

# Bidirectional Constructive Crossover for Evolutionary Approach to Travelling Salesman Problem

Semin Kang  
Tongmyong University  
Busan, KOREA  
Email: semin@tu.ac.kr

Sung-Soo Kim  
ETRI  
Daejeon, KOREA  
Email: sungsoo@etri.re.kr

Jong-Ho Won  
ETRI  
Daejeon, KOREA  
Email: jhwon@etri.re.kr

Young-Min Kang  
Tongmyong University  
Busan, KOREA  
Email: ymkang@tu.ac.kr

**Abstract**—In this paper, we proposed an improved crossover method for genetic approach to travelling salesman problem (TSP). Since the valid solution of TSP is an ordered permutation, traditional crossover methods do not guarantee the valid offsprings and fast convergence. Therefore, researchers have proposed special crossover, and it is known that SCX is superior to other methods in the aspect of convergence speed and fitness of the genes. In this paper, we extend the SCX to have bidirectional and circular search properties in the off-spring construction. The proposed BCSCX shows the better convergence speed and even better solution than those of SCX in the empirical experiments.

## I. INTRODUCTION

Travelling Salesman Problem (TSP) is a problem with an objective to find a path that has the lowest cost. Rules are simple. Nodes that must be visited should be visited only once and there should be no missing nodes. However this optimization problem is known as a NP-complete. This means TSP has a huge search space so that it's very hard to find an optimal solution [1]. Since Holland noticed that principle of natural evolution can be a good method to solve optimization problem in his book *Adaptation in natural and artificial system*. Genetic Algorithms (GA) have been widely used to solve search and optimization problem

GA is a search algorithm based on the rules of evolution like natural selection and natural genetics[?]. It uses selection, crossover and mutation to improve a population of candidate solutions and iterate it until terminating condition is satisfied. TSP is applicable to various fields of industry, and there have been many attempts to solve TSP [?], [?], [?]. In this paper, we propose an improved crossover operator for better performance of GA-based approach to TSP.

## II. GENETIC ALGORITHM FOR TSP SOLVER

GA is a heuristic search algorithm that mimics principle of evolution. Evolution of a population of candidate solution is derived by evolutionary rules such as selection and crossover. Reproduction mimics Darwinian survival-of-the-fitness. Therefore, individuals that have relatively worse fitness value cannot participate in reproduction process. Fitness value in TSP is simple. The multiplicative inverse of the total cost of tour path can be used as the fitness value: The smaller the tour cost of a gene is, the better the gene is.

### A. Genetic Representation

In order to implement a GA-based TSP solver, solutions should be expressed as a feasible chromosome. A sequence

TABLE I. EXAMPLE OF CONSTRAINTS VIOLATION

table	placed	here
-------	--------	------

representing the order of city visit is a possible solution. Suppose, for example, there are four nodes 1,2,3,4. The tour path  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$  can then be represented with a chromosome expression  $\langle 1,2,3,4 \rangle$ . When using GA to solve TSP, the phenotype is, in most cases, used as a chromosome for crossover. Since TSP constraints the form of valid solution, the conventional crossover with genotype expression cannot be easily implemented, and it is more difficult to devise a genotype-based crossover which rapidly converges to a sufficiently good solutions. Therefore, we also used path sequence directly as a chromosome.

### B. Problems in Conventional Crossover

The solutions must contain all nodes to be visited only once and there must be no missing nodes. In conventional crossover child is generated by associating the parts of parents' chromosomes. However, as shown in Table ??, the conventional crossover may easily produce duplicate nodes and missing nodes.

In order not to violate the constraints, many kinds of unconventional crossover operators such as Genetic Edge Recombination Crossover (ERX), Generalized n-Point Crossover (GNX) and Sequential Constructive Crossover (SCX) have been proposed, and SCX shows the fastest convergence speed [?].

### C. Sequential Constructive Crossover(SCX)

The SCX crossover method guarantees validity of child's chromosome and conserve goodness of parents' chromosome. This method tries to reduce the local distance in the adjacent nodes in the chromosome of off-springs by sequentially scan the chromosomes of parents. The algorithm can be described as follows[?]:

- 1) Set the starting node 1 to be the current node  $p$ .
- 2) Find the two unvisited node  $a$  and  $b$  respectively from each parent's chromosome by sequentially search the first unvisited node (legitimate node) after the current node  $p$ . If the search failed, select any unvisited node from the city template permutation such as  $\langle 1, 2, \dots, n \rangle$ .

TABLE II. THE COST MATRIX

table	placed	here
-------	--------	------

- 3) Compare the distances from  $p$  to  $a$  ( $d_{pa}$ ) and to  $b$  ( $d_{pb}$ ). If  $d_{pa}$  is less than  $d_{pb}$ , add  $a$  to the off-spring's chromosome and set  $a$  to be the current node. Otherwise,  $b$  is added and set to be the current node. Then go to step 2.

### III. PROPOSED CROSSOVER METHOD: BI-DIRECTIONAL CIRCULAR SCX

In order to improve the performance of SCX, we propose bidirectional circular SCX (BCSCX) which can search the next possible legitimate nodes in the parents chromosomes in two directions and the chromosome is regarded a circular data with no ends.

