computer sciences and mathematics forum

MDPI

Proceeding Paper

A Branch and Bound Algorithm for Counting Independent Sets on Grid Graphs †

GuillermoDe Ita *, Pedro Bello and Mireya Tovar

Faculty of Computer Science, Benemérita Universidad Autónoma de Puebla, Chignahuapan 73300 , Mexico; email1@email.com (P.B.); email2@email.com (M.T.)

* Correspondence: deitaluna63@gmail.com

† Presented at the 1st International Online Conference on Mathematics and Applications; Available online: https://iocma2023.sciforum.net/.

Abstract: The problem of counting independent sets of a graph G, denoted by $i(G)$, is not only mathematically relevant and interesting, but also has many applications in physics, mathematics, and theoretical computer science. Regarding hard counting problems, the computation of $i(G)$ for a graph G has been a key for determining the frontier between efficient counting and intractable counting procedures. In this article, a novel algorithm for counting independent sets on grid-like structures is presented. We propose a novel algorithm for the computation of $i(G_{m,n})$ for a grid graph with m rows and n columns based on the 'Branch and Bound' design technique. The splitting rule in our proposal is based on the well-known vertex reduction rule. The vertex in any subgraph from $G_{m,n}$ to be selected for the reduction rule must have 4 internal incident faces. The ramification process builds a computation tree. Our proposal consists of decomposing the initial grid graph until obtaining outerplanar graphs as the 'basic subgrids' associated to the leave nodes of the computation tree. The resulting time-complexity of our proposal for computing the number of independent sets for grid graphs is dramatically inferior to the time-complexity that the classic transfer matrix method requires for computing the same value.

Keywords: grid graphs; graph decomposition; transfer matrix method; branch and bound technique; counting independent sets

Citation: Ita, G.D.; Bello, P.; Tovar, M. A Branch and Bound Algorithm for Counting Independent Sets on Grid Graphs. *Comput. Sci. Math. Forum* **2023**, *1*, 0. https://doi.org/

Academic Editor: Firstname Lastname

Published: 28 April 2023

1. Introduction

Counting has become an important area in mathematics, as well as in computer science, even though it has received less attention than decision problems. This has caused less knowledge about the complexity of counting problems compared to the study of the complexity of decision problems. From the computational point of view, the counting of independent sets of a graph is a determining factor to establish the border between efficient counting and intractable counting procedures.

The study of the number of independent sets on grid structures $G_{m,n}$ is closely related to the "hard-square model" used in statistical physics and, of particular interest is the so-called hard-square entropy constant [1]. It has several applications in statistical physics [2,3], e.g., computation in the Potts and hardcore lattice gas model and the problem of counting q-particle Widom-Rowlinson configurations in graphs, where $q > 2$.

Euler [4] presents a method to calculate the Fibonacci number of a grid graph, calculating a new parameter $b(m, n)$ using the *transfer matrix method*. Calkin [5] calculates the number of independent sets on grid graphs using the *transfer matrix method*. In [6] an extension of the *transfer matrix method* is introduced in order to count the number of satisfying assignments of Boolean formulas in 2-CNF.

On the other hand, Merrifield and Simmons showed the correlation between the number of independent sets of G, denoted $i(G)$, and the boiling points of the molecular

graph represented by G [7]. This is one of the main reasons why the number of independent sets of a graph G, in mathematical chemistry, is called the Merrifield-Simmons topological index (M-S) of G. However, in graph theory, $i(G)$ is called the Fibonacci number of G.

A graph invariant is any function on a graph that does not depend on a labelling of its vertices. A topological index is a graph invariant applicable in chemistry. By IUPAC terminology, a topological index is a numerical value associated with chemical constitution purporting for correlation of chemical structure with various physical properties, chemical reactivities, or biological activities.

The branch and bound paradigm has been widely used to attack problems with an intrinsic combinatorial exponential character. This method allows us to branch the original problem into similar subproblems but with lower input parameters than the original problem. This branching process is continued by forming a computation tree where its leaves have associated instances derived from the original problem, but where such instances can be solved efficiently.

In this article, we present a branch and bound algorithm in order to compute the Merrifield-Simmons topological index of a grid $G_{m,n}$. Our proposal provides an exact algorithm that can be applied to any type of grid, whether regular, irregular or with non-squares internal faces. The resulting time-complexity of our proposal for computing $i(G_{m,n})$ is dramatically inferior to the time-complexity that the classic transfer matrix method spend in the computation of $i(G_{m,n})$.

