Mathematics Interdisciplinary Research **9** (4) (2024) 413 − 424

Original Scientific Paper

Genetic Algorithm for Finding the Global Forcing Number of Bipartite Graphs

Sara Oskoueian, Mostafa Tavakoli[⋆] *and Narjes Sabeghi*

Abstract

Consider a graph $G = (V(G), E(G))$, where a perfect matching in G is defined as a subset of independent edges with $\frac{|V(G)|}{2}$ elements. A global forcing set is a subset S of E such that no two disjoint perfect matchings of G coincide on it. The minimum cardinality of global forcing sets of G is called the global forcing number (GFN for short).
This paper addresses the NP-hard problem of determining the global forcing number for perfect matchings. The focus is on a Genetic Algorithm (GA) that utilizes binary encoding and standard genetic operators to solve this problem. The proposed algorithm is implemented on some chemical graphs to illustrate the validity of the algorithm. The solutions obtained by the GA are compared with the results from other methods that have been presented in the literature. The presented algorithm can be applied to various bipartite graphs, particularly hexagonal systems. Additionally, the results of the GA improve some results that have already been presented for finding GFN.

Keywords: Perfect matching, Global forcing set, Genetic algorithm, Hexagonal system.

2020 Mathematics Subject Classification: 05C70, 05C85.

How to cite this article
S. Oskoueian, M. Tavakoli and N. Sabeghi, Genetic algorithm for finding the global forcing number of Bipartite graphs, *Math. Interdisc. Res.* **9** (4) (2024) 413-424.

⋆Corresponding author (E-mail: m_tavakoli@um.ac.ir)
Academic Editor: Gholam Hossein Fath-Tabar
Received 20 July 2024, Accepted 3 October 2024
DOI: 10.22052/MIR.2024.255203.1472

1. Introduction

Matching theory plays a vital role in the connection between Graph Theory and Chemistry [1, 2]. Let $G = (V(G), E(G))$ be a graph with vertex set $V(G)$ and edge set $E(G)$, where $|V(G)| = n$ and $|E(G)| = m$. A subset $M \subseteq E(G)$ of non adjacent edges of G is called a *matching* of G. A *perfect matching* in a graph G is a matching M with $\frac{|V(G)|}{2}$ elements. A *forcing set* S of a perfect matching M in graph G is any subset of M that does not appear in any other perfect matching of G. The minimum cardinality of the forcing sets of M is called the *forcing number*. In earlier chemical literature, the concepts of perfect matching and the forcing number are also called the "Kekulé structure" and the "innate degree of freedom", respectively. These concepts were introduced in a paper by Harary, Klein and Zivković [3], with their origins traced back to the work of Randić and Klein [4]. The forcing number of a perfect matching finds resonance structures of a given molecule in organic chemistry. Note that forcing sets and forcing numbers are defined with respect to particular perfect matching of G. Vukičević et al. [5] classified all the Kekulé structures of C_{60} into six classes based on their innate degree of freedom. However, there were still considerable different structures among the Kekulé structures with the same innate degree of freedom. Hence Vukičević and Sedler [2] introduced the "global" concepts, considering all perfect matchings. They defined the *global forcing set (GFS)* of G as a set $S \subseteq E(G)$ that can distinguish all perfect matchings of G. In other words, let $\mu(G)$ denote the set of all perfect matchings in G and $S = \{e_1, ..., e_q\} \subseteq E(G)$. In what follows, we will use notation $M|_S$ to denote $M \cap S$. We define $r(M|_S) = (d_1, ..., d_q)$ where $d_i = 1$ if $e_i \in M$ and $d_i = 0$ otherwise, $i \in \{1, \ldots, q\}$. If $r(M_1|_S) \neq r(M_2|_S)$ for each $M_1, M_2 \in \mu(G)$, then S is termed the global forcing set for perfect matchings of G. Any GFS for $\mu(G)$ in G with the smallest cardinality is called a *minimum global forcing set* and its cardinality, denoted by φ_{g_m}, is called the *global forcing number (GFN)* for perfect matchings [6].

