

Improving the Louvain algorithm for community detection with modularity maximization

Olivier Gach^{1,2} and Jin-Kao Hao²

¹ LIUM, Université du Maine, Av. O. Messiaen, 72085 Le Mans, France
olivier.gach@univ-lemans.fr

² LERIA, Université d'Angers, 2 Bd Lavoisier, 49045 Angers Cedex 01, France
jin-kao.hao@univ-angers.fr

Abstract. This paper presents an enhancement of the well-known Louvain algorithm for community detection with modularity maximization which was introduced in [16]. The Louvain algorithm is a partial multi-level method which applies the vertex mover heuristic to a series of coarsened graphs. The Louvain+ algorithm proposed in this paper generalizes the Louvain algorithm by including a uncoarsening phase, leading to a full multi-level method. Experiments on a set of popular complex networks show the benefits induced by the proposed Louvain+ algorithm.

Keywords: Clustering; Optimization over networks; Heuristics.

1 Introduction

Complex networks are a graph-based model which is very useful to represent connections and interactions of the underlying entities in a real networked system such as social [1], biological [2], and technological networks [3]. A vertex of the complex network represents an object of the real system while an edge symbolizes an interaction between two objects. For example in a social network, a vertex corresponds to a particular member of the network and an edge represents a relationship between two members. Complex networks typically display non-trivial structural and functional properties which impact the dynamics of processes applied to the network [4]. Analysis and synthesis of complex networks help discover these specific features, understand the dynamics of the networks and represent a real challenge for research [5,6].

A complex network may be characterized by a community structure. Vertices of a community are grouped to be highly interconnected while different communities are loosely associated with each other. Community is also called cluster or still module [7]. All the communities of a network form a clustering. In terms of graph theory, a clustering can be defined as a partition of the vertices of the underlying graph into disjoint subsets, each subset representing a community.

Intuitively, a community is a cohesive group of vertices that are more connected to each other than to the vertices in other communities. To quantify the quality of a given community and more generally a clustering, modularity is certainly the most popular measure [8]. Under this quality measure, the problem

of community detection becomes a pure combinatorial optimization problem. Formally, the modularity measure can be stated as follows.

Given a weighted graph $G = (V, E, w)$ where w is a weighting function, i.e., $w : V \times V \mapsto \mathbb{R}$ such that for all $\{u, v\} \in E$, $w(\{u, v\}) \neq 0$, and for all $\{u, v\} \notin E$, $w(\{u, v\}) = 0$. Let $X \subseteq V$ and $Y \subseteq V$ be two vertex subsets, $W(X, Y)$ the weight sum of the edges linking X and Y , i.e., $W(X, Y) = \sum_{u \in X, v \in Y} w(\{u, v\})$ (in this formula, each edge is counted twice). The modularity of a clustering with k communities $C = \{c_1, c_2, \dots, c_k\}$ ($\forall i \in \{1, 2, \dots, k\}, c_i \subset V$ and $c_i \neq \emptyset$; $\cup_{i=1}^k c_i = V$; $\forall i, j \in \{1, 2, \dots, k\}, c_i \cap c_j = \emptyset$) is given by:

$$Q(C) = \sum_{i=1}^k \left[\frac{W(c_i, c_i)}{W(V, V)} - \left(\frac{d_i}{W(V, V)} \right)^2 \right] \quad (1)$$

where d_i is the sum of the degrees of the vertices of community c_i , i.e., $d_i = \sum_{v \in c_i} \text{deg}(v)$ with $\text{deg}(v)$ being the degree of vertex v .

It is easy to show that Q belongs to the interval $[-0.5, 1]$. A clustering with a small Q value close to -0.5 implies the absence of real communities. A large Q value close to 1 indicates a good clustering containing highly cohesive communities. The trivial clustering with a single cluster has a Q value of 0.

Community detection with modularity is an important research topic and has a number of concrete applications [9]. In addition to its practical interest, community detection is also notable for its difficulty from a computational point of view. Indeed, the problem is known to be NP-hard [10] and constitutes thus a real challenge for optimization methods.

