Łukasz JUROSZEK[1]

Opiekun naukowy: Stanisław ZAWIŚLAK[2]

WIZUALIZACJA EWOLUCYJNEGO UJĘCIA DWU KRYTERIALNEGO PROBLEMU DWU-PODZIAŁU GRAFU PRZY POMOCY TECHNOLOGII WEBOWYCH

Streszczenie: W pracy omówiono algorytmy oraz program komputerowy wyświetlający wybrane wyniki dla ewolucyjnego ujęcia dwu-kryterialnego problemu dwu-podziału grafu. Program jest oparty o technologie webową, co daje możliwość wyświetlania frontu-Pareto, wykresu sumy wag poszczególnej populacji oraz najlepszego osobnika w danej populacji. Graf generowany jest pseudolosowo, z prawdopodobieństwem występowania krawędzi. Sprawdzana jest spójność grafu. Zastosowano algorytm ewolucyjny, w którym osobnik reprezentuje podział grafu. Każdy z osobników posiada dwie oceny: liczbę krawędzi łączących dwa podzbiory na które dzieli się zbiór wierzchołków oraz sumę wag tychże krawędzi.

Słowa kluczowe: Algorytm ewolucyjny, graf, graf spójny, populacja, podział grafu, Pareto-front.

WEB BASED MULTI-CRITERIA CONSTRAINT GRAPH BI-PARTITIONING – EVOLUTIONARY APPROACH

Summary: In the paper, we discuss: algorithms as well as computer program which solves the problem and visualizes the chosen results related to evolutionary approach to two-criteria problem of graph bi-partitioning. The program utilizes web technologies. It allows for visualisation of Pareto-front, charts of the weights' sum for particular populations and the best chromosome of particular population. A graph is generated in a pseudo-random manner, taking into account the probability of an edge existence. Graph connectivity is checked. The evolutionary algorithm was applied in which every chromosome encodes (represents) a bi-partitioning of the graph. Each chromosome has two target functions: number of edges connecting two distinguished subsets of vertices (in which set of graph vertices is divided) and sum of weights of these edges.

[1] University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, lukjuroszek@gmail.com

[2] Assoc. Professor (Dr hab. inż.), University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, szawislak@ath.bielsko.pl

Keywords: evolutionary algorithm, graph, connected graph, population, graph partitioning, Pareto-front

1. Introduction

Genetic algorithm was invented between 1960 and 1970 at the University of Michigan by Professor John Henry Holland. Since this, the idea has been widely utilized for solving optimization or search problems [8,9]. Web-based programs for evolutionary algorithms can be resource-intensive, but if they are created for specific problem, they can be handy and effective. Presenting data in web browser is easy for basic usage, but very important to analyse data in evolutionary process. Therefore web-based approach was utilized for solving the problem given in the title.

2. Problem formulation

In the paper, we consider a problem of bi-partitioning of a particular graph [7] taking into account two criteria of optimization i.e. in both cases we are looking for minimal values of these target function. Additional functionalities of the prepared application is visualization of a graph and charts of some indicators of the implemented algorithm. Web-based applications are most familiar for an end user, moreover most of them are available on mobile phones.

Graph *G (V, E)* consists of two sets: V – set of vertices (non-empty) and E - set of edges, where $|V| = n$ and $|E| = m$ [1, 2, 3]. If E is empty than graph consists of separate vertices, only. Graph is named connected if for each pair of its vertices, the path exists – which connects these chosen vertices. We consider connected graphs. Edges of the considered graph have weights specified by user or generated randomly from the predefined range.

Pseudo-Random Number Generation (PRNG) is an algorithm for generating a sequence of the numbers by computer program. PRNG get approximately random numbers, because it is completely determined by an initial value, so it can generate similar number in some situation.

Graph is generated and stored in a matrix. This functionality will run in background. The matrix will be generated by button click based on probability and its result should be a connected graph. After generating a connected graph an evolution algorithm can be start. Program will generate default population on start. Population will have fixed number and fixed generating criteria. A population contains pop_size element i.e. chromosomes encoding a graph partitioning.