Example 1.1. Consider the graph Naphthalene ($C10H8$) shown in Figure 1, which is an organic compound with two hexagons. Naphthalene contains three different perfect matchings as follows:
$\mu(C10H8) = \{M_1 = \{e_2, e_5, e_7, e_{10}, e_{11}\}, M_2 = \{e_2, e_4, e_6, e_8, e_{10}\}, M_3 = \{e_1, e_3, e_5, e_7, e_9\}\}$.

The set $S_1 = \{e_2, e_5, e_7\}$ is a GFS for G, since we have: $r(M_1|_{S_1}) = (1, 1, 1)$; $r(M_2|_{S_1}) = (1, 0, 0)$; $r(M_3|_{S_1}) = (0, 1, 1)$. However S_1 is not minimum GFS because $S_2 = \{e_2, e_5\}$ is also a GFS with smaller cardinality. In addition, the set $S_3 = \{e_2\}$ is not GFS because $r(M_1|_{S_3}) = r(M_2|_{S_3}) = 1$.

The computation of the GFN for perfect matchings poses a formidable challenge due to the NP-hard nature of enumerating all perfect matchings, see [7]. Despite this complexity, efforts have been made to find GFN in certain families of graphs. In [8], Vukičević and Sedler presented a bound on the GFN for the particular triangular grid graphs. Vukičević et al. [9] obtained a formula for the GFN

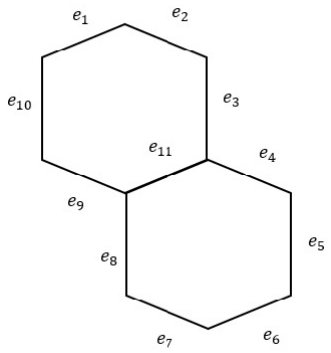

Figure 1: Naphthalene.

of grid graphs. Klavžar et al. [10] proposed an integer programming model for determining minimum GFS and GFN for perfect matchings of G. Their proposed model is based on the fact that the GFS should have a non-empty intersection with the symmetric difference of every pair of perfect matchings in G. Indeed, the computational applicability of their model is limited since the number of constraints of the presented model is huge. In this paper, our focus is on bipartite graphs, especially hexagonal systems. In the next section, we will discuss further on the related works.

Although the GFN problem is an NP-hard combinatorial optimization problem and the size of its solution space can grow with the problem dimension, heuristic methods can be useful in solving such problems. In this paper, we present a genetic algorithm that uses the binary encoding for the GFN problem. This algorithm can be applied to any bipartite graph.

The rest of this paper is organized as follows. In Section 2, we present a review of some results related to GFN for hexagonal systems and emphasize the necessity of employing algorithms such as what we propose. In Section 3, we propose a genetic algorithm for determining the GFN of perfect matchings in a given bipartite graph. The computational results of the GA implementation on various chemical graphs are presented in Section 4.

2. Background and some results

Let us now consider the related works on GFN of the hexagonal systems. We refer the reader to [11, 12] for terminology and notation not defined here. A *hexagonal (benzenoid) system* (HS) is a finite connected plane graph with no cut vertices such that every interior region is a regular hexagon with a side length of one. The benzenoid graphs are simple and bipartite, see [13, 14]. Let H be an HS. Graph H is called *catacondensed* if each vertex of H is on the border of H, otherwise

the graph is *pericondensed*. H is said to be *normal*, if each edge of H appears in some perfect matching of H. It is clear that all catacondensed HS are normal. In [11] Došlić et al. proved two results concerning GFN of catacondensed and pericondensed HS.

Theorem 2.1. *([11]). Let H be a catacondensed benzenoid with f hexagons. Then $\varphi_{g_m}(H) = f$.*

Theorem 2.2. *([11]). Let H be a pericondensed benzenoid with f hexagons. Then $\varphi_{g_m}(H) \leq f - 1$.*

Cai and Zhang [15] provided an exact formula for the GFN of two kinds of pericondensed HS, parallelogram $B_{p,q}$ and zigzag multiple chain $Z(l,k)$.

