

Genetic Diversity in the Multiobjective Optimization of Paths in Graphs

Lukasz Chomatek

Abstract Existing systems that allow Users to plan the route, usually do not support the multiple criteria during the search process. In the sense of the multi-criteria optimization, such a situation involves the search for the Pareto-optimal solutions, but the present services use a weighted sum of the supported criteria. For solving Multiobjective Shortest Path problem we incorporate genetic algorithms with modified genetic operators, what allows the reduction of the search space. In this paper we compare genetic diversity in the algorithms which incorporate our method. Conducted research shown that proposed modifications allowed to obtain better diversity without either changing parameters or apply some rules to the algorithm.

Keywords Genetic algorithms · Shortest path problems

1 Introduction

However the single objective shortest path (SOSP) problem can be solved in almost constant time with a reasonable operational memory allocation [1], while considering multiobjective case of finding paths between two nodes in graphs, there exists no solution with a comparable abilities. One of the most popular applications of such algorithms are the online services that allow users to plan an itinerary of the trips.

Such a services usually take into account more than one criterion that can be associated with each segment of the road network. I.e., one can decide to choose either the shortest route or the route with the smallest travel time. In some cases all of the criteria considered by a service are combined into their weighed sum, to present the path which fits the preferences of its typical users. The situation, where weights of the road network segments associated with some number of criteria are substituted with a single value is actually a single objective optimization.

Ł. Chomatek (✉)

Institute of Information Technology, Lodz University of Technology, Lodz, Poland
e-mail: lukasz.chomatek@p.lodz.pl

In the Multiobjective Shortest Path (MOSP) problem, the solution consist of all paths which are not dominated by any other paths. Such a set of paths is called a Pareto set. This means that there is no path outside a Pareto set that is better than any path in this set with respect to all of the criteria. In the multiobjective case, the set of Pareto paths can be presented to the user, who can choose one of the paths arbitrarily.

Algorithms for solving the MOSP problem are usually the extensions of the single objective algorithms. One of the most popular deterministic algorithms for this scenario is Shortcuts-Arcs (SHARC) algorithm [2] which reduces the search space to find a solution in an acceptable time.

Excepting the deterministic algorithms, there exist many heuristic algorithms that can be successfully applied to the problem of multiobjective optimization. There exist methods based on the tabu-search, simulated annealing and the genetic algorithms. In our work, we focus on the last group of mentioned algorithms. Due to the size of the search space in the MOSP problem, the genetic algorithms tends to stuck in the local minima. Such a phenomenon is called a premature convergence. It usually occurs when one or more individuals with a good values of the fitness function are frequently chosen for the crossover operation. In some number of epochs they can dominate the population what's consequence is an insufficient exploration of the search space. However there exist some methods that address the mitigation of the influence of premature convergence, it is still a problem to maintain it effectively.

In this work, we examined the phenotypic and genotypic diversity of the population which can be directly applied to different genetic algorithms. During the tests we included our modifications of the genetic operators, which allow to reduce the size of the search space in the MOSP problem.

2 Related Work

The phenomenon of the premature convergence was described shortly after the introduction of genetic algorithm paradigm. The idea of promoting the solutions with the best values of the objective function causes the problem when some individuals dominate the population. In the following section we describe possible ways of its mitigation. Then we discuss the application of the genetic algorithms to the problem of multiobjective shortest path, with focus on our former improvement associated with the reduction of the search space.

2.1 *Methods of Preventing the Premature Convergence*

Present methods of preventing the premature convergence can be related to [3–5]:

- crossover and mutation operator,
- selection operator,
- model of the population.

Regarding the crossover operator, there were attempts to preserve some patterns encoded in the individuals. Many of these operators were firstly applied for the Traveling Salesman Problem, but can be utilized for other combinatorial problems like MOSP. Maximum Preservative Operator [6] excluded the possibility of crossover that breaks the common sequence in the mated chromosomes. Greedy Crossover [7] applied some conditions that disallowed the operation in case of getting offspring worse than the ancestors.

