The Use of Genetic Algorithms in Mobile Applications

Plechawska-Wojcik Malgorzata
Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36 B, Lublin, Poland

Keywords: Genetic Algorithm, MOGA, Travelling Salesman Problem.

Abstract: The goal of the paper is to present the application of genetic algorithm in practice. The main result is the mechanism based on genetic algorithm applied in the application dedicated to tourists. The goal of the mechanism is to propose the most effective route between points - tourist facilities. Those objects are also chosen automatically based on user's interest as well as on his and his friends opinions expressed via social networking services Facebook. Genetic algorithm was implemented to obtain efficient way of solving the problem of matching the appropriate route regarding requirements concerning time and location. The results are obtained in short time by the genetic algorithm running on the web server. The paper presents also results of the application and mechanisms testing, including performance testing.

1 INTRODUCTION

The motivation of the paper is to adapt the multi-objective genetic algorithms to the problem of determining route in the mobile application dedicated to tourists. We developed mobile application offering tourist routes individually matched for a user who would like to visit new locations without prior preparation. Applying genetic algorithm should help to optimize results.

Genetic algorithms belong to a wide range of evolutionary algorithms. They were proposed by J. Holland (Holland, 1992) as a result of research on strict adaptation processes observed in nature (Streichert, 2001). All these algorithms search over possible solutions of the problem. They are inspired by real processes occurring in biological evolution, but they differ with use of individual operators (Dorronsoro, 2004; Goebel, et. al, 1998).

Common features of all evolutionary algorithms are: high randomness manifested during the initial population drawing and operators using, and quality improvement of individuals with successive generations reflecting the evolution occurring in nature (Fox and McMahon, 1991).

Unlike the standard analytical algorithms evolutionary algorithms do not process parameters of given problem directly, but they use their encoded form (McCarthy, 2007; Kumar, 2013; Kumar and Jyotishree, 2012). Crossover operators do not interpret the data (except heuristic operators), they modify only their representations (Guanci, et. al, 2006). However, interpretation, known also as data decoding, is required during the selection of individuals, when individual individuals must be evaluated using the objective function (El-Sharkawi, et. al., 2005; Chakraborttya, 2013).

Evolutionary algorithms might be applied to problems whose solution is impossible to achieve using analytical methods. For example, such problems are present in classes of optimization, decision or optimization problems (like NP-hard problems). Examples of evolutionary algorithms are genetic and evolutionary programming or learning and classifying systems (Yang, et. al, 2008).

2 PROBLEM STATEMENT

2.1 The Application Requirements

Application user selects initial and target location, and the time they can spend on travelling. Initial user's location might be assigned based on GPS built in mobile device. The application assigns the best route for user in given time, finding locations based on user's interest. Those interests are assigned based on Facebook portal and likes given to different locations by the user of his friends. Such locations are described with different attributes such as categories or coordinates. The identification of geographic coordinates of locations and routes

Malgorzata P..
The Use of Genetic Algorithms in Mobile Applications.
DOI: 10.5220/0004952805200525
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 520-525
ISBN: 978-989-758-027-7

between them might be defined based on Google Maps service.

2.2 Problem Formulization

The main goal of the study was to:
1. build and implement the algorithm automatically constructing the model of tourist facilities (objects) adjusted to individual user's preferences
2. construct and verify the mechanism of selection and arrangement of chosen points into the tourist route meeting the requirements of time as well as starting and finishing point.

The specificity of the mobile application project imposed required the application worked fast and efficient.

The construction of the solution (1) was performed based on activities of user's and his friends on social networking portal – Facebook.
The solution (2) was developed gradually to find the method meeting the requirement of its speed and performance. That is why the research question was "Is it possible to create mechanism of selection and arrangement of points and present user the travel map in relatively short time?". This question was verified based on efficiency testing of the constructed mechanism.

