THE TEACHING OF MATHEMATICS
2012, Vol. XV, 2, pp. 121–132

MAXIMALLY BALANCED CONNECTED PARTITION PROBLEM IN GRAPHS: APPLICATION IN EDUCATION

Dragan Matić and Milan Božić

Abstract. This paper presents the maximally balanced connected partition (MBCP) problem in graphs. MBCP is to partition a weighted connected graph into the two connected subgraphs with minimal misbalance, i.e., the sums of vertex weights in two subgraphs are as much equal as possible. The MBCP has many applications both in science and practice, including education. As an illustration of the application of MBCP, a concrete example of organizing the course Selected Topics of Number Theory is analyzed and one balanced partition is suggested. Several algorithms for solving this NP hard problem are also studied.

MathEduc Subject Classification: N64

AMS Subject Classification: 97N60

Key words and phrases: Graph partitioning; computers in education; course organization.

1. Introduction

Many real-life problems can be resolved by using mathematical structures and their properties. To create a mathematical model, each element of the problem must be represented as some mathematical structure and the relations between the elements induce appropriate relations between corresponding mathematical structures. For solving these problems, the researchers are striving to use effective mathematical formulations and structures for which theoretical results and practical techniques have been developed.

For many problems of combinatorial nature, graphs are often used and various techniques introduced in graph theory are applied in order to solve these problems.

Dealing with complex and large scale graphs, partitioning the graph into smaller subgraphs is an often used approach. If the problem is represented as a connected graph, the partitioning naturally requires that the subgraphs (partitions) are also connected. In situations when weighted connected graph has to be partitioned into two smaller components, the logical request is to ask that both subgraphs are connected and that the sum of vertex weights in both component are as similar as possible.

The problem described in the previous paragraph is called Maximally Balanced Connected Partition Problem (MBCP). The problem is proved to be NP hard [1], so there is no algorithm which solves it in polynomial time. This fact motivates the researchers to develop various heuristic methods for solving the MBCP. In this paper, mathematical formulation and application of this problem in education is presented. Also, some methods for solving the problem are explained.

2. Mathematical formulation

Let $G = (V, E)$, $V = \{1, 2, \ldots, n\}$ be a connected graph, $|E| = m$ and let w_i be weights on vertices. For any subset $V' \subset V$ the value $w(V')$ is denoted as a sum of weights of all vertices from V', i.e., $w(V') = \sum_{i \in V'} w_i$. The MBCP is to divide the set of vertices V into two nonempty disjoint sets V_1 and V_2, such that subgraphs of G induced by V_1 and V_2 are connected and the sums of the weights of vertices from V_1 and V_2 are almost uniform, ie. the difference between these two sums is as small as possible.

Formally, the MBCP is to find the partition (V_1, V_2), such that subgraphs of G induced by V_1 and V_2 are connected and the value $obj(V_1, V_2) = |w(V_1) - w(V_2)|$ is minimized.

Example 1. Let us consider the graph shown in Figure 1. The labels of the vertices are $1, 2, \ldots, 6$, while the weights are given in brackets, near to the labels. The total number of weights is 28. Ideally, the partition would contain two components with the equal weights (14). In that case, the component containing the vertex 1, with the weight 8 should also contain the vertex 4 (with the weight 6), or vertices 3 and 5. However, in both cases, the components $\{1, 4\}$ and $\{1, 3, 5\}$ are not connected. So, the optimal solution can not be equal to zero, but at least 2, because the total sum of all weights is even. The optimal solution is $(V_1, V_2) = (\{1, 2, 3\}, \{4, 5, 6\})$, and the $obj(V_1, V_2) = |13 - 15| = 2$. Note that the partition $(V_1', V_2') = (\{1, 3, 6\}, \{2, 4, 5\})$, also gives $obj(V_1', V_2') = |13 - 15| = 2$, but the graph induced by V_2' is not connected, so this partition is not considered as solution.

