Yugoslav Journal of Operations Research
23 (2013), Number 1, 31-41
DOI:10.2298/YJOR110704010K

AN ELECTROMAGNETISM-LIKE METHOD FOR THE MAXIMUM SET SPLITTING PROBLEM[1]

Jozef KRATICA
Mathematical Institute, Serbian Academy of Sciences and Arts,
Kneza Mihaila 36, 11 000 Belgrade, Serbia
jkratica@mi.sanu.ac.rs

Received: April 2011 / Accepted: May 2012

Abstract: In this paper, an electromagnetism-like approach (EM) for solving the maximum set splitting problem (MSSP) is applied. Hybrid approach consisting of the movement based on the attraction-repulsion mechanisms combined with the proposed scaling technique directs EM to promising search regions. Fast implementation of the local search procedure additionally improves the efficiency of overall EM system. The performance of the proposed EM approach is evaluated on two classes of instances from the literature: minimum hitting set and Steiner triple systems. The results show, except in one case, that EM reaches optimal solutions up to 500 elements and 50000 subsets on minimum hitting set instances. It also reaches all optimal/best-known solutions for Steiner triple systems.

Keywords: Electromagnetism-like metaheuristic, combinatorial optimization, maximum set splitting problem, Steiner triple systems.

MSC: 90C59, 90C27.

1. INTRODUCTION

Let S be a finite set with cardinality $m = |S|$ and let a family of subsets $S_1, ..., S_n \subseteq S$ be given. A partition of S is a disjoint pair of subsets (P_1, P_2) of S such that their union is equal to S, i.e. $P_1 \cap P_2 = \varnothing$ and $P_1 \cup P_2 = S$.

[1] This research was partially supported by Serbian Ministry of Education and Science under the grants no. 174010 and 174033.

We would like to stress that the style files and the template should not be manipulated and that the guidelines regarding font sizes and format should be adhered to. This is to ensure end product to be as homogeneous as possible.

Let us define the splitting condition: a subset $S_k \subseteq S$ is split by the partition (P_1, P_2) if and only if S_k is not disjoint with P_1 and P_2, i.e. $S_k \cap P_1 \neq \emptyset$ and $S_k \cap P_2 \neq \emptyset$. An equivalent expression of the splitting condition is the statement that there exist $a,b \in S_k$ for which holds $a \in P_1$ and $b \in P_2$.

Then, the maximum set splitting problem (MSSP) can be defined as finding the partition (P_1, P_2) that splits maximal number of given subsets S_1, ..., S_n. The MSSP, as well as weighted variant of the problem, is NP-hard in general ([11]). The variant of the problem, when all subsets in the family are of fixed size r, $r \geq 2$ is also NP-hard. Furthermore, the MSSP is APX complete, i.e. cannot be approximated in polynomial time within a factor greater than 11/12, as can be seen from [13].

Let us demonstrate some properties of MSSP on two small illustrative examples.

Example 1. Let our first set consist of four elements (m=4) and four subsets (n=4). The subsets are: S_1 = {1,3}; S_2 = {2,4}; S_3 ={1,4}; S_4 = {2,3}. One of the optimal solutions is the partition (P_1,P_2), P_1 = {1,2}; P_2 = {3,4}. The optimal objective value is equal to n=4, because $P_1 \cap S_k \neq \emptyset$ and $P_2 \cap S_k \neq \emptyset$, for all k=1,2,3,4.

Example 2. Let our second set consist of four elements (m=4) and five subsets (n=5). The subsets are: S_1 = {1,2,3}; S_2 = {1,4}; S_3 ={2,4}; S_4 = {3,4}; S_5 = {1,2,4}. One of the optimal solutions is the partition (P_1,P_2), P_1 = {1,2,3}; P_2 = {4}. The optimal objective value is 4 and all subsets are split, except the first subset.