In case of mutation, Kureichick et al. [7] proposed Social Disaster Technique. When the algorithm suspect that the premature convergence may occur, a catastrophic operator is applied to get the genetic diversity back. The variants of this operator are:

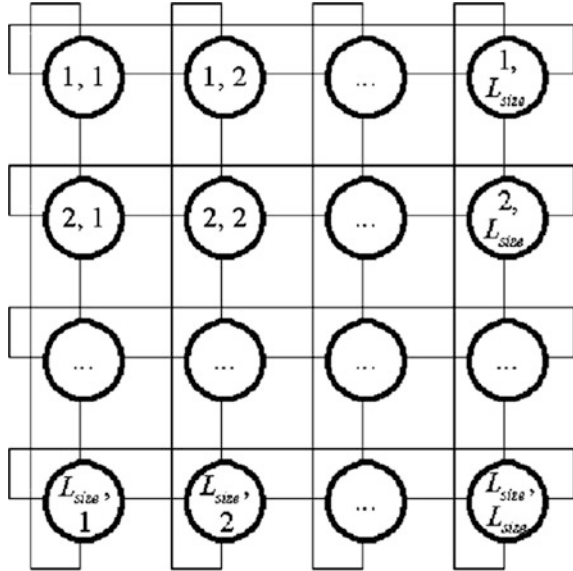
- Packing—while more than one individual have the same value of the fitness function, only one remains unchanged and all the others are fully randomized.
- Judgment day—only the individual with the best value of objective function remains unchanged and all the others are substituted by random chromosomes.

The idea of changing the selection operator depends on the observations of the Pareto front in the current iteration. The most popular algorithms in this category are Nondominated Sorting Genetic Algorithm II (NSGA-II) [8] and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [9]. In the first algorithm, the individuals are sorted with respect to the dominance relation. Then, sorting with respect to crowding distance is applied. This modifications promotes crossover that can preserve the genetic diversity. What is more, Nicoara [5] proposed dynamic mutation and crossover operators for NSGA-II to improve the performance of the algorithm. In the SPEA2 algorithm, the individuals are ranked with respect to the domination relation. Proposed ranking is treated as an objective function during a selection for crossover.

The second method of preserving the genetic diversity is to impose some restrictions on the individuals that can be mated with a certain individual without interposing the selection operator. Zhong [10] combined the paradigm of multi-agent systems and genetic algorithms. The individuals lived in a lattice and can interact only with their neighbors (Fig. 1). Furthermore author proposed unique operators like self-learning to improve the performance of the algorithm.

This influenced the possibility of obtaining the premature convergence, as very good individuals could not dominate the population in a small number of iterations. Another interesting algorithm is Hierarchical Fair Competition described in [11], where the population is divided into groups named islands. In each island live

Fig. 1 Population in multi-agent genetic algorithm [10]



individuals with a similar value of the objective function. This causes that elitist individuals does not compete with the worst and as a result population can evolve in different directions. After a certain number of iterations, a migration between islands is performed in order to preserve the genetic diversity on each island.

2.2 Genetic Algorithms for Solving MOSP

Let $G = (V, E)$ be a graph, where V and E represent its nodes and edges, respectively. Let $c = c_1, c_2, \dots, c_n$ be a cost associated with the edge $e \in E$. Moreover we denote a path between nodes s and t as $p(s, t) = \langle s, v_1, v_2, \dots, v_n, t \rangle$, where $v_1, v_2, \dots, v_n \in V$ denote nodes. In the path, all the neighbouring nodes are connected by the edge. In our experiments we consider a cost C of a path as a sum of cost of all edges that belong to the path:

$$C = \sum_{i=1}^n c(v_i, v_{i+1}) \tag{1}$$

We say that path p_1 dominates path p_2 , when $C_i(p_1) \leq C_i(p_2)$ and the inequality occurs for at least one element C_i of the cost vector associated with the path. If considered path is not dominated by any other path, we call it Pareto-optimal. The task of MOSP is to find the set of Pareto-optimal paths.

One of the first works on the solution of the shortest path problem with use of genetic algorithms was [12]. Authors proposed use of a simple genetic algorithm

