Minimum Vertex Cover, Distributed Decision-Making, and Communication Complexity*

Extended abstract

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 and we consider the *distributed decision-making* and the *communication complexity* frameworks. In the first framework we do not allow communication among the processors: in this case, we show an optimal algorithm whose performance ratio is equal to p where p is the number of processors. In the second framework two processors are allowed to communicate in order to find an approximate solution: in this latter case, we show a linear lower bound on the communication complexity of the problem.

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 [10]. A straightforward approximation algorithm, based on the idea of a maximal matching, was successively developed by Gavril (according to [5]) with a performance ratio no greater than 2. Several other approximation algorithms are presented in the lecture notes of Motwani [13]. In this paper we analyse the complexity of finding approximate solutions for the MVC problem under two basic frameworks.

In the first one, we study this problem as one of *distributed decision-making with incomplete information* [4, 15, 16, 17], that is, we assume that the vertex cover is chosen by independent processors, each knowing only a part of the graph and acting in isolation. In particular, we assume that the adiacency 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

* 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*.

distributed algorithms that always produce feasible solutions (that is, vertex covers) and achieve, in the worst case, a *reasonable* performance 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 Sect. 2 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 performance ratio of 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 graph problem has to be solved in a distributed fashion and neither a centralized control nor a complete information are available. 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' [17]. Intuitively, this theory should allow to compare in terms of performance ratios two different *information regimes*, that is, two different ways of distributing the input among the processors.

In the second framework, instead, we study the communication complexity of the MVC problem. In standard communication complexity, two processors interact in order to compute a function f of their respective inputs x and y. The question is: how much information do the two processors need to exchange to correctly compute the value $f(x, y)$? The minimum number of bits that must be communicated is the *deterministic communication complexity* of f. This complexity measure was introduced by Yao [22] and has been shown to be tightly related to time-area tradeoffs in VLSI [1, 7, 8, 11, 12, 19, 21], circuit complexity [9], and combinatorial optimization [20]. It has also been studied for its own sake as an interesting model of computation. Indeed, non-deterministic, probabilistic, and alternating variants have been considered and several complexity classes analogous to the more notorius ones in Turing machine complexity have been defined [2, 14, 18]. In this paper we consider the communication complexity of computing, for any graph G and for any rational $\epsilon > 1$, a vertex cover for G whose performance ratio is smaller than ϵ. In particular, we assume that ϵ is known by both processors and the adiacency list of each node of G is known by only one processor. In Sect. 3 we prove that the communication complexity of this function is at least $(1 - H(8(\epsilon - 1)))n/8$ where $H(x) = -(x \log x + (1 - x)\log(1 - x))$ is the entropy of x and n denotes the number of nodes of the graph. The proof is based on an interesting relation with the area of error-correcting codes. We also observe that in the case of planar graphs $n \log n$ bits are sufficient to compute an optimum solution so that, for these graphs, our lower bound is not far from being tight.

2 MVC and distributed decision-making

Suppose that a graph G is described by the n adjacency lists of its n nodes. Given p processors with $p \geq 2$, the ith processor knows the adjacency lists of a subset V_i of nodes (without loss of generality, we shall assume that, for any i

and j, $V_i \cap V_j = \emptyset$). Let G_i be the subgraph of G induced by the set of edges (u,v) such that either u or v belongs to V_i.

A *distributed decision algorithm* A is an algorithm which, for any graph G and for any i, on the basis of the subgraph G_i produces a subset $A(G_i)$ of V_i such that $A(G) = \bigcup_{i=1}^{p} A(G_i)$ is a vertex cover of G. Moreover, let $opt(G)$ be the cardinality of a minimum vertex cover for G. The *performance ratio* of A is

$$R(A) = \max_{G} \frac{|A(G)|}{opt(G)}.$$

Theorem 1. *A distributed decision algorithm A exists whose performance ratio is at most p.*

Proof. Consider the following algorithm where the edges are supposed to be lexicographically ordered.

begin
  $A(G_i) := \emptyset$; $B_i := \emptyset$;
  for each edge (u,v) such that $u \in V_i$ and $v \notin V_i$ **do**
    if $u \notin A(G_i)$ **and** $v \notin B_i$ **then**
    begin
      $A(G_i) := A(G_i) \cup \{u\}$;
      $B_i := B_i \cup \{v\}$;
    end;
  for each edge (u,v) such that $u, v \in V_i$ **do**
    if $u \notin A(G_i)$ **and** $v \notin A(G_i)$ **then**
      $A(G_i) := A(G_i) \cup \{u,v\}$;
end.