During solving the discussed problem, Pareto-front should be visible on the 2-dimentionl chart of the target functions for checking correctness of results of the evolutionary algorithm. Another chart should show sum of weights assigned to the edges connecting two distinguished subsets of V as well as the number of these edges, themselves. So, we have that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. Last important data will be showing info about best chromosome in iteration. Because, we consider a bi-criteria problem, therefore the evolutionary approach was utilized. This methodology allows for taking into account both goals within its flow. Considering of two criteria consists in a special type of selection procedure which is discussed in chapter 3. In the present work the evolutionary approach to the problem was utilized [5, 8]. Some other Artificial Intelligence methods were also used [4]. The problem itself has versatile application [6] e.g. in designing of printed boards.

Evolution algorithm should have fixed procedure to specify best way to solve problem of minimum-minimum (min-min) or maximum-maximum (max-max) for specified criteria. The graph bi-partitioning problem is a classical problem of graph theory. There are many approaches published in scientific journals and books. However, if the two criteria are considered, these algorithm have to be reformulated which was not done: The evolutionary approach does not assures the theoretical prove of effectiveness of the method. Based on general theory of evolutionary methods, we can stated that they realize a tendency to go to extreme in step by step manner. So, we neglect the precision of the solution. The advantage of the method is that it allows for dealing with many criteria simultaneously.

3. Algorithm

In the prepared allocation, the algorithm was utilized which is simple and effective and based on probability approach i.e. modified evolutionary algorithm. Modifications consists in interpretation of a chromosome and definition of evolutionary operators which relate to graph theory. The advantages of this approach are: greater independence and flexibility as well as defining elements of the algorithm according to the considered task, so finally we obtain tailored version which is precisely adequate to the problem.

We checked a connectivity of the generated graph by means of DFS-procedure. Depth-first search (DFS) is a recursive algorithm that uses the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, in other cases by backtracking. Backtrack means that when you are moving forward and there are no more nodes along the current edge, you move backwards on the same edge to find nodes to traverse. All the nodes will be visited on the current path till all the unvisited nodes have been traversed and after that the next edge will be selected. The result of this review is a sequence of nodes e.g. 0, 1, 3, 4, 2... If a sequence doesn't contain all the nodes of a graph it means that a graph is not connected.

Evolution algorithm is based on a concept of a population with contains individual (chromosome). Evolutionary algorithms make use of concepts in biology such as selection, reproduction and mutation they also depend on a random sampling. An individual (chromosome) represent a fragmentation of set V to two subsets vertices V_1, V_2 . For example: series of $(1,0,0,1,\dots,1)$, where length of series will be equal to $n = |V|$; 0 - vertex belongs to first subset, 1 - vertex belongs to second subset. Population size pop_size is a count of individuals, it is established by a program user. Algorithm start with default P_0 population. Evaluation of single individual C is based on two functions; $f_1(C)$ - sum of edges that connect two components $\mathrm{V}_1, \mathrm{V}_2$ form set V, $f_2(C)$ -sum of weigths $f_2(e)$ for all edges, where e connect two components $\mathrm{V}_1, \mathrm{V}_2$.

