Yugoslav Journal of Operations Research
Vol 18 (2008), Number 2, 143-151
DOI: 10.2298/YUJOR0802143K

COMPUTING STRONG METRIC DIMENSION OF SOME SPECIAL CLASSES OF GRAPHS BY GENETIC ALGORITHMS[1]

Jozef KRATICA

Mathematical Institute, Serbian Academy of Sciences and Arts
Kneza Mihaila 36, pp. 367, 11 001 Belgrade, Serbia
jkratica@mi.sanu.ac.yu

Vera KOVAČEVIĆ-VUJČIĆ, Mirjana ČANGALOVIĆ

Faculty of Organizational Sciences
University of Belgrade, Serbia
{verakov | canga}@fon.bg.ac.yu

Received: February 2008 / Accepted: November 2008

Abstract: In this paper we consider the NP-hard problem of determining the strong metric dimension of graphs. The problem is solved by a genetic algorithm that uses binary encoding and standard genetic operators adapted to the problem. This represents the first attempt to solve this problem heuristically. We report experimental results for the two special classes of ORLIB test instances: crew scheduling and graph coloring.

Keywords: Strong metric dimension, genetic algorithms, evolutionary approach.

1. INTRODUCTION

The strong metric dimension problem (SMDP) was recently introduced by A. Sebö and E. Tannier [25] and further investigated by O.Oellermann and J. Peters-Fransen [24]. The strong metric dimension of a graph is a more restricted invariant than the so called metric dimension, which arises in many diverse areas and has been widely investigated [3-7, 9, 13-17, 22, 23, 26, 27].

---

[1] This research was partially supported by Serbian Ministry of Science under the grant no. 144015G.

Given a simple connected undirected graph $G=(V,E)$, where $V=\{1,2,...,n\}$, $|E|=m$, $d(u,v)$ denotes the distance between vertices u and v, i.e. the length of a shortest u-v path. A vertex x of the graph G is said to resolve two vertices u and v of G if $d(x,u)\neq d(x,v)$. An ordered vertex set $S=\{x_1,x_2,...,x_k\}$ of G is a resolving set of G if every two distinct vertices of G are resolved by some vertex of S. Given a vertex t, the k- touple $r(t,S)=(d(t,x_1),d(t,x_2),...,d(t,x_k))$ is called the vector of metric coordinates of t with respect to S. The metric basis of G is a resolving set of the minimum cardinality. The metric dimension of G, denoted by $\beta(G)$, is the cardinality of its metric basis.

Cáceres et al. [4] introduce a more restricted invariant of G, based on the notion of a doubly resolving set, as follows. Vertices x, y of the graph G $(n\geq 2)$ are said to doubly resolve vertices u,v of G if $d(u,x)-d(u,y)\neq d(v,x)-d(v,y)$. A vertex set D of G is a doubly resolving set of G if every two distinct vertices of G are doubly resolved by some two vertices of D. Now the invariant $\psi(G)$ is defined as the cardinality of a minimal doubly resolving set of G. Note that if x, y doubly resolve u,v then $d(u,x)-d(v,x)\neq 0$ or $d(u,y)-d(v,y)\neq 0$ and hence x or y resolves u, v. Therefore a doubly resolving set is also a resolving set and $\beta(G)\leq\psi(G)$.

In this paper we consider the strong metric dimension of G, which is defined as follows. For two distinct vertices u and v of G let us denote by $I[u,v]$ the set of all vertices that belong to some shortest u-v path. A vertex w strongly resolves two vertices u and v if $v\in I[u,w]$ or if $u\in I[v,w]$. A vertex set W of G is a strong resolving set of G if every two distinct vertices of G are strongly resolved by some vertex of W. Now the strong metric dimension $\mathrm{sdim}(G)$ is defined as the cardinality of a minimal strong resolving set of G. It is easy to see that if a vertex w strongly resolves vertices u and v then w also resolves these vertices. Hence every strong resolving set is a resolving set and $\beta(G)\leq\mathrm{sdim}(G)$.

