

# Queen Bee Based Genetic Programming Method for a Hive Like Behavior

Veronika Bojtár, János Botzheim

*Department of Mechatronics, Optics and Mechanical Engineering Informatics*

*Faculty of Mechanical Engineering*

*Budapest University of Technology and Economics*

*4-6 Bertalan Lajos Street, 1111 Budapest, Hungary*

e-mail: veronika.bojtár@gmail.com, botzheim@mogi.bme.hu

**Abstract**—Designing the behavioral attributes of a robot is challenging, and the complexity of this task is even more increased in the case of swarm robotics. For effectively solving such problems special types of evolutionary algorithms can be used such as Genetic Programming and Queen Bee Based Genetic Programming method. The revolutionary idea behind these algorithms is that they use tree based representation for the individuals in a population, thus being able to solve structure optimization problems. The goal of this paper is to introduce the idea of the Queen Bee Based Genetic Programming method and compare its effectiveness with Genetic Programming through the evolution of a successful hive based behavioral program.

**Index Terms**—genetic programming, queen bee based genetic programming, hive like behavior

## I. INTRODUCTION

Genetic Programming (GP) and the Queen Bee method are both different types of evolutionary algorithms. These algorithms are inspired by biological evolution, utilizing its random nature to evolve adequate solutions to the presented problems. The process starts with a population consisting of randomly created individuals (each a viable, though not necessarily optimal solution to the presented problem), and through many iterations they become better, evolving through generations.

This paper proposes a novel variant of the Genetic Programming technique called Queen Bee Based Genetic Programming (QBBGP) which is a combination of Genetic Programming technique and Queen Bee evolution.

The problem presented in this article is to create the behavioral program of a hive whose goal is to find food piles in a map, and take the food they find back to their home location. This task can be achieved more effectively by utilizing the insect's cooperative nature, and it is an important part to emerge during the evolution of the behavioral program.

This task is originated from J. R. Koza's idea [1], and in this article it is used as the goal of the evolution process because it is a great example of how these methods can be utilized to solve very complicated problems. Moreover, with this example it is easy to compare the effectiveness of the two algorithms and show how they react to different circumstances.

## II. RELATED WORKS

### A. Genetic Algorithms

The idea of Genetic Algorithms were introduced by John Holland in the 1970's [2]. The individuals to be evolved have a string-like structure, and genetic operators are performed on them. These genetic operators are the selection, reproduction, crossover, and mutation.

Selection refers to selecting members from the current generation for reproduction and crossover. This can be carried out in various ways, for example by fitness proportionate selection (where each individual has a chance for being selected according to its fitness value compared to the total fitness of the population) or stochastic universal sampling (which is a development of fitness proportionate selection).

Reproduction is the process of copying a defined number of members of the previous generation to the next one without modifying them. This allows us to preserve good solutions. Crossover is done by selecting two individuals and swap parts of them in order to create two new individuals. Finally, mutation is randomly selecting part(s) of individuals and creating new parts instead of them. This operation is very useful for cases when the individuals are converging to a local optimum. This mutation operator has the ability to make them jump out of those local optima, so that they can continue to approach the global optimum.

This theory has been improved since then by many others. One such development is the so called Queen Bee method.

### B. Queen Bee Evolution

This method was suggested by S. H. Jung [3] as an enhancement of the Genetic Algorithm. The difference between the original and the new method lies in the way the genetic operators are performed. In the original Genetic Algorithm individuals for crossover are selected by a selection algorithm usually based on their fitness, and from this group they are selected two-by-two to make individuals for the new generation. In the Queen Bee method, only one of the two parents come from selection, the other one is always the best one in the current generation - very much like how bees reproduce in real life.

