RAIRO-Oper. Res. 49 (2015) 313–334
DOI: 10.1051/ro/2014040

RAIRO Operations Research
www.rairo-ro.org

ON MINIMUM WEAKLY CONNECTED INDEPENDENT SETS FOR WIRELESS SENSOR NETWORKS: PROPERTIES AND ENUMERATION ALGORITHM

Fatiha Bendali[1], Jean Mailfert[1] and Djelloul Mameri[1]

Abstract. Modeling topologies in Wireless Sensor Networks principally uses domination theory in graphs. Indeed, many dominating structures have been proposed as virtual backbones for wireless networks. In this paper, we study a dominating set that we call *Weakly Connected Independent Set* (*wcis*). Given an undirected connected graph $G = (V, E)$, we say that an independent set S in G is weakly connected if the spanning subgraph $(V, [S, V \setminus S])$ is connected, where $[S, V \setminus S]$ is the set of edges having exactly one end in S. The minimum weakly independent connected set problem consists in determining a *wcis* of minimum size in G. First, we discuss some complexity and approximation results for that problem. Then we propose an implicit enumeration algorithm which computes a minimum *wcis* in a graph with n vertices with a running time $O^*(1.4655^n)$ and polynomial space. Processing results are given that show that our enumeration program solves the *mwcis* problem for graphs whose number of vertices is less than 120.

Keywords. Dominating set, maximal independent set, minimum weakly connected independent set, wireless sensor networks, approximation, implicit enumeration.

Mathematics Subject Classification. 05C69, 05C85.

1. Introduction

Numerous civil and military applications use networked sensors [1,12]. Actually, sensors can be deployed to gather meteorological measures such as temperature

Received September 18, 2013. Accepted August 1, 2014.

[1] Laboratoire LIMOS – UMR 6158 CNRS, Campus des Cézeaux, 63173 Aubière Cedex, France. bendali@isima.fr

and pressure. They can also detect natural disasters such as earthquakes and conduct emergency response units to survivors. A Wireless Sensor Network (WSN) generally consists in a set of autonomous composants which collect data and broadcast messages to a base station. The communications are achieved *via* a shared bandwidth directly if the devices are close enough or through relays provided by intermediary sensors. Unfortunately, the network performance is reduced by interferences and unavoidable retransmissions can increase energy consumption. As there is no physical infrastructure like in wired networks, a virtual backbone needs to be created by choosing some sensors as dominator nodes. Thus, all nodes can communicate through the selected nodes straightforwardly or *via* dominee nodes.

An undirected communication graph $G = (V, E)$ [7, 12], is naturally associated to the sensors located in the region they monitor. The node set V is the set of sensors and an edge $e = \{u, v\}$ in E is a possible transmission link between two sensors u and v. This link depends on the euclidian distance between u and v and the energy to deploy for this connection. Usually the size of a Wireless Sensor Network is large and its nodes have very limited ressources. So the virtual backbone should be built with low communication and computation costs. *Connected Dominating Sets* [18] have been proposed as a solution by many authors [6, 20, 30, 32]. A node set D is *a connected dominating set*, or *cds* for short, if each vertex in G is in D or adjacent to at least one of the vertices in D (domination property) and if the subgraph induced by D is connected. Thus, communications are ensured between all the vertices *via* the set D. As one wants to reduce the number of exchanged messages and to avoid useless energy consumption, D must be of small size. But obtaining a minimum connected dominating set is an NP-hard problem [13]. Consequently, many approximation algorithms and heuristics have been proposed for that problem [2,16,23,29,30]. A greedy approximation algorithm has been described by Guha and Khuller [16] which gave a *cds* with a size of at most $(3 + \ln(\Delta(G)))$ the size of a minimum *cds* where $\Delta(G)$ is the maximum degree in the communication graph G.

The *cds* notion can be weakened by using a *weakly connected dominating set* or *wcds* [5,9]. A dominating set D is said weakly connected if the partial graph (V, F) is a connected graph where F is the set of edges of E which have at least one end in D. Yet, the problem of minimizing the cardinality of a *wcds* remains NP-hard [13]. In [5], a theoretical performance ratio of the approximation algorithms proposed for finding small *wcds* is $O(\ln(\Delta(G)))$ compared to the minimum size *wcds*.

An independent set is a subset of V that does not contain any edge of E. In [31, 32], the authors use algorithms which construct a connected dominating set by adding vertices of $V \setminus S$ to a maximal independent set S. It is easy to obtain greedily a maximal independent set in a graph G. Also it is known that a minimum maximal independent set can be found in polynomial time for some graph classes like interval graphs [4] and chordal graphs [10] whereas the problem remains NP-Hard in bipartite graphs and comparability graphs [8].

In [25], Moon and Moser showed that the number of maximal independent sets of a graph with n vertices is upper bounded by 1.443^n. Johnson *et al.* [21] gave a polynomial delay algorithm to generate all maximal independent sets. In [14], Gaspers and Liedloff present an $O^*(1.357^n)$ for solving the minimum maximal independent set. Recently, Bourgeois et alii. [3] have improved this result by a branching algorithm that computes a minimum maximal independent set with a running time $O^*(1.335^n)$.

Rather than lose the independence property for the connectivity property, we can specify conditions on an independent set to gain the *weak* connectivity. Indeed, in [2], it is showed that some particular maximal independent sets can be weakly connected. These sets of vertices are such that dominators may be connected through dominees. More formally, a weakly connected independent set is an independent set $W \subset V$ such that the partial graph $G_W = (V, [W, V \setminus W])$ is a connected graph. Such a set can be used as a structural basis for a cluster based architecture in wireless sensor networks [28]. Furthermore, the set V can be partitioned into three subsets: *slaves, masters and bridges* whose function is respectively to conduct detection activities, to collect data and to ensure cluster communication.

The present paper deals with properties about *weakly connected independent* sets, *wcis* for short, in a connected graph. We also describe a specific implicit enumeration algorithm in $O^*(1.4655^n)$ for the minimum *wcis* and display computational results. Despite its higher complexity compared to Bourgeois *et al.*'s prominent algorithm, our targeted approach provides a first wcis very fast, and finds an optimum solution on instances with more than a hundred vertices. A numerical comparison using results in [22] is made with implemented mis enumeration methods. Our results can be considered as a first theoretical and computational step towards a deeper study of this interesting structure.

The paper is organized as follows. In Sections 2 and 3, we give some definitions, notations and some basic properties of a *wcis* and related sets like maximal independent sets. Section 4 is dedicated to the complexity and approximation results for the minimum cardinality *wcis* problem. In Section 5 we study this problem in particular graph classes such as bipartite, split and comparability graphs. Section 6 describes an implicit algorithm for the *mwcis* problem and analyses its performance. Section 7 presents some numerical results and, in particular, we compare the running times of our procedure with the experimental tests stemming from [22]. A conclusion is in Section 8.

2. Notations and definitions

A finite undirected graph G is denoted by $G = (V, E)$ where V is the *vertex set* and E the *edge set*. In the following, we assume that the graph G is connected. $\Delta(G)$, (resp. $\delta(G)$) is the maximum (resp. minimum) degree in G. For two vertices u and v in G, the distance $d_G(u, v)$ is the minimum length of a path connecting u and v. If $S \subset V$ and $u \notin S$, $d_G(u, S) = \min_{v \in S}\{d_G(u, v)\}$. The *neighborhood* $N(u)$ of a vertex u is the set of nodes at distance 1 from u whereas the 2-neighborhood of u, $N^2(u)$, contains the nodes at distance 2 from u. The closed neighborhood of a vertex u is $N[u] = N(u) \cup \{u\}$. For any subset S of V, the outer neighborhood $N(S)$ is such that $N(S) = \{v \in V \backslash S;\ \exists u \in S, d_G(u,v) = 1\}$. Given S and S' two disjoint subsets of V, $[S, S']$ denotes the set of edges with exactly one end in S and in S'. If $S \subset V$, then denote by $G(S)$ the subgraph induced by the vertex set S.

