STUDIA UNIV. BABES-BOLYAI, INFORMATICA, VOLUME II, NUMBER 1, 1997

BRANCH AND BOUND METHOD FOR INDEPENDENT AND DOMINATED SETS IN GRAPH

TEODOR TOADERE ION COZAC

Abstract. In this paper a type of problem that may be solved using the branch and bound method is presented. There are also presented two graph theory problems in order to illustrate the branch and bound method. The problems considered here is to determine the (interior stable) independent and dominated sets of a graph.

1. Introduction

There are many ways to introduce the basic notions of graph theory, and in particular the notion of graph itself.

By a *partially directed graph* (or briefly, by a *graph*) we understand an ordered quadruple $G = (V, E, I, O)$, where V and E are disjoint sets, and I and O are binary relations on the cartesian product $E \times V$ so that for every $u \in E$ we have: $1 \leq |I(u)| \leq 2$, $|O(u)| \leq 1$, and $O(u) \subset I(u)$ (the symbol $|A|$ denotes the cardinality of the set A). If for every $u \in E$ $O(u) \neq \varnothing$ then G is a *directed graph*, and if for every $u \in E$ $O(u) = \varnothing$ then G is an *undirected graph*.

The sets V, E and $V \cup E$ respectively are called *vertex-set, edge-set* and *element set* of the graph G. The binary relations I and O will respectively be called *incidence* and *orientation* of the graph G.

If $(u, i) \in I$ we say that the edge u and the vertex i are *incident* in G. If $I(u) = \{i,j\}$, the edge u is said to *join* the vertices i and j in G; vertices i and j are the *ends* of the edge u. If $i = j$, i.e. $|I(u)|=1$, u is a *loop*, else, i.e. $i \neq j$ u is sometimes called a *link*. The vertices i and j are called *adjacent* if there is a link joing them.

Two edges u and v are called *adjacent* if $u \neq v$ and there is a vertex incident to both u and v.

Suppose that the edge u joins vertices i and j in G. If $O(u) = \{j\}$ the edge u is said to be *directed* from i to j; the vertices i and j are then called *initial* and *terminal* vertex, respectively, and the ordered pair (i, j) is the edge u.

Received by the editors: January 13, 1998.

1991 Mathematics Subject Classification. 05C45, 05C85, 68Q20, 68R10.

1991 CR Categories and Descriptors. G.2.2 [**Discrete Mathematics**]: Graph Theory - *graph algorithms*

An edge u of a graph G is called a *p-fold edge* if $p = |\{v \in E \mid I(u) = I(v) \text{ and } O(u) = O(v)\}|$. A graph having at least one p-fold edge, $p \geq 2$ is called a *multigraph*. A graph in which the multiplicity of each edge does not exceed p will be refered to as *p-fold graph* or *p-graph*.

An undirected graph without loops and without multiple edges will be referred to as *simple graph*. Directed graphs without loops and multiple edges will be called *digraph*. We may also define the notion of digraph in two different ways. Thus, a *digraph* G is a pair (V,E), where V is a nonempty vertex set and $E \subset V \times V$ is an edge set (set of ordered pair of vertex). Another definition for a digraph is a pair (V, Γ), where V is a non-empty vertex set and $\Gamma : V \to V$ is a multiple-value mapping in V. We have $(i, j) \in E$ if and only if $j \in \Gamma(i)$ or $\Gamma(i) = \{j \in V \mid (i, j) \in E\}$. For $A \subset V$, $\Gamma(A) = \cup_{i \in A} \Gamma(i)$.

By Γ^{1} we denote the multiple-value mapping $\Gamma^{1}: V \to V$, where $\Gamma^{1}(i) = \{j \in V | \ i \in \Gamma(j)\}$ for $i \in V$.

