Optimization of the Formula-1 race calendar using genetic algorithm

Gabriella Polyák[1],Miklós Póth[2]
[1]Faculty of Technical Sciences, University of Novi Sad, Serbia, gabriellapolyak8@gmail.com
[2]Subotica Tech, Subotica, Serbia, pmiki@vts.su.ac.rs

2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY) | 978-1-6654-8988-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/SISY56759.2022.10036315

The aim of this work is to solve an optimization problem using the genetic method. This could be considered similar to the Traveling Salesman Problem (TSP), as the topic is route optimization. The study is based on the Formula 1 race calendar and the geographical location of the Grand Prix venues.

The optimization was done in two ways, with an add-on of Microsoft Excel, Excel Solver, and a program written in MATLAB. Excel Solver is based on: examining the possible solutions taking into account the constraints, then selecting the one that fits the most. The program written in MATLAB uses methods used in the genetic algorithm, such as selection, mutation, crossover and insertion.In this work, the results obtained in the two programs are compared and summarized. The Traveling Salesman Problem has been used many times to solve problems, but for this specific problem it has not been used yet.

Keywords: route optimization, genetic algorithm, Formula-1, problem solving

1 INTRODUCTION

Genetic algorithms are a group of search techniques, which can be used to search for an optimum or an element with a given property. Genetic algorithms are special evolutionary algorithms, borrowing their techniques from evolutionary biology. They have many fields of application, i.e., many types of problems can be solved with their help, such as e.g., subset selection, structure development, technical design, etc. Genetic algorithms are explained in detail in Chapter 2.

In Chapter 3 the final calendar for 2021 has been optimized, first with Solver, a built-in application from Microsoft Excel, and then with a program written in Matlab that uses methods of the genetic algorithm such as selection, mutation and crossover. Since Solver is a built-in feature, it is not possible to know what is really happening while the program is running, only the input and output data can be known. This was the reason why a program was written in Matlab, so that it is known exactly how optimization takes place. The 2021 season consisted of 21 Grand Prix venues. There are many factors that affect how a year's race calendar is compiled, and weather and financial matters play a major role in this.

Using the same method, the optimal route between all tracks has been calculated in the history of Formula 1 since 1950 until the end of the 2021 season. There are 74 of them. In Chapter 4 the analysis and comparison of the obtained results is done.

2 GENETIC ALGORITHMS

In biological evolution, development of living beings can be observed as complex systems. The evolution of living beings is a process in which living beings learn to adapt to their changing environment. This adaptation is not equally successful for all living things, so in addition to increasingly diverse forms, there is also selection. Since Darwin's "The Origin of Species" (1859), this evolutionary process has become increasingly the subject of research.

Genetic algorithms are heuristic search approaches that are applicable to a wide range of optimization problems. This flexibility makes them attractive for many optimization problems in practice [7]. The model of the genetic algorithm is, of course, far from the actual biological evolutionary process, only a few concepts and techniques can be paralleled, such as population, individual, selection, mutation, reproduction and the generations of actual living beings are replaced by a series of ever-changing populations. The science of genetics helps us to differentiate between heredity and variations and seeks to account for the resemblances and differences due to the concepts of Genetic Algorithms and directly derived from natural heredity, their source and development [8].

Although the birth of the genetic algorithm was clearly motivated by biological evolution, other features of the established model can be highlighted. It can be considered a search process, a learning algorithm, or a population-based algorithm in which certain operations are trained based on biological analogies or information exchange characteristics. Based on this approach, the evolutionary algorithm can be classified as an independent search process for the artificial intelligence problem, or among the algorithms of the learner, or even as a population-based algorithm for operations research[1].

2.1 Principle and methods of operation of genetic algorithms

The genetic algorithm seeks to solve the problem through a genetic process. The procedure works cyclically, working with several individuals (solutions) at the same time in each cycle. It produces new offspring by sorting from one population and then through various search operations. The ability of each individual is measured by the fitness function, which assigns a larger (smaller) number to the more capable individuals than to the less capable individuals. At the end of each iteration, it evaluates individuals and offspring, or just offspring, and forms a new population from them. Cycles, i.e., generations, are repeated until a desired condition is met (e.g., the number of generations reaches a desired value).

