RAIRO
INFORMATIQUE THÉORIQUE

A. AIELLO
E. BURATTINI
A. MASSAROTTI
F. VENTRIGLIA

Towards a general principle of evaluation for approximate algorithms

RAIRO – Informatique théorique, tome 13, nº 3 (1979), p. 227-239.
<http://www.numdam.org/item?id=ITA_1979__13_3_227_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 13, n° 3, 1979, p. 227 à 239)

TOWARDS A GENERAL PRINCIPLE OF EVALUATION FOR APPROXIMATE ALGORITHMS (*)

by A. AIELLO, E. BURATTINI, A. MASSAROTTI and F. VENTRIGLIA ([1])

Communiqué par J. F. PERROT

Abstract. — *The concept of reducibility, the choosing of measures of approximate solutions and the evaluating functions for approximate algorithms are investigated in the present paper. The analysis carried out there shows that an evaluating function, which refers to the worst solution too, can be introduced. The adoption of such function is more advisable under many respects.*

Résumé. — *Le but de cet article est l'étude de la conception de la réductibilité, le choix des mesures de solutions approchées et les fonctions pour évaluer les algorithmes d'approximation. L'analyse montre qu'une fonction d'évaluation, qui considère aussi la solution pire, peut être introduite. Cette fonction élimine les contradictions qui étaient engendrées par les autres fonctions.*

1. INTRODUCTION

There is a large class of combinatorial problems involving the determination of properties of graphs, integers, finite families of finite sets, boolean expressions, etc. By now it is widely accepted, as a reasonable working hypothesis, that such problems can be all considered intractable since there exist no polynomial-bounded algorithms to solve them. Karp [5] has shown that some of them are strongly related so that if there exists a polynomial-bounded algorithm for one problem the same occurs for every other problem in the class, which is called NP-Complete.

The previous result of Karp has been obtained by formulating the problems as language-recognition ones. The concept of reducibility among languages plays, then, a fundamental rôle in this kind of approach.

(*) Reçu juin 1978, révisé mars 1979.

([1]) Laboratorio di Cibernetica C.N.R., Arco Felice (NA), Italia.

Many NP-Complete problems can be considered such as optimization problems. According to Johnson [4], an optimization problem $\mathscr{P}$ consists of:

(1) a set $\mathrm{INPUT}_{\mathscr{P}}$ of possible inputs;

(2) a map $\mathrm{SOL}_{\mathscr{P}}$ mapping each $u \in \mathrm{INPUT}_{\mathscr{P}}$ to a finite set of approximate solutions;

(3) a measure $m_{\mathscr{P}}: \bigcup_{u \in \mathrm{INPUT}_{\mathscr{P}}} \mathrm{SOL}_{\mathscr{P}}(u) \to Q^+$ defined for all possible approximate solutions;

(4) the assignement of the value MAX or MIN to the function BEST depending on whether $\mathscr{P}$ is a maximization or minimization problem.

From the firm belief in the intractability of NP-Complete problems comes the attempt of searching fast heuristic algorithms for obtaining "good" approximate solutions.

Evaluating the performance of approximation algorithms, that is defining quantitatively how much "good" their solutions are, is the first task of an approach of this kind.

D. S. Johnson [4] and S. K. Sahni *et al.* [6] have proposed to evaluate the approximation algorithms by analysing their worst case behaviors and have both introduced evaluating functions substantially equivalent each other.

The main result of those studies, besides setting up several fast approximate algorithms able to yield almost optimal solutions, has been the indication of a tentative classification of the NP-Complete problems using just the approximation algorithms as classification tools. In particular, problems can be quoted for which almost optimal solutions have been found by means of linear algorithms and, on the contrary, it has been prooved that the ascertainment of the existence of ε-approximate algorithms for some problems is itself a NP-Complete problem.

Remaining in the setting of the heuristic approach to the study of combinatorial problems we have tried to investigate the concept of reducibility, the choosing of measures of approximate solutions given for optimization problems and the evaluating function of approximate algorithms as well as the correlations which may exist among such elements.

