RAIRO. RECHERCHE OPÉRATIONNELLE

V. TH. PASCHOS

A relation between the approximated versions of minimum set covering, minimum vertex covering and maximum independent set

RAIRO. Recherche opérationnelle, tome 28, nº 4 (1994), p. 413-433

<http://www.numdam.org/item?id=RO_1994__28_4_413_0>

Recherche opérationnelle/Operations Research
(vol. 28, n° 4, 1994, p. 413 à 433)

A RELATION BETWEEN THE APPROXIMATED VERSIONS OF MINIMUM SET COVERING, MINIMUM VERTEX COVERING AND MAXIMUM INDEPENDENT SET (*)

by V. Th. Paschos ([1])

Communicated by Pierre Tolla

Abstract. – *Let ρ be a universal constant denoting the approximation ratio of a hypothetical polynomial time approximation algorithm for the instances of the independent set problem with* $\frac{9}{20}n \le \alpha(G) \le \frac{11}{20}n$, *where G is a graph of order n and stability number $\alpha(G)$. Let finally suppose the existence of a (universally) constant-ratio-polynomial-time-approximation-algorithm for set covering problem. Then, there exists a polynomial time approximation algorithm for vertex covering problem with a ratio bounded above by* $\max\left\{\frac{9}{5}, 2 - \frac{9}{10}\rho\right\} + \varepsilon$ *for a ε arbitrarily small.*

Keywords: NP-complete problem, polynomial time approximation algorithm, set covering, vertex covering, independent set.

Résumé. – *Soit ρ une constante universelle représentant le rapport d'approximation d'un algorithme approché hypothétique pour les instances du problème du stable maximum vérifiant* $\frac{9}{20}n \le \alpha(G) \le \frac{11}{20}n$ *où G est un graphe d'ordre n et de nombre de stabilité $\alpha(G)$. Supposons qu'il existe un algorithme approché de rapport constant pour le problème de recouvrement minimum d'ensembles. Il existe alors un algorithme polynomial approché pour le problème de transversal minimum avec un rapport majoré par* $\max\left\{\frac{9}{5}, 2 - \frac{9}{10}\rho\right\} + \varepsilon$ *avec ε arbitrairement petit.*

Mots clés : Problème NP-complet, algorithme polynomial approché, recouvrement d'ensembles, transversal, stable.

1. INTRODUCTION

Given a graph $G = (V, E)$ of order n, a vertex covering (or vertex cover) is a subset $V' \subseteq V$ such that, for each edge $uv \in E$, at least one of u

(*) Received September 1993.

([1]) LAMSADE, Université Paris-Dauphine, place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France.

Recherche opérationnelle/Operations Research, 0399-0559/94/04/$ 4.00

and v belongs to V' and the minimum vertex covering problem (VC) is to find a vertex cover of minimum size [in what follows, we shall denote the cardinality of a minimum vertex covering of a graph G by $\tau(G)$]; an independent set is a subset $V' \subseteq V$ such that not any two vertices in V' are linked by an edge in G and the maximum independent set problem (IS) is to find an independent set of maximum size [in what follows, we shall denote the cardinality of a maximum independent set of a graph G by $\alpha(G)$].

Given a graph G, a maximum (maximal) independent set is the complement of a minimum (minimal) vertex covering with respect to the vertex set of the graph; so, the sum of the cardinalities of a maximum (maximal) independent set and of the associated minimum (minimal) vertex covering equals the order of the graph.

Also, given a collection $\mathcal{S}$ of subsets of a finite set C, a set cover for C is a subcollection $\mathcal{S}'$ of $\mathcal{S}$ such that every element of C belongs to at least one member of $\mathcal{S}'$ and the minimum set covering problem (SC) is to find a set cover of minimum size.

There is a picturesque graph formalism for SC. Every instance I of SC characterized by two sets $\mathcal{S}$ and C ($\mathcal{S} = \{s_1, \ldots, s_n\}$ denoting the family of the subsets of the set $C = \{c_1, c_2, \ldots, c_m\}$, where n and m are the cardinalities of $\mathcal{S}$ and C, respectively), can be represented by a bipartite graph $B = (S, C, E)$, called the characteristic graph of I, where the vertex set S denotes the family $\mathcal{S}$, the vertex set C the elements of the set C and $E = \{s_i c_j : c_j \in s_i\}$. Then, finding a minimum set covering for C becomes to find a minimum size subset of vertices of $S(B)$ "seeing" (sending edges to) all vertices of $C(B)$. On the other hand, every instance I of VC is expressed in terms of a graph $G = (V, E_G)$, which can be equivalently represented by a bipartite graph $B_G = (V, E_G, E')$ where E', the edge set of B_G, contains the pairs $v_i e_j$ such that $e_j \in E_G$ is incident in G to $v_i \in V$. Clearly, the instances of VC are exactly the instances of SC where every element of C is contained in exactly two subsets of $\mathcal{S}$, or equivalently, in the characteristic graph of these instances of SC, the degrees of the C-vertices are equal to 2. Hence, we can treat every instance of VC as an instance of SC, by considering G or equivalently B_G as the characteristic graph of this instance.

One of the most interesting theoretical problems in the complexity theory is to be able to "transfer" approximation results (positive, negative or conditional) from an NP-complete problem to another one via reductions preserving approximations ratios or to condition the existence (or the improvement) of existing approximation performances for some problems on the existence (or the improvement) of approximation performances for other ones.

Minimum vertex covering is a famous combinatorial problem for which we know a polynomial time approximation algorithm (PTAA) with a ratio equal to 2, namely the maximal matching algorithm [6], consisting in picking a maximal matching M in G and in putting in the approximated solution for VC both the extremities of the edges of the obtained matching (so, the cardinality of the so-obtained approximated solution is $2 \mid M \mid$). Up to now, all the researchers have failed to find another approximation algorithm with better performance guarantees.

On the other hand, recently, some researchers [1] have proved that VC does not admit a polynomial time approximation schema unless P = NP; the remarks made previously on the relation between SC and VC (the latter is a sub-problem of the former), permit to conclude immediately that the result of [1] is valid for SC also.

In the light of this remarkable result, the evaluation of a value constituting the lower bound for the approximation ratio of VC, or an improvement of the known approximation ratio for VC, would be of a great theoretical interest. Concerning the improvement of this ratio, we mention here the works of Bar-Yehuda and Even ([2], [3]) as well as the work of Monien and Speckenmeyer [9]. Their results concern an improvement of VC approximation ratio from 2 to $2 - \varepsilon$, but for an $\varepsilon = \dfrac{\log \log n}{2 \log n}$ [3] which tends to 0 whenever $n \to \infty$.

In this paper, we propose a conditional method for the improvement of VC ratio by an absolute constant ε, by considering VC as a restriction of SC. In fact we link, from an approximability point of view, three optimization problems, the VC, SC and IS. We prove then that a sufficient condition for the improvement of VC approximation ratio is the simultaneous existence of an approximation algorithm for SC and an approximation algorithm for IS on graphs for which $\alpha(G) \asymp n$ holds[1], where $\alpha(G)$ denotes the stability number of the graph G and the two approximation algorithms are supposed of constant approximation guarantees.

