IMPROVING INITIAL POPULATION FOR GENETIC ALGORITHM USING THE MULTI LINEAR REGRESSION BASED TECHNIQUE (MLRBT)

Esra'a Alkafaween[1,*], Ahmad B. A. Hassanat[1,2,3], Sakher Tarawneh[1]

[1]IT Department, Mutah University, Karak, Jordan
[2]Computer Science Department, Community College, University of Tabuk, Tabuk, Saudi Arabia
[3]Industrial Innovation and Robotics Center, University of Tabuk, Tabuk, Saudi Arabia

*E-mail of corresponding author: esrakafaween86@gmail.com

Resume

Genetic algorithms (GAs) are powerful heuristic search techniques that are used successfully to solve problems for many different applications. Seeding the initial population is considered as the first step of the GAs.

In this work, a new method is proposed, for the initial population seeding called the Multi Linear Regression Based Technique (MLRBT). That method divides a given large scale TSP problem into smaller sub-problems and the technique works frequently until the sub-problem size is very small, four cities or less. Experiments were carried out using the well-known Travelling Salesman Problem (TSP) instances and they showed promising results in improving the GAs' performance to solve the TSP.

Article info

Received 9 March 2020
Accepted 11 May 2020
Online 20 October 2020

Keywords:

genetic algorithm,
population seeding,
TSP,
multi linear regression

Available online: https://doi.org/10.26552/com.C.2021.1.E1-E10

ISSN 1335-4205 (print version)
ISSN 2585-7878 (online version)

1 Introduction

Genetic Algorithm (GA) is one of effective and robust machine learning algorithms [1]. Many studies were concerned with Genetic Algorithms (GA) and exploited its capabilities in designing smart systems and solving problems [2-4]. The genetic algorithms are concerned, in general, with how to produce new chromosomes (individuals) that possess certain features through recombination (crossover) and mutation operators [5-6]. Therefore, individuals with appropriate characteristics have the strongest chance of survival [7]. Typically, the GAs start with a number of random solutions (initial population); this is the first phase in the GA. This phase generates a set of possible solutions randomly or by heuristic initialization. Although the initial population seeding phase is executed only once, it has an important role to improve the GA performance.

The GA aims to produce many solutions to solve specific problems, such as a problem of the TSP, which is a common issue in Artificial Intelligence area [8-9]. Several previous studies dealt with insight in the GA procedure to indicate its procedure and how it can be exploited to solve sophisticated problems, such as finance, medical, mathematical and technical ones [10-11].

Efficiency of the GA is based on many factors, such as initial population, crossover operator, mutation operator and selection strategy [12]. In this paper, a new enhanced initial population is proposed to increase the GA performance.

The proposed technique distinguishes from other previous techniques, where it divides the problem into sub problems using the regression line, which indicates the relationship between the points in the xy coordinates. Each sub problem results from intersection between the regression line and the rotated line at the center point. Then, the initial population is determined by reconnecting each sub problem. Results of experiments show that the proposed technique is of high efficiency through different aspects: improvement in error rate, average convergence and convergence diversity. This work contribution is in proposing a new enhanced approach to generate the initial population to be used in solving the decision-making problems, especially the TPS problem. Each individual in population is called chromosome, which will be presented as a solution [13]. Then, sampling this initial population will generate an intermediate population. Therefore, it is possible to apply reproduction, crossover and mutation on the new intermediate population [14]. This process is repeated until reaching the desired number of generations or a convergence adopted in the design is reached. Hence, initializing population is necessary in order to start the process of evolution in the GA [15].

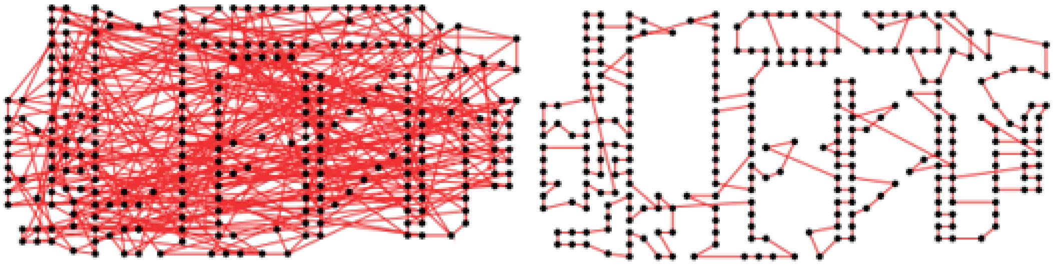


Figure 1 *Random Initialization and K-mean initialization*

2 Related work

Various initialization techniques have been introduced since the GA concepts appeared, such as random technique, nearest neighbor technique, Gene Bank (GB) technique, K-means clustering technique and Initialization Mechanism Based on Regression techniques. The randomization technique is considered as one of the most suitable and most used technique for generating the seed of the initial population. However, it may contain a poor fitness solution that reduces the possibility of finding the optimal solution.

Here is briefly presented a background review of several of the initial population seeding techniques that are used for the GAs.