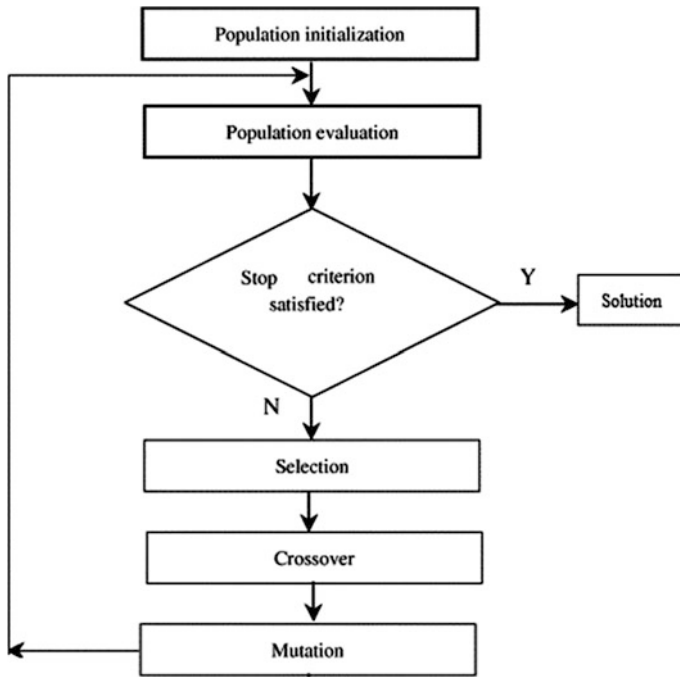


Fig. 2 Schema of the simple genetic algorithm

(SGA) to SSSP problem. A general scheme of simple genetic algorithm is given on the Fig. 2.

At the beginning of the algorithm, the population was initialized with a random set of paths. The crossover operator was a single point, and could be applied only if paths had a common vertex. The mutation operator substituted a fragment of selected path by a random one.

Kanoh [13] proposed a method specific for road networks which taken into account factors associated to traffic lights, road category and direction changes. Other authors utilized the conception of viruses [14] to inject the predefined segments into the optimized paths. Application of SPEA-II into the MOSP problem appeared in [15]. Chakraborty [16] proposed a method similar to VEGA [17], where population was divided into subpopulations and each of subpopulations was optimized with respect to single criterion.

The biggest disadvantage of mentioned algorithms was a small adjustment to the problem of path optimization. None of the known methods of the search space reduction, which are widely used in the deterministic methods, was applied into genetic algorithms. This can be treated as a possibility of the occurrence of premature convergence, because the algorithm may easier stuck in the larger search space with numerous local optima. Moreover, any of the algorithms used a

weighted sum method which actually is a kind of a single objective optimization and does not allow to find the Pareto set.

For the purpose of this paper we focused on the three algorithms: Simple Genetic Algorithm, Strength Pareto Evolutionary Algorithm and Multi-agent Genetic Algorithm. The first algorithm is often told to ‘get stuck’ in the local optimum. It has no operators to mitigate the premature convergence.

Schema of the Strength Pareto Evolutionary Algorithm [9] is similar as in SGA presented in Fig. 2. The main difference is the selection operator, which takes into account the number of individuals dominated by the current solution. What is more, the algorithm incorporates the concept of archive. Nondominated individuals from each iteration can be held in a separated set. To enter the archive, new individual cannot be dominated by any individual that is already placed there. After insertion of new solution to the archive, one have to remove from it all of the dominated solutions. Archived solutions allow to better control the results of the algorithm, because they will not be destroyed during the iterations.

Multi-agent Genetic Algorithm [10] differs significantly from the formerly described algorithms. In this algorithm individuals are organized in a grid presented on Fig. 1. Lines connect the solutions that can interact with one another. During each iteration authors propose four operations:

- competition in neighborhood—if the value fitness function of the current individual is worse than the fitness of the best individual in its neighborhood, genes of current individual are substituted by genes of the best individual
- orthogonal crossover—where each gene can be chosen from the first or the second parent
- mutation—performed as in the other algorithms
- learning—gathering good schemes from the individuals in the neighborhood.

As the algorithm presented in [10] was designed for the optimization of multi-dimensional functions, operators were adapted to the problem of MOSP. At first, we have to change the competition operator in the manner that the best individual in the neighborhood substitutes the current individual with some high probability. The crossover operator was changed to one known from the work [12], because orthogonal crossover cannot be easily adapted to the paths. Furthermore, the learning operator tries to substitute some part of the paths, if current individual has common nodes with the individuals in its neighborhood. As range of the genetic operators is limited in MAGA, it can be treated as a method of preserving the genetic diversity of the population.

2.3 Former Research on the Genetic Diversity Measures

Significant influence of the premature convergence induced researches to elaborate diversity measures for the population. First works regarding this problem appeared

in the end of the twentieth century [18, 19]. Overview of the general methods of measuring the population diversity can be found in [20, 21].

