

Effect of the initial population construction on the DBMEA algorithm searching for the optimal solution of the traveling salesman problem

Ali Jawad Ibada^{1*}, Boldizsár Túű-Szabó², and László T. Kóczy³

Abstract—There are many factors that affect the performance of the evolutionary and memetic algorithms. One of these factors is the proper selection of the initial population, as it represents a very important criterion contributing to the convergence speed. Selecting a conveniently preprocessed initial population definitely increases the convergence speed and thus accelerates the probability of steering the search towards better regions in the search space, hence, avoiding premature convergence towards a local optimum. In this paper, we propose a new method for generating the initial individual candidate solution called Circle Group Heuristic (CGH) for Discrete Bacterial Memetic Evolutionary Algorithm (DBMEA), which is built with aid of a simple Genetic Algorithm (GA). CGH has been tested for several benchmark reference data of the Traveling Salesman Problem (TSP). The practical results show that CGH gives better tours compared with other well-known heuristic tour construction methods.

Index Terms—Traveling Salesman Problem, Discrete Bacterial Memetic Evolutionary Algorithm, Genetic Algorithm, Nearest Neighbor heuristic, Second Nearest Neighbor heuristic, Alternating Nearest Neighbor heuristic, Circle Group Heuristic.

I. INTRODUCTION

THE Traveling Salesman Problem (TSP) is one of the most prominent members of the rich set of well-known combinatorial optimization problems with real life application potential. It is a Nondeterministic Polynomial hard (NP-hard) problem [1]. Given a set of cities (graph nodes) along with the costs of travel between each pair of them (the costs or lengths assigned to the edges), the TSP goal is to find the cheapest (shortest) way of visiting all the cities exactly once, and then returning to the starting point [1][2].

It must be realized that NP-hard problems are intractable (see e.g. [3]) and thus, there is no algorithm that gives guaranteed exact solution for them within a predictable time, nevertheless, there may be partially successful and guaranteed, but approximate solution methods constructed. Over the decades, there have been numerous approaches proposed in order to find the optimum (shortest, least cost) route. They may be classified to three classes: exact solution methods,

algorithms for approximate solution and heuristic approaches [4][5]. The Christofides algorithm is the most well-known approximation algorithm [6], which may be however, rather imprecise as the guaranteed solution may be maximally 50% greater than the global optimum. The most efficient heuristic solver so far is Helsgaun’s implementation of the classic Lin-Kernighan heuristic [7]. Many meta-heuristic researches have been published to find optimal or near-optimal solutions for the TSP; such as the Genetic Algorithm [8], the Ant Colony Optimization Algorithm [9], the Bacterial Evolutionary Algorithm (BEA) [10], the Particle Swarm Algorithm [11], Artificial Bee Colony [12], and their respective memetic versions [13]. In the next, we will only deal with a chosen, very efficient heuristic algorithm, the discrete memetic version of the Bacterial Evolutionary Algorithm (DBMEA) [14].

Returning to the matter of initial population selection, let us summarize that each initial population represents a feasible solution which is then subsequently improved over the course of several iterations through a heuristic (e.g., evolutionary) process [15]. The quality of the initial population of an evolutionary algorithm is rather important as it affects the search for the next (often numerous) generations, and has a significant influence on the quality of the final solution [16][17]. Improvement efforts on the initial population have shown to be effective in reducing the number of generations utilized while also improving the quality of the solution [18-23].

II. THE TRAVELING SALESMAN PROBLEM

The task of the TSP is to find a route through a given set of cities with the shortest possible length (cost). Mathematically, it means to find the shortest Hamiltonian tour in a graph [1].

$$G_{TSP} = (V_{cities}, E_{connections})$$

$$V_{cities} = \{v_1, v_2, \dots, v_n\}, E_{conn} \subseteq \{(v_i, v_j) \mid i \neq j\}$$

$$C: V_{cities} \times V_{cities} \rightarrow R, C = (c_{ij})_{n \times n} \quad (1)$$

Where: C is called the cost matrix, c_{ij} represents the cost of going from city i to city j .

The goal is to find an optimal permutation of vertices $(p_1, p_2, p_3, \dots, p_n)$ that gives the minimum total cost [24].

$$\text{Minimize } \left(\sum_{i=1}^{n-1} C p_i, p_{i+1} \right) + C p_n, p_1 \quad (2)$$

In general, the TSP can be classified into two different kinds, the Symmetric Travelling Salesman Problem (STSP) and the Asymmetric Travelling Salesman Problem (ATSP).

^{1,3} Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary.

^{2,3} Department of Information technology, Széchenyi István University, Győr, Hungary.

