Les Cahiers du GERAD ISSN: 0711–2440

Ascent – Descent Variable Neighborhood Decomposition Search for community detection by modularity maximization

D. Džamić, D. Aloise,
N. Mladenović

G–2016–35

May 2016

Cette version est mise à votre disposition conformément à la politique de libre accès aux publications des organismes subventionnaires canadiens et québécois.

Avant de citer ce rapport, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2016-35) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

This version is available to you under the open access policy of Canadian and Quebec funding agencies.

Before citing this report, please visit our website (https://www.gerad.ca/en/papers/G-2016-35) to update your reference data, if it has been published in a scientific journal.

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2016
– Bibliothèque et Archives Canada, 2016

Legal deposit – Bibliothèque et Archives nationales du Québec, 2016
– Library and Archives Canada, 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Ascent – Descent Variable Neighborhood Decomposition Search for community detection by modularity maximization

Dušan Džamić [a]

Daniel Aloise [b]

Nenad Mladenović [c]

Faculty of organizational sciences, University of Belgrades[a]

Departamento de Engenharia de Computação e Automação, Universidade Federal do Rio Grande do Norte[b]

GERAD & Mathematical Institute, Serbian Academy of Sciences[c]

dusan.dzamic@fon.bg.ac.rs
aloise@dca.ufrn.br
nenad@mi.sanu.ac.rs

May 2016

Les Cahiers du GERAD

G–2016–35

Copyright © 2016 GERAD

Abstract: In this paper we propose a new variant of the Variable Neighborhood Decomposition Search (VNDS) heuristic for solving global optimization problems and apply it in detecting clusters in complex community networks. The most used criterion for detecting comunities in networks, despite some recent criticism, is modularity maximization. Experimental results show that the proposed new heuristic outperforms the best algorithms found in the literature.

Key Words: clustering, community detection, modularity maximization, variable neighborhood search, decomposition

---

Acknowledgments: This research was partially supported by CNPq-Brazil grants 308887/2014-0 and 400350/ 2014-9, and Serbian Ministry of Education and Science grant 174010.

1 Introduction

Many systems in real world exist in the form of networks, such as biological networks, social networks, the World Wide Web, transportation networks etc., which are also called complex networks. Complex networks have been one of the most popular research areas in recent years due to its applicability to a wide spectrum of disciplines [1, 2]. Many networks display community structure: the nodes in networks are often found to cluster into tightly knit groups with a high density of within group edges and a lower density of between group edges [2]. The goal is to uncover the real community structure in complex networks. The research on complex network clustering is of fundamental importance for both the theoretical significance and practical applications on analyzing network topology, comprehending network function, unfolding hidden law of network and forecasting network activities, which have been used in many areas, such as organization management and recognition, biological network analysis, web community mining, topic based web document clustering, search engine, link prediction, etc. [3].

Community detection in networks (graph) refers to finding subset of vertices in a graph (called clusters, or communities, or modules) which are densely connected among themselves but less connected with vertices in other communities. There are many ways to formalize this idea. One way to identify communities is to specify an objective function to minimize or maximize. Various objective functions have been proposed such as multiway cut [4], normalized cut [5, 6], minimum-sum-of-squares [7, 8], ratio cut [9] and modularity [10]. One of the most popular initially proposed by Girvan and Newman in 2002 is to maximize the modularity, which represents the fraction of edges within clusters minus the expected fraction of such edges in a random graph with the same degree distribution.

In this paper we propose a new variant of Variable neighborhood decomposition search (VNDS) algorithm, that we call Ascent - Descent VNDS (ADVNDS) (for basic information regarding VNS and its variants see its recent surveys [11, 12]). The basic idea of ADVNDS is to accept not only better (ascent) but also worse (descent) solution with some probability. For that purpose, we choose an exponential probability distribution. Its single parameter value is a function of difference in two successive modularity values devided by the number of unsuccesful moves in all previous iterations. Thus, the smaller difference in modularity values and the larger number of unsuccessfull iterations imply the larger probability to move. In the ADVNDS framework, the local search in the whole solution space might be performed even if the subproblem resolution does not yield an improvement. Note that usually used Skewed (ascent - descent) VNS (SVNS) acceptance criterion [11] takes into account the distances between two solutions, and not the number of unsuccesful moves in previous iterations. Moreover, no distribution function is used in SVNS.

