RAIRO. INFORMATIQUE THÉORIQUE

P. BERTOLAZZI
M. LUCERTINI
A. MARCHETTI SPACCAMELA

Analysis of a class of graph partitioning problems

RAIRO. Informatique théorique, tome 16, nº 3 (1982), p. 255-261
<http://www.numdam.org/item?id=ITA_1982__16_3_255_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 16, n° 3, 1982, p. 255 à 261)

ANALYSIS OF A CLASS OF GRAPH PARTITIONING PROBLEMS (*)

by P. BERTOLAZZI,
M. LUCERTINI and A. MARCHETTI SPACCAMELA ([1])

Communicated by G. AUSIELLO

Abstract. — *The partitioning problem of a graph into m subgraphs with size constraint and an objective function given by the sum of links among vertices belonging to different subgraphs, is well-known and appears in a number of very diverse problem areas. This paper presents some simple results concerning some particular classes of graphs such as chains, cycles, stars and binary trees.*

Résumé. — *Cet article traite du problème de la partition d'un graphe en plusieurs sous-graphes avec des contraintes sur leurs tailles et en minimisant la somme des longueurs des arêtes reliant des sommets appartenant à des sous-graphes différents. Ce problème est bien connu. L'article donne quelques résultats simples sur des classes de graphes particulières telles que les chaînes, les cycles, les étoiles et les arbres binaires.*

1. INTRODUCTION

Given a graph $G(V, E)$ with $|V|=n$ and $|E|=p$, weights $w(v) \in Z^+$ for each $v \in V$ and $a(e) \in Z^+$ for each $e \in E$, and a positive integer B, the graph partitioning problem (GP) is the problem of finding a partition of V into disjoint sets $\{V_1, V_2, \ldots, V_m\}$ such that $(\sum_{(v \in V_i)} w(v) \leqq B)$ for $(1 \leqq i \leqq m)$ and such that if $E^c \subseteqq E$ is the set of edges that have their two end points in two different sets V_i, then $(\sum_{(e \in E^c)} a(e))$ is minimal.

This problem is relevant in many fields and in particular it frequently arises in computer system design and in the allocation of computer information to blocks of storage [1, 5, 8, 10].

(*) Received in October 1980.
([1]) Istituto di Analisi dei Sistemi ed Informatica del C.N.R. and Istituto di Automatica dell'Università di Roma, Via Eudossiana, 18 00184 Roma.
This work was supported by Consiglio Nazionale delle Ricerche.
A first version of this work was presented at the 18th Allerton Conference, Urbana, 1980.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1982/255/$ 5.

The (GP) problem has been proved to be NP-complete for fixed $B \geqq 3$, even if all vertex and edge weights are equal to 1 [2, 4]. It remains NP-complete with the additional constraint that the graph obtained from the partition is acyclic [2]. If G is a tree the general problem is NP-complete, but can be solved in pseudopolynomial time $O(n.B^2)$ [9]; if all edge weights or vertex weights are equal the tree problem can be solved in polynomial time [2, 3].

In this paper some simple additional results concerning some particular classes of trees are showed.

In particular the NP-completeness of the problem of partitioning simple graph structures such as stars and binary trees is proved, an $O(n)$ algorithm for the "chain partitioning" problem (CP) and an $O(n.B)$ algorithm for the "star partitioning" problem (SP) are presented. The (CP) algorithm is generalized in order to find an heuristic to solve the general (GP) problem.

2. NP-COMPLETENESS RESULTS

THEOREM 1: *The problem of finding an optimal partitioning of a star is NP-complete.*

Proof: We show that the subset sum problem (shown to be NP-complete by Karp [5]) is polynomially reducible to our problem. Given a set $N=\{a_1, a_2, \ldots, a_n\}$ such that:

$$\sum_{i=1}^{n} a_i = 2S,$$

of positive numbers, does exist a subset $J \subseteq N$ such that:

$$\sum_{j \in J} a_j = \sum_{j \notin J} a_j = S\ ?$$

Given any istance of subset sum we can construct an istance of the partitioning problem for stars in the following way:

- for each $i=1, 2, \ldots, n$ there is a node i with weight a_i;
- there is an additional node $(n+1)$ with weight $M=2S$;
- for each $i=1, 2, \ldots, n$ there is an edge e_i with weight a_i between the node i and the node $(n+1)$;
- $B=S+M$.