The process of mutation is also modified in the Queen Bee method. Only one part of the population is mutated with a

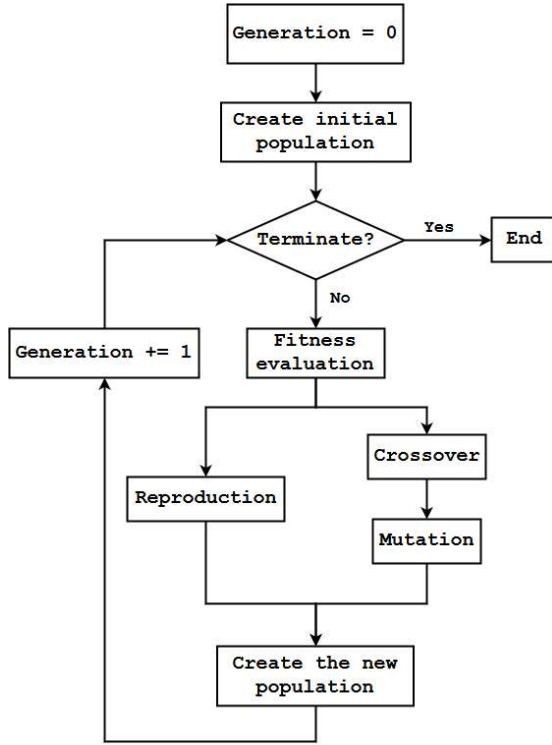


Fig. 1. The flowchart of Genetic Programming.

normal mutation rate, the other part is mutated with a so called *strong mutation rate*. The rate of how large part of the population should be mutated with the normal and how large should be mutated with the strong mutation rate are decided by a  $\xi$  parameter.

### C. Genetic Programming

Genetic Programming was invented by J. R. Koza in the early 1990's [4]. It can be categorized as a type of evolutionary algorithm, because the main idea is the same as in the case of any of them, but the specialty here is that individuals in a population are represented by tree structures, thus expanding the scope of problems to which this method can be used [5]–[7]. The flowchart of GP is presented in Fig. 1. The main genetic operators of GP are also reproduction, crossover, and mutation.

For executing the crossover operation illustrated in Fig. 2, first, two parents have to be selected from the population. After selecting the parents, a random node is chosen in both of them, then the sub-trees of the nodes are swapped, creating two children. This process is repeated until the required number of new individuals is reached. This type of reproduction method is called sexual reproduction, because it needs two parents for creating children, and the offspring will contain genetic information from both parents.

In the mutation process illustrated in Fig. 3, some of the individuals of the population are mutated. In the case of this tree-based representation this means that one node is selected

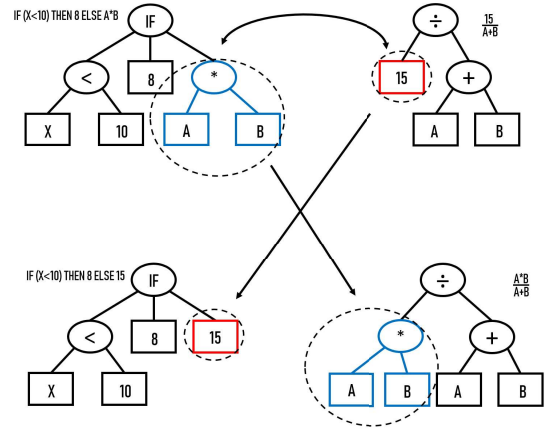


Fig. 2. Illustration of crossover operator.

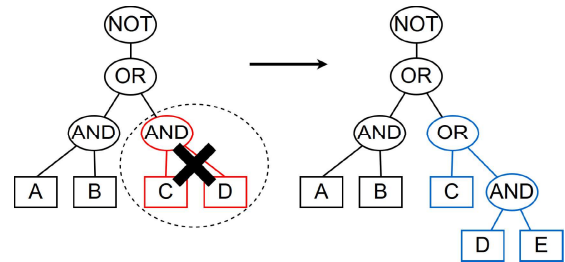


Fig. 3. Illustration of mutation operator.

randomly in the individual's tree structure and a new sub-tree is created in its place.

