Journal of Theoretical and Applied Information Technology
31st October 2021. Vol.99. No 20
© 2021 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW CROSSOVER VIA HYBRID ANT COLONY SYSTEM WITH GENETIC ALGORITHM AND MAKING STUDY OF DIFFERENT CROSSOVER FOR TSP

ASMAA HEKAL OMAR, AMANY AHMED NAIM*

Faculty of Science, Al-Azhar University (Girls Branch), Cairo, Egypt

E-mail: amany.naim@azhar.edu.eg*; asmaahekl@azhar.edu.eg

ABSTRACT

The traveling salesman problem (TSP) is a very famous NP-hard problem in computer science and operations research. In this study, proposed a new hybrid crossover (SPMX) combining the shuffle crossover and partially mapping crossover which served to develop Genetic algorithm (GA) to solve this problem since crossover is the main component of GA. And, apply the proposed SPMX on hybrid ant colony system with GA with Ant colony system (ACS), which called by (ACSGA) to solve TSP. Experimental results on some well-known TSPLIB instances of different types and sizes. The obtained comparative study clearly demonstrates the utility of the proposed a (SPMX) to enhance ACSGA for resolving TSP and the distance decrease by 0.82% with average 0.47%.

Keywords: *Travelling Salesman Problem (TSP), Genetic Algorithm (GA), Crossover Operators, Ant Colony System (ACS)*

1. INTRODUCTION

Travelling salesman problem is a famous problem (TSP) and is a NP- Hard problem [1]. It is very easy to define but difficult to solve, the find the shortest tour is aim. The journey is begin from a depot node and visit all nodes (cities) only once and then returns to the depot.

Many exact and metaheuristic algorithms have been mentioned for solving the TSP. "Branch and bound" [2] and "branch and cut" [3] are some exact algorithms. These algorithms give the exact optimal solution to the problem, but as the problem size increases the computational time increases dramatically. So, the small size of TSP can be solved to exact optimal. In another hand, the metaheuristic algorithms give an almost optimal solution in a reasonable computational time, but do not guarantee the superiority of the solution [4]. Some example of metaheuristic algorithms are genetic algorithm (GA) [5], ant colony optimization [6], simulated annealing [7], tabu search [8], artificial neural network [9], and Ant Colony System (ACS) [10, 11].

Genetic algorithm (GA) is an optimization algorithm and is the best and widely used algorithm to solve the TSP as well as other complex combinatorial optimization problems in computer science and operations research. GA has important operator is called by crossover, it plays an effective role to get the best results [12, 13]. Also, Ant Colony Optimization (ACO) is one of the most successful and uses systems now. ACO algorithm has attracted the interest of researchers due to its efficiency, effectiveness, and simplicity in solving complex optimization problems such as TSP and finding the shortest path at the lowest cost [14].

Hussain et al. [17] discussed the several crossover operators have been introduced, and proposed a new crossover operator for TSP by using GAs. This proposed operator ICX (improved form of cycle crossover) improves the path-represented CX and used to improve the quality of offspring.

Kumar [19] discussed TSP is solved using GA. An improved genetic algorithm is proposed which uses a hybrid initial population. The nearest neighbour greedy approach is used to generate a hybrid initial population.

Ahmed [16] proposed many crossover operators have been proposed for the TSP using GAs which can also be used for its variations. Eight simple GAs using eight different crossover operators.

Elsayed et al. [15] proposed a hybrid model Ant colony system and the genetic algorithm (ACSGA) to solve TSP. It was observed that starting with ACS in the case of hybrid ACSGA be effective to choose and estimate the starting solution.

Since the crossover operator plays an important roles in GA, so many crossover operators have been proposed to solve TSP. The aim of this paper is two folds. Frist it proposes a new crossover is called SPMX, which hybrid between PMX crossover and shuffle crossover. Second it makes study of different types of crossover and the effect on performance of GA. This paper applies proposed method with the hybrid ACSGA for TSP. The experiments are applied on five TSPLIB instance.
The remainder of this paper is organized as follows: Section 2 presents Travelling Salesman Problem. Section 3 states genetic algorithm and crossover. Ant colony optimization is listed in Section 4. Section 5 presents the proposed method. Section 6 presents the experimental results and analysis. Finally, the conclusion is presented in Section 7.

2. TRAVELLING SALESMAN PROBLEM

The Travelling Salesman Problem (TSP) is NP-hard problems and one of the most common and important optimization problems in operational research. In TSP, a salesman starts from any city in the tour and back to the starting city. A tour must include all the cities which given for visit and following the shortest path possible.
Assuming there is N number of cities that the salesman has to visit, then it means there are N different path that the salesman can choose.

Using Euclidean distance for calculating distance between cities *i* and j in Eq. (1) as follows:

$$d_{ij}= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \text{ ---------(1)}$$

Where, d_{ij} is the distance between cities *i* and *j*, x_i is *x* Cartesian coordinate for city *i*, x_j is *x* Cartesian coordinate for city *j*, y_i is *y* Cartesian coordinate for city *i*, y_j is *y* Cartesian coordinate for city j [21, 22].

3. THE BASIC GENETIC ALGORITHM

The genetic algorithm (GA) was presented in the 1970s by John Holland. GA is a form of evolutionary algorithms, which simulates natural selection [23].

The details of GA components as a fitness function, crossover, and mutation operators are presented.