A. Random initialization technique

This technique is widely used in machine learning algorithms and it is used widely in GA because it is the simplest way to initialize population seeding. In addition, the researchers prefer this technique especially when the prior data about expected optimal solution is trivial. As shown on the left-hand side of Figure 1, the successive cities of the initial solutions are chosen randomly, where the right-hand side shows initializing using the K-means [16]. This figure shows the difference between the random initialization and any other initialization techniques such as K-means. Most researchers use the sentence "generate an initial population" to indicate that they use random initialization technique. In the case of the TSP, random initialization selects the cities randomly and it generates random numbers from 1 up to (n) for each city. As shown in study [17], "if the current individual already contains the generated number, then it generates a new number. Otherwise, the generated number is added to the current individuals". Hence, this process is repeated until reaching the desired individual size (n).

B. Gene bank initialization technique

The Gene Bank is considered as a database of initial population to be used as solutions based on quality and diversity [18]. In the case of the TSP, each city that the salesman travels to is called N and permuted and assembled to build a gene bank. Then, encoding the nearest cities C to the city in order to be encoded to build a gene bank and noting that C should be less than (N-1). As a result, the Gene bank is given as a matrix A where its size is (C × N). For example, A[i][j] is an element in the matrix (Gene Bank) is the j-th city to city i. Therefore, the first row in the Gene Bank includes the C closest cities for city i. For each solution, the initial city i is initialized randomly from the row i in the Gene Bank. "After that, the method selects city j, where j is the nearby one in the unvisited elements of the i-th row. Then, city k is selected from the j-th row of gene bank as the next city. If all the city codes of the j-th row have been selected, then next city is chosen randomly from the set of unvisited cities" as mentioned by [18]. Thus, this process is repeated until generating the solution of the size N.

C. Nearest neighbor initialization technique

The nearest neighbor (NN) is known as the most common initial population seeding technique. In addition, NN can be used as an efficient random initial population method to be used for the purpose of generating initial population solutions, especially in the case of the TSP (to be solved with GA). The process of generating each individual begins by randomly selecting a city to be the starting city; then, adding the nearest city to the starting city as the new starting city. Thus, the nearest city that was not added to the current city is added to the individual until all the cities are included in the individual. Therefore, as the next generations that were created from a city nearest to the current city, the generated individuals would enhance the evolving search process [19].

D. K-means initial population (KIP)

To improve the process of initializing population of the GA, several studies used K-means clustering algorithm, especially in the case of the TSP problem, such as [20-22]. These studies used the k-means clustering to split a large-scale of the TSP into small groups k, where $K = [\sqrt{N} + 0.5]$ and N denotes number of cities. Then, the KIP is applied to the GA to find the local optimal path for each group and a global optimal path that connects each local optimal solution.

E. Initialization mechanism based on the regression techniques

A new initialization technique has been designed to improve the GA for solving the well-known TSP. It is based on the the Regression line and the perpendicular line that crosses the regression line at the center point to divide a large-scale TSP problem to small sub-problems. The resulting sub-problems are repeatedly classified to fit

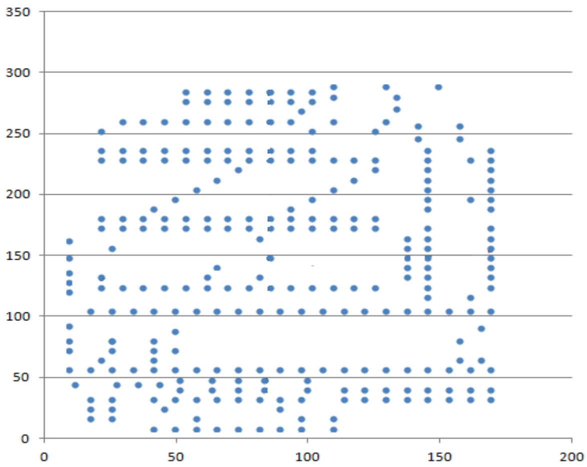


***Figure 2** The (x, y) scatter for the TSP city (a280)*

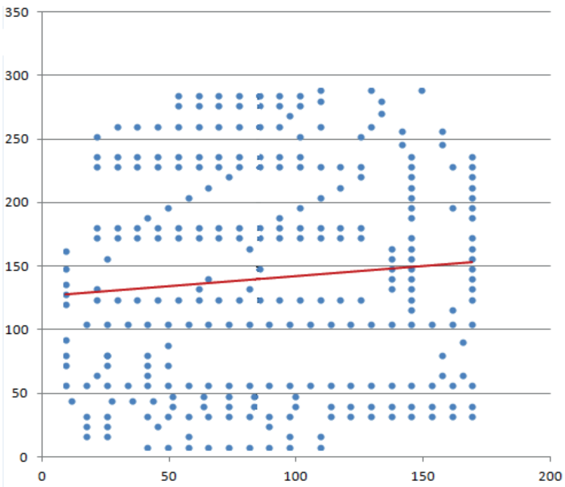


***Figure 3** The regression line*

into the four categories to obtain local optimal solutions [23].

Procedures of the proposed method **firstly** start with dividing the large-scale TSP into the four small sub-problems using Regression line and the perpendicular line and then classify the points into four categories. Each category is divided into four new categories recursively, by using the Regression line and the perpendicular line. The process carries on until having the target category that contains a small number of instances (x, y points). Maximum four cities or x, y points are assigned to each category that are considered as initial population for the TSP sub-problem. The process ends up when the local optimal solution is obtained for each category.

