal solution	Execution time hh:mm:ss
A280	88897	2949	87%	00:00:47
Berlin52	3640	7648	99%	00:00:11
Ch130	77474	6725	91%	00:00:12
Ch150	75583	6996	93%	00:00:13
D1291	100000	60024	84%	00:02:17
Gi1262	99858	2710	88%	00:00:26
Rat195	94661	2494	93%	00:00:17
Rat575	83152	8136	83%	00:01:47
Tsp225	89657	4504	87%	00:00:18

Table VII and Fig. 8 shows the result of proposed genetic algorithm using Mapped Partially Crossover (MPX), without NLSA, which show Convergence of results at Table V, but it is worse than result in Table II.

From the previous tables and when fixing crossover percentage and mutation for all problems, we noting the following:

1) The quality of result solutions from implementation of genetic algorithm on the benchmarks for travelling sales man approaching form the general optimal solution when using the proposed crossover mechanism without using any mechanisms solutions for local development algorithm.

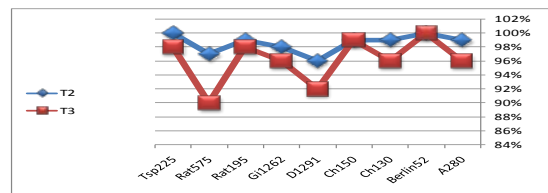


Fig 3. Proposed GA Output use Nearest GA with New Crossover + NLSA.

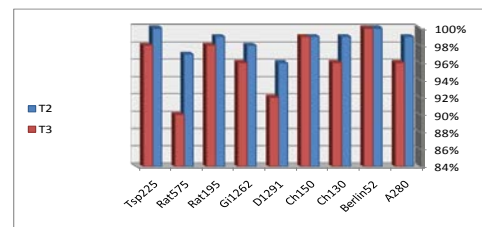


Fig 4. Proposed GA Output use Nearest GA with New Crossover + without NLSA.

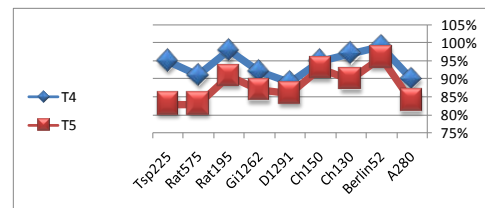


Fig 5. GA Output use Operation Crossover (OX) + NearLocal Search Algorithm (NLSA).

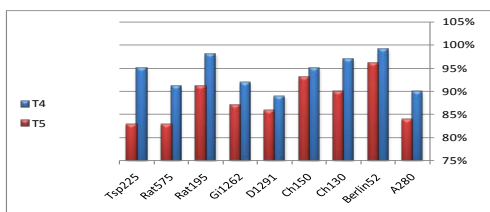


Fig 6. GA Output use OX with + without Near Local Search Algorithm (NLSA).

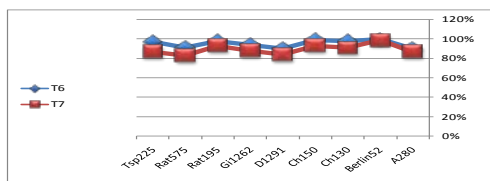


Fig 7. GA Output use Mapped Partially Crossover (MPX) with + NLSA.

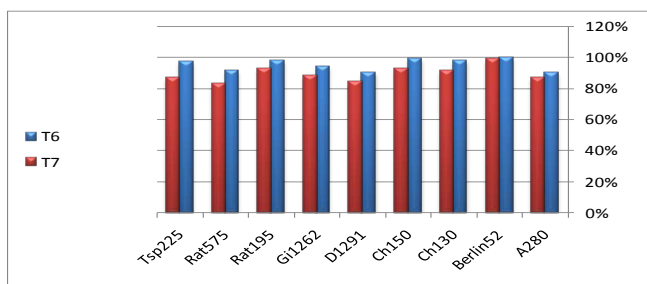


Fig 8. GA Output use MPX with + without NLSA.

2) By use local improvement algorithm on genetic algorithm by uses MPX: the percentage of performance solution is increase as shown below in Fig. 9 and Fig. 10:

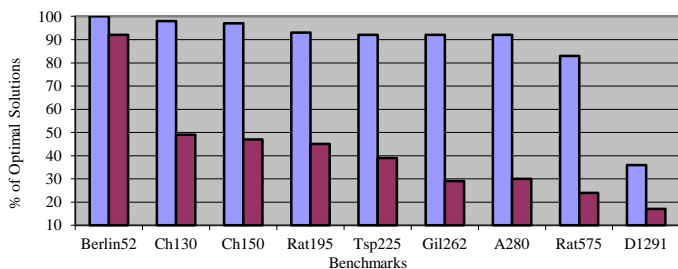


Fig 9. Compare (MPX+ NLSA) & (MPX+ No NLSA) Optimal Solution.