Theorem 2.3. *([15]). Let $B_{p,q}$ be a parallelogram with q rows, each row consisting of p hexagons. If $q \leq p$, then $\varphi_{g_m}(B_{p,q}) = \frac{pq+q}{2}$ if q is even and $\varphi_{g_m}(B_{p,q}) = \frac{pq+p}{2}$ otherwise.*

Theorem 2.4. *([15]). Let $Z(l,k)$ be zigzag multiple chain with l row and k columns of hexagons. Then $\varphi_{g_m}(Z(l,k)) = l\lceil\frac{k}{2}\rceil + \lfloor\frac{k}{2}\rfloor$.*

Cai and Zhang [12] gave a sharp lower bound on the GFN of a normal HS. Besides, they presented a formula to find the GFN for a divisible HS.

Theorem 2.5. *([12]). Let H be a normal HS with f hexagons. Then $\varphi_{g_m}(H) \geq \lceil\frac{2f - |S_0|}{3}\rceil$, where S_0 is a maximum subset of S such that any two edges of S are not in a hexagon of H.*

Based on the above theorems, there are few results for a pericondensed HS. It is obvious that according to Theorem 2.2, only the upper bound can be obtained for the GFN of pericondensed HS, in general. By Theorems 2.3 and 2.4, there exist exact formulas for the GFN of two kinds of pericondensed HS, parallelogram $B_{p,q}$ and zigzag multiple chain $Z(l,k)$. On the other hand, Došlić [11] proved that the GFN of an HS is equal to the sum of the GFN of its normal component. Hence it makes sense to achieve the GFN of normal HS. Theorem 2.5 provides a lower bound for GFN of a normal HS, and the explicit value for GFN is not available. Hence computing of the GFN for every kind of normal pericondensed HS is a challenging task. In response to this challenge, our paper presents a genetic algorithm for determining the GFN and minimum GFS of a bipartite graph including hexagonal systems, in general. We present the results of our algorithm's performance on several normal pericondensed HS which involve, Benzo and Ovalene. In the present work, we show that the results of our algorithm improve the bounds for GFN presented in the mentioned theorems. For instance, we implement our algorithm on two categories of HS to illustrate the process and the validity of the

algorithm. These categories are some normal pericondensed graphs which includes perylene, Benzo, coronene and Ovalene on one side, and HS including catacondensed, parallelogram, zigzag multiple chain on the other side, see Figures 2 to 4.

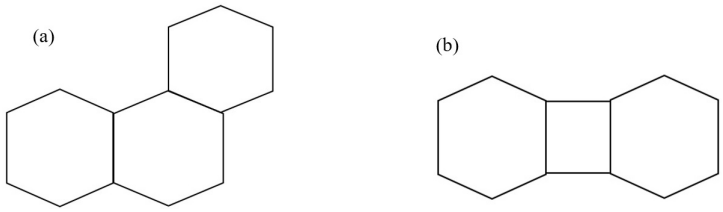

Figure 2: Catacondenesed: (a) Phenanthrene($C14H10$) (b) Biphenylene($C12H8$).

3. Genetic algorithm for the GFN problem

The GFN problem is a well-known NP-hard problem in combinatorial optimization, where the size of its solution space exponentially increases with the problem, see [16]. It is reasonable to seek an approach to solve such problems. One powerful technique to solve this category of problems is the use of meta-heuristics. Although meta-heuristic algorithms cannot always guarantee to obtain the true global optimal solution, they can provide results for a range of problems in a reasonable amount of time.

In this section, we employ Genetic Algorithm (GA) as one of the meta-heuristic tools. The basic steps of GA are described as follows:

1. Evaluation

At each iteration (generation), GA operates on a set of individual solutions known as a population. Each individual in the population, referred to as a chromosome, represents a potential solution to a given problem, starting from either randomly or heuristically produced one. The fitness function is applied to rank individuals and guides the algorithm in selecting candidates during the phase. Hence a fitness score is assigned to each individual. In our algorithm, we have considered fitness score as the cardinality of the GFS described by a chromosome.