In the digraph G a *walk* from a vertex i to vertex j is defined as a finite sequence $[k_0 \ u_1; \ k_1; u_2; \ \ldots; \ k_{n-1}; \ u_n; k_n]$ or a finite sequence $[k_0; k_1; \ldots; k_{n-1}; k_n]$ of vertices or a finite sequence $[u_1; \ u_2; \ \ldots; \ u_n]$ of edges, where $n \geq 0$, $k_0 = i$, $k_n = j$ and u_p is the edge joining k_{p-1} to k_p for $p = 1, 2, \ldots, n$. The i and j are called a *path* if no edges occur more that once in it.

A graph $G = (V, E)$ is a *bipartite graph* if exists $V_1 \subset V$ so that $E \subset V_1 \times (V - V_1)$. A bipartite graph G is denoted by $(V_1, \ V_2; \ E)$, where $V_2 = V - V_1$, or $(V_1, \ V_2; \ \Gamma)$, where $\Gamma : V_1 \to V_2$ is a multiple-value mapping.

A digraph $G = (V, \Gamma)$ is a *rooted tree* with the root $r \in V$ if only if $\Gamma^{1}(r) = \varnothing$ and every $i \in V - \{r\}$ there is only walk from r to i, i.e. $|\Gamma^{1}(i)|=1$. If $|\Gamma(i)| \leq 2$, for every $i \in V$ rooted tree is a *binar rooted tree*. Vertices $i \in V$ having the property $\Gamma(i) = \varnothing$ is called terminal vertex or leave.

2. The Branch bound method

This way of solving problems is applied usually to NP-complete problems which satisfy the following conditions:

(i) the set of possible solutions of the problem P is a finit set $S = \{s_1; \ s_2; \ \ldots; \ s_t\}$;

(ii) there is a function $f : S \to R$ with the property that if $a \in S$ and $f(a) = \min\{f(x) \mid x \in S\}$, then a is an optimal solution of the problem P;

(iii) there is a criterion to determinate (estimate) an upper bound of $f(A)$ (i.e. $f(A)$ can be that estimation);

(iv) there is a criterion for decomposition of any subset A of S in two or more subsets B_1, $\ldots$, B_s, i.e. $A = B_1 \cup B_2 \cup \ldots \cup B_s$.

Having such a procedure we can make an labeled rooted tree. In every label of vertices there is a subset $A \subset S$, in explicit or implicit form, and a superior (inferior) border which estimate the sup $f(A)$ called margin. Rooted tree will be developped by adding the adjacent vertices of the terminal vertex with smaller margin. The adjacent vertices correspond to the criterion of decomposition of A from the label of the selected terminal vertix.

The development will be done until an optimal solution is obtained or from the label of the vertex results that $A = \{s\}$, and in this case it retains the value b of the margin. Then the development continues only by the selection of those vertices which produce a value smaller or equal to b (i.e. the value of margins must indicate the obtaining of an admisible solution or a better or optimal solution).

The rooted tree is initialized with only vertex labeled $A = S$. Because only the terminal vertices from the rooted tree are of interest, we may keep only those vertices represented as list type structure.

For some problems $f(a) = \max \{f(x) \mid x \in S\}$ and borders estimate the inf $f(A)$, so they are minimal inferior borders for $f(A)$.

3. Independent sets

Let $G = (V, E)$ be a digraph. A subset $S \subset V$ is an *independent set* of vertices of G if and only if for any $i, j \in S$ we have $(i, j) \notin E$.

Properties 3.1. Let $G = (V, E)$ be a digraph.

a) $\varnothing$ is an independent set;
b) if S is an independent set then $\Gamma(S) \cap S = \varnothing$;
c) if S is an independent set and $A \subset S$, then A is also an independent set;
d) if S_1 and S_2 are independent sets, then $S_1 \cap S_2$ is also an independent set.

The independence number of the graph G is $\max\{|S| \mid S \in I\}$, and it is denoted by $\alpha(G)$. An independent set S is *maximum* if $|S|=\alpha(G)$ and it is *maximal* if does not exist another independent set B with $S \subset B$ and $S \neq B$.

Remark 3.2. Any maximum independent set is also maximal, but the reciproc generaly is not true.