Let $A_1(G_i)$ and $A_2(G_i)$ be the set of nodes included in $A(G_i)$ during the first and the second **for** instruction, respectively. Clearly, all edges 'seen' by processor P_i are covered by the set $A_1(G_i) \cup A_2(G_i) \cup B_i$. Then, in order to prove that $A(G)$ is a vertex cover for G it suffices to show that, for any i, $B_i \subseteq A_1(G) = \bigcup_{k=1}^{p} A_1(G_k)$. The proof is by induction on the number b of *bridge* edges, that is, edges whose extremes are known by different processors (observe that each B_i contains only extremes 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 $A_1(G'_i)$, B'_i, $A_1(G'_j)$, and B'_j be the sets computed by the algorithm on input $G' = (V, E - (u,v))$. By induction hypothesis, $B'_i, B'_j \subseteq A_1(G') = \bigcup_{k=1}^{p} A_1(G'_k)$. We shall now prove that $B_i \subseteq A_1(G)$ (the proof for B_j is similar). To this aim, we distinguish the following two cases.

1. $u \in A_1(G'_i) \vee v \in B'_i$: in this case $B_i = B'_i \subseteq A_1(G') \subseteq A_1(G)$.
2. $u \notin A_1(G'_i) \wedge v \notin B'_i$: in this case $B_i = B'_i \cup \{v\}$ and $u \notin B'_j$ (since $u \notin A_1(G'_i)$ and $B'_j \subseteq A_1(G')$). If $v \in A_1(G'_j)$ then, clearly, $B_i \subseteq A_1(G') \subseteq A_1(G)$, otherwise v will be put into $A_1(G_j)$ when considering edge (u,v) so that $B_i \subseteq A_1(G)$.

We have thus shown that $A(G)$ is a feasible solution. In order to prove that its performance ratio is at most p, let $n_k = \sum_{i=1}^{p} |A_k(G_i)|$ for $k = 1, 2$. Clearly, an index i must exist such that $|A_1(G_i)| \geq n_1/p$. This set $A_1(G_i)$ then corresponds to a set of at least n_1/p disjoint edges. Moreover, the set $\bigcup_{i=1}^{p} A_2(G_i)$ corresponds to another set of $n_2/2$ disjoint edges. It is also clear 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,

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

We can conclude that the performance ratio of A is at most p. □

In order to prove that the result of the previous theorem is tight, let us first show that, for any distributed decision algorithm A, $R(A) \geq 2$. Let $K_{i,j}^{m,n}$ denote the instance in which the complete bipartite graph $K^{m,n}$ with vertex classes V_1 and V_2 is distributed in the following way: V_1 is assigned to processor P_i, V_2 is assigned to processor P_j, and all other processors know nothing. Then, for any algorithm A, either P_i or P_j has to choose all its nodes when running algorithm A with input $K_{i,j}^{m,n}$ (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 instance in which vertices in V_2 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. If $m = n$, then the optimum solution contains n nodes while the solution computed by the algorithm contains at least $2n-1$ nodes. That is, the performance ratio is at least 2.

In order to improve the above bound, we will show in the next theorem how to find, for any distributed decision algorithm A, an instance G_A 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 algorithm A with input G_A. 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 the algorithm contains at least $p(n-1)$ nodes. That is, the performance ratio is at least p.

Lemma 2. *For any distributed decision algorithm A and for any integer N_0, a graph G, an index j, and an integer $n_0 > N_0$ exist such that*

1. $|V_j| = n_0$.
2. $\sum_{i \neq j} |V_i| \geq (p-1)(n_0 - 1)$.
3. *For any $i \neq j$, V_i and V_j are the two vertex classes of a complete bipartite graph.*
4. *For any $i \neq j$, $A(G_i) = V_i$.*

Proof. Recall that, for any i, j, m, and n, $K_{i,j}^{m,n}$ denotes the instance in which the complete bipartite graph $K^{m,n}$ with vertex classes V_1 and V_2 is distributed in the following way: V_1 is assigned to processor P_i, V_2 is assigned to processor P_j, and all other processors know nothing. Given a distributed decision algorithm A and an integer n, let $c_{i,j}^n$ be the maximum m such that $A(G_i) = V_1$ (see Fig. 1 where the black nodes have been chosen and the white nodes may or may not have been chosen).