Genotypic Diversity Measures The main difference between optimization of paths in graphs represented by a sequence of vertices with respect to multi-dimensional functions is that in the first case the length of the chromosome varies depending on number of vertices in examined paths. As a result, the genotypic diversity cannot be calculated as easy as calculating the distance between points representing the solution. In the case of paths, we can use Jaccard's index defined for sets and Jaro's distance defined for strings. During the literature review, one did not find any measures designed for paths.

$$m_{Jaccard} = \frac{E_{p1} \cap E_{p2}}{E_{p1} \cup E_{p2}} \quad (2)$$

Jaccard's index (Eq. 2) examines number of common edges in two paths with respect to the total number of edges in these paths. E_p denotes a set of edges that belong to the path p . However it does not take into account the order of the edges, it can give a general indication, whether paths differ or not.

On the other hand, Jaro's [22] distance shown in Eq. 3 defines the terms of matching and substitution which can be used to examine actual sequences. Furthermore it can be computed faster than a Levenshtein distance [22].

$$m_{Jaro} = \begin{cases} 0 & \text{for } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{for } m > 0 \end{cases} \quad (3)$$

where:

- $|p|$ the number of edges in path p ,
- m number of matching elements,
- t half of a number of substitutions.

In Jaro's distance, elements are called matching if they are in the same order in both sequences, and a difference in their position from a start of a sequence is smaller than a half of the length of shorter sequence. Substitution is an operation which occurs on the elements that are equal but not placed in the correct order.

Low values of measures (Eqs. 2 and 3) reveal that the paths represented by individuals differ significantly. If the phenotypic diversity is small, the values of mentioned measures should be larger.

Phenotypic Diversity Measure An example of the phenotypic diversity measure is proposed by Morrison [20] and given by Eq. 4. Its value is actually an overall distance from the centroid of values of the fitness function.

$$m_{Morrison} = \sum_{i=1}^N \sum_{j=1}^P (c_{ij} - k_i)^2 \quad (4)$$

$$k_i = \frac{\sum_{j=1}^P c_{ij}}{P} \quad (5)$$

where:

- c_{ij} i th cost component of the j th chromosome,
- P number of chromosomes in the population,
- N length of the cost vector.

However values of phenotypic diversity measures calculated for non-injective functions can be distorted, due to the fact that different paths can have the same cost, such measures are accepted in the literature.

Low values of the Morrison's distance mean that the solutions are gathered nearby the certain point in the search space. As genetic algorithm is supposed to converge, the values of this measure should not increase during the iterations. Higher values mean that some solutions are placed far away from the best solutions.

Solutions found by the genetic algorithm can also be compared to the actual solutions obtained by the deterministic exact method by use of the recall:

$$recall = \frac{|P_g \cap P|}{|P|} \quad (6)$$

where:

- P_g set of nondominated paths found by genetic algorithm
- P set of the Pareto-optimal paths.

3 Improvement of the Genetic Algorithm

In our former research [23, 24] we proposed a method to involve the reduction of the search space in genetic algorithms solving MOSP problem. The main idea is to preprocess the input graph by calculating the hierarchical division of its edges with the Highway Hierarchies [25] algorithm. This algorithm is based on the observations of drivers' behavior. While having long distance travel, they only decide to drive through minor roads when they are not far away from the start point and when they are close to their destination. Authors of [25] proposed a method that allows artificial assignment of the hierarchy level for all edges in the graph. In this division the most important edges are placed in the top hierarchy level. Such a division

performed independently for each objective allows us to mitigate many graph edges from the search process, what affects the speed of genetic algorithm's convergence.

What is more, we decided to take into account the hierarchy levels associated with each edge while evaluating the fitness function. Let l be a number of edges that belong to path p . For each edge $e_i^p, i \in 1, \dots, l$ we can denote the vector of hierarchy levels as $h(e_i^p)$. Let h_{max} be the highest possible hierarchy level in the hierarchical division for the input graph. In the Eq. 7 we define the vector $H(p)$ as a sum of differences between maximum hierarchy level and calculated hierarchy levels for all edges and all criteria.

$$H(p) = \sum_{j=1}^l \sum_{i=1}^n h_{max} - h_i(e_j^p) \quad (7)$$

This allows us to define the fitness function as a product of $H(p)$ and $C(p)$:

$$C_m(p) = \prod_{i=1}^n H_i(p) C_i(p) \quad (8)$$

where n is the number of criteria.

