The Cluster Crossover Operation for The Symmetric Travelling Salesman Problem

Ajchara Phu-ang[1] and **Duangjai Jitkongchuen**[2]

ABSTRACT

This paper proposed the new algorithm intended to solve a specific real-world problem, the symmetric travelling salesman problem. The proposed algorithm is based on the concept of the galaxy based search algorithm (GbSA) and embedded the new ideas called the clockwise search process and the cluster crossover operation. In the first step, the nearest neighbor algorithm introduces to generate the initial population. Then, the tabu list local search is employed to search for the new solution in surrounding areas of the initial population in the second step. The clockwise search process and the cluster crossover operation are employed to create more diversity of the new solution. Then, the final step, the hill climbing local search is utilized to increase the local search capabilities. The experiments with the standard benchmark test sets show that the proposed algorithm can be found the best average percentage deviation from the lower bound.

Keywords: Travelling Salesman, Cluster Crossover, Clockwise Search, Galaxy Based Search

1. INTRODUCTION

The travelling salesman problem (TSP) is categorized in the NP-Hard optimization problem field. The TSP has become very important in transportation businesses. In general, we found the TSP with the logistic planning, the traffic management and the airline crew landing. When discussing about transportation, the success of this kind of business will depend on the route planning. The company that controls the delivery using the least travel distance will be successful.

In the past year, there are many researchers that present algorithms to solve this problem; for instance. Mostafa et al. [1] presented some approaches to the application of hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. Eneko et al. [2] presented an improved discrete bat algorithm for TSP. Bat Algorithm (BA) is a metaheuristic algorithm for global optimization. It was inspired by the echolocation behaviour of microbats, with varying pulse rates of emission and loudness. The capability of echolocation of microbats is fascinating as these bats can find their prey and discriminate different types of insects even in complete darkness. The improvement in this paper have provided some kind of intelligence to all bats of the swarm. Thereby, each bat moves in a different way depending on its position in relation to the best bat of the swarm. When one bat is going to perform its movement, it has opportunity to assume that is far from the best bat of the swarm (large moves or short moves). This paper proposed 2-Opt for short moves, and the 3-Opt as large moves. Gustavo Erick Anaya Fuentes et al. [3] present the new metaheuristic algorithm called the CTSPMRILS. In the article, they combine the cluster technique , the NEH heuristic and the Multi Restart Iterated Local Search. At the beginning, they use the K-mean clustering algorithm to define a cluster of the cities. Then the NEH algorithm is applied to each of the clusters which created the smaller travel distance. Thereafter, they apply the MRSILS algorithm to each of the clusters from the initial solution generated by the NEH. This paper test the performance with 10 instances, the results in all the instances demonstrate that the proposed algorithm CTSPMRILS is more efficient than the genetic algorithms. Sebastian Meneses et al. [4] intended for solving the travelling salesman problem on tridimensional environments (TSP 3D-variation). This algorithm is contributed by the genetic algorithm. Because, the quality of the genetic algorithm depending on the quality of the initial population. Therefore, this study used a metaheuristic GRASP algorithm to generate the initial population. Then, the crossover operation and the mutation operation are applied. In the experiment, they compare the performance of the genetic algorithm with the GRASP versus the original genetic algorithm. The result shows that, the route obtained through the implementation of the genetic Algorithm is better than that obtained by the GRASP Algorithm to 13%. Jinmo Sung and Bongju Jeong [5] present the algorithm for solving the travelling salesman problem. They propose an adaptive evolutionary algorithm to improve the computational efficiency and the quality of the solutions. This paper contributed evolutionary algorithm liked

Manuscript received on June 8, 2018 ; revised on January 3, 2019.

Final manuscript received on January 27, 2019.

[1] The author is with the Department of Information Technology, Chandrakasem Rajabhat University, Thailand, E-mail: ajchara.p@chandra.ac.th