The distance between cities A and B is identical to the distance between cities B and A in the STSP. However, with the ATSP, it is possible to have two different costs or distances between two cities, depending on the direction. This may be realistic if, e.g., the altitude of the two places is different, and thus climbing would take more costs (time, fuel, etc.) than descending on the same route. Hence, the number of tours in the ATSP and STSP on n vertices is $(n-1)!$ and $(n-1)!/2$, respectively [25].

As mentioned above, due to the combinatorial complexity of the TSP, in practice, for larger instances (graphs), only approximate and/or heuristic procedures are applied in searching for the solution [26].

The TSP can be applied to a wide range of real life discrete optimization problems, especially in logistics, planning, and microchip manufacturing [24].

III. THE DISCRETE BACTERIAL MEMETIC EVOLUTIONARY ALGORITHM

In this section, a heuristic optimization approach will be briefly introduced which has proven rather efficient for a wide family of TSP related problems (mostly, extensions of the original TSP towards more realistic – and more complex – cases), while having two additional advantageous properties: the algorithm is generally applicable with high efficiency, and the runtime is rather predictable in terms of the size of the problem (number of nodes in the graph) [27]. Thus, even though for the basic TSP there is a better heuristic known, but it is very tailor made and not applicable for any other related optimization, and so we intend to investigate a complex approach which will be later extendable for other similar NP-hard problems.

Memetic algorithms extended the idea of using evolutionary algorithms for global search with nested local search methods, originally coming from more traditional mathematics. In each iteration for the individuals a local search step is applied [13]. DBMEA is a memetic algorithm that combines the very efficient Bacterial Evolutionary Algorithm as a global optimization with a simple combinatorial local search technique [25]. The drawbacks of both techniques, namely, the tendency to get stuck in a local optimum typical for the traditional (local) search techniques; and the very slow convergence speed of the outer (global) search cycle, are eliminated with this combined method. The evolutionary algorithms examine the global search space, and thus they only give a quasi-optimal solution because of their relatively slow convergence speed. Local search methods search only a certain neighborhood of the current candidate for solution, so they always converge to the closest local optimum; while their convergence speed is much faster. DBMEA was found to be rather efficient in solving a series of discrete nondeterministic polynomial-time hard optimization problems [25][26]. The DBMEA has four stages: initial population creation, bacterial mutation (coherent segment mutation and loose segment mutation), local search (2-OPT and 3-OPT), and gene transfer which is cyclically repeated.

1) Initial population creation step

Efficiency of an optimization algorithm is judged by its accuracy and speed. So, it is important to reach the (quasi-optimum) as fast as possible. The creation of the initial population can be crucial in reaching acceptably accurate solutions faster. In the literature, the initial population is often created randomly, but there are some deterministic approaches as well. Random creation guarantees the uniform distribution of the population in the search space. In the work of our group, several deterministic approaches were investigated [28]. The following heuristic construction algorithms are worth mentioning:

A. Nearest Neighbor (NN) heuristic

It constructs a tour in which, in the next step, the nearest unvisited city will always be visited. NN is easy to implement and fast to execute.

B. Secondary Nearest Neighbor (SNN) heuristic

It always visits the second nearest unvisited city in the next step of the tour.

C. Alternating Nearest Neighbor (ANN) heuristic

It combines the NN and SNN methods, here, the nearest and the second nearest unvisited cities are visited next in an alternating manner.

Among these three, the best convergence speed in most cases was achieved by the NN approach.

2) Bacterial Mutation Step

During this stage, each bacterium in the population is treated separately. A certain number of (identical) clones are made from the original bacterium (N_{clones}). Then, the bacterium and its clones are subdivided into chromosomes with a fixed length (I_{seg}). There are two semantic types of chromosomes (segments): coherent segments and loose segments. One from the segments of the bacterium is selected randomly and is modified in a clone, while the same gene in the original bacterium remains unaltered. So, it goes on with all the other clones as well. There is a particular clone in DBMEA; it contains the reverse order of the selected segment. Figure 1 shows the process of the clone creation.

The next step is the evaluation of the fitness values. As in the case of the TSP, traditional mathematics offers a possibility to determine the lower bound of the route length, based on the spanning tree of the whole graph, in this case the fitness function is obtained from the difference of the candidate solution individual from the theoretical lower bound, thus, the accuracy of all this way obtained clone bacteria (including the original). If one of the clones is better than the original bacterium, the mutated segment of the better clone is copied back to the original bacterium and to all the other clones. This process is consecutively applied until all the genes of the original bacterium have been mutated.