## III. PROBLEM FORMULATION

As learning algorithms appear more in the focus as methods to solve complex problems, we need to continuously improve them. This also aligns with the goals that this work aspires to achieve.

This article compares the performance of the two methods through a relatively easy and tested problem, shifting the focus on the properties of the algorithms from the problem itself.

Properties compared must include general properties such as best solution achieved or the convergence speed of the solution, but the focus of this article must be those properties that best differentiate the two approaches.

## IV. QUEEN BEE BASED GENETIC PROGRAMMING

This method is the implementation of the Queen Bee algorithm with the use of genetic programming-like structures. As explained above, the Queen Bee method itself is a good improvement of the Genetic Algorithm, but since it uses string-like structures it cannot be used for so broad range of problems like the Genetic Programming method. Combining the strengths of the two methods creates a powerful algorithm.

The basic operators remain the same as in Genetic Programming, but the execution of the crossover and the mutation is quite different.

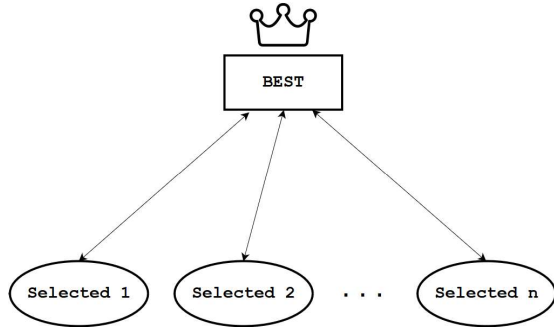


Fig. 4. Queen Bee reproduction.

When choosing parents for crossover only one of the two parents is selected by the above mentioned process. The other parent is always the best one in the current generation as depicted in Fig. 4, thus ensuring the subsistence of the genetic information of the best individual.

This reproduction method is very similar to the reproduction of bees in nature – where every male mates with the queen bee – this is why this algorithm is called Queen Bee Based Genetic Programming method.

The process of mutation is also modified in this new method. Only one part of the population is mutated with a normal mutation rate, the other part is mutated with a so called strong mutation rate. The rate of how large part of the population should be mutated with the normal and how large should be mutated with the strong mutation rate are decided by a  $\xi$  parameter.

## V. SIMULATION RESULTS

During the simulations, the most limiting factors for finding optimal solutions were the size of the map and the number of food pieces that can be found on it. This is evident since we have a limited number of time steps (400) for the simulation, and if the map is too big, or there is too much food to be taken back to the nest, the insects won't have time to complete the task. On the other hand, it is still interesting to examine how the algorithms react to unsolvable problems.

The simulations were carried out on three different maps which can be seen in Figure 5, Figure 6, and Figure 7. The smallest map is  $24 \times 24$  in size and contains 52 food pellets, the medium sized is  $26 \times 26$  with 80, and the big one is  $32 \times 32$  with 120 food pellets. These food pellets are organized into groups as seen in the figures. The nest of the ants is in the top-left corner – indicated as a black rectangle.

Since all these methods are stochastic methods, multiple runs are necessary for analyzing them. For the smallest and the medium sized map for every parameter setting nine simulations were run, and the results are obtained from the average of them. For the biggest map, four simulations were performed. The ripples in the curves are originating from both the selection method – in case of non-elitist selection there is some chance that the best individual won't be copied into the next generation, and the nature of the simulation – even with

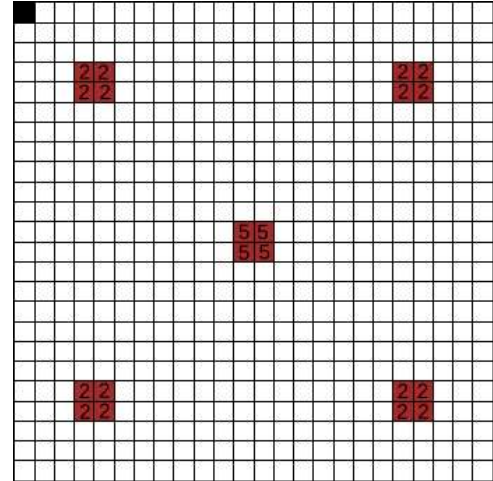


Fig. 5. Small map with 52 food on it.