Definition 2.1. (is) A subset $S \subset V$ is an *Independent Set* or a *stable set* in G if there is no edge in E between two vertices of S.

Definition 2.2. (mis) An independent set S is *maximal* if there does not exist an independent set in G which strictly contains S.

Definition 2.3. (wcis) An independent set W of G such that the partial graph $G_W = (V, [W, V\backslash W])$ is connected is called *Weakly Connected Independent Set*.

3. Basic properties of weakly connected independent sets

Given an undirected connected graph $G = (V, E)$ with $|V| \geq 2$, let $\mathcal{W}(G)$ be the set of weakly connected independent sets of G. The following Lemma is easily seen.

Lemma 3.1. *If* $W \in \mathcal{W}(G)$ *then*

(i) W *is a maximal stable set,*
(ii) $G_W = (V, [W, V\backslash W])$ *is a connected bipartite graph,*
(iii) There is a partition $V_1, \ldots, V_p$ *of* V *such that* $V_i \cap W = \{w_i\}$, $V_i \subseteq N[w_i]$ *for* $i = 1, \ldots p$ *and* $d_G(w_i, \cup_{j=1}^{i-1}\{w_j\}) = 2$, *for* $2 \leq i \leq p$.

The next property which characterizes a *wcis* can be found in [2].

Lemma 3.2. *Let* W *be a maximal independent set in* G. W *is a wcis in* G *if and only if, for any subset* $A \subset W$, *there exist a vertex* $u \in A$ *and a vertex* $v \in W \setminus A$ *such that* $d_G(u, v) = 2$.

We denote by MWCIS(G) a weakly connected independent set of minimum cardinality in G. The lemma below describes bounds for the cardinality of a MWCIS(G).

Lemma 3.3. *If* $W \in \mathcal{W}(G)$, *then*

(i) $\frac{|V|-1}{\Delta(G)} \leq |W|$,
(ii) $|W| \leq (\Delta(G) - 1)|MWCIS(G)| + 1$,
(iii) $|MWCIS(G)| \leq |V| - \Delta(G)$,
(iv) There do not exist a real number β, $0 < \beta < 1$, *and an integer* N_β *such that* $|MWCIS(G)| \leq \beta \times |V|$, *for all connected graph* $G = (V, E)$ *with* $|V| \geq N_\beta$.

Proof. Let $|\text{MWCIS}(G)| = \bar{p}$. Suppose first that $\bar{p} = 1$ and $\text{MWCIS}(G) = \{\bar{w}_1\}$. Then $\bar{w}_1$ is adjacent to any vertex in V, and $V = N[\bar{w}_1]$. So $|V| - 1 = \Delta(G)$. Therefore (i) holds for any $W \in \mathcal{W}(G)$. Moreover, as $W \subset N(\bar{w}_1)$, for any $W \in \mathcal{W}(G)$ which does not contain $\bar{w}_1$, we obtain (ii).

Assume now that $\bar{p} \geq 2$. From Lemma 3.1 (iii), denote by $\bar{V}_1, \ldots, \bar{V}_{\bar{p}}$ the partition induced by $\text{MWCIS}(G)$. As $\bar{V}_i \subseteq N[\bar{w}_i]$ for all $1 \leq i \leq \bar{p}$, and $d_G(\bar{w}_i, \cup_{j=1}^{i-1}\{\bar{w}_j\}) = 2$, for all $2 \leq i \leq \bar{p}$, we have that

$$|\bar{V}_1| \leq \Delta(G) + 1 \text{ and } |\bar{V}_i| \leq \Delta(G) \text{ for } i \geq 2. \tag{1}$$

(i) As $V = \cup_{i=1}^{\bar{p}} \bar{V}_i$, we get

$$|V| \leq \Delta(G) + 1 + \Delta(G)(\bar{p} - 1).$$

This implies that

$$\frac{|V| - 1}{\Delta(G)} \leq \bar{p}. \tag{2}$$

Moreover $|W| \geq \bar{p}$, for any set $W \in \mathcal{W}(G)$, then the inequality (2) yields (i).

(ii) Let W be in $\mathcal{W}(G)$. We give an upper bound on $|\bar{V}_i \cap W|$, for any i. If $\bar{w}_{i_0} \in W$, for some $i_0 \in \{1, \ldots \bar{p}\}$, then $|\bar{V}_{i_0} \cap W| = 1$. Thus, $|\bar{V}_i \cap W| \leq |\bar{V}_i| - 1$ for any $i \in \{1, \ldots \bar{p}\}$. By (1) we have $|\bar{V}_1 \cap W| \leq \Delta(G)$ and $|\bar{V}_i \cap W| \leq \Delta(G) - 1$ for all $i \geq 2$. As $W = \bigcup_{i=1}^{\bar{p}} (\bar{V}_i \cap W)$, we obtain that

$$|W| \leq \Delta(G) + (\Delta(G) - 1)(\bar{p} - 1),$$

and (ii) follows.

(iii) It suffices to take a wcis containing a vertex of degree $\Delta(G)$.

(iv) Suppose on the contrary that there are a real number β, $0 < \beta < 1$, and an integer N_β such that $|\text{MWCIS}(G)| \leq \beta \times |V|$, for all connected graph $G = (V, E)$ with $|V| \geq N_\beta$.

Let $p_\beta = \left\lceil \frac{\beta}{1-\beta} \right\rceil$ and $K_N = (\{u_1, \ldots u_N\}, E(K_N))$ be a clique of order $N \geq N_\beta$. Define the graph $H = (V(H), E(H))$ by

- $V(H) = \{u_1, \ldots, u_N\} \cup \{v_i^j : 1 \leq j \leq p_\beta, 1 \leq i \leq N\}$,
- $E(H) = E(K_N) \cup \{(u_i, v_i^j) : 1 \leq j \leq p_\beta, 1 \leq i \leq N\}$.

Note that the set $\bigcup_{i=1}^N \{v_i^1, \ldots, v_i^{p_\beta}\}$ is not a wcis of H. As K_N is complete, we see that $|W \cap \{u_1, \ldots u_N\}| = 1$, for any node set $W \in \mathcal{W}(H)$. W.l.o.g. $\text{MWCIS}(H) = \{u_1\} \cup (\bigcup_{i=2}^N \{v_i^1, \ldots, v_i^{p_\beta}\})$. Then

$$\frac{|\text{MWCIS}(H)|}{|V(H)|} = \frac{1 + p_\beta(N-1)}{N + p_\beta N} = \frac{p_\beta}{p_\beta + 1} + \frac{1 - p_\beta}{N(1 + p_\beta)}.$$

And this ratio exceeds β for large values of N. □

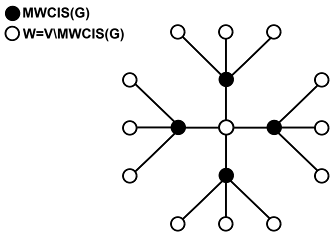

FIGURE 1. An example where the bounds (i) and (ii) in Lemma 3.3 are tight.

4. Complexity results

Given a connected graph $G = (V, E)$ and an integer k, the Weakly Connected Independent Set problem is to ask whether there exists a set $W \in \mathcal{W}(G)$ of size k or less in G.

Theorem 4.1. *Weakly Connected Independent Set is NP-Complete.*

We postpone the proof of Theorem 4.1 till paragraph 5 where it becomes a consequence of Theorem 5.6.