2.1.1 Individuals and population

The algorithm starts with an initial set of individuals called a population. Each individual is a potential solution to the problem. Individuals are made up of genes, usually

with binary representation. Since the method "treats" several individuals at once, the chance of finding a global optimum increases. The population evolves through the development of individuals. If the individuals are fitter, the population will get closer to solving the problem.

2.1.2 Fitness function

The ability of individuals is evaluated using the fitness function, it quantifies how fit (how competitive) they are.

2.1.3 Selection

By selection, is meant the process of selecting individuals from the population in order to pass on their properties to the next generation, the main idea of selection procedures has been proposed to accomplish this idea [10].Selection introduces the influence of the fitness function to the genetic algorithm optimization process. Selection must utilize the fitness of given individual, since fitness is the measure of the "goodness" of an individual [11].There are several types of selection techniques:

- Roulette wheel selection or fitness proportional selection (Figure 1): Where individuals with higher fitness values are more likely to cross over, but they may also be given a chance by underperforming individuals, albeit with a much lower chance. In this case, they are chosen randomly from the entire population, taking into account the proportions. It is also important to note that an individual may be selected several times.

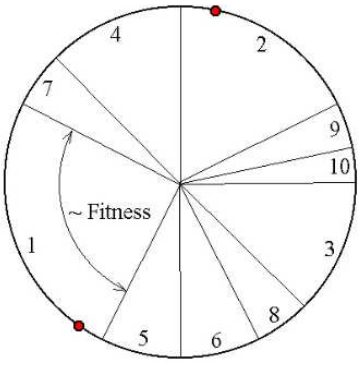

Figure 1. Roulette wheel selection [4]

- Tournament selection: When a randomly selected individual with a higher fitness value is crossed.
- Rank selection: Used when the algorithm gathers around the optimum, each individual has relatively the same slice on the roulette wheel. Instead of being competent, the opportunity for selection is allocated proportionally according to the skill ranking.
- Elitist selection: The most capable individuals are always transferred to the new population unchanged. This guarantees that the entire population will not be able to perform worse than the best solution found so far.

2.1.4 Mutation

In the case of mutation, genes of individuals are randomly changed, an example is shown in Figure 2. There are several possibilities for mutation:

- Replace bits from 0 to 1 and vice versa
- Replacement of the gene by a random number generated by even distribution between the upper and lower limits
- Adding a random value with a Gaussian distribution to the selected gene; if this is outside the specified lower or upper limit of the given gene, the new gene value is cut off at the limit

Initially, mutations result in an effective exploration of the state space, later their role is to break out of local minimums. However, for a population close to the global optimum, they are requested because they will be less fit than their counterparts.

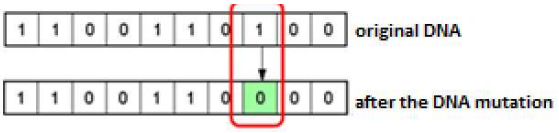

Figure 2. Mutation [4]

2.1.5 Crossover

Crossover or recombination means the process of uniting two selected individuals based on some "method". For each parent pair, one or two offspring are created. The advantage of the genetic algorithm comes mainly from crossing, so this part is presented in the most detailed way.

The crossing can be done in several ways, and it is always desirable to adapt the method to the problem:

- One-point crossing: Divides parents into two parts and replaces them with each other, as shown in Figure 3.

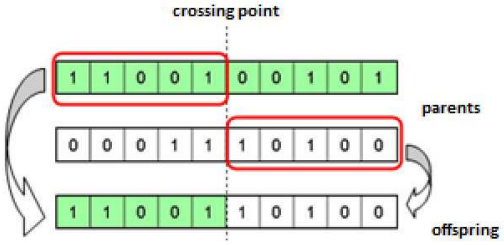

Figure 3. One-point crossing [4]

- K-accurate crossbreeding: Divides parents into k+1 parts and then creates the offspring from them, as shown in Figure 4.