A number of heuristic algorithms have been proposed recently in the literature for community detection with the modularity measure. These algorithms follow three general solution approaches. First, greedy agglomeration algorithms like [11,12] iteratively merge two clusters that yield a clustering by following a greedy criterion. Second, local optimization algorithms like [13,14,15] improve progressively the solution quality by transitioning from a clustering to another clustering (often of better quality) by applying a move operator. The quality of such an algorithm depends strongly (among other things) on the move operator(s) employed. Third, hybrid algorithms like [16,17,18,19] combine several search strategies (e.g., greedy and multi-level methods) in order to take advantage of the underlying methods. Among the existing community detection algorithms, the Louvain algorithm presented in [16] (see next section) is among the most popular methods.

The Louvain algorithm belongs to the hybrid approach and can be compared to the general multi-level framework which requires both a coarsening and uncoarsening phases [20]. The coarsening phase reduces the size of a graph at each level by grouping several vertices of the original graph into a single vertex. The uncoarsening phase does the inverse by unfolding the vertices of the coarsen graph and then applying a refinement (optimization) procedure. While Louvain algorithm does use a coarsening phase, it omits the uncoarsening phase. However, from an optimization point of view, it is known that the uncoarsening phase within the multi-level framework is useful to further improve the quality

of the solution (see the example given in Sect. 2). This paper aims to extend the Louvain algorithm by including a uncoarsening phase, making the algorithm a full multi-level method. Experiments on a set of popular complex networks show the benefits induced by the proposed Louvain+ algorithm.

2 The Louvain Algorithm

The Louvain algorithm presented by Blondel et al. [16] operates on multiple levels of graphs, applying the vertex mover (VM) procedure on each level to improve the modularity. In this Section, we recall the two key elements of the methods: the VM procedure and the coarsening phase.

2.1 Vertex mover procedure

For a given graph where each vertex represents a community, one iteration of VM explores all the vertices of the graph in a random order. For each vertex, one examines all the possible moves to a neighbor community with an increased modularity. The move giving the largest increase is chosen and realized. At the end of an iteration, all the vertices of the current graph are processed. One proceeds with a new iteration if at least one vertex has migrated. To ensure that the vertices are examined in a purely random order during each iteration, the exploration of the vertices follows a random permutation of $\{1, 2, \dots, n\}$ which is generated at the beginning once and for all. The procedure stops if no vertex has migrated when all the vertices have been examined. Another possible stop criterion is a minimum modularity gain: if the total gain obtained in one iteration is lower than the minimum gain required, the algorithm stops.

2.2 Coarsening phase

The coarsening phase of the Louvain algorithm starts with the initial graph G (call it level 0 graph G^0) and produces a hierarchy of coarser graphs G^1, G^2, \dots of decreasing orders. We use $G^l = (V^l, E^l, w^l)$ to denote the graph of level l . From the graph G^0 and the initial trivial clustering where each vertex of G^0 forms a singleton community, the VM heuristic is applied to generate an improved clustering C^0 . Then the graph G^1 of level 1 is created such that a vertex is introduced for each community of C^0 and an edge between two vertices is defined if they represent two neighboring communities in C^0 . Now the VM heuristic is applied to the new graph G^1 with the clustering of singleton communities. This process continues and stops at some level L if the VM heuristic can not improve the initial clustering with singleton communities of G^L .

Formally, the generation of the coarsened graph G^{l+1} from (G^l, C^l) are achieved according to the following steps [21].

1. A vertex in G^{l+1} corresponds to a community of clustering C^l and vice versa. Given a community c of clustering C^l , let $T^{l+1}(c)$ denote the corresponding vertex in G^{l+1} .