It has been proved in [24] that the problem of finding the strong metric dimension $\mathrm{sdim}(G)$ is NP-hard. Nevertheless, for some simple classes of graphs it is possible to determine $\mathrm{sdim}(G)$ explicitly: the complete graph K_n with n vertices has $\mathrm{sdim}(K_n)=n-1$, the cycle C_n with n vertices has $\mathrm{sdim}(C_n)=\lceil n/2\rceil$ and if T is a tree, $\mathrm{sdim}(T)$ is equal to the number of its leaves minus 1 [25].

If W is a strong resolving set of G, then the set $\{r(v,W)\,|\,v\in V\}$ uniquely determines graph G in the following sense. If G' is the graph with $V(G')=V(G)$ such that W is a strong resolving set of G' and if for all vertices v we have $r_G(v,W)=r_{G'}(v,W)$, then $G=G'$. If W is a resolving set, then the set $\{r(v,W)\,|\,v\in V\}$ need not uniquely determine G (see [25]). The next three examples illustrate previously defined invariants $\beta(G),\psi(G)$ and $\mathrm{sdim}(G)$.

Example 1. Consider the graph G_1 of Figure 1. The set $S_1 = \{A, B\}$ is a resolving set of G_1 since the vectors of metric coordinates for the vertices of G_1 with respect to S_1 are: $r(A, S_1) = (0,1)$; $r(B, S_1) = (1,0)$; $r(C, S_1) = (1,2)$; $r(D, S_1) = (2,1)$. On the other hand, a singleton set, e.g. $\{A\}$, is not a resolving set since $d(B, A) = d(D, A) = 1$. Therefore, S_1 is a minimal resolving set and $\beta(G_1) = 2$.

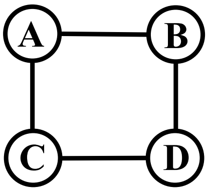

Figure 1: Graph G_1 from *Example 1*

Set S_1 is also a strong resolving set of G_1. Really, all possible sets $I[u,v]$, where u is a vertex of G_1 and v is a vertex of S_1, have the form: $I[A,A]=\{A\}$; $I[B,A]=I[A,B]=\{A,B\}$; $I[C,A]=\{A,C,\}$; $I[D,A]=\{A,B,C,D\}$; $I[B,B]=\{B\}$; $I[C,B]=\{A,B,C,D\}$; $I[D,B]=\{B,D\}$. Starting from these sets it is easy to check that for every two distinct vertices of G_1 there exists a vertex from S_1 which strongly resolves them. For example, since $C \in I[D,A]$, C and D are strongly resolved by A, while B and D are strongly resolved by B, as $B \in I[D,B]$. Hence, $2 = \beta(G_1) \le \mathrm{sdim}(G_1) \le |S_1| = 2$ which implies $\mathrm{sdim}(G_1)=2$.

Note that S_1 is not a doubly resolving set because $d(C,A) - d(A,A) = d(C,B) - d(A,B) = 1$. Similarly, it can be verified that none of the subsets consisting of two nodes is a doubly resolving set of G_1. But, $S_2=\{A,B,C\}$ is a doubly resolving set, as the vectors of metric coordinates with respect to S_2 are: $r(A,S_2)=(0,1,1)$; $r(B,S_2)=(1,0,2)$; $r(C,S_2)=(1,2,0)$; $r(D,S_2)=(2,1,1)$. Therefore, S_2 is a minimal doubly resolving set and $\psi(G_1)=3$. □

Example 2. For graph G_2 of Figure 2 values $\beta(G_2)$, $\psi(G_2)$ and $\mathrm{sdim}(G_2)$ are all different. Namely, it is easy to check that $\{A,B\}$ is a minimal resolving set, $\{A,B,E,F\}$ is a minimal doubly resolving set, while $\{A,B,E\}$ is a minimal strong resolving set of G_2. Therefore, $\beta(G_2)=2$, $\psi(G_2)=4$ and $\mathrm{sdim}(G_2)=3$. □