Our method consists, given an instance of SC, in constructing a new larger instance of the problem in which the cardinality of a set covering is a power of the cardinality of the solution in the initial instance.

[1] $f \asymp g$ if and only if $f = O(g)$ and $g = O(f)$.

This construction is performed by means of a kind of operation on bipartite graphs, called composition, where, given two bipartite graphs $B_i = (S_i, C_i, E_i)$ and $B_j\ (S_j, C_j, E_j)$, someone can construct the bipartite graph $B_i \star B_j = B_{ij} = (S_{ij}, C_{ij}, E_{ij})$ with $S_{ij} = S_i \times S_j$, $C_{ij} = C_i \times C_j$, and $E_{ij} = \{s_{tr}\, c_{kl} : s_t\, c_k \in E_i \wedge s_r\, c_l \in E_j\}$, where the operator $\times$ denotes the Cartesian product.

We denote by $B^p = (S^p, C^p, E^p)$ the graph obtained by the following inductive schema:

$$\begin{aligned} B_1 &= B \\ B^p &= B \star B^{p-1}. \end{aligned} \tag{1}$$

Given the graph B^p, we can see the set C^p as the union of $m = |C|$ sets of cardinality $m^{p-1} = |C^{p-1}|$, every set C^{p-1} as the union of $m = |C|$ sets of cardinality $m^{p-2} = |C^{p-2}|, \ldots$, every set C^2 as the union of $m = |C|$ sets of cardinality $m = |C|$ (the same correspondence holds also for the set S^p). For reasons of facility, $\forall\, i \leq p$, we will call by C^{i-1}-groups the m sets C^{i-1}, the union of which giving C^i (the same convention holds also for S^i); also, sometimes, for these C^{i-1}-groups (S^{i-1}-groups), we will say that they are embedded in C^i (S^i); moreover, whenever it is necessary, we will index them by the index of the vertex of B_G to which a group corresponds; finally, for every j, set C^{j-1} of the graph B^j is "seen" by two S^{j-1}-groups of B^j. This is due to the definition of B^j, since every c-vertex of B_G is "seen" by two s-vertices of B_G.

Let us, for example, consider the graph G of Figure 1(a) where we have denoted its vertices by s_1, s_2 and s_3 and its edges by c_1, c_2 and c_3, respectively; let us denote by S the set $\{s_1, s_2, s_3\}$ and by C the set $\{c_1, c_2, c_3\}$. The bipartite graph $B_G = (S, C, E')$ constructed as discussed previously is shown in Figure 1(b).

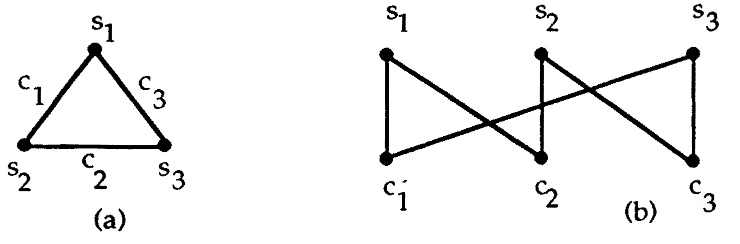

Figure 1. – (a) an instance B of VC; (b) the bipartite graph B_G.

In Figure 2, the graph $B^2 = B_G \times B_G$ constructed using schema 1 is shown. As one can see, every s-vertex (c-vertex) of B_G has been replaced by the whole set $S\,(C)$; we have so created three S-groups (S_1, S_2, S_3, where $S_i = s_i \times S$, $i = 1, 2, 3$) and three C-groups (C_1, C_2, C_3, where $C_i = c_i \times C$, $i = 1, 2, 3$), respectively. Every thick line between an S- and a C-group represents the whole of E_G (in other words, between an S- and a C-group corresponding to two vertices of B_G linked by an edge, we have drawn the whole graph B_G; we have chosen this representation for purposes of clarity of the figure). Everyone of the groups has three embedded vertices.

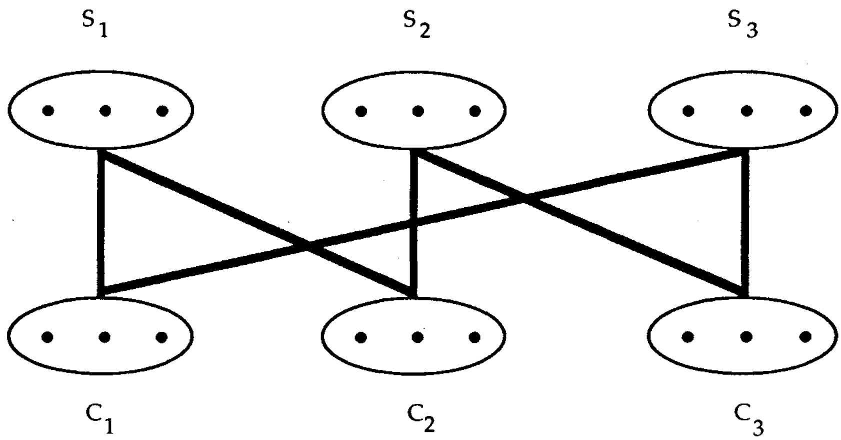

Figure 2. – The graph $B^2 = B \times B$B; the S- and C-groups are also indicated.

In Figure 3, the graph $B^3 = B \times B^2$ constructed using schema 1 is shown. Here, every s-vertex (c-vertex) of B_G has been replaced by the whole set $S^2\,(C^2)$; we have so created three S^2-groups (S_1^2, S_2^2, S_3^2, where $S_i^2 = S_i \times S$, $i = 1, 2, 3$) and three C^2-groups (C_1^2, C_2^2, C_3^2, where $C_i^2 = C_i \times C$, $i = 1, 2, 3$), respectively. Every thick line between two rectangles in Figure 3 represents the whole of the edges of B^2; in other words, between an S^2- and C^2-group corresponding to two vertices of B_G linked by an edge, we have drawn the whole graph B^2; we have, once more, chosen this representation for purposes of clarity of the figure[2]. Everyone of the S^2-groups (C^2-groups) has three embedded S-groups (C-groups), each one of them containing three embedded vertices.

[2] For example, the thick edge between the subsets S_1^2 and C_2^2 in Figure 3 means not only that there exist links between these two subsets, but also that these links are of the same nature as the ones of the initial graph B_G [Figure 1(b)].