Example 3.3. Consider the graph $G = (V, E)$ where $V = \{1, 2, 3\}$ and $E = \{(1,2); (2,3)\}$. The independent sets of this graph are: $\varnothing$; $\{1\}$; $\{2\}$; $\{3\}$; $\{1,3\}$, the number of independence is 2, the maximal independence sets are $\{2\}$; $\{1,3\}$ and the maximum independence set is $\{1,3\}$.

Problem 3.4. Compute the number of independence and all the maximum independent sets of a digraph G.

This problem is NP-complete [WH94]. We give a branch and bound algorithm for this problem. The labels of the rooted tree are of type (A, B, b) where:

- $A \subset V$ is an independent set and all the descendents of this vertex define the independent set that contain A;
- $B \subset V$ with the property that $\Gamma(A) \cup \Gamma^{-1}(A) \subset B$ is the set of vertices which will not go in the independent set defined by the label of the any descendent vertex in the rooted tree;
- $b = |V\text{-}B|$ is a superior margin of the number of the vertices which can be added to the set A to obtain another independent set.

Theorem 3.5. If (A, B, b) is the label of a vertex in the rooted tree then for every $i \in V\text{-}(A \cup B)$ the set $A \cup \{i\}$ is an independent set of the digraph G.

Proof.

$$\Gamma(A \cup \{i\}) \cap (A \cup \{i\}) = (\Gamma(A) \cap A) \cup (\Gamma(i) \cap A) \cup (\Gamma(A) \cap \{i\}) \cup (\Gamma(i) \cap \{i\}).$$

Since A is an independent set it results that $\Gamma(A) \cap A = \varnothing$.

From $\Gamma(A) \cup \Gamma^{-1}(A) \subset B \Rightarrow \Gamma(i) \cap A = \varnothing$ and $A \cap \Gamma(i) = \varnothing$ (otherwise $i \in B$).

The digraph is without loop, then $\Gamma(i) \cap \{i\} = \varnothing$. So $\Gamma(A) \cap A = \varnothing$, which means that A is independent set of the digraph G. □

Because of this property, the criterion of decomposition can be: choose $i \in V\text{-}(A \cup B)$ and add in rooted tree two descendent vertices with label $(A \cup \{i\}, A \cup B \cup \Gamma(i) \cup \Gamma^{-1}(i), |V\text{-}(A \cup B \cup \Gamma(i) \cup \Gamma^{-1}(i))|)$ and respectively $(A, B \cup \{i\}, |V\text{-}B|\text{-}1)$. From this criterion we obtain:

Theorem 3.6. The vertices which have in label the superior margin 0 are terminal vertices and in that label A is independent set of G.

After obtaining this type of vertex the α cardinal of all independent set found out will be held and only the vertices (A,B,b) of rooted tree for which $|A|+b \geq \alpha$ will be developed. The value α is modified every time when a vertex which margin 0 is obtained. For this operation the vertex for which the value $|A|+b$ is maxim will be selected.

4. Dominated sets

Let $G=(V,E)$ be a digraph. A set T of vertices is a *dominated set* if only if for any $i \in V\text{-}T$ there is $j \in T$ such that $(i,j) \in E$.

Properties 4.1. Let $G=(V,E)$ be a digraph.

a) V is dominated set of G;

b) If T is a dominated set and $T \subset A \subset V$, then A is also a dominated set of G;

c) If T_1 and T_2 are two dominated sets, then $T_1 \cup T_2$ is also a dominated set of G.

The dominance number of the digraph $G=(V,E)$ is $\min\{|T| \mid T$ is dominated set$\}$, and it is denoted by $\beta(G)$. The dominated set T with $\beta(G)=|T|$ is a *minimum dominated set*. The dominated set T for which if $A \subset T$ and A is dominated set then $A=T$, is a *minimal dominated set*.

Remark 4.2. Every minimum dominated set is a minimal dominated set. The reciprocal is not generally true.

Example 4.3. In the digraph $G=(V,E)$, where $V=\{1,2,3\}$ and $E=\{(1,2);\ (2,1);\ (2,3);\ (3,2)\}$, the dominated sets are: $\{2\}$; $\{1,3\}$ and $\{1,2,3\}$, the number of dominance is 1, the minimum dominated set is $\{2\}$ and the minimal dominated sets are $\{2\}$ and $\{1,3\}$.

