On the Distributed Decision-Making Complexity of the Minimum Vertex Cover Problem *,**

Pierluigi Crescenzi and Luca Trevisan

Dipartimento di Scienze dell'Informazione
Università degli Studi di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
E-mail: {piluc,trevisan}@dsi.uniroma1.it

Abstract

In this paper we study the problem of computing *approximate vertex covers* of a graph on the basis of partial information in the *distributed decision-making* model proposed by Deng and Papadimitriou [1]. In particular, we show an optimal algorithm whose competitive ratio is equal to p, where p is the number of processors.

1 Introduction

The minimum vertex cover (MVC) problem consists of finding, given a graph G, a minimum cardinality set of nodes V' such that, for any edge (u, v), either $u \in V'$ or $v \in V'$. This is a well-studied problem which appeared in the first list of NP-complete problems presented by Karp [4]. A straightforward approximation algorithm, based on the idea of a maximal matching, was successively developed by Gavril (according to [2]) with a performance ratio no greater than 2. Several other approximation algorithms are presented in the lecture notes of Motwani [5]. In this paper we analyse the complexity of finding approximate solutions for the MVC problem in the framework of distributed decision-making with incomplete information [1,3,6–8]. In particular, we assume that the vertex cover is chosen by independent processors, each knowing only a part of the graph and acting in isolation. More specifically, we assume

* Research partially supported by the MURST project *Algoritmi, Modelli di Calcolo, Strutture Informative* and by EEC Human Capital Mobility Program *Efficient Use of Parallel Computers.*
** The results contained in this paper have been presented at the *20th Workshop on Graph-Theoretic Concepts in Computer Science*, Munich, 1994

that the adjacency list of each node of the graph is known by only one processor which has to decide whether the node should belong to the vertex cover. We then want to develop distributed algorithms that always produce feasible solutions (that is, vertex covers) and achieve, in the worst case, a *reasonable* competitive ratio (that is, the ratio of the cardinality of the solution computed by the algorithm to the optimum cardinality should be as small as possible). In this paper, we show that a simple *double-matching algorithm* which essentially performs Gavril's algorithm first on the "bridge" edges and then on the "inner" edges achieves a competitive ratio equal to p where p is the number of processors. We also show, by means of a quite involved counting technique, that this algorithm is *optimal*, that is, no distributed algorithm can achieve a ratio smaller than p. These results fit into a more general context in which an optimization problem has to be solved in a distributed fashion and neither a centralized control nor a complete information are available (see [3] for several applications). Moreover, it has been argued that this kind of results "can be seen as part of a larger project aiming at an algorithmic theory of the value of information" [8]. Intuitively, this theory should allow to compare in terms of competitive ratios two different *information regimes*, that is, two different ways of distributing the input among the processors.

2 The Model

We consider a distributed system formed by p non-communicating processors $P_1, \ldots, P_p$. Given an instance I of an optimization problem Π, we assume that I is encoded as a set of objects $I = \{w_1, \ldots, w_n\}$. The instance is "distributed" among the processors according to a certain criterion, so that the processors P_i receives a subset $I_i \subseteq I$ and computes a partial solution S_i which depends only on I_i and i. A measure function u gives the value $u(S_1, \ldots, S_p, I)$ of the partial solutions $S_1, \ldots, S_p$ (without loss of generality, we assume that u is undefined whenever $S_1, \ldots, S_p$ do not result into a feasible solution).

More formally, an *information regime* (for a p-processor system) is a function $\mathcal{R}$ that, given an instance I, returns a p-tuple of subinstances $I_1, \ldots, I_p$. The p-tuple $I_1, \ldots, I_p$ is also called a *distributed instance*. A *decision strategy* $\mathcal{A}$ is a p-tuple of algorithms $A_1, \ldots, A_p$. The *competitive ratio* of a strategy $\mathcal{A}$ with respect to a regime $\mathcal{R}$ is defined as follows.