[2] The author is with the College of Innovative Technology and Engineering, Dhurakij Pundit University, Thailand, E-mail: duangjai.jit@dpu.ac.th

the genetic algorithm, include crossover and mutation processes along with adaptive search strategies. The ideas of the adaptive scheme is to adjust the values of instances to enhance the search efficiency. For the computational experiments, they compare the results of the proposed algorithm with the best known solutions. The comparison demonstrated that, the proposed approach outperforms other evolutionary algorithm in terms of the quality of solution and computation time. Yong WANG [6] solving the travelling salesman problem by using the genetic algorithm (GA). In the algorithm, they improve the GA with two local optimization strategies and call the proposed algorithm as the HGA. The second local optimization algorithm is used to reverse the local Hamiltonian paths in the mutation process. Thereafter, use the first local optimization algorithm designed based on the four vertices and three lines inequality to make the shorter solution. The computational results show that the hybrid genetic algorithm (HGA) found the better approximate solutions than the original GA. Ayman Srour et al. [7] presents a The water flow-like algorithm (WFA) for solving the travelling salesman problem. The WFA algorithm imitates the water flow, which flowing from higher to lower area. The flow can separate into sub flows and water flows merge with each other when they join at the same point. The initialization stage of the basic WFA is a random.While, in the proposed algorithm use WFA-TSP used the nearest neighbour heuristic for generating instead. Moreover, in the splitting and moving operation stage, they embedded the insertion method and 2-opt neighbourhood structure in order to find the best new solution. The performance of the proposed WFA-TSP is evaluated by conducting several experiments using 23 standard benchmark. The experimental results show that the proposed WFA found better solutions in terms of the average solution from the best-known solution. Thiago A.S.Masutti et al. [8] applied the Artificial Neural Networks and Artificial immune systems for solving the combinatorial problems, travelling salesman problem. The proposed algorithm is compared to other paper which employed the neural methods for solving the same problem. The comparison result shows that the proposed method is competitive to the other ones.

This paper aimed to propose the new ideas for solving the TSP. In our framework, the proposed is building on the galaxy based search algorithm. At the beginning, the nearest neighbor algorithm and the random rule were applied to generate the initial population. Then, the tabu list local search is employed to search for the promise solution in the surrounding area of the initial population. A new search methodologies called the clockwise search and the cluster crossover operation were introduced to enhance the search area efficiency. In the final step, the hill climbing local search used to increase the local search capability.

The remainder of this paper is organized as follows: in section 2, TSP is discussed in detail; in section 3, the original galaxy based search algorithm is described; in section 4, steps of our proposed algorithm are explained; in section 5, experimental results are shown; and section 6 is the conclusion of this paper.

2. THE TRAVELLING SALESMAN PROBLEM

The travelling salesman problem (TSP) is an NP-Hard combinatorial optimization problem. The objective of the TSP is to determine the sequence of the cities visited by a salesman. The objective function of this problem is to minimize length of salesman tour.

$$f(C) = \sum_{i=1}^{n} d_{i,i+1} + d_{n,depaeture} \tag{1}$$

$$d_{i,i+1} = \sqrt{(c_{ix} - c_{i+1x})^2 + (c_{iy} - c_{i+1y})^2} \tag{2}$$

A cost function of tour is defined as equation 1 where $\{d_{\mathrm{n,departure}}\}$ state as the Euclidean distance between the finally city (n) and the departure city. According to equation 2, it represent the Euclidean distance between the city number i (c_i) and the city number i+1 (c_{i+1}). Let c = $\{c_1,\ldots,c_n\}$ is a set of the cities. The condition of this problem is state as follows [9].

- The tour of a salesman can start at any city.
- Each city must be visited exactly once.
- A salesman always returns to the departure city.
- A salesman can travel between cities following to the specified route.

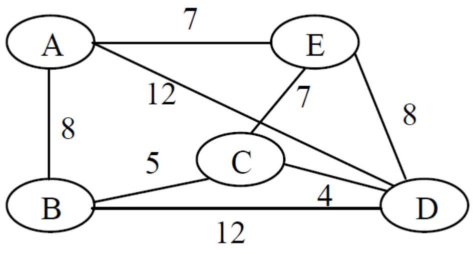

Fig.1: *A graph of the cities edges.*

Figure 1, shows the example of a set of the cities which consist of five cities c = {A,B,C,D,E}. There are various routes which the salesman can be visited, for the example,

Path 1: A,B,C,D,E,A

Path 2: A,E,C,D,B,A.

When calculating length of Path1 and Path 2 according to positive equation. Each path has a different of the total length, the length of the Path 1 is 32 and the Path 2 is 38.

3. THE GALAXY BASED SEARCH ALGORITHM

The Galaxy-based search algorithm (GbSA) is a new meta-heuristic algorithm has been proposed by Shahhoseini [10].