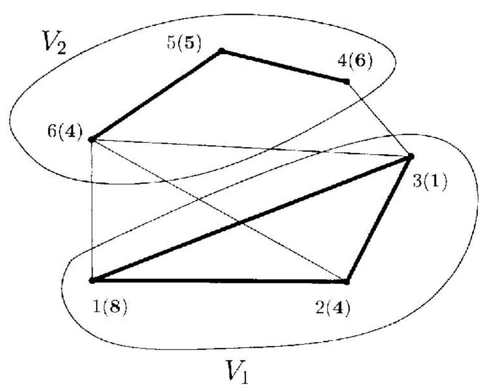

Fig. 1. The simple graph and the solution of the MBCP

The MBCP can be generalized if the number of sets in the partition is greater than 2. This generalized problem is called Balanced Connected Partition of graphs for q partitions (BCPq), where $q \geq 2$. In this problem, connected vertex weighted graph has to be partitioned into q partitions $V_1, \ldots, V_q$, so that each subgraph associated to each partition is connected and the weight of the lightest one is the highest possible, i.e., the distribution of the weights among the subgraphs should be the most homogeneous possible.

MBCP and BCPq belong to a wide class of graph partitioning problems which have a lot of direct and closely related applications, both in science and practice. In engineering field, MBCP and BCPq are applied in digital signal processing [2] and image processing [3]. Also, this problem solves some social issues like political districting [4] and organizing some public services in geographical areas [5].

3. Application in education

In education, solving MBCP is used for finding solutions of practical organizational problems, including course management issues, student grouping or organizing the researchers into connected groups.

The course material is usually divided into lessons and the appropriate difficulty is assigned to each lesson. To establish the connections between the lessons various criteria, like similarity, analogy, generality or conditionality can be used. The teacher would like to divide the course material into two disjoint connected sections, so that the sections are of similar difficulty, as much as it is possible.

Another example would be partitioning the group of students into two smaller groups. The "connectivity" between two students can be established as "the ability to work together". The objective should be to divide a student group into two smaller, having in mind that groups should be balanced by student abilities.

MBCP can be used to organize researchers engaged in a larger project. In this case, the researchers are represented as vertices and the vertex weights could be defined as "amount of time (in hours) to spend in project activities". The relation "coauthors of a scientific paper" is a natural connection between researchers and is in close relation to already mentioned "the ability to work together" property. Then, the problem is to determine the partition of the set of researchers such that total time planned for the project in each partition is as equal as possible.

Example 2. In this example, the partitioning the course Selected Topics of Number Theory into two balanced partitions is described. Whole course is divided into 32 lessons and the connections between the lessons are shown in the Figure 2.

Whole course is represented as a connected graph, where vertices correspond to lessons. Two lessons are connected only if there is a direct reason for connection in semantic or methodological sense. More precisely, the connections between two lessons are established in cases of analogy, conditionality or both lessons deal with same or similar mathematical structure or property in the field of number theory.

For example, the Fermat's little theorem is a direct consequence of Euler's theorem. Also it is in direct relation with the congruences and pseudo-prime numbers, because one important class of pseudo-prime numbers comes from Fermat's little theorem. Thus, the lesson Fermat's little theorem is connected to these three