Following a negative approximation result for the Minimum Maximal Independent Set problem is extended to the Minimum Weakly Connected Independent Set problem.

Theorem 4.2. *No polynomial algorithm can approximately solve the Minimum Weakly Connected Independent Set problem within ratio* $|V|^{1-\epsilon}$*, for any* $\epsilon > 0$*, unless* $P = NP$*.*

Proof. Suppose that there exists a polynomial algorithm A_ϵ which, given a connected graph $G = (V, E)$, constructs a wcis $A_\epsilon(G)$ such that

$$|A_\epsilon(G)| \leq |V|^{1-\epsilon} |\text{MWCIS}(G)| \tag{3}$$

with $\epsilon \in]0, 1[$.

We also know that the Minimum Maximal Independent Set problem is very hard from an approximation point of view [17]. But, algorithm A_ϵ can be transformed into a polynomial approximation algorithm B for this last problem as follows. Define the graph $G'' = (V'', E'')$ by

- $V'' = V \cup Z$ where $Z = \{z_u : u \in V\}$,
- $E'' = E \cup \{(u, z_u) : u \in V\} \cup \{(z_u, z_v) : u, v \in V, u \neq v\}$.

Note that Z is a clique of order $|V|$. We denote by MMIS(G) the minimum maximal independent set in G.

Claim 1.

$$|\text{MMIS}(G)| \leq |\text{MWCIS}(G'')| \leq |\text{MMIS}(G)| + 1.$$

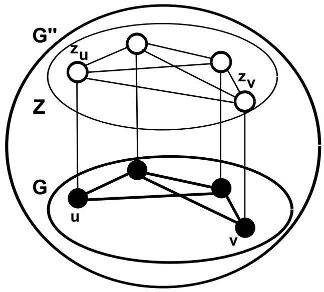

FIGURE 2. G and $G^{''}$ in the proof of Theorem 4.2.

Proof. As MMIS$(G) \neq V$, we can choose a vertex $z_{u_0} \in Z$ which is not a copy of a vertex in MMIS(G). Let $S_1 = \text{MMIS}(G) \cup \{z_{u_0}\}$, with $u_0 \in V \setminus \text{MMIS}(G)$. It is easy to see that S_1 is a wcis in G''. Then

$$|\text{MWCIS}(G^{''})| \leq |S_1| = |\text{MMIS}(G)| + 1. \tag{4}$$

As Z is a clique and there is at least an edge in E, we have $\text{MWCIS}(G'') \cap Z = \{z_{u_1}\}$, for some $u_1 \in V$. Let $T = \text{MWCIS}(G'') \setminus \{z_{u_1}\}$. T is an independent set of G since MWCIS(G'') is a stable set in $G^{''}$. Moreover, as MWCIS(G'') is a dominating set in G'', any vertex $v \in V \setminus T$, different from u_1, has a neighbour in T. Let S be such that

$$S = \begin{cases} T \cup \{u_1\}, & \text{if } T \cup \{u_1\} \text{ is stable in } G; \\ T, & \text{otherwise.} \end{cases}$$

It is easy to see that S is a maximal independent set in G, and

$$|S| \leq |T| + 1 = |\text{MWCIS}(G^{''})|.$$

Then, we obtain that

$$|\text{MMIS}(G)| \leq |S| \leq |\text{MWCIS}(G^{''})|. \tag{5}$$

Inequalities (4) and (5) give

$$|\text{MMIS}(G)| \leq |\text{MWCIS}(G^{''})| \leq |\text{MMIS}(G)| + 1,$$

which finishes the proof of the Claim 1 (*cf.* Figs. 3a and 3b). □

Consider now the wcis $A_\epsilon(G'')$ obtained by application of algorithm A_ϵ on $G^{''}$. Necessarily we have that $A_\epsilon(G'') = W_1 \cup \{z_a\}$ with $W_1 \subset V$ and $a \in V$. As in Claim 1, let the independent dominating set $W_2(G)$ be given by

$$W_2(G) = \begin{cases} W_1 \cup \{a\}, & \text{if } W_1 \cup \{a\} \text{ is stable in } G; \\ W_1, & \text{otherwise.} \end{cases}$$

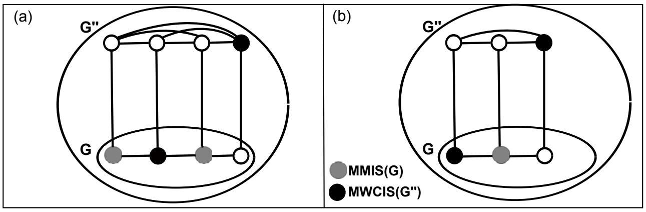

FIGURE 3. (a)|MWCIS(G'')|=|MMIS(G)|.(b)|MWCIS(G'')|=|MMIS(G)|+1.

From (3), we deduce that

$$|W_2(G)| \leq |A_\epsilon(G'')| \leq |V''|^{1-\epsilon}|\text{MWCIS}(G'')|.$$

Then, Claim 1 implies that

$$|W_2(G)| \leq (2 \times |V|)^{1-\epsilon}(|\text{MMIS}(G)| + 1).$$

So we have that

$$|W_2(G)| \leq 2^{2-\epsilon} \times |V|^{1-\epsilon}|\text{MMIS}(G)|.$$

Note that

$$2^{2-\epsilon}n^{1-\epsilon} \leq n^{1-\frac{\epsilon}{2}},$$

when $n \geq n_0$, for some $n_0 \in \mathbb{N}$. Hence,

$$|W_2(G)| \leq |V|^{1-\frac{\epsilon}{2}}|\text{MMIS}(G)|, \tag{6}$$

if $|V| \geq n_0$.

Thus, we can sketch a polynomial algorithm B which produces an Independent Dominating Set $B(G)$ for graph $G = (V, E)$ such that

$$B(G) = \begin{cases} W_2(G), & \text{if } |V| \geq n_0; \\ \text{MMIS}(G), & \text{otherwise (obtained by enumeration).} \end{cases}$$

Inequality (6) implies that B is a polynomial approximation algorithm for the Minimum Independent Dominating Set problem, which contradicts the theorem in [17]. □

5. Weakly connected independent sets in some graph classes

Definition 5.1. A graph $G = (V, E)$ is bipartite if there is a partition of its vertex set V into two disjoint sets A and B such that each edge of E joins a node in A to a node in B.

Definition 5.2. A graph $G = (V, E)$ is a split-graph if there is a partition of its node set V into a clique K and a stable set I. It is connected if the set of edges $[\{v\}, K] \neq \emptyset, \forall v \in I$.

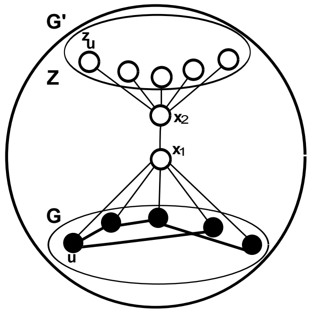

FIGURE 4. G and G' in the proof of Theorem 5.6.

Denote by $\mathcal{B}$ the connected bipartite graph class. The proofs of the first two results below are easy and will be omitted.

Theorem 5.3. *A graph $G = (A \cup B, E)$ in $\mathcal{B}$ has exactly two weakly connected independent sets, associated with A and B, respectively.*

Theorem 5.4. *In a connected split graph $G = (K \cup I, E)$, there are at most $(|K| + 1)$ weakly connected independent sets.*

Given a connected split graph $G = (K \cup I, E(G))$, we easily deduce from Theorem 5.4, that $|\text{MWCIS}(G)| = 1 + |I| - \max\{|[\{u\}, I]|; u \in K\}$.