2. Preliminares

Let $G = (V, E)$ be an undirected graph with a set of vertices V and a set of edges E. Two vertices v and w are adjacent if there is an edge $\{v, w\} \in E$ connecting them. Sometimes, we denote an edge $\{v, w\} \in E$ in abbreviated form as vw.

The neighborhood of $x \in V$ is the set $N(x) = \{y \in V : \{x, y\} \in E\}$, and its closed neighborhood is $N(x) \cup \{x\}$, which is denoted by $N[x]$. The cardinality of a set A is denoted by $|A|$. The degree of a vertex x in the graph G, denoted by $\delta(x)$, is $|N(x)|$, and the degree of the graph G is $\Delta(G) = max\{\delta(x) : x \in V\}$.

For a graph $G = (V, E)$, $S \subseteq V$ is an independent set of G, if for every $v_1, v_2 \in S$, it holds that $\{v_1, v_2\} \notin E$. $I(G)$ denotes the set of all independent sets of G. An independent set $S \in I(G)$ is "maximal" if it is not a subset of a larger independent set, and it is "maximum" if it has the largest size among all independent sets in $I(G)$.

Counting problems are not only mathematically interesting, but they also arise in many applications. Regarding hard counting problems, the computation of the number of independent sets of a graph G has been a key in determining the frontier between efficient counting and intractable counting algorithms.

The corresponding counting problem on independent sets, denoted by $i(G)$, consists of counting the number of independent sets of a graph G. The computation of $i(G)$ is a #P-complete problem for graphs G, where $\Delta(G) \geq 3$ [2,8,9]. There are several polynomial procedures to compute $i(G)$ when $\Delta(G) \leq 2$ [10–13]. All of them are methods of linear complexity with respect to the time.

Planar graphs play an important role both in the graph theory and in the graph drawing area. A graph G is planar if G admits an embedding in the plane. A planar drawing partitions the plane into connected regions called faces. The unbounded face is usually called outer face or external face. If all the vertices are incident to the outer face of the graph G then it is called an outerplanar graph. Given an embedding of a planar graph G, we differentiate as external vertices to all vertex of G incident to the outer face, otherwise it is an internal vertex of G.

A planar graph G has a set of closed non-intersected regions $F(G) = \{f_1, \ldots, f_k\}$, called internal faces (or just faces). Each face $f_i \in F(G)$ is represented by the set of edges that bound its inside area. We do not consider the outer face of the graph in $F(G)$, because we want to consider in $F(G)$ only the internal faces of G. Two faces $f_i, f_j \in F(G)$ are

adjacent if they have common edges, otherwise, they are independent faces. Notice that two independent faces can have common vertices, but they do not have common edges.

An special planar graph is the called grid graph. A grid graph of size mxn is a graph $G = (V, E)$ with vertex set $V = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq n\}$, and edge set $E = \{((j, i), (j + 1, i))|1 \leq j < m, 1 \leq i \leq n\} \cup \{((j, i), (j, i + 1))|1 \leq j < m, 1 \leq i < n\}$. In our case, we denote a grid graph $G_{m,n}$ where m is the number of rows and n is the number of columns, and let $k = m * n$ be the number of internal faces (tiling) in $G_{m,n}$.

There is a large volume of literature devoted to count structures in a grid graph, e.g., spanning trees, Hamiltonian cycles, independent sets, acyclic orientations, k-coloring, and so on [4–6,14]. Applications of the counting objects on grids also include for instance tiling and efficient coding schemes in data storage [15].

The classical method for computing the number of independent sets on grid graphs is based on the transfer matrix method [4,5]. The transfer matrix method consists of building an initial matrix of F_{m+2} rows and F_{m+2} columns that are indexed by $(m + 1)$-vectors of zeros and ones, and where F_{m+2} is the $m + 2$-th Fibonacci number.

3. A Branch and Bound Algorithm

Some reduction rules have been useful to count combinatorial objects on graphs. Particularly, the following rules are commonly used for counting independent sets:

1. Vertex reduction rule: let $v \in V(G)$,

$$i(G) = i(G - v) + i(G - (N[v]))$$

2. Edge division rule : let $e = \{x, y\} \in E(G)$,

$$i(G) = i(G - e) - i(G - (N[x] \cup N[y]))$$

On the other hand, $i(G) = \prod_{i=1}^{k} i(G_i)$ where $G_i, i = 1, \ldots, k$ are the connected components of G, then the total time complexity for computing $i(G)$, denoted as $T(i(G))$, is given by the maximum rule as $T(i(G)) = max\{T(i(G_i)) : G_i$ is a connected component of $G\}$. Thus, a first helpful decomposition of the graph is done via its connected components, and from here on, we consider as an input graph only one connected component.