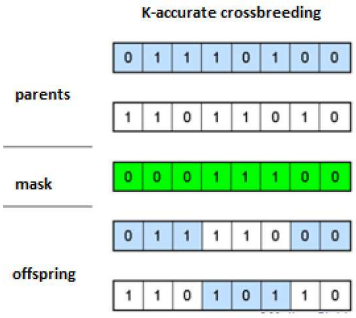

Figure 4. K-accurate crossbreeding [4]

- Uniform crossover - Each point can come from a certain probability, from one parent or another, as shown in Figure 5.
- Order crossover (OX): The first permutation is copied to the same part of the offspring by randomly selected section (8,7,3). The remaining locations are uploaded from the second permutation (parent), following the order, but omitting the points already used, an example is given in Figure 6. The offspring inherits from the first parent relative order, absolute positional and neighborhood information, while from the second only relative order information is inherited.

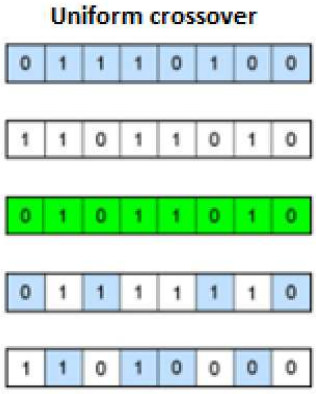

Figure 5. Uniform crossover [4]

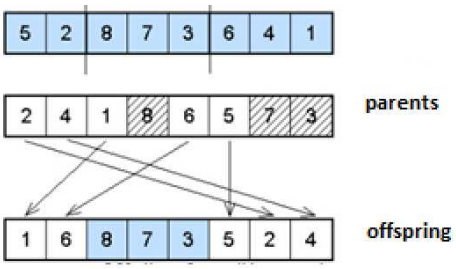

Figure 6. Order crossover (OX) [4]

- Order crossover 2: This method randomly selects a K point from the first parent, finds their location in the second parent, and places the K point in the places occupied in the second parent, but in the order specified in the first parent, places it in the offspring. The remaining points are copied directly from the second parent to the offspring.
- Partially mapped crossover (PMX): This operator was created by Goldberg and Lingle in 1985. The procedure differs from the method known at the cross-crossing in order in that it places points from the second parent differently (Figure 7). After the CDE excerpt copied from parent S1 (Figure 7 - 1), it first places in the U offspring the points that are not yet included in it and in the S2 parent are in the places that have already been filled with the offspring, these are points "b" and "a". Point b is placed by looking at which point in S1 is in place, this is point D, and then you place the b point in the place that d occupies in S2 (Figure 7 - 2). The placement of point "a": in its place there is E in S1, but in this place it is already C in U, so it is necessary to continue the method: c's location in U is still empty, so this is where the (Figure 7 - 3) is placed. It fills the remaining seats from S2 with dots in the same place (Figure 7 - 4).
- Edge crossing: The edge crossing procedure is a special operator for the traveling salesman problem. The algorithm takes into account the neighborhood properties of permutations, because this determines the length of a route. The procedure first expresses neighborhood information from the parents and stores them in a table, in fact drawing a graph containing the edges of both parents. Then, based on the table (graph), you create a new individual (route). The process of creating a new entity can reach a dead end several times while searching for the route – in Figure 8 it shows a possible successor [4].

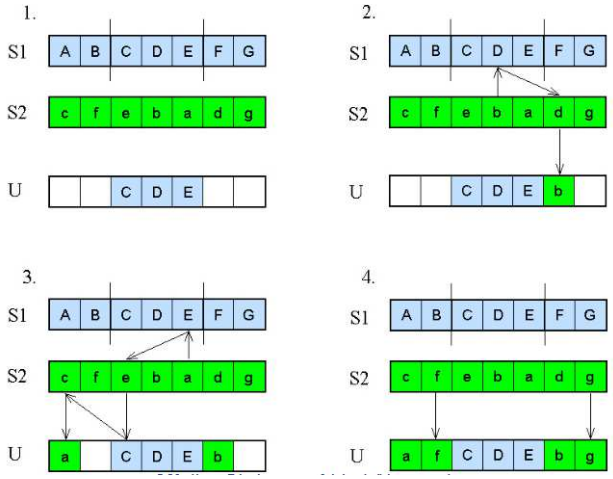

Figure 7. Partially mapped crossover (PMX) [4]