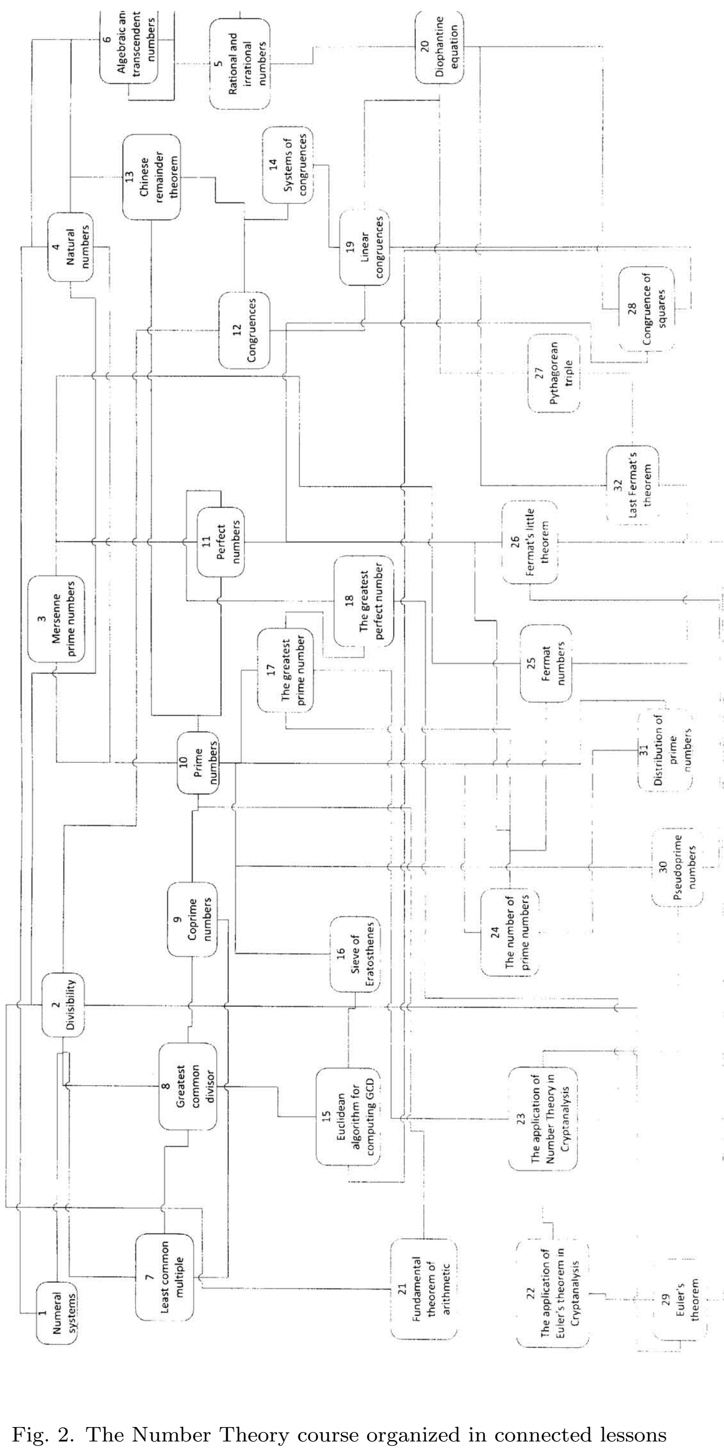

Fig. 2. The Number Theory course organized in connected lessons

lessons: Euler's theorem, congruences and pseudo-prime numbers. On the other hand, in methodological sense, the Fermat's little theorem can hardly be presented in the same context with the Fermat's last theorem, although both theorems are engendered by the same mathematician. So, the connection between these two lessons is omitted.

It should be noticed that determining the connections between lessons is a flexible task and in some way depends on the author's personal opinion and evaluation. Detailed analysis of established connections between lessons in the course is beyond the scope of this paper and can be investigated by using the references [6, 7].

The difficulty of each lesson presented in this example is based on the author's assessment and can vary, depending on professor's opinions about the required lesson's scope and level of the presentation, course objectives and student skills.

Nevertheless, several objective criteria for assessing the lesson difficulty can be established and they can follow the principles based on the influence of following factors:

- Factor I: the scope of the lesson;
- Factor II: the level of typical student skills achieved before the lesson is thought;
- Factor III: the importance of the lesson for forthcoming lessons;
- Factor IV: the importance of the lesson in mathematics, in general.

The Factor I influences to the lesson difficulty in a way that long lessons, containing a lot of definitions and theorems are generally harder than short ones. The second factor worries about the level of students pre-knowledge and their abilities to understand the content of the lesson. Last two factors indicate the level of comprehensiveness, in the sense that lessons which are pre-requisites for forthcoming lessons or general mathematical knowledge, should be studied in more comprehensive way.

To calculate the difficulty level to each lesson, the following notation is introduced. Let coefficients p_1, p_2, p_3 and p_4 be the influences of factors (I-IV) and let L be a lesson. If $f_1(L), f_2(L), f_3(L)$ and $f_4(L)$ denote the values assigned to each factor (I–IV) related to lesson L, then the difficulty level of lesson L is calculated as

$$w(L) = 10 \sum_{i=1}^{4} p_i f_i(L). \tag{1}$$

All values p_i, $i = 1, \ldots, 4$ are chosen from the interval $[0, 1]$ and the total sum

$$\sum_{i=1}^{4} p_i = 1. \tag{2}$$

For the purpose of the example, the values p_i, $i = 1, \ldots, 4$ are determined as: $p_1 = 0.45$, $p_2 = 0.25$, $p_3 = 0.15$ and $p_4 = 0.15$. Table 1 contains information about assignments of difficulty levels of each lesson of the course Selected Topics of Number Theory. First two columns are related to the lesson label (in the graph) and lesson name. The next four columns contain values related to the factors I–IV. Last column, named $w(L)$ contains the value of difficulty, calculated by the formula (1).