Second, rebuild the initial populations seeding by reconnecting all the local optimal solutions together. Finally, mutate the initial population N times to obtain N solutions, where N is the population size.

The research, presented in this paper, is compared to that research, because it already outperformed the two methods of seeding the population: random and the nearest neighbor.

3 The proposed method

This paper aims to propose a new initialization technique, which has been designed to enhance the GA at the aim of solving the problem of the TSP. The proposed technique is called (Multi Linear Regression Biased Techniques (MLRBT). This technique aims to divide the TSP problem into small sub-problems. In addition, it works depending on the Regression line on X and Regression line on Y, where it crosses the regression line at a point, which is not necessary to be in the center. Then, the previous process is repeated on the resulted sub-problems and classified in order to fit into the four categories to obtain local optimal solutions.

The main procedures of the proposed method include two main steps:

- Initially, the large-scale TSP problem is divided into four small sub-problems by using regression line on X and the regression line on Y. Then, the method classifies the resulting points into four categories. Then, recursively, dividing each category into new four sub-categories through the Regression line on X and Regression line on Y. The previous process will continue until reaching the optimal target category, which includes a small number of instances (x, y) points. Maximum is four cities or (x, y) points assigned to each category that are considered as initial population for TSP sub-problem. The process ends up when the local optimal solution is obtained for each category.
- Secondly, reconnecting all the local optimal solutions together in order to rebuild the initial populations seeding. Finally, to obtain N solutions, the method mutates the initial population N times, where N denotes the population size.

To show how regression population seeding works, Figure 2 shows the main steps by using one of the TSP cities (a280).

The following steps illustrate design of the MLRBT:

Step 1: dividing the points into two categories by using the regression line equation ($y=a + bx$). As shown in Figure 3, each section includes almost an equal number of nodes.

Step 2: this step aims to divide the points into two sections using the regression line equation ($x = a + by$), then, the diagram is divided into four equal sections where each section includes an equal number of cities as shown in Figure 4.

Step 3: In this step, the diagram is divided into four sections as shown in Figure 5:

- <u>Section 1:</u> (A is all the points above the positive x-axis and to the right of the positive y- axis).
- <u>Section 2:</u> (B is all the points above the negative x- axis and to the left of the positive y- axis).
- <u>Section 3:</u> (C is all the points below the negative x- axis and to the left of the negative y- axis).
- <u>Section 4</u>**:** (D is all the points below the positive x-axis and to the right of the negative y- axis).

