Independent $[1, 2]$-domination of grids via min-plus algebra

S.A. Aleid$^{1,3}$ J. Cáceres$^{2,3}$ M.L. Puertas$^{2,3}$

Department of Mathematics. University of Almería, Spain

Abstract

Domination of grids has been proved to be a demanding task and with the addition of independence it becomes more challenging. It is known that no grid with $m, n \geq 5$ has a perfect code, that is an independent vertex set such that each vertex not in it has exactly one neighbor in that set. So it is interesting to study the existence of an independent dominating set for grids that allows at most two neighbors, such a set is called independent $[1, 2]$-set. In this paper we develop a dynamic programming algorithm using min-plus algebra that computes the minimum cardinality of an independent $[1, 2]$-set for the grid $P_m \square P_n$.

Keywords: Domination, independence, grids, min-plus algebra.

1 Introduction

Let $G = (V, E)$ be a simple graph. A subset $S \subseteq V$ is called a dominating set of $G$ if every $v \in V \setminus S$ has at least one neighbor in $S$. Recall that the grid $P_m \square P_n$ is the cartesian product of paths $P_m$ and $P_n$. Domination in grids

---

1 Supported by Phoenix Project (Erasmus Mundus Programme). Corresponding author.
2 Partially supported by Junta de Andalucía FQM305 and MTM2014-60127-P.
3 Email: sahar.aleid@ual.es, jcaceres@ual.es, mpuertas@ual.es
has been extensively studied and the problem of determining the domination number \(\gamma(P_m \Box P_n)\), which is the minimum size of a dominating set of \(P_m \Box P_n\) was opened for almost 30 years, since it was first studied in [6]. In his 1992 Ph.D thesis Chang [1] proved that the domination number in grids is bounded by \(\lfloor \frac{(n+2)(m+2)}{5} \rfloor - 4\), for \(m, n \geq 8\). Chang also conjectured that equality holds for \(m = n\). In 1998 \(\gamma(P_m \Box P_n)\) was computed for \(m = 19\) and every \(n\) (see [8]). Finally in 2011, the problem was completely solved in [4] as authors were able to adapt the ideas in [5] to confirm Chang’s conjecture.

Independence is a property closely related to domination. A set \(S\) of vertices is called independent if no two vertices in \(S\) are adjacent. The independent domination number is denoted by \(i(G)\), which is the minimum cardinality of an independent dominating set for the graph \(G\). Recently in [3] the independent domination number has been computed for all grids.

A perfect code is an independent dominating set such that every vertex not in it has a unique neighbor in the set. It was proved in [7] that there exists no perfect code for grids \(P_m \Box P_n\), except in cases \(m = n = 4\) and \(m = 2, n = 2k + 1\). This leads us to work with independent \([1, 2]\)-sets, that is an independent vertex set \(S\) such that every \(v \in V \setminus S\) is adjacent to at least one but not more than 2 vertices in \(S\) (see [2]). We solve the open problem proposed in [2] about the existence of independent \([1, 2]\)-set in grids and we also compute the independent \([1, 2]\)-number \(i_{[1,2]}(P_m \Box P_n)\) which is the minimum cardinality of such a set.

2 Algorithm

We present a dynamic programming algorithm to obtain \(i_{[1,2]}(P_m \Box P_n)\), following the ideas in [3,8]. Consider \(P_m \Box P_n\) as an array with \(m\) rows, \(n\) columns and vertex set \(\{v_{ij} : 1 \leq i \leq m, 1 \leq j \leq n\}\). Let \(S\) be an independent \([1, 2]\)-set of \(P_m \Box P_n\). We define a labeling of vertices of \(P_m \Box P_n\) associated to \(S\) as follows

\[
l(v_{ij}) = \begin{cases} 
0 & \text{if } v_{ij} \in S \\
1 & \text{if } v_{ij} \notin S \text{ and } |\{v_{i(j-1)}, v_{(i-1)j}, v_{(i+1)j}\} \cap S| = 1 \\
2 & \text{if } v_{ij} \notin S \text{ and } |\{v_{i(j-1)}, v_{(i-1)j}, v_{(i+1)j}\} \cap S| = 2 \\
3 & \text{if } v_{ij} \notin S \text{ and } |\{v_{i(j-1)}, v_{(i-1)j}, v_{(i+1)j}\} \cap S| = 0
\end{cases}
\]

Given an independent \([1, 2]\)-set, we can identify each vertex with its label so we obtain an array of labels with \(m\) rows and \(n\) columns. Hereinafter, given an independent \([1, 2]\)-set of \(P_m \Box P_n\), the columns of the grid are words
of length $m$ in the alphabet $\{0, 1, 2, 3\}$ and the number of zeros in the array is the cardinality of the independent $[1, 2]$-set. The algorithm considers all the arrays of words (as columns) that come from some independent $[1, 2]$-set of $P_m \square P_n$ and it calculates the minimum among the number of 0's of each array. This minimum is equal to $i_{[1,2]}(P_m \square P_n)$.