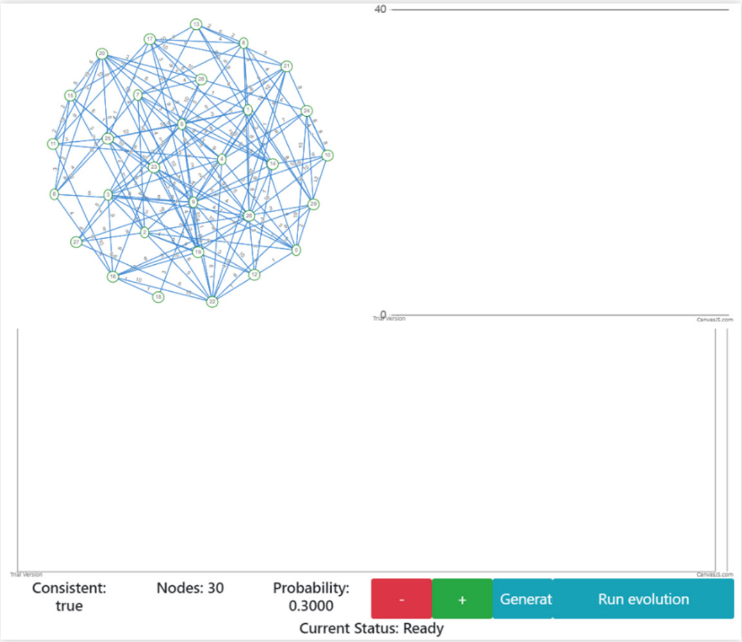
Procedure generates population as a collection of a codes representing graph bi-partitioning. Code is identifying a division with include limitation of total vertices difference between the subsets, so we consider a constrained problem. The code is not only a storage space but it is equipped in routine checking its correctness in the light of entered constraint. Each iteration starts with mutation after that the tournament selection occurs. Selection runs in two subpopulations based on $f_1(C)\ and\ f_2(C)$ respectively. So, it allows for taking into account two target functions in an alternating manner. At the end the shuffling is performed for avoidance of splitting of a population.

4. Program description

Program was written in TypeScript (Strongly Typed JavaScript), HTML. Graph is visualized by using "visjs". Other chars are created by using "ChartJs" library.

4.1. Application specification

An Application may be used for the visualization of a graph and consecutive steps of the evolutionary algorithm. The important information is visible for user; such as: connected graph, number of vertices, probability used to generate graph i.e. probability of existence of a particular edge.



Consistent: true

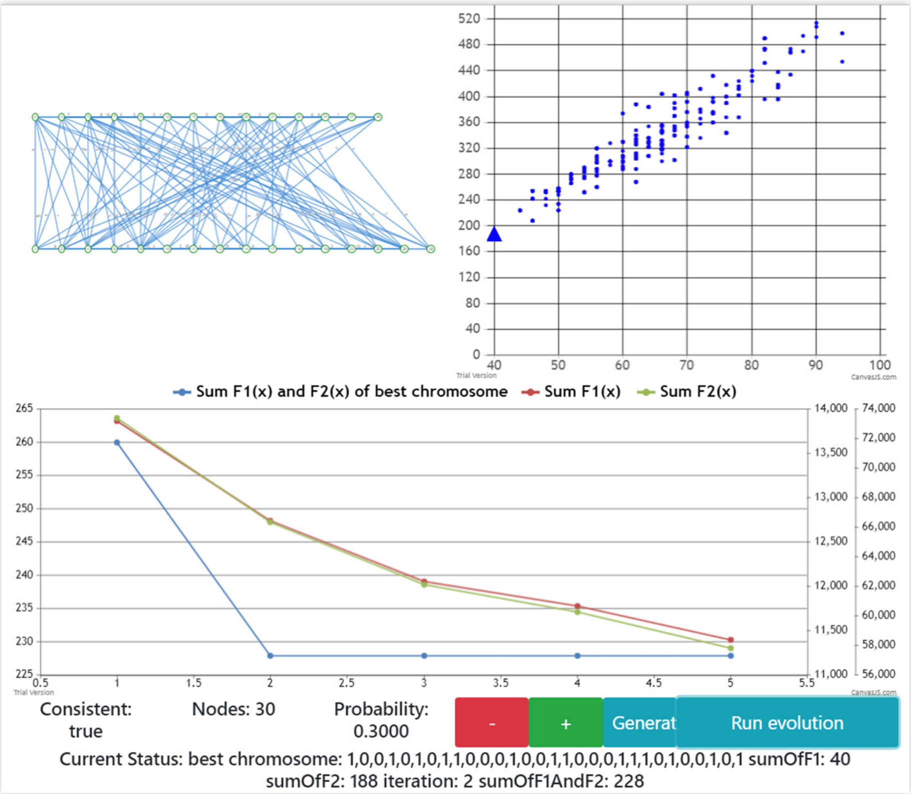
Nodes: 30

Probability: 0.3000

Current Status: Ready

Figure 1. Start screen of application

Other important functionality of the program is distinguishing and visualisation of Pareto-front chart. Pareto-front chart represents the so called non-dominated solutions related to the considered bi-partitioning problem. It a special discrete subset of all possible results which are also discrete (see Fig. 2). Second chart shows some indicators of the population in specific iteration. It could be e.g. sum of all values target functions for a whole members of a particular population or the adequate values fo best chromosome etc.