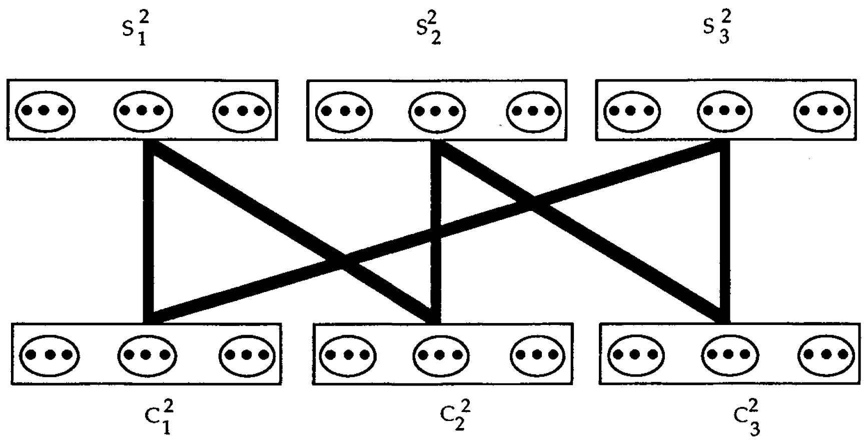

Figure 3. – The graph $B_3 = B \times B^2$; the S^2- and C^2-groups are also indicated.

In what follows, we will suppose that $\mathcal{A}$ is a hypothetical polynomial time ρ'-approximation algorithm for SC and $\mathcal{A}''$ is a hypothetical polynomial time ρ-approximation algorithm for IS on a family $\mathcal{G}_{\kappa_1 \kappa_2}$ of instances of IS, where ρ and ρ' are universal constants. The family $\mathcal{G}_{\kappa_1 \kappa_2}$ is defined as $\mathcal{G}_{\kappa_1 \kappa_2} = \{G : \kappa_1 n \leq \alpha(G) \leq \kappa_2 n\}$, where n is the order of the graph G and $\alpha(G)$ its stability number.

Throughout the paper and for reasons of facility, we fix the constants κ_1 and κ_2 to $\frac{9}{20}$ and $\frac{11}{20}$, respectively; we notice that the only changes performed in the result of section 2, due to the choice of precise values for the two constants, lie in the value of the constant substracted from 2 in the approximation ratio of VC. Moreover, for reasons of notation's simplifications, we shall denote by $\mathcal{G} = \left\{G : \frac{9}{20} n \leq \alpha(G) \leq \frac{11}{20} n\right\}$ the corresponding family of graphs.

We can suppose that the IS algorithm $\mathcal{A}''$, solving approximately the instances of the family $\mathcal{G}$, when applied to graphs not contained in $\mathcal{G}$, provides either solutions with ratio smaller[3] than ρ, or non-feasible solutions. Also, we will denote by τ' (indexed whenever necessary) the cardinality of the approximated (sub-optimal) solution for VC.

[3] $\rho \leq 1$, since IS is a maximization problem and the adopted approximation measure is defined as the ratio $\frac{\alpha'}{\alpha(G)}$, where α' is the size of the approximate solution for IS provided by an approximation algorithm [5].

2. THE RESULT

2.1 An algorithm for vertex covering

In what follows, we use the terminology of [10] where exposed vertices (with respect to a maximal matching M) are called the vertices that are not saturated by the edges of M; given also an edge uv of M, $u\,(v)$ is called the mate of $v\,(u)$. In order to use the least possible of notations concerning the optimal and the approximated solutions of the different instances, we will use the same notations for variables representing these quantities within the algorithm, and for constants representing the same quantities within the proof of the theorem of section 2.2. Finally, let us notice that the variables T', τ' appear twice in algorithm 1 (in steps [5] and [7]). This is no so missleading since we can see step [7] as an assignment of the final solution (solution value) to the variable $T'\,(\tau')$.

Algortihm 1 (algorithm $\mathcal{A}'$) is the polynomial time approximation algorithm, the (approximation) performance of which we analyze in section 2.2. In fact, $\mathcal{A}'$ is really polynomial since k depends only on ε which is independent of both n and m (as size of the VC instance can be considered the quantity n; on the other hand, m, the edge set cardinality of the instance, is bounded by n^2). In fact, step [5], which is the most expensive step of the algorithm, has a time complexity bounded above by $O\,(m^{k-1})$; on the other hand, in the worst case, the combination of steps [5] and [6] induce a complexity bounded above by nm^{k-1}, both these quantities being polynomial on n. As one can see, algorithm 1 uses three other algorithms as procedures, namely the hypothetical algorithm $\mathcal{A}$ for SC, the hypothetical algorithm $\mathcal{A}''$ for IS, and the maximal matching algorithm for VC.

Concerning step [5] for $k = 2$ for example, let us denote by M_i, $i = 1, 2, \ldots, n$, the subsets of T_2' in the S-group S_i; then, the solution T' is obtained by taking one of the subsets of T_2' of minimum cardinality which "sees" (covers) all of the vertices of a C-group of B^2 (the minimum being taken over the distinct C-groups), or more formally: $T' = \{M_i \cup M_j : |M_i \cup M_j| = \min_{l,\,t=1,\,2,\ldots,\,n} |M_l \cup M_t : s_l\, s_t \in (G)\}\,|\,\}$.

Really, since in B every vertex c_q "is seen by" two s-vertices, s_l, s_t, then by the construction of the graph B^p [expression (1)], vertex c_q corresponds to a C^{p-1}-group C_q^{p-1} of B^p receiving edges issued only from two S^{p-1}-groups S_l^{p-1}, S_t^{p-1} of B^p corresponding to the vertices s_l, s_t of B_G; consequently, the part of the solution T_p' covering the elements of the group C_q^{p-1} is contained in the groups S_l^{p-1}, S_t^{p-1}.

On the other hand, the "chain" of the conditions $\tau'_i \geq n \frac{\tau'_{i-1}}{2}$, $i = 2, \ldots, k$, verified in step [5] assures, as we shall see in section 2.2, that $\tau'^k \geq \left(\frac{n}{2}\right)^{k-1} \tau'_1$ and since the cardinality τ' of the final VC-solution found in step [6] is smaller than or equal to τ'_1, we have $\tau'^k \geq \left(\frac{n}{2}\right)^{k-1} \tau'$.

Moreover, concerning step [6], let us notice that in graph B^i, $2 \leq i \leq k$, every S^{i-1}-group of the set S^i corresponds to an S-vertex of B_G and, equivalently, to a vertex of G; so, the sets P and Q are well-defined on both graphs B_G and G. Furthermore, in section 2.2, we prove that the graph BG defined on color classes P and Q is really a bipartite graph.

Let us now revisit the example of section 1 in order to make clearer step [5] (and, partially, step [6]) of algorithm 1.

ALGORITHM 1: Algorithm $\mathcal{A}'$ solving approximately VC.

[1] *Given the graph* $G = (V, E)$, *pick a maximal matching on* G; *store as candidate solution for VC the vertices incident to the edges of the maximal matching just obtained.*

[2] *Construct the characteristic graph* $B = B_G = (S, C, E')$, *with* $S = V$ *and* $C = E$.

[3] *Fix an arbitrarily small universal constant* ε *and construct the graph* B^k *[inductive schema of expression (1)], where* k *is the smallest integer for which* $\rho'^{\frac{1}{k}} \leq 1 + \varepsilon$.