No.	Lesson Name	f_1	f_2	f_3	f_4	$w(L)$
1	Numeral systems	3	4	3	5	35.5
2	Divisibility	10	2	10	8	77
3	Mersenne prime numbers	3	4	2	3	31
4	Natural numbers	6	2	4	10	53
5	Rational and irrational numbers	6	2	4	8	50
6	Algebraic and trans. numbers	6	4	4	8	55
7	Least common multiple	5	1	6	10	49
8	Greatest common divisor	5	1	6	10	49
9	Coprime numbers	3	4	2	2	29.5
10	Prime numbers	10	4	10	10	85
11	Perfect numbers	5	2	3	8	44
12	Congruences	10	6	10	8	87
13	Chinese remainder theorem	6	4	2	3	44.5
14	Systems of congruences	6	6	4	4	54
15	Euclidean algorithm for comp. GCD	2	1	2	5	22
16	Sieve of Eratosthenes	2	1	2	5	22
17	The greatest prime number	2	5	2	6	33.5
18	The greatest perfect number	2	5	2	6	33.5
19	Linear congruences	6	4	4	3	47.5
20	Diophantine equation	6	4	6	6	55
21	Fundamental theorem of arithmetic	6	4	6	8	58
22	The appl. of Euler's theorem in Crypt.	6	4	4	3	47.5
23	The appl. of Number Theory in Crypt.	6	4	6	3	50.5
24	The number of prime numbers	6	4	2	3	44.5
25	Fermat numbers	3	4	2	3	31
26	Fermat's little theorem	3	4	2	6	35.5
27	Pythagorean triple	3	1	2	5	26.5
28	Congruence of squares	4	1	2	3	28
29	Euler's theorem	2	1	2	5	22
30	Pseudoprime numbers	3	4	2	2	29.5
31	Distribution of prime numbers	6	4	2	3	44.5
32	Last Fermats theorem	6	5	3	6	53

Table 1. The assignment of difficulty levels of lessons of the example course

From Table 1, it is obvious that values of all factors for each lesson are chosen from the interval $[0, 10]$ and because of (2), it can be easily shown that for each lesson L, the value $w(L)$ calculated by (1) is scaled into the interval $[0, 100]$.

The optimal solution obtained by total enumeration algorithm is 0.5, and one optimal partition is:

$V_1 = \{1, 2, 3, 4, 5, 6, 10, 11, 13, 16, 21, 22, 24, 29, 31\}$

$V_2 = \{7, 8, 9, 12, 14, 15, 17, 18, 19, 20, 23, 25, 26, 27, 28, 30, 32\}$

It can easily be calculated that $w(V_1) = 713.5$ and $w(V_2) = 714$.

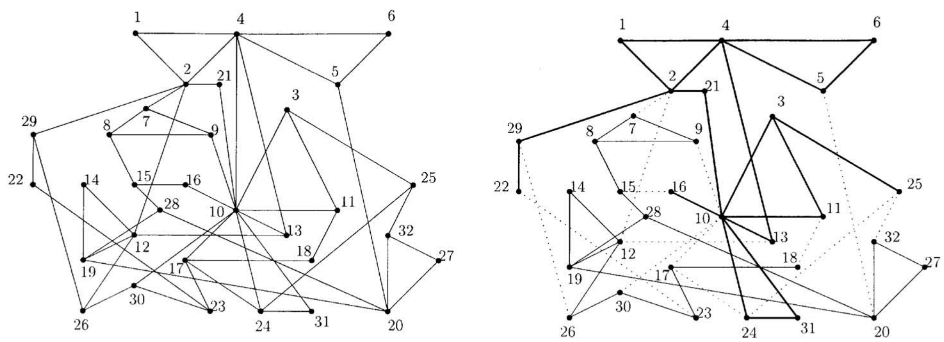

Fig. 3. Graph interpretation (left) and the solution of the example Course (right)

In Figure 3 (left), graph interpretation of the course Selected Topics of Number Theory is shown. At the right side of the Figure 1, edges belonging to the subgraph induced by V_1 are thicker, while edges belonging to the subgraph induced by V_2 are left in normal size. Edges between two subgraphs are dotted.