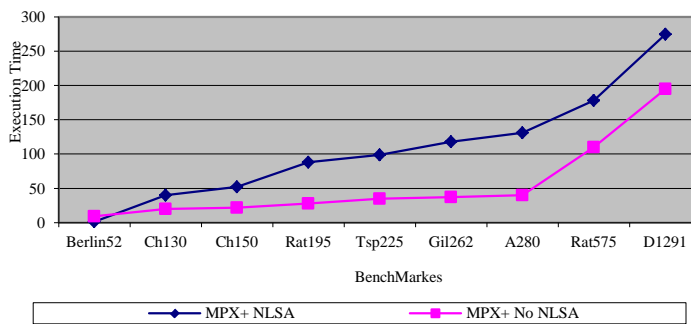


Fig 10. Compare (MPX+ NLSA) & (MPX+ No NLSA) Execution Time.

We applied genetic algorithm using new crossover technique and OX and MPX consider the follow-:

- Not use any of NLSA algorithms
 - Order initiate tour randomly.
 - Use same percentage (0.01, 0.8, 10000) ,
- as shown the following in Fig. 11.

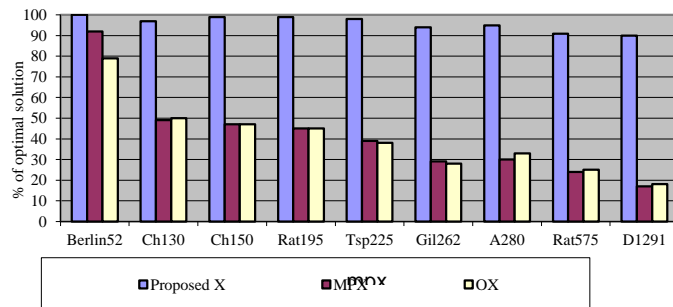


Fig 11. Compare (Proposed X) & (MPX) & (OX) Optimal Solution.

*New crossover(p) is the shortest execution time for genetic algorithm .The number of iteration for optimal solution is lowest for new crossover as shown in Fig. 12 below:-

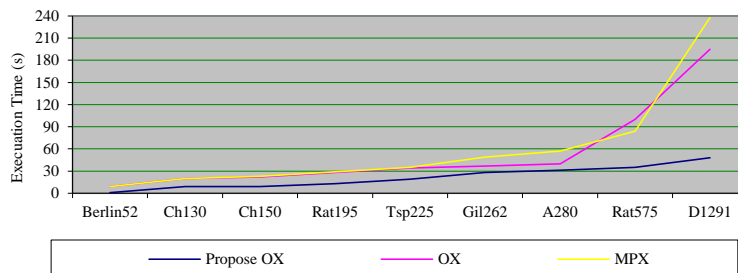


Fig 12. Compare (Proposed X) & (MPX) & (OX) Execution Time.

When we inject genetic algorithm by optimization algorithms we not the following-:

- 1) Increase the chance to have optimal solution.
- 2) Increase the result solution performance.
- 3) Increase in the execution time comparison by improvement quality of solutions.

As shown in Fig. 13 and Fig. 14 below:

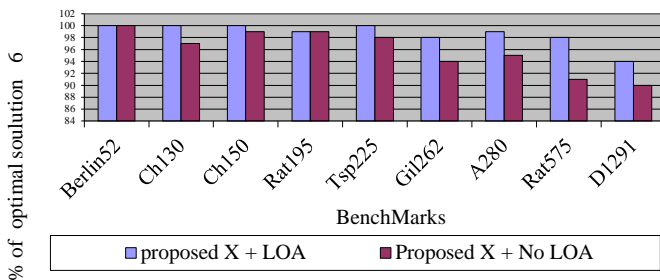


Fig 13. Compare (Proposed X+LOA) & (proposed X+ No LOA) Optimal Solution.

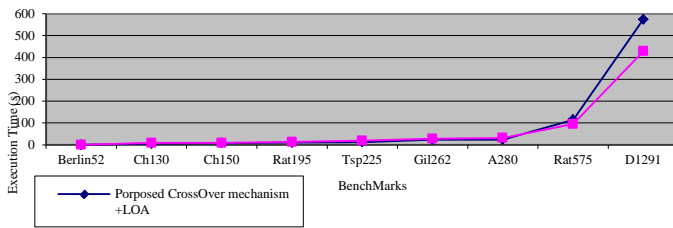


Fig 14. Compare (Proposed X+LOA) & (proposed X+ No LOA) Execution Time.