$$R(\mathcal{A}, \mathcal{R}) = \max_I \max \left\{ \frac{u(A_1(I_1), \ldots, A_p(I_p))}{\mathrm{opt}(I)} , \frac{\mathrm{opt}(I)}{u(A_1(I_1), \ldots, A_p(I_p))} \right\}$$

where $I_1, \ldots, I_p = \mathcal{R}(I)$ and $\mathrm{opt}(I)$ is the value of an optimum solution for I. Observe that $R(\mathcal{A}, \mathcal{R}) \geq 1$ for any strategy $\mathcal{A}$ and any information regime $\mathcal{R}$, and that $R(\mathcal{A}, \mathcal{R})$ is as close to one as the solutions computed by $\mathcal{A}$ are close to the optimum.

This definition can be easily generalized to a family $\mathcal{F}$ of information regimes as follows:

$$R(\mathcal{A}, \mathcal{F}) = \max_{\mathcal{R} \in \mathcal{F}} R(\mathcal{A}, \mathcal{R}).$$

The *competitive ratio* of a problem Π in a p-processors sysytem, with respect to a family $\mathcal{F}$ is

$$R(\Pi, \mathcal{F}) = \min_{\mathcal{A}} R(\mathcal{A}, \mathcal{F}).$$

It has been argued that such a ratio is a reasonable measure of the *value of the information* that has been distributed to the processors according to the information regimes in $\mathcal{F}$.

For any optimization problem, two approaches are interesting within such a framework:

(i) Fix a natural family of "homogenous" information regimes, usually those with fewer redundancy, and try to characterize as tightly as possible the competitive ratio (that is, the value of information with respect to such a distribution scheme).
(ii) Consider several (families of) different information regimes and show the existence of trade-offs between redundancy and competitiveness.

In this paper we will study the MVC problem within this framework following the first approach.

2.1 The Distributed MVC Problem

In the following we will identify a graph $G = (V, E)$ with the set L of its adjacency lists and we will consider the family $\mathcal{F}_{\text{part}}$ of information regimes that partition L into disjoint sets $L_1, \ldots, L_p$.

A *distributed strategy* $\mathcal{A} = A_1, \ldots, A_p$ for the MVC problem is a p-tuple of algorithms with the following property. For any graph $G = (V, E)$, for any information regime $\mathcal{R}$, such that $\mathcal{R}(G) = L_1, \ldots, L_p$, $A_i(L_i) \subset V_i$ and $\mathcal{A}(G, \mathcal{R}) = \bigcup_{i=1}^{p} A_i(L_i)$ is a vertex cover for G, where V_i denotes the set of nodes whose adjacency lists are in L_i.

```
Input: L_i
{ V_i denotes the set of nodes whose adjacency lists are in L_i }
begin
  C_i := ∅; B_i := ∅;
  for each edge (u, v) such that u ∈ V_i and v ∉ V_i do
    if u ∉ C_i and v ∉ B_i then
    begin
      C_i := C_i ∪ {u};
      B_i := B_i ∪ {v};
    end;
  for each edge (u, v) such that u, v ∈ V_i do
    if u ∉ C_i and v ∉ C_i then
      C_i := C_i ∪ {u, v};
  return C_i
end.
```

Fig. 1. The double-matching algorithm.

The *competitive ratio* of $\mathcal{A}$ with respect to the family $\mathcal{F}_{\text{part}}$ is

$$R(\mathcal{A}, \mathcal{F}) = \max_{\mathcal{R} \in \mathcal{F}_{\text{part}}} \max_{G} \frac{|\mathcal{A}(G, \mathcal{R})|}{\text{opt}(G)}.$$

In the following sections we will show that $\min_{\mathcal{A}} R(\mathcal{A}, \mathcal{F}_{\text{part}}) = p$.

3 The upper bound

Recall that Gavril's algorithm looks for a maximal matching in the graph and then returns both the endpoints of any edge in the matching. It is easy to see that such a set of nodes is a vertex cover and that its cardinality is at most twice the cardinality of the minumum cover.

In the next theorem we will apply the same idea first to edges "shared" by two processors and then to the remaining edges.