It is clear that not every word of length $m$ can belong to such labeling, for instance there can not be consecutive 0's in a word, because of independence. The first objective is to identify the words that belong to the labeling associated to some independent $[1, 2]$-set, that we will call suitable words.

We need to have in mind the not every suitable word can be in the first column nor in the last one. For instance a word with the sequence 120 can not be in the first column, because the vertex labeled with 2 does not follow the labeling rules, in the absence of a previous column. Also in the last column can not be vertices with label 3, because they would not be dominated. The second task is to identify those words that can be in the first column and those words that can be in the last one.

It is also clear that not any two suitable words can follow each other in a labeling associated to some independent $[1, 2]$-set, for instance if a word has 3 in the $r^{th}$ position, then the following word must have 0 in the $r^{th}$ position, to preserve domination. The last objective regarding to words is to identify which of them can follow a given one.

The final part of the algorithm calculates the minimum number of zeros in an array, among all possible arrays of labels associated to an independent $[1, 2]$-set. This can be done by the successive addition of columns, beginning with just one and ending with $n$.

2.1 The suitable words

Let $S$ be an independent $[1, 2]$-set of $P_m \square P_n$ and consider the associated labeling. It is clear that a column can not contain two consecutive 0's because of independence, nor two consecutive 2's because in the previous column should be two consecutive 0's, also it can not contain two consecutive 3's because in the following column will be two consecutive 0's. Moreover a column can not contain any of the sequences 03, 30, 010 by definition of the labeling. If a column contains the sequence 11 then the previous label or the following label in the column (or both) must be 0, because in other case would be two consecutive 0's in the previous column. If a column contains the sequence 32 then the other label next to 2 in the column must be 0, by definition of the labeling.
Similarly sequence 23 must be preceded by 0 and both sequences 21, 12 must be placed between two 0’s. Words satisfying all those rules are called suitable. We denote the cardinal of the set of all suitable words by \( k \), so every suitable word can be identified with \( p \in \{ 1, \ldots, k \} \).

2.2 The first column, the last column and the initial vector

Let \( S \) be an independent \([1, 2]\)-set of \( P_m \square P_n \) and consider the associated labeling. Note that the first column is a suitable word such that every 2 is placed between two 0’s and every 1 is preceded or followed (but not both) by 0, because in other case the labeling would not follow the definition. A suitable word that satisfies these properties is called initial.

Similarly a word can be placed in the last column if it does not contain any 3, because in other case one vertex would be not dominated. We call this word final.

Finally we define the initial vector \( X^1 = (X^1(1), X^1(2), \ldots, X^1(k)) \) which is a vector of size \( k \) such that for every word \( p \in \{ 1, 2, \ldots, k \} \), \( X^1(p) \) is number of zeros of \( p \), if it is an initial word, and \( X^1(p) = \infty \) in other case.

2.3 The rules for adding a column and the transition matrix

Let \( S \) be an independent \([1, 2]\)-set of \( P_m \square P_n \) and consider the associated labeling. Note that not every pair of suitable words can be consecutive columns, in order to preserve both independence and \([1, 2]\)-domination. Next we show the conditions needed to ensure that word \( p = p_1 \ldots p_m \) can follow word \( q = q_1 \ldots q_m \), for the case \( q_i = 0 \).

If \( q_i = 0 \) then

\[
\begin{align*}
\text{if } i = 1 \quad & \begin{cases}
p_1 = 1, & p_2 \neq 0 \text{ or } \\
p_1 = 2, & p_2 = 0
\end{cases} \\
\text{if } 1 < i < m \quad & \begin{cases}
p_i = 1, & p_{i+1} \neq 0, \ p_{i-1} \neq 0 \text{ or } \\
p_i = 2, & p_{i+1} \neq 0, \ p_{i-1} = 0 \\
p_i = 2, & p_{i+1} = 0, \ p_{i-1} \neq 0
\end{cases} \\
\text{if } i = m \quad & \begin{cases}
p_m = 1, & p_{m-1} \neq 0 \text{ or } \\
p_m = 2, & p_{m-1} = 0
\end{cases}
\end{align*}
\]

Conditions can be defined similarly for cases \( q_i = 1, q_i = 2 \) and \( q_i = 3 \).