3) Local Search Step

During this step, individual improvement is carried out. The approach uses the exhaustive investigation of rearranging

Effect of the initial population construction on the DBMEA algorithm searching for the optimal solution of the traveling salesman problem

Original bacterium	...	3	6	2	5	1	4	...
1. clone (reverse order)	...	3	5	2	6	1	4	...
2. clone	...	3	2	6	5	1	4	...
3. clone	...	3	5	6	2	1	4	...
4. clone	...	3	6	5	2	1	4	...

Fig. 1. Clones creation in mutation stage

DBMEA, 2-opt and 3-opt could improve the individual with bounded size sub-graphs, optimizing them locally. Our group has investigated 2-opt and 3-opt local search for reasonable time, and we found that subsequently carried out 2-opt and 3-opt local search cycles are useful, and so, they are applied [28].

A. 2-opt local search

To shorten the TSP tour, in this simple method, two edge pairs in the original graph are exchanged. Assume we have two edge pairs, *AB* and *CD*; then these two will be replaced with *AC* and *BD* edges, resulting in a new potential tour. The truth of the following inequality is examined in the case of the new tour:

$$|AB| + |CD| > |AC| + |BD| \tag{3}$$

If the inequality is true, the edge pairs are swapped; the *AB* and *CD* edges are removed from the graph and replaced with *AC* and *BD*, as illustrated in Figure 2. The 2-opt move requires reversing one of the sub-tours between the initial edges. This iterative process is terminated if no further improvement can be made.

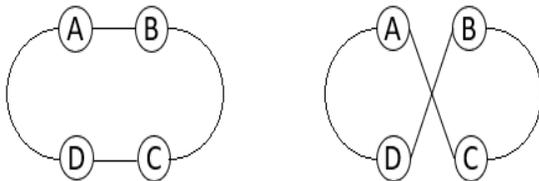


Fig. 2. Example for 2-opt local search

B. 3-opt local search

In this method, three edges are replaced with three others, producing eight alternative ways to reconnect the TSP tour, however, four of them have already been checked as they distort into 2-opt steps, therefore they are not considered here. The possible new replacements in the 3-opt local search are shown in Figure 3.

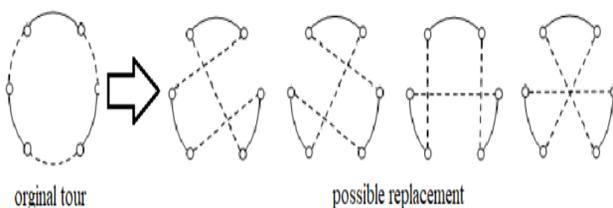


Fig. 3. Possible replacement of 3-opt local search

4) Gene transfer

In this stage, the population is initially sorted in decreasing order according to their fitness values, which are then sorted and separated into two (a superior and an inferior half). The operator repeats the following N_{inf} times: it picks one random bacterium (source bacterium) from the superior half and another random bacterium (destination bacterium) from the inferior part. Then it transfers some randomly picked segments with a pre-defined length ($l_{transfer}$) from the source bacterium to the destination bacterium. The bacterium length will not alter since the duplicate occurrences will be removed. Figure 4 shows the segment transfer in the gene transfer stage.

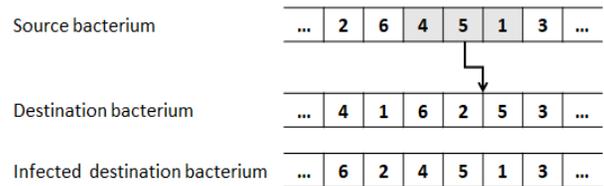


Fig. 4. Gene transfer

IV. THE GENETIC ALGORITHM

Genetic Algorithm (GA) is the most prototypical evolutionary algorithm, which is very widely used in the solution of simpler heuristic optimization problems, and which has many standardized toolbox type implementations. It uses a stochastic search algorithm imitating in a simplified way the natural selection process of living organisms and copying natural genetics. The original Bacterial Evolutionary Algorithm was created by enhancing and modifying some ideas within the GA [29], this way essentially speeding up the convergence. Its continuous memetic extension and its discrete versions were proposed by our group and were applied for a variety of optimization problems rather successfully. The original GA has five steps: Initial population creation, Candidate selection, Crossover, Mutation, and Fitness function evaluation [30-33].

1) Initial population creation

The size of the population varies depending on the problem, but it usually encompasses several hundred or thousands of potential solutions. The starting population is frequently created at random, providing for a wide variety of possible solutions (however, lacking the opportunity of a deterministic improvement approach already applied in this step).

2) Candidate selection

The best offspring solutions must be chosen to be parents in the new parental population in order to facilitate convergence towards optimum solutions. Because of this, an excess of offspring solutions is developed, and the best are chosen in order to make progress toward the optimum. This selection method is based on the fitness values in the population.