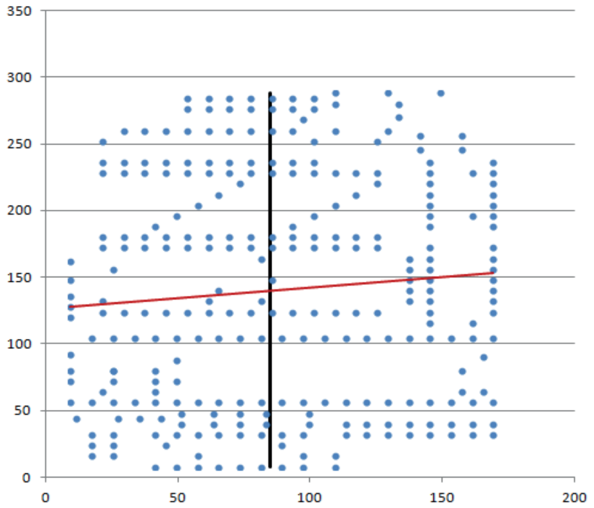


***Figure 4** Regression on x line, which intersects with the regression y line*

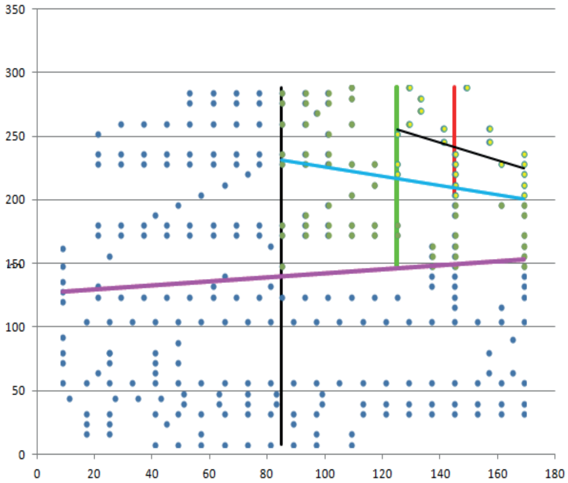


***Figure 6** Implementation of the regression line algorithm on category A (+, +)*

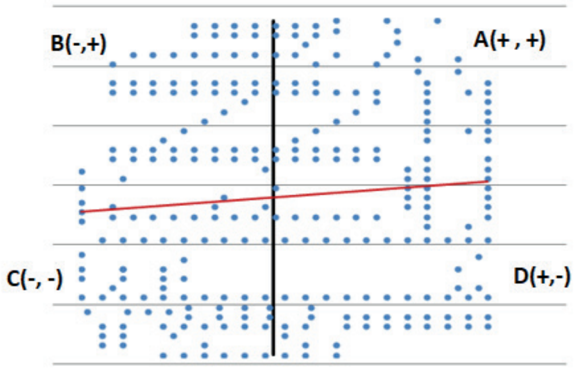
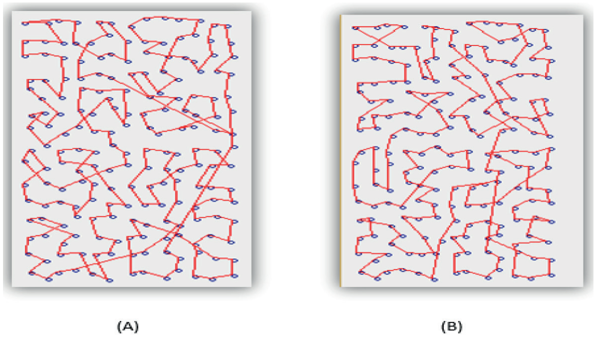


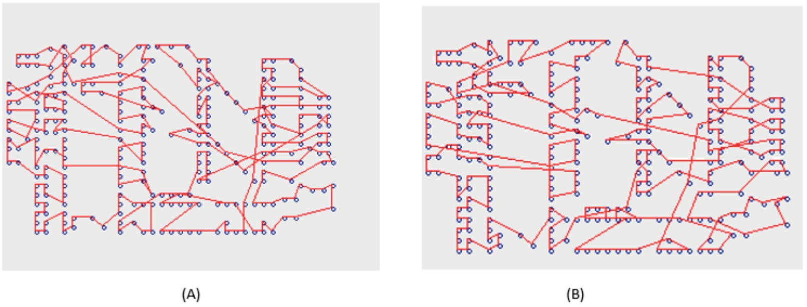
***Figure 5** A, B, C and D categories that generated from the intersect between regression line on x and y*



(A)

(B)

***Figure 7** (A) The initial solution for a280 using [23], (B) The initial solution for a280 using MLRBT*



(A)

(B)

***Figure 8** (A) The initial solution for rat195 using [23], (B) the initial solution for rat195 using MLRBT*

Step 4: repeating the method recursively four times on all categories: (A), (B), (C) and (D), as shown in Figure 6.

Step 5: Terminate the recursive call, if the number of points (cities) is less than or equal four. Since the algorithm uses the regression line, it guarantees that the small number of cities that it ends with are more likely to be neighbors and closer to each other from the other cities. Therefore, connecting them with each other is better than connecting any of them with further cities, as these local links are minimized and minimizing the local links attributes in finding a smaller global route.

Step 6: Select a random city to be the starting city and then add the nearest city as a new starting city until having all the cities connected in the category of the local path. The group in each category is connected with the nearest group in other categories until all the groups are connected in a global path.

This method gives only one solution, and to derive solutions with a specific population size, the mutation process is used. It is used to mutate the seed solution (n-1) times, in order to derive the other solutions, where n is e size of the population.

***Table 1** GA configuration parameters for experiments*

no.	parameter	value/technique
1	population size	100
2	generation limit	3000
3	initialization technique	regression based technique [23] for GA's population initialization techniques and multi linear regression based technique (MLRBT)
4	crossover method	one-point modified crossover
5	crossover probability	0.82
6	mutation method	exchange mutation
7	mutation probability	1.0
8	selection	roulette wheel
9	termination condition	generation limit

***Table 2** Experimental results for the MLRBT and RBT*

city	optimal solution	MLRBT			regression based technique		
		best solution before GA	best sol	average	best solution before GA	best sol	average
KroA100	21282	32210	27493	29440.8	40675	29234	30506.2
eil51	426	550	465	478.6	680	475	500.6
pr76	108159	156926	128039	133572.8	209897	132694	139536.6
KroA200	29368	51191	46269	47877.4	64374	51427	52770.8
lin318	42029	73676	68490	70237.6	81754	75180	76764.7
pr144	58537	152275	124763	131974.3	213332	124881	136572.1
att532	27686	160340	145128	157423.6	184976	178600	182420.33
rat783	8806	15711	14659	15308.1	16730	16214	16355.8
d2103	80450	172550	169253	171890	175701	175147	175573.5
fnl4461	182566	361538	351280	358443	365414	363900	364567.5

Figures 7 and 8 show a comparison of the two initialization population methods: Initialization Mechanism Based on regression Techniques [23] and MLRBT, applied to two TSP instances: a280 and rat195.

As shown in Figures 7 and 8, the process of the GA is continuing to enhance and optimize the solutions.

4 Results and analysis

To evaluate the proposed methods, he experiments on different TSP problems were conducted. Experiments include conducting the proposed method, which includes implementing the Multi Linear Regression Based Technique MLRBT together with the Regression based technique in [23], which was found superior to both the Random and the NN techniques. Table 1 shows the selected GA parameters.

Experiments include applying each technique 10 times for each of the TSP instances. Then, computing the average of all the executions results for the purpose of experimental analysis. To conduct the experiments, Microsoft visual studio 2008 tool was used, as well as the TSP benchmark datasets obtained from TSPLIB [24]. Results of experiments are shown in Table 2. Hence, the results are divided into three different solutions: best solution, worst solution and the average solution as well.