The results show that:

(1) The existence of reducibility between two combinatorial problems when they are represented in form of language recognition does not necessarily imply that one of them is reducible to the other also when both are represented in the form of optimization problems. Nevertheless, the fact that two combinatorial problems can be represented as optimization problems, together with the existence of one, but not necessarily only one, polynomial-bounded reduction

function, are indispensable conditions in order we could just consider approximate algorithms and reducibility between approximation problems;

(2) the evaluation function of approximation algorithms, which also is arbitrary, is bound by invariance conditions induced by available reductions;

(3) certain results concerning the behavior of approximation algorithms pertaining to some optimization problems must be revised, as well as it is necessary to revise certain conclusions relating to the classification of *NP*-Complete problems. The work reported here was first presented at an I.R.I.A. Seminar in December 1976. Further papers on the same subject include [1, 2, 3].

2. REDUCIBILITY AND APPROXIMATION ALGORITHMS

In the paper of Karp quoted before, among other things, it is reported under the title « Main Theorem » a list of *NP*-Complete problems from which it follows that:

(*a*) *K*-GRAPH-COLORING $\propto$ EXACT-COVER;

(*b*) EXACT-COVER $\propto$ KNAPSACK,

where the symbol $\propto$ stands for "is reducible to".

Each of the three problems above can be formulated as an optimization problem [4]:

KNAPSACK:

$\text{INPUT}_{\mathscr{K}} = \{ \langle T, s, b \rangle$: T is a finite set, s: $T \to Q^+$ is a map which assigns to each $x \in T$ a "size" $s(x)$, and $b > 0$ is a single rational number $\}$,

$$\text{SOL}_{\mathscr{K}}(\langle T, s, b \rangle) = \{ T' \subseteq T : \sum_{x \in T'} s(x) \leqq b \},$$

$$m_{\mathscr{K}}(T') = \sum_{x \in T'} s(x);$$

EXACT-COVER:

$$\text{INPUT}_{\mathscr{E}\mathscr{C}} = \{ F : F \text{ is a finite family } \{ S_1, S_2, \ldots, S_p \} \text{ of finite sets} \},$$

$$\text{SOL}_{\mathscr{E}\mathscr{C}}(F) = \{ F' \subseteq F : \bigcup_{S \in F'} S = \bigcup_{S \in F} S \},$$

$$m_{\mathscr{E}\mathscr{C}}(F') = \sum_{S \in F'} |S|;$$

GRAPH-COLORING:

$\text{INPUT}_{\mathscr{GC}}=\{G(N, A)$: G is a finite undirected graph with nodes N and arcs $A\}$,

$\text{SOL}_{\mathscr{GC}}(G)=\{h:N\rightarrow\{1, 2, \ldots, |N|\}$: if there is an arc between nodes x and y, $h(x)\neq h(y)\}$,

$$m_{\mathscr{GC}}(h)=|\{z:z=h(x), \text{ for some } x\in N\}|.$$

At first sight it could seem that it is possible to transfer the results given by some approximation algorithm of KNAPSACK to the solution of the general problem of EXACT-COVER and then, through the transitive property of reducibility among NP-Complete problems, from KNAPSACK to GRAPH-COLORING. But this is not the case since the reducibility among combinatorial problems in the form of language recognition does not change immediately into reducibility among optimization problems.

In fact, considering the reduction function between EXACT-COVER and KNAPSACK, let us remark that some optimal solution of the former $(F'\subseteqq F : \bigcup_{S\in F'} S=\bigcup_{S\in F} S$ and $\sum_{S\in F}|S|=|\bigcup_{S\in F} S|)$ does always correspond to every given optimal solution of the latter $(T'\subseteqq T: \sum_{x\in T'} s(x)=b)$, but the same reduction function does not put any solution of the EXACT-COVER $(\nexists F'\subseteqq F: \bigcup_{S\in F'} S=\bigcup_{S\in F} S)$ into correspondence with the approximate solutions of KNAPSACK $(T'\subseteqq T: \sum_{x\in T'} s(x)<b)$.

Analogous considerations can be done for the reduction between K-GRAPH-COLORING and EXACT-COVER.

Now we want to show how some properties coming from the reducibility, in particular the equivalence among NP-Complete problems formulated in terms of optimization problems, limit the choice of the evaluating function of the approximate algorithms. For these reasons, we shall then refer to a specific problem and its possible schematizations.