The GbSA is imitated the spiral arm of spiral galaxies. This algorithm has two main elements: Spiral Chaotic Move and Local Search. The Spiral Chaotic Move seem as exploration process and the Local Search seem as exploitation process.

The first stage of GbSA is started with initial L-dimension solution ($S = S_1, S_2, \ldots, S_L$). This solution is given to the Local Search process. Then the variable Flag is set to false then send the solution to Spiral Chaotic Move. The Spiral Chaotic Move process is performed to search the role neighbouring area of the current solution.

A chaotic variable is state as the Spiral Chaotic Move process that is generated by equation 3.

$$x_{n+1} = \lambda x_n(1 - x_n), \quad n = 0, 1, 2, \ldots \tag{3}$$

When the Spiral Chaotic Move found a solution that is better than the current solution, the current solution is updated with the improved solution then set variable Flag to true.

While the Flag variable is true, the Local Search process is started. This process searched solution from the nearest neighborhood. If the input space is L-D, the Local Search procedure receives an input vector as $S = S_1, S_2, \ldots, S_L$, and considers the nearest neighbors with radius r1 around the s1 component using equation 4.

$$S_{r_1} = S_1 \pm \alpha \times \Delta S \tag{4}$$

Where S1 is the first component of the current solution. α increases as the process runs by a constant parameter $\Delta\alpha$. ΔS is fixed during run of the Local Search process. At the same time, it changes during run of the GbSA algorithm by the NextChaos() function. Then NextChaos() function generates a chaos number between (0, 1).

When a better solution is found in the r1 neighborhood area, it is selected as the current best solution then the neighborhood radius increases and the search process is repeated with new radius, on the other hand, the process will be stops and the current best solution is presented as output. The whole process is recursive until a condition is satisfied.

4. THE PROPOSED ALGORITHM

This work present the new algorithm which based on the concepts of the galaxy based search algorithm. In the proposed algorithm, the tabu list local search is employed to improved the initial population. The new search technique called the clockwise search process and the cluster crossover operation is embedded.

```
-Randomly generate the initial population
-Apply the local search
-While loop
   Define Flag = False
        Use the SpiralChaoticMove
  Set Flag = True(when found the better population)
        If Flag = True
                Apply the local search
        End if condition
-End while loop
-End process
```

Fig.2: *The pseudo-code of the basic GbSA.*

The flowchart of the proposed algorithm is shown in Fig 4. Each stage in the chart is explained in detail below.

```
-Generate the initial population  by the nearest
  neighbor algorithm and the random rule
-Apply the tabu list local search
-While loop
   Define Flag = False
        Use the cluster crossover  operation
        and the clockwise search process
   Set Flag = True(when found the better population)
        If Flag = True
                Apply the hill climbing local search
        End if condition
-End while loop
-End process
```

Fig.3: *The pseudo-code of the proposed algorithm.*

4.1 The Solution Encoding

For the solution representation, the structure of the solution used in this paper consists of a series of the cities which the salesman need to tour. An example of the solution encoding is shown in Fig 5.

In Fig. 5, the solution consists of the sequence of cities 5, 4, 3, 2, 1 and 5 which can be explained as follows.

- The number 5 is a representation of the city named E, determined as the departure city.
- The number 4 is a representation of the city named D, acts as the 2nd city which a salesman toured.
- The number 3 is a representation of the city named C. The salesman must travel to the C as the 3rd city.
- The number 2 is a representation of the city named B, it is the 4th city which a salesman reached,
- The number 1 is a representation of the city named A, acts as the 5th city that the salesman need to travel.

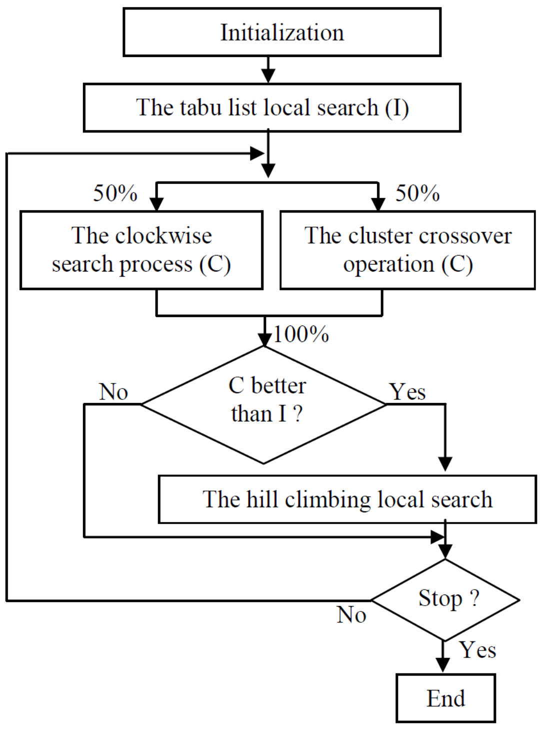
Initialization
The tabu list local search (I)
50%
50%
The clockwise search process (C)
The cluster crossover operation (C)
100%
C better than I ?
No
Yes
The hill climbing local search
Stop ?
No
Yes
End