3.1. Fitness Function

The fitness function is the main component of GA which evaluates the chromosomes, and objective function is defined as the sum of the costs of edges in the tour. The GA is used for the maximization problem. For the maximization problem, the fitness function is the same as the objective function [15, 16]. But, for a minimization problem, one way of defining a fitness function in Eq. (2) as follows:

$$F(x) =1/f(x) \text{ ----------------------------(2)}$$

3.2. Selection Operator

The objective of this step is to select the suitable individual for matting. Selection algorithm describes the methodology to choose the suitable individual of the parents for a meeting pool. These parents will generate children for the next generation. Roulette selection is an example of selection, chromosomes are represented in the roulette wheel proportionally to their fitness functions [15, 24].

3.3. Crossover Operator

Crossover operators play a vital important role in GA. Crossover technique is inspired from biology: children by inheriting their parents' genes can be more able and may have better fitness than their parent.

Several crossover operators are created to reach in minimum iterations the optimum solution [25]. Seven crossovers are described: Order Crossover Operator (OX), Partially Mapped Crossover (PMX), Cycle Crossover (CX), Uniform Crossover, Shuffle Crossover, Generalized N Crossover (GN) and Sequential Constructive Crossover (SCX). So many crossover operators have been proposed for the TSP [26].

3.3.1. Partially mapped crossover operator

Goldberg and Lingle [27] developed the partially mapped crossover (PMX) that used two crossover points. It defines an exchange mapping between these points. The first crossover used in GA to solve TSP is PMX [12]. For example, the two paths cities where chromosome representing the path P1: (2, 1, 3, 4, 6, 9, 5, 7, 8) and P2: (2, 3, 5,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7, 8, 9, 4, 1, 6). Consider the same pair of cities for stating all the crossover operators considered here. Suppose the randomly selected cut-off points are between 3rd and 4th genes and between 7th and 8th genes as follows (these cut points are marked with "|"):

P1: (2, 1, 3 | 4, 6, 9, 5 | 7, 8) and
P2: (2, 3, 5 | 7, 8, 9, 4 | 1, 6)

In this example, the mapping systems are 4↔7, 6↔8, 9↔9, and 5↔4. Now these mapping systems are copied with each other to build offspring as follows:

O1: (2, *, * | 7, 8, 9, 4 | *, *),
O2: (2, *, * | 4, 6, 9, 5 | *, *)

Then we can add more genes from the original parents as follows:

O1: (2, 1, 3 | 7, 8, 9, 4 | *, *),
O2: (2, 3, * | 4, 6, 9, 5 | 1, *)

The first * in the first offspring(O1) should be 7 that comes from first parent, but it is present in this offspring, so, we review mapping 4↔ 7, but 4 is also present in this offspring, again review mapping 5↔4, so 5 is added. Similarly, the second * in (O1) should be 8 that comes from first parent, but it is present in this offspring, so, we review mapping 6↔8 and hence, we add 6 at second *. Thus, the first offspring becomes

O1: (2, 1, 3 | 7, 8, 9, 4 | 5, 6),
Similarly, the second offspring as:
O2: (2, 3, 7 | 4, 6, 9, 5 | 1, 8).

3.3.2. Ordered crossover operator

The method is a trivial variation of PMX. To create offspring chromosomes, the ordered crossover (OX) selects a subsection of a route from one parent chromosome and then it maintains the relative arrangement of genes from the other [26].

We choose parent chromosomes and cut points marked with "|" as:

P1: (2, 5, 4 | 7, 8, 1 | 3, 6) and
P2: (2, 8, 3 | 4, 5, 6 | 1, 7)

We always fix first gene as 'node 1'. At first,

The offspring is generated by copying the parts between these pieces as follows:

O1: (2, *, * | 7, 8, 1 | *, *),
O2: (2, *, * | 4, 5, 6 | *, *)

Then, starting from 2nd cut of one parent chromosome, the genes un-available from the other chromosome are copied in the same sequence. The arrange of genes in P2 from the 2nd cut is

{1 →7→8→3→4→5 →6}. After ignoring the already available genes 7, 8 and 1 in O1, the order becomes {3→4→5 →6}, which is added in O1 starting from the 2nd cut point:

O1: (2, 5, 6 | 7, 8, 1 | 3, 4).

Similarly, second offspring is created as:
O2: (2, 8, 1 | 4, 5, 6 | 3, 7).

3.3.3. Sequential constructive crossover

The sequential constructive crossover (SCX) operator constructs an offspring using better edges on the basis of their values present in the parents' constructing. Furthermore, it also uses the better edges, which is not found in the parents' structure.

SCX sequentially searches both of the parent chromosomes and take into account the first legitimate node (i.e. unvisited node)

that appeared after the previous visited node and in the absence of legitimate node is found in either of the parent chromosomes,

it sequentially searches for the legitimate node (s) and then compares the associated cost to locate the next node of the child chromosome [28].

Step 1: Start from "first node " (for example, current node p = 1).

Step 2: Conduct a sequential search of each parent chromosome and consider the first "legitimate node" (the node that has not yet been visited) that appeared after the "p-node" in each parent. If there is no "legitimate node" after "node p" in either parent, search in sequence from the beginning of the parent and consider the first "legitimate node", and go to step3.

Step 3: Assume that 'node α' and 'node β' are present in the first and second parent, respectively, and to select the next node, proceed to step 4.

Step 4: If cpα <cpβ then select " node α", otherwise, " as node β" is the next node and chain it to the partially formed offspring chromosome. If the offspring is a complete chromosome, then stop, otherwise, rename the current node to 'node p' and go to step 2.

3.3.4. Cycle crossover operator

In this technique bits are come circular motion from both parents together with their position which gives a good result in less iteration. However, it produces offspring same as the parents [29].

3.3.5. Generalized n-point crossover operator

Radcliffe and Surry developed generalized N crossover (GNX) [30, 16].