Theorem 1 *For any $p \geq 2$, a strategy $\mathcal{A}$ for a p-processor system exists whose competitive ratio is at most p.*

Proof. Consider the strategy $\mathcal{A} = \{A_1, \ldots, A_p\}$ where algorithm A_i is described in Fig. 1, and assume that the edges are picked in any specified order in the **for** loops (e.g. in lexicographic order).

Let G be an input graph and $\mathcal{R}$ be an information regime in $\mathcal{F}_{\text{part}}$. We denote by $A_i^{(1)}(L_i)$ and $A_i^{(2)}(L_i)$ the set of nodes included in C_i during the first and the second **for** instruction, respectively. Clearly, all edges "seen" by processor P_i are covered by the set $A_i^{(1)}(L_i) \cup A_i^{(2)}(L_i) \cup B_i$. Then, in order to prove that $\mathcal{A}(G,\mathcal{R})$ is a vertex cover for G it suffices to show that, for any i,

$$B_i \subseteq \mathcal{A}^{(1)} = \bigcup_{k=1}^{p} A_k^{(1)}(L_k). \tag{1}$$

The proof is by induction on the number b of *bridge* edges, that is, edges whose endpoints "belong" to different processors (observe that each B_i contains only endpoints of bridge edges). If $b = 0$, then the proof is trivial. Suppose that we have $b+1$ bridge edges and that (u,v) is the last of these edges in the lexicographic order with $u \in V_i$ and $v \in V_j$. Let $\hat{L}_i$ and $\hat{L}_j$ denote the adjacency lists obtained from L_i and L_j, respectively, by deleting the edge (u,v), moreover, let $A_i^{(1)}(\hat{L}_i)$ and $\hat{B}_i$ (respectively, $A_j^{(1)}(\hat{L}_j)$ and $\hat{B}_j$) be the sets computed by the algorithm on input $\hat{L}_i$ (respectively, $\hat{L}_j$). By induction hypothesis,

$$\hat{B}_i, \hat{B}_j \subseteq \hat{\mathcal{A}}^{(1)} = \bigcup_{\substack{k=1\\k\neq i,j}}^{p} A_k^{(1)}(L_k) \cup A_i^{(1)}(\hat{L}_i) \cup A_j^{(1)}(\hat{L}_j).$$

We shall now prove that $B_i \subseteq \mathcal{A}^{(1)}$ (the proof for B_j is similar). To this aim, we distinguish the following two cases.

(i) $u \in A_i^{(1)}(\hat{L}_i) \vee v \in \hat{B}_i$: in this case $B_i = \hat{B}_i \subseteq \hat{\mathcal{A}}^{(1)} \subseteq \mathcal{A}^{(1)}$.

(ii) $u \notin A_i^{(1)}(\hat{L}_i) \wedge v \notin \hat{B}_i$: in this case $B_i = \hat{B}_i \cup \{v\}$ and $u \notin \hat{B}_j$ (since $u \notin \hat{\mathcal{A}}^{(1)}$ and $\hat{B}_j \subseteq \hat{\mathcal{A}}^{(1)}$). If $v \in A_j^{(1)}(\hat{L}_j)$ then, clearly, $B_i \subseteq \hat{\mathcal{A}}^{(1)} \subseteq \mathcal{A}^{(1)}$, otherwise v will be put into $A_j^{(1)}(L_j)$ when considering edge (u,v) so that $B_i \subseteq \mathcal{A}^{(1)}$.

We have thus shown that $\mathcal{A}(G,\mathcal{R})$ is a vertex cover. In order to prove that its competitive ratio is at most p, let $n_k = \sum_{i=1}^{p} |A_i^{(k)}(L_i)|$ for $k = 1, 2$. Clearly, an index i must exist such that $|A_i^{(1)}(L_i)| \geq n_1/p$. This set $A_i^{(1)}(L_i)$ then corresponds to a set of at least n_1/p disjoint edges. Moreover, the set $\bigcup_{i=1}^{p} A_i^{(2)}(L_i)$ corresponds to another set of $n_2/2$ disjoint edges. From (1) It also follows that the union of these two sets is still a set of disjoint edges. That is, G contains a matching of at least $n_1/p + n_2/2$ edges. Thus, any vertex cover for G must contain at least $n_1/p + n_2/2 \geq (n_1+n_2)/p$ nodes, that is,

$$\frac{|\mathcal{A}(G,\mathcal{R})|}{\mathrm{opt}(G)} \leq \frac{n_1+n_2}{(n_1+n_2)/p} = p.$$