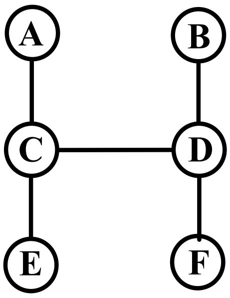

Figure 2: Graph G_2 from *Example 2*

Example 3. The Petersen graph G_3, given on Figure 3, has $\beta(G_3)=\psi(G_3)=3$ and $\mathrm{sdim}(G_3)=8$. Really, it can be shown that $\{A,C,I\}$ is both a minimal resolving and a minimal doubly resolving set, while $\{A,B,C,D,E,F,G,I\}$ is a strong resolving set of G_3. □

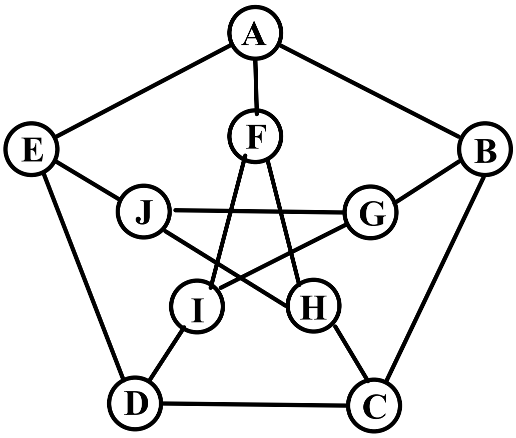

Figure 3: Petersen graph G_3 from Example 3

The paper is organized as follows. Section 2 includes the main features of a genetic algorithm (GA) implementation designed for the SMDP, while in Section 3 we report computational results on two special classes of ORLIB test instances: crew scheduling and graph coloring.

2. GENETIC ALGORITHM FOR SMDP

GAs are stochastic search techniques which imitate some spontaneous optimization processes in the natural selection and reproduction. At each iteration (generation) GA manipulates with a set (population) of encoded solutions (individuals), starting from either randomly or heuristically generated one. Individuals from the current population are evaluated using a fitness function to determine their qualities. Good individuals are selected to produce the new ones (offspring), applying operators inspired from those of genetics (crossover and mutation), and they replace some of the individuals from the current population. Detailed description of GAs is out of this paper's scope and it can be found in [11, 21, 29]. Extensive computational experience on various optimization problems shows that GA often produces high quality solutions in a reasonable time [8, 11, 17-20, 28-30]. In most cases GA has shown to be robust with respect to parameter choice in reasonable bounds.

In [17] we developed a genetic algorithm for determining the metric dimension of an arbitrary graph which was successfully tested on several classes of graph instances. A similar genetic approach has also been successfully applied for finding the cardinality of a minimal doubly resolving set [18]. As the strong metric dimension problem is closely related to the metric dimension problem, the existing genetic algorithm from [17] was easily accommodated to solve the SMDP. It was done by replacing the procedure of checking whether the current individual corresponds to a resolving set or not by the procedure of identifying it as a code of a strong resolving set.

Here, we will give only the main features of our GA implementation for the strong metric dimension problem (for more details see [17]).

The algorithm uses a binary encoding of the individuals, where each solution $W \subseteq V$ (i.e. a candidate for a strong resolving set) is represented by a binary string of length n. Digit 1 at the i-th position of the string denotes that the vertex i belongs to W, while 0 shows the opposite.