3) Crossover

Crossover is a function that permits the genetic material of

two or more solutions to be combined. The reason for such an operator is that both strings might represent successful components of solutions that, when combined, outperform their parents. This operator may easily be expanded to more points, where the solutions are alternately separated and rebuilt. This is not unlike the Gene transfer step in the DBMEA.

4) *Mutation*

Mutation is the second main character in Genetic Algorithms. Mutation operators change a solution by disturbing them. Random alterations are the foundation of mutation. Mutation is the part of the GA which is related to the “exploration” of the search space. It has been discovered that mutation is required for GA convergence. There are different operators for mutation such as: bit flip mutation, random resetting, swap mutation, scramble mutation, and inversion mutation. Again, here, the Mutation step of the DBMEA has its “ancestor”.

5) *Fitness function evaluation*

In this step, the phenotype of a solution is evaluated. The fitness function measures the quality of the solutions that is generated by the GA. The proper design of the fitness function is part of the overall modeling process of the overall optimization approach. The practitioner may have an influence Genetic Algorithms by designing choices of the fitness function and thus guiding the search.

V. THE PROPOSED TOUR CONSTRUCTION HEURISTIC

We proposed a novel approach that quasi-optimizes, but definitely improves, the initial population for the TSP, from the point of view of applying the DBMEA on this quasi-optimized initial population; by introducing the novel idea of applying the concept of “neighborhood circle” (NC). The NC has a pre-specified radius which will limit and speed up the search for the best possible initial population candidate. The new heuristic method is called Circle Group Heuristic (CGH). CGH is built into the Discrete Bacterial Memetic Evolutionary Algorithm (DBMEA), as its first step, this way increasing the efficiency of this memetic meta-heuristic algorithm that has already proven rather efficient in handling the optimization of TSP type tasks. Next, the CGH method will be explained in detail.

1) *The CGH construction step*

Starting at the initial node, City 0, which represents the center of the first circle, a circle is drawn with a given radius R. In the first step, the closest unvisited city within the circle will be marked for visit. The tour continues at the next unvisited city within the circle, until all nodes within the circle have been included in the tour. In the subsequent step, the node/city outside the circle, which is the closest to the last visited node/city, is marked as the next city in the tour. This new city on the outside of the circle will become the center of a new circle, and the algorithm starts again as in the case of

the first circle. So, on it goes until all the cities have been visited exactly once. A simple tour created by the CGH is shown in Figure 5.

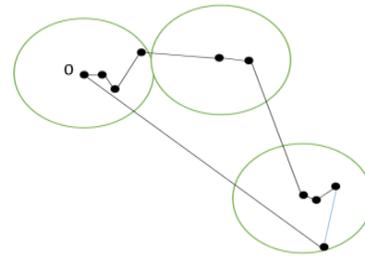


Fig. 5. Example for a simple CGH tour

A. *Brute force optimization of the radius*

We tested the new method on benchmarks of national TSP instances up to 10639 nodes [34]. In every case, an exhaustive search for the optimal radius was done in discrete steps by testing range of integer numbers [1-100] with brute force method. With these series of tests we have established that the most effective radius of the circle in the CGH generating the best initial population in the sense that the subsequent steps of the DBMEA result in the best approximation in the shortest runtime, is different from graph to graph. Examining the road network graphs of number of countries in this benchmark, with the respective city and road networks assigned with costs obtained from the road section lengths of the respective benchmark data, we found that these graphs differ from each other essentially in the behavior, because of different topologies and sizes of these road networks.

B. *GA optimization of the radius*

Later, we attempted to find the optimal radius for the CGH by applying genetic algorithm, thus allowing a continuous range for the radius. The GA is a standard toolbox, which is used one independent variables for the fitness function to return the optimal radius, and the test the range of rational numbers [1 – 100]. All the other parameters are defaults of the toolbox [35].

In the next Section, the results obtained by GA optimization of the radius method is presented and evaluated.

VI. RESULTS

The novel CGH tour construction was tested on more than 25 national TSP instance benchmarks up to 10639 nodes. In all cases, the initial population individuals obtained by the above mentioned three heuristics in the previous investigations, NN, SNN, and ANN, were compared with the ones got by the new CGH approach, the radius of CGH is calculated in both methods: simple exhaustive search and GA. In the case of all thus optimized initial individuals, the DBMEA method was applied for solving the respective TSP optimization task. Table 1 shows the tour costs of the optimal tours in the case of the initial populations generated by the previous three older approaches and both sub-versions of the CGH.

Effect of the initial population construction on the DBMEA algorithm searching for the optimal solution of the traveling salesman problem

TABLE I
THE LENGTH OF THE DETERMINISTIC INITIAL INDIVIDUAL.