4. Methods for solving MBCP

The MBCP is NP hard so any deterministic algorithm can not solve the problem in polynomial time. So, exact methods for solving this problem can be only used for small scale instances because the execution time of such kind of algorithms exponentially increases with the problem dimensions. For larger instances (containing hundreds or thousands of vertices), the only way to achieve quality results is to use some heuristic methods.

This section describes one exact (total enumeration approach) and two heuristic algorithms: greedy algorithm and genetic algorithm (GA), for solving this problem.

4.1. Total enumeration algorithm

The total enumeration algorithm creates and evaluates every possible solution. This fact guarantee that optimal solution will be found when algorithm is finished. At the other hand, total enumeration algorithm is mostly unusable for real-world problems due to time and memory limitations. So this method is only used for solving small scale instances and probably, with some adoptions and restrictions, for verifying results achieved by other methods.

For the given graph $G = (V, E)$, each potential solution of the MBCP can be represented as a binary array x with the length $|V|$. The elements of the array correspond to vertices, indicating in which of two subsets of V vertices are arranged, i.e. $i \in V_1$ if $x_i = 1$ and $i \in V_2$ if $x_i = 0$. It is evident that each partition (potential solution of total enumeration algorithm) corresponds to one sequence of the length $|V|$ with the elements from $\{0, 1\}$. Since the total number of sequences of $|V|$ elements is $2^{|V|}$ the total enumeration algorithm has to analyze total of $2^{|V|}$ potential solutions. This fact proves that this algorithm can not be used for larger graphs.

The pseudo-code of the total enumeration algorithm for solving MBCP is shown in Figure 4.

1. Initialize current best solution $bs := w(V)$.
2. Construct all partitions (V_1, V_2) of set of vertices V.
3. For each partition (V_1, V_2) do
   If both subgraphs induced by V_1 and V_2 are connected
      then
         calculate the weights $w(V_1)$ and $w(V_2)$
         calculate the difference $d = |w(V_1) - w(V_2)|$
         if $d < bs$ then $bs := d$
4. Return bs.

Fig. 4. The total enumeration algorithm

At the beginning of the algorithm, current best solution bs is initialized as the maximum possible value $w(V)$. For each partition, the algorithm firstly investigates if both subgraphs are connected and if so, the difference d between the sums of weights in two components is calculated. If that difference is less than current best solution, the variable bs is assigned to that value d. The algorithm stops after all possible solutions are analyzed and the variable bs contain the optimal value.

EXAMPLE 3. Suppose that input graph is the same as in Example 1. The total enumeration algorithm has to analyze total of $2^6 = 64$ binary arrays of the length 6. Each array corresponds to one potential solution. For example, the array 100100 corresponds to the partition $\{1, 4\}$ and $\{2, 3, 5, 6\}$ and this partition induces unconnected components, so it is discarded. The optimal solution given in the Example 1 is represented as 111000.

4.2. Greedy algorithm for solving MBCP

A greedy algorithm is a heuristic method for solving the problem by making the locally optimal choice at each stage, with the hope of finding the global optimum.

Greedy algorithms do not always yield optimal solutions, but could be useful for obtaining starting feasible solutions or solutions which quality can be determined by some other techniques.

In the case of MBCP, presented greedy algorithm is based on the block-balance algorithm from [1].

Let $G = (V, E)$ be the graph. The algorithm starts with the connected partition (V_1, V_2), such that V_1 contains only the single vertex with the maximal weight. During the iteration process, the algorithm looks for a vertex u of V_2, such that both $V_1 \cup \{u\}$ and $V_2 \setminus \{u\}$ are connected.

Among all these vertices, the one with the minimal weight is chosen. The algorithm finishes when adding a new vertex to V_1 can not anymore increase the balance of the partition. The pseudo-code is shown in Figure 5.