3 ALGORITHM SELECTION AND ADJUSTMENT

The propose of created algorithm is to propose user a route satisfying its requirements in terms of his interests, location of starting and destination point and the time. This task can be formed as selection of points from a given set of all available locations maximizing all the attributes imposed by the user.

3.1 Applied Algorithm

The used algorithm is multi-objective genetic algorithms (MOGA) (Shang and Zhang, 2009; Yang, et. al., 2008). The main advantage of using an evolutionary algorithm in this task is the ability to targeting the solutions in terms of all occurring dimensions due to appropriate use of the objective function and selection of parents. Therefore, the fulfilment of all the conditions determining the route targeting the solutions in terms of all occurring dimensions due to appropriate use of the objective function and selection of parents. Therefore, the fulfilment of all the conditions determining the route quality can be obtained after only one run of the algorithm.

3.2.1 Coding, Operators, Parameters

The problem domain must define the type of data representation used to encode individuals in the genetic algorithm. Natural chosen solution is an ordered list of all the sites considered to present to the user.

The algorithm implements two-point crossover which gives good results in solving the travelling salesman's problem. This choice results in necessity to use the order encoding (OE). Crossing is done with probability 80%, so it plays the major role in genes development through forcing the genes exchange between individuals. This is consistent with one of the distinguishing features of genetic algorithms from other evolutionary algorithms: the dominant role of the crossover operator during the reproduction of individuals.

The algorithms uses two mutation operators. The first one replaces two separate strings within a single chromosome. It is also responsible for abrupt changes, and thus fulfils the basic functions of mutation in the genetic algorithms, independent of the problem domain. The second mutation operator

is necessary to obtain the correct route when the target time does not allows to visit all the places. This operator negates the substring value in the second array describing the individual that stores information whether a place a given position is a part of the route or not.

Therefore, this operator can operate in two ways: first, removing excess locations for routes with limited time, and also, in later stages, of restoring locations excluded from the route. In some cases it can improve the quality of the route in terms of user interests.

Of course this is not guaranteed, but in the worst case individuals altered in this way will be eliminated over the next generations through the natural selection process. A mutation is performed with the probability of 20%, so much less than the crossover operation, but still quite significant. This probability cannot be too low, because the algorithm could not react properly to excluding locations form chromosomes.

3.2.2 Objective Function

The algorithm used in the application uses three objective functions responsible for routes development in terms of their quality, user interests, and time. Values returned by these functions are used to generate the Pareto front during determining the ranking of individuals. The values of all objective functions are converted in such a way that they can be minimized.

Before running the algorithm, ranking of categories of available location is created on the basis of information collected with Facebook. Ranking is directly proportional to the number of likes of category by the user or his friends. After that all values in the ranking are normalized. The objective function, during the route evaluation, calculates the average values of categories for locations assign to them. As the values are always in the interval (0, 1), the final step is subtracting the resulting value from one in order that the function could be minimized.Finally, the objective function responsible for estimation of route in terms of user interests has the form (1)

$$f_z = 1 - g_z \tag{1}$$

where

$$f_z = \frac{\sum_{i=1}^{n} k(x_i)}{n}$$

where $X = (x_1, x_2, \cdots, x_{1n})$ is a chromosome representing a route, n – number of locations at the route, $k(x_i)$ – category value for location *i*.

The objective function assessing the length of route should promote routes with the least differing the set time. To do it, a mapping between locations lengths and the time domain.

The conversion between lengths and time was made for medium walking speed, 5 km/h. The absolute value of the difference between given time and the route time (2).

$$f_t(x) = |t_d - t_X| \tag{2}$$

where X – a chromosome, t_d- the time set by user, t_X - the time needed for finishing the route. t_X is given with (3).

$$t_X = \sum_{i=1}^{n} t(i, i+1) + t_p * (n-1) \tag{3}$$

where t(i,i+1) is the time needed to move from particular locations of the route, n - number of locations along the route, t_p - average time spent in each of the locations.