For each pair of distinct vertices $u,v \in V$ we check whether there exists $w \in W$ such that $d(u,w) = d(u,v)+d(v,w)$ or $d(v,w) = d(v,u)+d(u,w)$. If this condition is satisfied, then W is a strong resolving set as the condition is equivalent to $v \in I[u,w]$ or $u \in I[v,w]$. If it is not satisfied, then the following "reparation" technique is applied: an additional randomly chosen vertex from $V \setminus W$ is added to W and this procedure is iterated until W becomes a strong resolving set. The cardinality of the obtained strong resolving set W is the objective value of the current individual.

Population in each generation contains $N_{pop} = 150$ individuals. The fitness f_{ind} of individual $ind = 1,2, ..., N_{pop}$ is computed by scaling objective values obj_{ind} of all individuals into the interval [0,1], so that the best individual ind_{min} has fitness 1 and the worst one ind_{max} has fitness 0. More precisely, $f_{ind} = \frac{obj_{ind_{\max}} - obj_{ind}}{obj_{ind_{\max}} - obj_{ind_{\min}}}$. Next, individuals are arranged in non-increasing order of their fitness: $f_1 \geq f_2 \geq ... \geq f_{N_{pop}}$.

The population of the first generation is randomly generated, providing the maximal diversity of the genetic material. In order to prevent an undeserved domination of some individuals in the current population an elitist strategy is used. The fitness of $N_{elite} = 100$ elite individuals over the population are decreased by folowing formula.

$$f_{ind} = \begin{cases} f_{ind} - \overline{f}, & f_{ind} > \overline{f} \\ 0, & f_{ind} \leq \overline{f} \end{cases} \quad , \quad 1 \leq ind \leq N_{elite} \ , \quad \overline{f} = \frac{1}{N_{pop}} \sum_{ind=1}^{N_{pop}} f_{ind}$$

The first N_{elite} individuals are directly passing to the next generation. Genetic operators are applied to the rest of the population, so that only one third is replaced in every generation.

The selection operator, which chooses the parent individuals that will produce offspring in the next generation, is an improved tournament selection operator known as the fine-grained tournament selection (see [10, 11]). This operator uses a real (rational) parameter F_{tour} which denotes the desired average tournament size. Two types of tournaments are performed: the first type is held k_1 times on $\lfloor F_{tour} \rfloor$ individuals, while the second type is applied k_2 times with $\lceil F_{tour} \rceil$ individuals participated, so

$$F_{tour} \approx \frac{k_1 \lfloor F_{tour} \rfloor + k_2 \lceil F_{tour} \rceil}{N_{nnel}} \text{ , where } N_{nnel} = N_{pop} - N_{elite} \text{ .}$$

In our implementation $F_{tou} = 5.4$ and, therefore, the corresponding values k_1 and k_2 for $N_{nnel} = 50$ non-elitist individuals are 20 and 30, respectively.

When a pair of parents is selected, an one-point crossover operator is applied with probability p_{cross} = 0.85. It means that approximately 85% pairs of individuals exchange segments of their genetic codes.

The simple mutation operator is performed by changing a randomly chosen gene in the code of the individual, with a certain mutation rate (probability) p_{mut} . In order to prevent a premature convergence the mutation rate is increased on each so-called frozen gene, i.e. a gene on a certain position with the same value which appears in all individuals in the current population. In our implementation the mutation rate depends on dimension n and it is 2.5 times higher on frozen genes ($p_{mut} = 1.0/n$) then on non-frozen ones ($p_{mut} = 0.4/n$).

Duplicated individuals are removed from the current population by setting their fitness to zero. This is a very effective method for saving diversity of the genetic material and keeping the algorithm away from a premature convergence. Individuals with the same objective value and different genetic codes, in some cases may dominate in the population and lead the algorithm to a local optimum. For that reason, number of such individuals in one generation is limited to $N_{rv} = 40$.

The run-time performance of GA is optimized by a caching technique. The main idea is to avoid computing the same objective value every time when genetic operators produce individuals with the same genetic code. Evaluated objective values are stored in a hash-queue data structure using the least recently used (LRU) caching technique. When the same code is obtained again, its objective value is taken from the cache memory, that provides time-savings. In this implementation the number of individuals stored in the cache memory is limited to 5000.