Fig.4: *The flow chart of the proposed algorithm.*

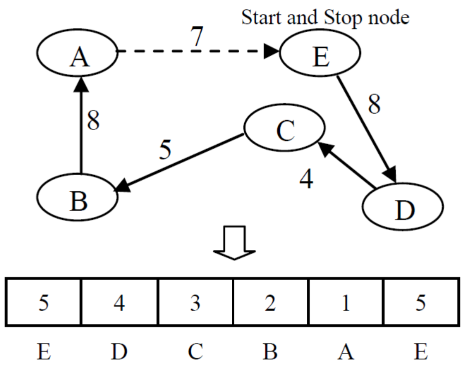

Fig.5: *The example of solution encoding method.*

- The number 5 is shown again, because the salesmen always return to the departure city.

To evaluate quality of the solution, the Euclidean distance between two cities is executed according to the equation 2. Then plus all the Euclidean distances according to equation 1.

4.2 Initialization

In the initialization process, each of the initial solution is contributed by randomly selecting between the nearest neighbor algorithm and the random selection rule. The procedure of this step can be explained as follows, firstly, we randomly pick the number between zero and one. If the random number is zero, the nearest neighbor algorithm is employed to generate the initial solution. The random selection rule is chosen when the random number equal to number one.

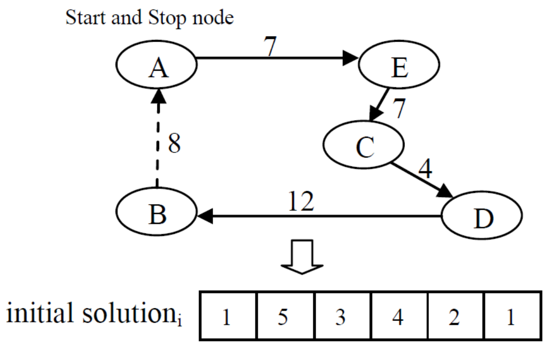

Fig.6: *The example of the nearest neighbor algorithm.*

- At the beginning of the nearest neighbor algorithm, randomly selected the departure city. Secondly, calculate the distance between the departure city and other city that have not been toured yet. Then the city which minimum distance is selected as the next city. This process repeats until all cities are visited.
- The following step is the example when used the nearest neighbor algorithm to generate the initial population.
  1. In figure 6, randomly selected the departure city. In this case, select A as the starting city.
  2. Calculate the distance between A:B, A:D, A:E
  3. Because, the distance of A to E is the shortest path . Therefore, E is selected as the 2nd city.
  4. Calculate the distance between E:C, E:D
  5. Defined C as the 3rd city.
  6. Choose the city which is next closest to that city having the minimum distance.
  7. Repeat this procedure until all the cities have been visited.
  8. The resulting path is {A, E, C, D, B, A } as show in figure 6.
- For the random selection rule, it's not complicated and unconditional. Firstly, we randomly selected the departure city. Secondly, randomly choose the 2nd city. Then, select the 3rd city. This process repeats until all cities are chosen.

4.3 The tabu list local search algorithm

This paper uses the tabu search algorithm proposed by Fred W. Glover [11] to find the promise solution in the surrounding area of the initial solution and prevent the algorithm select the same city repeatedly. In this stage, the tabu list is used to record a set of the cities which have been randomly selected. The rationale behind this idea is to prevent the algorithm select the same city repeatedly. The step of this process can be explained as follows.

At the beginning, uses the new swap technique find the promise solution$_i$ in enclosing space of the initial

solution_i as shown in Fig 7. The procedure of the new swap technique is as follow.