We can conclude that the competitive ratio of $\mathcal{A}$ is at most p. $\Box$

4 The lower bound

In order to prove that the result of the previous section is tight, let us first show that, for any strategy $\mathcal{A}$, an information regime $\mathcal{R} \in \mathcal{F}_{\mathrm{part}}$ exists such that $R(\mathcal{A}, \mathcal{R}) \geq 2$. Let $K_{i,j}^{n,n}$ denote the distributed instance in which the complete bipartite graph $K^{n,n}$ with vertex classes U and W is distributed in the following way: $V_i = U$, $V_j = W$, and all other processors know nothing (recall that $\mathcal{R}(K^{n,n}) = L_1, \ldots, L_p$ and, for any k, V_k is the set if nodes whose adjacency lists are in L_k). Then, for any strategy $\mathcal{A}$, either P_i or P_j has to choose all its nodes when running $\mathcal{A}$ with input $K_{i,j}^{n,n}$ (otherwise, an uncovered edge exists). Without loss of generality, we can assume that P_i chooses all its nodes. Let us then consider the new distributed instance in which the vertices in W are pairwise connected, thus forming a clique of order n. Clearly, P_i still chooses all its nodes since its subinstance is not changed. Moreover, P_j is also forced to choose at least $n-1$ of its nodes. The optimum solution then contains n nodes while the solution computed by $\mathcal{A}$ contains at least $2n-1$ nodes. That is, the competitive ratio is at least 2.

In order to increase the above lower bound, we will show in the next theorem how to find, for any strategy $\mathcal{A}$, and for infinitely many n, a distributed instance $\mathcal{G}$ in which a processor P_j knows n nodes and the other processors P_i share at least $(p-1)(n-1)$ nodes which are all connected to the n nodes of P_j. Moreover, each P_i with $i \neq j$ chooses all its nodes when running strategy $\mathcal{A}$ with input $\mathcal{G}$. We can then modify the instance by pairwise connecting all nodes of P_j. The optimum thus contains n nodes while the solution computed by $\mathcal{A}$ contains at least $p(n-1)$ nodes. That is, the competitive ratio is at least p.

In order to prove the theorem, we need the following technical result.

Lemma 2 *Let $a_1, \ldots, a_N, b_1, \ldots, b_N$ be $2N$ nonnegative numbers such that, for any n,*

(i) $0 \leq a_n, b_n \leq N$.
(ii) If $a_n < n-1$, then $b_k \geq n$ for $k = a_n + 1, \ldots, n-1$.
(iii) If $b_n < n-1$, then $a_k \geq n$ for $k = b_n + 1, \ldots, n-1$.

Then

$$\sum_{n=1}^{N}(a_n + b_n) \geq 2\sum_{n=1}^{N}(n-1). \tag{2}$$

Proof. We proceed by induction on N. For $N = 1$ the proof is trivial since both a_1 and b_1 are nonnegative.

Assume that (2) has been proven for any $N' < N+1$ and let $a_1, \ldots, a_N, a_{N+1}$, $b_1, \ldots, b_N, b_{N+1}$ be $2(N+1)$ nonnegative numbers satisfying the hypothesis of the lemma. Let us consider the case in which both a_{N+1} and b_{N+1} are smaller than N (the other cases are proved similarly). Then $a_{N+1} = N - h$ and $b_{N+1} = N - k$ with $h, k > 0$. From the hypothesis it follows that

$$a_{N-k+1}, \ldots, a_N = N + 1 \qquad \text{and} \qquad b_{N-h+1}, \ldots, a_N = N + 1.$$