Country	Number of cities	NN	SNN	ANN	Exhaustive search		Genetic Algorithm	
					R	CGH	R	CGH
Djibouti	38	9748.946	13509.088	10474.948	78-100	8306.575	79.834	8306.575
China	70	2570.329	4148.675	3565.305	18	2267.889	21.137	2263.575
Burma	80	5477.026	8674.245	6079.910	98-100	4526.302	98.272	4526.302
Qatar	194	11892.888	18980.443	17199.801	9	11649.869	56.280	11255.296
Uruguay	734	102594.358	165796.643	130793.380	57-59	95536.209	54.832	95461.998
Zimbabwe	929	117733.696	200063.995	160430.512	10	114813.039	48.955	114484.256
Luxembourg	980	14212.721	26240.107	20397.885	8	13995.032	16.254	13958.023
Rwanda	1621	32276.665	68487.437	45630.149	10	31596.476	9.768	31596.476
Oman	1979	120542.129	204249.064	152503.643	13	110747.729	12.688	110029.190
Nicaragua	3496	122412.147	229749.240	179481.992	16	118141.497	15.935	118141.497
Canada	4663	1668707.230	2852242.400	2320011.780	71-74	1603709.500	73.834	1603709.500
Tanzania	6117	501427.829	852834.843	696141.743	16	499513.302	16.567	499513.302
Egypt	7146	222335.231	391416.939	306664.393	4	217487.431	3.981	217487.431
Yemen	7663	298953.459	523144.599	417814.532	5	298150.565	5.185	297972.031
Panama	8079	146660.520	277856.032	210793.510	16	142277.699	15.984	141949.651
Ireland	8246	259165.057	421610.120	350446.111	12	255167.585	11.971	255167.585
Argentina	9152	1034964.600	1951034.190	1527346.320	16	1034084.030	15.635	1034084.030
Japan	9847	625031.710	1104954.110	909941.924	2	624849.337	2.215	624849.337
Greece	9882	391415.926	637638.279	523514.157	14	384948.298	13.158	384900.881
Kazakhstan	9976	1346903.560	2320848.050	1863893.670	13	1325094.710	13.058	1325094.710
Finland	10639	657774.773	1081658.770	889702.220	20	649477.672	20.158	649477.672

In the initial population level, the CGH produced on average 4%, 44%, and 30% shorter tours than NN heuristic, SNN heuristic, and ANN heuristic respectively. While the best tour for CGH was 17% shorter than NN heuristic in Burma (80 cities), 54% shorter than SNN heuristic in Rwanda (1621 cities), and 37% shorter than ANN heuristic in China (70 cities).

We made comparisons for the deterministic initial population individuals with the known absolute optimum after each stage of the DBMEA on the China (70 cities) and Oman (1979 cities) instances. The optimal tour for China70 is 2023, while the one for Oman1979 is 86891. The CGH generated for the initial population roughly 20% longer tours than the optimum solutions, making it a useful starting point for a memetic evolutionary algorithm.

The goal of this investigation was to show that introducing a few promising deterministic individuals would improve the efficiency of the evolutionary/memetic algorithm. The convergence speed would be unambiguously faster when using better deterministic initial individuals.

Table 2 illustrates the tour construction heuristics run times for China70 and Oman1979. It shows that the CGH not only provides better tours, but also does it faster than the NN, SNN, and ANN heuristics previously utilized. Tour costs of initial individuals following each stage of DBMEA are illustrated in Table 3.

TABLE II
THE TOUR CONSTRUCTION HEURISTICS RUN TIMES

Instance	China(70)	Oman(1979)
NN	0.002 sec	1.38 sec
SNN	0.002 sec	2.32 sec
ANN	0.002 sec	1.563 sec
CGH	0.001 sec	0.839 sec

TABLE III
TOUR COSTS OF INITIAL INDIVIDUALS FOLLOWING EACH STAGE OF DBMEA

DBMEA stage	Instance	NN	SNN	ANN	CGH	Time (Sec)
Mutation stage	China	2471.1 391	4021.1 8307	3052. 31702	2221.0 5428	0.008
	Oman	120207 .162	195634 .363	14987 1.306	108658 .897	3.873
Local search	China	2038.3 6554	2081.7 0514	2059. 382	2024.7 4914	0.106
	Oman	91178. 9462	93608. 4377	91659 .2708	90324. 38	5018. 752
Improved version of local search	China	2118.8 5481	2126.6 373	2125. 59152	2085.9 1453	0.357
	Oman	93182. 4458	102739 .247	10051 2.507	94981. 3311	197.5 64

Figures 6, 7, and 8 show the graphs of the deterministic individuals after the initial populations stage, bacterial mutation stage, and local search stage respectively for China70.