We have designed a typical branch and bound algorithm in order to count the number of independent sets from a grid graph that we denote as BB algorithm. BB builds a computation tree. Into the ramification processes of the computation tree, BB considers two main points: the criterion to choose a vertex v for applying the vertex division rule, and a halting criterion to stop the branching on any node of the computation tree. The selected vertex v from any subgraph in order to apply the vertex division rule, holds:

- v is incident to four internal faces from the current graph.
- One of the internal faces incident to v has a maximum size with respect to the other internal faces in the subgraph, or it is incident to at least one face with vertices in the outerface.

The application of the division rule on the vertex v builds two new child nodes v_1 and v_2 from the current node in the computation tree. The subgraph associated to v_1 is formed as $G_1 = G - \{v\}$, and the subgraph associated to v_2 is formed as $G_2 = G - N[v]$.

At this stage, we have a similar problem for each subgraph $G_i, i = 1, 2$ that we had with the original grid G. If we solve the problem in a recurrent way and r_i is its solution, then the complete solution for $r = i(G)$ is $r = r_1 + r_2$. The above procedure determines a enumerative tree whose leaves correspond to the basic subgraph instances.

This process of ramification iterates until obtain as subgraph associated to each child nodes base instances G_p. The main characteristic on the topology of any base instance G_p is that there not exists a vertex $u \in V(G_p)$ that is incident to 4 internal faces, and in this case, we have achieved a child node for the enumerative tree. In Figure 1, an example of the computation tree formed by the algorithm BB can be seen for an initial input grid $G_{4,5}$.

We have shown that for those basic prime graph G_p, $i(G_p)$ can be computed in linear time on its number of edges, since we can consider those basic prime graph as outerplanar graphs [16,17]. Let us define $H(E_G) = \{G_p : G_p$ is the graph associated to a leaf node of the previous enumerative tree. $\}$. After E_G has been built, we have that $i(G) = \Sigma_{G_p \in H(E_G)} i(G_p)$. As the computation for each $i(G_p)$ can be done in linear time, then the complexity time for $i(G)$ lies on the time complexity time of the number of nodes of the enumerative tree.

Let $k = n \cdot m$ be the number of internal tiling faces existing on the initial grid $G_{m,n}$. Then, the complexity time of the branch and bound procedure depends on the recurrence $i(G) = i(G - \{u\}) + i(G - N[u])$. This splitting rule has different cases according of the number of internal faces incident to the vertices in $N[u]$. In the best case, $N[u]$ can be incident to 8 tiling faces and in this case, the decomposition rule expressed by the number of rectangles that are decomposed is given by the recurrence: $T(k) = T(k-8) + T(k-3)$.

But in the worst case, if G does not correspond to a basic case, then $N[u]$ should be incident to at least 5 internal tiling. And in this case, the splitting rule expressed by the number of rectangles that are decomposed is given by the recurrence:

$$T(k) = T(k-5) + T(k-3) \tag{1}$$

We seek a solution of the form $x^k = T(k)$. Substituting this into the previous recurrence relation (1) leads to the characteristic polynomial: $P(x) = x^5 - x^2 - 1$, whose 5 roots $r_i, i = 1, \ldots, 5$ give rise to solutions of the form $T(k) = r_i^k$.