3. EXPERIMENTAL RESULTS

In this section we present the computational results for two different ORLIB classes of graph instances: crew scheduling [1, 2] and graph coloring [1, 12]. This is the first computational study of the strong metric dimension problem and it was not possible to make a comparison with other approaches. The GA tests were performed on an AMD Sempron 1.6 GHz with 256 MB memory, under Linux (Knoppix 5.0) operating system. The stopping criterion was the maximum number of generations equal to 5000 or at most 2000 generations without improvement of the objective value. The GA has been run 20 times for each instance and the results are summarized in Table 1 and Table 2. The tables are organized as follows:

- the first three columns contain the test instance name, the number of nodes and edges respectively;
- the fourth column contains the best GA solution (named GA_{best}) obtained in 20 runs;
- the average execution time (t) used to reach the final GA solution for the first time is given in the fifth column, while the sixth and seventh column (t_{tot} and gen) show the average total execution time and the average number of generations for finishing GA, respectively;
- the eighth and the ninth column ($agap$ and σ) contain information on the average solution quality: $agap$ is a percentage gap defined as $agap = \frac{1}{20}\sum_{i=1}^{20} gap_i$, where

$gap_i = 100 \cdot \frac{GA_i - GA_{best}}{GA_{best}}$ and GA_i represents the GA solution obtained in the *i*-th run, while σ is the standard deviation of gap_i, $i = 1,2,...,20$, obtained by formula $\sigma = \sqrt{\frac{1}{20}\sum_{i=1}^{20}(gap_i - agap)^2}$.

- in the last two columns *eval* represents the average number of the objective function evaluations, while *cache* displays savings (in percent) achieved by using the caching technique.

Table 1: Results on crew scheduling ORLIB instances

Inst.	*n*	*m*	GA_{best}	*t*	t_{tot}	*gen*	*agap*	σ	*eval*	*cache*
csp50	50	173	29	4.108	26.769	2271	3.103	3.429	30309	73.5
csp100	100	715	61	100.787	528.21	2420	3.279	3.864	49099	59.1
csp150	150	1355	98	540.584	3166.3	2867	8.98	7.418	70178	50.7
csp200	200	2543	144	4978.7	8047.6	3010	8.125	5.135	81943	45.4
csp250	250	4152	178	4941.3	17060	2478	3.034	2.372	70516	43.2

Table 2: Results on graph coloring ORLIB instances

Inst.	*n*	*m*	GA_{best}	*t*	t_{tot}	*gen*	*agap*	σ	*eval*	*cache*
gcol1	100	2487	91	18.119	172.732	2142	0.769	0.531	32411	69.6
gcol2	100	2487	91	28.443	182.663	2294	0.769	0.531	34094	70.1
gcol3	100	2482	91	49.108	198.800	2500	0.989	0.811	38425	68.8
gcol4	100	2503	91	33.926	173.360	2343	0.879	0.463	32905	72.4
gcol5	100	2450	91	22.734	166.339	2191	0.989	0.624	31639	71.2
gcol6	100	2537	91	46.603	194.784	2478	0.330	0.531	36725	70.3
gcol7	100	2505	91	15.826	170.525	2128	1.099	0.518	33391	68.6
gcol8	100	2479	90	16.111	177.131	2134	2.000	0.703	33075	68.8
gcol9	100	2486	91	15.676	178.285	2133	1.099	0.518	34353	67.8
gcol10	100	2506	91	17.327	167.539	2137	0.659	0.567	31565	70.5
gcol11	100	2467	91	18.700	169.150	2146	0.769	0.531	31415	70.7
gcol12	100	2531	91	20.990	166.088	2184	0.440	0.567	31359	71.5
gcol13	100	2467	91	25.398	171.383	2237	0.989	0.348	32965	70.5
gcol14	100	2524	91	27.909	169.664	2282	1.099	0.518	33341	70.9
gcol15	100	2528	91	44.315	182.577	2436	0.989	0.624	36497	70.6
gcol16	100	2493	91	33.121	173.897	2349	0.659	0.567	32784	72.0
gcol17	100	2503	91	25.073	172.929	2214	0.440	0.567	31882	71.4
gcol18	100	2472	91	34.893	173.495	2337	0.440	0.567	31481	73.1
gcol19	100	2527	91	14.407	163.039	2106	0.659	0.567	31065	70.5
gcol20	100	2420	91	31.080	175.908	2287	1.209	0.962	33936	70.1