The quality of the route needs to be controlled in terms of the locations order. Genetic algorithms randomly search the solutions space. Chromosomes will be created with good time rates and user interest, which are, however, not acceptable because of locations disorder.

The objective function responsible for assessing the route quality takes into account two factors. The first is the number of intersections between lines joining locations appearing on the route, the second – the sum of angles between successive lines.

Evaluation of these aspects reduces the probability of reproduction of individuals making suboptimal route topologies. The algorithm should be able to reject such solution or create a route, which reorders locations, or removes or adds them in order to minimize the difference between the assumed and real time.

After calculating the total values of objective functions for the entire population, individuals ranking is created based on the Pareto front. Afterwards it is used in selecting individuals to be reproduced using tournament method. The place in the ranking is determined by the number of chromosomes dominated by an individual. Individuals from all populations that make up the Pareto front are transferred to a separate file. Before each stage of reproduction chromosomes from this collection are attached to the processed population. This treatment significantly affects performance of the algorithm, because it makes simultaneous generating the population ranking and updating Pareto front of all individuals. When the algorithm is finished, routes that should be considered to be presented to the user are located in the Pareto front collection.

4 RESULTS

The characteristic feature of genetic algorithms is randomness of generated results, due to their mechanisms based on probabilistic. What is more, the task is dependent on two factors. The first is the same user and associated with it data hosted on Facebook, determining the ranking of his interests. The second factor is all the variables specified by the user before starting the search for the route, such as the place of start and destination and the total travel time and waiting time. Of course it is not possible to check all combinations of parameters that can be passed to the genetic algorithm.

The following section presents the average results obtained during the search process of ten routes, each of them with the same parameters.

4.1 Parameters of the Algorithm

Genetic algorithm was running in following configuration:

- The number of starts: 10,
- The number of chromosomes: 100,
- The maximum number of generations: 200,
- The probability of crossing: 80%,
- The probability of mutation: 20%,
- Duration of the tour: 4 hours,
- Average time spent at location: 30 minutes.

Each instance of the algorithm worked on a fifty locations, without the start and destination points.

4.2 Results

Before starting the searching process the ranking of categories which is used to determine the quality of location in terms of user interest. This action requires to use objects marked by Facebook feature "I like it", either by the user as well as related friends. All these elements are available in Facebook Graph API. This solution provides processing of significant number of tagged objects (Table 1).

Table 1: Example of category ranking.

Category name	Number of objects marked	Ranking value
Sport	14	0.3783783784
Religion	9	0.2432432432
Culture	28	0.7567567568
Health	17	0.4594594595
Gastronomy	23	0.6216216216
Entertainment	37	1

The first column contains the name of the category, the second the number of all elements in the category, and the last column - value in the ranking. Category names from the portal are mapped to the names used internally by the application.

Table 2 presents data related to the initial location, destination and all other attributes available during the route searching process. The first column contains the identifier for each location corresponding to the identification of objects in the Graph API of Facebook service. The second column contains the ranking of categories value assigned to the location allocated based on the ranking category. Ranking values are normalized before running the algorithm, they are in the interval <0, 1>. The third and fourth columns, respectively, present width and length of geographical location.

Figures 1, 2 and 3 show graphs of the best results of specific criteria for the individuals evaluation, depending on the single generation, respectively, to time of the route, interests and its quality. Charts also considers initial generation, randomly calculated. Values occurring in each dimension has been transformed in such a way that the algorithm can minimize them.

Table 2: Data set of the algorithm.

Identifier	Ranking value	Geographical width	Geographica l length
495980820459035	0.45945945	51.24462	22.558498
509027512493449	0.6216216	51.244976	22.558838
537887462921307	0.45945945	51.242558	22.55469

A lower value means a greater quality of the individual. All of these graphs present the phenomenon of convergence of genetic algorithm, achieving the point where the algorithm is not able to create better individuals for next generations.