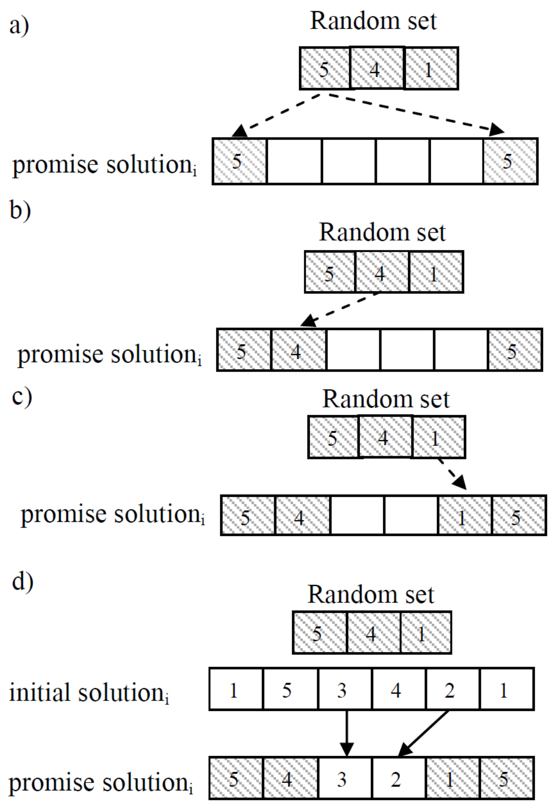

Fig. 7: *The example of the new swap technique.*

1. Randomly selected a set of the cities. In the example, the random set is R = 5,4,1
2. Use the 1st city of the random set as the starting city and the end city of the tour. In this case, the city number 5 is used as the departure and the end city of the promise solution_i as shown in fig. 7a.
3. Pick the city number 4 of the random set as the 2nd city of the promise solution_i as shown in fig. 7b.
4. Set the city 1 of the random set as the end of the tour as shown in fig. 7c.
5. Insert the city of the initial solution_i which is not a member of the random set to the blank position of the promise solution_i from left to right as shown in fig. 7d.

When already been created the promise solution_i, save the random set in the tabu list. Mean that, this random set will not be selected again for five iterations of this stage.

When the promise solution_i is better than the initial solution_i, it replaces the original. Then, the process is ended. Otherwise, if the promise solution_i is worse than the initial solution_i, uses the new swap technique searching for the promise solution_{i+1} in neighboring areas of the initial solution_i. Then the better solution is survivors. This process repeats until met the predefined criteria.

After already been applied the tabu list local search algorithm, the initial population is updated.

In addition, the initial solution is randomly divided into 2 groups. The initial solution in the first group is transferred to the clockwise search process and another group is sent to the cluster crossover operation.

4.4 The clockwise search process

This paper proposed the clockwise search process to deeply explore for the novel solution in surrounding areas of the initial solution from the previous stage. The search process is similar to the clockwise rotation, it slightly changes some city of the solution with the hope that the total length of the solution will decrease.

At the beginning, the initial solution which transferred to this process is defined as the clock solution.

Fig.8 show the example of the first iteration of the clockwise search process, The novel solution_i is created by copying all cities from the clock solution_i, then swap the city in the last position (city 1) with the city in the first position (city 2).

Next, created the novel solution_{i+1} by pick the city of the nearly last position of the novel solution_i (city 2) to the second position. Then, move the city in the second position (city 3) for placing into the defined position. The novel solution_{i+2} is created by moving the city in the third position (city 5) to the defined position. Thereafter, move the city in the nearly last position (city 3) instead the city in the third position.

The clockwise search process has been repeatedly until all the cities already been moved. However, if all the cities moved and not found the satisfy solution. The first step of the clockwise search is repeated again with the novel solution_{i+2}.

When the clockwise search process is terminated. All novel solution which better than the clock solution will be a survivor. Transferred all of the clock solution and the survivor to the 4.6 stage.

4.5 The cluster crossover operation

This paper presents the new crossover operation called, the cluster crossover operation. The aim of this process is to inherit the strong solution in order to keep it alive as long as possible and to enhance the diversity of the new solution by combining the cities of the initial solution in different cluster. Fig. 9 show the example of the cluster generation process and the details of the cluster crossover operation is explained as follow.

Step1: Start a crossover operation by sort the initial solution in descending order, then define the best of the initial solution as the alive solution. I addition, determined the initial solution which transferred to this process as the dig solution.

