DOI: 10.15439/2016F273
ACSIS, Vol. 8. ISSN 2300-5963

A new polynomial class of cluster deletion problem

Sabrine Malek
University of Sfax - Tunis
Faculty of Economics and Management of Sfax
Email: sabrine.malek@gmail.com

Wady Naanaa
University of Monastir - Tunis
Faculty of Sciences of Monastir
Email: wady.naanaa@gmail.com

***Abstract*—Cluster Deletion (CD) problem asks to transform a given graph into a cluster graph by at most k edge deletions. CD is a combinatorial problem arising in the field of classification. In this paper, we introduce a graph transformation which enabled the identification of new polynomially solvable classes of CD problem. We show that if a graph is K_3-free or (*diamond*, *kite*, *house*, *xbanner*)-free then cluster deletion problem can be solved in polynomial time on that graph.**

***Index Terms*—graph clustering; cluster deletion; Line Graph; P_3 adjacency graph; forbidden patterns; diamond graph; claw graph; fork graph**

I. Introduction

CLASSIFICATION is the problem of identifying to which of a set of categories a new observation belongs [1]. In the terminology of machine learning, classification may be supervised or unsupervised. The corresponding unsupervised procedure is known as clustering which is considered the most important unsupervised learning problem [2]. On the other hand, many combinatorial problems are modelled using graphs, in particular, the partitioning of graph vertices into clusters is a classification task that may be used to better manage many real-world problems. From the theoretical point of view, the clustering task is closely related to partitioning problems. As every other problem of this kind, clustering aims to finding structures or patterns in a collection of unlabeled data. The goal is to partition these elements into subsets called clusters such that two meta-criteria are satisfied: homogeneity (elements in a same cluster should be highly similar to each other) and separation (elements from different clusters may have low similarity to each other). In the graph theoretic approach to clustering, data are often represented in the form of a graph G. Ideally, the resulting graph would be a cluster graph, that is, a graph in which every connected component is a clique, i.e., a complete subgraph. From the practical point of view, clustering algorithms can be applied in many fields, for instance in social networks, in Wireless sensor network (WSN) [3], in particular in optimizing energy distribution between access points [4], [5], or in designing electronic integrated circuits [6]...

In this paper, we deal with a specific version of the graph clustering problem, namely, cluster deletion (CD), which allows a graph partitioning, into a set of complete subgraphs, just by removing edges. This problem is known to be NP-hard [7] for general graphs. However, it may become easier and polynomial-time solvable in specific graphs, for instance split graphs, block graph, proper interval graph, cographs [8], [9]. Graph classes on which CD is polynomial-time solvable can also be specified by forbidding the occurrence of certain (small) subgraphs in the input graph. For instance, CD is polynomial-time solvable on a sub-class of P_4-sparse graphs that strictly includes P_4-reducible graphs (which are, in turn, a superclass of cographs) [10]. Those results were obtained for unweighted graphs. For weighted graphs, the cluster deletion problem can be solved in polynomial time on the class of K_3-free graphs for which the CD equivalent to maximum weighted matching [8], [11].

On the other hand, there are several works showing that CD problem is NP-hard on some subclasses of weighted graphs such, (C_5, P_5)-free graphs,($2K_2$, $3K_1$)-free graphs and (C_5, P_5, bull, 4-pan, fork, co-gem, co-4-pan)-free graphs [9].

Our aim is to derive polynomial subproblems of *CD* by resorting to graph transformation. In the literature, there are several graph transformation, among them the widely used line graph [12]. A graph H is a line graph of a graph G if the vertices of H are in a one-to-one correspondence with the edges of G, with two vertices being adjacent in H if and only if the corresponding edges of G have a vertex in common. Clearly, any graph matching in G corresponds to an independent set in its line graph, and therefore, the maximum independent set problem in the class of line graphs is equivalent to the maximum matching problem in general graphs. What is worth noting here is that finding a maximum matching in a graph is a polynomial problem [13], which implies that the maximum independent set problem in polynomially solvable in line graphs [14]. For this reason this graph transformation has been widely used both for reducing and solving the maximum independent set problem. Another transform, known as conic reduction [15], transforms a graph G into a graph G' with $\alpha(G') = \alpha(G) - 1$, where $\alpha(G)$ denotes the maximum cardinality of a maximum independent set in a graph G, that is, the stability number of G. Yet another interesting transformation is the one based on the removal of simplicial vertices [16]. Let v be a vertex and let $N(v)$ be the neighbours set of v. v is simplicial if $N(v)$ is a clique. For each simplicial vertex v, we have $\alpha(G) = \alpha(G \setminus N[v]) + 1$ holds such that $N[v] = N(v) \cup \{v\}$. Thus, given a simplicial vertex, it is easy to reduce the problem of determining $\alpha(G)$ to the same problem on a smaller graph. There are other graph transformation, such as clique reduction [17], C-reduction [18], graph reduction for QoS Prediction [19],