#### A. Bi-directional Property

SCX operator proceeds crossover process in a single direction. In other words, it chooses the candidate nodes to be added to the chromosome of off-spring only from the nodes appearing after the currently visited node in parents chromosome sequence. For example, assume that a uncompleted chromosome sequence  $\langle 1,5 \rangle$  has been inherited from two parents during the crossover operation. Then the current node is node 5. If, within the chromosome sequence of one parent, the node 3 is the closest unvisited node ('closest' in the aspect of the location in the sequence string) among the nodes before the current node (the node 5), and the node 2 is the closest unvisited node after the current node in the sequence. The proposed BCSCX takes both nodes (3 and 2) as legitimate candidate nodes. Two more candidates are taken from the other parent's chromosome. Let us assume that the candidates from the other parent are node 2 and 6. The candidates for the next node in the off-spring's chromosome right after the node 5 are the union of the candidate sets (i.e., node 2, 3, and 6), and they are tested similarly as SCX selects the next node. If the cities are connected as described in Table 1,  $d_{56}$  is less than  $d_{52}$  and  $d_{53}$ . Therefore, in this case, the chromosome of the off-spring will grow to be  $\langle 1,5,6 \rangle$ .

#### B. Circular Property

SCX does not assume that chromosome strings are circularly concatenated. Therefore, if the last node of the chromosome string is the current node  $p$ , no candidate legitimate nodes can be obtained. In order to avoid this problem, SCX employed a pre-defined template. To solve this problem SCX uses pre-defined template. For example, assume that the chromosome of a parent is  $\langle 1,3,7,6,2,4,5 \rangle$  and that of the other is  $\langle 1,5,7,2,6,3,4 \rangle$ , and currently constructed partial chromosome of the off-spring is  $\langle 1,5 \rangle$ . The original SCX then fails to find the legitimate node from the first parent, and the first unvisited node (in this case, node 2) from the template  $\langle 1,2,3,4,5,6,7 \rangle$  will be selected as the legitimate node. However, our method regards the chromosome a circular data, and jumps to the first character so that the node 3 will be selected as the legitimate node.

## IV. EXPERIMENT AND ANALYSIS

In order to verify the validity and the efficiency of the proposed BCSCX method, we empirically compared the method with original SCX method. Since it is known that SCX method is better than any other crossover methods previously proposed, we did not implement other crossover methods such as edge recombination, generalized n-point crossover methods. In order for the empirical comparison, we devised 4 methods and named them as follows:

- **SCX**: The original SCX method proposed by Ahmed[?]. Two good parents are randomly selected, and two identical off-springs are created. One of the off-springs is immediately mutated after the creation.
- **TWSCX**: Two-way SCX, an improved version of SCX to produce two different off-springs. The first offspring  $C_1$  is created with SCX of two parents  $A$  and  $B$ , and another one  $C_2$  is the crossover of  $A$  and  $B'$ , the inverted string of  $B$  (i.e.,  $B_i = B_{n-1-i}$ ).
- **BCSCX**: Bidirectional circular SCX proposed in this paper.
- **mBCSCX**: Multi-population BCSCX. The population is subdivided into multiple gene pools, and BCSCX method is independently applied to each pool. After the evolution is performed, the best gene of each pool is copied to randomly selected pools.

For the experiment, we generated randomly located  $n$  cities on 2-D plane and the cost from a city to another was computed as Euclidean distance on the plane. Fig. ?? compares the convergence speed of the four methods in the early stage of the evolution (10 generations). In this experiment, 1024 cities were given and 100 genes were employed. For the mBCSCX method, the population was subdivided into 5 gene pools (i.e., each gene pool contains 40 genes). The BCSCX method showed the best convergence speed as shown in the figure. The temporary solutions of the methods (SCX, TWSCX, mBCSCX and BCSCX clockwise from top left) at the tenth generation are also shown in the figure. As shown in the figure, the BCSCX method not only converges fast but also produces better solution. does not guarantee the better convergence and it would be better to divide population into smaller groups to maintain the variety of genes and utilize the parallel computing power of current computer architecture. Fig. ?? shows the long-term evolution results. In this experiment, 100 genes are evolved with 1000 generations to find the route for 300 cities, and mBCSCX produced the best result. The convergence trends of the long-term evolution are also visualized as shown in Fig. ?? In this experiment, 50 genes are evolved to find the route for 100 cities. BCSCX converges fast but immediately stays at a local minimum. However, mBCSCX shows slightly slower convergence than BCSCX, but it continues to improve the result during the whole evolution period. Although the original SCX also continuously improves the result, its convergence speed is too slow compared to BCSCX or mBCSCX. TWSCX stays at a worse local minimum compared to BCSCX and mBCSCX.

## V. CONCLUSION

In this paper, we proposed an improved crossover operator (BCSCX) based on the sequential constructive crossover (SCX). Since TSP constrains the off-spring chromosome, many special crossover methods that do not violate the constraints have been proposed. Among those methods, it is known that SCX shows the best performance. However, SCX does not exploit the symmetry of the problem. When we handle symmetric TSPs, the search directions for legitimate nodes may be forward and backward, and the solution paths obtained from TSP are naturally circular. We extended SCX to take those aspects into account, and the proposed BCSCX showed better performance. Moreover, the accuracy and the efficiency of BCSCX-based evolution were not greatly influenced by the population size. Therefore, interestingly, multi-population version of BCSCX (mBCSCX) showed slightly slower convergence speed while it yielded better solutions than single-population BCSCX. Those results mean population size doesn't affect much on solving TSP. And the way of escape from local minima by keeping variety of population with multi-population is more important. This implies that BCSCX can be successfully applied to parallel processing frameworks to find better solution in shorter time.

## VI. ACKNOWLEDGMENT

This work was supported in part by ETRI R&D Program (Development of Big Data Platform for Dual Mode Batch-Query Analytics, 14ZS1400) and also in part by Brain Busan 21 (BB21) support program supervised by Busan Human Resources Development Institute.