Although the ADVNDS idea is general, we applied it just for solving Modularity maximization problem (MMP). Our heuristic for solving MMP may also be seen as a modification of the current state-of-the-art, i.e., strictly ascent VNDS based heuristic proposed in [13]. Beside including the possibility of accepting non improving (descent) solutions, in this paper we also propose a new neighborhood structure for solving sub-problem and use it in the shaking phase of the algorithm. Our ADVNDS was tested on 21 instances from the 10th DIMACS Implementation Chalenge. On 13 instances where optimal solutions were not known we were able to improve best known ones on 9 instances and be equal on 4.

The paper is organized as follows. In Section 2 we present formulations of the problem. In Section 3, we give steps of ADVNDS algorithm and details regarding its components. In Section 4, a computational study is performed comparing our ADVNDS with the two best methods from the literature. Finally, some concluding remarks are given in Section 5.

2 Modularity

Let $G = (V, E)$ be a graph, or network, with vertex set V of cardinality n and edge set E of cardinality m. The following precise definition of modularity is given in [10]:

$$Q = \sum_{c \in C} (a_c - e_c)$$

where C is partition of the graph into disjoint communities, (i.e., into sets such that each vertex belongs to exactly one set), a_c is the fraction of all edges that lie within community $c \in C$, and e_c is the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random. Modularity can then be written equivalently as:

$$Q = \sum_{c \in C} \left(\frac{m_c}{m} - \frac{K_c^2}{4m^2} \right)$$

where m_c denotes the number of edges in community $c \in C$ (i.e., which belong to the subgraph induced by the vertex set V_c of that community), and K_c denotes the sum of degrees of the vertices in community $c \in C$. A good partition of the vertex set of a graph into communities is obtained when the modularity is maximized. Using the second definition of modularity, the corresponding problem can be expressed by means of a convex Mixed Integer Quadratic Programming (MIQP) formulation [14]. Numerous heuristics have been proposed to maximize modularity such as simulated annealing, genetic algorithm, spectral clustering, dynamical clustering, and quantum mechanics [15].

3 Ascent - Descent Variable Neighborhood Decomposition Search

Variable Neighborhood Search (VNS) metaheuristic [16] combines local search with systematic changes of neighborhood in the descent and escape from local optimum phases. Since its inception, VNS has undergone many developments and has been applied in numerous fields. For small and medium size problems, descent local searches are very fast and general heuristics usually use much longer CPU times. But, for very large problem instances, local search algorithms often require substantial amounts of running time. One way to reduce the running time is using decomposition search which explores the structure of the problem concentrating on small parts of it.

Variable Neighborhood Decomposition Search (VNDS) [17] is two level VNS variant where the outer level is responsible for managing subproblems while they are solved (usually by VNS) in the inner level. The maximum size of the subproblems is controlled by the parameter s_{max}. If that parameter is equal to the dimension of the problem, then the subproblem is equivalent to the original one. In addition, each time a subproblem solution is improved, some local search is performed in the whole solution space in order to tackle a phenomenon called the *boundary effect*, in which modifications to the solution of a subproblem might lead to improvements in the overall solution (cf. [18])

We propose a modification to VNDS algorithm described in [13]. Our new variant of VNDS, called Ascent − Descent VNDS (ADVNDS) allows local search improvement even for non improving incumbent solutions. A non improving solution is refined by local search in the whole solution space with a probability that increases with the number of unsuccessful moves. For that purpose, we use an exponential distribution with parameter $1/a$. In the sequel, parameter a returns to its initial value.

3.1 Shaking and local search components

The shaking component our algorithm explores different types of neighborhoods, each of them focusing on different aspects and properties of the solutions to the MMP:

1. All the vertices in a community are made singleton communities, with an example given in Figure 1 (a).
2. Splits a community into two equal parts. Vertices are assigned to each part randomly, with an example given in Figure 1 (b).
3. Move each vertex of a community to one of its neighborhoods or to a new community, with an example given in Figure 1 (c).
4. Splits a community into two equal parts and then merges two or more communities into a single one, with an example given in Figure 1 (d).