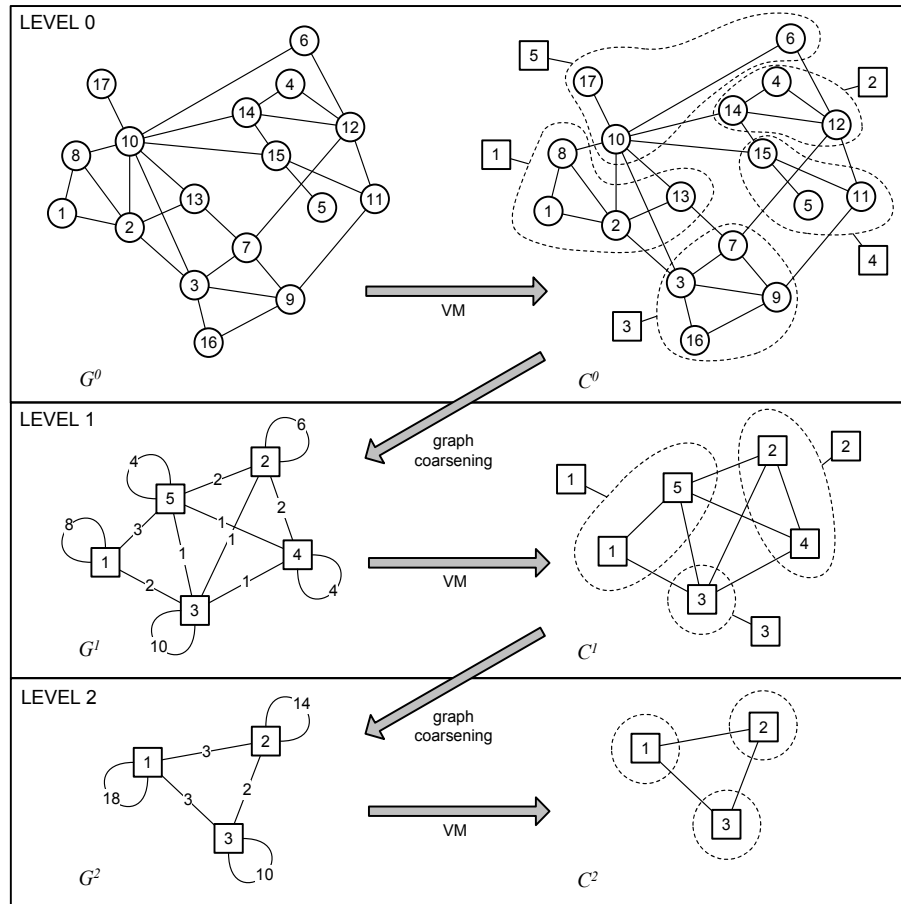


Fig. 1. Illustration of the Louvain algorithm. The initial graph G^0 contains 17 vertices and 29 edges. A first application of VM procedure to the trivial clustering of singleton communities gives the clustering C^0 composed of 5 communities. Then the coarsen graph G^1 is built with weighted edges and loops (squares in this graph represents communities from a lower level). The VM procedure is applied to the new graph G^1 to obtain the clustering C^1 with 3 communities. At level 2, the application of the VM procedure to the initial clustering of singleton communities does not change the clustering. The algorithm stops.

2. Given two communities c and c' of clustering C^l , if they are connected by at least one edge in G^l , then their corresponding vertices $T^{l+1}(c)$ and $T^{l+1}(c')$ are linked by an edge in G^{l+1} . Additionally, the edge is weighted by $\frac{W^l(c,c')}{2}$.
3. A loop is added to each vertex $T^{l+1}(c)$ corresponding to community c weighted by $w^{l+1}(T^{l+1}(c), T^{l+1}(c)) = W^l(c, c)$.

This Louvain algorithm is illustrated on Figure 1 with a simple graph containing 17 vertices and 29 edges.

3 Algorithm Louvain+

We extend the Louvain algorithm by introducing an uncoarsening-refinement phase at the end of the standard Louvain algorithm. Our Louvain+ algorithm executes the following steps:

1. Run the Louvain algorithm to obtain a series of coarsened graphs G^1, G^2, \dots, G^L and clusterings C^1, C^2, \dots, C^L , assuming the highest level is L .
2. Run the uncoarsening phase from C^{L-1} and project the current clustering to a new clustering \bar{C}^{L-2} where each coarsened community of the current clustering is unfolded (uncoarsened) into its composing communities. The new clustering \bar{C}^{L-2} is immediately refined by the VM heuristic to improve its quality. The improved \bar{C}^{L-2} serves then as the initial clustering for the next projection application. This process continues until level 0 is reached. (Notice that it is useless to start the uncoarsening phase from C^L since no moves are made by the VM heuristic during the last iteration of the coarsening phase.)