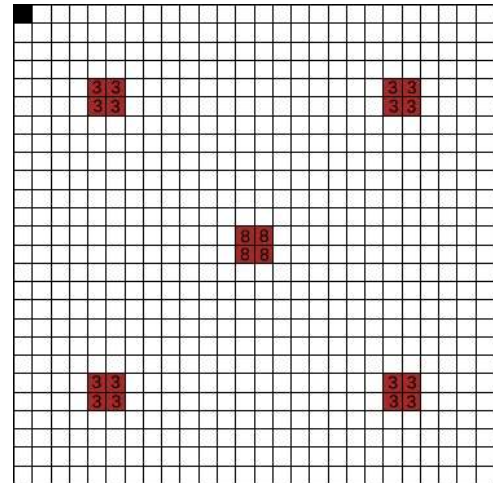


Fig. 6. Medium map with 80 food on it.

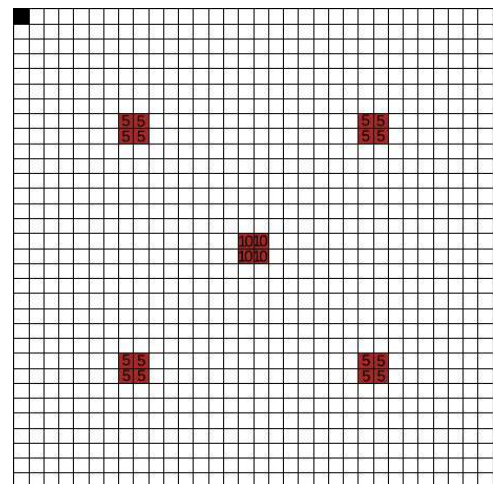


Fig. 7. Big map with 120 food on it.

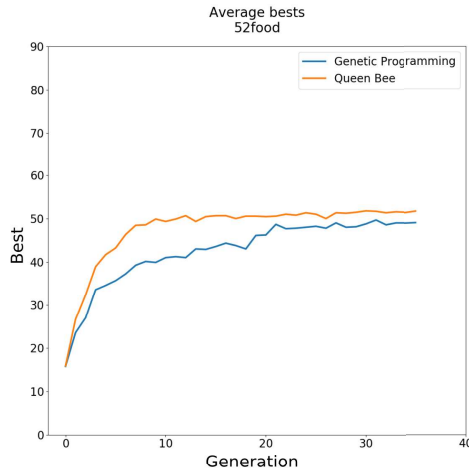


Fig. 8. Smallest map average bests results.

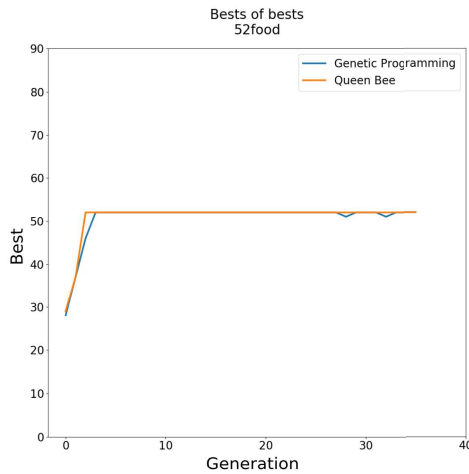


Fig. 9. Smallest map best results.

