Izabela SKOCZEŃ[1], Kamil KOCUR[2]

Opiekun naukowy: Stanisław ZAWIŚLAK[3]

EWOLUCYJNE UJĘCIA 3-KRYTERIALNEGO PROBLEMU CYKLU HAMILTONA

Streszczenie: W pracy omówiono wielokryterialny problem optymalizacji z zakresu teorii grafów. Nie ma ogólnie znanego algorytmu dla problemu cyklu Hamiltona. Zatem dla jego ujęcie z trzema kryteriami można zastosować algorytm ewolucyjny. Omówiono przygotowaną aplikację, a także zamieszczono przykładowe wyniki. Wagi w rozważanej klice ustalono losowo. Zastosowano wersję algorytmu z sukcesją w trzech podpopulacjach oraz suplementarnym przetasowaniem.

Słowa kluczowe: sukcesja turniejowa, Pareto front, klika K_n

THREE-CRITERIA HAMILTONIAN CYCLE PROBLEM – EVOLUTIONARY APPROACH

Summary: In the paper, the multi-criteria optimization problem of graph theory is discussed. There is not known the commonly used and effective algorithm for this problem (i.e. minimal Hamiltonian cycle). Therefore, especially for the problem with three criteria – the evolutionary algorithm has been applied. The prepared application is described. Exemplary results are included. The weights of edges in cliques were generated randomly. The utilized version of the algorithm consists in incorporation of succession in three subpopulation with next immediate reshuffling.

Keywords: tournament succession, Pareto front, clique K_n

1. Introduction

Graph theory [9, 10] is a branch of discrete mathematics which can be divided into a pure theoretical field and an additionary part i.e. algorithmic approach to chosen

[1] University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, izabela.s94@gmail.com
[2] University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, kamilkocur95@gmail.com
[3] Assoc. Professor [Dr hab. inż.], University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, Chair of Informatics & Automation, szawislak@ath.bielsko.pl

graph theory problems. Algorithms can solve versatile problems e.g.: checking a particular graph property (planarity, connectivity, Eulerian feature etc.) as well as solving optimization or enumeration problems. There are some classical optimization graph problems like e.g. minimal spanning tree [4, 6, 8], the shortest path problem or maximal flow through a network (i.e. weighted graph with distinguished source and destination vertices). The mentioned problems have effective classical algorithms solving them [9]. There are still some problems which do not have fully effective algorithms in a general case e.g. checking isomorphism of two arbitrary graphs or finding a Hamiltonian cycle. Moreover, multi-criteria formulations [3] of every of the listed problems are also not solved via commonly utilized algorithms. Therefore, an application of evolutionary approach [1, 2, 5] is reasonable and it was taken into account in the underneath considerations.

2. Problem formulation

In the present paper, we consider a 3-criteria problem of finding the minimal Hamiltonian cycle in clique K_n. Hamiltonian cycle is a cycle in which every vertex is visited exactly one time whereas Eulerian cycle passes through every edge exactly one time. In general, there are approx. n! cycles in a clique because every permutation of the vertices is a solution due to a fact that all possible edges belong to a clique. Some of them are the same because we start the route in the consecutive vertices. But we consider 3-criteria which means that it is even more complicated. Every edge has three weights assigned randomly in our case. These weights can be interpreted e.g. as distance in [km], time in [hours] and cost in [PLN]. These are objectives which could be considered by every management team. In case of real traveling by means of vehicles distance is connected with available roads, different times of journey could be related to utilization of highways, staying in stacks, staying before highway gates, waiting on the borders, running through the center instead of using bypass road and cost depends on fuel consumption, fees for highways.

We consider the random edge weights. Cliques are relatively small $n < 20$. Simplified evolutionary algorithm was utilized. In a clique K_n , there n vertices and $m = 05 \times n \times (n-1)$.

3. Algorithm applied in own application

The algorithm solves the problem of 3–criteria minimal Hamiltonian cycle in a particular weighted graph – for simplicity the cliques were considered. In this case every route can be represented as a permutation of vertices, 3 weights of a route are calculated as 3 sums of three different edges' weights f1, f2 and f3. Evolutionary algorithm was utilized. The consecutive populations of constant number of elements (chromosomes) are considered within a flow of the algorithm until the stop condition.