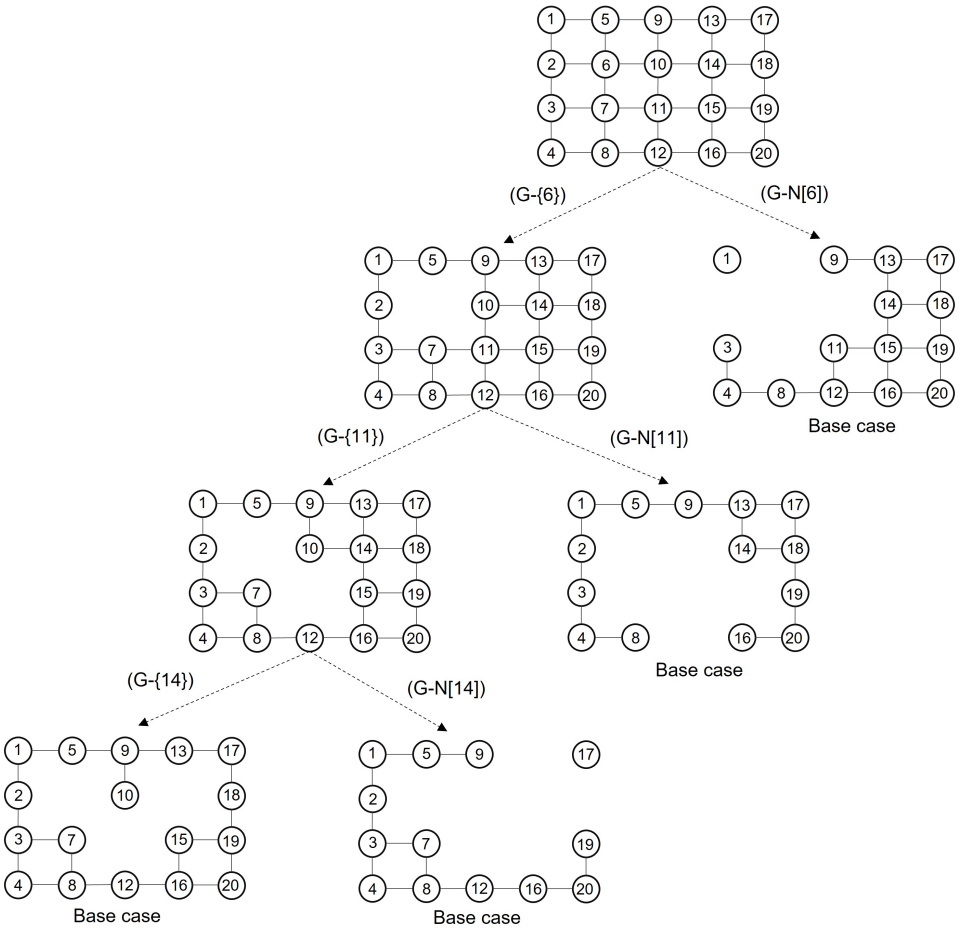

Figure 1. Processing the grid $G_{4,5}$.

As we are interested in the asymptotic behavior of the recurrence $T(k)$ we only consider the real root r_1 such that $|r_1| \geq |r_i|, i = 2, \ldots, 5$. In this case, the maximum real root is $r_1 \approx 1.194$ and, then we obtain a worst-case upper bound of $O(r_1^k \cdot poly(k))$, where $poly(k)$

is a polynomial function that corresponds to the time processing of the basic case graphs. Thus, the complexity-time of our proposal has an upper bound of $O(1.1939^k \cdot poly(k)) = O(1.1939^{m \cdot n} \cdot poly(k))$.

Our proposal works without any problem on irregular grids, or on variants of grids as the aztec diamonds graphs [18], since it lies of looking for vertices incident to more than three internal faces. On the other hand, although our proposal continues having an exponential time-complexity, it does not have the explosive combinatorial character that the classic transfer matrix method has for the computation of $i(G_{m,n})$.

4. Conclusions

We have introduced a branch and bound algorithm for the computation of $i(G_{m,n})$, for a grid graph with m rows and n columns. We apply as splitting rule the well-known vertex reduction rule. The vertex in any subgraph from $G_{m,n}$ to be selected for the reduction rule must be incident to 4 internal faces. Our proposal consists of decomposing the initial grid graph until obtaining as basic cases only outerplanar subgraphs.

The resulting time-complexity of our proposal for computing the number of independent sets for grid graphs is dramatically inferior to the time-complexity that the classic transfer matrix method requires for computing the same value.

Author Contributions:

Funding:

Author Contributions:

Institutional Review Board Statement:

Informed Consent Statement:

Data Availability Statement:

Acknowledgments: The authors give thanks to Conacyt-México for the economical support for this researching.