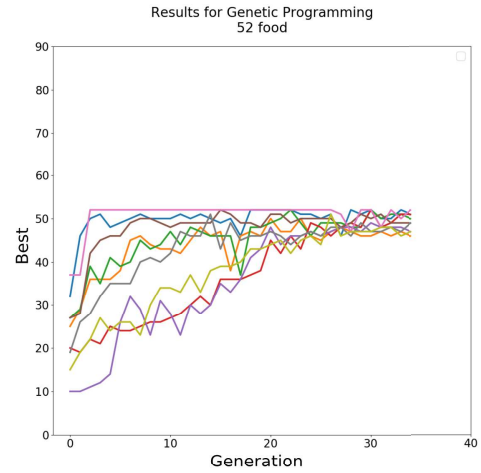


Fig. 10. Smallest map Genetic Programming runs.

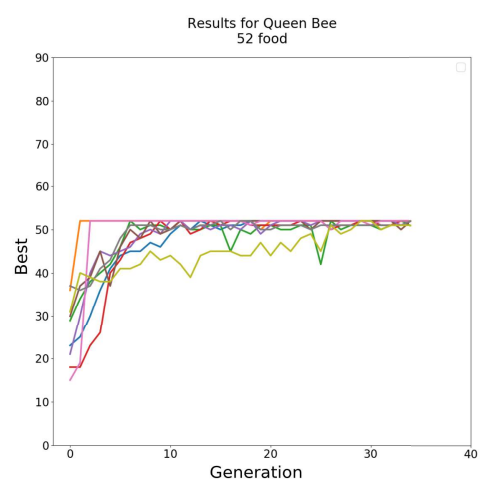


Fig. 11. Smallest map QBBGP runs.

the same behavior tree we won't always get the exact same result, since an important item in the program is the random searching motion of the ants.

#### A. Smallest Map with 52 Food

Figure 8 shows the average bests of the runs from the Queen Bee based and the original Genetic Programming method using the smallest map. This gives an overview of the general performance of the algorithms. In this and every consecutive figure the term "Queen Bee" refers to the Queen Bee Based Genetic Programming. The average best for a generation is calculated by looking at all the values for the different runs at that generation step and taking the average of them.

Figure 9 shows the best result in each generation selected from the multiple runs, this is indicated as "bests of bests" in the figures. This gives an overview of the maximum capacity of the algorithm.

From these figures it can be seen that both algorithms perform relatively well, and the best results of both of them reach the maximum value very quickly. However, it is also apparent

that the Queen Bee Based Genetic Programming method performs better than the original Genetic Programming. For the purpose of analyzing this, Figure 8 is more useful. It shows that the Queen Bee Based Genetic Programming is faster, and reaches a better average maximum than Genetic Programming does. What is also very important that the *average* of the runs reaches a value very close to the maximum – which was 52 – while in the case of Genetic Programming this is not the case.

Figures 10 and 11 show the individuals' simulation results from which the average and bests were obtained. The results obtained using Genetic Programming are more spread out at the final generations than the results obtained by using QBBGP.

It is also useful to examine the robustness of the methods. One way of this is to decrease the population. This creates a smaller initial resource pool for the algorithms to work with, and also the chance of having initially very good solution is smaller. Figures 12 and 13 show the results for this decreased populations.

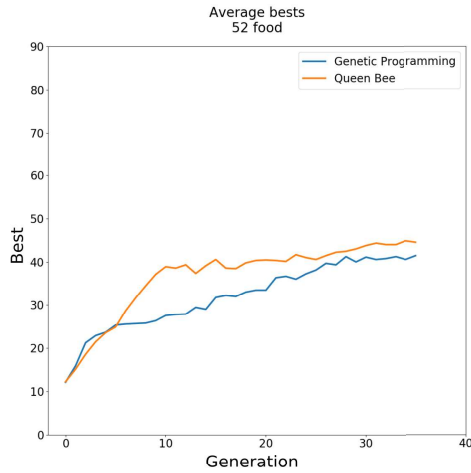


Fig. 12. Smallest map average bests results with decreased population size.