2. Selection

The selection operator among the population selects the best individual that has higher fitness scores for the next generation. Here, the selection operator utilizes the roulette wheel. This selection is a technique to randomly choose parents for reproduction based on their fitness scores.

3. Crossover

In the crossover stage, two individuals are chosen to be parents for the next generation and some portions of chromosome are exchanged between the individuals to produce the new chromosomes. There we have single-point crossover. This

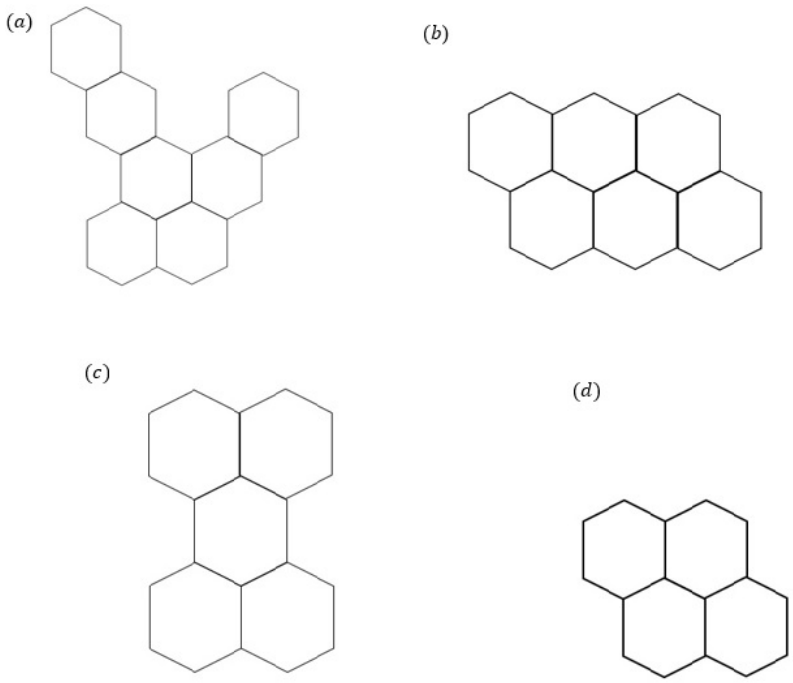

Figure 3: (a) Multiplechain (b) $B_{3,2}$ (c) Perylene($C20H12$) (d) $Z(2,2)$.

crossover is a technique where the selected two parents, is cut at the crossover point to create the child.

4. Mutation

By applying the mutation operator, new individuals are created and they replace some of the individuals from the current population.

A cycle of the selection, crossover, and mutation forms one iteration in GA. In this section, we describe a GA algorithm for determining the GFN for perfect matchings of a given bipartite graph G. The main reason for focusing on bipartite graphs is that the enumerating of all the perfect matchings in a graph is NP-hard in general [17]. Fukuda[18] presented a polynomial time algorithm to find all the perfect matchings for bipartite graph, where the input of this algorithm is an initial perfect matching of the graph. Thereby, we apply Fukuda's algorithm to find all the perfect matchings of the bipartite graph. The details of our GA implementation are given in Algorithm 1. The binary encoding of the fitness score of individuals used in presented Algorithm 2. Each solution S (candidate for a GFS) is presented in the population by a string of length m. Character 1 at the i-th place of the string shows that edge i belongs to S, while 0 denotes that it is not. If vectors of $r(M_i|_S)$ for all $M_i \in \mu(G)$ are different, then S is a GFS.