At the beginning, the initial population is randomly generated. The parameter population size was assumed as 150 members In some phases of the algorithm, this population is divided into 3 subpopulation, having 50 members, each. The consecutive epochs are performed one after another until the fulfilment of the stop condition.

Succession is performed in every epoch via dividing the population into 3 subpopulations. Succession consists in tournament selection – in one tournament two chromosomes take part. These pairs are randomly chosen by 50 times in every subpopulation. The winner of every tournament is a cycle of lower value of adequate weight, because we are looking for the min-min-min solutions i.e. all three criteria are minimized simultaneously. So in the first 50 positions we have winners in relations to f_1, in next 50 positions there winners due to second criterion (f_2), and in last 50 items – winners according to the third criterion f_3. Then the population is subjected to reshuffling to assure the different evaluation criteria in consecutive iterations.

On the current population usually some evolutionary operations are performed. In our case, we restricted these operation to mutation, only. Mutation consists in drawing randomly of two natural numbers belonging to the interval (1,n) and exchange their positions.
Exemplary mutation:

Kody krawędzi	Wynik mutacji
(1,2,3,4,**5**,6,7,**8**,9,10)	(1,2,3,4,**8**,6,7,**5**,9,10)

Mutation is performed based upon the indicator i.e. probability of mutation which is a parameter of the method and the parameter entered in the discussed application. Evolutionary Algorithm consists in repetition of the afore described steps for prescribed number of iterations – the parameter is entered by a user.

4. Program describtion

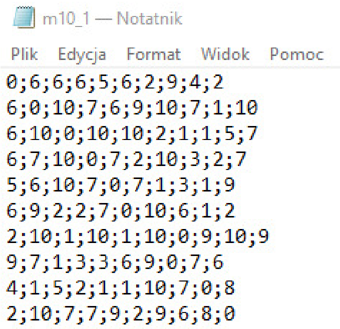
Computer program was written in algorithmic language C#, in the environment of Microsoft Visual Studio 2017.

4.1. General remarks about the application

Application can be utilized for solution of solving the 3–criteria problem of minimal Hamilton cycle. The values of the weights are read/input to the program via some text files, it allows for comparison results, starting the program several times. The result of the program is shown in a graphical sub-window where a 3D chart can be seen in an isometric view. Two ultimate populations are shown simultaneously – in two colors: red and blue. It allows for dynamic observation of the performance of consecutive populations throughout of flow of the program. Additionally, the Pareto-front is shown for the final population. The program show also the generated permutations (i.e. Hamiltonian cycles) together the sum of weights of edges belonging to the cycle. Application allows for viewing of chosen permutation on the graph presenting in the background using different colors.

4.2. Utilization of the application

After starting of the program, there is needed to input three files containing the values of weights for all edges. Graph (clique) is a simple graph therefore its adjacency matrix is symmetrical, therefore the weight matrices are also symmetrical. The format of the file is presented in *Figure 1*. The file is a square ($n\times n$) - matrix. In consequence, for $n = 10$ vertices, it is a square matrix 10x10. Particular values are divided mutually via the sign ‘;’, the values belong to the interval (1,10) and they are natural.



m10_1 — Notatnik

Plik Edycja Format Widok Pomoc

```
0;6;6;6;5;6;2;9;4;2
6;0;10;7;6;9;10;7;1;10
6;10;0;10;10;2;1;1;5;7
6;7;10;0;7;2;10;3;2;7
5;6;10;7;0;7;1;3;1;9
6;9;2;2;7;0;10;6;1;2
2;10;1;10;1;10;0;9;10;9
9;7;1;3;3;6;9;0;7;6
4;1;5;2;1;1;10;7;0;8
2;10;7;7;9;2;9;6;8;0
```

Figure 1. Input values of weights

The introductory view of the program - which can be seen – is presented in *Figure 2.*