The first three types of neighborhoods are the same as in [13], and the fourth type is composition of a second type and the neighborhood which is obtained by merging two or more communities into a single one.

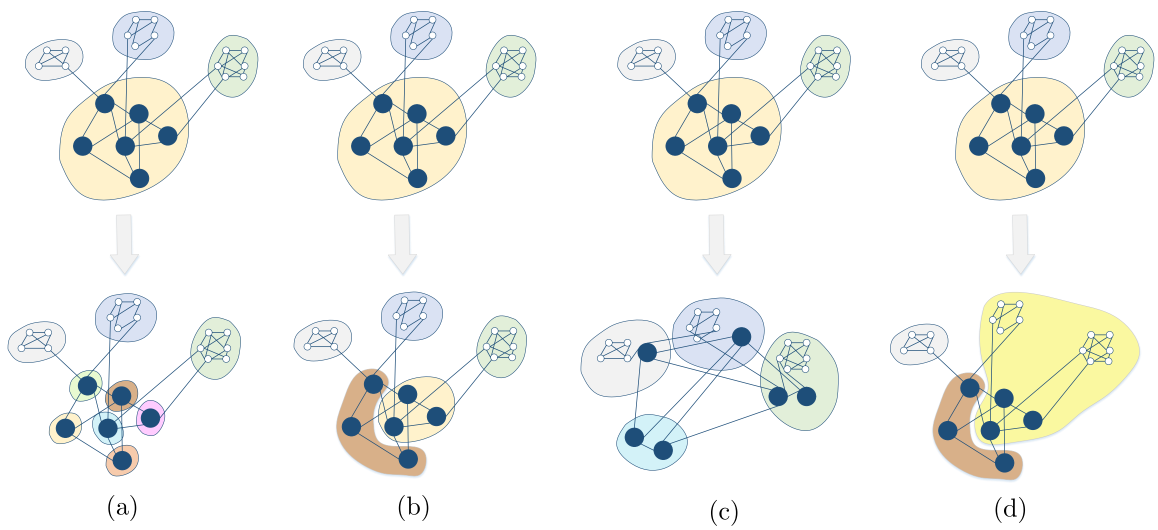

Figure 1: Different types of neighborhoods used in the shaking step

The local search component is exactly the same as the one used in [13].

3.2 ADVNDS schema

The decomposition procedure uses the shaking and local search algorithms presented in the previous subsection, within a variable neighborhood schema. For that purposes local search receives two input parameters: x_{temp} - the solution to be improved, and S - the solution space on which an improvement will be searched. Subproblem (in this case subgraph S) is constructed with $s-1$ randomly selected neighboring communities from one solution of the graph G. An example of decomposition, where cluster marked with * presents the main cluster, is given in Figure 2. The increase of the parameter a is made multiplying the initial value $a_{initial}$ by a factor r (that in fact increases exponentially with respect to the number of unsuccessful moves). The pseudo-code of the ADVNDS based heuristic is given in Algorithm 1.