Now we show that the subset sum has solution if and only if the corresponding istance of the graph partitioning problem has a solution less or equal to S.

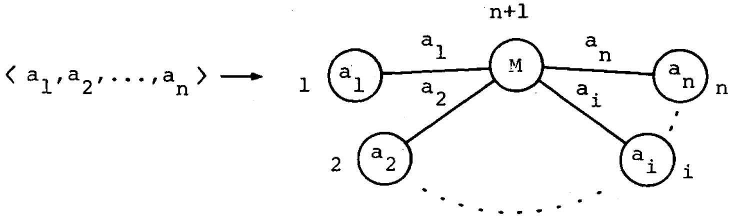

Figure 1

(A) $\Rightarrow$ Let J be a solution for the subset problem. We can obtain a solution for the GP problem by joining node $(n+1)$ with the nodes j, $j \in J$. It is easy to see that the capacity constraint is satisfied and the value of the solution is S.

(B) $\Leftarrow$ Let it be given a solution of the GP problem whose value is less or equal to S and let I be the set of nodes in the same partition of node $(n+1)$.

For the capacity constraint of the bin we have:

$$(1) \qquad \sum_{i \in I} a_i + M \leqq S + M \quad \Rightarrow \quad \sum_{i \in I} a_i \leqq S.$$

Furthermore as the value of the solution is less or equal to S we have:

$$(2) \qquad \sum_{i \notin I} a_i \leqq S.$$

By considering (1) and (2) we have $\sum_{i \in I} a_i = S = \sum_{i \notin I} a_i$. This completes the proof. ◀

With a slight modification of the proof we can obtain the following theorem.

THEOREM 2: *The problem of finding an optimal partition of a binary tree is NP-complete.*

Proof: Given an istance of subset sum we can construct an instance of GP problem for binary tree in the following way:

– for each $i = 1, 2, \ldots, n$ there are two nodes $(i, 1)$ and $(i, 2)$ with weight a_i and $M = 2S$ respectively;

– for each $i = 1, 2, \ldots, n$ there is an edge e_i between $(i, 1)$ and $(i, 2)$ with weight a_i;

– for each $i = 1, 2, \ldots, n-1$ there is an edge e_{n+i} between $(i, 2)$ and $((i+1), 2)$ with weight $2S$;

– $B = (2n+1)S = nM + S$.

We show that the subset sum problem has solution if and only if the graph partitioning problem has solution with value less or equal to S.

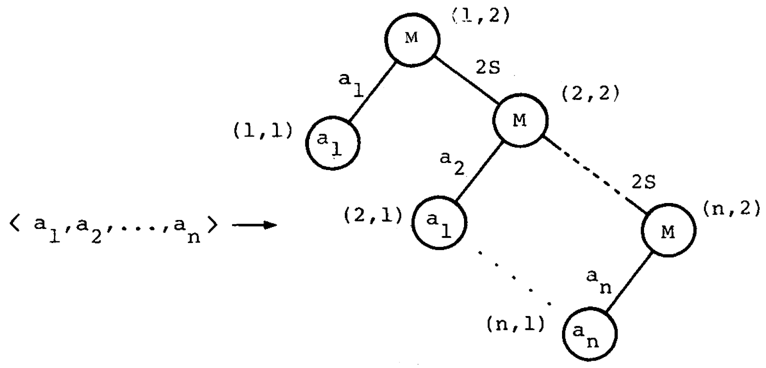

Figure 2

(A) $\Rightarrow$ If J is a solution of the subset sum problem then we obtain a feasible solution for the graph partitioning problem joining the nodes $(j, 2)$ $j=1, 2, \ldots, n$ and the nodes $(j, 1), j \in J$. The cost of this solution is S.