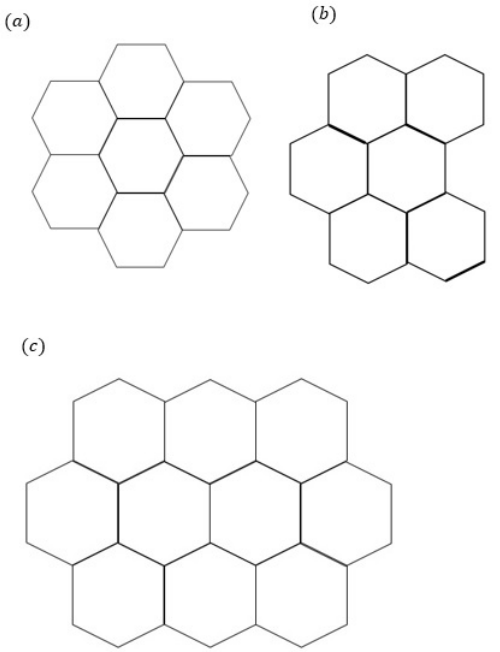

Figure 4: Normal pericondensed HS: (a) Coronone($C24H12$) (b) Benzo($C22H12$) (c) Ovalene($C32H14$).

Algorithm 2 is applied in order to determine whether S, current individual by code ind, is a GFS and if the answer is positive, to reduce its cardinality.

Algorithm 1 GA

1: Generate the initial population of m chromosomes randomly.
2: **while** number of iterations<Max **do**
3: fitness evaluation by Fitness-Function.
4: selection based Roulette Wheel selection.
5: crossover based Single-Point crossover.
6: mutation based Swapping.
7: **return** the GFN of the best solution.

Algorithm 2 *Fitness − Function*

1: for solution S, sort all vectors $r(M_i|_S)$ for all $M_i \in \mu(G)$, $i = 1, \ldots, t$ according to the lexicographically increasing (or decreasing) order in the sequence $r(M_1|_S), ..., r(M_t|_S)$.
2: **for** $i = 1, ..., t-1$ **do**
3: find $j(i)$ as the minimal coordinate where the vectors $r(M_i|_S)$ and $r(M_{i+1}|_S)$ are different.
4: **end for**
5: **if** for some i, $r(M_i|_S) = r(M_{i+1}|_S)$ (S is not a GFS) **then** choose an additional edge randomly from $E \setminus S$ and add to S and go to step 1.
6: determine $r = max\{j(1), j(2), ..., j(t-1)\}$ and keep in S the first r elements.

As mentioned in step 5 of Algorithm 2, if the solution S is not a GFS, then the "reparation" technique is applied: S is updated randomly by adding an edge from $E \setminus S$ to S, and Algorithm 2 is applied again. This would be iterated until S becomes a GFS. The cardinality of the final GFS, S, becomes the fitness score for the current individual. In step 6, r is the maximum of the minimal coordinate $j(i)$ where the vectors $r(M_i|_S)$ and $r(M_{i+1}|_S)$ are different. Thus the vectors $r(M_i|_S)$ for all $M_i \in \mu(G)$, $i = 1, ..., t$ are different at the first r elements i.e. r is minimum GFS for the current solution.

4. Experimental results

In this paper, we provide a genetic algorithm for determining the minimum GFS and GFN for perfect matchings of a bipartite graph G. Oskoueian et al. [19] presented a hybrid greedy approximation algorithm to find the minimum GFS and GFN for the perfect matchings of a bipartite graph G. Their algorithm generates a solution with acceptable approximation ratio. They implemented the proposed algorithm on several functional chemical graphs and obtained the minimum GFS and GFN for these graphs. Table 1 compares the results obtained by the GA with the result of the greedy algorithm (Gd) presented in [19]. The fourth column of Table 1 contains the best greedy GFN and the fifth column contains the minimum GFS. It can be seen that for graphs such as Coronone, Benzo, and Ovalene the generated GFN obtained by GA is better than the generated GFN obtained by the greedy algorithm. Besides, the mentioned graphs are in the category of normal pericondensed. For this category of HS, Došlić [11] computed a bound for the GFN. In Table 1, U_b denotes the upper bound of GFN for pricondensed HS. Meanwhile, GA gave a better GFN than bound. Also, in Table 1, E_v denotes the exact value of GFN for catacondensed HS and some pericondensed HS that is described by Theorems 2.1, 2.3 and 2.4. Observe that the results obtained by GA match the exact value of GFN for these graphs. In addition, it is important

to notice that the minimum GFS is not necessarily unique and so the set S obtained by GA may be different from the set S obtained by the greedy algorithm. Algorithms 1 and 2 has been coded in MATLAB R2019b (Intel(R) Pentium(R) CPU2117U@1.80GHz). The performance plot of GA for Ovalene are shown in

Table 1: Results on chemical graphs.