1. Sort vertices from V in such a way that $w(v_1) \geq w(v_2) \geq ... \geq w(v_n)$.
2. Set $V_1 := \{v_1\}$, $V_2 := V \setminus \{v_1\}$.
3. If $w(V_1) \geq \frac{1}{2}w(V)$ then goto Step 7 else goto Step 4.
4. Form the set $V_0 = \{u \in V_2 |\ (V_1 \cup \{u\}, V_2 \setminus \{u\})$ is a connected partition for $G\}$.
5. Choose $u \in V_0$ such that $w(u) = \min_{v \in V_0} w(v)$.
6. If $w(u) < w(V) - 2w(V_1)$
   then $V_1 := V_1 \cup \{u\},\ V_2 := V_2 \setminus \{u\}$, goto Step 3
   else goto Step 7.
7. Return (V_1, V_2).

Fig. 5. The greedy algorithm for solving MBCP

Example 4. Let us consider the graph shown in the Figure 6.

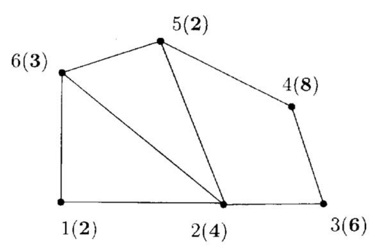

Fig. 6. The graph used for illustrating the greedy algorithm

The vertex 4 is the one with the highest weight 8. So, the starting partition is $(V_1, V_2) = (\{4\}, \{1, 2, 3, 5, 6\})$. In the next step, the set V_0 is formed, and $V_0 = \{3, 5\}$. The vertex 5 is chosen (the vertex with minimal weight 2). The condition in the step 6 of the algorithm is satisfied and the sets V_1 and V_2 are updated:

$(V_1, V_2) = (\{4,5\}, \{1,2,3,6\})$. In the next iteration of the algorithm, the set V_0 is formed again, and $V_0 = \{3,6\}$. The vertex 6 is chosen as the vertex with minimal weight (3). The condition from the step 6 of the algorithm is satisfied again: $3 < 25 - 2 \cdot (8+2)$, and the partition is updated: $(V_1, V_2) = (\{4,5,6\}, \{1,2,3\})$. In the next iteration, the set V_0 is formed $V_0 = \{1,3\}$. The vertex 1 is chosen, but the condition from the step 6 of the algorithm is not satisfied, because the right side of the inequality is negative ($2w(V_1) = 2 \cdot 13 = 26 > w(V) = 25$). The algorithm is finished and the partition $(V_1, V_2) = (\{4,5,6\}, \{1,2,3\})$ is the result. It is obvious that in this case the greedy algorithm achieves the optimal solution.

4.3. Genetic algorithms (GA) for solving MBCP

Genetic algorithms are complex and adaptive algorithms usually used in solving robust optimization problems. They involve working with population of individuals where each individual represents one (not necessary feasible) solution. In the iterative process, the population is evolving in a way that old individuals are changing to new, potentially the better ones. Each individual is assigned a value called fitness, which is used for comparing the individuals, according to the quality of the corresponding solution. During the iteration process, good individuals are selected to (re)produce better ones, while applying genetic operators crossover and mutation. The detailed description of GA is out of the scope of this paper and can be found in [8].

In this section, the GA for solving MBCP described in [9] is presented. The representation of individuals is similar as it is described in the section related to total enumeration algorithm. Each individual is represented by the vector x of $|V|$ binary genes $x_i \in \{0,1\}, i = 1, \ldots, |V|$. Each gene is corresponding to one vertex, indicating the components of V in which the vertex is arranged, i.e. $i \in V_1$ if $x_i = 1$ and $i \in V_2$ if $x_i = 0$.

Certain problems can arise from this representation, because unconnected subgraphs can appear. For overcoming these situations, the GA uses penalty function. The penalty function calculates the number of connected components for each subgraph. If $ncc_1(ind)$ and $nnc_2(ind)$ denote numbers of connected components in the first and second subgraphs of the individual ind and max_1 and max_2 denote maximal weights of vertices in the first and the second subgraphs, then the penalty function is calculated as follows:

$$Pen\_Fun(ind) = (ncc_2(ind) - 1) \cdot max_1 + (ncc_1(ind) - 1) \cdot max_2. \tag{3}$$

So, the objective value of the individual ind, corresponding to the partition (V_1, V_2) is calculated as

$$objective(ind) = |w(V_1) - w(V_2)| + Pen\_Fun(ind). \tag{4}$$

From (3) it is obvious that if both subgraphs are connected, then the penalty function is equal to 0, because $ncc_1 = ncc_2 = 1$ and in that case, the penalty function has no influence on the objective value (4).