[4] *Construct* B^k; *execute* $\mathcal{A}$ *on the instance of SC represented by* B^k.

[5] *For* $i = k$ *do the following:*

for every C^{i-1}*-group* C_r^{i-1} *of set* C^i *in the graph* B^i, *consider the two* S^{i-1}*-groups "seeing"* C_r^{i-1} *and find the elements of* T'_i, *lying into these* S^{i-1}*-groups, which cover* C_r^{i-1}; *form a solution* $T'_{i-1,\,r}$ *for* C_r^{i-1} *by taking the set of the elements of* T'_i *found just above, by projecting their indices onto their* $i-1$ *last coordinates and then by removing (eventual) duplications;*

form a SC-solution T'_{i-1} $(|T'_{i-1}| = \tau'_{i-1})$ *for the graph* B^{i-1}, *where* $T'_{i-1} = \min_{1 \leq r \leq m} \{T'_{i-1,\,r}\}$;

if, for the value of τ'_{i-1}, $\tau'_i \geq n \frac{\tau'_{i-1}}{2}$, *then repeat step [5] until* $i = 2$, *with* $i-1$ *instead of* i, *else go to step [6];*

execute $\mathcal{A}''$ *on* G *and let* α' *be the cardinality of the obtained solution* S';

if S' *is feasible and moreover* $\alpha' \geq \frac{9}{20} \rho n$, *then store the set* $T'' = V \backslash S'$ *as candidate solution; go to step [7].*

[6] *(We are in the case where* $\exists\, i \leq k$, $\tau'_i < n \frac{\tau'_{i-1}}{2}$.*)*

Let $P\,(Q)$ *be the set of the* S^{i-1}*-groups of* B^i *with more than, or equal to (less than),* $\frac{\tau'_{i-1}}{2}$ *elements of* T'_i;

construct the graph $BG = (P, Q, E')$, *which is the bipartite graph resulting from* G *by removing all the edges between the vertices of* G *corresponding to the elements of* P *and obtain a maximum matching* M *on* BG; *let* PS, PE (QS, RE) *be the saturated and the exposed vertices in* $P\,(Q)$ *with respect to* M;

if $PE = \varnothing$, *or* M *is perfect, then* $P = PS$ *is an optimal vertex covering for* G;

else, start from set QE and consider the set PS' of the members of PS adjacent to the members of QE; consider the set QS' of the mates of the members of PS'; augment PS' by inserting in this set all the vertices adjacent to the members of QS' that are not already in PS'; augment also QS' by taking into account the mates of the vertices recently added to PS' and repeat this procedure until no more vertices can be added to PS'; let G' and G'' be the subgraphs of G induced by the sets $PS' \cup QS' \cup QE$ and $(P \backslash PS') \cup (Q \backslash (QS' \cup QE))$, respectively;

take as solution of G' the set PS';

go to step [1] and replace G by G''.

[7] *The final solution T' ($|T'| = \tau'$) for G is the smallest set between*

(i) *the candidate solution obtained in step [1],*

(ii) *the union of T_1' (obtained in step [5]) with the union of the sets PS' created from the (eventually multiple) executions of step [6] and*

(iii) *the union of T'' (obtained in step [5]) with the union of the sets PS' created from the (eventually multiple) executions of step [6].*

For $k = 2$ and following the above remark, let us suppose that $\mathcal{A}$ has found a solution T_2' for B^2 constituted from the first vertex of the S_1-group, the second vertex of the S_2-group and the third vertex of the S_3-group (*Fig.* 2)[4]; it is easy to see that these three vertices "see" all of the vertices of C^2, constituting so a solution for the SC-instance represented by B^2. Now, the C^1-group of B^2 is seen by the set $\{s_{11}, s_{33}\}$, the projection on the second index for both vertices (that is the set $\{s_1, s_3\}$) constituting a solution for the original SC instance; other solutions could be the sets $\{s_1, s_2\}$ (for the C_2-group) and $\{s_2, s_3\}$ (for the C_3-group)[5]. Since, for all of the C-groups of B^2, the obtained solutions have the same cardinality, the solution T' can be one of the three sets just mentioned; if this was not true, then T' would be the minimum cardinality so obtained set.

Finally, let us discuss the case $k = 3$ (*Fig.* 4) and show how a solution for the first (leftist) C_1-group embedded in the C_1^2-group of set C^3 can be constructed[6]. The subset of T_3' which "sees" the C-group C_1 embedded in the C^2-group C_1^2 embedded in set C^3 contains the vertex-set $\{s_{111}, s_{122}, s_{133}, s_{311}, s_{322}, s_{333}\}$. The projection of the indices of these vertices onto their two last components gives the set $\{s_{11}, s_{22}, s_{33}\}$ constituting a solution for the SC instance represented by B_2; next, we can obtain a solution $T' = \{s_1, s_3\}$ for the considered C-group as described just above.

[4] Following the notation we have used when we defined the composition of two bipartite graphs, we could call these vertices s_{11}, s_{22} and s_{33}, respectively.

[5] Let us remark here that another solution could be the S-vertices of B_G corresponding to those S-groups of B^2 containing non-empty subsets of T_2'; for our example this solution is trivial since all S-groups of the set S^2 contain some members of T_2'.

[6] This group, following the adopted notation, contains the vertices c_{111}, c_{112} and c_{113}.

On the other hand, concerning the second part of step [5] of algorithm $\mathcal{A}$, the feasibility test can be performed polynomially by taking the candidate solution S' and by verifying that it really constitutes an independent set; also, since ρ is supposed to be a priori known and, moreover, n is the order of the graph (instance of VC), the test $\alpha' \geq \frac{9}{20}\rho n$ is meaningfull (in fact, it is the case (**a2**) of the theorem in section 2.2).

Let us now have a small discussion on step [7] of $\mathcal{A}'$. Cases (ii) and (iii) are due to the following configuration: for an $i_1 < k$, step [6] is executed and a set PS' is obtained, as well as a partition of G into two subgraphs G', G''; then, algorithm 1 is re-executed with G'' instead of G; (a) let us suppose that $G'' \notin \mathcal{G}$; then, if the condition of step [5] $\left(\tau_i' \geq n\frac{\tau_{i-1}'}{2}, \forall i \leq k\right)$ is verified, the final solution is the union of the set PS' and of the solution-set obtained from the execution of step [5]; on the other hand, if the condition of step [5] is not verified, then step [6] is executed and a new set PS' and a new partition G', G'' of G (recall that G is now the graph G'') is obtained; then, the solution of VC will be the union of the two sets PS' obtained and of the solution of the new graph G''; moreover, algorithm $\mathcal{A}'$ is re-executed with the new G'' in place of G;... [case (ii)]; (b) let us now suppose that during an iteration (re-execution) of algorithm 1 the graph G'', replacing G, belongs to the class $\mathcal{G}$; then, the final solution for VC is the union of the sets PS' produced during the anterior executions of $\mathcal{A}'$ with the set T'' produced by step [5] during the last execution of the algorithm [case (iii)], or the solution obtained as we have just described in case (a).