(B) $\Leftarrow$ If there is a solution with value less than or equal to S then all the nodes $(i, 2)$ must be in the same cluster with some other nodes $(j, 1)$ for $j \in J$. It is not difficult to see that $\sum_{j \in J} a_j \leqq S$ (for the capacity constraint) and that $\sum_{j \notin J} a_j \leqq S$ (because the solution is less than or equal to S). Therefore $\sum_{j \in J} a_j = S$ and so there is a solution for the subset sum problem. ◀

3. A $O(n^2)$ ALGORITHM FOR CHAIN PARTITIONING

Let $1, e_{12}, 2, e_{23}, \ldots, n-1, e_{(n-1)n}, n$ be a sequence of nodes and edges: this sequence is called a chain leading from 1 to n. The (CP) problem is the (GP) problem on a chain, where w_i is the weight of vertex i and a_i is the weight of the edge e joining i to $i+1$.

Let $\Omega_{ji}: \{ j+1, \ldots, i-1, i \}$ be a subset of the nodes of the chain. Let be $\Omega = \{ \Omega_{ji} \}$ $\forall j, \forall i > j$; it is easy to see that $|\Omega| = n \cdot (n+1)/2$. Let's associate a weight to each subset Ω_{ji}, $w(j, i) = \sum_{k=j+1}^{i} w_k$: the set $\Omega' = \{ \Omega_{ji} : w(j, i) \leqq B \} \subset \Omega$ has cardinality $|\Omega'| < n \cdot (n+1)/2$.

It's easy to see that the optimal partition of the chain is obtained by choosing in Ω' a subset:

$$P = \{ \Omega_{0j}, \Omega_{j+1k}, \ldots, \Omega_{m+1l}, \Omega_{l+1n} \}$$

with minimum cost of the objective function. Remark that the subsets of V that are candidates to the optimal solution are all connected subchains.

We can define the following quantities:

$f(i)$ = optimal value of (CP) on the first i vertices $(1, 2, \ldots, i)$;

$$\varphi(j, i)=\left\{\begin{array}{ll} a_i & \text{if } w(j, i)\leqq B \\ +\infty & \text{if } w(j, i)>B \end{array}\right\} j<i,$$

$w(j, i)$ $\forall (j, i)$ (and then Ω') can obviously be computed in time:

$O(n^2)$ (infact $w(j, i)=\bar{w}_i-\bar{w}_j$ with $\bar{w}_1=w_1$ and $\bar{w}_j=\bar{w}_{j-1}+w_j (j=2, \ldots, n)$).

It is easy to see that the optimal solution of (CP) can be found by solving the following set of recursive equations.

$$f(O)=a,$$

$$f(i)=\min_{j<i}(f(j)+\varphi(j, i)) \qquad (i=1, 2, \ldots, n).$$

In the worst case the complexity of this algorithm is obviously $O(n^2)$.

The problem remains $O(n^2)$ even if the constraints on the maximum size of the subsets are more complicated (for example if w_i is a r-vector and the constraint has the general form $g_k(w_i, i\in V_k)\leqq 0$).

This algorithm can be utilized for solving the "cycle partitioning" problem (CLP), i. e. the (CP) problem with an additional edge between vertices 1 and n. Given an instance of the (CLP), let $\sum_{k=1}^{n} w_k>B$, in this case at least two cuts will exist in the cycle; let c_i be the optimal value of the solution of the (CP) problem obtained from the cycle by dropping the edge $(i, i+1)$. The optimal solution of (CLP) problem is given by $\min_i (c_i+a_i)$. If we start from vertex 1, the procedure can stop when $i=j$ with $j=\min\left\{k \mid \sum_{i=1}^{k} a_i>B\right\}$. In the worst case the complexity is $O(n^3)$.

4. A GENERALIZATION OF THE CHAIN PARTITIONING ALGORITHM

Let us define a new problem: the graph partitioning problem (GP) with the additional constraint that, given an ordered list of vertices $L=\{r_1, r_2, \ldots, r_n\}$ each subset of the optimal partition only contains adjacent nodes of L. Without loss of generality we can suppose that the vertices are renumbered such that $L=\{1, 2, \ldots, n\}$; let T_i be the set of edges belonging to the cut

$\{(1, 2, \ldots, i), (i+1, \ldots, n)\}$ (i.e. edges e_{hk}) such that $h \in (1, 2, \ldots, i)$ and $k \in (i+1, \ldots, n)$. The set of recursive equations of the previous section holds to find the optimal solution, if we define $\varphi(j, i)$ and $f(O)$ as follows:

$$\varphi(j, i) = \left\{ \begin{array}{ll} \sum_{(e \in T_i,\, e \notin T_j)} a(e) & \text{if } w(j, i) \leqq B \\ +\infty & \text{if } w(j, i) > B \end{array} \right\} j < i,$$

$$f(O) = 0.$$

Remark that in this case, if $w(j, i) \leqq B$, $\varphi(j, i)$ depends both on i and j, instead of depending only on i as in the (CP) algorithm.