Ten TSP instances were selected to implement the experiment; these instances include KroA100, eil51, pr76, KroA200, in318, pr144, att532, rat783, d2103, fnl4461 and e experiments were repeated ten times for each instance. It resulted in that the MLRB technique was more beneficial than the Regression Based Technique [21] across all the instance categories (small, medium and large) using the same parameters, which have been fixed according to the previous research, i.e. the Regression Based Technique [21]. The RBT was previously compared to other techniques, the random and the NN in different researches and was found superior. In addition, Table 2 shows that the MLRB is found to have advantage for all the cities in terms of the best solution. It can also result in that these techniques are more beneficial than the random and the NN techniques, where the successful performance has been achieved by the proposed solution and the achieved performance was close to the optimal solution.

It is necessary to consider the performance factors (which have been identified as measurements) when investigating several initialization techniques such as error rate, average convergence and convergence diversity.

Table 3 *Error Rate (%)*

problem	optimal solution	random		NN		regression based T		MLRBT	
		best	worst	best	worst	best	worst	best	worst
KroA100	21282	5.70759	6.15248	3.11968	6.03017	0.83437	0.93065	0.40767	0.54403
eil51	426	2.12911	2.43897	0.94366	1.11972	0.46714	0.61502	0.17606	0.32629
pr76	108159	3.41266	3.70083	0.87348	1.03310	0.88179	1.04367	0.28720	0.50809
KroA200	29368	9.11206	9.74299	5.11244	5.25940	1.19021	1.31865	0.53790	0.76202
lin318	42029	11.64106	12.19720	5.21618	5.68427	0.91615	0.95808	0.70366	0.78900
pr144	58537	10.81328	11.65166	1.26527	1.56317	0.84842	0.90172	0.79873	0.84839
att532	27686	28.14841	54.83620	27.91581	54.87900	5.75349	6.14578	4.36174	5.29621
rat783	8806	18.01306	18.65478	9.13570	9.41154	0.87247	0.90427	0.78605	0.83046
d2103	80450	38.15525	38.67478	10.16072	38.59516	1.16288	1.20272	1.13421	1.16910
fnl4461	182566	22.10100	43.99948	22.06794	43.88047	1.01834	1.03906	0.97270	0.98494

Table 4 *Average Error rate (100%)*

city	MLRBT	regression based T
KroA100	0.339273518	0.476779348
eil51	0.225454545	0.373529412
pr76	0.310764309	0.484704403
KroA200	0.426305405	0.543790972
lin318	0.429542863	0.485908946
pr144	0.615583648	0.725606098
att532	0.827329425	0.850326529
rat783	0.439500987	0.473640167
d2103	0.533758331	0.542119851
fnl4461	0.495029568	0.500385864

However, in any problem, the error rate denotes the percentage difference between the known optimal solution and the fitness value of the solution [19]. The Error Rate can be given by the following formula:

$$Error\ Rate(\%) = \frac{Fitness - Optimal\ fitness}{Optimal\ fitness} \times 100\,. \quad (1)$$

This factor measures the quality of the generated population by finding the effect of applying initial population technique on the GAs' performance to obtain a solution near to optimal one. The error rates are also classified into two types, depending on the fitness values in the given problem population. In other words, individuals with high error rate are given according to the initial population with the worst fitness value. In addition, individuals with low error rate are given according to the initial population with the best fitness. The experimental results of error rate for random, NN, RTB and MLRBT are given in Table 3.

From Table 3 follows that MLRBT technique achieved the minimum error rate likened to Random, NN and RBT techniques. It can be noted that the Multi Linear Regression based technique for GA's achieved lower error rate than the other seeding techniques, which are "Random and NN and RBT". Hence, it indicates that the produced individuals by the MLRBT for GA's are better fit the quality measures than the individuals that were produced by each of: NN, Random and RBT techniques. Since the proposed technique divides the problem (which is improvement for the RBT), this difference occurred. Table 3 also shows the experimental results of ("the initial population techniques w.r.t. error rate for the best individuals and the worst individuals in the initial population for each technique").

The results of Table 3 are extracted from Table 2, which includes the error rate for the MLRBT and RBT [23] for each city and there is a clear superiority in ratios in the table in favor of the MLRBT.

The Average error rates, obtained from both initial population techniques (RBT and MLRBT), for different problems are given in Table 4.

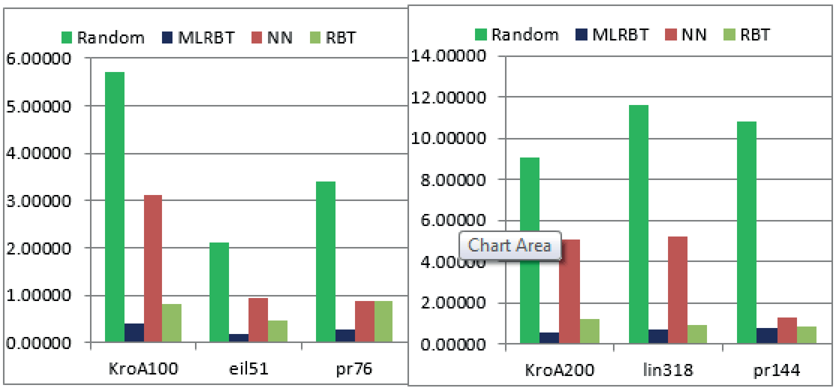
Besides on the size of cities, Table 5 shows the selected TSP examples, which were classified into three classes according to their problem size. The error rates of several initial population techniques are given for different classes of a problem example by Class A and Class B.