Conflicts of Interest:

References

1. Baxter, R.J. Planar lattice gases with nearest-neighbor exclusion. *Ann. Comb.* **1999**, *3*, 191–203. https://doi.org/10.1007/bf01608783.
2. Vadhan, S.P. The Complexity of Counting in Sparse, Regular, and Planar Graphs. *SIAM J. Comput.* **2001**, *31*, 398–427, [https://doi.org/10.1137/S0097539797321602]. https://doi.org/10.1137/S0097539797321602.
3. Dyer, M.E.; Greenhill, C.S. Corrigendum: The complexity of counting graph homomorphisms. *Random Struct. Algorithms* **2004**, *25*, 346–352. https://doi.org/10.1002/rsa.20036.
4. Euler, R. The Fibonacci number of a grid graph and a new class of integer sequences. *J. Integer Seq. [Electron. Only]* **2005**, *8*, 1–16.
5. Calkin, N.J.; Wilf, H.S. The Number of Independent Sets in a Grid Graph. *SIAM J. Discret. Math.* **1998**, *11*, 54–60, [https://doi.org/10.1137/S089548019528993X]. https://doi.org/10.1137/S089548019528993X.
6. Guillen, C.; Lopez, A.L.; DeIta, G. Computing #2-SAT of Grids, Grid-Cylinders and Grid-Tori Boolean Formulas. In Proceedings of the 15th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, Udine, Italy, 12–13 December 2008; Gavanelli, M., Mancini, T., Eds.; CEUR-WS.org; CEUR Workshop Proceedings; 2008, Volume 451, pp. 152–167.
7. Merrifield, R.E.; Simmons, H.E. *Topological Methods in Chemistry*; Wiley: New York, NY, USA, 1989.
8. Greenhill, C.S. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. *Comput. Complex.* **2000**, *9*, 52–72.
9. Luby, M.; Vigoda, E. Approximately Counting up to Four (Extended Abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*; STOC '97; Association for Computing Machinery: New York, NY, USA, 1997; pp. 682–687. https://doi.org/10.1145/258533.258663.
10. Bubley, R. *Randomized Algorithms: Approximation, Generation, and Counting*; BAR International, Springer: London, UK, 2001.
11. Dahllöf, V.; Jonsson, P. An Algorithm for Counting Maximum Weighted Independent Sets and Its Applications. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*; SODA '02; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002; p. 292–298.

12. Roth, D. On the hardness of approximate reasoning. *Artif. Intell.* **1996**, *82*, 273–302. https://doi.org/https://doi.org/10.1016/0004-3702(94)00092-1.
13. De Ita, G.; Contreras, M.; Bello, P.; Moyao, Y.; López, C. Recognition of graph topologies for the efficient counting of its number of independent sets. In *Combinational Algorithms and Learning*; Zacarias, F., Moyao, Y., Contreras, M., De Ita, G., Bello, P., Eds.; Montiel & Soriano: 2017; pp. 33–44.
14. Golin, M.J.; Leung, Y.; Wang, Y.; Yong, X. Counting Structures in Grid Graphs, Cylinders and Tori Using Transfer Matrices: Survey and New Results. In Proceedings of the Seventh Workshop on Algorithm Engineering and Experiments and the Second Workshop on Analytic Algorithmics and Combinatorics, ALENEX /ANALCO 2005, Vancouver, BC, Canada, 22 January 2005; Demetrescu, C.; Sedgewick, R.; Tamassia, R., Eds.; SIAM, 2005, pp. 250–258.
15. Roth, R.; Siegel, P.; Wolf, J. Efficient coding schemes for the hard-square model. *IEEE Trans. Inf. Theory* **2001**, *47*, 1166–1176. https://doi.org/10.1109/18.915673.
16. DeIta, G.; Rodríguez, M.; Bello, P.; Contreras, M. Basic Pattern Graphs for the Efficient Computation of Its Number of Independent Sets. In Proceedings of the Pattern Recognition-12th Mexican Conference, MCPR 2020, Morelia, Mexico, 24–27 June 2020, Proceedings; Figueroa-Mora, K.M., Anzurez-Marin, J., Jacobo, J.C., Carrasco-Ochoa, J.A., Trinidad, J.F.M., Olvera-López, J.A., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2020; Volume 12088, pp. 57–66. https://doi.org/10.1007/978-3-030-49076-8_6.
17. Medina, M.A.L.; Marcial-Romero, J.R.; Luna, G.D.I.; Moyao, Y. A Linear Time Algorithm for Computing #2SAT for Outerplanar 2-CNF Formulas. In Proceedings of the Pattern Recognition-10th Mexican Conference, MCPR 2018, Puebla, Mexico, 27–30 June 2018, Proceedings; Trinidad, J.F.M., Carrasco-Ochoa, J.A., Olvera-López, J.A., Sarkar, S., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2018; Volume 10880, pp. 72–81. https://doi.org/10.1007/978-3-319-92198-3_8.
18. Zhang, Z. Merrifield-Simmons index of generalized Aztec diamond and related graphs. *MATCH Commun. Math. Comput. Chem.* **2006**, *56*, 625–636.