978-83-60810-90-3/$25.00©2016, IEEE

SWR reduction [20]. All these transformations may simplify combinatorial problems on graphs.

In the present paper, we introduce a new transformation called the P_3-*adjacency graph* and we use it to identify new polynomially-solvable classes of *CD*.

In the next section, we present a new proof of the tractability of the CD problem for unweighted K_3-free graphs which is much easier that the one proposed in [11]. Indeed, we prove that when the initial unweighted graph *G* is K_3-free, its $P_3(G)$ will correspond to its line graph and then finding a minimum deletion edge-set is equivalent to finding a maximum independent set in $P_3(G)$. In other words, solving the *CD* problem on a K_3-free graph amounts to finding a maximum matching in the line graph of *G*. Next, we show that if *G* is diamond-free then any maximum independent set of $P_3(G)$ provides a solution of *CD*. Secondly, we introduce a new collection of forbidden patterns, namely *kite*, *house*, *open-envelope* and *xbanner*, and prove that $P_3(G)$ is claw-free when *G* is (*kite*, *house*, *open-envelope*, *xbanner*)-free. This enables a polynomial computation of a maximum independent set of $P_3(G)$, and then, provides an optimal solution for *CD* on *G* in polynomial-time.

II. Definitions and notations

A graph is a mathematical structure consisting of a set of vertices and a set of edges connecting the vertices. There are several types of graphs, among which, one can distinguish simple graphs, which are defined by an ordered pair (V, E), where V is a finite set of vertices and $E \subseteq \mathcal{P}_2(V)$ is the set of edges, with $\mathcal{P}_2(V)$ being the set of all pairs of V. From a simple graph $G = (V, E)$, one can extract a partial graph $G_p = (V, E_p)$ obtained by removing some of the edges of *G*, we have therefore $E_p \subseteq E$.

Definition 1. *Let $G = (V, E)$ be a simple graph and let $U \subseteq V$. The simple graph $(U, E(U))$ is the sub-graph of G induced by U, where $E(U) = E \cap \mathcal{P}_2(U)$*

A sub-graph of a given simple graph is therefore defined as follows:

Definition 2. *A graph $G_s = (V_s, E_s)$ is the sub-graph of a graph $G = (V, E)$ if there exists $U \subseteq V$ such that G_s is the sub-graph of G induced by U i.e $V_s = U$ and $E_s = E(U)$.*

A complete graph is a simple graph in which every pair of distinct vertices is connected by a unique edge. The complete graph with n vertices is denoted by K_n. A clique of a simple graph $G = (V, E)$ is a complete subgraph of *G*. The K_3 graph is the complete graph with three vertices and a P_3 graph is the path on three vertices as it is illustrated in Fig. 1.
Observe that a complete graph cannot contain any P_3 as an induced subgraph.

Let C be a collection of small graphs, that will be designated by patterns. *G* is said to be C-*free* if *G* contains no member of C as an induced subgraph.

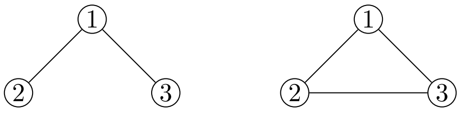

Fig. 1. A P_3 graph (left) and K_3 graph (right)

Besides, in the literature there exists a known class of graph called line graph which represents the adjacencies between the edges of a given graph.