Definition 5.5. (*comparability graph*) A connected graph $G = (V, E)$ is a comparability graph if G has an acyclic transitive orientation.

Theorem 5.6. *Minimum Weakly Connected Independent Set problem is NP-hard for comparability graphs.*

Proof. Given a comparability graph $G = (V, E)$, let the graph $G' = (V', E')$ be such that (*cf.* Fig. 4)

(i) $V' = V \cup \{x_1, x_2\} \cup Z$ where $Z = \{z_u : u \in V\}$,
(ii) $E' = E \cup \{(u, x_1) : u \in V\} \cup \{(x_1, x_2)\} \cup \{(x_2, z_u) : u \in V\}$.

Note that Z is an independent set of order $|V|$.

Claim 2. G' is a comparability graph.

Proof. Indeed, it is straightforward to deduce a transitive orientation of G' from an acyclic transitive orientation of G (*cf.* Fig. 5). □

Then, for any maximal independent set S of G, the set $S' = S \cup \{x_2\}$ is a wcis of G'. As $Z \cup \{x_1\}$ is the only wcis in G' which does not contain x_2, the minimum maximal independent set in G can be associated to the minimum weakly connected independent set in G' as above. Therefore, the MMIS problem and the MWCIS problem have the same complexity in the comparability graph class. □

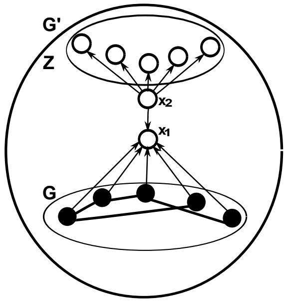

FIGURE 5. An orientation of $G^{'}$.

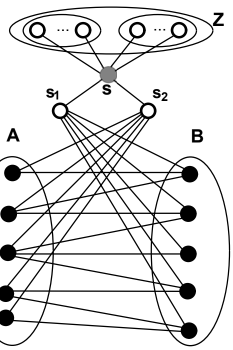

FIGURE 6. The graph G_1 obtained from G.

We consider now the graph class $\mathcal{B}_1$ which is slighly broader than $\mathcal{B}$ defined as follows. A connected graph $G = (V, E)$ belongs to $\mathcal{B}_1$ if G is a connected bipartite graph, or if there exists a node $u_0 \in V$ such that $G \setminus \{u_0\}$ is a connected bipartite graph.

Theorem 5.7. *Weakly Connected Independent Set is NP-complete in* $\mathcal{B}_1$*.*

Proof. The problem of determining whether a connected bipartite graph $G = (A \cup B, E)$ has a maximal independent set of size less than k was shown to be NP-complete in [19]. We transform a connected bipartite graph G into a graph $G_1 = (V_1, E_1)$ of $\mathcal{B}_1$ as follows (*cf.* Fig. 6).

(i) $V_1 = A \cup B \cup \{s, s_1, s_2\} \cup Z$, where $Z = \{z_u : u \in A \cup B\}$,
(ii) $E_1 = E \cup \{(s, s_1), (s, s_2)\} \cup \{(s_1, v) : v \in B\} \cup \{(s_2, u) : u \in A\} \cup \{(s, z_u) : u \in A \cup B\}$.

Note that $G_1 \setminus \{s_1\}$ and $G_1 \setminus \{s_2\}$ belong to $\mathcal{B}$. Any maximal independent set M in G corresponds with a weakly connected independent set $W = M \cup \{s\}$ in G_1. Furthermore, any wcis of G_1 that is included in $V_1 \setminus \{s\}$, contains the set Z and s_1 or s_2. And its cardinality is bigger than $|A| + |B| + 1$. Then, it is straightforward

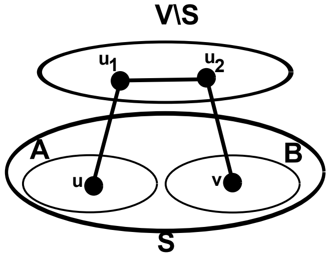

FIGURE 7. A partition (A,B) of S such that $d_G(A, B) = 3$.

to verify that G has a maximal independent set M such that $|M| \leq k$ if and only if G_1 has a weakly connected independent set W of size $k+1$ or less. □

As any wcis in a graph G is a mis in G, it can be interesting to study the following property.

Definition 5.8. (*wcis-property*) A connected graph G has the *wcis-property* if any maximal independent set in G is a weakly connected independent set.

Note that the cycle C_5 has the wcis-property whereas P_4 has not. Actually, we have not characterized these graphs, but we have the following result.

Lemma 5.9. *Let $G = (V, E)$ be an undirected connected graph. G and* all *its induced connected subgraphs have the wcis-property if and only if G is P_4-free.*

Proof. Let G be a P_4-free connected graph. Suppose that there exists a mis S which is not a wcis in G. From Lemma 3.2, there is a non empty subset A of S such that $l^* = \min\{d_G(u, v); u \in A, v \in S \setminus A\} \geq 3$ (*cf.* Fig. 7). As S is a dominating set, we have $l^* = 3$. Henceforth, the minimum length path $\{u, u_1, u_2, v\}$ between A and $S \setminus A$ induces a P_4, this yields a contradiction.

Now, if G and all its connected subgraphs verify the wcis-property, then G must be P_4-free since P_4 does not satisfy the wcis-property. □

Obviously, in P_4-free graphs, the problems of determining the minimum size mis and of finding the minimum size wcis have the same polynomial complexity [11].

6. AN IMPLICIT ENUMERATION ALGORITHM

In independent set problems, trivial algorithms that simply enumerate subsets of vertices and check for feasible solutions can be applied. Thus, all the solutions can be obtained in $O^*(2^n)$ (notation $O^*(.)$ is used to measure the complexity of an algorithm ignoring polynomial terms). But, it is possible to design algorithms that are significantly faster than exhaustive search, though still not polynomial [3, 14].

We present a $O^*(1.4655^n)$ time algorithm for solving the Minimum Weakly Connected Independent Set Problem. Actually, this result can be seen as a first step for directly obtaining the Minimum Weakly Connected Independent Set.

For an undirected connected graph $G = (V, E)$, let $n = |V|$ and $m = |E|$. Denote by $T(n)$ the worst case time for an algorithm to resolve an instance on at most n vertices. If someone can prove that computing a solution on an instance of n vertices is done in a running time which is at most the time for running a sequence of k instances of respective sizes $n - \alpha_1, \ldots, n - \alpha_k$, then one can write

$$T(n) \leq \sum_{i=1}^{k} T(n - \alpha_i) + p(n)$$

where $p(n)$ is a polynomial term. Thereafter, the running time $T(n)$ is bounded by $O^*(c^n)$ where the branching factor c is obtained as the maximum root of the equation $\sum_{i=1}^{k} \frac{1}{x^{\alpha_i}} = 1$.

Our enumeration algorithm is based on an implicit binary search tree [15].

An independent set W is said a *partial wcis* of G if the subgraph $G_W = (W \cup N(W), [W, N(W)])$ is connected. Obviously, if $W \cup N(W) = V$, then W is a wcis of G. A *completion* of a partial wcis W is a subset C of vertices in V for which $W \cup C$ is a wcis in G. We have the following easy lemma.