Problem 4.4. Find the dominance number and all the minimum dominated sets of a digraph G?

This problem is also NP-complete and in order to solve it an algorithm of branch and bound type is preferable.

Having a digraph $G=(V,\Gamma)$ we associate to G a bipartite graph $G'=(V,V';\Delta)$ like this: $V'=\{i' \mid i\in V\}$ a new vertices set and $\Delta:V\text{--}\!\!\to V'$ is a multiple-value mapping defined as $\Delta(i)=\{j' \mid j\in\Delta(i)$ or $j=i\}$. Using the digraph G' the dominated sets of G are characterized by the following theorem.

Theorem 4.5. In the digraph $G=(V,\ \Gamma)$ the set $T\subset V$ is a dominated set if and only if in the bipartite digraph $G'=(V,V';\ \Delta)$ defined as above we have $\Delta\ (T)=V'$.

Proof. ($\Rightarrow$) By definition of Δ we have $\Delta\ (T)\subset V'$. Let be $i'\in V'$. If $i\in T$ then of course $i'\in\Delta(T)$. If $i\in V\text{-}T$ then there is $j\in T$ such that $j\in\Gamma i$ (from the definition of an dominated set), and then $i'\in\Delta j\subset\Delta T$.

($\Leftarrow$) If $i\in V\text{-}T$ then $i'\in V'=\Delta T$ and there exists $j\in T$ such that $i'\in\Delta j$ (from the definition of the Δ function), from where it results that $i\in\Delta^{-1}(j)$. Therefore T is a dominated set of G.

$\square$

Branch and bound method for the determination of the minimum dominated sets. This method develop a rooted tree which has its vertices' labels $(G';T;b)$, where G' is an reduced bipartite digraph (the procedure of reduction will be presented in the sequel); $T\subset V$ (obtained after reduction) will be contained in all dominated sets defined by the terminal and descendent vertex of this vertex of rooted tree; $b=|V'|$ is a superior margin of the number of the vertices which must be put to the set T to obtain a dominated set of digraph G.

The root is labeled with $(G';T;b)$, where G', T and b are obtained by using the procedure of reduction applied to G' associated to digraph G which was given.

For the development of the rooted tree the vertex with $(G';T;b)$ with the smallest $b+|T|$ is selected. To this vertex two succesor vertices are added, choosen as it follows:

- it is $i\in V$ selected with $\Delta(i)=0$;
- a succesor will be labeled with the G' obtained through the procedure of reduction applied to the bipartite digraph obtained from $G'=(V,V';\Delta)$ by $V=V\text{-}\{i\}$ and $V'=V'\text{-}\Delta(i)$; $T=T\cup\{i\}$ and $b=|V'|$;

- the other succesor will be labeled with the G' obtained through the reduction procedure applied to the G' with $V=V-\{i\}$ and $V'=V'$; T and b.

The development of the rooted tree goes on until a vertex whose label has a G' that verifies $V=\varnothing$ will be obtained.

Remarks 4.6. (1) The bipartite graphs from the labels of the succesor are obtained from those labels of the father vertex by eliminating some vertices and of the edges incident to these vertices.

(2) The bipartite digraphs from the labels of the rooted tree have the property that $\forall i\in V$ $|\Delta^{-1}(i')|\geq 2$, and so by the elimination of some vertices and of the incident edges to them, the vertices from V' are not becoming isolated.

The reduction procedure. Parameters are $G'=(V,V';\Delta)$ and T.

while there exists $j'\in V'$ with $|\Delta^{-1}(j')|=1$ **do**

$i:=\Delta^{-1}(j')$;

$T:=T\cup\{i\}$;

$V:=V-\{i\}$;

$V'=V'-\Delta(i)$;

end.