Definition 3. *The line graph of a simple graph G= (V,E) is the graph $L(G)$= (V',E'), where:*

- *Each vertex of $L(G)$ represents an edge of G; and*
- *two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G.*

Example 1. *Fig 2. shows a simple graph and its line-graph.*

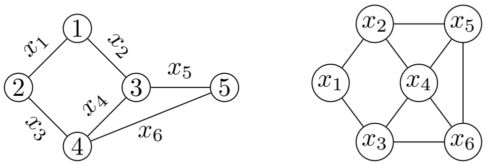

Fig. 2. Graph *G* (left) and its line graph L(G) (right)

The relevance of the line graph class is that many combinatorial problems, that are NP-hard on general graphs, are polynomially solvable on line graphs. The clustering problem is one of these combinatorial tasks. It consists in making the fewest changes to the set of edges of an input graph in order to obtain a set of cliques. There are three variations of graph clustering: cluster completion, cluster deletion and cluster editing. In the graph completion variations, edges can only be added. In cluster deletion, edges can only be deleted. In cluster editing, both edge additions and edge deletions are allowed. More formally, the cluster deletion (*CD*) problem consists in finding, for a given graph *G*, a P_3-free partial graph. An optimal solution for a *CD* instance given by a simple graph $G = (V, E)$ is a P_3-free partial graph $G_p = (V, E_p)$ of *G* that minimizes $|E - E_p|$.

III. A polynomial CD class

In this paper, we precisely consider the cluster deletion problem which allows a graph partitioning just by removing edges. This version partitions the graph into a set of complete subgraphs, i.e., cliques, and the goal is to remove the fewest edges from the input graph. We propose an algorithm that solves CD by resorting to a new graph transformation, which is defined as follows:

Definition 4. *The P_3 adjacency graph of a simple graph G= (V,E) is a simple graph, which we denote by $P_3(G)$, such that:*

- *Each vertex of $P_3(G)$ represents an edge of G; and*

- *two vertices of $P_3(G)$ are adjacent if and only if their corresponding edges form a P_3 sub-graph of G.*

Example 2.

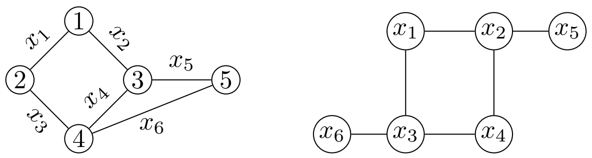

Fig. 3. Graph G (left) and its P_3(G) graph (right)

Lemma 1. *$P_3(G) = L(G)$ if and only if G is K_3-free*

Proof. Let $G = (V, E)$ be a simple graph with the associated P_3 adjacency graph $P_3(G) = (E, E_{P_3})$ and line graph $L(G) = (E, E_L)$.

($\Rightarrow$) Assume that $P_3(G) = L(G)$ while G contains a K_3 as an induced subgraph, and proceed to get a contradiction.

By Definition 3, a K_3 graph formed by edges x, y, z of G will be transformed into a K_3 in $L(G)$ whose edges are $\{x, y\}$,$\{x, z\}$,$\{y, z\}$. On the other hand, since any K_3 of G cannot contain a P_3, by Definition 4, x, y and z will not be connected in $P_3(G)$. We deduce that $E_L \neq E_{P_3}$, which contradicts our assumption. So, $P_3(G) = L(G)$ cannot hold true unless G is K_3-free.

($\Leftarrow$) Suppose that G is K_3-free and show that $P_3(G) = L(G)$, which amounts to establishing that $E_{P_3} = E_L$ since $P_3(G)$ and $L(G)$ have the same vertex-set by definition. So, let $\{x, y\}$ be any edge of E_{P_3}. According to Definition 4, the edges of G that correspond to vertices x and y in $P_3(G)$ must form an induced P_3 subgraph in G, and then, they must share a common endpoint. This implies that, $\{x, y\} \in E_L$. Thus, we have $E_{P_3} \subseteq E_L$. Conversely, let $\{x, y\}$ be in E_L, which implies that the edges x and y must share a common endpoint in G. Moreover, since G is K_3-free, it cannot contain a third edge which form a K_3 subgraph with x and y. This implies that x and y form a P_3 in G, which implies that $\{x, y\} \in E_{P_3}$. Thus, $E_L \subseteq E_{P_3}$. It follows, that $P_3(G) = L(G)$. □

Example 3. *Consider the graph G_1 depicted in Fig. 4. Its $L(G_1)$ and $P_3(G_1)$ are as shown in Fig. 5.*

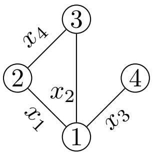

Fig. 4. A graph that contains both P_3 and K_3.