Consistent: true

Nodes: 30

Probability: 0.3000

Current Status: best chromosome: 1,0,0,1,0,1,0,1,1,0,0,0,1,0,0,1,1,0,0,0,1,1,1,0,1,0,0,1,0,1 sumOfF1: 40 sumOfF2: 188 iteration: 2 sumOfF1AndF2: 228

Figure 2. Running first 5 iterations of evolution process

An interface of the prepared application is simple to use. It don't show all options that are used in evolution algorithm, because they are specific for two-criteria problem. Graph once generated is used in the whole process. User can generate new graph and interface will clear all previous chars, so the new realisation does not contain old data. The chart changes are shown when evolution process is started. The best individual from population in all iterations is shown. Graph chart forms two parallel lines. It was done due to the idea that the connection inside the components are not important for us. Therefore, only the edges connecting the components are visible. Edges have also information about their weights. Maximum weight is permanently set to 10, that improves visibility and data readability.

After 100 iterations data from Pareto-front will be deleted. Such procedure will make application running faster in later iteration.

4.2. How to use the application

Application is running in web browser; all screen size and mobile screen resolution are supported by using a responsive component. UI (aka. User Interface) is minimalistic, only buttons that are needed to run Generic process.

First main screen shows an initialized application (Fig. 1). The initialization process contains generations of a connected graph, with default probability. User can run evolution process immediate after start. In the left-upper corner we see a representation of the generated Graph.

The edges have a text (placed nearby) which is a value of weight in range from 1 to 10. In program we determine that weight of value 0 means that edge of graph don't exists. On right-upper corner is an empty space, this is reserved for Pareto-Front chart. Under presented parts of application is another reserved space for chart with values of current iterations, more information is in later part.

On Figure we can see change after clicking in "Run evolution" button. User starts evolution process, and default number of iterations is preformed, in this exemplary running we see that 5 iteration were completed. The graph chart is presented in other way. We can see closer look on Fig. 3.