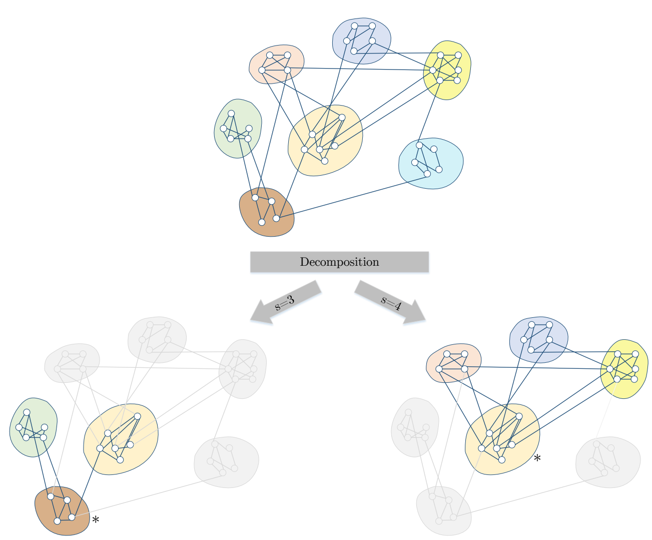

Figure 2: Decomposition

Initial solution $x_{current}$ is obtained in the first two steps from a random solution followed by local search; it is claimed to be the best (x_{best}) in step 3. The initial value of variables are defined at step 4: s - the number of communities used in the decomposition, and a - the parameter of exponential distribution used in acceptance criterion. The central part of the ADVNDS algorithm consists of the loop executed in lines 5-23. In step 6, subproblem S is constructed with s randomly selected neighboring communities from $x_{current}$. In step 7 a neighborhood type k is randomly selected for shaking phase. Then in step 8, the selected shaking routine is performed on the solution $x_{current}$ in the search space defined by subproblem S. After shaking, local search is applied over x_{temp} in line 9 only in the space defined by the current subproblem S. The condition in line 11 defines that local search is performed in x_{temp} in step 12 if it is better than $x_{current}$ or if it is worse conditioned to a specific probability. This probability is larger the smaller the difference in modularity values between x_{temp} and $x_{current}$, and the larger the number of unsuccessfull iterations. In steps 13–15, the best solution is updated if $x_{current}$ is better then x_{best}. Then in step 16, the variables s and a, that control the current decomposition size and the probability of accepting a non improving solution for local search are set to 1 and $a_{initial}$, respectively. Otherwise, if x_{temp} is not accepted for local search, the size of the decomposition is increased by one and parameter a is increased by multiplying it by the given parameter r in line 18. The value of parameter s is reset to 1 in step 20 if it exceeds the minimum value between a given parameter s_{max} and the number of clusters in $x_{current}$ solution. Finally, after the main loop terminates, the best found solution by ADNDS is claimed to be x_{best}.

Algorithm 1: ADVNDS scheme

input : Graph G, Parameters $(a_{initial}, r, s_{max})$
output: membership vector x_{best}

1 $x_{current} \leftarrow$ RandomSolution();
2 $x_{current} \leftarrow$ LocalSearch($x_{current}$, G);
3 $x_{best} \leftarrow x_{current}$;
4 $s \leftarrow 1$; $a \leftarrow a_{initial}$;
5 **while** *maximum allowed time not attained* **do**
6     $S \leftarrow$ Decomposition($x_{current}$, s);
7   Select randomly type of neighborhood $k \in \{1, 2, 3, 4\}$;
8     $x_{temp} \leftarrow$ Shaking($x_{current}, k, S$);
9     $x_{temp} \leftarrow$ LocalSearch(x_{temp}, S);
10     $d \leftarrow$ modularity(x_{temp})-modularity($x_{current}$);
11   **if** $d \geq 0$ *or* $e^{d/a} > R(0, 1)$ **then**
12         $x_{current} \leftarrow$ LocalSearch(x_{temp}, G);
13   **if** modularity($x_{current}$) $>$ modularity(x_{best}) **then**
14             $x_{best} \leftarrow x_{current}$;
15   **end**
16         $s \leftarrow 1$; $a \leftarrow a_{initial}$;
17   **else**
18         $s \leftarrow s + 1$; $a \leftarrow a \times r$;
19   **if** $s >$ min(NumberOfCommunities($x_{current}$), s_{max},) **then**
20             $s \leftarrow 1$;
21   **end**
22   **end**
23 **end**

4 Computational results

The proposed algorithm is programmed in C++, and compiled with GNU g++ on Intel Core i5-2400 CPU @ 3.10GHz $\times$ 4 and 4GB of RAM. The instances are taken from the Clustering chapter of the 10th DIMACS Implementation Challenge.[1] The collection of benchmark inputs of the 10th DIMACS Implementation Challenge includes both synthetic and real-world data. All graphs are undirected. Formerly directed instances

[1] All instances can be downloaded from *http://www.cc.gatech.edu/dimacs10/downloads.shtml*

were symmetrized by making every directed edge undirected. While this procedure necessarily loses information in a number of real-world applications, it appeared to be necessary since most existing software libraries can handle undirected graphs only [19].