Figures 2-4 present detailed performance results. Figure 2 is a graph of convergence of the algorithm in time of the route. Values correspond to the absolute value of the difference between the actual time of the route, which also includes the time spent in each location, and the target time specified by the

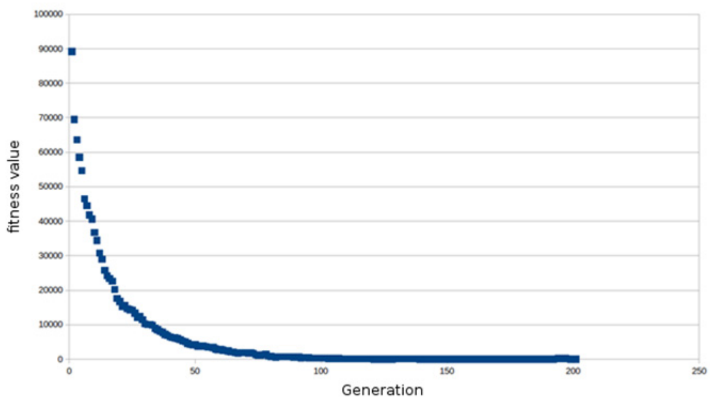

Figure 2: Best value for the criterion of the route, depending on the generation.

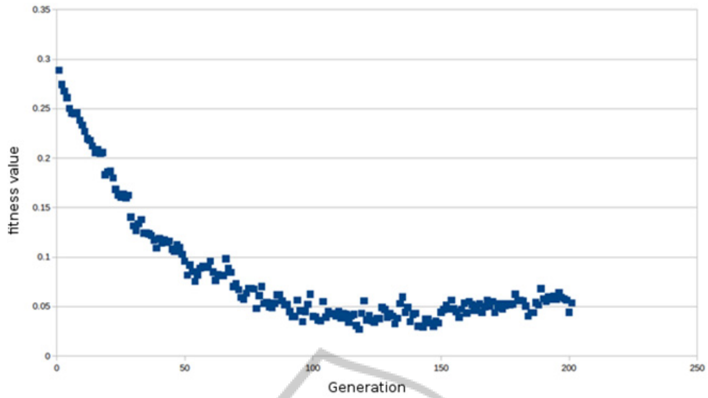

Figure 3: Best value for the criterion of interests, depending on the user.

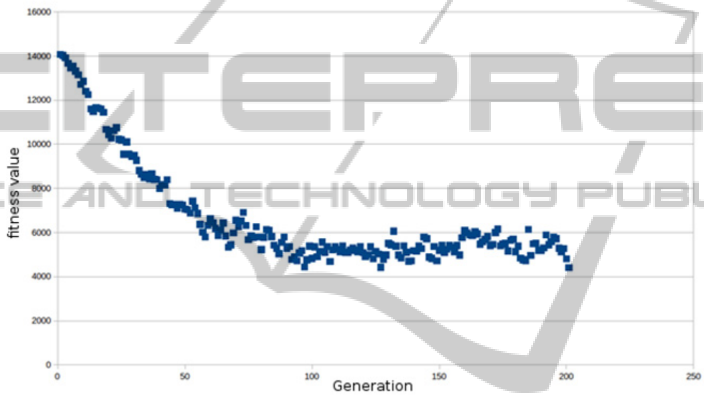

Figure 4: Best value for the criterion of the quality of the route, depending on the generation.

user. The value obtained from the best individual form the randomly created initial generation differs from the average user expectations about 90,000 seconds (26 hours). Then, generation after generation, the algorithm obtains better results until it achieves alignment between the hundredth and one hundred and fiftieth generation, where the difference between the obtained and the expected length of the route is reduced almost to zero.

Similar characteristics have graphs presenting the convergence for criterion of interests and quality of the route. In developing routes in terms of user interests values can reach values in the range (0,1), as a result of the values provided by the rating category. Values corresponding to the quality of the route have no upper limit , while the lower limit is value zero. In practice, however, there is a very small probability of taking the minimum value for the fitness function for the route quality dimension, because the route would have to be a perfect straight line from the initial to the target point, by all locations occurring on the path.