Chemical name	GA_{best}	S_{GA}	Gd_{best}	S_{Gd}	U_b	E_v
Phenanthrene	3	$\{e_9, e_{10}, e_{14}\}$	3	$\{e_1, e_3, e_4\}$	-	3
Biphenylene	3	$\{e_9, e_{11}, e_{14}\}$	3	$\{e_1, e_3, e_4\}$	-	3
$Z(2,2)$	3	$\{e_7, e_8, e_{17}\}$	3	$\{e_3, e_{12}, e_{13}\}$	3	3
Perylene	4	$\{e_9, e_{10}, e_{14}, e_{17}\}$	5	$\{e_1, e_2, e_3, e_{14}, e_{16}\}$	4	-
Multiplechain	5	$\{e_3, e_{13}, e_{17}, e_{23}, e_{29}\}$	5	$\{e_1, e_4, e_{17}, e_{21}, e_{25}\}$	6	5
$B_{3,2}$	4	$\{e_1, e_6, e_{10}, e_{19}\}$	4	$\{e_5, e_{11}, e_{15}, e_{17}\}$	5	4
Coronone	6	$\{e_5, e_{11}, e_{15}, e_{17}, e_{24}, e_{28}\}$	7	$\{e_1, e_2, e_5, e_8, e_9, e_{17}, e_{20}\}$	6	6
Benzo	5	$\{e_6, e_9, e_{10}, e_{22}, e_{24}\}$	6	$\{e_1, e_2, e_6, e_9, e_{15}, e_{24}\}$	5	-
Ovalene	9	$\{e_3, e_9, e_{15}, e_{16}, e_{20}, e_{31}, e_{38}, e_{40}\}$	9	$\{e_1, e_6, e_7, e_{12}, e_{15}, e_{20}, e_{21}, e_{31}, e_{39}\}$	9	-

Figure 5. The parameters setting of GA in our algorithm are detailed in Table 2.

Table 2: The GA parameters setup.

Parameters	Value
Population size	25
Mutation probability	0.3
Number of iteration	100
Crossover probability	0.8

5. Conclusion

The development of the GFS concept in graph theory has been significantly influenced by applications in chemistry. It would be useful to precisely determine the GFN for perfect matchings in a graph G. However, this is challenging to calculate, as finding all perfect matchings is generally considered NP-hard. As discussed in Section 3, there have been efforts to find the GFN for perfect matchings in specific

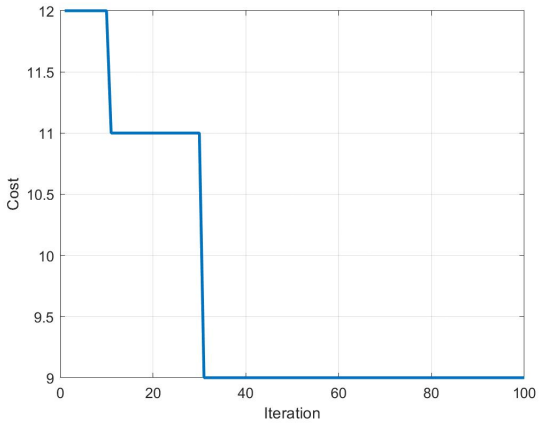

Figure 5: Performance plot for Ovalene with 100 generations.

graphs, including hexagonal systems. The algorithm described here can be applied to various bipartite graphs, particularly hexagonal systems. Additionally, the results of the genetic algorithm (GA) improve the results presented in the theorems mentioned in Section 3 for finding the GFN.

Conflicts of Interest. The authors declare that they have no conflicts of interest regarding the publication of this article.

Acknowledgments. Research of the first author, Sara Oskoueian, and the second author, Mostafa Tavakoli, supported in part by the Ferdowsi University of Mashhad.