In the following section, the existing integer programing models for MSSP and some previous work are given. Section 3 describes EM solution procedure. Experimental results on two classes of instances, and short discussion of the results obtained from the proposed EM solution procedure are presented in Section 4. The final section presents conclusions and ideas for a future work.

2. PREVIOUS WORK

Kernelization method based on a probabilistic approach is proposed in [4,5]. Running time of a subset partition technique is bounded by $O(2^q)$, where q is the number of split subsets. That algorithm can be de-randomized, which leads to a deterministic parameterized algorithm of running time $O(4^q)$ for the weighted maximum set splitting problem. This indicates that the problem is fixed-parameter tractable. The kernelization technique is consequently used in [7,8,17,18].

The first quadratic integer programming (QIP) formulation of the MSSP, given by (1)-(3), is introduced in [2]. That formulation and its semidefinite programming (SDP) relaxation were used for constructing the 0.724-approximation algorithm of the MSSP. By improving the rounding method and applying a tighter analysis in [21], the SDP was strengthened to a slightly better, 0.7499-approximation algorithm. Variables of QIP formulation are defined as:

$$y_i = \begin{cases} 1, & i \in P_1 \\ -1, & i \in P_2 \end{cases} \qquad z_k = \begin{cases} 1, & S_k \ split \\ 0, & otherwise \end{cases}$$

Then QIP model is defined as:

$$\max \sum_{k=1}^{n} z_k \tag{1}$$

subject to

$$\frac{1}{|S_k|-1} \sum_{\substack{i_1, i_2 \in S_k \\ i_1 \neq i_2}} \frac{1- y_{i_1} \cdot y_{i_2}}{2} \geq z_k, \quad k = 1,...,n \tag{2}$$

$$z_k \in \{0,1\}, \quad k = 1,...,n; \quad y_i \in \{-1,1\}, \ i = 1,...,m \tag{3}$$

In contrast to the classical branching on parts of the solution, inclusion/exclusion branching proposed in [19] is used to branch on the requirements imposed on problems. That technique was consequently used for the partial dominating set and the parameterised problem of the *k*-set splitting.

The MSSP is taken into account in the stationary set splitting game ([15]). Two players participate in this game: the *unsplit* and the *split*, where the unsplit are choosing stationarily many countable ordinals and the split are trying continuously to divide them into two stationary pieces. In [15], it is shown that it is possible to force a winning strategy either for both players, or for none of them. This gives a new insight into the second-order monadic logic of order.

The first integer linear programming (ILP) formulation of MSSP, given by (4)-(8) is introduced in [16]. In that paper, a genetic algorithm (GA) for solving MSSP is also proposed. The GA uses the binary encoding, standard genetic operators adapted to the problem and caching technique. Experimental results using CPLEX solver based on the ILP formulation and proposed GA were performed on two sets of instances from the literature: minimum hitting set and Steiner triple systems. The results show that the Steiner triple systems seem to be much more challenging for maximum set splitting problems since the CPLEX solved to optimality, within two hours, only two instances up to 15 elements and 35 subsets. Parameters and decision variables of ILP formulation are defined as:

$$s_{ik} = \begin{cases} 1, & i \in S_k \\ 0, & i \notin S_k \end{cases} \qquad y_i = \begin{cases} 1, & i \in P_1 \\ 0, & i \in P_2 \end{cases} \qquad z_k = \begin{cases} 1, & S_k \ split \\ 0, & otherwise \end{cases}$$

Then MSSP is modeled as ILP program:

$$\max \sum_{k=1}^{n} z_k \tag{4}$$

subject to

$$z_k \leq \sum_{i=1}^{m} s_{ik} \cdot y_i \ , \quad k = 1,...,n \tag{5}$$

$$z_k + \sum_{i=1}^{m} s_{ik} \cdot y_i \leq |S_k| \ , \quad k = 1,...,n \tag{6}$$

$$z_k \in \{0,1\}, \quad k = 1,...,n; \quad y_i \in \{0,1\}, \quad i = 1,...,m \tag{7}$$