Lemma 6.1. *Any partial wcis of a connected graph can be completed.*

Along the enumeration procedure, each node of the tree is characterized by a *partial solution*. A partial solution L is an ordered list of vertices of V assigned to be in a partial wcis W_L or forbidden to be used in any completion of W_L. A forbidden vertex $u \in L$ is written as $\overline{u}$. Denote also by V_L the set $W_L \cup N(W_L)$ and by F_L the set of forbidden vertices stemming from L. A node belonging to $V \setminus (V_L \cup F_L)$ is called *free*. A free vertex v is *accessible* if $v \in N^2(u)$ for some $u \in W_L$. Let A_L be the set of accessible free vertices of V from W_L. Note that $W_L \cup \{v\}$ is a partial wcis for any $v \in A_L$. Thus, at a tree node, the decision is to add a vertex v_0 in a partial wcis or not. So the right subtree of a node is formed by all the wcis containing v_0 whereas the left subtree contains all the wcis *not* containing v_0.

A completion L' of a partial solution L is a list of vertices such that L is a prefix of L' and $A_{L'} = \emptyset$. Thus a partial solution L determines at most $2^{n-|V_L \cup F_L|}$ different completions. A completion L' is said *feasible* if $W_{L'} \setminus W_L$ is a completion of W_L, *i.e.* $W_{L'}$ is a wcis.

For example, let G be a graph with $V = \{u_1, u_2, u_3, u_4, u_5\}$, and $E = \{(u_1, u_2), (u_1, u_3), (u_2, u_3), (u_3, u_4), (u_3, u_5), (u_4, u_5)\}$. For $L = \{\overline{u}_1, u_2\}$, we have that $W_L = \{u_2\}$, $F_L = \{u_1\}$ and $A_L = \{u_4, u_5\}$. (see Fig. 8a). So, u_1 cannot belong to any completion of W_L and the subgraph $G_{W_L} = (\{u_2\} \cup \{u_1, u_3\}, \{(u_1, u_2), (u_2, u_3)\})$ is connected (see Fig. 8b). For the above example

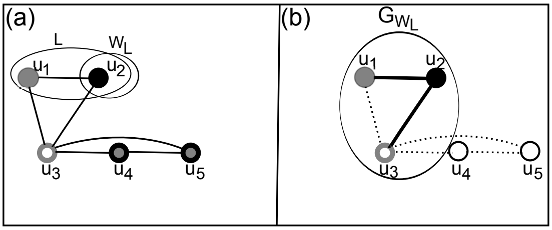

FIGURE 8. (a) $G = (V, E)$. (b) G_{W_L}.

there are three completions, the last one is not feasible:

$$\{\overline{u}_1, u_2, u_4\}, \{\overline{u}_1, u_2, \bar{u}_4, u_5\}, \{\overline{u}_1, u_2, \bar{u}_4, \bar{u}_5\}.$$

Our implicit enumeration algorithm involves generating a sequence of partial solutions. As the calculations proceed, feasible completions are discovered and the best one yet found is kept. At each step of the algorithm, characterized by a partial solution L, we try to add an accessible vertex v_0 to W_L, otherwise we fathom the node L. Then we make a backtrack at every fathoming.

Let us introduce some notations. For a subset $S \subseteq V$ and a node $v \in V$, we define $N_S(v) = N(v) \cap S$ and $d_S(v) = |N_S(v)|$, the S-degree of the node v.

6.1. Initialization

We choose a minimum degree vertex w_0. Let $N(w_0) = \{w_1, w_2, \ldots, w_{\delta(G)}\}$. Any wcis of G must contain w_0 or a neighbour of it. Indeed, $\mathcal{W}(G)$ can be partitionned in $\delta(G) + 1$ sets. Each of them are identified by an *initial* partial solution of the form:

$$L_0 = \{w_0\} \text{ or } L_0^k = \{\overline{w_0}, \overline{w_1}, \ldots, \overline{w_{k-1}}, w_k\}, \text{ for } 1 \leq k \leq \delta(G).$$

Our algorithm successively uses these $\delta(G) + 1$ partial solutions as initial lists.

6.2. An iteration

Denote by L a current partial solution. L can be fathomed in one of the following cases:

Fathoming condition $(F1)$ $V_L = V$,
Fathoming condition $(F2)$ $A_L = \emptyset$, and $V_L \neq V$,
Fathoming condition $(F3)$ $\exists u \in F_L, N(u) \subset N(W_L) \cup F_L$.

Indeed, Condition $(F1)$ indicates that W_L is a wcis, which may replace the best known solution if it is smaller. With Condition $(F2)$, L is an infeasible completion of the current initial list. A forbidden vertex verifying Condition $(F3)$ cannot be dominated in any completion L' of L, since $W_L \subset W_{L'}$ and $F_L \subset F_{L'}$. So we may backtrack.

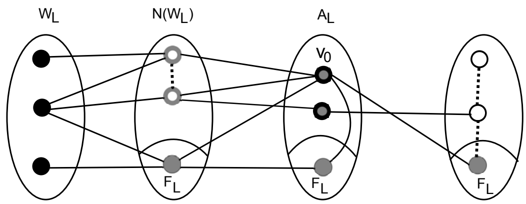

FIGURE 9. Branching on a vertex according to Rule ($R1$).

Suppose now that none of the above conditions is satisfied. We select an accessible node v_0 satisfying:

Rule ($R1$) $d_{V\setminus(V_L\cup F_L)}(v_0) = 0$,
Rule ($R2$) $d_{V\setminus(V_L\cup F_L)}(v_0) = 1$,
Rule ($R3$) $d_{V\setminus(V_L\cup F_L)}(v_0) \geq 2$.

6.2.1. Branching on a vertex according to Rule ($R1$)

Lemma 6.2. *Assume that an accessible node* v_0 *is not adjacent to any free vertex. Then we can add* v_0 *without branching.*

Proof. As v_0 is an accessible node whose neighboorhood is included in $N(W_L)\cup F_L$, any wcis extending W_L must contain this vertex (see Fig. 9). So the partial solution $L' = L \cup \{\overline{v_0}\}$ has no feasible completion. □

6.2.2. Branching on a vertex according to Rule ($R2$)

Lemma 6.3. *Assume that an accessible node* v_0 *is adjacent to exactly one free vertex. Then we can remove at least two vertices and* $T(p) \leq 2T(p-2)$. *Thus we obtain a branching factor* $\lambda \leq \sqrt{2} = 1.4142$.

Proof. Given a partial solution L, let v_0 be an accessible vertex such that $N(v_0)\setminus(N(W_L)\cup F_L) = \{x_0\}$. So, with $L' = L\cup\{v_0\}$ we have $T(p) \leq T(p-2)$. Assume now that v_0 is forbidden. Consider a partial solution L'' which admits $L \cup \{\overline{v_0}\}$ as prefix. Suppose that $W_{L''}$ cannot contain x_0, e.g. $x_0 \in F_{L''}$ or $x_0 \in N(W_{L''})$. As $N(v_0) \subset (N(W_L)\cup F_L\cup\{x_0\}) \subset (N(W_{L''})\cup F_{L''})$, v_0 satisfies Fathoming Condition (3) for L''. This implies that $T(p) \leq T(p-2)$. Finally, the branching gives $T(p) \leq 2T(p-2)$ (see Fig. 10). □

6.2.3. Branching on a vertex according to Rule ($R3$)

Consider a node v_0 satisfying Rule ($R3$). When we take v_0 in L, we must remove at least two free vertices (see Fig. (11)). So we get that $T(p) \leq T(p-3)+T(p-1)$. Here the branching factor λ is less than 1.4655. Therefore we get as an immediate consequence of Lemmata 6.2 and 6.3 the following theorem.

Theorem 6.4. *The implicit algorithm solves Minimum Weakly Connected Independent Set problem in polynomial space and in time* $O^*(1.4655^n)$.