As shown in Figure 9, the error rates of several initial population techniques are given for different classes of a problem example by Class A and Class B.

Average Convergence (%) is the convergence rate of solutions in the initial population, and it is given by the following formula [25]:

Table 5 *Different sized TSP problem*

Class	Size	city
Class 1	Size<= 100	KroA100, pr76, eil51
Class 2	100 <Size<= 500	KroA200, pr144, lin318
Class 3	500<Size<= 1000	att532, rat783, fnl4461, d2103



Class A **Class B**

Figure 9 *Performance of initialization techniques*

Table 6 *The Average Convergence (%)*

city	MLRBT	RBT
KroA100	0.660726482	0.523220652
eil51	0.774545455	0.626470588
pr76	0.689235691	0.515295597
KroA200	0.573694595	0.456209028
lin318	0.570457137	0.514091054
pr144	0.384416352	0.274393902
att532	0.172670575	0.149673471
rat783	0.560499013	0.526359833
d2103	0.466241669	0.457880149
fnl4461	0.504970432	0.499614136

$$Average\ convergence = \\ = 1 - \frac{Average\ Fitness - Optimal\ fitness}{Optimal\ fitness} \times 100. \quad (2)$$

Optimal fitness denotes the recognized optimal value of identical instance, and Average fitness denotes the average value of the initial population fitness.

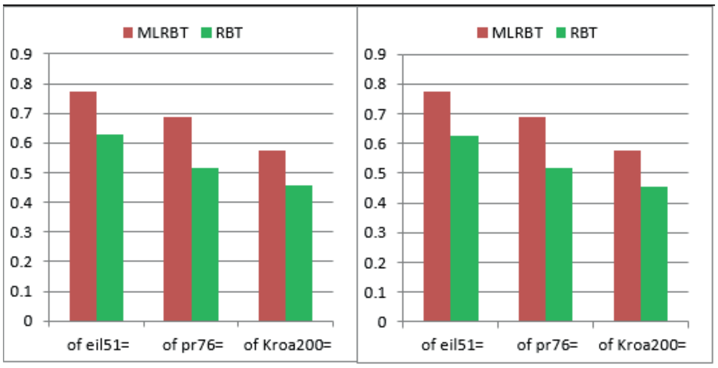
From the results, it was found that the MLRBT regression based technique for the GA's population initialization had average convergence rate higher than the RBT. In addition, it was noted that the MLRBT works in better way, especially in the case of large size problems. Hence, it can result that individuals that are produced by the MLRBT are the closest to the optimal solution. Moreover, Table 6 includes the experimental results of the initial population techniques w.r.t. average convergence (%).

As shown in Figure 10, the Multi Linear Regression Based Technique (MLRBT) achieved larger convergence than the Regression based technique. Hence, the MLRBT was found slightly greater than Random and NN, especially the RBT, [23] is better than Random and NN technique.

Furthermore, the final solution error rate denotes the difference between the known optimal solution and the final solution that resulted when applying the GAs on the TSP instances using one of initial population techniques. This can be given by the following formula:

$$Final\ Solution\ Error\ Rate\,(\%) = \\ = \frac{final\ solution - Optimal\ fitness}{Optimal\ fitness} \times 100. \quad (3)$$

In addition, this factor aims to compute the produced population quality and this can be conducted by determining



Class A **Class B**

Figure 10 *The Average Convergence (%)*

Table 7 *Final Solution Error Rate (%)*

city	MLRBT	RBT
KroA100	0.291842872	0.373649
eil51	0.091549296	0.115023
pr76	0.183803475	0.226842
KroA200	0.57549033	0.751124
lin318	0.629589093	0.788765
pr144	1.13135282	1.133369
att532	4.241927328	5.450914
rat783	0.664660459	0.841245
d2103	1.103828465	1.177091
fnl4461	0.924126069	0.993252