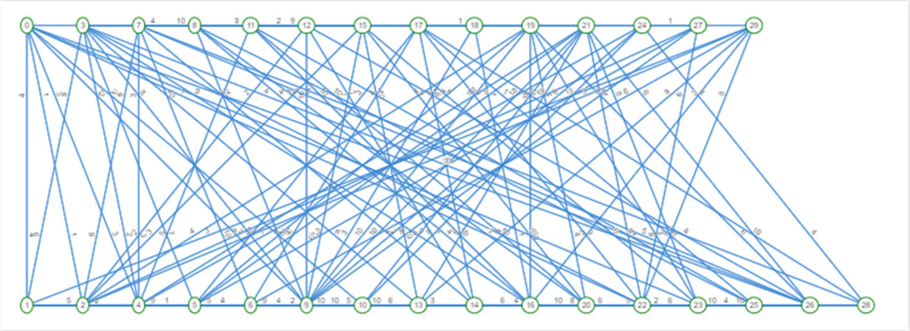
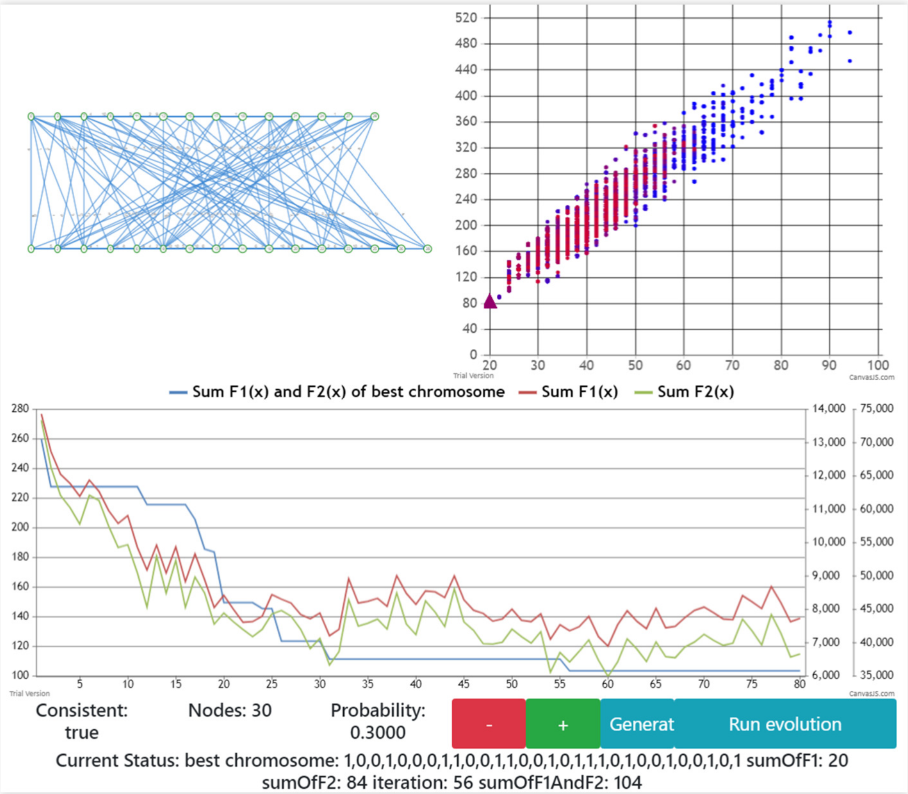


Figure 3. Graph shows best chromosome

Figure shows closer look on the considered graph. In this form best chromosome is presented. Nodes are arranged in two parallel lines; this represents division of graph to two parts i.e. set of graph vertices is divided into two subsets. The vertices belonging to these subsets are placed on the two parallel lines – for simplicity and clear view. Edges between those lines are marked, the edges between the vertices belonging to each component are one on another. So, in consequence they form one line (a section stretching between the ultimate vertices). As we can see the difference in number of nodes is equal to 2, but maximum allowed difference is 6.

Running the program for different parameters, a user can observe graphically their importance.



Consistent: true

Nodes: 30

Probability: 0.3000

Current Status: best chromosome: 1,0,0,1,0,0,0,1,1,0,0,1,1,0,0,1,0,1,1,1,0,1,0,0,1,0,0,1,0,1 sumOfF1: 20 sumOfF2: 84 iteration: 56 sumOfF1AndF2: 104

Figure 4. 80 iteration of evolution process

In Fig. 4, we can see result after preformed 80 iterations (see the horizontal axis in the lower chart). One can see that between star and approximately 30-ties generation all indicators are going down which is consistent with our task, we are looking for minimum. Later, we can observe same variations of indicators which means that there is same period of stagnation.

On the bottom strip of the program panel, there is a “Current Status” field. It tells user with chromosome is best until now. In this situation we have string of 0 and 1 i.e. encoded bi-partitioning of the graph. “Sum of F1” is $f_1(C) - sum\ of\ edges\ that\ connect\ two\ components\ V_1, V_2\ form\ set\ V$, and by an analogy the SumOfF2 is $f_2(C) - sum\ of\ weigths\ f_2(e)\ for\ all\ edges,\ where\ e\ connect\ two\ components\ V_1, V_2\ form\ set\ V$. Other information tells in with iteration was generated best chromosome and sum of $f_1(C)\ and\ f_2(C)$.

There is also a legend – informing about the meaning of colours; e.g. red for sum of F1(x) and green for sum of F2(x). It allows for clear recognition of variability of the considered indicators. The assessment is related to the whole population, despite these variable way of changes, the Pareto-front could be better and better in the consecutive iterations.

