Jurnal Teknologi, 41(C) Dis. 2004: 65–72
© Universiti Teknologi Malaysia

A VISUAL TOOL FOR DETERMINING THE MAXIMUM INDEPENDENT SET OF A GRAPH USING THE POLYNOMIAL FORMULATION METHOD

FONG WAN HENG[1] & SHAHARUDDIN SALLEH[2]

Abstract. Maximum independent set is a prototype problem in a graph theory that has many applications in engineering and science. In this paper, we study the problem of determining the largest number of maximum independent sets of a graph $G(V, E)$ of order n using polynomial formulations. This method uses the first and second order neighbourhoods of nodes to determine whether a node is in the maximum independent set. This involves the maximization of an n-variable polynomial, where n is the number of nodes in G. Finally, we present a visual tool developed using the Visual C++ programming language for solving the maximum independent set problem.

Keywords: Independent set, polynomial formulation, second-order neighbourhood.

Abstrak. Set tak bersandar maksimum merupakan satu masalah prototaip dalam teori graf yang meliputi banyak aplikasi dalam sains dan kejuruteraan. Dalam kertas ini, kami mengkaji masalah menentukan set tak bersandar maksimum bagi graf $G(V, E)$ peringkat n menggunakan kaedah huraian polinomial. Kaedah ini menggunakan jiran nod-nod peringkat pertama dan kedua bagi menentukan sama ada nod-nod tersebut suatu set tak bersandar maksimum. Ia meliputi pengmaksimuman polinomial n pembolehubah, di mana n ialah jumlah nod dalam G. Seterusnya, suatu perisian berbentuk penglihatan bagi menyelesai masalah pencarian set tak bersandar maksimum menggunakan Visual C++ dicadangkan dalam kertas ini.

Kata kunci: Set tak bersandar, huraian polinomial, jiran peringkat kedua.

1.0 INTRODUCTION

The maximum independent set problem often arises in the following form: given n tasks which compete for the same resources, compute the largest set of tasks which can be performed at the same time, without sharing resources.

Let $G = (V, E)$ be an undirected graph, where $V = \{1, 2, ..., n\}$ is the set of vertices and E is the set of edges. The complement graph of $G = (V, E)$ is the graph $\overline{G} = (V, \overline{E})$, where $\overline{E}$ is the complement of E. Let I be a subset of V. I is an independent set if no two vertices in I is connected by an edge in E. The set I is the maximal independent set if further addition of any vertex destroys its independent set property. The set I is called maximum if there is no larger independent sets in the graph. Thus, a maximum

[1&2] Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor.

independent set is maximal but the converse is not always true. The independence *number* is called the cardinality of a maximum independent set in G.

In this paper, a visual tool is presented for computing the maximum independent set in a graph using the polynomial formulation method. In the polynomial formulation method, the neighbourhoods of points are taken into account, and the objective function is expressed in the polynomial form. We developed a simulation model using C++ to solve maximum independent set problem. In addition, several applications of the maximum independent set problem are also discussed.

2.0 PROBLEM BACKGROUND

The problem of finding the maximum independent set of a graph G has been extensively studied [1], and there is a number of works on counting the number of maximum independent sets [2-5]. Erdõs and Moser first raised the problem of determining the maximum value of maximal independent sets for a general graph G of order n, and those graphs achieving this maximum value. Erdõs then solved this problem, followed by Moon, and Moser [6]. Two decades later, the problem was extensively studied for various classes of graphs, including trees, forests, (connected) graphs with at most one cycle [7], bipartite graphs, connected graphs, k-connected graphs, triangle-free graphs, and connected triangle-free graphs. Besides, there are different approaches in finding the maximum independent sets in a graph. This includes genetic algorithm which involves natural selection processes such as crossover and mutation [8], parallel algorithm [9,10], improved approximation ratios [11], approximation chains [12], and approximation by excluding subgraphs [13].