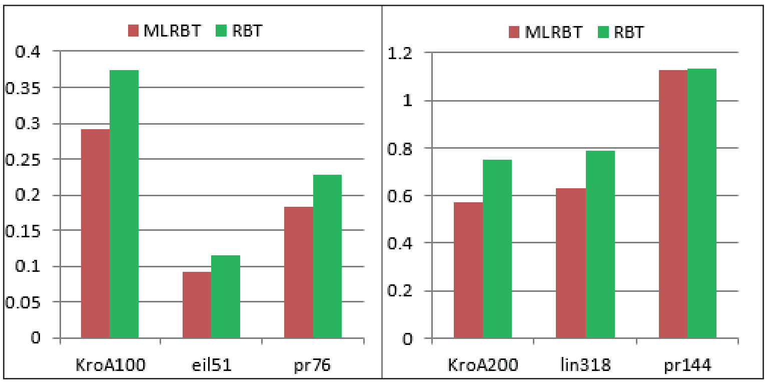


Figure 11 *Final solution error rates for MLRBT and RBT [23]*

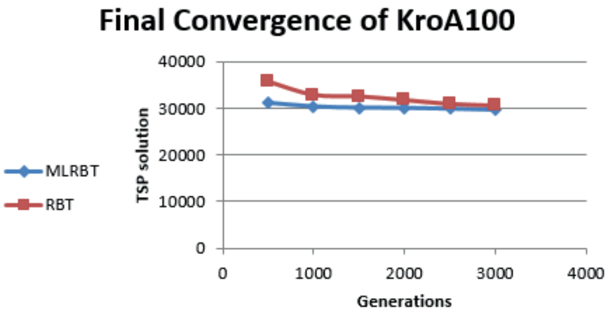


Figure 12 *KroA100 - overview of performance*

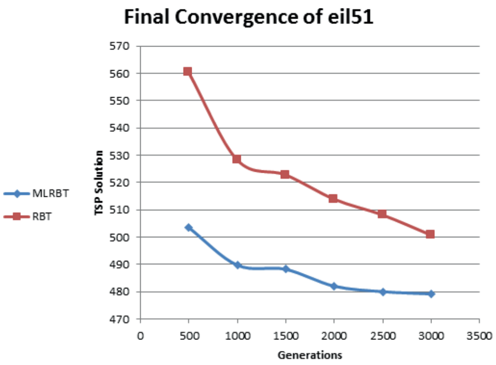


Figure 13 *eil51- overview of performance*

the impact of applying initial population technique on the GAs performance for obtaining a solution, which is close to the optimal one. However, the error rate of the final solutions indicated that the MLRBT for the GA's population initialization achieved error rate lower than the RBT. In other words, this indicates that the produced individuals from the MLRBT achieved a higher fit quality more than the individuals that are produced from RBT. This is evaluated in Table 7.

Figure 11 shows the final solution error rate obtained from several initial population techniques used for different classes of problem instances. The Multi Linear Regression Based Technique (MLRBT) achieved the maximum final solution error rate over the Regression based technique [23].

The performance MLRBT and RBT [23] techniques' was observed from the instances: kroA100 and eil51 as shown in Figures 12 and 13. In addition, the two figures show the produced initial population from the MLRBT, RBT and the final solution after 3000 generations.

5 Conclusion

This paper introduced a new, enhanced, initial population technique for genetic algorithm in order to solve the problem of the TSP and this technique was called the Multi Linear Regression Based Technique for GAs population initialization. In addition, the proposed technique has been implemented and analyzed, and to test the efficiency of the proposed technique, it has been compared to three other population techniques: Random, Nearest Neighbor (NN) initial population and Regression based techniques.

In this context, a set performance criteria were considered in order to compute the performance factors for the proposed technique and the other seeding techniques including the convergence diversity, error rate, and average convergence.

Moreover, to conduct experiments, the study extracted several TSP examples from the standard TSPLIB. After conducting experiments, the results indicated that the proposed Multi Linear Regression Based Technique for the GA's population initialization achieved the higher efficiency than the other initial population techniques to be depended in developing GA-based applications. Hence, it can be concluded that the Multi Linear Regression Based Technique for the GA's population initialization produces the high quality and efficient fittest individuals, which enable the GA to enhance the solution using the best-fit individuals.

References

[1] HENDRICKS, D., GEBBIE, T., WILCOX, D. High-speed detection of emergent market clustering via an unsupervised parallel genetic algorithm. *South African Journal of Science* [online]. 2016, **112**(1/2), p. 1-9. eISSN 1996-7489. Available from: https://doi.org/10.17159/sajs.2016/20140340

[2] GOLBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. Boston: Addion - Wesley Longman Publishing Co., 1989. ISBN 978-0-201-15767-3.

[3] MOHAMMED, A. A., NAGIB, G. Optimal routing in Ad-Hoc network using genetic algorithm. *International Journal of Advanced Networking and Applications*. 2012, **3**(5), p. 1323-1328. ISSN 0975-0290, eISSN 0975-0282.

[4] AYALA, H. V. H., DOS SANTOS COELHO, L. Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications* [online]. 2012, **39**(10), p. 8968-8974. ISSN 0957-4174. Available from: https://doi.org/10.1016/j.eswa.2012.02.027

[5] HOLLAND, J. H. Genetic algorithms. *Scientific American* [online]. 1992, **267**(1), p. 66-72. ISSN 0036-8733. Available from: http://dx.doi.org/10.1038/scientificamerican0792-66

[6] EIBEN, A. E., SMITH, J. E. *Introduction to evolutionary computing* [online]. Berlin Heidelberg: Springer Science & Business Media, 2003. ISBN 978-3-642-07285-7, eISBN 978-3-662-05094-1. Available from: https://doi.org/10.1007/978-3-662-05094-1

[7] GENDREAU, M., POTVIN, J.-Y. *Handbook of metaheuristics*. Vol. 2. Springer International Publishing, 2010. ISBN 978-1-4419-1663-1.