Step2: Randomly selected a four of the dig solution. After that, determined the above solution as

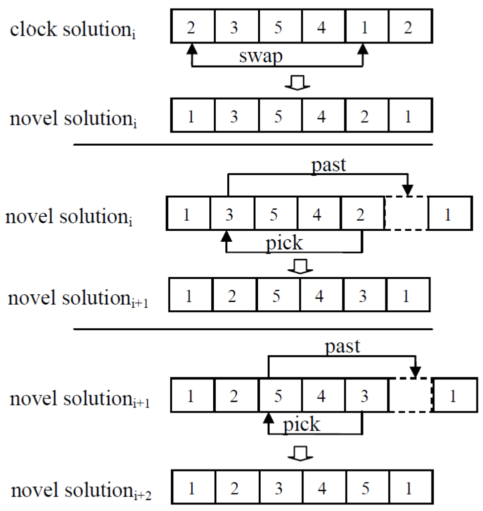

Fig.8: *The example of the clockwise search technique.*

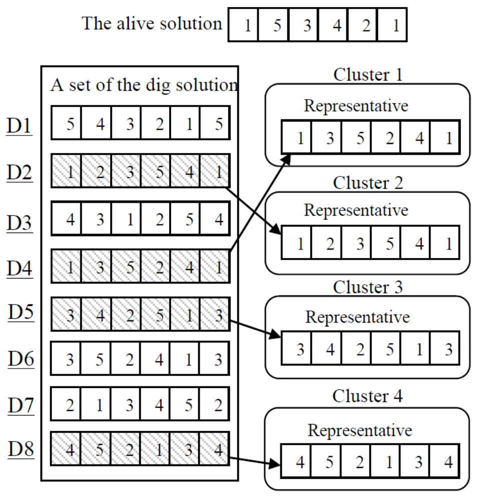

Fig.9: *The example of the cluster generation process.*

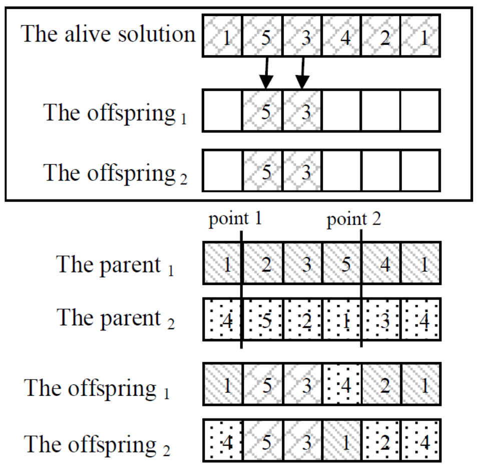

Fig.10: *The example of the N-point crossover operation.*

the representative solution of each cluster as shown in fig.9.

Step3: Calculated the distance between the dig solution$_j$ and the representative solution of each cluster by using equation 1. Consequently, the dig solution which a cities sequence is similar will be allocated to identical cluster. The dig solution$_j$ will be assigned to be a member of the nearest cluster.

Step4: A pair of the parent solution is chosen by random selected the dig solution from different cluster. According to this concept, a set of parents whose a sequence of the cities is dissimilar have more chance of getting crossed over. The rationale behind this idea is to increase the diversity of the offspring solution.

Step5: To create the offspring solution, randomly chosen the cities of the alive solution to inherit to the offspring solution. At the same time, the N point crossover operation is applied to parent pairs solution. as shown in fig. 10.

In fig. 10, the cities which inherit from the alive solution to the offspring solution is cities: [5,3]. The parent solution$_1$ (D2) and the parent solution$_2$ (D8) is randomly selected from cluster 2 and cluster 4 respectively.

This process is stopped when all solution in each cluster has already been crossed over. Then, transferred the parent solution and the offspring solution to the 4.6 stage.

4.6 The hill climbing local search algorithm

The hill climbing local search algorithm used in this stage is to find the neighbor solution in surrounding area of the target solution. The descriptions of this process are as follows:

At the beginning, the novel solution, the clock solution, the parent solution and the offspring solution are combined together. Then assigned to be the target solution.

Next, divided each of the target solution into N group. After that, applied the insertion technique proposed by Fogel [12] to find the neighbor solution$_i$ in enclosing space of the target solution$_i$ as shown in fig. 11.