In this paper, the maximum independent set problem is formulated as the polynomial formulation problem. Based on these formulations, we can characterize the independence number of a graph G as an optimization problem. We consider a polynomial of n variables here.

$$F(x) = \sum_{i=1}^{n} (1 - x_i) \prod_{(i,j)\in E} x_j, x \in [0,1]^n \tag{1}$$

The independence number of the graph G is actually the maximization of objective function, $F(x)$, as shown in the following theorem:

Theorem 1.
Let $G = (V, E)$ be a graph of n nodes, named as $V = \{1,2,\ldots,n\}$ and set of edges E. $\alpha(G)$ denotes the independence number of G. Then,

$$\alpha(G) = \max_{0 \le x \le 1, i=1,\dots,n} F(x) = \max_{0 \le x \le 1, i=1,\dots,n} \sum_{i=1}^{n} (1 - x_i) \prod_{(i,j) \in E} x_j, \tag{2}$$

where each variable x_i, corresponds to node $i \in V$.

Proof.

First, we denote the objective function by:

$$f(G) = \max_{0 \le x \le 1, i=1,\dots,n} F(x) = \max_{0 \le x \le 1, i=1,\dots,n} \sum_{i=1}^{n} (1 - x_i) \prod_{(i,j) \in E} x_j$$

We need to show that:

(1) (2) always has an optimal 0-1 solution.

(2) $\alpha(G) = f(G)$.

$F(x)$ is a continuous function and $[0,1]^n = \{(x_1, x_2, \dots, x_n) : 0 \le x_i \le 1, i = 1, \dots, n\}$ is a compact set, from the definition of compact set: A set S is compact if, from any sequence of elements $x_1,\dots,x_n$ of S, a subsequence can always be extracted, which tends to limit some element X of S. Compact sets are therefore, sets which are both closed, and bounded.

Thus, there always exists $x^* \in [0,1]^n$ such that $F(x^*) = \max_{0 \le x \le 1, i=1,\dots,n} F(x)$, where $F(x)$ is in the following form:

$$F(x) = (1 - x_i) A_i(x) + x_i B_i(x) + C_i(x)$$

and $i \in V$. From this definition, we obtain the first-order neighbourhood of vertex i,

$$A_i(x) = \prod_{(i,j) \in E} x_j$$

and the second-order neighbourhood of vertex i,

$$B_i(x) = \sum_{(i,k) \in E} (1 - x_k) \prod_{(k,j) \in E, j \ne i} x_j$$

The neighbourhood of all other vertices other than i, which is not characterized by $B_i(x)$, in other words, $C_i(x)$ is complementary to $B_i(x)$,

$$C_i(x) = \sum_{(i,k)\notin E} (1 - x_k) \prod_{(k,j)\in E} x_j$$

$F(x)$ is linear with respect to each variable since x_i is absent. From the above representation, if x^* is any optimal solution of (2), then,

$$x_i^* = 0, \quad \text{if } A_i(x^*) > B_i(x^*),$$
$$\text{and } x_i^* = 1, \quad \text{if } A_i(x^*) < B_i(x^*)$$

Finally, if $A_i(x^*) = B_i(x^*), x_i^* = 1$. Since $F(x)$ is linear with respect to each variable, an optimal solution cannot be fractional. This proves that (2) always has an optimal 0-1 solution.We need to show that $f(G) \geq \alpha(G)$. First assume that $\alpha(G) = m$, and I is a maximum independent set. Let

$$x_i^* = \begin{cases} 0, & \text{if } i \in I; \\ 1, & \text{otherwise.} \end{cases}$$

Then, $f(G) = \max\limits_{0\leq x \leq 1, i=1,\ldots,n} F(x) \geq F(x^*) = m = \alpha(G)$. Next, we show that $f(G) \leq \alpha(G)$. Since $F(x)$ is linear with respect to each variable, hence the fractional optimal solutions do not exist. Hence, $f(G)$ is an integer. We assume $f(G) = m$ and takes any optimal 0-1 solution of (2). Without loss of generality, we can assume that this solution is $x_1^* = x_2^* = \ldots = x_k^* = 0; x_{k+1} = x_{k+2} = \ldots = x_n^* = 1$, for some k. We have:

$$\left(1 - x_j^*\right) \prod_{(1,j)\in E} x_j^* + \left(1 - x_2^*\right) \prod_{(2,j)\in E} x_j^* + \ldots + \left(1 - x_k^*\right) \prod_{(k,j)\in E} x_j^* = m,$$