Table 3 shows the measured time of several algorithm runs. Differences are caused by diversified size of global for all generations Pareto front, which is included in each stage of the individuals selection for the reproduction.

Table 3: The timing of the genetic algorithm runs.

Instance no	1	2	3	4	5	
Time [s]	2,20	5,66	2,58	2,21	2,42	
Instance no	6	7	8	9	10	Mean
Time [s]	2,31	2,14	2,12	2,5	2,44	2,66

Increasing the number of Pareto front elements increases the computational effort, as the calculation it for n individuals requires n^2 comparisons. Resizable Pareto front for each run of the algorithm and for each generation is a result of randomness of mechanisms existing in genetic algorithms.

5 CONCLUSIONS

From the obtained results we can draw the following conclusions. Even the best individuals included in the initial generation consisting of randomly created chromosomes are very low quality. This phenomenon can be seen primarily in an example of the route criterion. What is more, the algorithm optimizes all three dimensions equally describing all individuals.

For all optimized dimensions algorithm achieves convergence between a hundredth and one hundred and fiftieth generation. In order to reduce the run-time of the algorithm one can reduce the maximum number of generations to, for example, 150 or 100. This treatment can be combined with parallel implementation of a dynamic stopping condition of the algorithm which will stop the algorithm, if the quality in all dimensions will not be improved among the last generations.

In spite of the randomness present in the genetic algorithms and the differentiation of the Pareto front size, duration of the individual instance of the algorithm for the same parameters varies slightly.

ACKNOWLEDGEMENTS

This work is supported with the scholarship under the project "Qualifications for the labour market - employer friendly university", cofinanced by European Union from European Social Fund.

REFERENCES

Dorronsoro, B., 2004. *Cellular evolutionary algorithms*. Springer-Verlag.

El-Sharkawi, M. A., Ngatchou, P., Zarei, A., 2005. Pareto multi objective optimization. *Intelligent Systems Application to Power Systems. Proceedings of the 13th International Conference*, 84 – 91.

Fox, B., McMahon, M., 1991. Genetic operators for sequencing problems, in *Foundations of Genetic Algorithms*, G. Rawlins, Ed. Morgan Kaufmann Publishers, San Mateo, CA, 284-300.

Goebel, R., Poole, D., Mackworth *A*., 1998. *Computional Intelligence: A Logical Approach.* Oxford University Press.

Guanci, Y., Quingsheng, X., Shaobo, Li., 2006. Studies on fast Pareto genetic algorithm based on fast fitness identification and external population updating scheme. *Global Design to Gain a Competitive Edge An Holistic and Collaborative Design Approach based on Computational Tools.* Ed. Xiu-Tian Yan, Benoit Eynard, William J. Ion.

Kumar, A., 2013. Encoding schemes in genetic algorithm. *International Journal of Advanced Research in IT and Engineering*, 2(3).

Kumar, R, Jyotishree, 2012, Novel Encoding Scheme in Genetic Algorithms for Better Fitness, International *Journal of Engineering and Advanced Technology* (IJEAT), 1(6).

Holland, J., 1992. *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.

McCarthy, J., 2007. *What is artificial intelligence?* http://www-formal.stanford.edu/jmc/whatisai/whatisai.html.

Shang, Y., Zhang, J., 2009. Improved multi-objective adaptive niche genetic algorithm based on Pareto front. *Advance Computing Conference*, 2009. IACC 2009. IEEE International, 300 – 304.

Streichert, F., 2001. Introduction to evolutionary algorithms. *Frankfurt MathFinance Workshop*.

Yang, Y., Li, J., Dai, W., 2008. Multi-objective genetic algorithm based on the correlation coefficient and its application. *Control and Decision Conference*, 3898 - 3902.

Chakraborttya, R., Hasinb M, 2013. Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach. International Journal of Industrial Engineering Computations, 4, 1–12.