We describe now the process of projection. Given two vertices v_1^l and v_2^l of graph G^l , we use $v_1^l \Gamma^l v_2^l$ to denote the relation “ v_1^l and v_2^l belong to the same community in C^l .” Furthermore, we use $\gamma^l(v^l)$ to denote the community to which vertex v^l belongs in C^l . By convention, let $\bar{C}^{L-1} = C^{L-1}$ denote the first projected clustering. At each level $l = L - 2, L - 3, \dots$, the clustering \bar{C}^l is the result of the projection of \bar{C}^{l+1} onto C^l which is optimized by the VM heuristic. In \bar{C}^l , two vertices v_1^l and v_2^l belong to the same community if the vertices in G^{l+1} corresponding to the communities $\gamma^l(v_1^l)$ and $\gamma^l(v_2^l)$ from C^l belong to the same community in C^{l+1} . Formally, this is denoted by $v_1^l \Gamma^l v_2^l \equiv T^l(\gamma^l(v_1^l)) \Gamma^{l+1} T^l(\gamma^l(v_2^l))$. This relation defines entirely the new clustering \bar{C}^l . The number of communities in \bar{C}^{l+1} is the same as in \bar{C}^l . The uncoarsening phase with refinement by the VM heuristic is illustrated on Figure 2 which starts with the result of Louvain algorithm (i.e., C^1) obtained in Figure 1.

4 Experimental results

4.1 Benchmark and protocol of test

To evaluate the efficiency of our Louvain+ algorithm, we compare it with the Louvain algorithm on a set of 13 networks from different application domains

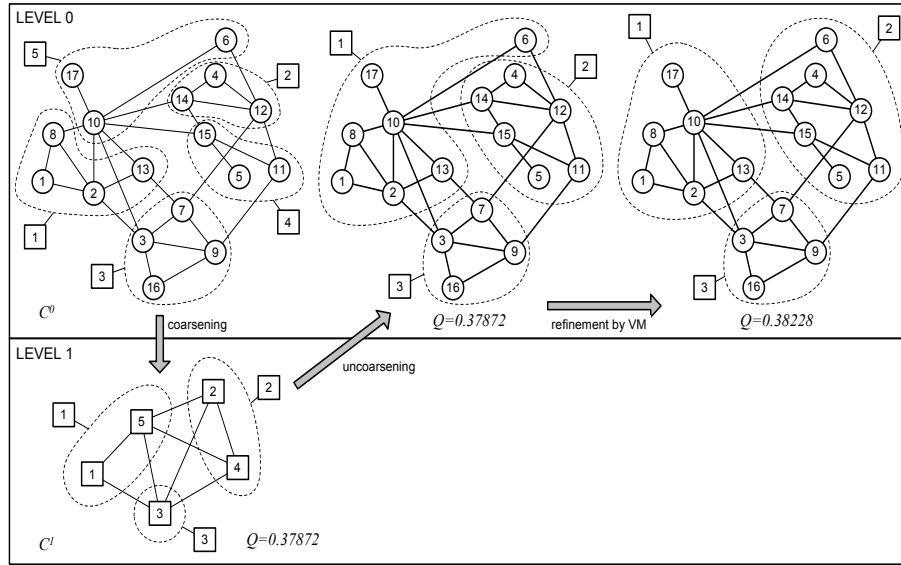


Fig. 2. Illustration of the uncoarsening-refinement phase of the Louvain+ algorithm. We start with level 1 from the example of Figure 1. The clustering C^1 of level 1 has three communities containing communities from level 0. With the uncoarsening operation, the clustering C^1 is projected to a new clustering where new communities are formed. For instance all the vertices from communities 1 and 5 of C^0 now form a new community of the projected clustering while communities 2 and 4 of C^0 lead to another new community. Since the structure of communities in the projected clustering has changed, the VM procedure can be applied to the projected clustering to obtain an improved clustering with an increased modularity. We see that displacing vertex 6 of the projected clustering from community 1 to community 2 leads to a higher modularity (0.38228 vs 0.37872).

shown in Table 1. Both algorithms are coded in Free Pascal and executed on a PC equipped with a Pentium Core i7 870 of 2.93 GHz and 8 GB of RAM³. Since the algorithm is sensitive to the order of vertices, we generate 100 instances of each graph with random vertices order. We use a deterministic version of the Louvain and Louvain+ algorithms (i.e. without preliminary random vertex reordering) and execute them on these 100 instances. For each graph, we present the distribution or average of different measures (modularity, number of vertices misplaced etc.) obtained over the 100 instances.