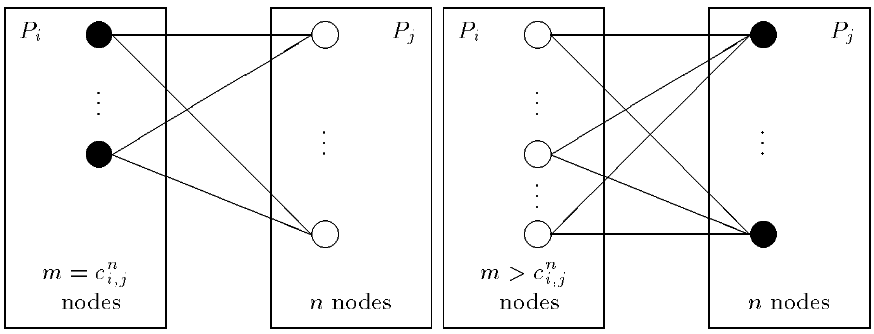

Fig. 1. The definition of $c_{i,j}^n$

Observe that, for any i, j, and n, if $c_{i,j}^n = m < n-1$ then $c_{j,i}^k \geq n$ for $k = m+1, \ldots, n-1$. This observation intuitively suggests that the sum of all these quantities is large enough. Indeed, in the Appendix we prove the following result.

Proposition 3. *For any i, j, and N, the following inequality holds:*

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

From the above proposition it then follows that

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

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

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

which, for N sufficiently large, contradicts (1).

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 graph 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. Clearly, this graph satisfies Conditions 1-4. □

As a consequence of the above lemma, we then have the following result.

Theorem 4. *No distributed decision algorithm has a performance ratio smaller than p.*

3 MVC and communication complexity

Let V_1 and V_2 be two set of nodes, $E_i \subseteq (V_1 \times V_2) \cup V_i^2$, for $i = 1, 2$, be two set of edges, and $G = (V_1 \cup V_2, E_1 \cup E_2)$ be a graph. For any ϵ, let $V(G, \epsilon)$ be the set of vertex covers V' of G such that

$$\frac{|V'|}{opt(G)} \leq \epsilon$$

where $opt(G)$ denotes the cardinality of a minimum vertex cover. Roughly speaking, $V(G, \epsilon)$ is the set of *approximate solutions within performance ratio* ϵ with respect to the instance G of the MVC problem.

The *MVC communication problem* with error ϵ is then as follows. Two processors P_1 and P_2 get inputs $G_1 = (V_1 \cup V_2, E_1)$ and $G_2 = (V_1 \cup V_2, E_2)$, respectively. Their task is to choose two sets $V_i' \subseteq V_i$ for $i = 1, 2$ such that $V_1' \cup V_2' \in V(G, \epsilon)$. The *communication complexity* of this problem is the minimum number of bits exchanged by the best protocol on a worst case input.

Let us consider the two instances shown in Fig. 2, that is, two rings of eight nodes such that the subinstance held by P_1 is the same in the two cases. The main difference is the 'parity' of the nodes seen by P_1 so that, in the first case,