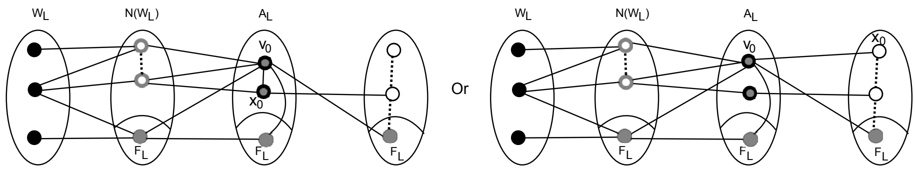

FIGURE 10. Branching on a vertex according to Rule ($R2$).

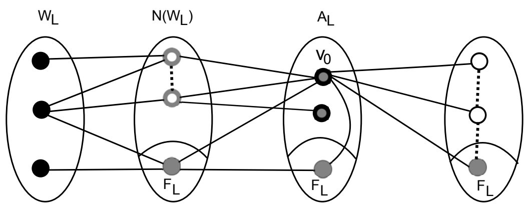

FIGURE 11. Branching on a vertex according to Rule ($R3$).

7. Computational experiment

In this section, we present our graph instances and discuss experimental results. The algorithms are implemented in C. All runs are performed on a Machine HP 8 CPU 2.7 Ghz, AMD Opteron QuadCore, with 256 Go of RAM in CentOS 5.5, running under Linux. We have fixed the maximum CPU time to 6 h.

7.1. Description of graph instances

We use three graph classes for our tests: graphs from the TSPLIB[2] library [27], random graphs and s-grid graphs.

Regarding the first class, we used the *node-coord-section* proposed by the TSPLIB library. For any node we fix a transmission range r. An edge between two nodes u and v is generated if the euclidian distance between u and v is less than r. For the random graphs, points are uniformly distributed in an unit square and links are created according to a transmission threshold. The number of nodes rises from 50 to 120 and the magnitude of the density D, given by $D = \frac{2*|E|}{|V|*(|V|-1)}$, is 10%.

The two-dimensional s-grid graph $G_{m\times n} = (V_{m\times n}, E_{m\times n})$ is defined as follows:

$$\begin{aligned} V_{m\times n} &= \{(i,j) | 1 \le i \le m, 1 \le j \le n\} \cup \{s\},\\ E_{m\times n} &= \{((i,j),(i,j+1)); 1 \le i \le m, 1 \le j \le n-1\}\\ &\cup \{\{(i,j),(i+1,j)\}; 1 \le i \le m-1, 1 \le j \le n\}\\ &\cup \{(s,(1,j)); 1 \le j \le n\}. \end{aligned}$$

s-grid graphs are not bipartite and belong to the class $\mathcal{B}_1$. They can model sensors dispersed on cultivable lands. These devices are generally arranged in the form of

[2]www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/.

a regular grid and they communicate with a base station (*i.e.* the node s) outside of the field.

Each instance is given by its name followed by an extension representing the number of nodes of the graph.

7.2. Results: heuristic procedures, tests and analysis

For a graph instance $G = (V, E)$, we denote by Opt the number of nodes of a MWCIS(G) built by the exact algorithm and by $\sharp Opt$ the total number of optimal solutions.

We also give the results of one heuristic, called H_{120s}. It is a modified version of the greedy routine of [28]. H_{120s} gives the best solution obtained after several runs of that greedy procedure during a lapse of two minutes. Accessible vertices are successively added *at random* in the current partial *wcis*.

As the deep first search method builds a feasible solution very quickly, we also keep the best *wcis* found after two minutes of processing time of the enumeration algorithm. This solution is denoted by $A_{120s}(G)$. The other entries of the various tables are:

- D: density of the graph $\left(D = \frac{2*|E|}{|V|*(|V|-1)}\right)$;
- CPU: running time in hours:min:sec;
- TNET: total number of nodes of the enumeration tree (in millions);
- NFOS: number of nodes of the enumeration tree for finding the first optimal solution;
- $\frac{NFOS}{TNET}$: indicates the share of the whole enumeration tree for finding the first optimal solution;
- Gap: the relative error between the optimal solution (when the problem has been solved to optimality) and the best heuristic solution, given by $Gap = \frac{\min(|A_{120s}(G)|,|H_{120s}(G)|)-Opt}{Opt}$.

Tables 1–3 summarize the results for the three graph classes.

First, we have to choose a transmission radius for each graph stemming from the $TSPLIB$ and random graphs. Table 1 shows that the enumeration algorithm can quickly solve instances whose number of nodes is less than 70 for any density. For higher cardinalities, the Minimum Weakly Connected Independent Set problem becomes easier when the number of edges increases in the graph, which is illustrated by Figures 14 and 15. Around a density of 6%−8%, instances exceeding 100 vertices are very hard to solve as it appears in Table 1 and Figure 15. The instances indicated with "*" in the Table 1 are those whose CPU time exceeded 6 hours. With a density of 10%, our exact algorithm can treat graphs up to one hundred of nodes in a reasonable time. The average size of minimum *wcis* in graphs with a fixed density $D \geq 10\%$ is relatively constant (Fig. 17). The CPU time grows up exponentially with the number of nodes (Fig. 16).

For Table 2, we generate ten occurrences for each cardinality and solve them to optimality with our enumeration program.

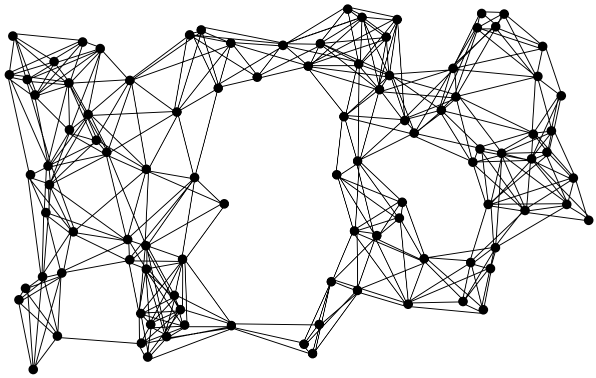

FIGURE 12. Instance kroC100 with density of 10%.

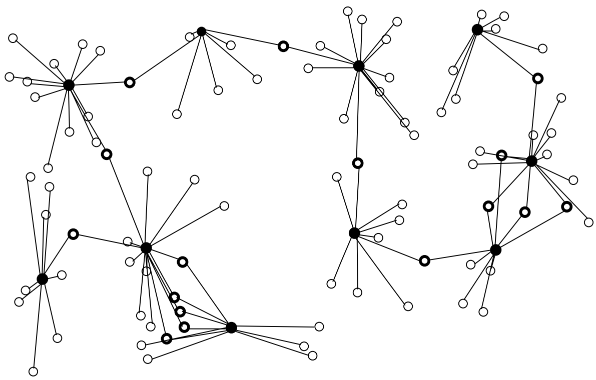

FIGURE 13. Optimal solution of the instance kroC100.

We can see that the *wcis* obtained after two minutes in a running of the enumeration algorithm is pretty good. It outperforms the solution given by H_{120s} in almost all tests. The gap between the optimal solution and the best heuristic result is within 14% at worst for graphs with less than 150 nodes. Thus, a quite satisfying solution can be very quickly obtained as in many combinatorial problems. For s-grid graphs, the enumeration algorithm discovers an optimal solution at the beginning of the tree, but it is facing major difficulties when the number of nodes rises. When they have more than 100 nodes, these graphs are very difficult examples for our algorithm (their density decreases lower than 4 %). In contrast, for 50% TSPLIB examples, more than 60% of the enumeration tree was needed for finding the first optimal solution. The situation for random graphs is somewhat median.

7.3. Running time comparison with indirect approaches

As a weakly connected independent set is a maximal stable set, any maximal independent set enumeration algorithm, combined with a connectivity test applied

Table 1. Exact algorithm and heuristic results for TSPLIB instances.