We use the minimal modularity gain ϵ between two consecutive iterations (see Sect. 2.1) as the stop condition of the VM procedure. We use ϵ_c and ϵ_r to distinguish the minimal modularity gain for the coarsening phase (for both

³ The source code of our Louvain+ algorithm will be made available at www.info.univ-angers.fr/pub/hao/Louvainplus.html

Table 1. Benchmark graphs in the literature for community detection with the number of vertices (n) and the number of edges (m). These are undirected graphs with medium size (from about 2000 to almost 1 million of edges).

Graph	Description	n	m	Source
Jazz	jazz musician collaborations network	198	2742	[23]
Email	university e-mail network	1133	5451	[24]
Power	topology of the Western States Power Grid of the United States	4941	6594	[25]
Yeast	Protein-Protein interaction network in yeast	2284	6646	[26]
Erdos	Erdős collaboration network	6927	11850	[27]
Arxiv	network of scientific papers and their citations	9377	24107	[28]
PGP	trust network of mutual signing of cryptography keys	10680	24316	[29]
Condmatt2003	scientific coauthorship network in condensed-matter physics	27519	116181	[30]
Astro-ph	collaboration network of arXiv Astro Physics	16046	121251	[31]
Enron	email network from Enron	36692	183831	[32]
Brightkite	friendship network from a location-based social networking service	58228	214078	[33]
Slashdot	social network from Slashdot news web site	77359	469180	[32]
Gowalla	location-based social network from a website	196591	950327	[33]

Louvain and Louvain+) and for the uncoarsening phase (only Louvain+). It is clear that a smaller ϵ induces more applications of the VM heuristic and thus more computing time. In all of our experiments, we set $\epsilon_r = 10^{-5}$.

It is obvious that with the uncoarsening-refinement phase, the proposed Louvain+ algorithm will increase or leave unchanged the modularity which is achieved by Louvain. In the rest of this section, we assess experimentally the impact of the uncoarsening phase of Louvain+ on the run time cost, the modularity improvement and the structural changes of the clustering.

4.2 Execution time and modularity

Figure 3 shows a comparison of accumulated average runtime between Louvain and Louvain+ when they are applied to the set of 13 graphs with the same parameter value $\epsilon_c = \epsilon_r = 10^{-5}$. With the same coarsening phase in both algorithms, we can measure the extra time required by the uncoarsening-refinement phase of Louvain+. We observe that the curve of Louvain+ is slightly above that of Louvain but with a similar linear growth on m (number of edges in graph). The time complexity seems to be in $O(m)$. Curve *delta* shows a linear increase of runtime required by the uncoarsening-refinement phase. Louvain+ does not change the complexity of the Louvain algorithm. Over the 13 tested graphs, the average increase of runtime caused by the refinement is about 20%.

Figure 4 presents for each graph the gain of modularity given by the refinement of Louvain+. We observe that Louvain+ leads to an increase of modularity between 0.002 and 0.01 with respect to the results obtained by Louvain. This is achieved thanks to the uncoarsening phase introduced in Louvain+.

On the other hand, as shown in Figure 3, Louvain+ consumes more CPU time than Louvain to achieve the reported (better) results. One interesting question is