[8] HASSANAT, A., ALMOHAMMADI, K., ALKAFAWEEN, E., ABUNAWAS, E., HAMMOURI, A., PRASATH, V. B. Choosing mutation and crossover ratios for genetic algorithms - a review with a new dynamic approach. *Information* [online]. 2019, **10**(12), 390. ISSN 2078-2489. Available from: https://doi.org/10.3390/info10120390

[9] HASSANAT, A. B., ALKAFAWEEN, E. A., AL-NAWAISEH, N. A., ABBADI, M. A., ALKASASSBEH, M., ALHASANAT, M. B. Enhancing genetic algorithms using multi mutations: experimental results on the travelling salesman problem. *International Journal of Computer Science and Information Security*. 2016, **14**(7), p. 785-801. ISSN 1947-5500.

[10] HUSSAIN, A., MUHAMMAD, Y. S., SAJID, M. N. A simulated study of genetic algorithm with a new crossover operator using traveling salesman problem. *Journal of Mathematics*. 2019, **51**(5), p. 61-77. ISSN 1016-2526.

[11] ZHONG, J., HU, X., GU, M., ZHANG, J. Comparison of performance between different selection strategies on simple genetic algorithms. In: International Conference on Computational Intelligence for Modelling, Control and Automation

CIMCA ´05 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce CIMCA-IAWTIC´06: proceedings. Vol. 2. 2005. p. 1115-1121.

[12] ALKAFAWEEN, E., HASSANAT, A. Improving TSP solutions using GA with a new hybrid mutation based on knowledge and randomness. arXiv preprint arXiv:1801.07233. 2018.

[13] HASSANAT, A., ALKAFAWEEN, E. On enhancing genetic algorithms using new crossovers. *International Journal of Computer Applications in Technology* [online]. 2017, **55**(3), p. 202-212. ISSN 0952-8091. Available from: https://doi.org/10.1504/IJCAT.2017.10005868

[14] ALKAFAWEEN, E. O. *Novel methods for enhancing the performance of genetic algorithms*. Master´s degree thesis. Karak: Mutah University, 2018.

[15] VEDAT, T., DALOGLU, A. T. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers and Structures* [online]. 2008, **86**(11-12), p. 1204-1218. ISSN 0045-7949. Available from: https://doi.org/10.1016/j.compstruc.2007.11.006

[16] YUGAY, O., KIM, I., KIM, B., KO, F. I. Hybrid genetic algorithm for solving traveling salesman problem with sorted population. In: 3rd International Conference on Convergence and Hybrid Information Technology: proceedings [online]. IEEE, 2008. ISBN 978-0-7695-3407-7. Available from: https://doi.org/10.1109/ICCIT.2008.373

[17] WAGSTAFF, K., CARDIE, C., ROGERS, S., SCHRODL, S. Constrained K-means clustering with background knowledge. In: 8th International Conference on Machine Learning ICML´01: proceedings. 2001. p. 577-584.

[18] WEI, Y., HU, Y., GU, K. Parallel search strategies for TSPs using a greedy genetic algorithm. In: 3rd International Conference on Natural Computation ICNC 2007: proceedings [online]. Vol. 3. IEEE, 2007. Available from: https://doi.org/10.1109/ICNC.2007.537

[19] RAY, S. S., BANDYOPADHYAY, S., PAL, S. K. Genetic operators for combinatorial optimization in TSP and microarray gene ordering. *Applied Intelligence* [online]. 2007, **26**(3), p. 183-195. ISSN 0924-669X, eISSN 1573-7497. Available from: https://doi.org/10.1007/s10489-006-0018-y

[20] PAUL, P. V., RAMALINGAM, A., BASKARAN, R., DHAVACHELVAN, P., VIVEKANANDAN, K., SUBRAMANIAN, R., VENKATACHALAPATHY, V. Performance analyses on population seeding techniques for genetic algorithms. *IJET International Journal of Engineering and Technology*. 2013, **5**(3), p. 2993-3000. ISSN 2319-8613, eISSN 0975-4024.

[21] CHANG, D.-X., ZHANG, X.-D., ZHENG, C.-W. A genetic algorithm with gene rearrangement for K-means clustering, Pattern Recognition. 2009, **42**(7), p. 1210-1222. ISSN 0031-3203. Available from: https://doi.org/10.1016/j.patcog.2008.11.006.

[22] SALLABI, O. M., EL-HADDAD, Y. An improved genetic algorithm to solve the traveling salesman problem. World Academy of Science, Engineering and Technology. 2009, **52**(3), p. 471-474. ISSN 2010-376X, eISSN 2010-3778.

[23] HASSANAT, A., PRASATH, V., ABBADI, M., ABU-QDARI, S., FARIS, H. An improved genetic algorithm with a new initialization mechanism based on regression techniques. *Information* [online]. 2018, **9**(7), 167. ISSN ‎2078-2489. Available from: https://doi.org/10.3390/info9070167

[24] REINELT, G. TSPLIB [online] [accessed 2018]. University of Heidelberg 1996. Available from: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95

[25] DENG, Y., LIU, Y., ZHOU, D. An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering* [online]. 2015, **2015**, 212794. ISSN 1024-123X, eISSN 1563-5147. Available from: https://doi.org/10.1155/2015/212794