While constructing the initial paths, we use a bidirectional search as in the Highway Hierarchies algorithm. After visiting a node, we write the information about the hierarchy levels of the incoming edge. Then, we do not allow to add any dominated edges to the priority queue. All possible edges are added to the priority queue with a random priority (instead of cost of traveling from the source node to the visited node). This ensures that the path will be found and it is not violate the rules of the hierarchical search.

The use of the hierarchical division while generating the initial paths enforces to handle its rules during the whole run of the algorithm. The crossover operator can only be applied for two individuals when there exist at least one common node on the paths it represents. Moreover, we check before substitution of the path fragments if the next edge after a crossing point is not dominated by the preceding edge [24].

The mutation operator works as follows: we randomly choose two nodes on the mutated path and generate new random path between them, considering only edges that are not dominated by these preceding the nodes selected for mutation. Adapted crossover and mutation operators allowed us to take into account the search space reduction in the remaining parts of the genetic algorithm.

4 Experimental Results and Discussion

In this section we present the results of our research regarding the population diversity in genetic algorithm applied for solving the MOSP problem. We performed the research on the map of the city of Lodz, Poland, which contains about 7000 nodes and 17,000 edges.

We have chosen two nodes on the map which represented the starting and the finishing point on the route that was supposed to be found. The first node was located in the suburbs and the second one was placed in the opposite side of the city. We used Matsim as a simulation environment integrated with our algorithms. The data was prepared using OpenStreetMap (spatial information) and the PostGIS plug-in (spatial functions to extract values for the different criteria).

We tested our approach for three different criteria: the length of the road, inverted maximum speed limit and the distance to the buildings. We decided to minimize all of the criteria, so the optimal path will be a short drive through the highly urbanized area but with the highest possible speed limits.

In our experiments we gathered results for three genetic algorithms: Simple Genetic Algorithm, Strength Pareto Evolutionary Algorithm and Multi-agent Genetic Algorithm. All the algorithms was run for 200 iterations. Other parameters were as follows:

- SGA—100 individuals, mutation rate: 10 %, crossover probability: 90 %, tournament selection with 2 competitors and 1 winner
- MAGA—lattice size: 10×10 , mutation probability: 5 %
- SPEA—100 individuals, mutation rate: 5 %, tournament selection with 2 competitors and 1 winner.

For all algorithms we tested its four variants:

- SR—classical version of the algorithm with no search space reduction
- CR—search space reduction included in the fitness function only
- SH—search space reduction included during the random population generation
- CH—search space reduction included in both fitness function and initialization.

As the fitness function in the SR and CR algorithm is a plain cost of the path, and in the SH and CH it is given by Eq. 8, values of the measures can be compared pairwise. In all simulations we calculated values of Morrison's measure, recall and Jaccard's index after every iteration.

4.1 Phenotypic Concentration

At first we decided to examine the concentration of individuals after 200 iterations of all algorithms. To do this we compared the average values of the Morrison's distance for all algorithms during all iterations. The results for Simple Genetic