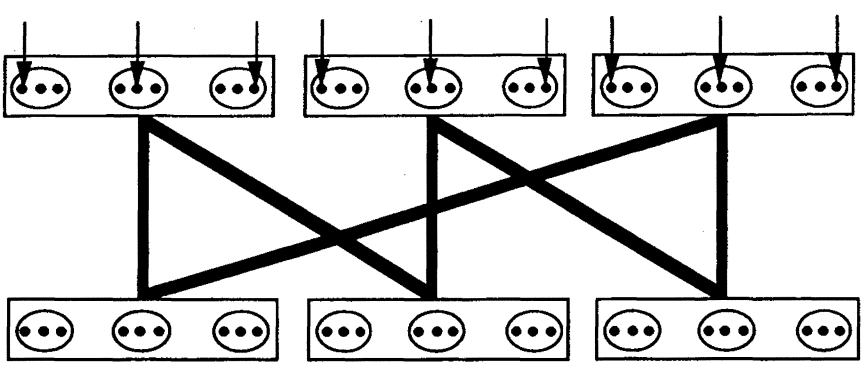

Figure 4. – A solution T_3' for B^3; the arrows show the members of T_3' (using the notation adopted in the definition of the composition of two bipartite graphs, the indicated vertices are s_{111}, s_{122}, s_{133}, s_{222}, s_{311}, s_{322}, s_{333}).

2.2 The theorem

THEOREM: *Let ρ be the approximation ratio of a polynomial time approximation algorithm $\mathcal{A}''$ solving independent set on graphs in $\mathcal{G}$, and let us suppose the existence of a polynomial time constant-ratio-approximation algorithm $\mathcal{A}$ for set covering. Then, algorithm 1 is a polynomial time approximation algorithm for vertex covering achieving an approximation ratio bounded above by* $\max\left\{\frac{9}{5}, 2 - \frac{9}{10}\rho\right\} + \varepsilon$, *for a positive constant ε arbitrarily small.*[7]

In order to prove the theorem, we examine two cases, namely, $\alpha(G) \le \frac{9}{20}n$ and $\alpha(G) \ge \frac{9}{20}n$. The proof of the first case is easy and straightforward. For the second case, starting from an instance G (or equivalently B_G) of VC, we use iteratively the inductive schema of expression (1) and we examine the cases **(a)** $\tau'_i \ge n\frac{\tau'_{i-1}}{2}$, where i denotes the ith iteration of schema (1) and τ'_i, τ'_{i-1} denote the approximated vertex covering cardinalities for the bipartite graphs B^i and B^{i-1} produced, respectively, during the iterations i and $i-1$ of schema (1), and **(b)** $\tau'_i < n\frac{\tau'_{i-1}}{2}$, where i, τ'_i and τ'_{i-1} are as in case **(a)**. For case **(a)**, we distinguish two subcases, namely **(a1)** $\alpha(G) \ge \frac{11}{20}n$ and **(a2)** $\frac{9}{20}n \le \alpha(G) \le \frac{11}{20}n$. In all, for case **(a)**, we prove that there always exists a $\beta > \frac{1}{2}$ such that $\tau'_k \ge \beta^{k-1}\tau'^k$, this fact, as we prove at the end of section 2.2, entailing an approximation ratio for VC strictly smaller than 2. For case **(b)**, we partition the vertices of G into two subsets such that the subgraph G' induced by the one of these sets is an instance where VC is polynomially solved, and the subgraph G'' induced by the other one admits the hypotheses of case **(a)**; moreover, we prove that the union of the approximated solutions of G' and G'' constitutes a solution for G and moreover that the cardinality of an optimal vertex covering of G is greater than, or equal to, the cardinality of the union of the optimal solutions of G' and G''. So, if for case **(a)** one can find a polynomial time approximation

[7] $2 - \frac{9}{10}\rho + \varepsilon > 2 - \frac{9}{10}\rho \overset{\rho<1}{>} 2 - \frac{9}{10} > 0.$

algorithm with ratio strictly smaller than 2, then one can obtain an algorithm of ratio even smaller than the one case (**a**) also for the graphs admitting the hypotheses of case (**b**). Then, the only remaining question is to show how algorithm 1 achives such a ratio; hence, we conclude the proof by answering this question.

Whenever $\alpha(G) \leq \frac{9}{20} n$, and since $\alpha(G) + \tau(G) = n$, we have $\tau(G) \geq \frac{11}{20} n$; consequently, given that any minimal vertex covering is at most of cardinality n (recall that by n we denote the order of G), any suboptimal algorithm for VC (for example, the maximal matching one) has an approximation ratio bounded above by

$$\frac{n}{\frac{11}{20} n} = \frac{20}{11} < 1.82.$$

Step [1] of $\mathcal{A}'$ serves to treat the cases where $\alpha(G) \leq \frac{9}{20} n$.

Thus, the main part of the proof concerns the case $\alpha(G) \geq \frac{9}{20} n$.

In what follows, we assume the existence of a PTAA $\mathcal{A}$ with approximation ratio ρ' (absolute constant) for SC which provides us with a solution T_i' of cardinality τ_i' for B^i [inductive schema of expression (1)], by means of which we shall derive a solution T' of cardinality τ' for B_G (or equiv. for G).

Let us suppose that a graph G, instance of VC, is given. We apply $\mathcal{A}'$ (algorithm 1) to G and we examine the following two cases corresponding to steps [5] and [6] of algorithm $\mathcal{A}'$, respectively:

(**a**) $\forall i \leq k,\ \tau_i' \geq n \frac{\tau_{i-1}'}{2}$;

(**b**) $\exists i \leq k,\ \tau_i' < n \frac{\tau_{i-1}'}{2}$.

(**a**) $\forall \mathbf{i} \leq \mathbf{k},\ \tau_{\mathbf{i}}' \geq \mathbf{n} \frac{\tau_{\mathbf{i-1}}'}{2}$.

Here, we have to examine two subcases concerning $\alpha(G)$:

(**a1**) $\alpha(G) \geq \frac{11}{20} n$;

(**a2**) $\frac{9}{20} n \leq \alpha(G) \leq \frac{11}{20} n$.

(**a1**) Let $\bar{m}$ be the cardinality of a maximum matching in G. Given that [4] $\alpha(G) + \tau(G) = n$ and $\bar{m} \leq \tau(G)$, we have

$$\bar{m} \leq \tau(G) \leq \frac{9}{20} n. \tag{2}$$

We have already mentioned that given a maximum matching M, the set of vertices incident to the edges of M constitutes a solution Λ for VC of cardinality $\lambda = 2\,\bar{m}$; thus, by using expression (2), we get $\lambda \leq \frac{9}{10}\,n$ and, by taking into account the fact that the exposed vertices of a graph with repect to a maximal matching form an independent set of the graph, we obtain immediately such a set of cardinality $\alpha' \geq \frac{n}{10}$. So, $n = \lambda + \alpha' \geq \lambda + \frac{n}{10}$, or $\lambda \leq \frac{9}{10}\,n$, this expression implying

$$n \geq \frac{10}{9}\,\lambda. \tag{3}$$

In fact, solution Λ is the one obtained during step [1] of algorithm 1; moreover, this solution is compared to the ones obtained in steps [6] and [7] of the algorithm in order to select the minimum among them. So, if the constraint of case (**a1**) holds, then, for the finally selected solution T', $\tau' \leq \lambda$ and, consequently, $n \geq \frac{10}{9}\,\tau'$.