3. EM IMPLEMENTATION

An electromagnetism-like (EM) metaheuristic is a powerful algorithm for global optimization that converges rapidly to the optimum ([3]). In the field of combinatorial optimization, the method is used either as a stand-alone approach or an accompanying algorithm for other methods. A detailed description of EM is not in the scope of this paper, but several recent successful applications should be mentioned:

- Global optimization ([1]);
- Response time variability ([10]);
- Flow path design of undirectional AGV systems ([12]);
- Strong minimum energy topology ([14]);
- Blind multiuser detection over the multipath fading channel ([20]).

EM is a population-based algorithm that can solve nonlinear optimization problems. In the following text, each member p_j, $j = 1, 2, \ldots, N_{pop}$ of the population maintained by the algorithm will be referred to as EM point (or solution). The population itself will be referred to as a solution set. Since each point is a real vector of the length m, whose meaning is described in detail later, the i-th coordinate of point p_j is denoted as p_{ji}. The proposed EM algorithm for solving MSSP is given by the following pseudo code:

Program 1: EM pseudo-code

```
program MSSP_EM(Output)
begin
 MSSPInput;
 Init;
 iter:=0;
 while iter < N_iter do
  begin
   iter:=iter+1;
   for j:=1 to N_pop do
    begin
      fv:=ObjFunction(p_j,y,z);
      LocalSearch(y,z,fv);
      Scaling(p_j,y);
    end;
   CalculateChargesForces;
   Moving;
  end;
 PrintResults;
end.
```

When the reading of a test instance is completed by a procedure *MSSPInput*, EM points in the first iteration are randomly initialized from set $[0,1]^m$ (procedure *Init*).

In each iteration and for each EM point, the program calculates the value of the objective function, applies the local search, and performs the scaling procedure (*ObjFunction*, *LocalSearch* and *Scaling*, respectively). Afterwards, calculation of charges and forces using EM attraction-repulsion mechanism is applied, resulting in moving the points towards a local maxima (procedures *CalculateChargesForces* and *Moving*). At the end, all obtained results are exhibited by procedure *PrintResults.*

3.1 Objective function and local search

This section gives a description of the evaluating the objective function `ObjFunction(p`$_j$`,y,z)` mentioned in Program 1. In that procedure, the objective function has only one input parameter, which is a given EM point p_j, while arrays y and z are output parameters defined in the same way as decision variables y and z in ILP formulation (4)-(7). Therefore, y_i=1 means that the element i belongs to P_1, while y_i=0 means the opposite (i belong to P_2). In the case when the subset k is split, holds z_k=1, otherwise z_k=0.

For a given EM point p_j, a partition (P_1,P_2) is established by rounding in the following way: if the i-th coordinate of the p_j is equal to, or greater than 0.5, then the element i is assigned to P_1, otherwise it is assigned to P_2. Mathematically, by using the decision variable y, it can be defined as $y_i = \begin{cases} 1, & p_{ji} \geq 0.5 \\ 0, & p_{ji} < 0.5 \end{cases}$. Values of decision variable z are obtained by checking if the subset S_k is split by the given partition (P_1,P_2), or not, while the objective value is the number of split subsets, i.e. the number of decision variables $z_{i,,}$ which has the value 1, or is equal to $\sum_{k=1}^{n} z_k$. Note that all EM points are feasible, since the problem has no forbidden partitions.

After objective function for each EM point is computed, a possible improvement is tried by local search (LS) procedure. Local search (LS) is a supplemental procedure to perform a quick exploration around a solution. The motivation behind the utilization of LS is to explore the possibility of finding a solution with a better objective function. In this work, a 1-swap local search is used and adapted to MSSP into a simple, but very effective procedure *LocalSearch* described in Algorithm 2.

The proposed local search procedure uses the first improvement strategy, which means that it is immediately applied after the detection of an improvement of the solution. After that, it is continuously applied until no more improvements in the number of split sets are observed, i.e. when for each $i = 1, \ldots, m$ local search does not produce a greater number of split sets than the current one.