VI. CONCLUSIONS

This paper Suggested new Crossover mechanism to solve travelling salesman problem using genetic algorithms, has also proposed anew local improve algorithm to naming search for neighbor solutions algorithm entered on genetic algorithms to improve its performance and get better results and less and an increase of up to (10%) of the time implementation required to obtain the final result compared with other algorithms that have been comparison.

The following points can be drawn from this paper:

1) The proposed genetic algorithm addressed the travelling salesman problem and has a solution up to the percentage (100%) of standard solutions in some of the issues and to the proportion of (96% -100%) of the ideal solution in others matters.

2) Solve the travelling salesman problem using a proposed Crossover mechanism in genetic algorithms gave better results than in previous studies and reasonable time relative to the time required in the crossover mechanisms of the other private and tests were performed on the benchmark, number of cities where large up to (1291) city.

3) using proposed Local optimization algorithm as a tool to enhance the performance of genetic algorithms using different Crossover mechanisms raise the result efficiency and reduced the number of generations needed to get to the final solution.

4) Despite the increase in execution time when you enter local optimization algorithm on genetic algorithm, but this increase raised the efficiency of the final solution and significantly.

ACKNOWLEDGMENTS

The researcher gratefully acknowledges and highly appreciates the financial support and the remarkable resources provided by WISE University, Jordan.

REFERENCES

[1] Bardy, 1985, Optimization Strategies Gleaned from Biologic Evolution. Nature, 317: 804-806.

[2] Burke, 1995, A Hybrid Genetic Algorithm for Highly Constrained Timetabling problem, Proc. Of the 6th Int. Genetic Algorithms, San Francisco, Morgan Kaufmann, 1995.

[3] David, 1989, Proceedings of the 3rd International Conference on Genetic Algorithms, George Mason University, Fairfax, Virginia, USA, June 1989 ICGA 1989.

[4] Freisleben and Merz, 1997, Genetic Local Search for the TSP: New Results. In T. Bäck, Z. Michalewicz, and X. Yao, eds., Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, 159-164, Piscataway, NJ, IEEE

[5] Goldberg D. 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[6] Goldstein, 1991, Genetic Algorithm Simulation of the SHOP Scheduling Problem, by Jonathan M. Goldstein, in September 1991 published by An ICMS/Shell Oil Business Consultancy.

[7] Hoffman, 1991, M., Improving LP-representations of zero-one linear programs for branch-and cut. ORSA Journal of Computing, 1991, 3(2), 121—134.

[8] Johnson David and L.A McGeoch, 1998, “The Travel Salesman Problem: A Case Study in Local Optimization”, chapter in the book Local Search in Combinatorial Optimization edited by E.H.L. Arts and J.K. Lenstra. Vol. E81A, no.5, 1998, pp. 738-750.

[9] John Holland's 1992, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, 1975 (second edition: MIT Press, 1992).

[10] Lalena, 2003, Travelling Salesman Problem Using Genetic Algorithms, [http:// www.lalena.com/ai/tsp](http://www.lalena.com/ai/tsp).

[11] Louis S. 1997, Genetic Algorithms with Memory for Travelling Salesman Problem, 1997.

[12] Martin O., S.W. Otto, and W. Feten, “Large-Step Markov Chains for the TSP incorporating Local Search Heuristics”, Operations Research letters, vol. 11, 1992, pp. 219-224.

[13] Man, K. F. 1999, Tang, K. S. and Kwong, S., Genetic Algorithms concepts and Designs, Springer-Verlag London, Great Britain, 1999.

[14] Michalewicz Z. 1999, Genetic Algorithms + Data Structures = Evolution Programs, 3rd edition, Springer-Verlag Berlin Heidelberg, New York, 1999.

[15] Michal Genetic Algorithms, Numerical Optimization and Constraints, Proceeding Michalewicz, 1995, s of the 6th International Conference on Genetic Algorithms, Pittsburgh, July 15-19, 1995, pp. 151-158.

[16] Michalewicz, 1996, M., Evolutionary Algorithms for Constrained Parameter Optimization Problems, Evolutionary Computation, Vol.4, No.1, 1996, pp.1-32

[17] Mitchell, 1996, an Introduction to Genetic Algorithms, MIT Press, 1996 New York

[18] Newall, 1998, Hybrid Methods for Automated Timetable, Ph.D. Thesis, University of Nottingham, 1998.

[19] Obitko, 1998 Obitko, M. (1998). V. Operators of GA. Accessed from

[20] Schaerf A., “Tabu Search Techniques for Large High-School Times tabling”, Proceedings of the 13th American Conference on Artificial Intelligence, 1996.

[21] Katayama and Narihisa, 1999, Iterated Local Search Approach Using Gene Transformation to the Travelling Salesman Problem. In W. Banzhaf, ed., Proceedings of the Genetic and Evolutionary Computation Conference, 321-328, Morgan Kaufmann, 1999.