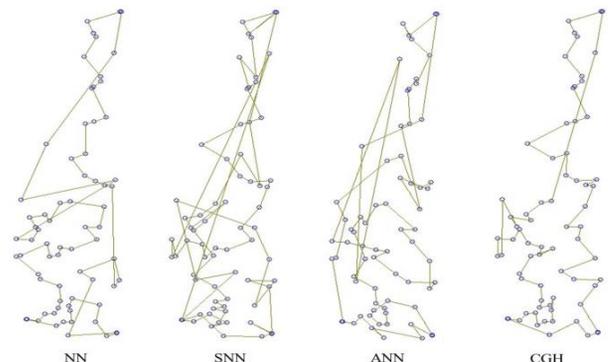


Fig. 6. Initial individual tours of China70

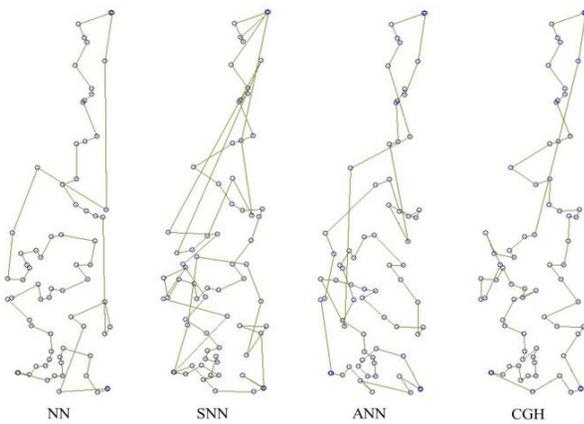


Fig. 7. Tours of the predefined individuals for China70 after bacterial mutation stage

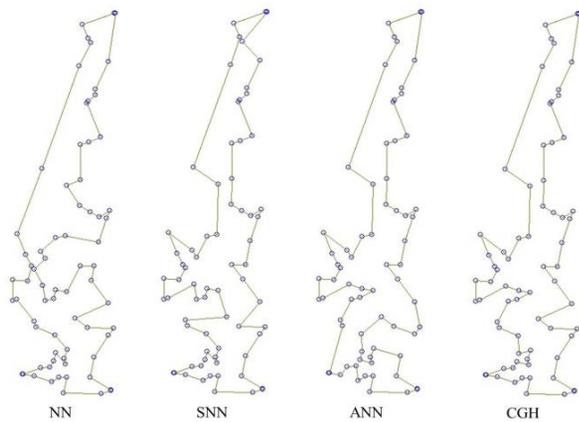


Fig. 8. Tours of the deterministic individuals for China70 after local search stage

VII. CONCLUSION

In this paper, we proposed a new and efficient initial tour construction method for DBMEA to solve TSP. We had studied and compared the other three known deterministic tour construction heuristics of DBMEA with the here proposed CGH algorithm, and we found that CGH gives better results in all DBMEA stages. During the investigations of the population’s behaviors in bacterial mutation and the local search stages, we conclude that in almost all cases, better initial population individuals will lead to faster convergence speed and better approximation of the optimal tour length. Based on our experiments, we suggest the use of our novel proposed CGH in the initial population creation stage of the DBMEA and very likely, other heuristic optimization algorithms for solving the TSP.

REFERENCES

[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, “The Traveling Salesman Problem: a computational study,” Princeton University Press, Princeton, Princeton University Press, 2006, PP 1–8. [doi: 10.1515/9781400841103.541](https://doi.org/10.1515/9781400841103.541).

[2] Y. Bartal, L.-A. Gottlieb, R. Krauthgamer, “The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme,” *SIAM J. Comput.*, Vol. 45, No. 4, pp. 1563–1581, 2016. [doi: 10.1145/2213977.2214038](https://doi.org/10.1145/2213977.2214038).

[3] A. V. Aho, J. E. Hopcroft, J. D. Ullman, “The Design and Analysis of Computer Algorithms,” Addison-Wesley, 1974.

[4] A. J. Ibada, B. Tüü-Szabó, L. T. Kóczy, “A new efficient tour construction heuristic for the Traveling Salesman Problem,” In *2021 5th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI 2021)*, Association for Computing Machinery, New York, NY, USA, 2021. pp. 71–76. [doi: 10.1145/3461598.3461610](https://doi.org/10.1145/3461598.3461610).

[5] L. T. Kóczy, P. Földesi, B. Tüü-Szabó, R. Almahasneh, “Modeling of Fuzzy Rule-base Algorithm for the Time Dependent Traveling Salesman Problem,” in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6 [doi: 10.1109/fuzz-ieee.2019.8858853](https://doi.org/10.1109/fuzz-ieee.2019.8858853).