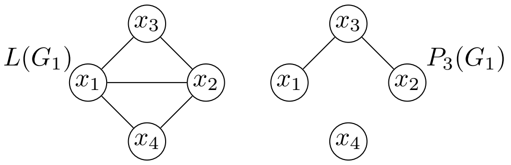

Fig. 5. The line graph and the P_3 adjacency graph of G_1.

We notice that $L(G_1)$ and $P_3(G_1)$ are different. This is due to the K_3 subgraph induced by vertices $1, 2$ and 3.

Example 4. *The figure below represents a graph G_2 composed of four vertices and containing two P_3 but no K_3:*

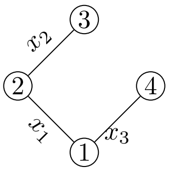

Fig. 6. A simple graph containing two P_3 but no K_3.

Fig. 7 represents the $L(G_2)$ associated with $P_3(G_2)$.

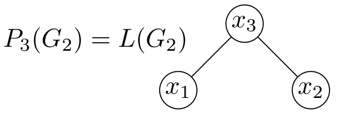

Fig. 7. The line graph and P_3 adjacency graph of G_2

Transforming G_2 into $L(G_2)$ and $P_3(G_2)$ gives the same simple graph because G_2 is K_3-free. This observation is in accordance with Proposition 1.

Lemma 2. *Let $G = (V, E)$ be a K_3-free graph. If $E' \subseteq E$ is a maximum independent set of $P_3(G)$ then (V, E') is a CD solution for G.*

Proof. ($\Leftarrow$) Assume that (V, E') is a *CD* solution for *G*. Since (V, E') is composed of a set of cliques and a clique cannot contain any P_3 as an induced subgraph, the vertices of $P_3(G)$ that correspond to the edges of E' will be pairwise not connected. This means that they form an independent set of $P_3(G)$.

It remains to prove that E' is a maximum independent set of $P_3(G)$. To this end, assume the converse is true, that is, there exists $E'' \subseteq E$ such E'' is an independent set of $P_3(G)$ and $|E''| > |E'|$. Since E'' is an independent set of $P_3(G)$, then, by Lemma 1, the partial graph of G defined by (V, E'') is P_3-free. Moreover, we have $|E - E''| < |E - E'|$ since $|E''| > |E'|$. This implies that (V, E') is not a CD solution for G which contradicts the hypothesis.

($\Rightarrow$) Let E' be a maximum independent set of $P_3(G)$. Assume that (V, E') is not a solution of *CD* for G and proceed to get a contradiction.

If (V, E') is not a solution of the *CD* instance defined by *G* then either (V, E') contains a P_3 or there exists a P_3-free partial graph (V, E'') such that $|E - E''| < |E - E'|$. By Lemma 1, the first case cannot hold true. In turn, the second case cannot hold because, otherwise, E' will not be a maximum independent set of $P_3(G)$ since $|E''| > |E'|$. □

Combining Lemma 1 and 2, we deduce that

Theorem 1. *The CD problem limited to the class of K_3-free graphs can be solved in polynomial time for simple graphs.*

Proof. According to Lemma 2, solving a *CD* instance defined by *G* amounts to finding a maximum independent set in $P_3(G)$. On the other hand, according to Lemma 1, if *G* is K_3-free, then $P_3(G)$ is identical to its line graph. In addition, it is well established that the maximum independent problem is polynomial on the class of line graphs. It follows that the *CD* problem for K_3-graphs can be solved by first constructing a P_3 adjacency graph , and then, by computing, in polynomial time, a maximum independent set in the latter graph. Finally, since the P_3 adjacency graph can be built in $O(|V|^4)$ step, the overall process is polynomial. □

In what follows, we identify a wider tractable class of *CD*, which is also defined via forbidding certain graph patterns.