Instances	D	Opt	♯Opt	CPU	$\frac{NFOS}{TNET}$	A_{120s}	*Gap*	H_{120s}
eil51	8%	13	2	0:00:20	16%	13	0%	13
eil76	11%	12	834	0:00:20	0%	12	0%	13
pr76	19%	8	3230	0:00:02	7%	8	0%	8
kroA100	5%	20	39672	1:03:38	68%	21	5%	21
kroB100	7%	15	52	0:41:48	74%	16	7%	17
kroC100	5%	25	1248	0:03:14	0%	25	0%	26
kroD100	6%	18	1328	0:33:14	71%	19	6%	20
kroE100	7%	16	264	0:56:29	95%	17	6%	17
kroA100	10%	11	76596	0:12:28	16%	12	9%	12
kroB100	10%	11	1954	0:10:24	37%	12	9%	12
kroC100	10%	10	60	0:15:34	87%	11	10%	12
kroD100	10%	11	21074	0:13:58	2%	11	0%	12
kroE100	10%	11	14070	0:17:20	91%	12	9%	12
eil101	10%	12	8	0:31:11	65%	13	8%	15
lin105	16%	9	12824	0:00:58	0%	9	0%	9
ch130*	8%	15	–	6:00:00	–	18	7%	18
ch130	10%	12	154670	5:59:50	85%	13	8%	14
pr136	20%	6	376	0:00:16	25%	6	0%	7
pr144	13%	10	1644624	0:08:39	60%	11	10%	11
pr144	15%	7	787143	0:11:54	38%	8	14%	8
ch150*	4%	28	–	6:00:00	–	29	4%	29
ch150*	10%	11	8268	6:00:00	12%	12	9%	13
ch150	15%	8	47937	1:19:45	46%	9	12%	9
kroA150*	4%	34	–	6:00:00	–	35	0%	34
kroA150*	10%	11	85608	6:00:00	13%	12	9%	13
kroA150	15%	7	1440	0:20:52	11%	7	0%	8
kroB150*	5%	30	–	6:00:00	–	31	0%	30
kroB150*	10%	11	–	6:00:00	–	12	9%	13
kroB150	15%	7	6	0:54:04	90%	8	14%	9
pr152	30%	4	924	0:00:01	8%	4	0%	4
pr226	15%	8	11075899	1:00:50	63%	9	12%	9
pr226	20%	5	842	0:14:04	83%	7	20%	6

Table 2. Exact algorithm and heuristic results for random graphs.

Graphes	*D*	Opt	♯Opt	CPU	$\frac{NFOS}{TNET}$	A_{120s}	*Gap*	H_{120s}
Random50	10%	11.00	378	0:00:00	62%	–	–	–
Random60	10%	11.40	534	0:00:02	41%	–	–	–
Random70	10%	10.75	7399	0:00:05	32%	–	–	–
Random80	10%	10.66	4584	0:00:39	25%	–	–	–
Random90	10%	10.50	1819	0:02:20	40%	10.75	2%	11.75
Random100	10%	11.33	34499	0:14:44	30%	12.10	7%	12.50
Random110	10%	10.88	16925	0:36:51	35%	12.10	11%	12.44
Random120	10%	10.90	7985	1:58:34	42%	12.10	11%	12.70

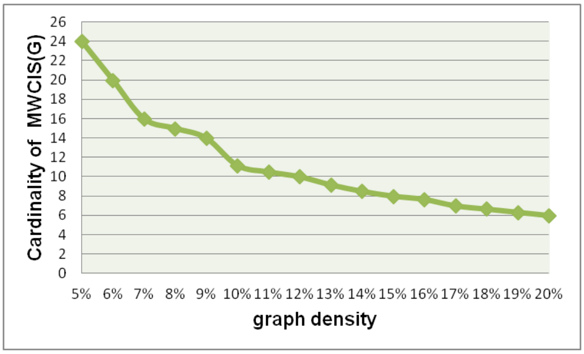

FIGURE 14. Average number of nodes of the minimum $WCIS(G)$ when $|V| = 120$ and $D_{\min} \leq D \leq 20\%$.

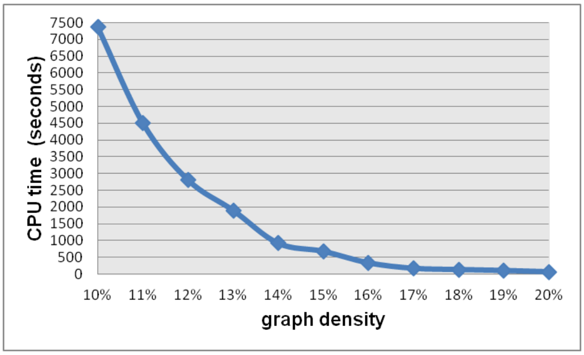

FIGURE 15. Average time of exact algorithm when $|V| = 120$ and $D_{\min} \leq D \leq 20\%$.

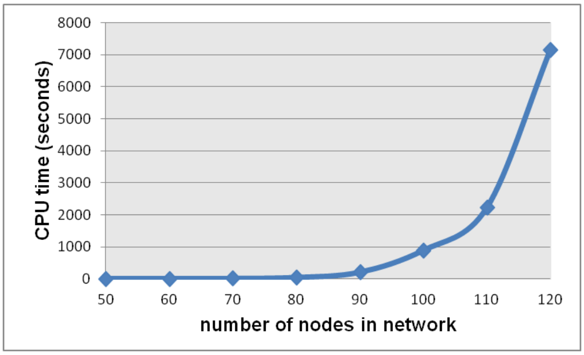

FIGURE 16. Average time of exact algorithm when $D = 10\%$.

TABLE 3. Exact algorithm and heuristic results for s-grid graphs.

s-Grids	$\|V\|$	Opt	♯Opt	CPU	$\frac{NFOS}{TNET}$	A_{120s}	Gap	H_{120s}
s-Grid$_{6\times12}$	73	21	802	0:00:23	2%	–	–	–
s-Grid$_{12\times6}$	73	24	140	0:00:12	5%	–	–	–
s-Grid$_{8\times9}$	73	21	4	0:00:21	56%	–	–	–
s-Grid$_{9\times8}$	73	22	196	0:00:18	8%	–	–	–
s-Grid$_{5\times16}$	81	22	396	0:02:50	0%	22	0%	24
s-Grid$_{16\times5}$	81	25	4	0:00:32	0%	25	0%	35
s-Grid$_{8\times10}$	81	24	574	0:02:04	3%	24	0%	29
s-Grid$_{10\times8}$	81	25	494	0:01:38	4%	25	0%	31
s-Grid$_{8\times11}$	89	25	12	0:11:00	23%	26	4%	31
s-Grid$_{11\times8}$	89	27	576	0:08:21	26%	28	4%	35
s-Grid$_{6\times16}$	97	27	5372	1:27:48	2%	28	4%	31
s-Grid$_{16\times6}$	97	32	936	0:21:29	5%	32	0%	41
s-Grid$_{5\times20}$	101	27	1592	4:55:38	21%	36	15%	31
s-Grid$_{20\times5}$	101	31	4	0:21:45	59%	32	3%	45
s-Grid$_{10\times10}$	101	30	1520	2:19:18	4%	32	3%	37

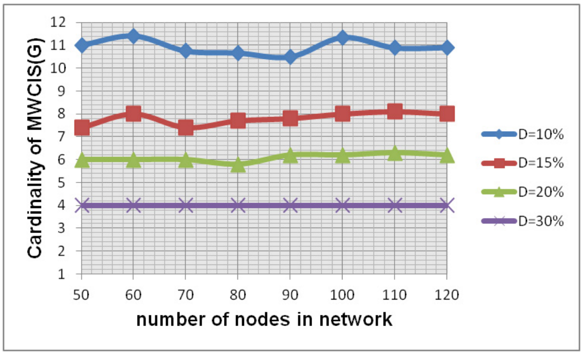

FIGURE 17. Average number of nodes of the minimum $WCIS(G)$ when $D \in \{10\%, 15\%, 20\%, 30\%\}$.

to each detected mis, can provide a way to search for a minimum wcis. We present here a comparison of our algorithm with the Laforest and Phan's experiments and the tests of [24, 26] stemming from [22].