As noted earlier, ADVNDS algorithm stops after running for t_{max} (a parameter) seconds. In the first category (C_1) of instances, t_{max} is set to 180 seconds (3 minutes) and for all instances in the second category (C_2), t_{max} is set to 3600 seconds (1 hour). Other parameter values of the ADVNDS algorithm are set as follows: $a_{initial} = 0.001$, $r = 1.0001$ and $s_{max} = 15$. The algorithm has not been tested on very large instances for which available memory was too small.

Table 1 presents the computational results obtained in ten independent runs of proposed ADVNDS algorithm. The first and second columns refers to the category and name of the instance. Third and fourth columns refer to the average modularity values Q_{avg} and the average computing times t_{avg}, respectively. The fifth column, denoted by w_{avg}, refers to the average number of times that non improving solutions were refined by local search. The sixth and seventh columns present the modularity value Q_{best} and the corresponding number of clusters C_{best} respectively, that correspond to the best solution found. Finally, the eighth column reports the % standard deviation.

Table 1: Experimental results - ADVNDS algorithm

category	instance name	Q_{avg}	t_{avg}	w_{avg}	Q_{best}	C_{best}	% deviation
	adjnoun	0,31336747	0,25	0	0,31336747	7	0,000
	celegans_metabolic	0,45324822	1,66	0	0,45324822	9	0,000
	celegansneural	0,50378240	0,15	0	0,50378240	6	0,000
	chesapeake	0,26579585	0,33	0	0,26579585	3	0,000
	dolphins	0,52851944	0,00	0	0,52851944	5	0,000
C_1	football	0,60456956	1,08	0	0,60456956	10	0,000
	jazz	0,44514385	3,05	0	0,44514385	4	0,000
	karate	0,41978961	0,25	0	0,41978961	4	0,000
	lesmis	0,56668798	0,00	0	0,56668798	6	0,000
	polbooks	0,52723659	0,00	0	0,52723659	5	0,000
	as-22july06	0,67789771	1795,12	8	0,67838109	38	0,025
	astro-ph	0,74502039	2088,75	93	0,74524637	1051	0,013
	cond-mat-2003	0,77843785	3428,28	75	0,77924749	1673	0,055
	cond-mat-2005	0,74607478	3412,24	11	0,74718099	1903	0,054
	cond-mat	0,85387268	1842,00	266	0,85414744	1244	0,020
C_2	email	0,58282897	29,73	12	0,58282897	10	0,000
	hep-th	0,85792731	1786,35	1495	0,85808847	1366	0,011
	netscience	0,95989999	0,12	0	0,95989999	407	0,000
	PGPgiantcompo	0,88639112	1800,00	119	0,88664690	103	0,016
	polblogs	0,42710511	0,12	0	0,42710511	278	0,000
	power	0,94086519	1643,94	283	0,94097423	42	0,005

Proposed ADVNDS algorithm has reached the optimal solution for all instances from C_1 set, except for "celegans_metabolic" for which the optimal solution is not known. It should be noted that our ADVNDS based heuristic improve the best known solutions [2] to all 13 hard instances from the 10th DIMACS Implementation Challenge [19] for which the optimal solutions are not known.

In Table 2 we compare three methods, but only on those 13 hard instances for which the optimal solutions are not known. Besides the previous version of VNDS proposed in [13], we also include in our comparison the so-called CGGCi scheme [20] – both heuristics are considered as the current state-of-the-art. Main idea of Core Groups Graph Clustering Scheme(CGGC) scheme is to combine several weak classifiers into a strong classifier. From the maximal overlap of clusterings computed by weak classifiers, the algorithm searches for a solution with high quality [19, 20]. The first column of Table 2 refers to the instance name. The second, fourth and sixth columns refer to the best modularity values Q_{best}, obtained by CGGCi, VNDS and ADVNDS heuristics, respectively. The third, fifth, and seventh columns give the rank of corresponding three methods among 15 participating teams in Challenge.