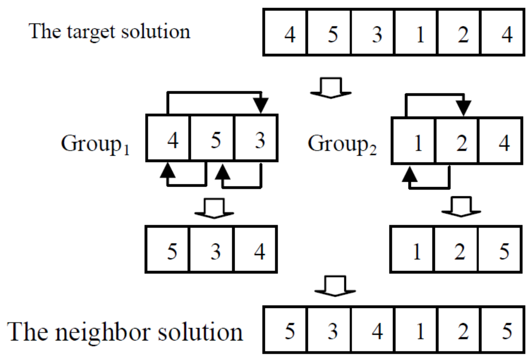

Fig.11: *The example of insertion technique.*

When the neighbor solution$_i$ is fitter than the target solution$_i$, replace the target solution$_i$ by the neighbor solutioni.Otherwise, if the neighbor solution$_i$ is worse than the target solution$_i$, uses the insertion technique searching for the neighbor solution$_{i+1}$ in neighboring areas of the neighbor solution$_i$. Then compare it again with the target solution$_i$. Mean that, the better solution is survivors. The insertion technique stops either when it finds a satisfying neighbor solution or the iteration is reached to a predefined value.

4.7 The final process

The best N number of the target solution is selected as the half of the initial solution used in the next iteration. While this work generates the other half of the initial solution by randomly. Then, the algorithm goes back to the 4.3 stage The proposed algorithm terminated when the predefined criteria is reached.

5. THE EXPERIMENTAL RESULT

This following presents the performance of the proposed algorithm on twelve of standard benchmarks. The performance test was run on an Intel Core i7 2.80 GHz processor with 8 GB of RAM and a 64-bit operating system.

The twelve sets of benchmarks obtained from the TSPLIB consist of Eli51, Eli76, Berlin52, kroA100, kroA150, kroA200, kroB100 kroB150, kroB200, kroC100, kroD100 and kroE100.

The performance of the proposed algorithm was compared to three other meta-heuristic algorithms detailed in the literature review. This paper evaluates the performance of the proposed algorithm in terms of average percentage deviation from the best known solutions for symmetric TSPs.

The rationale behind selected the three algorithm named HGA[6], WTF[7] and REBNET[8] because the above three algorithms are a recent paper tested with the same data benchmarks as the proposed algorithm.

Table 1: *The experimental results.*

Instances	**Experimental Result**				
	Best Known Solution	***Proposed***	***HGA [6]***	***WFA [7]***	***REB NET[8]***
Eli51	426	427	428	426	427
Eli76	538	540	544	538	541
Berlin52	7542	7542	7544	7542	7542
kroA100	21282	21283	21285	21282	21333
kroA150	26524	26526	26524	26524	26678
kroA200	29368	29370	29369	29368	29600
kroB100	22141	22141	-	22141	22343
kroB150	26130	26135	26130	26160	26264
kroB200	29437	29520	29450	29634	29637
kroC100	20749	20749	20750	20749	20915
kroD100	21294	21309	21294	21342	21374
kroE100	22068	22073	-	22106	22395
Avg. Dev.		0.100	0.168	-	-
BKS (%)		0.085	-	0.099	0.597

Table 1 illustrates the summaries experimental results of the proposed algorithm with the other algorithm on the twelve test sets. The column labeled with “Best Know Solution” displays the minimum length of each problem data set. The row labeled with “Avg. Dev. BKS (%)” reports the average percentage deviation from the best known solution.

Their reported best known solution were used:

- The Eli51 test set is composed of 51 cities. The reported best known solution is 426, the proposed algorithm reached the path length is 427

- The Eli76 test set is composed of 76 cities. The reported best known solution is 538, the proposed algorithm reached the path length is 540

- The Berlin52 test set is composed of 52 locations. The reported best known solution is 7542, the proposed algorithm reached the path length is 7542

- The kroA100 test set is composed of 100 cities. The reported best known solution is 21282, the proposed algorithm reached the path length is 21283

- The kroA150 test set is composed of 150 cities. The reported best known solution is 26524, the proposed algorithm reached the path length is 26526

- The kroA200 test set is composed of 200 cities. The reported best known solution is 29368, the proposed algorithm reached the path length is 29370

- The kroB100 test set is composed of 100 cities. The reported best known solution is 22141, the proposed algorithm reached the path length is 22141

- The kroB150 test set is composed of 150 cities. The reported best known solution is 26130, the proposed algorithm reached the path length is 26135

- The kroB200 test set is composed of 200 cities. The reported best known solution is 29437, the proposed algorithm reached the path length is 29520

- The kroC100 test set is composed of 100 cities. The reported best known solution is 20749, the proposed algorithm reached the path length is 20749

- The kroD100 test set is composed of 100 cities. The reported best known solution is 21294, the proposed algorithm reached the path length is 21309

- The kroE100 test set is composed of 100 cities. The reported best known solution is 22068, the proposed algorithm reached the path length is 22073

The followings are summaries of what we observed from the comparison results;

For Berlin52, kroB100 kroC100 and kroE100, the proposed algorithm come in first place.