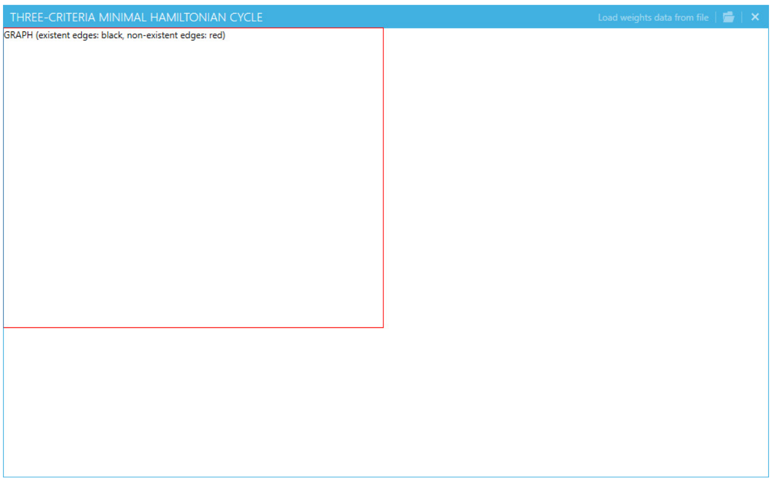


Figure 2. Introductory view of the program window

To input the introductory data to the program, a user should click on *Load weights data from file* (Figure 3).

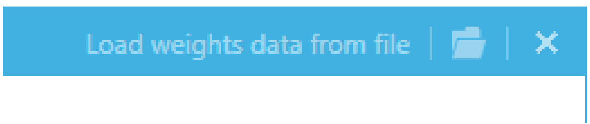


Figure 3. Button ‘Load weights data from file’.

Panel of reading (window) – is presented in *Figure 4.*

After clicking on the button *Generate*, the whole evolutionary algorithm is launched. Results are presented in 3D chart (*Figure 6*).

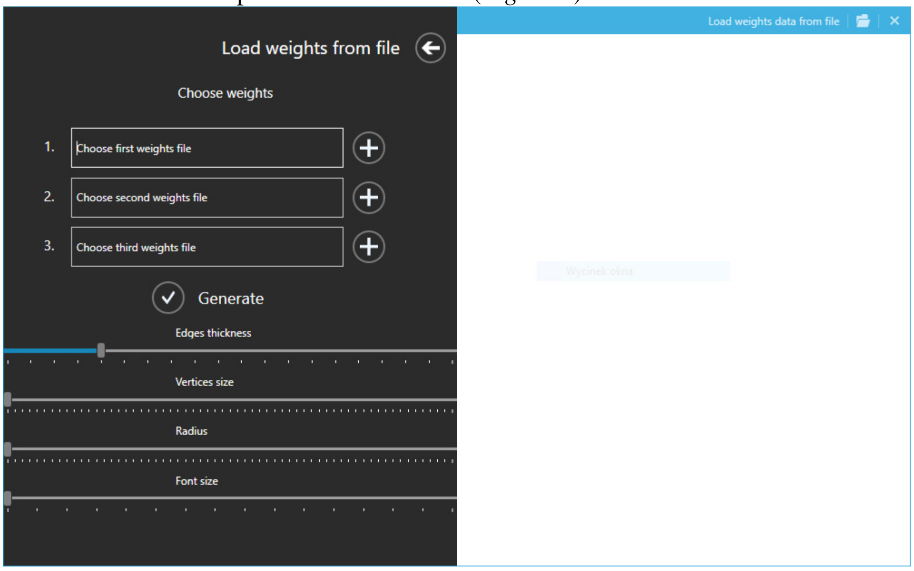


Figure 4. Panel of choosing the options, setting and input of weights

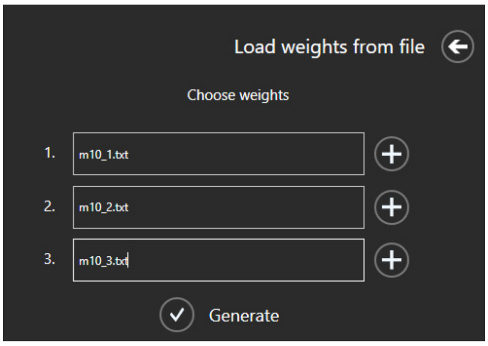


Figure 5. Files of weights which are input

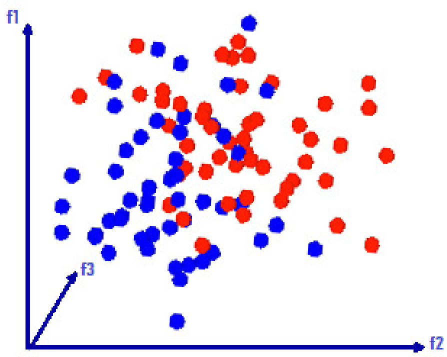


Figure 6. 3D charts of elements of final populations (blue points) and first population (red points)