References

[1] R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors*, John Wiley & Sons, (2008).

[2] N. Trinajstic, *Chemical Graph Theory*, CRC press, 2018.

[3] F. Harary, D. J. Klein, and T. P. Živkovič, Graphical properties of polyhexes: perfect matching vector and forcing, *J. Math. Chem.* **6** (1991) 295 – 306, https://doi.org/10.1007/BF01192587.

[4] D. J. Klein and M. Randić, Innate degree of freedom of a graph, *J. Comput. Chem.* **8** (1987) 516 – 521, https://doi.org/10.1002/jcc.540080432.

[5] D. Vukičević and M. Randić, On kekulé structures of buckminsterfullerene, *Chem. Phys. lett.* **401** (2005) 446 − 450, https://doi.org/10.1016/j.cplett.2004.11.098.

[6] D. Vukičević, S. Zhao, J. Sedlar, S.-J. Xu and T. Došlić, Global forcing number for maximal matchings, *Discrete Math.* **341** (2018) 801 − 809, https://doi.org/10.1016/j.disc.2017.12.002.

[7] L. Lovász and M. D. Plummer, *Matching Theory*, American Mathematical Soc. **367** 2009.

[8] D. Vukičević and J. Sedlar, Total forcing number of the triangular grid, *Math. Commun.* **9** (2004) 169 − 179.

[9] D. Vukicevic and T. Došlić, Global forcing number of grid graphs, *Australas. J. Combin.* **38** (2007) 47 − 62.

[10] S. Klavžar, M. Tavakoli and G. Abrishami, Global forcing number for maximal matchings in corona products, *Aequationes Math.* **96** (2022) 997 − 1005, https://doi.org/10.1007/s00010-022-00869-3.

[11] T. Došlić, Global forcing number of benzenoid graphs, *J. Math. Chem.* **41** (2007) 217 − 229, https://doi.org/10.1007/s10910-006-9056-2.

[12] H. Zhang and J. Cai, On the global forcing number of hexagonal systems, *Discrete Appl. Math.* **162** (2014) 334 − 347, https://doi.org/10.1016/j.dam.2013.08.020.

[13] M. Sinan Oz and I. Naci Cangul, Computing the Hosoya and the Merrifield-Simmons indices of two special benzenoid systems, *Iranian J. Math. Chem.* **12** (2021) 161 − 174, https://doi.org/10.22052/IJMC.2021.243008.1580.

[14] Z. Yarahmadi, A note on the first geometric-arithmetic index of hexagonal systems and phenylenes, *Iranian J. Math. Chem.* **2** (2011) 101 − 108, https://doi.org/10.22052/IJMC.2011.5217.

[15] J. Cai and H. Zhang, Global forcing number of some chemical graphs, *MATCH Commun. Math. Comput. Chem.* **67** (2012) 289 − 312.

[16] P. Adams, M. Mahdian and E. S. Mahmoodian, On the forced matching numbers of bipartite graphs, *Discrete Math.* **281** (2004) 1 − 12, https://doi.org/10.1016/j.disc.2002.10.002.

[17] L. G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979) 410 − 421, https://doi.org/10.1137/0208032.

[18] K. Fukuda and T. Matsui, Finding all the perfect matchings in bipartite graphs, *Appl. Math. Lett.* **7** (1994) 15 − 18, https://doi.org/10.1016/0893-9659(94)90045-0.

[19] S. Oskoueian, M. Tavakoli and N. Sabeghi, Finding the minimum global forcing set for maximal and perfect matchings using a greedy algorithm, Submitted.

Sara Oskoueian
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
Ferdowsi University of Mashhad,
Mashhad, I. R. Iran
e-mail: sara.oskoueian@hotmail.com

Mostafa Tavakoli
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
Ferdowsi University of Mashhad,
Mashhad, I. R. Iran
e-mail: m_tavakoli@um.ac.ir

Narjes Sabeghi
Department of Mathematics,
Faculty of Basic Sciences,
Velayat University,
Iranshahr, I. R. Iran
e-mail: sabeghinarjes@gmail.com