For Eli51, Eli76, kroA100 and kroA150, the WFA found the best answer while the proposed algorithm come in second place.

The HGA comes in first place with kroB150, kroB200 and kroD100 then followed by the proposed algorithm.

For kroA200, the WFA comes in first place and follow by the HGA. The proposed algorithm found the result at third place.

When considered all instances together met that, the average percentage deviation from the best known solution of the proposed algorithm is the best among the comparison algorithms at 0.085%. In addition, when test with 10 instances found the average percentage deviation from the best known at 0.100 better than the HGA.

6. THE CONCLUSION

For solving the symmetric travelling salesman problem, this paper utilized the galaxy based search algorithm with the new idea called cluster crossover operation and the clockwise search process. The rationale behind this idea is to deep search in the promise areas and to enhance the diversity of the solution. In addition, the proposed algorithm uses the tabu list local search to prevent the algorithm repeated to the visited position. Finally, the hill climbing local search algorithm is utilized to enhance the neighbor search capabilities. The result shows the performance of the proposed algorithm when test with the test data.

References

[1] M. Mahi, O. K. Baykan and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing*, vol. 30, pp. 484-490, 2015.

[2] E. Osaba, X. S. Yang, F. Diaz, P. Lopez-Garcia and R. Carballedo, "An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems," *Engineering Applications of Artificial Intelligence*, vol 48, pp. 59-71, 2016.

[3] G. E. A. Fuentes, E. S. H. Gress, J. C. S. T. Mora and J. M. Marín, "Solution to travelling salesman problem by clusters and a modified multi-restart iterated local search metaheuristic," *PLoS ONE*, vol. 13(8), pp.1-20, 2018.

[4] S. Meneses, R. Cueva, M. Tupia and M. Guanira, "A Genetic Algorithm to Solve 3D Traveling Salesman Problem with Initial Population Based on a GRASP Algorithm," *Journal of Computational Methods in Sciences and Engineering*, vol. 17, pp. 1-10, 2017.

[5] J. Sung and B. Jeong, "An Adaptive Evolutionary Algorithm for Traveling Salesman Problem with Precedence Constraints," *The Scientific World Journal*, vol.2014, pp.1-14.

[6] Y. Wang, "The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem," *Computers & Industrial Engineering*, vol. 70, pp. 124-133, 2014.

[7] A. Srour, Z. A. Othman and A. R. Hamdan, "A Water Flow-Like Algorithm for the Travelling Salesman Problem," *Advances in Computer Engineering*, vol. 2014, pp. 1-14.

[8] T. A.S. Masutti and L. N. de Castro, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem," *Information Sciences*, vol. 179, pp. 1454-1468, 2009.

[9] R. Matai, S. Singh and M. Lal Mittal, Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9, InTech., 2010.

[10] H. Shah-Hosseini, "Principal components analysis by the galaxy based search algorithm: A novel metaheuristic for continuous optimization," *Int. J. Computational Science and Engineering*, vol. 6, pp. 132-140, 2011.

[11] Fred Glover, "Tabu Search - Part 2," *ORSA Journal on Computing*, vol. 2, no.1, pp. 4-32, 1990.

[12] D.B. Fogel, "An evolutionary approach to the travelling salesman problem," *Biological Cybernetics*, vol. 60, pp.139-144, 1988.

Ajchara Phu-ang received the Ph.D. degree from king mongkut's institute of technology ladkrabang, Bangkok, Thailand, in Information Technology. Her worked in university, where her is now an lecturer in the Information Technology Department, Chandrakasem Rajabhat University. Her teaching and research involves discrete optimization techniques, evolutionary algorithms, memetic algorithms and computer intelligence.

Duangjai Jitkongchuen graduated Ph.D. in Information Technology from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 2014 and now she is working at Dhurakij Pundit University, College of Innovative Technology and Engineering as a deputy dean administration. Her research area is about machine learning, data mining and bio-inspired algorithms.