So, using the hypothesis of case (**a**) and expression (3), we conclude that

$$\tau'_k \geq n\,\frac{\tau'_{k-1}}{2} \geq n\,\frac{n\,\frac{\tau'_{k-2}}{2}}{2} \geq \ldots \geq \frac{n^{k-1}}{2^{k-1}}\,\tau'$$

or

$$\tau'_k \geq \left(\frac{5}{9}\right)^{k-1}\,\tau'^{k}. \tag{4}$$

(**a2**) In this case, if α' is the cardinality of the independent set S' obtained from $\mathcal{A}''$ (second part of step [5] of algorithm 1), it verifies $\frac{\alpha'}{\alpha\,(G)} \geq \rho$ or $\alpha' \geq \rho\alpha\,(G) \geq \frac{9}{20}\,\rho\,n$. Then, $T'' = V \backslash S'$ is a vertex covering for G of cardinality $\lambda \leq n - \frac{9}{20}\,\rho\,n = \frac{20 - 9\,\rho}{20}\,n$ or,

$$n \geq \frac{20}{20 - 9\,\rho}\,\lambda. \tag{5}$$

Let τ' be the cardinality of the solution for G found by the application of $\mathcal{A}$ on B^k (first part of step [5] of algorithm $\mathcal{A}'$). For τ', given that $\forall\, i \leq k$, $\tau'_i \geq n\,\frac{\tau'_{i-1}}{2}$, we have

$$\tau'_k \geq n\,\frac{\tau'_{k-1}}{2} \geq n\,\frac{n\,\frac{\tau'_{k-1}}{2}}{2} \geq \ldots \geq \left(\frac{n}{2}\right)^{k-1}\,\tau'$$

and by expression (5),

$$\tau_k' \geq \left(\frac{10\,\lambda}{20 - 9\,\rho}\right)^{k-1} \tau'. \tag{6}$$

If $\lambda \geq \tau'$, then expression (6) gives $\tau_k' \geq \left(\frac{10}{20 - 9\,\rho}\right)^{k-1} \tau'^k$, while if $\lambda < \tau'$, it gives $\tau_k' \geq \left(\frac{10}{20 - 9\,\rho}\right)^{k-1} \lambda^k$.

So, the solution obtained in step [7] of algorithm 1 always verifies[8]

$$\tau_k' \geq \left(\frac{10}{20 - 9\,\rho}\right)^{k-1} \tau'^k. \tag{7}$$

This concludes case (**a**).

(**b**) $\exists\, \mathbf{i} \leq \mathbf{k},\ \tau_{\mathbf{i}}' < \mathbf{n}\,\frac{\tau_{\mathbf{i-1}}'}{2}$

Of course, the inequality $\tau_i' \leq n\,\frac{\tau_{i-1}'}{2}$ imposes in B^i the existence of some S^{i-1}-groups with less than $\frac{\tau_{i-1}'}{2}$ vertices, elements of the solution T_i' (these groups form the set Q).

As we have already seen, in B^i there are $n = |\,S\,|\ S^{i-1}$-groups, each one of these n groups representing a vertex of G when seen with respect to the whole graph B^i. Thus, we have equivalently a partition of the vertices of G into two sets P and Q, the set Q being an independent set of G. The argument: since the S^{i-1}-groups that form Q contain each one less than $\frac{\tau_{i-1}'}{2}$ members of T_i', the existence of a C^{i-1}-groups of B^i (equiv. an edge of G) "seen" in common by two S^{i-1}-groups of Q (let us denote them by $\bar{S}^{i-1}$ and $\tilde{S}^{i-1}$) would lead to a smaller solution τ_{i-1}' (contradicting so the minimality of T_{i-1}' assured by step [5] of algorithm 1); this solution could be obtained by considering the vertices of T_i' belonging to $\tilde{S}^{i-1}$ and $\tilde{S}^{i-1}$, by projecting their indices onto their $i-1$ last coordinates and by considering, finally, the union of these vertices.

[8] Recall that, in algorithm 1, if after the excecution of step [5], step [7] is immediately executed, then in step [7] the minimum between the maximum matching solution of step [1], the solution provided by step [5] and the solution found by the execution of algorithm $\mathcal{A}''$ is selected as final solution for VC; so, always, $\tau' \leq \lambda$.

Let us examine, for a while, set PE and QE.

If QE is empty, then the constraint $\tau_i' < n \frac{\tau_{i-1}'}{2}$ is not true. Really, let us consider the maximum matching M obtained on BG in step [6] of algorithm 1. In terms of the graph B^i, $i = 1, \ldots, k$, one can see M as the set of C^{i-1}-groups (elements if $i = 1$) such that there is no S^{i-1}-group (set if $i = 1$) simultaneously "seeing" two of them; moreover, for each one of the C^{i-1}-groups corresponding to the edges of M, the cardinality of set, covering it in B^i is greater than, or equal to, τ_{i-1}' (recall that in the first part of step [5] of algorithm 1, the minimum of the solutions for the $m\, C^{i-1}$-groups has been retained). So, if p_e is the cardinality of PE, we have $\tau_i' \geq \frac{n - p_e}{2} \tau_{i-1}' + p_e \frac{\tau_{i-1}'}{2} = n \frac{\tau_{i-1}'}{2}$, contradicting so the hypothesis on the size of τ_i'.

On the other hand, if PE is empty ($P = PS$) or M is perfect ($PE = QE = \varnothing$), then the optimal solution for G is found. The arguments: since (i) $PS = P$ is saturated by the matching M[9], (ii) the mates of this set is set QS (iii) P, being the complement of an independent set, *i.e.* $V \backslash Q$[10] is a solution for G and, moreover (iv) in every graph, the cardinality of a vertex covering is greater or equal to the cardinality of a maximum matching [4], then the minimum over all the possible solutions is found.

Thus, we can suppose that the sets PS, PE, QS, QE provided by the execution of step [6] of algorithm 1, are all non-empty.

Of course, the fact that M is a maximum matching implies that there will never be a vertex member of PE added in PS' during the described procedure. In fact, during step [6] of algorithm 1, we proceed by creating sets of alternating paths[11]. If for instance we suppose that, by this construction of alternating paths, we attain member of PE, this means exactly that we have discovered an augmenting path[12] and of course the hypothesis that M is a maximum matching is contradicted. Also, the fact that there are no more vertices that can be added in PS' during step [6] of algorithm 1, implies that all the members of the so-formed QS' are adjacent exclusively to the

[9] It is easy to see that M is also maximal for G.