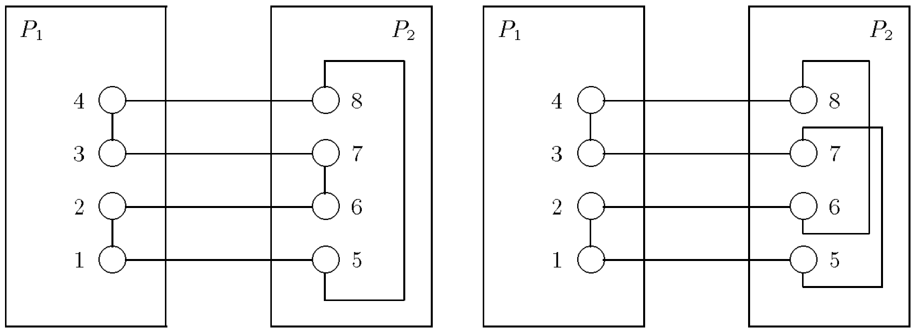

Fig. 2. Two rings of 8 nodes

node 2 belongs to the optimum solution if and only if node 3 does not belong to it while, in the second case, node 2 belongs to the optimum solution if and only if node 3 belongs to it. This means that if the processors execute a protocol that always yields an optimum solution, than P_1 will compute different subsolutions in the two cases, even if its subinstance is the same. This implies that at least one bit of communication is required. This reasoning can be generalized as follows: let $\mathcal{G}_n$ be the set of graphs G containing n disjoint rings $R_1, \ldots, R_n$ where each of these rings can be of one of the two types shown in Fig. 2. Clearly, $|\mathcal{G}_n| = 2^n$ and all these graphs are equal from P_1's point of view. On the other hand, an optimum solution for a graph $G \in \mathcal{G}_n$ cannot be an optimum solution for any other graph in $\mathcal{G}_n$. Thus, while executing a protocol yielding optimum solutions, P_1 will find different subsolutions for any instance in $\mathcal{G}_n$, that is, 2^n different computations are performed with respect to the same P_1's input. This, in turn, implies that different sequences of messages must be exchanged every time and that a sequence of messages exists whose length is at least n.

Let us now consider the case of ϵ-approximate solutions. Since the optimum solutions of any instance in $\mathcal{G}_n$ contains $4n$ nodes, an ϵ-approximate solution will contain at most $4n\epsilon = 4n + 4n(\epsilon - 1)$ nodes. Thus, the number of rings that are not optimally covered is at most $4n(\epsilon - 1)$. It is also easy to see that if $V(G, \epsilon) \cap V(G', \epsilon) \neq \emptyset$ where $G, G' \in \mathcal{G}_n$, then G differs from G' in at most $8n(\epsilon - 1)$ rings. Let $\mathcal{G}'_n$ be a subset of $\mathcal{G}_n$ such that any two instances in $\mathcal{G}'_n$ differ in less than $8n(\epsilon - 1)$ rings. Then $\log |G'_n|$ is a lower bound on the communication complexity of approximating the MVC problem within factor ϵ on graphs with $8n$ nodes. The problem of finding a large set $\mathcal{G}'_n$ with the required property is equivalent to the problem of finding a large set $\mathcal{C} \subset \{0,1\}^n$ such that for any two strings $s_1, s_2 \in \mathcal{C}$, the Hamming distance $d(s_1, s_2)$ (i.e., the number of bits where s_1 differs from s_2) is smaller than $8n(\epsilon - 1)$. Such a set $\mathcal{C}$ is indeed an *error correcting code* and a classical result of Gilbert [3, 6] allows us to state that, for any $\epsilon < 17/16$, an error correcting code $\mathcal{C}$ exists with $\log |\mathcal{C}| > (1 - H(8(\epsilon - 1)))n/8$ where H is the entropy function defined as $H(x) = -x \log x - (1 - x) \log(1 - x)$, for $0 < x < 1/2$.

We have thus shown the following result

Theorem 5. *For any $\epsilon < 17/16$, the communication complexity of the MVC communication problem with error ϵ is at least* $(1 - H(8(\epsilon - 1)))n/8$.

For any fixed $\epsilon < 17/16$ we thus have a *linear* lower bound. This result clearly holds even if we restrict ourselves to planar graphs. Since any planar graph with n nodes contains $O(n)$ edges, a trivial upper bound on the communication complexity of exactly solving the MVC problem is $O(n \log n)$, so that our lower bound is not far from being tight.

4 Conclusion and open problems

We studied the problem of computing approximate vertex covers of a graph on the basis of partial information and we analysed two basic frameworks. In the first one we do not allow communication among the processors: in this case, we showed an optimal algorithm whose performance ratio is equal to the number of processors. In the second framework two processors are allowed to communicate in order to find an approximate solution: in this latter case, we showed a linear lower bound on the communication complexity of the problem.

Some problems are left open by this work. From the distributed decision-making point of view, it would be interesting to find tradeoffs between the quantity of information available to each processor and the performance ratio, that is, to compare two different information regimes. From the communication complexity point of view, it would be interesting to pin down the precise communication complexity of approximating the MVC problem. For example, is the linear lower bound optimal? What about $\epsilon > 17/16$?