Program 2: Local search pseudo-code

```
procedure LocalSearch(y,z,fv)
begin
 repeat
  impr:=false;
  i:=0;
  while not impr and (i<m) do
   begin
```

```
        i:=i+1;
        nfv:=Change(y,z,i);
        if(nfv > fv) then
         begin
          impr:=true;
          fv:=nfv;
          y[i]:=1-y[i];
         end
      end
  until not impr;
  end;
```

Function *Change*(*y*,*z*,*i*) firstly computes the number of sets S_k split by exchanging element *i* from P_1 to P_2 , if the element *i* previously belonged to P_1 (or conversely, from P_2 to P_1 if *i* previously belonged to P_2). Then, the number of sets S_k not split by exchanging the element *i* is counted. Subsequently, the new objective value *nfv* is equal to the old objective value *fv* plus the difference between the numbers of split and not-split sets produced by the exchanging the element *i*. Note that, in the function *Change*(*y*,*z*,*i*), it is enough to search only the subsets S_k that contain the element *i* ($P_k \ni i$),whose number is usually substantially smaller than the total number of all subsets *n*. Therefore, in order to speed-up the evaluation of *LocalSearch*() function, in the preprocessing part of the program (procedure *Init*), for each element *i*, an array of indices of the subsets P_k, containing element *i* is memorized. Therefore, to evaluate the function *Change*(*y*,*z*,*i*), the only thing needed is to search inside these arrays instead to search all subsets .

3.2. Scaling procedure

In this implementation, scaling procedure is applied, which additionally moves points towards solutions obtained by local search. It is considered only with some factor $\lambda \in [0,1]$ in order to prevent falling into a local optimum and being trapped there. An EM point p_j is moved by the following formula:

$$p_{ji}^{new} = \lambda \cdot y_i + (1-\lambda) \cdot p_{ji} \tag{8}$$

where p_{ji}^{new} is the new value of the *i*-th coordinate of EM-point p_j while y_i denotes a sequence *y* of the *j*-th EM point in the current iteration after the local search procedure is finished.

Choosing an appropriate value of the scale factor λ is a significant step for governing the search process. In the extreme case, when λ is close to 1, the search process will likely fall into a local optimum and be trapped. Another extreme case, when λ is equal to 0, obviously represents no-scaling situation. Experiments have showed that λ = 0.1 is a good compromise that yields satisfactory results.

3.3. Attraction-repulsion mechanism

As it can be seen from the literature, the strength of the EM algorithm lies in the idea of directing EM points towards local optima utilizing an attraction-repulsion mechanism. Therefore, after applying the local search procedure to each solution in the

current population, the solutions must be moved towards promising regions in order to get closer to the optimal solution.

In this process, each EM point is considered as a charged particle. The amount of charge relates to the value of the objective function at the point, which also determines the magnitude of attraction or repulsion of the point over the solution set. Mathematically, the charge of each sample point is calculated by the following formula:

$$q_j = \exp\left(-N_{pop} \cdot \frac{f(p^{best}) - f(p_j)}{\sum_{l=1}^{N_{pop}} f(p^{best}) - f(p_l)} \right), \quad j=1, ..., N_{pop} \tag{9}$$

The force between two points is computed using a mechanism similar to electromagnetism theory for the charged particles. In this mechanism, the force exerted on a point via other points is inversely proportional to the distance between the points and directly proportional to the product of their charges. The point that has a better objective value attracts the other points, and the point with the worse objective value repels the others. The computation of this force is given by (10). The power of attraction or repulsion of charges is calculated as follows:

$$F_j = \sum_{l=1, l \neq j}^{N_{pop}} F_j^l, \text{ where}$$

$$F_j^l = \begin{cases} \left(\dfrac{q_l q_j}{\| p_j - p_l \|^2} \right) \cdot (p_l - p_j), & f(p_l) > f(p_j) \\ \left(\dfrac{q_l q_j}{\| p_j - p_l \|^2} \right) \cdot (p_j - p_l), & f(p_l) \leq f(p_j) \end{cases} \tag{10}$$

where $\|p_l - p_j\|$ is the Euclidean distance between EM points p_l and p_j.