1) Suppose N=2, and P1: (2, 5, 4, 7, 8, 1, 3, 6) and P2: (2, 8, 3, 4, 5, 6, 1, 7). If the points of crossover are 4 and 6, then the nodes of the bold face are usually determined by G2X.
2) Assume that the sections are tested in the

order (3, 2, 1). Then the third part of the random parent will be added, suppose P2, to give the first child (*, *, *, *, *, *, 1, 7). Then, the nodes will be tested in the second part of P1 in random order. Node 8 is accepted to give the first child (*, *, *, *, 8, *, 1, 7). Then the nodes are tested in the first part of P2, and nodes 2, 3 and 4 are accepted to give the final first child after the first stage: (2, *, 3, 4, 8, *, 1, 7).

3) Then the untested parts are being tested on both parents in arbitrary order. Only the second part of P2 is appropriate here and node 6 is acceptable. So, the first child after the second stage is (2, *, 3, 4, 8, 6, 1, 7).
4) Since this offspring is not yet complete, we fill it randomly. Therefore, the ultimate offspring may be (2, 5, 3, 4, 8, 6, 1, 7). Only four edges are chosen from either parent.

3.3.6. Uniform crossover

Uniform crossover as the best operator for certain problems because it can avoid the disadvantages of destroying building blocks [31]. The steps for generating offspring of uniform crossover can be described in [32].

3.3.7. Shuffle crossover

Shuffle crossover has been proposed by Eshelman et al. [33] to reduce bias introduced by other crossover techniques, and it is associated with uniform crossover. a single crossover position of is determined (as in single-point crossover).

It shuffles the individual solution values before the crossover, they are randomly shuffled in both parents and unshuffles them after performing the crossover process so that the point of crossover does not introduce any bias in crossover as the variables are randomly reset every time a crossover is made.as shown in this example:

Parent 1: 1 1 0 0 0 1 1 0
Parent 2: 1 0 0 1 1 0 1 1

Parent 1: 0 0 1 0 1 1 0 1
Parent 2: 0 1 1 1 0 1 0 1 , After shuffling bits

Offspring1: 0 0 1 0 1 1 0 1
Offspring 2: 0 1 1 1 0 1 0 1single point crossover

3.4. Mutation Operator

The main purpose of Mutation operation is to prevent the algorithm from being caught in local minima. The mutation operator performs modification of information in the chromosome by selecting two nodes randomly and altering it or flipping a single gene in the chromosome [34].

4. THE BASIC ANT COLONY OPTIMIZATION ALGORITHM

ACO was introduced by Marco Dorrigo. This algorithm is inspired by the behavior of ants in the real world.

Ants follow the principle of survival of colonies. Ants are known for performance very difficult tasks in a very simple way. They search for path which is the shortest one between a food sources and nest. Initially, they move randomly While exploring the path, they deposit a chemical, pheromone on their path [35]. If an ant finds food, it again goes back to its nest by depositing that pheromone trail on the ground. Other ants smell the depositing pheromone trail.

Ant selection for path depends on the amount of pheromone deposited on this pathway.

The higher concentration of the pheromone on the pathway, the more probability that pathway was selected by another ant. Therefore, the probability of choosing the path and intensity of the pheromone are directly proportional.

The ACO is successfully in finding good solutions to difficult optimization problems such as (TSP) [36].

4.1. Ant Colony System (ACS)

Ant colony system (ACS) is a forms Ant colony optimization (ACO).

ACS (Ant Colony System) is a method of heuristically generating "best solutions" to the TSP that described by Artificial researcher, where (ACS) one of the best effective methods in finding the shortest path [37]. ACS contains a set of ants in random cities intended to move to other cities, and Ants can make local and global updates of the pheromone pathway. Pheromone Path is which determines for choosing the next city to move and every Ants movement on Pheromone Path creates a path with a stronger Pheromone Path [18]. This process is repeated get even the shortest tour, or all ants converge in the same path. Ants move from place to another according to the move rule. The probability of ant k moves from i to next node j by following transmission rule in Eq. (3) as follows.

$$P_k(i,j) = \begin{cases} \frac{\tau(i.j).[\eta(i.j)]^{\beta}}{\sum \tau(i.u).[\eta(i.u)]^{\beta}} & i\in J_k(i).\ u\in J_k(i) \\ 0 & \text{otherwis} \end{cases} \quad \text{-(3)}$$

where, τ is the amount pheromone, $\eta = 1/\delta$ is the inverse of the distance (i, j), $Jk(i)$ is the set of not visited node by ant k. β is a parameter determines the importance of pheromone versus distance $(\beta > 0)$. Ants after each step update pheromone locally when selecting their next node, so Pheromone is locally updated by following rule in Eq. (4) as follows

$\tau(i,j) \leftarrow (1-\rho).\tau(i,j)+\rho.\Delta\tau(i,j)$ ---------------(4)

Where, p is pheromone evaporation [18], $0 < \rho < 1$, and the acceptable value of $\rho = 0.5$. And $\Delta\tau(i, j) = \tau 0$ where τ is pheromone. Then Pheromone is globally updated by following rule in Eq. (5) as follows.

$\tau(i,j) \leftarrow (1-\alpha).\tau(i,j)+\alpha.\Delta\tau(i,j)$ ----------------(5)

α is a decomposition parameter of the pheromone, α = 1. The value of $\Delta\tau\ (i, j)$ is determined as the value of $\Delta\tau\ (i, j)$ is determined According to Eq. (6) as follows.

$$\Delta\tau(i,j)=\begin{cases} \frac{1}{l_{gb}} & \text{if}(i.j)\epsilon \text{ global best tour} \\ 0 & \text{otherwise} \end{cases} \quad ----(6)$$

Where, Lgb is the length of the globally best tour. The Pheromone trail updating to improve the performance through Max–min ant system (MMAS).then the updating rule is given by Eq. (7).