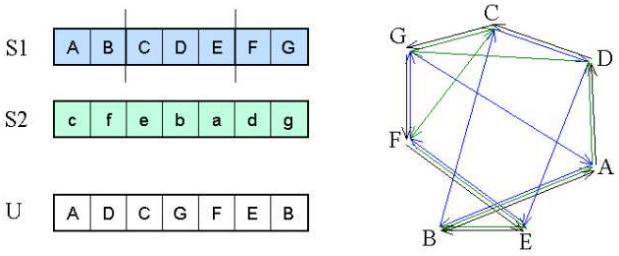

Figure 8: Edge crossing [4]

2.1.6 Reposition

After the creation of offspring, it is necessary to form a new population, in which even the offspring are already housed. The new population can be produced in several ways. The offspring can replace the previous population, in some proportion it will consist of individuals from the previous population and offspring, or, for example, the algorithm will decide on the basis of the fitness function which individuals of the new population will be [1].

2.2 Application areas

The term genetic algorithm, almost universally abbreviated nowadays to GA, was first used by John Holland, whose book Adaptation in Natural and Artificial Systems of 1975 was instrumental in creating what is now flourishing field of research an application that goes much wider than the original GA [9].

By adaptation, he meant a process that constantly modifies the structure of something, thereby achieving ever better efficiency in its environment. As an adaptive system, it investigated biological evolution, which allows adaptation to changing environmental conditions by generating ever-changing genetic structures and spreading them through inheritance.

In some ways, the production of increasingly efficient or increasingly adaptable structures is actually an optimization process. Of course, the overall achievement of optimum is not ensured by the adaptive system, but it can be easily improved to solve optimization problems. Thus, the evolutionary algorithm is considered as a method of optimization, which can be applied to the most complex problems. For many tasks, the solution of which can be traced back to optimization, the genetic algorithm can be successfully applied.

In addition to optimization, genetic algorithms are applied in various fields. In aircraft design, simulation of economic processes, work scheduling, setting market prices, designing regulators, retrieving information, etc. successful applications can be found.

The types of problems that can be classified in most application areas are:

1. Subset selection: a subset must be selected from a set of objects (e.g., error diagnostics).
2. Scenario generation: simulating the system so that the dynamics and developmental tendencies of the system can be examined in different conditions (e.g., economic process modeling).
3. Parameter setting: e.g., parameter estimation for mathematical formulas, neural networks.
4. Structure development: development of optimal, high-quality system structure (e.g., neural network structure, formation of clusters).
5. Assignment problems: Mapped a set of objects to another set of objects, taking into account the criteria you specify.
6. Technical design: determines the sequence of operations in which a starting state is transformed into a desired state [1].

3 Route Optimization

3.1 Excel Solver

Solver programs are mathematical optimization applications that can be used to solve a mathematical task. The goal is to create a general model with the help of which many similar problems can be solved. The Microsoft Excel add-in is the principle of Excel Solver to examine possible solutions, taking into account restrictions, and then select the best solutions for you.

Excel Solver can be used to find the optimal value (minimum, maximum, or specific target value) of a formula in the so-called target cell by setting constraints or restrictions in the values of other formula cells on the worksheet. To do this, Solver uses a group of cells called decision variables or simple variable cells that can be used to calculate formulas in the target value or constraint cell. Solver modifies the values of the decision variable cells to meet the constraints of the constraint cell and produce the result you want for the target cell.

3.1.1 The 2021 calendar

The 2021 calendar consisted of 21 Grand Prix venues and 22 races. Once there was a double weekend, which means that two consecutive weekends of Grand Prix were held at the same Grand Prix venue with two Grand Prix of different names.

Table 1 shows track names and their geographical coordinates.

The next step is to calculate the distance between the geographical coordinates of the tracks using the Euclidean method.

In the left part of Table 2, distances, order and their total are shown before optimization, and this is the order the races originally took place in the race calendar.

The original distance was 630.497 (66288.536 km), while the optimized result was 396.3041 (41957.074 km). This is a significant improvement. It can be stated that Excel Solver has done an excellent job.