5. A $O(n.B)$ ALGORITHM FOR STAR PARTITIONING

For clarity sake we assume that vertex 1 of the star is the vertex connected with all other vertices; let w_i be the weight of vertex i and a_i be the weight of the edge between vertices 1 and i $(i=2, 3, \ldots, n)$.

Let $f(i, y)$ be defined as:

$f(i, y)$ optimal solution of (SP) on the first i vertices with weight of node 1 equal to y;

the optimal solution of (SP) can be found solving the following set of recursive equations:

$$f(1, y) = 0, \qquad \forall y,$$

$$f(i, y) = +\infty, \qquad \forall i, \forall y > B,$$

$$f(i, y) = \min\{a_i + f(i-1, y), f(i-1, y+w_i)\}, \quad i = 2, 3, \ldots, n;$$

$$y = w_1, w_1 + 1, \ldots, B.$$

The algorithm finds the optimal solution in time $O(n.B)$.

Remark that we can apply the same techniques used for the knapsack problem in order to obtain ε-approximate solutions in polynomial time [7].

6. FURTHER RESEARCH

Algorithms (exact or approximate) for solving (GP) can be based on the (CP) (or the (CLP)) algorithm, on the ground of the results of section 4.

If you suppose that n_i is the number of vertices in the set i in the optimal solution and m is the number of sets, the problem becomes how to find one of the

$\left(\left(\sum_{i=1}^{m} n_i\right) m\right)$ optimal sequences of vertices among the $\left(n = \left(\sum_{i=1}^{m} n_i\right)\right)$ lists that we can generate. The approach can be in two different directions: the first one is to find a set of dominance relations among the sequences, in order to eliminate as many sequences as possible; the second one is to point out a heuristic in order to find better sequences, given a sequence and the associate optimal solution.

REFERENCES

1. N. CHRISTOFIDES and P. BROOKER, *The Optimal Partitioning of Graphs*, S.I.A.M. J. App. Math., January 1976.
2. M. R. GAREY and D. S. JOHNSON, *Computers and Intractability*, Freeman and Co., 1979.
3. F. O. HADLOCK, *Minimum Spanning Forest of Bounded Trees*, Proc. 5th Southeastern Conf. on Combinatorics, Graph theory and Computing, Winnipeg., 1974.
4. L. HYAFIL and K. L. RIVEST, *Graph Partitioning and Constructing Optimal Decision Trees are Polynomial Complete Problems*, Re. 33 I.R.I.A.-Laboria, Rocquencourt, France, 1973.
5. R. M. KARP, *Reducibility among Combinatorial Problems, in* R. E. MILLER and J. W. THATCHER Eds., *Complexity of Computer Computations*, Plenum Press, N.Y., 1972.
6. B. W. KERNIGHAN, *Some Graph Partitioning Problems Related to Program Segmentation*, Ph. D. Thesis, Princeton Univ., N.J., January 1969.
7. E. L. LAWLER, *Fast Approximation Algorithms for Knapsack Problems*, Proc. 18th Ann. Symp. on Foundations of Computer Science, I.E.E.E. Comp. Soc., Long Beach, 1977.
8. E. L. LAWLER, K. N. LEVITT and J. TURNER, *Module Clustering to Minimize Delay in Digital Networks*, I.E.E.E. Tr. on Comp. C-18, 1969.
9. J. A. LUKES, *Efficient Algorithm for the Partitioning of Trees*, I.B.M. J. Res. Develop., Vol. 18, 1974.
10. G. S. RAO, H. S. STONE and T. C. HU, *Assignment of Tasks in a Distributed Processor System with Limited Memory*, I.E.E.E. Tr. on Comp., C-28, 1979.