Using the *Move* procedure of the electromagnetism approach, current solutions are by (11) shifted towards the best ones. All the EM points are moved, except the current best solution. The vector of the total force exerted on each point from the other points, determines the direction of movement for the corresponding EM point. Therefore, the total forces are normalized ($F_j = \dfrac{F_j}{\|F_j\|}$), which also implies that infeasible solutions cannot be produced. The movement of each EM point (except the best EM solution) is calculated by (11), using a random step length β generated from uniform distribution from the set [0,1]. This step length is used, since, as can be seen in [3], the candidate solutions have a nonzero probability to move to the unvisited solution in this direction when random step length is selected.

$$p_{ji} = \begin{cases} p_{ji} + \beta \cdot F_{ji} \cdot (1 - p_{ji}), & F_{ji} \geq 0 \\ p_{ji} + \beta \cdot F_{ji} \cdot p_{ji}, & F_{ji} < 0 \end{cases} \tag{11}$$

4. COMPUTATIONAL RESULTS

The tests are performed on a single processor Intel 2.5 GHz with 1GB memory, under Windows XP operating system. The algorithm is coded in C programming language and tested on two classes of instances from literature: minimum hitting set (MHS) instances introduced in [6] and Steiner triple systems (STS) described in [9]. For MHS instances, all optimal solutions are known and are equal to *n*. All optimal solutions are reported in [16]; they are obtained by CPLEX solver, except the largest MHS instance, when CPLEX stopped its work with "out of memory" status. In that situation, with *m*=500, *n*=50000, GA in [16] obtained solution, with all split subsets (objective value is equal to *n*=50000), which verified the optimality of that solution. In the case of the STS instances, optimal solutions are known only for the first two instances (also obtained by CPLEX solver in [16]), and they are strictly smaller than *n*.

The parameters of EM are: λ = 0.1, N_{iter}=20 and N_{pop}=5. The EM ran 20 times for each instance, and the results are summarized in Table 1 and Table 2. The tables are organized as follows:

- the first and the second column contain *m* and *n*;
- the third column contains the optimal solution if it is known in advance. If an optimal solution is not known, next column displays best-known solution up to date;
- next three columns present the EM best solution (EM_{best}), running time in seconds needed to reach that solution (t) and the average total running time (t_{tot}), respectively;
- the last two columns (*agap* and σ) contain information on the average solution quality: agap is a percentage gap defined as $agap = \frac{1}{20}\sum_{r=1}^{20} gap_r$, where $gap_r = 100 \cdot \frac{opt - EM_r}{opt}$ in cases when an optimal solution is known or $gap_r = 100 \cdot \frac{best - EM_r}{best}$ in other cases. EM_r represents the EM solution obtained in the *r*-th run, while σ is the standard deviation of gap_r, r=1,2,...,20, obtained by formula $\sigma = \sqrt{\frac{1}{20}\sum_{r=1}^{20} gap_r - agap}^{2}$.

Table 1: EM results on MHS instances

m	*n*	Opt	EM_{best}	*t* (sec)	t_{tot} (sec)	*agap* (%)	σ (%)
50	1000	1000	opt	0.014	0.158	0.000	0.000
50	10000	10000	opt	0.333	3.212	0.000	0.000
100	1000	1000	opt	0.024	0.334	0.000	0.000
100	10000	10000	opt	0.665	10.593	0.000	0.000
100	50000	50000	49998	81.305	216.316	0.008	0.002
250	1000	1000	opt	0.068	1.062	0.000	0.000
250	10000	10000	opt	2.454	45.393	0.000	0.000
500	1000	1000	opt	0.150	2.336	0.000	0.000
500	10000	10000	opt	4.841	94.473	0.000	0.000
500	50000	50000	opt	26.984	486.124	0.000	0.000