TABLE I.
NAMES AND GEOGRAPHICAL COORDINATES OF THE TRACKS

	Circuit	Latitude	Longitude
1	Bahrain International Circuit	26.0325	50.510556
2	Imola Circuit	44.34111	11.713333
3	Algarve International Circuit	37.232	-8.632
4	Circuit de Barcelona-Catalunya	41.57	2.261111
5	Circuit de Monaco	43.734722	7.420556
6	Baku City Circuit	40.3725	49.853333
7	Circuit Paul Ricard	43.250556	5.791667
8	Red Bull Ring	47.219722	14.764722
9	Silverstone Circuit	52.078611	-1.016944
10	Hungaroring	47.582222	19.251111
11	Circuit de Spa-Francorchamps	50.437222	5.971389
12	Circuit Zandvoort	52.388819	4.540922
13	Monza Circuit	45.620556	9.289444
14	Sochi Autodrom	43.410278	39.968271
15	Istanbul Park	40.951667	29.405
16	Circuit de Americas	30.132778	-97.64111
17	Autódromo Hermanos Rodriguez	19.406111	-99.0925
18	Interlagos Circuit	-23.70111	-46.69722
19	Losail International Circuit	25.49	51.454167
20	Jeddah Corniche Circuit	21.631944	39.104444
21	Yas Marina Circuit	24.467222	54.603056

TABLE II.
ROUTE LENGTH BEFORE AND AFTER OPTIMIZATION

1	42.9002	7	3.9101
2	21.5516	4	11.7251
3	11.7251	3	71.8457
4	5.5952	18	67.8491
5	42.5658	17	10.8244
6	44.1556	16	99.0851
7	9.8117	9	5.5665
8	16.5127	12	2.4197
9	20.7608	11	9.3635
10	13.5832	8	4.5010
11	2.4197	10	12.1271
12	8.2679	15	10.8456
13	30.7583	14	10.3413
14	10.8456	6	16.5993
15	127.5059	21	3.3108
16	10.8244	19	1.0884
17	67.8491	1	12.2256
18	109.7883	20	35.5806
19	12.9383	2	2.7408
20	15.7558	13	2.6550
21	4.3816	5	1.6993
1		7	
Total	630.4969	Total	396.3041

3.1.2 All 74 tracks so far

In the history of Formula 1 since 1950, there were a total of 74 circuits by 2021. It is an interesting optimization task to find the shortest route between them. The optimization was also based on the geographical coordinates of the tracks and the distance calculated by the Euclidean method between them.

When the tracks were set in alphabetical order, the distance between them was 5232.484 (553973.546 km), after which Excel Solver was used and after optimization this distance was cut down to 953.542 (139465.186 km). Solver again performed well, finding a much shorter optimal route than the original one.

3.2 Matlab

The program, written in Matlab, uses methods used in genetic algorithms such as selection, mutation, crossover and reposition. The program is looking for the optimal route for 8 individuals at the same time, but the program could be run for arbitrary number of individuals. In order to get the best, (optimal) result, it is better to run the program for as many generations as possible, to a certain point until no improvement could be seen in the result.

3.2.1 Calendar for 2021

Figure 9 shows the coordinates of the tracks specified in the coordinate system and the optimal route drawn as a result after optimization for the 2021 race calendar. After optimization, the result obtained, i.e., the optimal route length is 397.38 (40848.594 km).