[6] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388.” Graduate School of Industrial Administration, CMU, 1976.

[7] K. Helsgaun, “An effective implementation of the Lin-Kernighan traveling salesman heuristic,” *European Journal of Operational Research* 126, pp. 106–130, 2000. [doi: 10.1016/s0377-2217\(99\)00284-2](https://doi.org/10.1016/s0377-2217(99)00284-2).

[8] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, “Genetic algorithms for the travelling salesman problem: a review of representations and operators,” *Artificial Intelligence Review*, Vol. 13, No. 2, pp. 129–170, 1999.

[9] M. Dorigo, L. M. Gambardella, “Ant colonies for the travelling salesman problem,” *Biosystems*, Vol. 43, No. 2, pp. 73–81, 1997. [doi: 10.1016/s0303-2647\(97\)01708-5](https://doi.org/10.1016/s0303-2647(97)01708-5).

[10] N. E. Nawa and T. Furuhashi, “Fuzzy system parameters discovery by bacterial evolutionary algorithm,” in *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 5, pp. 608–616, 1999. [doi: 10.1109/91.797983](https://doi.org/10.1109/91.797983).

[11] K. Wang, L. Huang, C. Zhou, W. Pang, “Particle swarm optimization for traveling salesman problem,” In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, 2003. [doi: 10.1109/ICMLC.2003.1259748](https://doi.org/10.1109/ICMLC.2003.1259748).

[12] L. Li, Y. Cheng, L. Tan, B. Niu, “A Discrete Artificial Bee Colony Algorithm for TSP Problem,” In *Proceedings of International Conference on Intelligent Computing*, 2011. [doi: 10.1007/978-3-642-24553-4_75](https://doi.org/10.1007/978-3-642-24553-4_75).

[13] P. Moscato, “On evolution, search, optimization, genetic algorithms and martial arts towards memetic algorithms” Technical Report Caltech Concurrent Computation Program, Report. 826. Pasadena: California Institute of Technology, 1989.

[14] L. T. Kóczy, P. Földesi, B. Tüü-Szabó, “An effective discrete bacterial memetic evolutionary algorithm for the traveling salesman problem,” *International Journal of Intelligent Systems*, Vol. 32, No. 8, pp. 862–876, 2017. [doi: 10.1002/int.21893](https://doi.org/10.1002/int.21893).

[15] H. Razip and M. N. Zakaria, “Combining approximation algorithm with genetic algorithm at the initial population for NP-complete problem,” 2017 IEEE 15th Student Conference on Research and Development (SCORED), pp. 98–103, 2017. [doi: 10.1109/scored.2017.8305413](https://doi.org/10.1109/scored.2017.8305413).

[16] G. Zhang, L. Gao and Y. Shi, “An effective genetic algorithm for the flexible job-shop scheduling problem,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3563–3573, 2011. [doi: 10.1016/j.eswa.2010.08.145](https://doi.org/10.1016/j.eswa.2010.08.145).

[17] R. R. Hill, “A monte-carlo study of genetic algorithm initial population generation methods,” *Proceedings of the 31st Conference on Winter Simulation: Simulation-a Bridge to the Future*, vol. 1, pp. 543–547, 1999. [doi: 10.1109/wsc.1999.823131](https://doi.org/10.1109/wsc.1999.823131).

[18] J. Wang, O. K. Ersoy, X. Chen, F. Wang, “A Method of Initial Population Generation of Intelligent Optimization Algorithms for Constrained Global Optimization,” *International Journal of Hybrid Information Technology*, al of Hybrid Information Technology Vol. 10, No. 6, 2017, pp. 47–56. [doi: 10.14257/ijhit.2017.10.6.05](https://doi.org/10.14257/ijhit.2017.10.6.05).

[19] H. Maaranen, K. Miettinen, M.M. Mäkelä, “Quasi-random initial population for genetic algorithms,” *Computers & Mathematics with Applications*, Vol. 47, No. 12, pp. 1885–1895, 2004. [doi: 10.1016/j.camwa.2003.07.011](https://doi.org/10.1016/j.camwa.2003.07.011).

[20] B. Kazimipour, X. Li, A. K. Qin, “A review of population initialization techniques for evolutionary algorithms,” 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 2585–2592, 2014. [doi: 10.1109/cec.2014.6900618](https://doi.org/10.1109/cec.2014.6900618).