The transition matrix is the square matrix \( A \) of size \( k \) such that, for every pair \( p, q \in \{ 1, 2, \ldots, k \} \), the entry \( A_{pq} \) in row \( p \) and column \( q \) is defined to be the number of zeros of the word \( p \) if \( p \) can follow \( q \), and \( A_{pq} = \infty \) otherwise.
2.4 Adding many columns via the min-plus multiplication

Using the particular matrix multiplication in \((\min, +)\)-algebra see [3], we can successively obtain vectors \(X^2 = A \boxtimes X^1, \ldots, X^n = A \boxtimes X^{(n-1)}\). For every word \(p \in \{1, \ldots, k\}\), the entry \(X^n(p)\) is the minimum number of zeros in an independent vertex subset of \(P_m \square P_n\), which has word \(p\) in the \(n^{th}\)-column and \([1, 2]\)-dominates the graph, except for may be some vertices in the last column with label 3. This allows us to obtain the following general formula

\[
i_{[1, 2]}(P_m \square P_n) = \min\{X^n(p) : p \text{ is a suitable final word}\}.
\]

Note that if the minimum is not infinite then there exists an independent \([1, 2]\)-set in the grid \(P_m \square P_n\).

2.5 The recursion rule to calculate \(i_{[1, 2]}(P_m \square P_n)\)

The above procedure allows us to obtain the value of \(i_{[1, 2]}(P_m \square P_n)\) for fixed \(m\) and \(n\). However a recurrence argument gives a formula for \(i_{[1, 2]}(P_m \square P_n)\) for fixed \(m\) and any \(n \geq m\). We use the following result which is similar to Theorem 2.2 in [3].

**Theorem 2.1** Suppose that there exist integers \(n_0, c, d > 0\) satisfying the equation \(X^{n_0 + d}(p) = X^{n_0}(p) + c\), for every suitable word \(p\). If \(P_m \square P_r\) has an independent \([1, 2]\)-set for \(n_0 \leq r \leq n_0 + d - 1\), then

(i) \(P_m \square P_n\) has an independent \([1, 2]\)-set, for all \(n \geq n_0\).
(ii) \(i_{[1, 2]}(P_m \square P_{n+d}) = i_{[1, 2]}(P_m \square P_n) + c\), for all \(n \geq n_0\).

Now let \(m\) be fixed and consider the suitable words of length \(m\) and the initial vector \(X^1\). Then apply the operation \(\boxtimes\) to calculate vectors \(X^i\), until obtaining a natural number \(n_0\) such that \(X^{n_0 + d} = c \boxtimes X^{n_0}\). Using the above theorem we get the finite difference equation \(i_{[1, 2]}(P_m \square P_{n_0 + d}) - i_{[1, 2]}(P_m \square P_{n_0}) = c\). The unique solution obtained from solving this equation for \(n \geq n_0\), is the desired formula for \(i_{[1, 2]}(P_m \square P_n)\).

2.6 Results

Applying the construction described in the previous section we have obtained that the independent \([1, 2]\)-number agrees with the independent dominating
number in almost every grid of small size $1 \leq m \leq 13, n \leq n$

$$i_{[1,2]}(P_m \Box P_n) = \begin{cases} 
  i(P_m \Box P_n) + 1 & \text{if } m = 12, n \equiv 10(\text{mod } 13) \\
  i(P_m \Box P_n) & \text{otherwise}
\end{cases}$$

We would like to point out that applying the algorithm for large values of $m, n$ is possible but the running time needed is extremely long. However the calculation of $i_{[1,2]}(P_m \Box P_n)$ for $14 \leq m \leq n$ can be solved using a constructive regular pattern, following the ideas in [1,3]. We have obtained in these cases

$$i_{[1,2]}(P_m \Box P_n) = \begin{cases} 
  i(P_m \Box P_n) & \text{if } m = 14, 15, m \leq n \\
  \gamma(P_m \Box P_n) & \text{if } 16 \leq m \leq n
\end{cases}$$

References