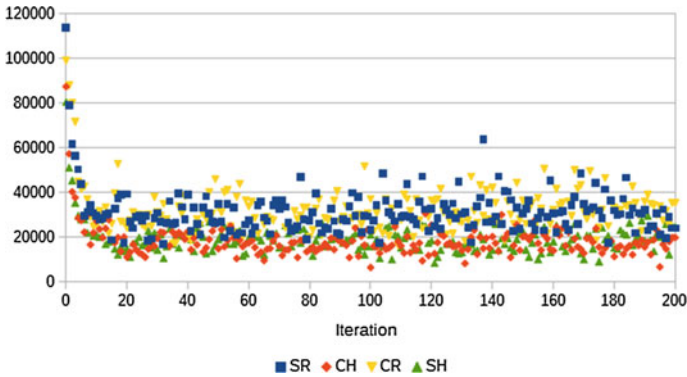


Fig. 3 Morrison's distance for simple genetic algorithm

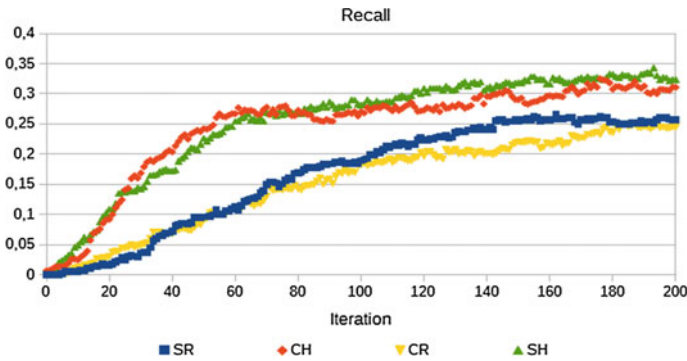


Fig. 4 Values of the recall

Algorithm are presented on Fig. 3. Presented data show the difference between the algorithm with modified operators and its classic version. It turned out, that values of the Morrison's distance are lower for the modified algorithms with the search space reduction applied in the genetic operators, which means that the individuals are gathered closer to each other. For other algorithms, we obtained similar results.

Values of the recall (Eq. 6) measure shown on Fig. 4. We observed that the variants of the modifications including the search space reduction in the operators (CH, SH) performs better than the base version (SR) and the algorithm with modified fitness function (CR). This corresponds with the values of the Morrison's distance.

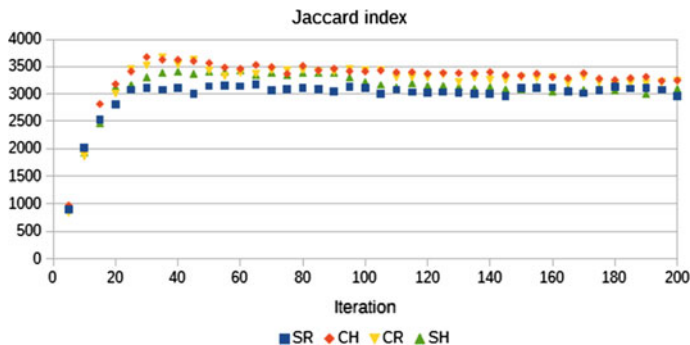


Fig. 5 Jaccard index for Strength Pareto Evolutionary Algorithm

4.2 Jaccard’s Index

Conducted research shown that for examined algorithms the values of the Jaccard’s Index are similar for all the versions of the algorithm. We observe that at the beginning of the optimization process the genotypic diversity is very high due to the low values of the Jaccard’s Index. During the iterations, one can see that the Jaccard’s Index has higher values for the variants of the algorithms with the search space reduction included in the genetic operators (CH and CR). However for the individuals in this versions the genotypic diversity is lower, these values does not mean that the algorithm stuck in the local optimum. It is caused by the better exploration of the part of the search space which is placed near the Pareto front (Fig. 5).

4.3 Values of Measures for Selected Algorithms

Conducted research shown that for SGA the modifications which include the search space reduction do not help significantly to preserve the genetic diversity of the population. Values of Morrison’s measure do not vary very much during the iterations, which is not a desirable effect, because it means that the search space is not properly explored. On the other hand, SGA is often told to converge very fast to suboptimal solution due to its simplicity and it is very hard to adapt any schema to improve its performance.

The Multi-agent Genetic Algorithm is known from its good abilities in preserving the genetic diversity. During the experiments we noticed the fastest convergence of the SH and CH variants. It was confirmed by high values of the recall measure. The genotypic diversity was preserved for all of the variants.

The last algorithm we examined was SPEA. Overall values of the recall were greater than in MAGA and SGA algorithms for the same size of the population.

5 Conclusion

Conducted research shown that proposed modifications of the fitness function and the genetic operators affects significantly the convergence of the examined genetic algorithms. The modified genetic operators allow to obtain the faster convergence and the better exploration of the Pareto front with respect to the base version of the genetic algorithm.

What is more, proposed modifications can be applied to all genetic algorithms with no modifications because they only change the fitness function and the method of performing the crossover and mutation.