Table 2: EM results on STS instances

m	*n*	*Opt*	*Best*	EM_{best}	*t* (sec)	t_{tot} (sec)	*agap* (%)	σ (%)
9	12	10	10	opt	0.001	0.001	0.000	0.000
15	35	28	28	opt	0.001	0.003	0.000	0.000
27	117	-	91	best	0.001	0.005	0.000	0.000
45	330	-	253	best	0.010	0.030	0.000	0.000
81	1080	-	820	best	0.054	0.173	0.000	0.000
135	3015	-	2278	best	0.384	0.905	0.000	0.000
243	9801	-	7381	best	8.066	14.953	0.000	0.000

As it can be seen from Tables 1 and 2, EM reaches all optimal/best-known solutions, except one MHS instance (*m*=100, *n*=50000). Overall running time is relatively short, for example, for MHS instances it is less than 9 minutes, while for STS instances the running time is less than 15 seconds.

In order to clarify EM performance, direct comparison with the previous GA approach from [16] is performed. Tables 3 and 4 contain data organized as follows:

- the first and the second column contain *m* and *n*;
- the third column contains the optimal solution if it is known in advance. If an optimal solution is not known, the next column displays currently best-known solution;
- next two columns present the GA best solution (*best*) and average total running time (t_{tot}), respectively;
- last two columns contain the EM results, presented in the same way as for the GA.

Table 3: Direct comparison of the results on MHS instances

Inst.			*GA*		EM	
m	*n*	Opt	*best*	t_{tot} (sec)	*best*	t_{tot} (sec)
50	1000	1000	opt	2.582	opt	0.158
50	10000	10000	opt	60.039	opt	3.212
100	1000	1000	opt	4.67	opt	0.334
100	10000	10000	opt	168.603	opt	10.593
100	50000	50000	opt	683.147	49998	216.316
250	1000	1000	opt	8.626	opt	1.062
250	10000	10000	opt	336.894	opt	45.393
500	1000	1000	opt	13.325	opt	2.336
500	10000	10000	opt	437.909	opt	94.473
500	50000	50000	opt	2086.517	opt	486.124

Table 4: Direct comparison of the results on STS instances

Inst.			*GA*		EM	
m	*n*	Opt	*best*	t_{tot} (sec)	*best*	t_{tot} (sec)
9	12	10	best	0.193	best	0.001
15	35	28	best	0.233	best	0.003
27	117	91	best	0.382	best	0.005
45	330	253	best	0.914	best	0.030
81	1080	820	best	2.893	best	0.173
135	3015	2278	best	7.858	best	0.905
243	9801	7381	best	65.409	best	14.953

The direct comparison between GA and EM shows that, although GA has reached all optimal/best-known solutions, EM is much faster, sometimes more than one order of magnitude. Therefore, computational results confirm proposed EM approach as an efficient and robust method for solving MSSP.

5. CONCLUSIONS

This paper is devoted to exploring the results of the new electromagnetic like approach applied to the maximum set splitting problem. Combining scaling technique with a basic attraction-repulsion mechanism boosts the performances of the proposed algorithm. The fast local search procedure additionally improves performances of the system.

In order to show the efficiency of the proposed hybrid EM, a number of experiments are carried out, and the results are compared with the optimal/best-known solutions taken from the literature. The obtained results clearly indicate that EM is a useful tool for solving this problem.

Further research should be directed to parallelization of the EM and run it on a powerful multiprocessor computer. Another direction can be incorporation of this method in some exact solution framework.

REFERENCES

[1] Ali, M.M., Golalikhani, M., "An electromagnetism-like method for nonlinearly constrained global optimization", *Computers & Mathematics with Applications*, 60(8) (2010) 2279-2285.