$\tau_{ij}(t+1)=\tau_{ij}(t)+\Delta t^{best}$ --------------------------(7)

Δt^{best} proposed in ACS [38] where $\Delta t^{best}=1/f\ (s\ best)$, and $f\ (s\ best)$ denotes the solution cost of either the iteration best solution. Then, making the compare for F in each iteration to get the best solution [39].

The hybrid model (ACSGA) combining the ant colony system (ACS) with the genetic algorithm (GA). ACSGA faster than other algorithms in obtaining the shortest path. GA is a strong global optimization tool to produces the initial population randomly, in GA crossover is the most important operator. So, we consider seven crossover operators in GAs to comparison results for TSPLIB, so proposed a new hybrid crossover (SPMX) that combines shuffle and partial mapping crossover for applying in ACSGA to solve the TSP [15].

5. PROPOSED METHOD

Many crossover operators are present in GA that are used in solving optimization problems.

The effect of crossover operators in GA is application dependent as well as encoding.

Shuffle is a type crossover that reduce bias introduced by other crossover techniques.

It shuffles the individual solution values before the crossover and unshuffled them after performing the intersection process so that the point of intersection does not present any bias in the intersection.

Partially mapping crossover (PMX) is the most frequently used crossover operator.

PMX that performs better than most other crossover operators.

5.1. The Design of Hybrid Crossover (SPMX)

To solve the TSP and obtain effective solutions, we use the features of both shuffle crossover and PMX have been used in our proposed crossover that applying in the hybrid model ACSGA.

The proposed hybrid SPMX crossover is displayed, begins with PMX that evaluating the start-up then apply shuffle on the offspring the proposed SPMX crossover is shown in this step.

1- PMX Crossover choose the two parents for crossover.

2- Select two-point crossover in parents and copy the swab from first parent (P1) into the first offspring.

3- From first crossover point, look for elements in swab of second parent (P2) that have not been copied.

4- For each element (such as i) that not copied, look in the first offspring to find what elements have been copied from P1 (such as j).

5-Put i in the position that occupied by j in P2.

6- If the position occupied by j in P2 is already full in the offspring by o, then place i in the position occupied by o in P2.

7- Fill in all positions of the first offspring similarly for all elements in P2.

8- Repeat this set of steps for the second offspring reversing arrangement of parents.

9- Then applying shuffle crossover to the offspring which was obtained from PMX and consider them as parents (P1, P2).

10- Shuffle randomly the genes in the both parents but in the same way.

11- Apply the 1-Point crossover technique by randomly selecting a point as crossover point and then combines both parents to create two offspring

For example, if we apply the proposed (SPMX) to solve TSP.

Suppose, there are nine nodes that are paths, first we apply PMX on the paths (represent parents) that chosen according to fitness function:

Consider parents

P1: (3 2 4 1 5 6 7 8 9)

P2: (4 6 7 5 2 3 9 8 1)

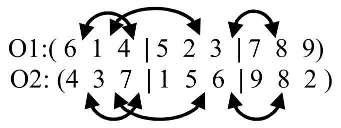
Then assumed the cut points that are marked with "|", as follows:

P1: (3 2 4 | 1 5 6 | 7 8 9)
P2: (4 6 7 | 5 2 3 | 9 8 1)

Then the offspring from PMX produced as follows: $1 \leftrightarrow 5$, $5 \leftrightarrow 2$, $6 \leftrightarrow 3$

O1:(6 1 4 | 5 2 3 | 7 8 9)
O2: (4 3 7 | 1 5 6 | 9 8 2)

Consider two offspring as anew parent to apply shuffle crossover



O1:(4 2 6 5 1 | 7 3 8 9) After
O2: (7 5 4 1 3 | 9 6 8 2) Shuffling bits

Then the new offspring from SPMX produced as follows:

O`1:(4 2 6 5 1 | 9 8 7 3)
O`2: (7 5 4 1 3 | 8 9 2 6)

5.2. The Proposed Hybrid Shuffle and Partially Mapped Crossover with (ACSGA)

The proposed SPMX that enhance ACSGA start with ACS where all the ants are initialized in the starting city and finally return to the same city of the beginning, initial pheromone is supposed as the inverse of the distance between the cities to solve TSP. all cities are visited once in different paths, which makes the local update of pheromone on all the discovered paths and the global update of the pheromone on the shortest path. Parameters α and β determine the pheromone pathway and the distance between two cities.

After applying ACS and MMACS, the best paths are chosen based on the path length from the available paths, where the shortest path is chosen and then GA executed. So, GA starts working on results of ACS. each fitness score starting from the initial population (ACS results), We used roulette wheel selection which is the most popular method either fitness appropriate selection to select chromosomes with more fitness values and crossover procedure in our application. After that the proposed crossover SPMX, mutation function of GA. these results lead to the shortest path and finding space for search both locally and globally.

This paper apply Proposed SPMX crossover on ACSGA algorithm to solve the TSP [15].

Algorithm for ACSGA that use the proposed SPMX:

Step1: Initialize parameters of ants and pheromone values (Iteration *i =1*).

Step2: Generating the primary distribution of pheromones according to the optimization solution:

a) Calculate the transition probability for each ant, the ant moves according to the probability.
b) Choose paths randomly based on an updated pheromone trail.
c) Update pheromone values (local update).
d) Store and update the best global tour for ant.