[10] Recall that $V = S$ (step [2] of algorithm 1).

[11] Given a matching M in a graph G, an alternating path is a simple (elementary) path $P = v_{i_1} - v_{i_2} - \ldots - v_{i_k}$, where an edge in $M \cap P$ alternates an edge of $(E \backslash M) \cap P$.

[12] An augmenting path is an alternating path where the vertices v_{i_1} and v_{i_k} are exposed with respect to M; a matching M is maximum if and only if it does not contain augmenting paths.

members of PS' formed throughout the procedure. At the end of step [6] of algorithm 1, we have a partition of the vertices of P into two sets, namely PS' and $P' = P \backslash PS'$.

Figure 5 shows an example of how step [6] of algorithm 1 works. Set QE is considered first and, next, the set of the neighbours of QE (the first rectangle marked PS'); after, the mates of the vertices of PS' (first circle marked QS') are considered; all of the neighbours of these new vertices are then entered to PS' (if they do not belong already); these newly introduced vertices are in the second rectangle marked PS' and so on; this procedure will go on until the vertices lastly introduced to QS' have all of their vertices already in PS' (this is the case of the rightest circle marked QS'). With respect to Figure 5, let us suppose that one of the neighbours of the lowest vertex of the second circle marked QS' belongs to PE. Moreover, let us denote by v_{i_1} the lowest vertex of the circle marked QE, by v_{i_2} the lowest vertex of the first rectangle marked PS', by v_{i_5} the lowest vertex of the first circle marked QS', by v_{i_4} the lowest vertex of the second rectangle marked PS', by v_{i_5} the lowest vertex of the second circle marked QS' and, finally, by v_{i_6} the hypothetical neighbour of v_{i_5} belonging to PE (this vertex, as well as the edge $v_{i_5} v_{i_6}$, are not shown in *Fig.* 5). Then, it is easy to see that the path $v_{i_1} - v_{i_2} - v_{i_3} - v_{i_4} - v_{i_5} - v_{i_6}$ is an augmenting path and, in this case, we could obtain a greater matching by replacing the set $\{v_{i_2} v_{i_3} v_{i_4} v_{i_5}\}$ of matched edges on this path by the set $\{v_{i_1} v_{i_2} v_{i_3} v_{i_4} v_{i_5} v_{i_6}\}$ (considering the latter set as the set of matched edges along this path).

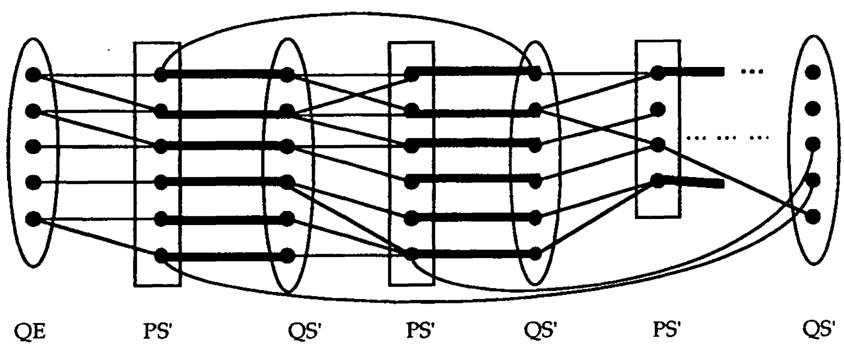

Figure 5. – An example of how step [6] of algorithm $\mathcal{A}'$ works.

We claim that PS' is an optimal solution of VC in G' (created by algorithm 1). Clearly, PS' is a solution for G', since its members are adjacent to all other vertices ($QS' \cup QE$) of G'. Moreover, this solution is optimal for G'. The arguments: the way we have constructed PS' implies that all the members of this set are endpoints of edges contained in M (this subset of M constitutes, obviously, a maximal matching for G'); furthermore, all of the other edges emanate from those vertices; finally, the edges of the set $(E(G) \backslash E(BG)) \cap E(G')$ (where by $E(G)$, $E(G')$, $E(BG)$, we denote the edge set of G, G' and BG, respectively) removed from G to obtain BG are all incident to members of $P \cap PS' = PS'$. Thus, the cardinality of PS' is exactly the cardinality of a matching in G' and thus the solution induced by PS' is minimum [4].

Also, by the way we have conceived step [6] of $\mathcal{A}'$, there are no edges between the members of $QS' \cup QE$ and the vertices of the graph G'' (where all of the vertices of the set $Q \backslash (QS' \cup QE)$ are saturated by M).

By referring to Figure 5, one can see that the union of the vertices of all of the rectangles marked PS' (that is the set PS' finally produced by algorithm 1) cover all of the edges of the graph induced by the vertex-set $QE \cup QS' \cup PS'$.

Finally (the multiple executions of), algorithm $\mathcal{A}'$ produces a partition of G say $G'_1, G'_2, \ldots, G'_l$ such that G'_k, $k < l$, are polynomially solved and G'_l is either polynomially solved or admits the constraint $\tau'_i \geq n \frac{\tau'_{i-1}}{2}$ where, now, τ'_i, n and τ'_{i-1} concern G'_l, for which case (**a**) is applicable. Let us denote by G' the union $\bigcup_{1 \leq k \leq l-1} G'_k$ of the graphs produced by the (eventually multiple) execution of step [6] and by G'' the graph G'_l.

For this partition of G into the graphs G' and G'', we can prove that *the approximation ratio ρ of a polynomial algorithm solving approximately the VC in G is smaller than the approximation ratio ρ_2 of a polynomial algorithm solving approximately the VC in G''.*

Really, let us consider the independent set Q_1 associated with the solution T'_1 ($|T'_1| = \tau'_1$) of G'. We denote by T_1 ($|T_1| = \tau_1$) the quantity $\tau(G'')$, *i.e.* the optimal solution for G'. We have already proved that $T'_1 = T_1$ ($\tau'_1 = \tau_1$).

Let T'_2 ($|T'_2| = \tau'_2$) and T_2 ($|T_2| = \tau_2$) be the approximate and optimal solutions, respectively, for G'' ($\tau_2 = \tau(G'')$) and let $\frac{\tau'_2}{\tau_2} \leq \rho_2$ for a fixed constant ρ_2.