[2]Results can be downloaded from *http://www.cc.gatech.edu/dimacs10/results/Modularity_Quality/*

Table 2: Comparison on the Clustering instances from the 10th DIMACS Implementation Challenge

instance / algorithm	CGGCi		VNDS		ADVNDS	
	Q_{best}	rank	Q_{best}	rank	Q_{best}	rank
celegans metabolic	0,45214778	#2	0,45324822	#1	0,45324822	#1
as-22july06	0,67826683	#2	0,67757548	#3	0,67838109	#1
astro-ph	0,74384869	#3	0,74462084	#2	0,74524637	#1
cond-mat-2003	0.77868235	#2	0,77671700	#3	0,77924749	#1
cond-mat-2005	0,74625408	#2	0,74506494	#3	0,74718099	#1
cond-mat	0.85309726	#3	0,85340200	#2	0,85414744	#1
e-mail	0,58187658	#2	0,58282897	#1	0,58282897	#1
hep-th	0.85655364	#3	0,85769200	#2	0,85808847	#1
netscience	0.95989999	#1	0,95989999	#1	0,95989999	#1
PGPgiantcompo	0,88656423	#2	0,88608182	#3	0,88664690	#1
polblogs	0,42709696	#2	0,42710511	#1	0,42710511	#1
power	0,94033178	#3	0,94085057	#2	0,94097423	#1
kron g500-simple-logn16	0,06372656	#4	0,06505580	#2	0,06566119	#1

It appears that our new proposed ADVNDS based heuristic is ranked as the first on all 13 instances. For nine of them, ADVNDS are strictly better whereas for four instances the quality of VNDS and ADVNDS are equal; in one instance, all three methods reported the same best known solution (for "netscience"). Comparing VNDS and CGGCi, in 8 out of 13 instances, the VNDS outperforms CGGCi. Therefore, the heuristic proposed in this paper may be seen as the new state-of-the art heuristic for solving MMP.

5 Conclusions

Community detection in complex networks has been one of the most popular research areas in recent years due to its applicability to the wide scale of disciplines. In this paper we proposed Ascent – Descent Variable Neighborhood Decomposition Search (ADVNDS) for solving the modularity maximization problem (MMP). The new ADVNDS involves a new acceptence criterion and a new neighborhood structure for the MMP. We evaluate the performance of our ADVNDS algorithm on a subset of clustering instances from the 10th DIMACS Implementation Challenge. Experimental results show that the proposed ADVNDS algorithm outperforms the best algorithms found in the literature.

The acceptance criterion proposed here within VNDS, which decides whether to perform the local search for all space (after solving subproblem), may obviously be used within VNS as well in future works with the objective of solving any combinatorial optimization problem.

References

[1] D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks, Nature 393 (6684) (1998) 440–442.

[2] M. Girvan, M. E. Newman, Community structure in social and biological networks, Proceedings of the national academy of sciences 99 (12) (2002) 7821–7826.

[3] M. A. Porter, J.-P. Onnela, P. J. Mucha, Communities in networks, Notices of the AMS 56 (9) (2009) 1082–1097.

[4] O. Goldschmidt, D. S. Hochbaum, Polynomial algorithm for the k-cut problem, Proceedings of 29th Annual Symposium on Foundations of Computer Science (October 1988), 444–451.

[5] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[6] P. Hansen, M. Ruiz, D. Aloise, A vns heuristic for escaping local extrema entrapment in normalized cut clustering, Pattern Recognition 45 (12) (2012) 4337–4345.

[7] E. Carrizosa, N. Mladenović, R. Todosijević, Sum-of-squares clustering on networks , Yugoslav Journal of Operations research , 21 (2011) 157–161.

[8] E. Carrizosa, N. Mladenović, R. Todosijević, Variable neighborhood search for minimum sum-of-squares clustering on networks, European Journal of Operational Research 230 (2) (2013) 356–363.

[9] C. J. Alpert, S.-Z. Yao, Spectral partitioning: the more eigenvectors, the better, in: Proceedings of the 32nd annual ACM/IEEE Design Automation Conference, ACM (1995) 195–200.

[10] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical review E 69 (2) (2004) 026113.