It is important to note that the checking whether a set is a strong resolving set has the complexity $O(n^2 \cdot \mathrm{sdim}(G))$, while the checking whether a set is a resolving set requires only $O(n \cdot log\ n \cdot \beta(G))$ operations. Also, the strong metric dimension of a graph *G* is usually much larger than the metric dimension. Consequently, the GA running times in Table 1 and Table 2 are much larger than the GA running times for the metric dimension problem in [17]. For example, for instance csp250 the ratio $\frac{n^2 \cdot \mathrm{sdim}(G)}{n \cdot \log n \cdot \beta(G)}$ is aproximately 300, while the real running time ratio is $17060/25.4 \approx 672$. For this reason, experiments in this paper are performed only on a subset of crew scheduling instances with $n \leq 250$ in Table 1 and graph coloring instances with $n = 100$ in Table 2.

The running times displayed in column t_{tot} would be considerebly larger without the caching technique, which can be seen from the last column of Table 1 and Table 2. Time savings vary from 43.2% to 73.5% for crew scheduling instances, i.e. from 67.8% to 73.1% for graph coloring instances.

4. CONCLUSIONS

In this paper an evolutionary metaheuristic for solving the strong metric dimension problem is presented. It uses the binary representation, the mutation with frozen genes, a limited number of different individuals with the same objective value and the caching technique. Checking whether a set is a strong resolving set requires that for each pair of the vertices of the graph one member of that set which strongly resolves them is found. Infeasible individuals, candidates which are not strongly resolving sets, are corrected by adding necessary vertices in order to become feasible.

Further research should be directed to testing instances of larger dimensions on more powerful and/or parallel computers as well as to using computational results to generate theoretical hypotheses about the strong metric dimension.

REFERENCES

[1] Beasley, J.E., "Obtaining test problems via Internet", *Journal of Global Optimization*, 8 (1996) 429-433. http://www.brunel.ac.uk/depts/ma/research/~jeb/orlib

[2] Beasley, J.E., and Cao, B., "A tree search algorithm for the crew scheduling problem", *European Journal of Operational Research*, 94 (1996) 517-526.

[3] Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffmann, M., Mihalak, M., and Ram, L.S., "Network discovery and verification", *IEEE Journal on Selected Areas in Communications,* 24(12) (2006) 2168-2181.

[4] Cáceres, J., Hernando, C., Mora, M., Pelayo, I.M., Puertas, M.L., Seara, C., and Wood, D.R., "On the metric dimension of Cartesian products of graphs", *SIAM Journal in Discrete Mathematics,* 21(2) (2007) 423-441.

[5] Chartrand, G., Eroha, L., Johnson, M.A., and Oellermann, O.R., "Resolvability in graphs and the metric dimension of a graph", *Discrete Applied Mathematics,* 105 (2000) 99-113.

[6] Chartrand, G., Poisson, C., and Zhang, P., "Resolvability and the upper dimension of graphs", *Computers & Mathematics with Applications,* 39 (2000) 19-28.

[7] Currie, J.D., and Oellerman, O.R., "The metric dimension and metric independence of a graph", *Journal of Combinatorial Mathematics and Combinatorial Computing,* 39 (2001) 157-167.