For any n with $N-k+1 \leq n \leq N$ and for any m with $N-h+1 \leq m \leq N$, let us define $a'_n = N$ and $b'_m = N$. The $2N$ numbers $a_1, \ldots, a_{N-k}, a'_{N-k+1}, \ldots, a'_N$, $b_1, \ldots, b_{N-h}, b'_{N-h+1}, \ldots, b'_N$ clearly still satisfy the hypothesis of the lemma. This, in turn, implies that

$$\begin{aligned}
\sum_{n=1}^{N+1}(a_n + b_n) &= \sum_{n=1}^{N}(a'_n + b'_n) + (h+k) + (a_{N+1} + b_{N+1}) \\
&\geq 2\sum_{n=1}^{N}(n-1) + (h+k) + (a_{N+1} + b_{N+1}) \\
&= 2\sum_{n=1}^{N+1}(n-1)
\end{aligned}$$

where the inequality is due to the inductive hypothesis. The lemma thus follows. □

We are now in a position to prove the main result of this section.

Theorem 3 *For any strategy $\mathcal{A} = \{A_1, \ldots A_p\}$, and for any integer N_0, a graph G, an information regime $\mathcal{R} \in \mathcal{F}_{\text{part}}$, an index j, and an integer $n_0 > N_0$ exist such that*

(i) $|V_j| = n_0$.
(ii) $\sum_{i \neq j} |V_i| \geq (p-1)(n_0 - 1)$.
(iii) For any $i \neq j$, each vertex in V_i is connected to each vertex in V_j.
(iv) For any $i \neq j$, $A_i(L_i) = V_i$.

Proof. For any i, j, m, and n, let $K_{i,j}^{m,n}$ denote the distributed instance in which the complete bipartite graph $K^{m,n}$ with vertex classes U and W is distributed in the following way: $V_i = U$, $V_j = W$, and all other processors know nothing. For any integer n, let $c_{i,j}^n$ be the maximum m such that P_i with input $K_{i,j}^{m,n}$ chooses all its nodes (see Fig. 2 where the black nodes have been chosen and the white nodes may or may not have been chosen).

Observe that, for any i, j, and n, if $c_{i,j}^n = m < n-1$ then $c_{j,i}^k \geq n$ for $k = m+1, \ldots, n-1$: from Lemma 2, we have that for any i, j, and N, the

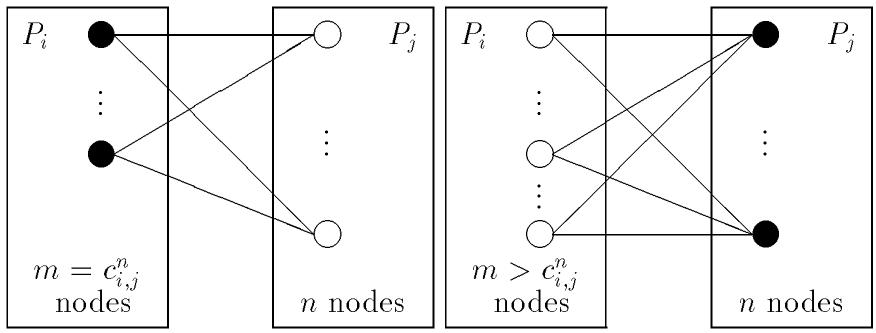

Fig. 2. The definition of $c_{i,j}^n$

following inequality holds:

$$\sum_{n=1}^{N}(c_{i,j}^n + c_{j,i}^n) \geq 2\sum_{n=1}^{N}(n-1).$$

It then follows that

$$\sum_{n=1}^{N}\sum_{j=1}^{p}\sum_{\substack{i=1\\i\neq j}}^{p} c_{i,j}^n = \sum_{\substack{i,j=1\\i<j}}^{p}\sum_{n=1}^{N}(c_{i,j}^n + c_{j,i}^n) \geq p(p-1)\sum_{n=1}^{N}(n-1). \tag{3}$$