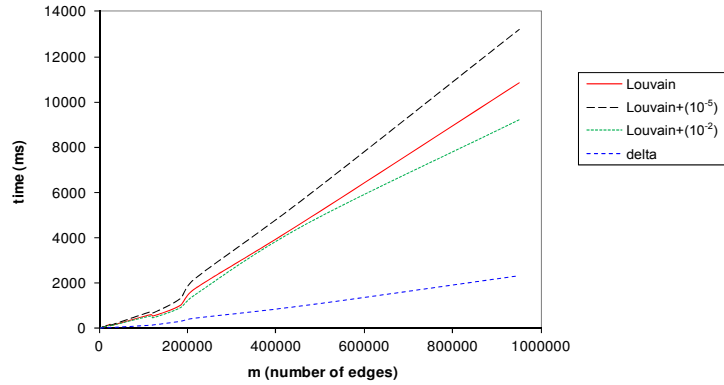


Fig. 3. Runtime comparison between Louvain and Louvain+. The three upper curves show the average run time of Louvain, Louvain+ ($\epsilon_c = 10^{-5}$) and Louvain+ ($\epsilon_c = 10^{-2}$) on the set of 13 graphs over 100 instances in milliseconds as a function of the number of edges m . The blue curve *delta* (lowest one) represents the time difference between Louvain+ ($\epsilon_c = 10^{-5}$) and Louvain.

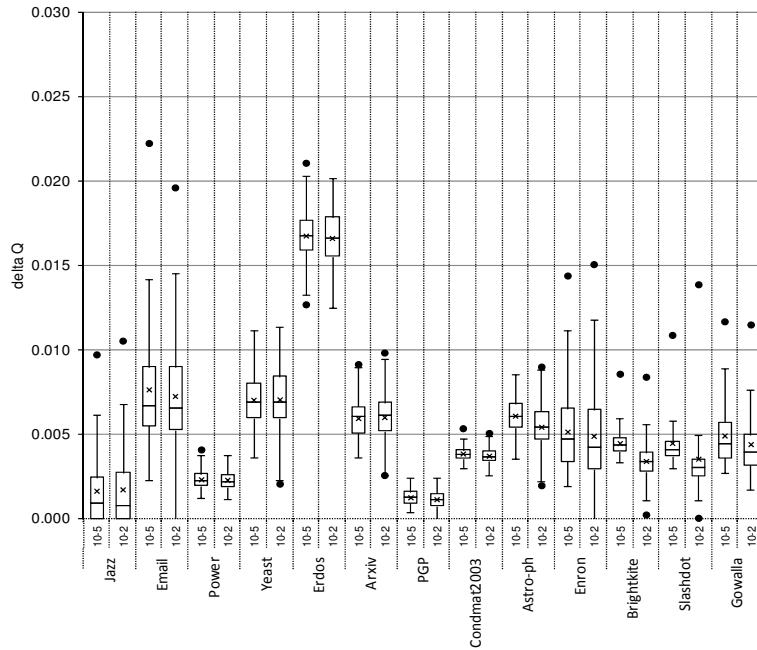


Fig. 4. Box-and-Whisker Plots for the modularity gain obtained by the uncoarsening-refinement phase on 100 instances of each graph. Two versions are tested, one with $\epsilon_c = 10^{-5}$ and one with $\epsilon_c = 10^{-2}$ (parameter value on x-axis). In both cases, the final expected precision is the same, $\epsilon_r = 10^{-5}$.

to know whether Louvain+ is able to attain the same results with less computing time. To verify this, we carry out another experiment where we run Louvain+ with a relaxed coarsening phase by using a much larger ϵ_c value ($\epsilon_c = 10^{-2}$ instead of $\epsilon_c = 10^{-5}$).

Now observe again Figure 4 for the modularity gain of Louvain+. It can be seen that Louvain+ with $\epsilon_c = 10^{-2}$ leads to a modularity performance comparable to that with $\epsilon_c = 10^{-5}$ while the computing time is decreased, and becomes lower than the computing time of Louvain. This can be explained as follows. With the relaxed ϵ_c value, the coarsening phase is reduced. Even if this generally leads to a clustering with a decreased modularity at the end of the coarsening phase, the modularity is improved during the uncoarsening-refinement phase.