The chart presents the initially generated points in the evolutionary algorithms. Program allows for tracing the work of the program – populations are shown in 3D-charts which can be seen from different perspectives via rotation of the whole chart. Additionally, the program shows every generated population one by one (switching on, switching out). Every chromosome/element of population is presented as a code (consecutive vertices) and calculated value f. The list of population is divided into three subpopulations. After choosing the particular list, there is a possibility to see the data as codes and the graph with the indicated cycle - see *Fig. 7.*

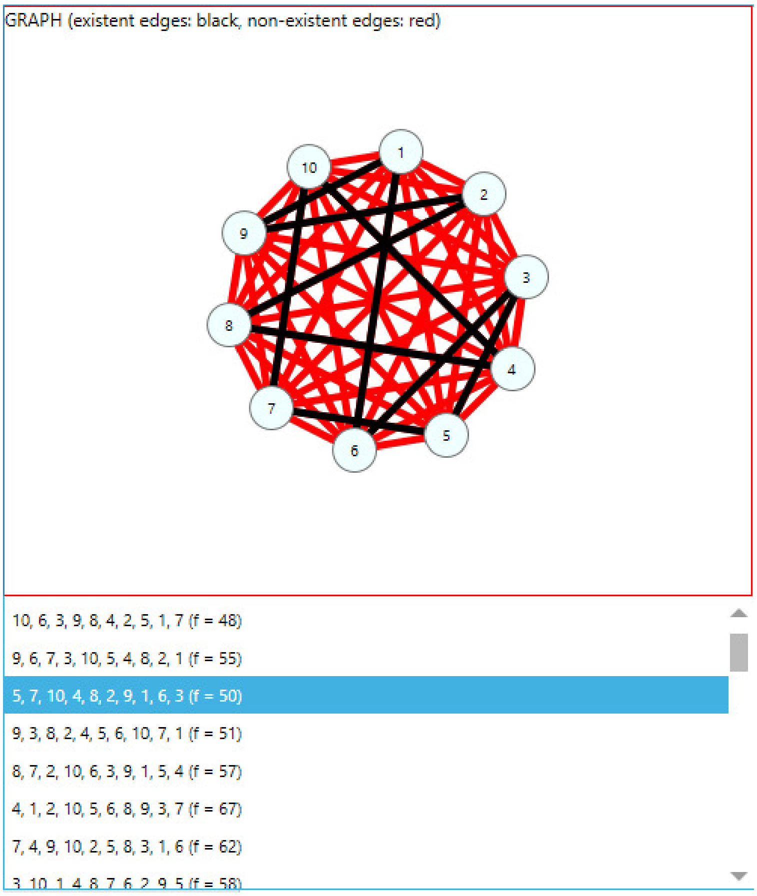


Figure 7. Generated graph and the chosen cycles – pointed on the list via highlighting

The chosen position in the list is highlighted in blue color. The edges belonging to the cycle are shown as black ones, the remaining edges are shown in red as a background. Program allows for presenting the graph with displaced vertices (*Figure 8).*

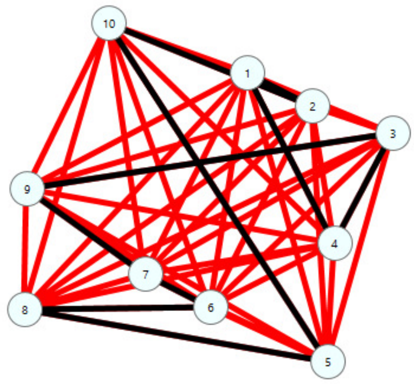


Figure 8. Graph shown in the manner of randomly distributed vertices

Other options of the program consist in possibilities of changing of chosen parameters: e.g. related to graphical presentation of the graph - like as follows: *Edges thickness, Vertices size, Radius and Font size.*