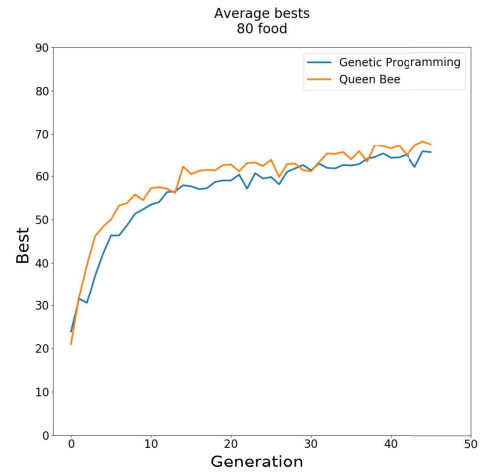


Fig. 14. Medium map average bests results.

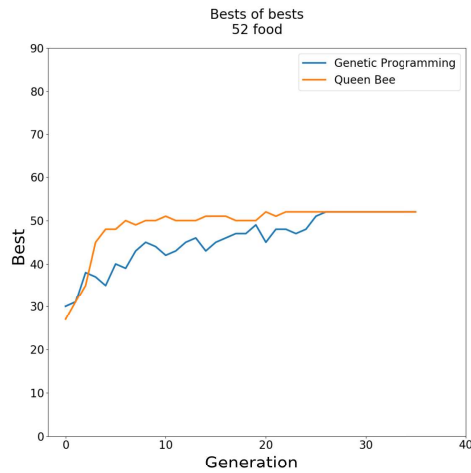


Fig. 13. Smallest map best results with decreased population size.

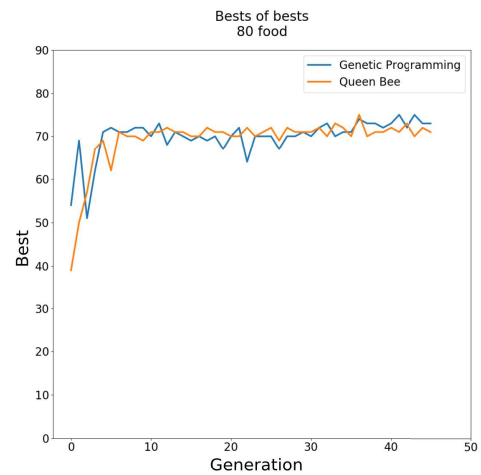


Fig. 15. Medium map bests results.

In this case also the QBBGP gives better results. Here examining the bests of the best results tells us more. Both curves reach the maximum, however the one belonging to the QBBGP reaches it very much faster while GP struggles to find an optimal solution, and the curve of QBBGP itself is also smoother. This smoothness can occur because of the speciality of the crossover – it is more likely to preserve good solutions by the method the QBBGP uses.

#### B. Middle Sized Map with 80 Food and Biggest Map with 120 Food

Figure 14 shows the average best results for the medium sized, and Fig. 16 shows the average best results for the biggest map. Figures 15 and 17 depict the best in each generation for the medium sized map and for the biggest map, respectively.

In these cases the map is so big, that the insects cannot collect all food in time, so these problems can also be used for the examination of the behavior of the algorithms in sub-optimal circumstances.

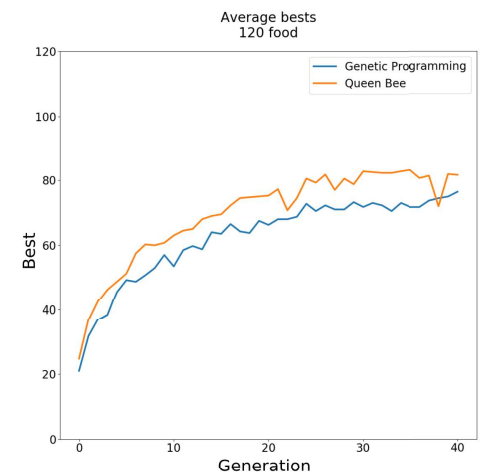


Fig. 16. Big map average bests results.