Step3: Choose a short tour from the output (ACS/MAX-MIN) of Step2 and perform following steps:

a) Use the chosen tours as the initial population and produce population.
b) Calculate fitness function and do Selection.
c) Apply SPMX Crossover of 2 tours to produce an offspring.
d) Apply mutation operators on offspring
e) Calculate the fitness of a new generation (new path).

Step4: if the cost new tour < cost global best tour then goes to step 7

Step5: Iteration *i = i+1*

Step6: Repeat Step 2 to Step 5 until all the ants converge in one path (the shortest path) or a suitable result is achieved

Step7: End

The proposed hybrid ACSGA with SPMX shown below of the flowchart is shown in Fig. 1.

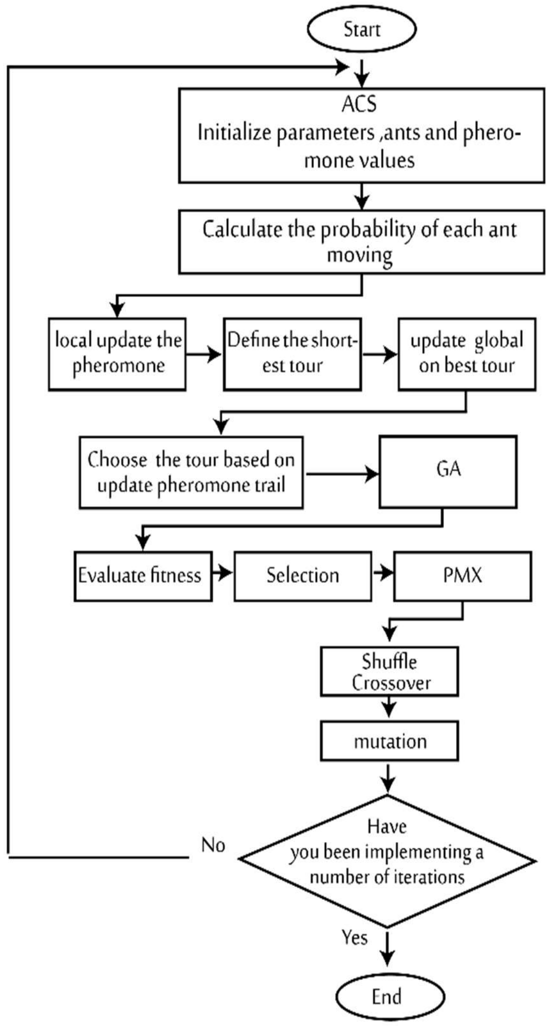

Start
ACS
Initialize parameters ,ants and pheromone values
Calculate the probability of each ant moving
local update the pheromone
Define the shortest tour
update global on best tour
Choose the tour based on update pheromone trail
GA
Evaluate fitness
Selection
PMX
Shuffle Crossover
mutation
Have you been implementing a number of iterations
No
Yes
End

Fig. 1 The Flowchart of the Proposed Model (ACSGA (SPMX))

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

To perform compare study among seven different crossover operators and proposed SPMX, simple GA using these crossover operators have been encoded in hybrid ACSGA, and then run for five TSPLIB instances. The five instances olvir30, eil51, berline52, eil76 and eil101 are symmetric. We run ACSGA for different setting of parameters, and the parameters used are listed in Table 1.

Table 1. The Different Setting for Parameters of Proposed SPMX that used in ACSGA

parameters	Cross over rate	Mutation rate	Pop Size	α	β	ρ
value	1	0.0001	100	1	10	0.5

The problems that are used for experimental results can be used from TSPLIB [40]. Most problems in TSPLIB are solved and the optimum values are presented. The numbers in the name of problem refer to the number of cities.

Used problems and their optimum values are Oliver30 (423.74), Eil51 (428.87), Berlin 52 (7542/7544.37), Eil101 (642.31) In addition to problem Eil76 (545.39) [41]. Then comparison the results with best known solution which gives results without rounding [15, 18].

6.1. Comparison crossovers Cycle, GNX N=2, Uniform, SCX, OX, Shuffle, PMX and SPMX

To comparisons eight crossover Cycle, GNX N=2, Uniform, SCX, OX, Shuffle, PMX and SPMX. Result by ACSGA using different Crossover Operators for Olvir30, Eil51, Eil76, Eil101 and Berline52 as shown in Table 2.

From Table 2 it is seen that the order crossover, Cycle, GNX, Uniform, OX, SCX, shuffle, PMX and SPMX crossover. The proposed SPMX obtains the best solution with short distance for (Olvir30, Berline52, Eil76, and Eil101) problem, it gives the better result (short distance) for Eil101 problem.

The crossover OX gives the best solution (short distance) for Eil51problem and the the crossovers PMX obtains best solution for Eil76 problem as the proposed method.

These results are shown in Fig. 2 that show the crossovers OX, PMX and SPMX are competing, Cycle crossover is the worst. Also show the usefulness of crossover SPMX.