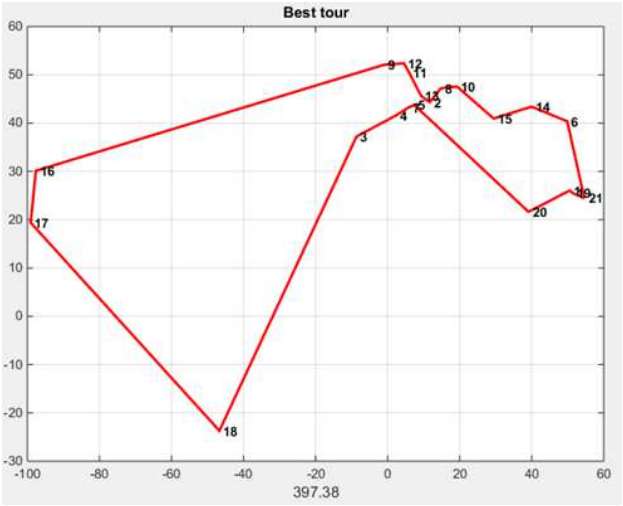

Figure 9. Path after optimization shown in graph

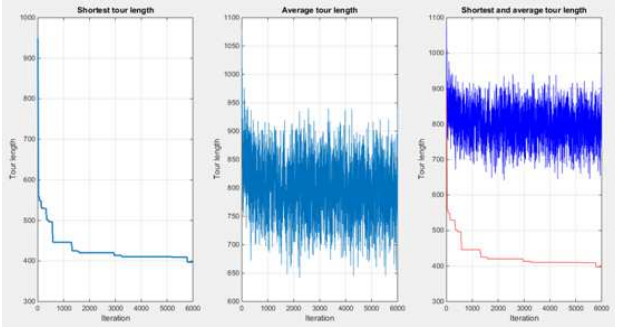

Figure 10. The shortest route, the average route and the two compared in one graph

The program was run on 6000 generations (Figure 10). Increasing the number of generations above that number resulted in no significant improvement, and the program did not produce a shorter route. The figure shows three graphs, the value of the shortest routes on the left, the average paths in the middle and both are compared on the right.

3.2.2 All 74 tracks so far

After a slightly simpler task, a more complicated one was given to the program written in Matlab. It worked with the same coordinates in the same order as Excel, but produced a different result. The optimal route is shown in Figure 11. The program was run on 50 individuals and 10000 generations.

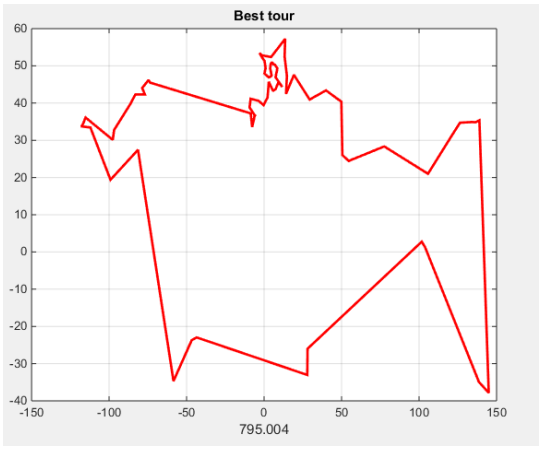

Figure 11. The route after optimization and its length

It can be seen that after running the program, the result was 795.004 (86131.007 km), which is better than the one obtained in Excel. Figure 12 shows the optimal route projected onto the world map.

Figure 12. The optimized route projected onto the world map

Figure 13 shows that optimization started in the same way as in Excel, but the optimal route length was shorter.

The program can be run with Partially Mixed Crossover (PMX) and Order crossover (OX) or without crossover. Similar results are achieved on all three occasions. When the program runs without crossover, it runs much faster, but early convergence occurs, i.e., it reaches the optimum quickly, and then it does not improve further.

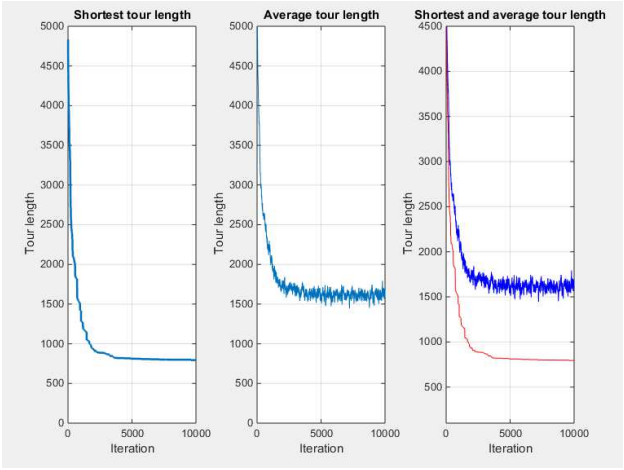

Figure 13. The shortest, the average route and the two are compared in one graph

3.2.3 2-opt method

In 1958, Croes invented the method to solve the Travelling Salesman Problem. It is an iterative correction algorithm, so it is developing an already existing path.

The 2-opt algorithm works as follows: it takes 2 arcs from the route, reconnect these arcs with each other and calculate new travel distance. If this modification has led to a shorter total travel distance the current route is updated. The algorithm continues to build on the improved route and repeats the steps. This process is repeated until no more improvements are found or until a pre-specified number of iterations is completed(Figure 14)[13].