Assume now that an N_0 exists such that, for any $n > N_0$ and for any j,

$$\sum_{\substack{i=1\\i\neq j}}^{p} c_{i,j}^n < (p-1)(n-1)$$

and let

$$\sigma = \sum_{n=1}^{N_0}\sum_{j=1}^{p}\sum_{\substack{i=1\\i\neq j}}^{p} c_{i,j}^n.$$

Then, for any $N > N_0$, we have that

$$\begin{aligned}\sum_{n=1}^{N}\sum_{j=1}^{p}\sum_{\substack{i=1\\i\neq j}}^{p} c_{i,j}^n &= \sigma + \sum_{n=N_0+1}^{N}\sum_{j=1}^{p}\sum_{\substack{i=1\\i\neq j}}^{p} c_{i,j}^n \\ &\leq \sigma + p(p-1)\sum_{n=N_0+1}^{N}(n-1) - (N-N_0)p\end{aligned}$$

which, for N sufficiently large, contradicts (3).

Thus, for any integer N_0, an index j and an integer $n_0 > N_0$ exist such that

$$\sum_{\substack{i=1 \\ i \neq j}}^{p} c_{i,j}^{n_0} \geq (p-1)(n_0-1).$$

The distributed instance $\mathcal{G}$ is then defined as a star of bipartite graphs in which processor P_j knows n_0 nodes and each processor P_i with $i \neq j$ knows $c_{i,j}^{n_0}$ nodes which are all connected to each node of P_j [1]. Clearly, this graph satisfies the theorem. $\square$

As a consequence of the above theorem, we then have the following result, which states the optimality of the upper bound shown in the previous section.

Corollary 4 $R(MVC, \mathcal{F}_{\mathrm{part}}) \geq p$.

5 Conclusion and Open Problems

We studied the problem of computing approximate vertex covers of a graph on the basis of partial information. We showed an optimal algorithm whose competitive ratio is equal to the number of processors.

In this paper we fixed a particular family of information regimes: it would be interesting to consider other families of information regimes and find trade-offs between competitive ratio and redundancy.

The algorithm given in Sect. 3 runs in polynomial time, even if the lower bound holds for algorithms of unbounded complexity. In general, however, the competitive ratio of a problem may increase if we restrict ourselves to polynomial-time algorithms. Investigating the relationship between time complexity and competitive ratio may be an interesting direction for further research.

In particular, in the case of the minimum vertex cover problem, given full information, no polynomial time algorithm is known to achieve a better asymptotic performance ratio than 2: it would be interesting to find a non-trivial information regime where a competitive ratio equal to 2 is achievable in polynomial time.

[1] To be more precise, we should assume that, for any k, each vertex assigned to processor P_k has a “name” depending on the value of k, such that no two distinct processors own two nodes with the same name. For instance, we can assume that if processor P_k has n_k nodes, then their identity is $k, p+k, \ldots, (n_k-1)p+k$.

Acknowledgements

We wish to thank Alberto Marchetti-Spaccamela for several helpful discussions.

References

[1] Deng, X. and C. H. Papadimitriou. "Distributed decision-making with incomplete information". In *Proceedings of the 12th IFIP Congress*, 1992.

[2] Garey, M. R. and D. S. Johnson. "Computers and Intractability. A Guide to the Theory of NP-Completeness". Freeman, 1979.

[3] Irani, S. and Y. Rabani "On the value of information in coordination games". In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 12–21, 1993. An extended version is to appear in the *SIAM Journal of Computing*.

[4] Karp, R. M. "Reducibility among combinatorial problems". In *Complexity of Computer Computations*. Plenum Press, 1972.

[5] Motwani, R. "Lecture notes on approximation algorithms". 1992.

[6] Papadimitriou, C.H. "The value of information". In *Proceedings of World Congress of Economics*, 1992.

[7] Papadimitriou, C.H. and M. Yannakakis. "On the value of information in distributed decision making". In *Proceedings of 10th ACM Symposium on Principles of Distributed Computing*, pages 61–64, 1991.

[8] Papadimitriou, C.H. and M. Yannakakis. "Linear programming without the matrix". In *Proceedings of 25th ACM Symposium on Theory of Computing*, 1993.