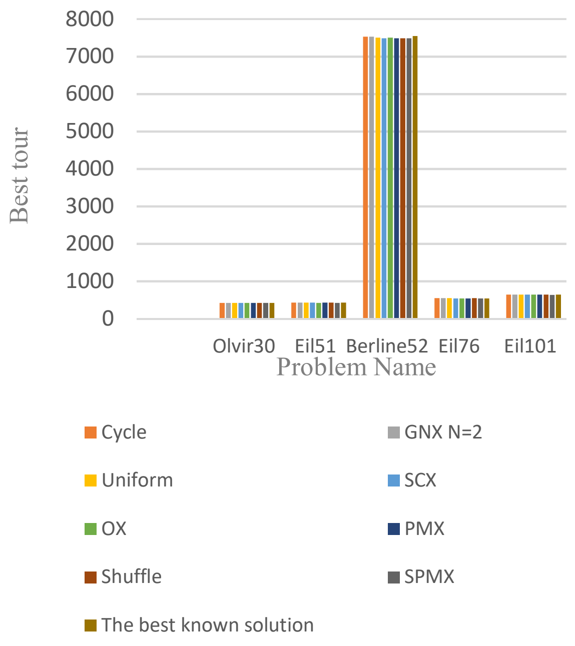


Fig. 2 Comparative Study of Eight Types for Crossover

6.2. Comparisons ACSGA that use SPMX with Different Algorithms

To Evaluate the proposed ACSGA that enhance by SPMX crossover by comparing the solutions with other methods such as GA (Collision crossover) [42], GA (CSCX) [43], SCGA [20], LevyACO [44], PDLSO [45], Multi-rules base SSPSO + 3-opt + GA [46], Nested hybrid ACS [18], MN2 [47], STASA [48] and ACSGA [15] and best known solutions from [15-18]. For example, TSPLIB data sets were used Shown in the Table 3 and Fig. 3.

From Table 3 and Fig. 3, the path length achieved by SPMX that enhance ACSGA is better than the other algorithms. All the above comparisons proved the high performance of using SPMX that enhance ACSGA on different TSPLIB.

Table 4 show the compared of the results with the best-known solutions of TSPLIB [20, 42-48] and quality
of solutions of proposed ACSGA (SPMX) has been made by calculating the percentage decrease which is defined by Eq. (8):

Percentage Decrease =

$$\frac{(\text{val. of best known sol.})-(\text{val. of proposed})}{|\text{val.of best known sol.}|} \times 100$$

----------------(8)

7. CONCLUSION

Crossover operators are classified as distance-based crossover operators. There are several crossover operators available in the study of literature. This paper proposes hybrid shuffle crossover and partially mapping (SPMX) to enhance ACSGA for the TSP. To demonstrate the utility of our proposed SPMX we applied distance-based crossover operators, such as Cycle, GNX, Uniform, OX, SCX, Shuffle, PMX and SPMX for five TSPLIB instance using ACSGA algorithm. The results show that the proposed method the crossover SPMX is the best for four TSPLIB instance (Olvir30, Berline52, Eil76 and Eil101) ,and gives better result for Eil101problem, the crossover OX obtains short distance for one TSPLIB instance(Eil51) and Cycle crossover is the worst.

This study aimed to develop the hybrid ACSGA to find high quality solutions to TSP. SPMX on ACSGA worked well for combinatorial optimization problems such as TSP when compared with other algorithms. In terms of solution quality and get a short distance, it is found that our proposed crossover SPMX is the best where the distance decreasing by 0.82% with average 0.47%, thus enhancing the results.

REFERENCES

[1] C. Yang, K. Szeto, “Solving the traveling salesman problem with a multi-agent system”, IEEE Congress on Evolutionary Computation, 2019, pp. 158-165.
[2] D. Little, G. Murty, W. Sweeney, and C.Karel, “An algorithm for the traveling salesman problem”. Operations research, Vol. 11(6), 1963, pp. 972-989.
[3] M. Padberg, G. Rinaldi, “Optimization of a 532-city symmetric traveling salesman problem by branch and cut”. Operations research letters, Vol. 6(1), 1987, pp. 1-7.
[4] Y. Deng, Y. Liu and D. Zhou, “An improved genetic algorithm with initial population strategy for symmetric TSP”. Mathematical Problems in Engineering, 2015, pp. 1-6.

[5] H. Zhou, M.Song, "An improvement of partheno-genetic algorithm to solve multiple travelling salesmen problem". IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1-6.

[6] S. Chowdhury, M. Marufuzzaman, H.Tunc, L. Bian, and W. Bullington, "A modified Ant Colony Optimization algorithm to solve a dynamic traveling salesman problem: A case study with drones for wildlife surveillance". Journal of Computational Design and Engineering, Vol. 6(3), 2019, pp. 368-386.

[7] Y. Lin, Z. Bian and X. Liu, "Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem". Applied Soft Computing, Vol. 49, 2016, pp. 937-95.

[8] W. Dewantoro , P. Sihombing, "The combination of ant colony optimization (ACO) and tabu search (TS) algorithm to solve the traveling salesman problem (TSP)". 3rd International Conference on Electrical, Telecommunication and Computer Engineering. IEEE, 2019, pp. 160-164.

[9] F.La Maire, M. Mladenov, "Comparison of neural networks for solving the travelling salesman problem". In 11th Symposium on Neural Network Applications in Electrical Engineering, IEEE, 2012, pp. 21-24.

[10] M. Dorigo, L. Gambardella, "Ant colony syste m: a cooperative learning approach to the traveling salesman problem". IEEE Transactions on evolutionary computation, Vol .1(1), 1997, pp.53-66.

[11] K. Thirugnanasambandam, R. Raghav, D. Saravanan, U., Prabu, and M. Rajeswari, "Experimental analysis of ant system on travelling salesman problem dataset TSPLIB". EAI Endorsed Transactions on Pervasive Health and Technology, Vol .5(19), 2019, pp. 1-15.

[12] Z. Ahmed, "Adaptive sequential constructive crossover operator in a genetic algorithm for solving the traveling salesman problem". GAs, Vol. 11(2), , 2020, pp. 594-605.

[13] J. Kaabi, Y. Harrath, "Permutation rules and genetic algorithm to solve the traveling salesman problem". Arab Journal of Basic and Applied Sciences, Vol. 26(1), 2019, pp. 283-291.