[8] Djurić, B., Kratica, J., Tošić, D., and Filipović, V., "Solving the maximally balanced connected partition problem in graphs by using genetic algorithm", *Computing and Informatics,* 27 (2) (2008), in press.

[9] Fehr, M., Gosselin, S., and Oellermann, O.R., "The metric dimension of Cayley digraphs", *Discrete Mathematics,* 306(1) (2006) 31-41.

[10] Filipović, V., "Fine-grained tournament selection operator in genetic algorithms", *Computing and Informatics,* 22 (2003) 143-161.

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

[12] Fleurent, C., and Ferland, J.A., "Genetic and hybrid algorithms for graph coloring", *Annals of Operations Research,* 63 (1996) 437-461.

[13] Harary, F., and Melter, R.A., "On the metric dimension of a graph", *Ars Combinatoria,* 2 (1976) 191-195.

[14] Hernando, C., Mora, M., Pelayo, I.M., Seara, C., Cáceres, J., and Puertas, M.L., "On the metric dimension of some families of graphs", *Electronic Notes in Discrete Mathematics,* 22 (2005) 129-133.

[15] Hernando, C., Mora, M., Pelayo, I.M., Seara, C., and Wood, D.R., "Extremal graph theory for metric dimension and diameter", *Electronic Notes in Discrete Mathematics,* 29 (2007) 339-343.

[16] Khuller, S., Raghavachari, B., and Rosenfeld, A., "Landmarks in graphs", *Discrete Applied Mathematics,* 70 (1996) 217-229.

[17] Kratica, J., Kovačević-Vujčić, V., and Čangalović, M., "Computing the metric dimension of graphs by genetic algorithms", *Computational Optimization and Applications,* (2008), in press, DOI:10.1007/s10589-007-9154-5

[18] Kratica, J., Čangalović, M., and Kovačević-Vujčić, V., "Computing minimal doubly resolving sets of graphs", *Computers & Operations Research*, (2009), in press, DOI: 10.1016/j.cor.2008.08.002

[19] Ljubić, I., and Raidl, G.R., "A memetic algorithm for minimum-cost vertex-biconnectivity augmentation of graphs", *Journal of Heuristics*, 9 (2003) 401-427.

[20] Ljubić, I., "Exact and memetic algorithms for two network design problems", PhD thesis, Institute of Computer Graphics, Vienna University of Technology, 2004.

[21] Mitchell, M., *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Massachusetts, 1999.

[22] Oellermann, O., Fehr, M., and Gosselin, S., "The metric dimension of Cayley digraphs", *Discrete Mathematics*, 306 (2006) 31-41.

[23] Oellermann, O., and Peters-Fransen, J., "The metric dimension of Cartesian products of graphs", *Utilitas Mathematica,* 69 (2006) 33-41.

[24] Oellermann, O., and Peters-Fransen, J., "The strong metric dimension of graphs and digraphs", *Discrete Applied Mathematics,* 155 (2007) 356-364.

[25] Sebö, A., and Tannier, E., "On metric generators of graphs", *Mathematics & Operations Research,* 29(2) (2004) 383-393.

[26] Shanmukha, B., Sooryanarayana, B., and Harinath, K.S., "Metric dimension of wheels", *Far East Journal of Applied Mathematics,* 8 (2002) 217-229.

[27] Slater, P.J., "Leaves of trees", *Congr. Numerantium*, 14 (1975) 549-559.

[28] Stanimirović, Z., "Solving some discrete location problems by using genetic algorithms", Master's thesis, Faculty of Mathematics, University of Belgrade, 2004. (in Serbian)

[29] Stanimirović, Z., "Genetic algorithms for solving some NP-hard hub location problems", Ph.D. thesis, University of Belgrade, Faculty of Mathematics, 2007. (in Serbian),

[30] Stanimirović, Z., "A genetic algorithm approach for the capacitated single allocation p-hub median problem", *Computing and Informatics*, 27 (2008), in press.