Lemma 3. *Let $G = (V, E)$ be a diamond-free graph and let $E' \subseteq E$. (V, E') is a CD solution for G if and only if E' is a maximum independent set of $P_3(G)$.*

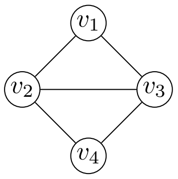

Fig. 8. Diamond graph

Proof. (⇐) Let E' be a maximum independent set of $P_3(G)$. Assume that (V,E') is not a *CD* solution for *G*. This implies that either (V, E') contains P_3 as an induced subgraph or there is a P_3-free partial graph (V, E'') such that $|E''| > |E'|$. Assure the former case, i.e., (V, E') contains a P_3, say u-v-w, and proceed to get a contradiction. Denote by e, the edge $\{u, v\}$ and by e', the edge $\{v, w\}$. Since e and e' are in E' and $E' \subseteq E$, which is a maximum independent set of $P_3(G)$, the latter two edges must not form a P_3 in (V, E). On the other hand, e and e' form a P_3 in (V, E'). This implies that $\{u, w\}$ is in E but not in E'. This cannot occur unless there is another edge $\{x, u\} \in E'$ that forms a P_3 with $\{u, w\}$ but neither with e nor with e'. Moreover, since $\{x, u\}$ and $\{u, w\}$ form a P_3, $\{x, w\}$ must not be in G. It follows that $(\{u, v, w, x\}, E(\{u, v, w, x\}))$ is an induced diamond subgraph of G, which contradicts the hypothesis. In the latter case, there is a P_3-free partial graph (V, E'') such that $|E''| > |E'|$. This implies that E' is not a maximum independent set of $P_3(G)$, which also contradicts the hypothesis.

(⇒) Let (V,E') be a CD solution for *G* and assume that E' is not a maximum independent set of $P_3(G)$, then proceed to get a contradiction.

Since E' is not a maximum independent set of $P_3(G)$ then either E' is not an independent set of $P_3(G)$ or it is not maximum. From the definition of $P_3(G)$, the former case implies that there exist $e, e' \in E'$ that form a P_3 in G, which contradicts the fact (V, E') is a *CD* solution for G. The latter case implies that there exists a maximum independent set $E'' \subseteq E$ such that $|E''| > |E'|$. Using (⇐), we deduce that (V, E') is not a *CD* solution for *G* and this contradicts the hypothesis. □

The complete bipartite graph $K_{1,3}$ is known as the *claw* graph. It is illustrated by Fig. 9. In what follows, the goal is to obtain a $P_3(G)$ free from claw. Recall that this allows to polynomially solve the maximum independent set problem in $P_3(G)$. Thus, interested by determining the patterns which entail a claw in $P_3(G)$.

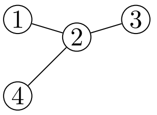

Fig. 9. A complete bipartite graph $K_{1,3}$ (Claw graph or Y graph)

Lemma 4. *Let $G = (V, E)$ be a simple graph. $P_3(G)$ has a claw as an induced subgraph if and only if G contains one of the graphs in Fig. 10 as an induced subgraph.*

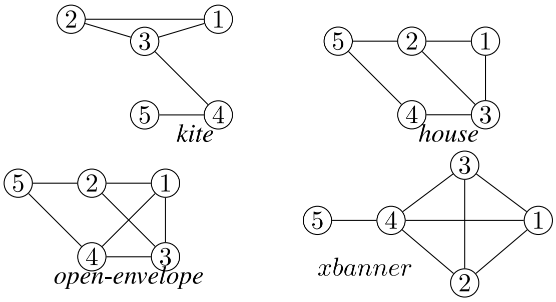

Fig. 10. The set of minimal graphs, which when they are present in G as subgraphs entail a claw in P_3(G) [21]

Proof. (⇒) Let *G* be a simple graph and let $P_3(G)$ be the P_3 adjacency graph of *G*. Assume that $K_{1,3}$ is a subgraph of $P_3(G)$ and, by the same time, *G* does not contain any *kite*, *house*, *open-envelope* and *xbanner* as an induced subgraph and proceed to get a contradiction. By referring to Definition 4 and since $K_{1,3}$ is composed by a vertex which has three not connected neighbors, if $P_3(G)$ contains a $K_{1,3}$ then *G* must contain three P_3 subgraphs sharing a common edge, say a. So, we should have in *G* one of the following two subgraphs:

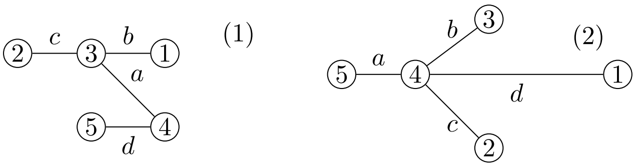

Fig. 11. Minimal patterns that entail a claw in $P_3(G)$

To ensure that a claw graph (Fig. 9) occurs in the P_3 adjacency graph of G, we add, in all possible manners, the edges that keep the claw graph as an induced subgraph of $P_3(G)$. This results in the following four patterns, which correspond exactly to the forbidden patterns of Fig 12 and contradicts the hypothesis.

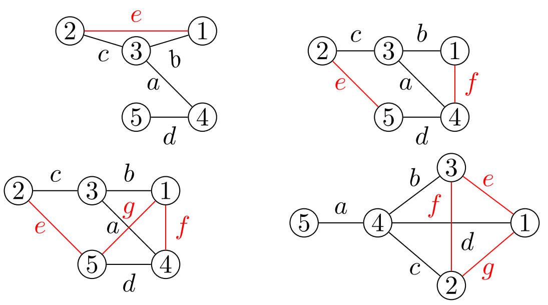

Fig. 12. After change: the set of minimal graphs, which when they are present in G as subgraphs, entail a claw in $P_3(G)$

(⇐) If G contains a *kite*, *house*, *open-envelope* or *xbanner* as an induced subgraph, then $P_3(G)$ will respectively contain one of the following graphs as an induced subgraph (see Fig. 13, Fig. 14, Fig. 15).

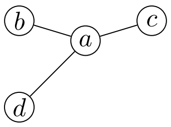

Fig. 13. P_3(kite) and P_3(xbanner)

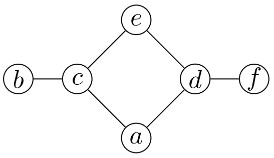

Fig. 14. P_3(house)

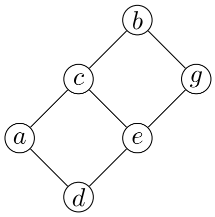

Fig. 15. P_3(open-envelope)

Observe that all these subgrahs contain a $P_{1,3}$ as an induced subgraph. □

Theorem 2. *The CD problem is polynomial in the class of (kite, house, xbanner, diamond)-free graphs.*

Proof. Let G be a (*kite*, *house*, *xbanner*, *diamond*)-free graph. Since *diamond* is an induced subgraph of *open-envelope* and by Lemma 4, $P_3(G)$ is claw-free. On the other hand, by Lemma 3, a maximum independent set of $P_3(G)$ corresponds to a *CD* solution for G. Since maximum independent set problem can be solved in polynomial time in claw-free graphs [12], *CD* can be solved in polynomial time in (*kite*, *house*, *xbanner*,*diamond*)-free graphs. □

IV. Conclusion

In this paper, we identified polynomial classes of the *CD* problem. We introduced a new graph transformation, namely, the P_3 adjacency graph. We used this transformation in order to show that whenever a simple graph G is diamond-free, any maximum independent set of $P_3(G)$ provides a solution of CD. Next, we showed that *CD* problem can be solve in polynomial time in K_3-free and (*kite*, *house*, *xbanner*,*diamond*)-free unweighted graphs.