In fact, Q_1 is the set QS' formed during step [6] of algorithm 1; moreover, since the construction of graph G' stops when all neighbours of QS' are already in set PS', in graph G' induced by $PS' \cup QS' \cup QE$ all of the neighbours of the members of QS' are included in G' and, consequently, all edges between G' and G'' are edges outcoming from PS'. Furthermore, PS' being a complement of an independent set (the set $QS' \cup QE$), it is a vertex covering for G' and, since its size equals the size of a maximal matching of this graph, PS' constitutes a minimum size vertex covering; moreover, PS' covers all of the edges between G' and G'' (we notice, once more, that algorithm 1 constructs polynomially PS'; so, $|PS'| = |T_1'| = |T_1|$). Consequently, once the edges of G' and the ones between G' and G'' are covered, in order to all of the edges of G do so, the edges of G'' remain to be covered. One can do that by calling the approximation algorithm anounced by the emphasized proposition to obtain the solution T_2'; since there are no edges between members of Q_1 and vertices of G'', there are no more edges between Q_1 and the independent set Q^2 associated with T_2'; thus, the set $T' = T_1' \cup T_2'$ covers all of the edges of G, constituting so a solution for G (it is exactly the candidate solution of case (ii) in step [7] of algorithm 1).

For the optimal solutions on G' and G'', respectively, T_1 optimally covers the edges of G', as well as the edges between G' and G''; on the other hand, T_2 optimally covers the edges of G''. These sets (T_1 and T_2) being disjoint, we have $\tau(G) = \tau_1 + \tau_2$.

Thus, given that $\frac{\tau_1'}{\tau_1} = 1$ and $\frac{\tau_2'}{\tau_2} \leq \rho_2$, we get

$$\rho = \frac{\tau'}{\tau(G)} = \frac{\tau_1' + \tau_2'}{\tau_1 + \tau_2} = \frac{\tau_1 + \tau_2'}{\tau_1 + \tau_2} \leq \rho_2. \tag{8}$$

This completes the proof of the emphasized proposition.

The last line of step [6] of algorithm 1 implies the application of steps [1] ÷ [6] of the algorithm on G''.

It remains now to explore the approximation ratio for VC induced by solutions for SC found after the kth composition of G'' (step [5] of algorithm 1). In any case [see expressions (4) and (7)], the cardinalities of the solutions obtained in this step are of the form

$$\tau_k' \geq \beta^{k-1} \tau'^k \tag{9}$$

where $1 \geq \beta \geq \frac{1}{2}$ and equal either to $\frac{5}{9}$ [expressions (4)] or to $\frac{10}{20 - 9\rho}$ [expression (7)].

Moreover, for the optimal solutions $\tau(G)$, τ_k of G and B^k, respectively, we have

$$(\tau(G))^k \geq \tau_k. \tag{10}$$

From expressions (9), (10), and the fact that the approximation algorithm $\mathcal{A}$ for SC has approximation ratio ρ', we have

$$\rho' \geq \frac{\tau'_k}{\tau_k} \geq \beta^{k-1} \left(\frac{\tau'}{\tau(G)}\right)^k$$

or

$$\frac{\tau'}{\tau(G)} \leq \frac{1}{\beta^{\frac{k-1}{k}}} \rho'^{\frac{1}{k}}.$$

We have already seen that if the composition of algorithm $\mathcal{A}'$ is performed on G'', then the solution for G obtained in step [7] approximates the optimal one within an error smaller than the one for the solution of G'' [expression (8)].

So, we get (recall that in step [2] of algorithm 1, we have fixed $\rho'^{\frac{1}{k}} \leq 1+\varepsilon$)

$$\frac{\tau'}{\tau(G)} \leq (1+\varepsilon) \max \left\{\frac{20-9\rho}{10}, \frac{9}{5}\right\}$$

or

$$\frac{\tau'}{\tau(G)} \leq \max \left\{\frac{9}{5}, 2-\frac{9}{10}\rho\right\} + \varepsilon$$

for $\varepsilon \leq \varepsilon \max \left\{\frac{9}{5}, \frac{20-9\rho}{10}\right\}$.

Henceforth, since we can choose ε arbitrarily small, the approximation ratio of algorithm 1 tends to $\max \left\{\frac{9}{5}, 2-\frac{9}{10}\rho\right\} < 2$.

3. DISCUSSION

The result of section 2 has brought to the fore an aspect of the complex relation, concerning their approximation behaviour, between three known and difficult combinatorial optimization problems. We think that such results in a theoretical level contribute to produce a deeper knowledge of the approximation mechanisms in the class NP-complete. On the other hand, they could help us in deeper understanding of the properties of this class as well as of the relations between its problems, relations that are not exhausted in the fact that the existence of an exact polynomial algorithm for one of them would imply the existence of such an algorithm for all of the problems. Moreover, the investigation of this type of relation, from a "practical" point of view, could produce immediate positive or negative results for some of the problems concerned. If for example, the conditions of the theorem concerning IS and SC were true, a new improved algorithm for VC would be immediately found.

Unfortunately, this "practical" significance of the above result is not valid. In fact, in [8] (*see* also [7]), Lund and Yannakakis have proved a strong negative result for SC approximability: SC cannot be approximated with ratio $c \log m$ for any $c < \frac{1}{4}$ unless NP $\subseteq$ DTIME $[n^{\text{poly} \log n}]$ (conjecture weaker than P $=$ NP but highly improbable). On the other hand, the approximability of IS in the class $\mathcal{G}$, even if such a result has not be proved yet, is very improbable[13]. For one more time, in theoretical computer science it is very frequent, we have produced theoretical results, we have eventually increased the number of open questions, without, unfortunately, increasing the number of the answers.

ACKNOWLEDGEMENTS

I want to thank Lalo Fernandez de la Vega for his helpful comments and our fruitful discussions on this work. I thank also Rajeev Motwani for his very carefull reading of an early version of the paper and for our discussions-by-e_mail. Many thanks to Virginie Gabrel for having helped me in drawing the figures of the paper. Finally, I am gratefull to an anonymous referee for pertinent suggestions and remarks contributing to improve this paper.

REFERENCES

1. S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof Verification and Intractability of Approximation Problems*, Proc. IEEE/FOCS, 1992, pp. 14-23.
2. R. Bar-Yehuda, S. Even, A Linear Time Approximation Algorithm for the Weighted Vertex Cover Problem, *J. of Algorithms*, 1981, *2*, pp. 198-203.
3. R. Bar-Yehuda, S. Even, A Local-Ratio Theorem for Approximating the Weighted Vertex Cover Problem, *Annals of Discr. Appl. Maths*, 1985, *25*, pp. 27-46.
4. C. Berge, *Graphs and Hypergraphs*, North Holland, Amsterdam, 1973.
5. M. R. Garey, D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.
6. F. Gavril, cited in [5], p. 134.
7. D. S. Johnson, The NP-Completeness Column: On Ongoing Guide, *J. of Algorithms*, 1992, *13*, pp. 502-524.

[13] In [1], the authors prove that there is no constant ratio approximation algorithm for IS unless P $=$ NP.

8. C. LUND, M. YANNAKAKIS, *On the Hardness of Approximating Minimization Problems*, Proc. ACM/STOC, 1993, pp. 286-293.
9. B. MONIEN, E. SPECKENMEYER, Ramsey Numbers and an Approximation Algorithm for the Vertex Cover Problem, *Acta Informatica*, 1985, *22*, pp. 115-123.
10. C. H. PAPADIMITRIOU, K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity, Prentice Hall*, New Jersey, 1981.