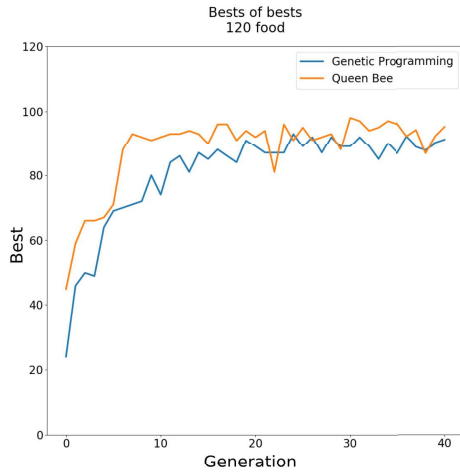


Fig. 17. Big map bests results.



Fig. 18. Example of a behavior tree.

Looking at the curves which show the average values, it can be stated that QBBGP is better for both sizes, it gives higher values faster than GP does. The bigger ripples can occur because of the size of the map is increased and there is more room for random motion than it was in the case of the smallest map.

Figure 18 illustrates an example of a behavior tree obtained during the optimization process. The possible nodes from which a tree could be built of in this task are the followings. “If carrying food” represents a function which checks whether the ant is currently carrying any food pellets in that timestep

and then decides the next action based on that. “If food here” checks whether there is food where the ant is standing. “Progn” is used for running multiple child functions without any conditions. “Move to adjacent food Else” makes the ant move to the next food pellet – if there is any – neighboring the ants location, if there are no such food it carries out its child function. “Move to adjacent pheromone Else” works similarly as “Move to adjacent food Else”, except now the ant wants to move to the neighboring pheromone spot. With “Move randomly” the ant is instructed to take a two-step long motion in a random direction and “Move towards nest” instructs the ant to move one step toward the nest. “Pick up food” makes the ant pick up a food pellet – if there is any available and with the “Drop pheromone” command the ant emits a pheromone cloud.

## VI. CONCLUSION

The paper proposed a novel Genetic Programming variant, the Queen Bee Based Genetic Programming. In almost all cases the selected best results have a very close value for GP and QBBGP, and this is logical – the two methods are based on the same operations, so statistically it is logical to have similar appearances. However, in average QBBGP always gives better results and faster.

For future work we intend to try the proposed Queen Bee Based Genetic Programming in other applications as well. We also aim to use its basic idea in other evolutionary algorithms.

## ACKNOWLEDGMENT

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

János Botzheim was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

## REFERENCES

- [1] J. R. Koza, “Genetic Evolution and Coevolution of Computer Programs,” *Langton et al. (editors): Artificial Life II, Addison-Wesley, Redwood City, CA*, pp. 603–029, 1990.
- [2] J. H. Holland, *Adaption in Natural and Artificial Systems*. The MIT Press, Cambridge, MA, 1992.
- [3] S. H. Jung, “Queen-Bee Evolution for Genetic Algorithms,” *Electronic Letters, Volume 39*, pp. 575–576, 2003.
- [4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- [5] A. Adhikari, N. Adhikari, and K. C. Patra, “Genetic programming: A complementary approach for discharge modelling in smooth and rough compound channels,” *Journal of The Institution of Engineers (India): Series A*, vol. 100, no. 3, pp. 395–405, 2019.
- [6] Q. U. Ain, H. Al-Sahaf, B. Xue, and M. Zhang, “A genetic programming approach to feature construction for ensemble learning in skin cancer detection,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. Association for Computing Machinery, 2020, p. 1186–1194.
- [7] M. Kovacic and U. Zuperl, “Genetic programming in the steelmaking industry,” *Genetic Programming and Evolvable Machines*, vol. 21, no. 1–2, pp. 99–128, 2020.