Acknowledgments This research was funded by the Dean's of the Faculty of Technical Physics, Information Technology and Applied Mathematics grant supporting the skills of young scientists. Development of the application for handling the simulation data was financed from the project "Platforma Informatyczna TEWI" funded from the European Union Innovative Economy, grant no. POIG.02.03.00-00-028/09

References

1. Abraham, I., Delling, D., Goldberg, V.A., Werneck, R.: A hub-based labeling algorithm for shortest paths in road networks. *LNCS* **5038**, 319–333 (2011)
2. Bauer, R., Delling, D.: SHARC: fast and robust unidirectional routing. *J. Exp. Algorithmics (JEA)* **14**(4) (2009)
3. Ramadan, S.Z.: Reducing premature convergence problem in genetic algorithm: Application on traveling salesman problem. *Comput. Inf. Sci.* **6**(1), 47–57 (2013)
4. Rocha, M., Neves, J.: Preventing premature convergence to local optima in genetic algorithms via random offspring generation. In: *Multiple Approaches to Intelligent Systems*, pp. 127–136. Springer, Berlin (1999)
5. Nicoara, E.S.: Mechanisms to avoid the premature convergence of genetic algorithms. In: *PetroleumGas University of Ploiesti Bulletin. Mathematics–Informatics–Physics*, series 61, pp. 87–96 (2009)
6. Muhlenbein, H.: Evolution in time and space—the parallel genetic algorithm. In: *Foundations of Genetic Algorithms* (1991)
7. Kureichick, V., Melikhov, A.N., Miaghick, V.V., Savelev, O.V., Topchy, A.P.: Some new features in the genetic solution of the traveling salesman problem. In: Parmee, I., Denham, M. J. (eds.) *Adaptive Computing in Engineering Design and Control 96 (ACEDC'96)*, 2nd International Conference of the Integration of Genetic Algorithms and Neural Network Computing and Related Adaptive Computing with Current Engineering Practice, Plymouth, UK (1996)
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
9. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)
10. Zhong, W., Liu, J., Xue, M., Jiao, L.: A multiagent genetic algorithm for global numerical optimization. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **34**(2), 1128–1141 (2004)

11. Hu, J., Goodman, E., Seo, K., Fan, Z., Rosenberg, R.: The hierarchical fair competition (hfc) framework for sustainable evolutionary algorithms. *Evol. Comput.* **13**(2), 241–277 (2005)
12. Munetomo, M., Takai, Y., Sato, Y.: A migration scheme for the genetic adaptive routing algorithm. In: *IEEE International Conference on Systems, Man, and Cybernetics, 1998*, vol. 3, pp. 2774–2779 (1998)
13. Kanoh, H., Hara, K.: Hybrid genetic algorithm for dynamic multiobjective route planning with predicted traffic in a real-world road network. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 657–664. ACM (2008)
14. Kubota, N., Shimojima, K., Fukuda, T.: The role of virus infection in virus-evolutionary genetic algorithm. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 182–187 (1996)
15. Maria, J., Pangilinan, A., Janssens, G.K.: Evolutionary algorithms for the multiobjective shortest path planning problem. *Int. J. Comput. Inf. Sci. Eng.* **1**, 54–59 (2007)
16. Chakraborty, B.: Simultaneous multiobjective multiple route selection using genetic algorithm for car navigation. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) *Pattern Recognition and Machine Intelligence, LNCS 3776*, pp. 696–701. Springer, Berlin (2005)
17. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, USA*, pp. 93–100 (1985)
18. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge (1992)
19. Mori, N., Kita, H., Nishikawa, Y.: Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm. In: *Parallel Problem Solving from Nature PPSN V*, pp. 149–158. Springer, Berlin (1998)
20. Morrison, R.W., De Jong, K.A.: Measurement of population diversity. In: *Artificial Evolution*, pp. 31–41. Springer, Berlin (2002)
21. Hien, N.T., Hoai, N.X.: A brief overview of population diversity measures in genetic programming. In: *Proceedings of 3rd Asian-Pacific Workshop on Genetic Programming, Hanoi, Vietnam*, pp. 128–139 (2006)
22. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. *Kdd workshop on data cleaning and object consolidation* **3**, 73–78 (2003)
23. Chomatek, L.: Efficient algorithm for multi-criteria optimisation in the road networks. In: *Information Systems Architecture and Technology: System Analysis Approach to the Design, Control and Decision Support*, pp. 233–242 (2014)
24. Chomatek, L., Zakrzewska, D.: Search space reduction in the combinatorial multi-agent genetic algorithms. In: *Intelligent Systems' 2014*, pp. 461–472. Springer International Publishing (2015)
25. Sanders, P., Schultes, D.: Engineering highway hierarchies. In: *Algorithms ESA 2006*, pp. 804–816. Springer, Berlin (2006)