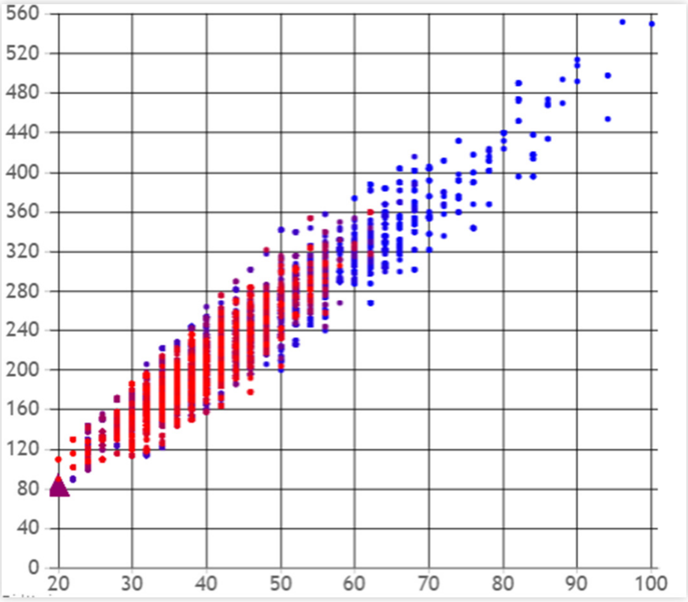


Figure 5.. Pareto-Front chart

Fig. 5 shows Pareto-Front chart. The scales on axes are very important, because we are solving min-min problem, we are looking on minimal values by those axes. The points on axis represent chromosomes in population, X value represents $f_1(C)$ and Y value represents $f_2(C)$, respectively. The colour of point indicates in with iterations was generated. The triangle shape shows best chromosome in current sate in all iterations. The X-axis is in range from 20 to 100 and Y-axis is from 0 to 560, that means that best chromosome is heavily connected with Y-axis value. Pareto-Front can be analysed by user, and user can decide that evolution process is going in right direction. The charts confirms a tendency that in consecutive iterations we obtain better and better solutions, tending down along both axes.

All charts in application have possibility to zoom and pan view. This process is shown by Pareto-Front chart example on Figure. The X and Y axis value are lower and user can see more points.

The last chart in view (Figure) show most important state of application. The "Sum F1(x) and F2(x) shows current best chromosome in iteration its value should be descending in next iterations. The "Sum F1(x)" and "Sum F2(x)" represents a sum of $f_1(C)$ and $f_2(C)$ in populations. This value can oscillate but in longer run will go lower and lower.