5. Proposed problems and their models

Problem 5.1. If we have a chess table of 8x8 or generally nxm, what is the maximum number of the pieces of the same type (queens, horses, etc.) which can be set on the table such that any two pieces can not atack each other. Find out all the solutions to place the pieces.

Solution. A digraph with 8x8=64 vertices or nm vertices in general case, will be built. Each vertex of the digraph corresponds to a chess table square. Two different vertices i and j are linked by an edge if and only if the pieces in the corespondent squares atack each other. We will find all the independent set of the digraph. An independent set shows the way of putting the pieces on the chess table.

Problem 5.2. If we have a chess table of 8x8 or generaly nxm, what is the minimum number of the pieces of the same type (queens, horses, etc.) which can be set on the table so that any square of the table must be controlled and it's required to find all the solutions to display the pieces.

Solution. Similarly we construct a digraph, two different vertices i and j are linked by an edge if and only if the chess piece set in the square which coresponds to the vertex i controls the square which corresponds to the vertex j. We'll find out all the minimum dominated sets of the digraph. A minimum dominated set indicates the way of putting the pieces on the chess table.

6. Conclusions

We have presented an analytical form of the branch and bound method, and some methods to solve graph problems with it.

In [RA74] and [TT92] a branch type algorithm (just branched!) which finds out all the independent sets of a digraph is presented. This algorithm was updated with the bound procedure as the one described before. In [TI75] and [RR84] an algorithm (Bednareck-Taulbee) which finds out maximal independent sets of a digraph is presented. But the family of the maximal independent sets is bigger that the one of the maximum independent set, so this algorithm doesn't work for our problem.

In [BC69], [RA74], [RR84] and [TT92] an algorithm which finds out some (!) minimum dominated sets is presented. Why doesn't it determine all of them? This algorithm uses a pretty tough condition to eliminate the vertices: if there are two vertices i and j with the property that $\Delta(i) \subset \Delta(j)$ in the digraph G', then it eliminates the vertex i from the digraph G'. This condition is not proven and it can provoke the losing of some solutions lot of times, like in the following example.

Example 6.1. Let $G=(V,E)$ be a digraph where $V=\{1,2,3,4\}$ and $E=\{(1,2); (2,3); (3,1); (3,2); (4,3)\}$. It may easily be verified that the dominance number of this digraph is 2. The minimum dominated sets of G are $\{1,3\}$, $\{2,3\}$ and $\{2,4\}$. The bipartite digraph associated to G is built, it may be noted that $\Delta(1) \subset \Delta(2)$ and $\Delta(4) \subset \Delta(3)$. If we will follow old algorithm we will obtain just a single solution $\{2,3\}$.

The algorithm presented in the papers mentioned before was completed with the brach procedure and the assignment of labels to the vertices of the rooted tree as it was described even more. Moreover, in algorithm the procedure coresponding to the condition $\Delta(i) \subset \Delta(j)$ was eliminated.

REFERENCES

[BC69] C. Berge, *Theorie des graphes et ses applications*, Paris, Dunod, 1967.

[RA74] A. Rosu, *Graph Theory. Algorithms and Applications*, Ed. Militara, Bucuresti, 1974. (in romanian)

[RR84] N. Radescu; E. Radescu, *Problems of Graph Theory*, Editura Scrisul românesc, Craiova, 1984. (in romanian)

[TI75] I. Tomescu, *Introduction to combinatorics*, Collet's (Publishers) Ltd.,London and Willingbough,1975

[TT92] T.Toadere, *Basics of Graph Theory*, Lito. Univ. "Babes-Bolyai", Cluj-Napoca, 1992. (in romanian)

[WH94] H.S.Wilf, *Algorithms and Complexity*, 1994, Internet resource, ftp://ftp.cis.upenn.edu/pub/wilf/AlgComp.ps.Z

Babeş-Bolyai University, Faculty of Mathematics and Informatics, RO 3400 Cluj-Napoca, str. Kogalniceanu 1, România.

E-mail address: toadere@cs.ubbcluj.ro

Petru Maior University, Faculty of Science, RO 4300 Tg. Mures, România.

E-mail address: icozac@uttgm.ro