Enhanced Security of Software-defined Network and Network Slice Through Hybrid Quantum Key Distribution Protocol

[21] S. Helwig, R. Wanka, "Theoretical analysis of initial particle swarm behavior," in *Parallel Problem Solving from Nature-PPSN X*. Springer, pp. 889–898, 2008. doi: 10.1007/978-3-540-87700-4_88.

[22] B. Kazimipour, X. Li, A. Qin, "Initialization methods for large scale global optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, pp. 2750–2757, 2013. doi: 10.1109/cec.2013.6557902.

[23] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "A novel population initialization method for accelerating evolutionary algorithms," *Computers & Mathematics with Applications*, vol. 53, no. 10, pp. 1605–1614, 2007. doi: 10.1016/j.camwa.2006.07.013.

[24] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, D. B. Shmoys, "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization" John Wiley & Sons, New York, 1985.

[25] L. T. Kóczy, P. Földesi and B. Tüü-Szabó, "Adiscrete bacterial memetic evolutionary algorithm for the traveling salesman problem," 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 3261–3267, 2016. doi: 10.1109/cec.2016.7744202.

[26] H. H. Hoos, T. Stutzle, "Stochastic Local Search: Foundations and Applications," Morgan Kaufmann, San Francisco, 2005.

[27] B. Tüü-Szabó, P. Földesi, L. T. Kóczy, "An Efficient Evolutionary Metaheuristic for the Traveling Repairman (Minimum Latency) Problem," *International Journal of Computational Intelligence Systems*, Vol. 13, No. 1, Pages 781–793, 2020. doi: 10.2991/ijcis.d.200529.001.

[28] L. T. Kóczy, P. Földesi, B. Tüü-Szabó, "Enhanced discrete bacterial memetic evolutionary algorithm – An efficacious metaheuristic for the traveling salesman optimization," *Information Sciences*, Vol. 460–461, pp. 389–400, 2018. doi: 10.1016/j.ins.2017.09.069.

[29] N. E. Nawa, T. Furuhashi, "A study on the effect of transfer of genes for the bacterial evolutionary algorithm," 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111), pp. 585–590, 1998. doi: 10.1109/kes.1998.726026.

[30] O. Kramer, "Genetic Algorithm Essentials," Springer International Publishing, 2017. doi: 10.1007/978-3-319-52156-5_2.

[31] M. Gen, R. Cheng, L. Lin, "Network Models and Optimization Multiobjective Genetic Algorithm Approach," *British Library Cataloguing*, Springer, 2008.

[32] L. Nagy, "Classical and quantum genetic optimization applied to coverage optimization for indoor access point networks," *Infocommunications Journal*, Vol. 4, No. 4, 2012.

[33] K. Kubíček, J. Novotný, P. Švenda, and M. Ukrop, "New results on reduced-round tiny encryption algorithm using genetic programming," *Infocommunications Journal*, vol. 8, no. 1, pp. 2–9, 2016.

[34] VLSI TSP dataset, April 2015. <http://www.math.uwaterloo.ca/tsp/world/countries.html>.

[35] *Genetic Algorithm and Direct Search Toolbox*, User's guide version 1, The MathWorks Inc. 2004.



Ali Jawad Ibada Received his M.Sc. degree in computer engineering from the Middle Technical University, Baghdad, Iraq which was achieved with first-class honors in 2015. Currently, he is a Ph.D. candidate in computer engineering in the Department of Telecommunications and Media Informatics at the Budapest University of Technology and Economics, Budapest, Hungary.

His research interests include evolutionary and memetic algorithms, neural networks, computational intelligence and machine learning algorithms in modeling complex systems.



Boldizsár Tüü-Szabó received the PhD degree in information sciences from the Szechenyi Istvan University, Győr, Hungary in 2021 and the M.Sc. degree in electrical engineering in 2015. Currently he is senior lecturer at the Department of Information Technology of Szechenyi Istvan University. His research interests are cover optimization, evolutionary computation and fuzzy modelling.



Laszlo T. Koczy received the M.Sc., M.Phil. and Ph.D. degrees from the Technical University of Budapest (BME) in 1975, 1976, and 1977, respectively; and the D.Sc. degree from the Hungarian Academy of Science in 1998. He spent his career at BME until 2001 and from 2002 at Szechenyi Istvan University (Győr, SZE). He has been a visiting professor in Australia, Japan, Korea, Austria, Italy, etc. His research interests are fuzzy systems, evolutionary and memetic algorithms, and neural networks, as well as applications in infocommunications, logistics, management, and others. In the last years, he has focused on NP-complete problems, especially route selection and optimization and the application of metaheuristics for approximate solution of such complex tasks. He has published over 775 articles, most of those being refereed papers, and several text books on the subject. His Hirsch-index is 40 by Google Scholar (based on 7300 citations there).