where each term in (8) is either 0 or 1. Therefore $k \geq m$ and there exists a subset $I \subset \{1, \ldots, k\}$ such that $|I| = m$ and

$$\forall i \in I : \prod_{(i,j)\in E} x_j^* = 1$$

Therefore, if $(i, j) \in E$, then $x_j^* = 1$. Then, to prove that I is an independent set, we see that $x_1^* = x_2^* = \ldots = x_k^* = 0$, it follows that $\forall \{i, j\} \subset I$ we have $(i, j) \notin E$ and so I is an independent set. Thus, $\alpha(G) \geq |I| = m = f(G)$. Since $f(G) \geq \alpha(G)$ and $\alpha(G) \geq f(G)$, we have $\alpha(G) = f(G)$, and the proof is complete.

The function $F(x)$ is linear with respect to each variable, so $A_i(x)$ and $B_i(x)$ can be computed for any $i \in \{1,2,...,n\}$. To produce a maximal independent set using $F(x)$, first we take $x^0 \in [0,1]^n$ as any starting point. The procedure produces a sequence of n points $x^1, x^2, ..., x^n$ such that x^n corresponds to a maximal independent set.

From (4-6), we set

$$x_i^1 = \begin{cases} 0, \text{ if } A_i\left(x^0\right) > B_i\left(x^0\right); \\ 1, \qquad\qquad \text{otherwise.} \end{cases}$$

and $x_j^1 = x_j^0$, if $j \neq i$ such that we obtain for the point $x^1 = \left(x_1^1, x_2^1, ..., x_n^1\right)$ that $F\left(x^1\right) \geq F\left(x^0\right)$.

Then we update $V = V \setminus \{i\}$ and construct the next point x^2 from x^1 in the same manner. Running this procedure n times, we obtain a point x^n which satisfies the inequality:

$$F\left(x^n\right) \geq F\left(x^0\right)$$

We will prove that x^n has an independent set associated with it in the following theorem.

Theorem 2.

If $I = \left\{i \in \{1,2,...,n\} : x_i^n = 0\right\}$, then I is an independent set.

Proof.

Consider any $(i, j,) \in E$. We need to show that $\{i, j\}$ is not a subset of I. Without loss of generality, let us assume that we check x_i on the k-th iteration of the above procedure. If $x_i^k = 1$, then $i \notin I$. Alternatively if $x_i^k = 0$, i.e. $i \in I$, we need to show that $j \notin I$. Now, let $l > k$ be an iteration on which we check x_j. Then, $A_j\left(x^{l-1}\right) = \prod_{(i,j) \in E} x_i = 0$, and, therefore, $A_j\left(x^{l-1}\right) \leq B_j\left(x^{l-1}\right)$ and $x_j^l = 1$ which implies that $j \notin I$.

To obtain the maximum independent set, we run the iteration for $2n$ times. The nodes which return a value of zero are in the maximum independent set. This output varies depending on our initial value for x^0, since the problem of finding such a point cannot be solved in polynomial time, unless P = NP. There, we set $x^0 = 0.5$ in this problem as a initial point, where $F(x^0) \leq 1$. In any case, if the initial point is not suitable, the maximal independent set is found, but not necessarily the maximum independent set.

3.0 OUR SIMULATION MODEL

Our simulation model is based on a connected graph having a maximum of 30 vertices. The main objective is to develop a user-friendly interface for computing the maximum independent set in the graph. The points are drawn from a mouse, which generate a connected graph. Here, the method used to solve this problem is polynomial formulation which involves n variables, and the objective function is a continuous function. We considered all the nodes in the graph, and determined whether each of them is in the maximum independent set, by looking at their neighbourhoods.