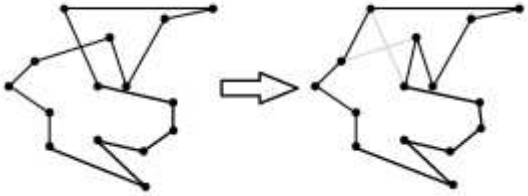

Figure 14. 2-opt example

The program that includes 2-opt method has also been incorporated. The advantage of it is a visibility of changing edges as the route becomes shorter and shorter. Similar results were achieved as with the aforementioned methods: 40848.594 km for optimization of the annual 2021 calendar and 86181.925 km for all Grand Prix venues.

4 CONCLUSION

This paper consists of theoretical and practical parts. In the theoretical part, the concept, principle of operation and fields of application of genetic algorithms are introduced. It has many areas of application, i.e., many types of problems can be solved with using GAs, such as subset selection, structure development, technical design, etc.

In the practical section, route optimization was performed using the Solver add-in in Microsoft Excel and Matlab. Both methods are based on the same basis, using a genetic algorithm. Since Solver is a built-in feature, it is not exactly known what really happens and how its engine works. On the other hand, in Matlab, each step can be followed and how optimization actually happens. The 2021 calendar has been optimized, which includes 21 Grand Prix venues, so the route between 21 pairs of coordinates had to be optimized, so the shortest one was to be found. The distance between the coordinates was calculated using Euclidean methods, with these distances later worked.

Solver can only achieve good results with a certain amount of input data, because when all the Grand Prix locations were given as input data, it could not perform nearly as much optimization as before with less data.

As for the race calendar in Excel, the length of the pre-optimization route in km is 66288.536 km, while the optimized route was 41957.074 km. In Matlab (both with crossover and 2-opt) optimization, 40848.594 km was obtained (approx. the length of the equator). The difference between the two methods is almost 1000 kilometers.

As for all the 74 tracks, the length of the route before optimization was 553973.546 km. In Excel it is 139465.186 km after optimization and in Matlab it is 86131.107 km. In Matlab, the optimal route is more than 50000 kilometers shorter than in Excel. Both methods found a much shorter path than the original, but the program written in Matlab in this task performed better than Excel. With 2-opt. method the optimal route is 86181.925 km long(a little more than double the length of the equator), so there is only a 50 km difference between crossing methods and 2-opt for such a distance. Both methods worked much better than Excel. The next step would be to incorporate the 3-opt method into the program.

REFERENCES

[1] Borgulya István: Evolúciós algoritmusok, Dialóg Campus Kiadó, 2004, ISBN 963-9542415

[2] John Henry Holland: Adaptation in Natural and Artificial Systems, 1975

[3] Bence Keresztury: Genetic Algorithms and the Traveling Salesman Problem, 2017

[4] https://mogi.bme.hu/TAMOP/szamitogepes_szimulacio/ch05.html

[5] Takács Á., dr. Kamondi L. - A genetikus algoritmusok, 2006

[6] Kulcsár Z. – Az utazóügynök probléma és alkalmazásai, 2017

[7] Kramer, O. (2017). Genetic Algorithms, Studies in Computational Intelligence, 11-19. Doi: 10.1007/978-3-319-52156-5_2

[8] Sivanandam, S. N., & Deepa, S. N. (2008). Genetic Algorithms. Introduction to Genetic Algorithms, 15-37. Doi: 10.1007/978-3-540-73190-0_2

[9] Reeves, C.R. (2010) – Genetic Algorithms. International Series in Operations Research & Management Science, 109-139. Doi: /10.1007/978-1-4419-1665-5_5

[10] Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic Algorithms. Search Methodologies, 97-125. https://doi.org/10.1007/0-387-28356-0_4

[11] Shukla, A., Pandey, H. M., & Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). Doi: 10.1109/ablaze.2015.7154916

[12] Rishal Hurbans: Algoritmi veštačke inteligencije, edicija Grokking, Kompjuter biblioteka – Beograd (2021) ISBN: 978-8673105611

[13] Michael Negnevitsky: Artificial Intelligence, A Guide to Intelligent Systems. Pearson Education Canada; 3rd edition (2011) ISBN: 978-1408225745