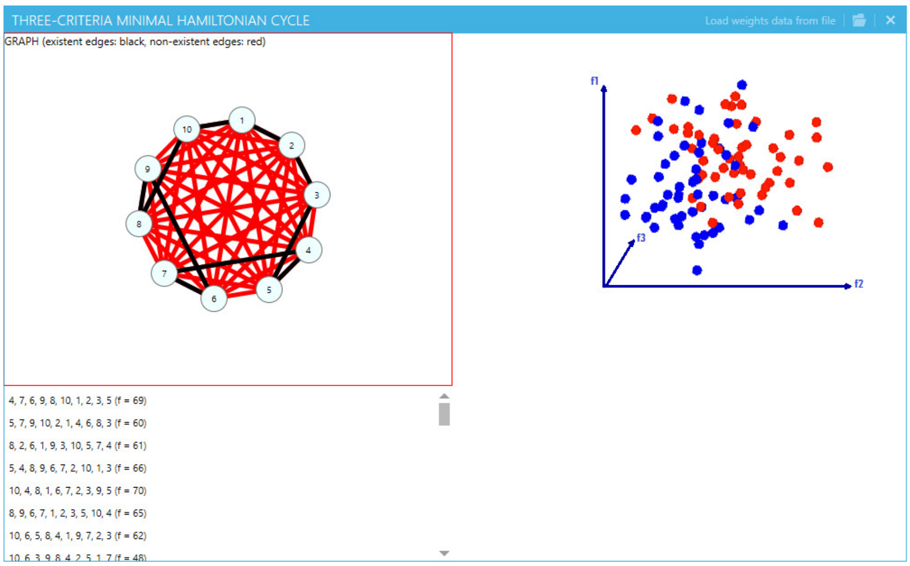


Figure 9. Window of the program after generation of population and 3D charts

The window of the program showing the generated population and the 3D-chart is presented in *Figure 9.*

5. Summary

The presented application allows for solving the 3-criteria minimal Hamiltonian cycle problem for the simple weighted graph. Simplified evolutionary algorithm was utilized. The data for launching the program and algorithm can be modified by a user. The program could be used in didactics connected with the subjects related to graph theory.

REFERENCES

1. ARABAS J.: Wykłady z algorytmów ewolucyjnych, WNT, Warszawa (2001).
2. MICHALEWICZ Z.: Genetic Algorithms + Data Structures = *Evolution* Programs. Third, Revised, Extended Edition. SPRINGER, Berlin, New York (1999).
3. EHRGOTT M., GANDIBLEUX X.: A survey and annotated bibliography of multi-objective combinatorial optimization. Or Spectrum, 22.4 (2000), 425-460.
4. FERNÁNDEZ F.R., HINOJOSA M.A., PUERTO J.: Multi-criteria minimum cost spanning tree games. European Journal of Operational Research, 158.2 (2004), 399-408.
5. OSYCZKA A.: Evolutionary algorithms for single and multi-criteria design optimization, Physica-Verlag a Springer-Verlag Company, Heidelberg (2002).
6. PAGACZ A., RAIDL G., ZAWISLAK S.: Evolutionary Approach to Constrained Minimum Spanning Tree Problem - commercial software based application. IX National Conference Evolutionary Computation and Global Optimization 2006, Editor: Jarosław Arabas, Prace Naukowe. Elektronika, z.156, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa (2006).
7. STAROSTKA Ł., KOCZUR K., JENKNER M., PATAS M.: Bi-criteria TSP: visualization program, Opiekun: S. Zawiślak. Konferencja „Inżynier XXI wieku", ATH, Bielsko-Biała, (2016), 373-382.
8. WALUS J., RUDYK.: Evolutionary approach to bi-criteria problem of minimal spanning tree in a particular graph, (Opiekun: S. Zawiślak) Konferencja „Inżynier XXI wieku", ATH, Bielsko-Biała, (2017), 369-376.
9. WOJCIECHOWSKI J., PIEŃKOSZ K.: Grafy i sieci, PWN, Warszawa, 2013.
10. WILSON R.J.: Introductory graph theory, (Polish translation: „Wprowadzenie do teorii grafów"), PWN, Warszawa (2017).