Table 2. Structural comparison between Louvain and the two versions of Louvain+ (with $\epsilon_c = 10^{-5}$ and $\epsilon_c = 10^{-2}$ respectively). The average percentage of vertices, misplaced before the uncoarsening-refinement phase and correctly placed after, are computed over the 100 instances of graphs. We also show the similarity, computed by the NMI, between clusterings before and after the refinement phase (column 'similarity').

graph	% corrections		similarity
	$\epsilon_c = 10^{-5}$	$\epsilon_c = 10^{-2}$	
Jazz	0.5%	1.3%	0.953
Email	1.2%	1.2%	0.879
Power	0.1%	0.1%	0.947
Yeast	0.6%	0.4%	0.869
Erdos	1.4%	1.4%	0.856
Arxiv	0.5%	0.4%	0.907
PGP	0.1%	0.1%	0.981
Condat2003	0.4%	0.4%	0.884
Astro-ph	0.7%	0.5%	0.906
Enron	0.2%	0.1%	0.955
Brightkite	0.3%	0.1%	0.931
Slashdot	0.3%	0.0%	0.845
Gowalla	0.3%	0.2%	0.930
<i>average</i>	0.5%	0.5%	0.911

4.3 Bad vertices and structural changes in clustering

We now turn our attention to evaluate the structural changes in clustering made by the uncoarsening-refinement phase of Louvain+. For this purpose, we compare the clusterings obtained before and after the uncoarsening-refinement phase, corresponding to the results of Louvain and Louvain+ respectively. An interesting measure for this evaluation is the percentage of misplaced vertices according

to the strong sense of community criterion [34] correctly placed by the refinement. A community is defined in the strong sense if the internal degree of all the vertices of the community is greater than the external degree (there are more adjacent vertices in a community than outside). This is a very strong condition of existence of a community which is rarely satisfied in real networks, but it is interesting to count the number of vertices that do not satisfy this condition for a given clustering. To simplify our discussion, we use the term '*correction*' to designate these vertices misplaced by Louvain (i.e., those vertices with an internal degree smaller than some external degree), but correctly placed by Louvain+, i.e. by the refinement phase. Generally, according to our observations, the maximum of modularity goes with the minimum of misplaced vertices.

We show in Table 2 the percentage of vertices corrected by the refinement phase of Louvain+ and the similarity between clusterings before and after this phase. We find that the percentage of corrections is positive for all the tested graphs. This percentage represents 0.1% to 1.4% of the total vertices, with an average of 0.5% over all the tested graphs. This information allows us to confirm once again the usefulness of the uncoarsening-refinement phase introduced in the Louvain+ algorithm.

We also calculate the global structural difference between clusterings before and after the refinement phase, measured by the similarity called NMI [35]. This measure is based on information theory and mostly used in community detection. The range of NMI goes from 0 (completely different clusterings) to 1 (identical clusterings). Table 2 discloses that structural changes made by the Louvain+ refinement is quite important with a NMI between 0.84 and 0.98. As the NMI scale is logarithmic, a value of 0.9 implies a significant structural difference.

5 Conclusion and perspectives

In this work, we have presented an improved algorithm for community detection with modularity. The proposed Louvain+ algorithm extends the well-known Louvain algorithm by adding an uncoarsening-refinement phase, leading to a fully multi-level method. From the result of the Louvain algorithm, this extension goes backward and uncoarsens successively each intermediate graph generated during the Louvain algorithm and applies the vertex mover heuristic to each uncoarsened graph to improve the modularity. We have assessed the performance of the proposed algorithm on a set of 13 popular real networks. The comparisons with Louvain show that with comparable computing times, Louvain+ achieves systematically better modularity than Louvain does, thanks to the optimization during the uncoarsening-refinement phase. Experiments also disclosed that the extension of the uncoarsening phase does not change the linear complexity of the initial Louvain algorithm.

Like Louvain, the proposed Louvain+ algorithm is conceptually simple and computationally fast. As a consequence, it can be applied to very large networks that can be encountered in numerous real situations. Additionally, it can be used

within more sophisticated methods, e.g. to generate initial clusterings that are further improved by search-based heuristics.

Acknowledgment

We are grateful to the referees for their comments and questions which helped us to improve the paper. The work is partially supported by the Pays de la Loire Region (France) within the RaDaPop (2009-2013) and LigeRO (2010-2013) projects.