References

[1] M. Azad and M. Moshkov, "Classification and optimization of decision trees for inconsistent decision tables represented as MVD tables," in *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015*, 2015, pp. 31–38. [Online]. Available: http://dx.doi.org/10.15439/2015F231

[2] R. Ziembinski, "Unsupervised extraction of graph-stream structure for purpose of knowledge retrieval and information fusion," in *Position Papers of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015.*, 2015, pp. 53–60. [Online]. Available: http://dx.doi.org/10.15439/2015F288

[3] C.-T. Cheng, C. K. Tse, and F. C. M. Lau, "A clustering algorithm for wireless sensor networks based on social insect colonies," *IEEE Sensors Journal*, pp. 711–721, 2010. [Online]. Available: http://dx.doi.org/10.1109/JSEN.2010.2063021

[4] C. S. Nam, Y. S. Han, and D. R. Shin, "Multi-hop routing-based optimization of the number of cluster-heads in wireless sensor networks," *Sensors Journal*, pp. 2875–2884, 2011. [Online]. Available: http://dx.doi.org/10.3390/s110302875

[5] N. Amini, A. Vahdatpour, W. Xu, M. Gerla, and M. Sarrafzadeh, "Cluster size optimization in sensor networks with decentralized cluster-based protocols." *Computer Communications*, pp. 207–220, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2011.09.009

[6] J. Cong and S. K. Lim, "Edge separability-based circuit clustering with application to multilevel circuit partitioning," *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp. 346–357, 2004. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2004.823353

[7] R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," *Discrete Applied Mathematics*, pp. 173–182, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2004.01.007

[8] F. Bonomo, G. Durán, and M. Valencia-Pabon, "Complexity of the cluster deletion problem on subclasses of chordal graphs," *Theor. Comput. Sci.*, pp. 59–69, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2015.07.001

[9] Y. Gao, D. R. Hare, and J. Nastos, "The cluster deletion problem for cographs," *Discrete Mathematics*, pp. 2763–2771, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.disc.2013.08.017

[10] F. Bonomo, G. Durán, A. Napoli, and M. Valencia-Pabon, “A one-to-one correspondence between potential solutions of the cluster deletion problem and the minimum sum coloring problem, and its application to P_4-sparse graphs,” *Inf. Process. Lett.*, pp. 600–603, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ipl.2015.02.007

[11] J. Edmonds, “Maximum matching and a polyhedron with 0, 1-vertices,” *Journal of Research of the National Bureau of Standards B*, pp. 125–130, 1965. [Online]. Available: http://archive.org/details/jresv69Bn1-2p125

[12] V. V. Lozin and M. Milanic, “A polynomial algorithm to find an independent set of maximum weight in a fork-free graph,” *J. Discrete Algorithms*, pp. 595–604, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.jda.2008.04.001

[13] J. Edmonds, “Paths, trees, and flowers,” *Canad. J. Math.*, pp. 449–467, 1965. [Online]. Available: http://dx.doi.org/10.4153/CJM-1965-045-4

[14] C. Berge, “Two theorems in graph theory,” *Proceedings of the National Academy of Sciences of the United States of America*, pp. 842–844, 1957. [Online]. Available: http://www.jstor.org/stable/89875

[15] V. V. Lozin, “Conic reduction of graphs for the stable set problem,” *Discrete Mathematics*, pp. 199–211, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0012-365X(99)00408-2

[16] A. Brandstädt and P. L. Hammer, “On the stability number of claw-free P_5-free and more general graphs,” *Discrete Applied Mathematics*, pp. 163–167, 1999. [Online]. Available: http://dx.doi.org/10.1016/S0166-218X(99)00072-4

[17] L. Lovàsz and M. D. Plummer, *Matching theory*. Amsterdam, New York: North-Holland, 1986. [Online]. Available: http://opac.inria.fr/record=b1086098

[18] P. A. Catlin, “A reduction method for graphs,” in *Proc. 19th Southeastern Conf. (Baton Rouge), Congressus Numerantium 65*, 1988, pp. 159–170.

[19] A. Goldman and Y. Ngoko, “On graph reduction for qos prediction of very large web service compositions,” in *IEEE Ninth International Conference on Services Computing, Honolulu, HI, USA*, 2012, pp. 258–265. [Online]. Available: http://dx.doi.org/10.1109/SCC.2012.21

[20] J. Cardoso, A. P. Sheth, J. A. Miller, J. Arnold, and K. Kochut, “Quality of service for workflows and web service processes,” *J. Web Sem.*, pp. 281–308, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2004.03.001

[21] H. N. de Ridder *et al.*, “Information System on Graph Classes and their Inclusions (ISGCI),” http://www.graphclasses.org.