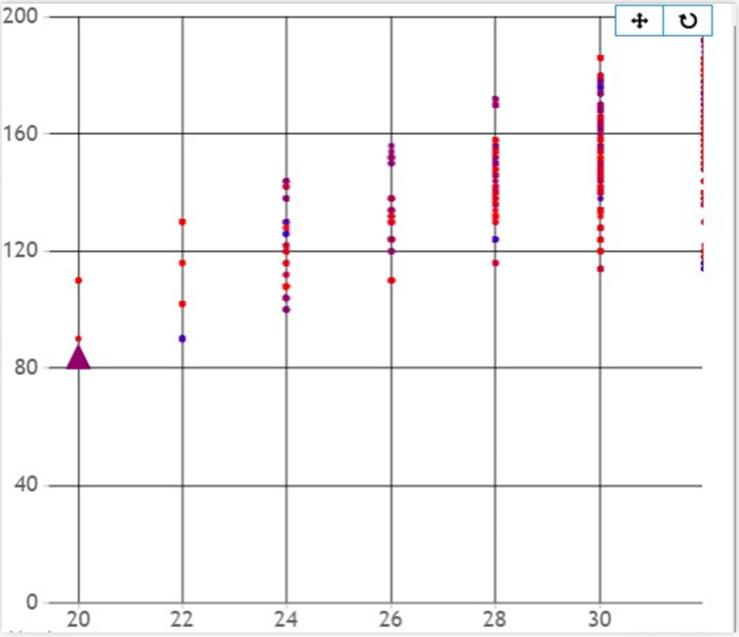


Figure 6. Pareto-Front Chart zoomed version

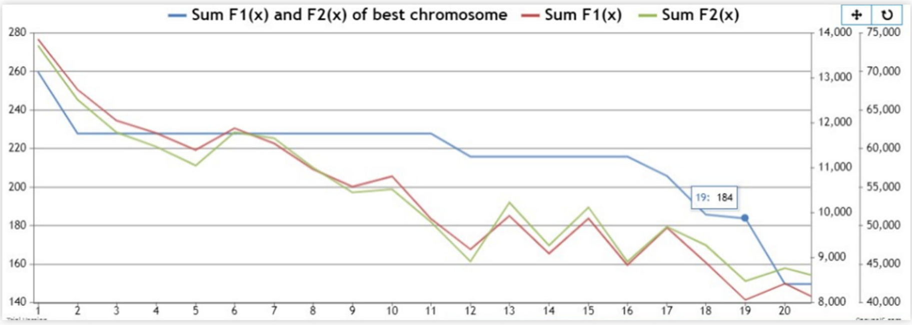
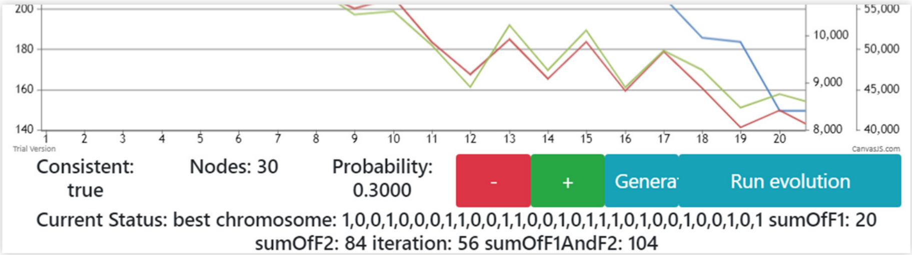


Figure 7. Chart with sums of F, F2 and sum of F1 and F2 of best chromosome

There is possibility to generate new graph by clicking “Generate” button (Fig. 7). Other buttons controls probability of occurrence of edge between nodes. The node number is constant value, that because the limitations must be selected to specific case. This assures that there is enough space for showing all graph vertices in the visual panel as well as there is enough distance (space) between consecutive vertices.



Consistent: true Nodes: 30 Probability: 0.3000

Current Status: best chromosome: 1,0,0,1,0,0,0,1,1,0,0,1,1,0,0,1,0,1,1,1,0,1,0,0,1,0,0,1,0,1 sumOfF1: 20 sumOfF2: 84 iteration: 56 sumOfF1AndF2: 104

Figure 8. Control buttons

5. Summary

The discussed application, thanks to web-based technology, allows visualize problem of finding best division by specific limitations. The performance depends on numbers of nodes and limitations. In edge case can take quite long. But effect will be human-friendly. The program can be used in didactics of subjects related to graph theory.

REFERENCES

1. BEINEKE, L. W., WILSON R. J., CAMERON P. J. (Eds.): *Topics in algebraic graph theory*. Vol. 102. Cambridge University Press, 2004.
2. WILSON, R. J. *Introduction to graph theory*. Pearson Education India, 1979.
3. GROSS, J. L.; YELLEN, J. (Eds.): *Handbook of graph theory*. CRC Press, USA, 2004.
4. MERZ, P., FREISLEBEN, B.: Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In: *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, (1998) 765-774.
5. INAYOSHI, H., MANDERICK, B.: The weighted graph bi-partitioning problem: A look at GA performance. In: *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 1994. p. 617-625.
6. YANG, Jae-Seok, et al. A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography. In: *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*. IEEE Press, (2010) 637-644.
7. KIM, Jin, et al. Genetic approaches for graph partitioning: a survey. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, (2011) 473-480.
8. BÄCK, T., FOGEL, D. B., MICHALEWICZ, Z. (Eds.): *Evolutionary computation 1: Basic algorithms and operators*. CRC Press, USA, (2018).
9. DASGUPTA, D., MICHALEWICZ, Z. (Eds.): *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, (2013).