[14] H. Ali, M. Haris, F. Hadi, S. Ahmadullah and Y. Shah, "Solving Traveling Salesman Problem through Optimization Techniques Using Genetic Algorithm and Ant Colony Optimization". Journal of Applied Environmental and Biological Sciences, Vol. 6(4), 2016, pp. 55-62.

[15] E.Elsayed, A.Omar and K.Elsayed, "Smart Solution for STSP Semantic Traveling Salesman Problem via Hybrid Ant Colony System with Genetic Algorithm", International Journal of Intelligent Engineering and Systems, Vol. 3(5), 2020, pp. 476- 489.

[16] Z. Ahmed, "A Comparative Study of Eight Crossover Operators for the Maximum Scatter Travelling Salesman Problem", operations research, Vol. 11(6), 2020, 317-329.

[17] A. Hussain, Y. Muhammad and M. Sajid, "An Improved Genetic Algorithm Crossover Operator for Traveling Salesman Problem". Turkish Journal of Mathematics and Computer Science, Vol. 9, 2018, pp.1-13.

[18] S. Sahana, "Hybrid optimizer for the travelling salesman problem", Evolutionary Intelligence, Vol. 12(2), 2019, pp.179-188.

[19] A. Kumar, "Improved Genetic Algorithm to Solve Small Scale Travelling Salesman Problem".4th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2020, pp. 516-520.

[20] Y.Deng, J. Xiong, and Q. Wang, "A Hybrid Cellular Genetic Algorithm for the Traveling Salesman Problem", Mathematical Problems in Engineering, 2021, pp. 1-16.

[21] N. Sözen, "Solving Travelling Salesman Problem: A Hybrid Optimization Algorithm", 1st International Informatics and Software Engineering Conference (UBMYK). IEEE, 2019, pp. 1-6.

[22] A .Cinar, S. Korkmaz and M. Kiran, "A discrete tree-seed algorithm for solving symmetric traveling salesman problem", Engineering Science and Technology, an International Journal, Vol. 23(4), 2020, pp. 879-890.

[23] H. Bennaceur, E. Alanzi, "Genetic algorithm for the travelling salesman problem using enhanced sequential constructive crossover operator", International Journal of Computer Science and Security, Vol. 11(3), 2017, pp. 42-52.

[24] P.Paul, P. Dhavachelvan and R. Baskaran. "A novel population initialization technique for Genetic Algorithm", International Conference on Circuits, Power and Computing Technologies (ICCPCT), 2013, pp.1235-1238.

[25] A. Umbarkar, P. Sheth, "Crossover operators in genetic algorithms: a review", ICTACT journal on soft computing, Vol. 6(1), 2015, pp. 1083-1092.

[26] W. Hameed, A. Kanbar, "A Comparative

Study of Crossover Operators for Genetic Algorithms to Solve Travelling Salesman Problem", International Journal of Research, Vol. 5(2), 2017, pp. 284-291.
[27] D. Goldberg, R. Lingle, "Alleles, loci, and the traveling salesman problem", In: Proc. of an international conf. on genetic algorithms and their applications. Hillsdale, Vol. 154, pp. 154-159, 1985, pp. 154-159.
[28] M. Al-Omeer and Z. Ahmed, "Comparative study of crossover operators for the MTSP", International Conf. on Computer and Information Sciences, IEEE, 2019, pp. 1-6.
[29] S. Akter, N. Nahar, M. Shahadat Hossain, and K. Andersson, "A new crossover technique to improve genetic algorithm and its application to TSP", International Conf. on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2019, pp. 1-6.
[30] N. Radcliffe and P. Surry, "Formae and the variance of fitness. Foundations of Genetic Algorithms, Vol. 3, 1995, pp. 51-72.
[31] A. Silitonga, E. Nababan and O. Sitompul, "Reducing Image Noises Using Genetic Algorithm's Uniform Crossover", 2nd International Conf. on Informatics and Computational Sciences, IEEE, 2018 , pp. 1-6.
[32] K. Ishola, O. James, "Cost Reduction of Traveling Salesman Problem with an Enhanced Genetic Algorithm", International Journal of Research and Scientific Innovation (IJRSI), Vol. VII(I), Issue I, 2020, pp. 110-117.
[33] F. Burkowski, "Shuffle crossover and mutual information", In Proc. of the Congress on Evolutionary Computation-CEC99, IEEE, Vol. 2, 1999, pp. 1574-1580.
[34] S. Akter, M. Murad, R. Chaity, M. Sadiquzzamanand, and S. Akter, "Genetic Algorithm with Updated Multipoint Crossover Technique and its Application to TSP", IEEE Region 10 Symposium (TENSYMP), 2020, pp. 1209-1212.
[35] L. Whitley, T. Starkweather, and D. Fuquay, "Scheduling problems and traveling salesmen: The genetic edge recombination operator", In ICGA, Vol. 89, 1989, pp. 133-40.
[36] W. Gao, "New ant colony optimization algorithm for the traveling salesman problem. International Journal of Computational Intelligence Systems", Vol. 13(1), 2020, pp. 44-55.
[37] J. Renders, H. Bersini, "Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways", In: Proc. of the First IEEE Conf. on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 1994, pp. 312-317.
[38] U. Hacizade, I. Kaya, "Ga based traveling salesman problem solution and its application to transport routes optimization", IFAC-PapersOnLine, Vol. 51(30), 2018, pp. 620-625.
[39] V. Carvalho, F. Esteban, C. Johnson, F. Osorio, "Exploratory Path Planning Using the Max-Min Ant System Algorithm", 2016 IEEE Congress on Evolutionary Computation, 2016, pp. 1-7.
[40] G. Reinelt, "TSPLIB—A traveling salesman problem library, ORSA J. Comput", Vol. 3(4), 1991, pp. 376–384.
[41] A. Cinar, S. Korkmaz and M. Kiran, "A discrete tree-seed algorithm for solving symmetric traveling salesman problem", Engineering Science and Technology, an International Journal, Vol. 23(4), 2020, pp. 879-890.
[42] A. Hassanat, E. Alkafaween, "On enhancing genetic algorithms using new crossovers", International Journal of Computer Applications in Technology, Vol. 55(3), 2017, pp. 202-212.
[43] Z. Ahmed, "Genetic algorithm with comprehensive sequential constructive crossover for the travelling salesman problem", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 11(5), 2020, pp. 245-254.
[44] Y. Liu, B. Cao, "A novel ant colony optimization algorithm with levy flight", IEEE Access, Vol. 8, 2020, pp. 67205-67213.
[45] Z. Daoqing, J. Mingyan, "Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem", Journal of Systems Engineering and Electronics, Vol. 31(4), 2020, pp.751-760.
[46] I. Khan, S. Pal, and M. Maiti, "A hybrid PSO-GA algorithm for traveling salesman problems in different environments", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 27(05), 2019, pp. 693-717.
[47] I. Ahmia and M. Aider, "A novel metaheuristic optimization algorithm: the monarchy metaheuristic", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 27(1), 2019, pp.362-376.
[48] X. Han, Y. Dong, L. Yue, and Q. Xu, "State transition simulated annealing algorithm for discrete-continuous optimization problems", IEEE, 2019, pp. 44391-44403.