The GA uses modified mutation operator including the concept of frozen genes, standard one-point crossover and fine-grained tournament selection with the value of $F_{tour} = 5.4$.

Example 5. Let us again consider the graph shown in the Example 1 and two individuals corresponding to encoding 100100 and 111000. As it is already mentioned in the Example 1, the first encoding corresponds to the partition $\{1,4\}$ and $\{2,3,5,6\}$ and the sets V_1 and V_2 induces graphs with 2 connected components each. The penalty function is calculated as

$$Pen\_Fun('100100') = (2-1)\cdot 8 + (2-1)\cdot 5 = 13. \tag{5}$$

As we can see, although the balance of the individual is best possible, the penalty function significantly increases the objective function because of the unconnected components. For this individual, objective value calculated by (4) has the value 13. The objective value of the individual represented by the array '111000' corresponding to optimal solution is equal to 2.

5. Conclusions

In this paper the Maximally balanced connected partition problem in graphs is presented. MBCP belong to a wide class of graph partitioning problems and has a lot of applications in science, technical fields and social issues. In this paper, an educational problem of organizing course lessons into two connected balanced components is analyzed in details. The criteria for determining the difficulty of lesson are presented and according to those criteria, partition of the course Selected Topics of Number Theory is constructed.

MBCP is an NP hard problem, so no deterministic algorithm can solve it in polynomial time. Several algorithms for solving this problem are developed and one exact and two heuristic algorithms are presented in this paper: total enumeration algorithm, greedy algorithm and genetic algorithm.

This research can be extended by applying the suggested system for determining difficulties of lessons to other mathematical courses. Also, the criteria for determining difficulties of lessons can be extended by involving additional factors influencing on lessons difficulty.

REFERENCES

[1] J. Chlebikova, *Approximating the Maximally Balanced Connected Partition Problem in graphs*, Inform. Process. Lett. **60** (1996) 225–230.

[2] I. Slama, B. Jouaber, D. Zeghlache, *Topology Control and Routing in Large Scale Wireless Sensor Networks*, Wireless Sensor Network, **2** (2010) 584–598.

[3] M. Lucertini, Y. Perl, B. Simeone, *Most uniform path partitioning and its use in image processing*, Discrete Appl. Math. **42**(2–3) (1993) 227–256.

[4] B. Bozkaya, E. Erkut, G. Laporte, *A tabu search heuristic and adaptive memory procedure for political districting*, Eur. J. Oper. Res. **144** (1) (2003) 1226.

[5] T. Assunção, V. Furtado, *A Heuristic Method for Balanced Graph Partitioning: An Application for the Demarcation of Preventive Police Patrol Areas*, in: H. Geffner, R. Prada, I. M. Alexandre, N. David (Eds.), Advances in Artifical Intelligence – Iberamia 2008, LNCS 5290/2008 Springer Berlin/Heidelberg, (2008), pp. 62–72.

[6] V. Mićić, Z. Kadelburg, D. Đukić, *Introduction to Number Theory* (fourth ed.) (in Serbian), Društvo matematičara Srbije, Beograd (1999).

[7] Hardy, G. H.; Wright, E. M., *An Introduction to the Theory of Numbers* (fifth ed.), USA: Oxford University Press,(1979)

[8] Filipović, V., *Selection and migration operators and Web services in parallel evolutionary algorithms* (in Serbian). PhD thesis, University of Belgrade, Faculty of Mathematics, (2006).

[9] B. Djurić, J. Kratica, D. Tosić, V. Filipović, *Solving the Maximally Balanced Connected Partition Problem in Graphs by Using Genetic Algorithm*, Comput. Inform. **27**(3) (2008) 341–354.

Faculty of Science and Mathematics, University of Banja Luka, Bosnia and Herzegovina

Faculty of Mathematics, University of Belgrade, Serbia

E-mail: matic.dragan@gmail.com, milan@sezampro.rs