Figure 1 shows the output from our model for determining the number of nodes in the maximum independent set of a graph, and the corresponding nodes. The model was developed using the Microsoft Visual C++ compiler on the Windows environment. The graph was drawn by the user, where the left mouse is used to click point or nodes, and the right mouse is used to draw edges between nodes. To compute the maximum independent set in the graph, the "COMPUTE" button is clicked, and the total nodes

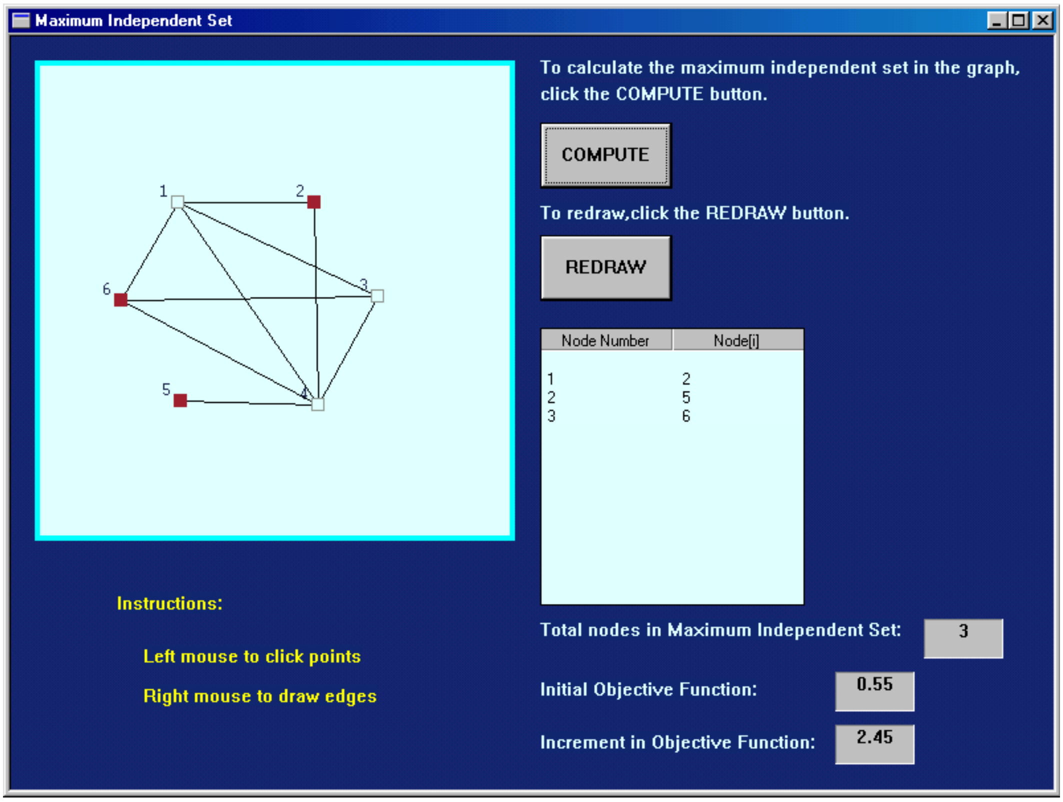

Figure 1 Screen snapshot of our model

in maximum independent set is displayed with the corresponding nodes shown in the list box. To add any nodes and edges, just add with the left and right mouse, and click the "COMPUTE" button. To redraw the graph, click the "REDRAW" button, and start plotting the graph again. The initial objective function and increment in objective function (difference between number of nodes in maximum independent set and initial objective function) are displayed for references.

In this example,

$$
\begin{aligned}
&x=(x_1,x_2,x_3,x_4,x_5,x_6)\in[0,1]^6;\\
&F(x)=(1-x_1)x_2x_3x_4x_5x_6+(1-x_2)x_1x_4+(1-x_3)x_1x_4x_6\\
&\qquad +(1-x_4)x_1x_2x_3x_5x_6+(1-x_6)x_1x_3x_4+(1-x_5)x_4;\\
&A_1(x)=x_2x_3x_4x_6;A_2(x)=x_1x_4;A_3(x)=x_1x_4x_6;A_4(x)=x_1x_2x_3x_5x_6;\\
&\qquad A_5(x)=x_4;A_5(x)=x_1x_3x_4;\\
&B_1(x)=(1-x_2)x_4+(1-x_3)x_4x_6+(1-x_4)x_2x_3x_5x_6+(1-x_6)x_3x_4;\\
&B_2(x)=(1-x_1)x_3x_4x_6+(1-x_4)x_1x_3x_5x_6;\\
&B_3(x)=(1-x_1)x_2x_4x_6+(1-x_4)x_1x_2x_5x_6+(1-x_6)x_1x_4;\\
&B_4(x)=(1-x_1)x_2x_3x_6+(1-x_2)x_1+(1-x_3)x_1x_6+(1-x_5)+(1-x_6)x_1x_3;\\
&B_5(x)=(1-x_4)x_1x_2x_3x_6;\\
&B_6(x)=(1-x_1)x_2x_3x_4+(1-x_3)x_1x_4+(1-x_4)x_1x_2x_3x_5;
\end{aligned}
$$