[49] N. Mouttaki, J.Benhra, and G. Rguiga, " Genetic Algorithm for Optimizing Distribution with Route Restriction Constraint due to Traffic Jams", The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 44, 2020, pp. 295-301.

Table 2. Comparative Study of Eight Types Crossover Based on ACSGA for Symmetric TSPLIB instances

Instance	Olvir30	Eil51	Berline52	Eil76	Eil101
Cycle	424.9	431.14	7524.54	549.82	648.3
GNX N=2	424.67	430.4	7523.98	549.06	645.96
Uniform	423.74	429.9	7500.84	548.16	645.96
SCX	423.95	431.9	7485.98	545.16	645.06
OX	423.74	**426.87**	7500.59	545.39	642.03
PMX	423.74	428.04	7482.49	**545.39**	641.93
Shuffle	423.74	434.75	7482.99	550.28	646.66
Proposed SPMX	**423.52**	427.04	**7482.44**	**545.39**	**639.95**
The best known solution	423.74	429.98	7544.37	545.39	642.03

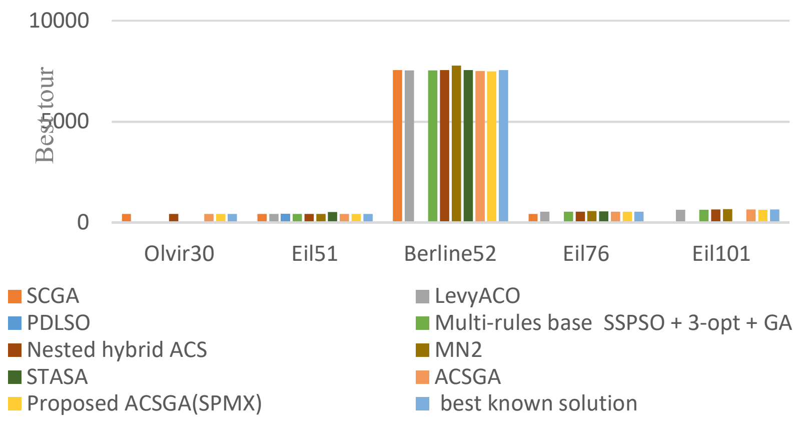


Fig. 3 A comparison of Tour Length for TSP on Different Algorithms and Proposed ACSGA with SPMX Crossover

Table 3. Comparison of Proposed ACSGA(SPMX) with GA(Collision crossover), GA(CSCX), SCGA, LevyACO, PDLSO,Multi-rules base SSPSO + 3-opt + GA, Nested hybrid ACS, MN2, STASA and ACSGA, and Show the Percentage of Decrease

Problem Name / Methods	Olvir30	Eil51	Berline52	Eil76	Eil101
GA(Collision crossover)(2017)	-	649	10224	-	-
GA(CSCX)(2020)	-	437	7646	-	-
SCGA(2021)	423.7406	428.8718	7544.3659	**428.8718**	-
LevyACO(2020)	**-**	426	7542	538	629
PDLSO(2020)	**-**	430.46	7 544.37	**-**	**-**
Multi-rules base SSPSO + 3-opt + GA(2019)	**-**	426.20	7542.0	538.08	629.40
Nested hybrid ACS(2020)	423.74	429.98	7544.37	545.39	642.31
MN2(2019)	**-**	431.17	7774.24	563.66	664.258
STASA(2019)	**-**	529.8	7544	554.5	-
ACSGA(2020)	423.74	426.87	7500.59	545.39	642.03
Proposed ACSGA(SPMX)	**423.52**	**427.04**	**7482.44**	**545.39**	**639.95**
The best-known solution	423.74	429.98	7544.37	545.39	642.03

Table 4. Show the Percentage of Decrease

Problem Name / Methods	Olvir30	Eil51	Berline52	Eil76	Eil101
Proposed ACSGA(SPMX)	**423.52**	**427.04**	**7482.44**	**545.39**	**639.95**
The best-known solution	423.74	429.98	7544.37	545.39	642.03
Percentage of decrease	0.51%	0.68%	0.82%	0%	0.32%
Average	.47%				