References

1. M. Girvan and M. E. J. Newman, *Community structure in social and biological networks*, 2002 *Proceedings of the National Academy of Sciences of the United States of America* 99(12):7821–7826
2. R. Guimerà and L. A. N. Amaral, *Functional cartography of complex metabolic networks*, 2005 *Nature* 433(7028):895–900
3. G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, *Self-organization and identification of web communities*, 2002 *Computer* 35(3):66–70
4. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, *Complex networks: Structure and dynamics*, 2006 *Physics Reports* 424(4-5):175–308
5. S. H. Strogatz, *Exploring complex networks*, 2001 *Nature* 410(6825):268–276
6. R. Albert and A.-L. Barabási, *Statistical mechanics of complex networks*, 2002 *Rev. Mod. Phys.* 74:47
7. M. Newman, *Networks: An Introduction*, 2010 *Oxford University Press*
8. M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, 2004 *Physical Review E* 69(2):026113
9. S. Fortunato, *Community detection in graphs*, 2010 *Physics Reports* 486(3-5):75–174
10. U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, *On modularity clustering*, 2008 *IEEE Transactions on Knowledge and Data Engineering* 20(2):172–188
11. A. Clauset, M. E. J. Newman, and C. Moore, *Finding community structure in very large networks*, 2004 *Physical Review E* 70(6):066111
12. M. E. J. Newman, *Fast algorithm for detecting community structure in networks*, 2004 *Physical Review E* 69(6):066133
13. P. Schuetz and A. Cafilisch, *Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement*, 2008 *Physical Review E* 77(4):046112
14. Z. Lü and W. Huang, *Iterated tabu search for identifying community structure in complex networks*, 2009 *Physical Review E* 80(2):026130
15. J. Duch and A. Arenas, *Community detection in complex networks using extremal optimization*, 2005 *Physical Review E* 72(2):027104
16. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, 2008 *Journal of Statistical Mechanics: Theory and Experiment* 10:8–+.
17. X. Liu and T. Murata, *Advanced modularity-specialized label propagation algorithm for detecting communities in networks*, 2009 *Physica A: Statistical Mechanics and its Applications* 389(7): 1493–1500

18. A. Noack and R. Rotta, *Multi-level algorithms for modularity clustering*, 2009 *Lecture Notes in Computer Science* 5526: 257–268
19. O. Gach and J.K. Hao, *A memetic algorithm for community detection in complex networks*, 2012 *Lecture Notes in Computer Science* 7492: 327–336
20. C. Walshaw, *Multilevel Refinement for Combinatorial Optimisation Problems*, 2004 *Annals of Operations Research* 131: 325–372
21. A. Arenas, J. Duch, A. Fernández, and S. Gómez, *Size reduction of complex networks preserving modularity*, 2007 *New Journal of Physics* 9(6):176
22. Pajek project, *Us airports connected by a direct flight*, <http://vlado.fmf.uni-lj.si/pub/networks/data/>, 1997.
23. P. Gleiser and L. Danon, *Community structure in social and biological networks*, 2003 *Advances in Complex Systems* 6:565–573
24. R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, *Self-similar community structure in a network of human interactions*, 2003 *Physical Review E* 68(6):065103
25. D. J. Watts and S. H. Strogatz, *Collective dynamics of "small-world" networks*, 1998 *Nature* 393(6684):440–2
26. D. Bu, Yi. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen, *Topological structure analysis of the protein-protein interaction network in budding yeast*, 2003 *Nucleic Acids Research* 31(9):2443–2450
27. J. Grossman, *The erdős number project*, <http://www.oakland.edu/enp/>, 2007
28. KDD, *Cornell kdd cup*, <http://www.cs.cornell.edu/projects/kddcup/>, 2003
29. M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, *Models of social networks based on social distance attachment*, 2004 *Phys. Rev. E* 70(5):056122
30. M. E. J. Newman. *The structure of scientific collaboration networks*, 2001 *Proceedings of the National Academy of Sciences of the United States of America* 98(2):404–409
31. J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters*, 2006 *ACM Transactions on Knowledge Discovery from Data* 1(1):2–es
32. J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, *Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters*, 2008 *Internet Mathematics* 6(1):66
33. E. Cho, S. A. Myers, and J. Leskovec, *Friendship and mobility: user movement in location-based social networks*, 2011 *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090
34. F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, *Defining and identifying communities in networks*, 2004 *Proceedings of the National Academy of Sciences of the United States of America* 101(9):2658–2663
35. L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. *Comparing community structure identification*, *Journal of Statistical Mechanics: Theory and Experiment* 2005(09):P09008, September 2005.