Applying the algorithm for this graph with initial point $x^0=\left(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\right)$ we obtain, at the end of step 1, the solution (1,0,1,1,0,0), which corresponds to the maximum independent set $I=\{2,5,6\}$. In this case, we have $|I|=3, F(x^0)=\frac{35}{64}$, and the objective function improvement value is $\frac{157}{64}$.

4.0 CONCLUSION

In this paper, we discussed the polynomial formulation method for solving the maximum independent set problem. To produce maximal independent set using objective function $F(x)$, we take a starting point. Then, we calculate the first and second-order neighbourhoods of vertex i. This produces a sequence of points in the maximal independent set. The iteration is then run $2n$ times to produce maximum independent set, whereby the nodes corresponding to a value of zero is in the maximum independent set. However, this output varies depending on our initial value for x^0.

An algorithm to determine the maximum independent set is proposed in this paper. A program using C++ programming with MFC was developed for computing the maximum independent set of a given graph using the polynomial formulation. We also discuss some applications of the maximum independent set problem such as computing the largest set of tasks which can be performed at the same time without sharing resources, given n tasks which compete for the same resources.

REFERENCES

[1] Golumbic., M. C. 1980. Algorithmic Graph Theory and Perfect Graphs. New York: *Academic Press*.

[2] Hopkins, G., and W. Staton. 1985. Graphs with Unique Maximum Independent Sets. *Discrete Math.* 57: 245-251.

[3] Jou, M. J. 1996. Counting Independent Sets. *Ph.D Dissertation,* Dept. of Applied Math. National Chiao Tung University. Taiwan.

[4] Liu, J. 1992. Maximal and Maximum Independent Sets in Graphs. *Ph.D. Dissertation*. Dept. of Math. and Statistics. Western Michigan University.

[5] Zito, J. 1991. The Structure and Maximum Number of Maximum Independent Sets in Trees. *J. Graph Theory.* 15: 207-221.

[6] Moon, J. W., and L. Moser. 1965. On CLiques in Graphs. *Israel J. Math.* 3: 23-28.

[7] Jou, M. J., and G. J. Chang. 1997. Maximal Independent Sets in Graphs with at Most Be Cycle. *Discrete Applied Math*. 79: 67-73.

[8] Xingzhao, L., A. Sakamoto, and T. Shimamoto. 1996. *A Genetic Algorithm for Maximum Independent set problems*. IEEE.

[9] Takefuji, T., L. Chen, K. C. Lee, and J. Huffman. 1990. Parallel Algorithms for Finding a Near-maximum Independent Set of a Circle Graph. *IEEE Transactions On Neural Networks*. 1(3): 263-267.

[10] W. S., Evan. 1991. Comments on Parallel Algorithms for Finding a Near-maximum Independent Set of a Circle Graph. *f*. 2(2): 328-329.

[12] Paschos, V., and M. Demange. 1997. A Generalization of König-Egervary Graphs and Heuristics for the Maximum Independent Set Problem with Improved Approximation Ratios. *European Journal of Operational Research*. 580-592.

[13] Demange, M., and V. T. Paschos. 1997. Improved Approximations for Maximum Independent Set Via Approximation Chains. *Applied Math. Lett*. 10(3): 105-110.

[14] Boppana, R., and M. M. Halldórsson. 1992. Approximating Maximum Independent Sets by Excluding Subgraphs. BIT 32. 2.