References

1. Aho, A. V., J.D. Ullman and M. Yannakakis. "On notions of information transfer in VLSI circuits". In *Proceedings of 15th ACM Symposium on Theory of Computing*, pages 133–136, 1983.
2. Babai, L., P. Frankl and J. Simon. "Complexity classes in communication complexity theory". In *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, pages 337–347, 1986.
3. Berlekamp, E.R. "Algebraic Coding Theory". McGraw-Hill, 1968.
4. Deng, X. and C. H. Papadimitriou. "Distributed decision-making with incomplete information". In *Proceedings of 12th IFIP*, 1992.
5. Garey, M. R. and D. S. Johnson. "Computers and Intractability. A Guide to the Theory of NP-completeness". Freeman, 1979.
6. Gilbert, E.N. "A comparison of signaling alphabets". *Bell System Tech. J.*, 31:504–522, 1952.
7. Hambrusch, S. E. and J. Simon. "Solving undirected graph problems on VLSI". *SIAM Journal of Computing*, 14:527–544, 1985.
8. Ja'Ja', J. "The VLSI complexity of selected graph problems". *Journal of the ACM*, 31:840–849, 1984.

9. Karchmer, M. and A. Wigderson. "Monotone circuits for connectivity require super-logarithmic depth". In *Proceedings of 20th ACM Symposium on Theory of Computing*, pages 539–550, 1988.
10. Karp, R. M. "Reducibility among combinatorial problems". In *Complexity of Computer Computations.* Plenum Press, 1972.
11. Lipton, R. J. and R. Sedgewick. "Lower bounds for VLSI". In *Proceedings of 13th ACM Symposium on Theory of Computing*, pages 300–307, 1981.
12. Melhorn, K. and E. M. Schmidt. "Las Vegas is better than determinism in VLSI and distributed computing". In *Proceedings of 14th ACM Symposium on Theory of Computing*, pages 330–337, 1982.
13. Motwani, R. "Lecture notes on approximation algorithms". 1992.
14. Papadimitriou, C. H. and M. Sipser. "Communication complexity". In *Proceedings of 14th ACM Symposium on Theory of Computing*, pages 196–200, 1982.
15. Papadimitriou, C.H. "The value of information". In *Proceedings of World Congress of Economics*, 1992.
16. 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.
17. Papadimitriou, C.H. and M. Yannakakis. "Linear programming without the matrix". In *Proceedings of 25th ACM Symposium on Theory of Computing*, 1993.
18. Paturi, R. and J. Simon. "Probabilistic communication complexity". *Journal of Computer and System Sciences*, 33:106–123, 1986.
19. Thompson, C. D. "Area-time complexity for VLSI". In *Proceedings of 11th ACM Symposium on Theory of Computing*, pages 81–88, 1979.
20. Yannakakis, M. "Expressing combinatorial optimization problems by linear programs". In *Proceedings of 29th IEEE Symposium on Foundations of Computer Science*, pages 223–228, 1988.
21. Yao, A. "The entropic limitations of VLSI computation". In *Proceedings of 13th ACM Symposium on Theory of Computing*, pages 308–311, 1981.
22. Yao, A. C. "Some complexity questions related to distributive computing". In *Proceedings of 11th ACM Symposium on Theory of Computing*, pages 209–213, 1979.

Appendix: Proof of Proposition 1

We shall prove the following. Let $a_1, \ldots, a_N, b_1, \ldots, b_N$ be $2N$ nonnegative numbers such that, for any n,

1. $0 \leq a_n, b_n \leq N$.
2. If $a_n < n-1$, then $b_k \geq n$ for $k = a_n + 1, \ldots, n-1$.
3. If $b_n < n-1$, then $a_k \geq n$ for $k = b_n + 1, \ldots, n-1$.

Then

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

The proof is by induction on N. For $N = 1$ the proof is trivial since, by property 1,

$$a_1 + b_1 \geq 0.$$

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 conditions 1-3. 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 properties 1-3 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 satisfy conditions 1-3. This, in turn, implies that

$$\sum_{n=1}^{N+1}(a_n + b_n) \geq 2\sum_{n=1}^{N}(n-1) + (h+k) + (a_{N+1} + b_{N+1}) = 2\sum_{n=1}^{N+1}(n-1).$$

The proposition thus follows. $\square$