[11] P. Hansen, N. Mladenović, J. A. M. Pérez, Variable neighbourhood search: methods and applications, 4OR 6 (4) (2008) 319–360.

[12] S. Hanafi, J. Lazić, N. Mladenović, C. Wilbaut, I. Crevits, New variable neighbourhood search based 0-1 mip heuristics, Yugoslav Journal of Operations Research 25 (3) (2015) 343–360.

[13] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, M. Ruiz, Modularity maximization in networks by variable neighborhood search., Graph partitioning and graph clustering, American Mathematical Soc., 588 (2012) 113–127.

[14] G. Xu, S. Tsoka, L. G. Papageorgiou, Finding community structures in complex networks using mixed integer optimisation, The European Physical Journal B 60 (2) (2007) 231–239.

[15] S. Fortunato, Community detection in graphs, Physics reports 486 (3) (2010) 75–174.

[16] N. Mladenović, P. Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097–1100.

[17] P. Hansen, N. Mladenović and D. Perez-Brito, Variable neighborhood decomposition search, Journal of Heuristics 7 (4) (2001) 335–350.

[18] D. Urosević, J. Brimberg, N. Mladenovic, Variable neighborhood decomposition search for the edge weighted k-cardinality tree problem, Computers and Operations Research 31 (2004) 1205–1213.

[19] D. A. Bader, H. Meyerhenke, P. Sanders, D. Wagner, Graph partitioning and graph clustering, American Mathematical Soc., 588 (2013).

[20] M. Ovelgönne, A. Geyer-Schulz, An ensemble learning strategy for graph clustering, Graph partitioning and graph clustering, American Mathematical Soc., 588 (2012) 187.

Dušan Džamić is a PhD student at the Department for Informatics and Computer Science, Faculty of Mathematics, University of Belgrade. At the same Faculty, he graduated as a Bachelor of Mathematics and Computer Science in 2013. He defended his master thesis in 2014. He is working as a teaching assistant in the Department for Mathematics at the Faculty of Organizational Sciences, University of Belgrade, and also as a researcher at the Mathematical Institute SANU in Belgrade. His research is mainly focused on Mathematical Programming Methods and Optimization.

Daniel Aloise received his degree in computer engineering from the Universidade Federal do Rio Grande do Norte, Brazil, in 2002, and his Ph.D. degree from Polytechnique-Montral in 2009. In 2010, Daniel Aloise received the Classification Society Distinguished Dissertation Award for his Ph.D. thesis on exact methods for minimum sum-of-squares clustering, under the supervision of Pierre Hansen. He is currently assistant professor in the Department of Computer Engineering and Automation at the Universidade Federal do Rio Grande do Norte, Brazil. His research interests are mainly focused on classification/clustering methods based on optimization tools.

Professor Nenad Mladenović is currently at Mathematical Institute, Belgrade and LAMIH, University of Valenciennes, France. He was also teaching in United Kingdom, Canada and Belgium. He had visiting professor positions in more than dozen Universities all around the globe. He was keynote speaker on several international conferences in Russia, Brazil, Spain, Portugal, Canada, France, etc. He is editor in chief of one and serves as editorial member of 11 international journals. He was editor of 7 special issues as well. He has published more than 150 research papers (according to Web of Knowledge) in leading OR journals, such as Management Science, Operations Research, SIAM J on Scientific Computing, Pattern Recognition, Mathematical Programming, European J of Operations Research, Computers and OR, OR Letters, INFORMS J on Computing, Annals of OR, Discrete Applied Mathematics, etc. Mladenovic is highly cited researcher. He has more than 11,500 citations according to Google scholar (4654 - Scopus). His paper with P Hansen was selected among 30 most influential published in EJOR in 30 years history of EJOR. His current (H, G) research indices are (24, 55) (Web of Knowledge) and (42, 101) (Google Scholar). He is a member of Academia Europea (The Academy of Europe), Academy of Nonlinear Sciences, Serbian Scientific Society, acts as a President of National (Serbian) Society of Industrial and Applied Mathematics, under his supervision more than ten doctoral students completed their PhD in several countries, etc.