[2] Andersson, G., and Engebretsen, L., "Better approximation algorithms for set splitting and not-all-equal sat", *Information Processing Letters*, 65 (1998) 305-311.

[3] Birbil, S.I., and Fang, S.C, "An electromagnetism-like mechanism for global optimization", *Journal of Global Optimization*, 25 (2003), 263-282.

[4] Chen, J., and Lu, S., "Improved algorithm for weighted and unweighted set splitting problems", *LectureNotes in Computer Science*, 4598 (2007) 573-547.

[5] Chen, H., and Lu, S., "Improved parameterized set splitting algorithms: A probabilistic approach", *Algorithmica*, 54 (2009) 472-489.

[6] Cutello, V.., and Nicosia, G., "A clonal selection algorithm for coloring, hitting set and satisfiability problems", *Lecture Notes in Computer Science*, 3931 (2006) 324-337. http://www.dmi.unict.it/~nicosia/cop.html

[7] Dehne, F., Fellows, M. and Rosamond, F., "An FPT algorithm for set splitting", *Lecture Notes in Computer Science*, 2880 (2003) 180-191.

[8] Dehne, F., Fellows, M., Rosamond, F., and Shaw, P., "Greedy localization, iterative compression, modeled crown reductions: New FPT techniques, and improved algorithm for set splitting, and a novel 2k kernelization of vertex cover", *Lecture Notes in Computer Science*, 162 (2004) 127-137.

[9] Fulkerson, D. R., Nemhauser, G.L. and Trotter, L.E., "Two computationally difficult set covering problems that arise in computing the l-width of incidence matrices of Steiner triple systems", *Mathematical Programming Study*, 2 (1974) 72-81. http://www.research.att.com/~mgcr/data/steiner-triples.tar.gz

[10] Garcia-Villoria, A., and Moreno R.P., "Solving the response time variability problem by means of the electromagnetism-like mechanism", *International Journal of Production Research*, 48(22) (2010) 6701-6714.

[11] Garey, M., and Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
[12] Guan, X., Dai, X., Li, J., "Revised electromagnetism-like mechanism for flow path design of unidirectional AGV systems", *International Journal of Production Research*, 49(2) (2011) 401-429.
[13] Guruswami, V., "Inapproximability results for set splitting and satisfiability problems with no mixed clauses", *LectureNotes in Computer Science*, 1913 (2000) 155-166.
[14] Kartelj, A., "Electromagnetism metaheuristic algorithm for solving the strong minimum energy topology problem", submitted to *Yugoslav Journal of Operations Research*.
[15] Larson, P.B., and Shelah, S., "The stationary set splitting game", *Mathematical Logic Quarterly*, 54(2) (2008) 187-193.
[16] Lazović, B., Marić, M., Filipović, V., and Savić, A., "An integer linear programming formulation and genetic algorithm for the maximum set splitting problem", *Publications de l'Institut Mathematique*, in press
[17] Lokshtanov, D., and Saurabh, S., "Even faster algorithm for set splitting!", *Proceedings of the International Workshop on Parameterized and Exact Computation - IWPEC 2009*, 288-299.
[18] Lu, S., "Randomized and deterministic parameterized algorithms and their applications in bioinformatics", Ph. D. thesis, Texas A&M University, 2009.
[19] Nederlof, J., and van Rooij, J.M.M., "Inclusion/exclusion branching for partial dominating set and set splitting", *LectureNotes in Computer Science*, 6478 (2010) 204-215.
[20] Tsai, C.Y., Hung, H.L., Lee S.H., "Electromagnetism-like method based blind multiuser detection for MC-CDMA interference suppression over multipath fading channel", *Proceedings of the IEEE International Symposium on Computer Communication Control and Automation -3CA*, 2010, 470-475.
[21] Zhang, J., Ye, Y., and Han, Q., "Improved approximations for max set splitting and max NAE SAT", *Discrete Applied Mathematics*, 142 (2004) 133-149.