Note that, for this subsection, our program runs on a machine Intel(R) Core (TM)2 Duo CPU at 3.00 GHz with 3.25 GB RAM.

Table 4 summarizes the running times (in seconds) on Grid Graphs from 5×5 to 8×8.

The comparison with the Laforest and Phan's algorithm[3] is given in Table 5. These tables show that our direct approach for the minimum weakly connected independent set problem is experimentally more efficient than an indirect method based on the implemented mis enumeration procedures from [22].

[3]The authors thank Raksmey Phan for gracefully lending his examples.

TABLE 4. Grid graphs: Running time comparison.

$\lvert V\rvert$	5×5	6×6	7×7	8×8
CPU	0	0	0.02	0.88
Laforest [22]	0	0	9.90	630
IEA [26]	1	254	141 242	–
Liu[24]	11	39 225	–	–

TABLE 5. Random graphs: Running time comparison.

$\lvert V\rvert$	80	90	100	110
CPU	9.95	118.47	284.48	563.89
Laforest [22]	740.20	8049.50	38 460.00	126 985.00

8. Conclusion

In this paper, we discussed the problem of determining the Minimum Weakly Connected Independent Set in graphs. We showed that the MWCIS problem is $NP-hard$ in general graphs, and studied its complexity in some well known graph classes. We also proposed the first exact algorithm designed specifically for the MWCIS problem whose time and space complexities are respectively $O^*(1.4655^n)$ and $O(n^2)$.

Experimental results point out that our implicit enumeration method can satisfactorily handle instances up to 120 nodes but that it has difficulty with sparse graphs from 100 nodes.

We believe that future works should focus on a decrease in the theoretical complexity of the wcis enumeration, and on the status of the Minimum Weakly Connected Independent Set Problem for s-grid graphs, which are practically hard to solve.

Acknowledgements. The authors would like to thank the anonymous referees for their valuable comments.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: A survey, *Comput. Networks* **38** (2002) 393–422.

[2] K.M. Alzoubi, P.J. Wan and O. Frieder, *Weakly Connected Dominating Sets and Spanners in Wireless Ad Hoc Networks. Proceedings of the 23rd International Conference on Distributed Computing Systems* (2003).

[3] N. Bourgeois, F. Della Croce, B. Escoffier and V.Th. Paschos, Fast algorithms for min independent dominating set, *Discrete Appl. Math.* **161** (2012) 558–572.

[4] M.S. Chang, Efficient algorithms for the domination problems on interval and circular-arc graphs, *SIAM J. Comput.* **27** (1998) 1671–1694.

[5] Y.P. Chen and A.L. Liestman, Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks, *The 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing* (2002).

[6] X. Cheng, X. Cheng, D.Z. Du and M. Cardei, Connected Domination in multihop Ad Hoc Wireless Networks. *Proceedings of the 6th International Conference on Computer Science and Informatics* (2002).

[7] B.N. Clark, C.J. Colbourn and D.S. Johnson, Unit Disk Graphs. *Discrete Math.* **86** (1990) 165–177.

[8] D.G. Corneil and Y. Perl, Clustering and domination in perfect graphs. *Discrete Appl. Math.* **9** (1984) 27–39.

[9] J.E. Dunbar, J.W. Grossman, J.H. Hattingh, S.T. Hedetniemi and A.A. McRae. On Weakly connected Domination in graphs, *Discrete Math.* **167-168** (1997) 261–269.

[10] M. Farber, Independent domination in chordal graphs, *Operation Res. Lett.* **1** (1982) 134–138.

[11] M. Farber and J. M. Keil, Domination in permutation graphs, *J. Algorithms* **6** (1985) 309–321.

[12] E. Fleury and D. Simplot-Ryl, *Réseaux de capteurs, théorie et modélisation.* Lavoisier (2009).

[13] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness. *Freeman, New York* (1979).

[14] S. Gaspers and M. Liedloff, A branch-and-reduce algorithm for finding a minimum independent dominating set in graphs, *Fomin, Lect. Notes Comput. Sci.* **4271** (2006) 78–89.

[15] A.M. Geoffrion, Integer programming by implicit enumeration and Balas' Method, *Siam Review* **9** (1967) 178–190.

[16] S. Guha and S. Khuller, Approximation algorithms for connected dominating sets, *Algorithmica* **20** (1998) 374–387.

[17] M.M. Halldórsson, Approximating the minimum maximal independence number, *Information Processing Lett.* **46** (1993) 169–172.

[18] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, Fundamentals of Domination in graphs: Advanced Topics, *Marcel Dekker, Inc.* (1998).

[19] R.W. Irving, On approximating the minimum independent dominating set, *Information Processing Lett.* **37** (1991) 197–200.

[20] B. Jeremy, D. Min, T. Andrew and C. Xiuzhen, Connected Dominating set in Sensor networks and manets, *Handbook of Combinatorial optimization. Springer.* (2004).

[21] D.S. Johnson, C.H. Papadimitriou and M. Yannakakis. On generating all maximal independent sets, *Inform. Processing Lett.* **27** (1988) 119–123.

[22] C. Laforest and R. Phan, *Experimentations on an exact algorithm for the minimum independent dominating set problem in graphs using clique partition. RAIRO* **47** (2013) 199–221.

[23] Y.S. Li, M.T. Thai, F. Wang, C.W. Yi, P.J. Wan and D.Z. Du, On Greedy Construction of Connected Dominating Sets in Wireless Networks, *J. Wireless Communications and Mobile Computing* **5** (2005) 927–932.

[24] C. Liu and Y. Song, Exact algorithms for finding the minimum independent dominating set in graphs. *ISAAC, Lect. Notes Comput. Sci.* **4288** (2006) 439–448.

[25] J.M. Moon and L. Moser, On cliques in graphs, *Israel J. Math.* **3** (1965) 23–28.

[26] A. Potluri and A. Negi, Some observations on algorithms for computing minimum independent dominating set, *IC3, Communications in Computer and Information Science*, **168** (2011) 57–68.

[27] G. Reinelt, TSPLIB-A Traveling Salesman Problem Library, *Informs J. Comput.* **3** (1991) 376–384.

[28] A.C. Santos, F. Bendali, J. Mailfert, C. Duhamel and K.M. Hou, *Heuristices for designing Energy-efficient Wireless Sensor Network Topologies. J. Networks* **4** (2009) 436–444.

[29] I. Stojmenovic, M. Seddigh and J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, *IEEE Transactions on parallel and distributed systems* **13** (2002) 14–25.

[30] P.J. Wan, K.M. Alzoubi and O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications* **9** (2004) 141–149.

[31] P.J. Wan, L. Wang and F. Yao, Two-Phased Approximation Algorithms for Minimum CDS in Wireless Ad Hoc Networks, *IEEE ICDCS* (2008) 337–344.

[32] W. Wu, H. Du, X. Jia, Y. Li and S.C.H. Huang, Minimum connected dominating sets and maximal independent sets in unit disk graphs, *Theoret. Comput. Sci.* **352** (2006) 1–7.