3. CONCRETE PROBLEMS AS LANDMARKS OF THE THEORY

Consider a multiprogramming operating system and suppose that, at a certain moment, there are n programs allocated in a HSM (high speed memory) starting at the addresses $x_1, \ldots, x_n$ with lengths $l_1, \ldots, l_n$. For a correct operating, at each instant, one has to fulfill the conditions

$$\forall i, j, \quad \left\{ \begin{array}{ll} \sum_{k=i}^{j-1} l_k \leqq (x_j - x_i) & \text{if} \quad j > i \\ \sum_{k=j}^{i-1} l_k \leqq (x_i - x_j) & \text{if} \quad i > j \end{array} \right\} \tag{1}$$

and the hypothesis

$$\sum_{i=1}^{n} l_i \leqq \text{available space in HSM.} \tag{1'}$$

If during the operations it happens that some programs change their memory-space requirements, one has to take care to move certain programs in order that condition (1) be always satisfied.

Since relocating one or more programs costs in terms of time and space, assuming the cost of every relocation equal to a costant, the problem arises of determining either which is the smallest number of programs to be reallocated for evoiding overlappings or, equivalentely, which is the largest number of programs to be not reallocated.

Given the n initial addresses, to each assignment of lengths corresponds a set of solutions of the problem, a subset of which constitutes the set of the optimal solutions. Consider now the set T of the non-ordered pairs $\{i, j\}$ of programs transgressing condition (1) and let F be the family of the finite sets $\{S_1, \ldots, S_n\}$ where

$$S_j = \{ p \in T \mid j \in p \},$$

that is, the set of the pairs containing j and some program located too close to j.

Every subset $I \subseteqq \{1, \ldots, n\}$ such that

$$\bigcup_{h \in I} S_h = T,$$

determines a possible correct reallocation: that concerning only the programs belonging to I. So, being $T = \bigcup_{S_j \in F} S_j$, the considered problem is reduced to the well-known SET-COVERING and its optimal solution is given by the MINIMUM SET-COVERING of the family F.

Another way to formulate the same problem is obtainable generating the undirected graph $G(N, A)$ in which the i-th node represents the i-th program and there is an arc between nodes i and j if and only if the corresponding programs do satisfy to the condition (1). According to this formulation every completely connected subgraph $G^*(N^*, A^*)$ of $G(N, A)$ singles out a set of mutually consistent programs, those corresponding to N^*. In other words every MAXIMUM CLIQUE of the graph $G(N, A)$ yields one of the largest available sets of programs which do not need to be reallocated.

Moreover, consider the graph $R(N, B)$ in which the i-th node represents the i-th program and there is an arc between the nodes i and j if and only if the relative programs do not satisfy to the condition (1). In this case a subset $\overline{N}^* \subseteq N$ such that every arc of B has, at least, one node belonging to $\overline{N}^*$ represents a subset of programs whose reallocation can restore consistence among all the programs. So that the smallest set of programs which need to be reallocated is given by the MINIMUM NODE-COVER of the graph $R(N, B)$.

Now given n programs to be stored in HSM, let their addresses be $x_1, \ldots, x_n$ and $l_1, \ldots, l_n$ their memory-space requirements (lengths), that is an assigned input I_{PR} of the PROGRAMS REALLOCATING problem, let:

$\hat{\mathscr{P}}$ be a generic corresponding solution, that is a reallocation operation in HSM;

$\mathscr{P}^*$ be an optimal solution, that is the less "expensive" among all the solutions $\hat{\mathscr{P}}$;

$\hat{\mathscr{P}}_{\mathscr{S}\mathscr{C}}$ and $\mathscr{P}^*_{\mathscr{S}\mathscr{C}}$ be respectively a generic solution and an optimal solution of the SET-COVERING problem related to the given PROGRAMS REALLOCATING problem and to the input I_{SC} corresponding to I_{PR};

$\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}$ and $\mathscr{P}^*_{\mathscr{N}\mathscr{C}}$ the analogous for the related NODE-COVER problem, and $\hat{\mathscr{P}}_{\mathscr{C}}$ and $\mathscr{P}^*_{\mathscr{C}}$ the analogous for the related CLIQUE problem.

It can be easely seen that:

$$|\mathscr{P}^*_{\mathscr{S}\mathscr{C}}| = |\mathscr{P}^*_{\mathscr{N}\mathscr{C}}| = n - |\mathscr{P}^*_{\mathscr{C}}|, \tag{2}$$

and that for every approximate $\hat{\mathscr{P}}$ there exist corresponding $\hat{\mathscr{P}}_{\mathscr{S}\mathscr{C}}$, $\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}$ and $\hat{\mathscr{P}}_{\mathscr{C}}$ such that

$$|\hat{\mathscr{P}}_{\mathscr{S}\mathscr{C}}| = |\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}| = n - |\hat{\mathscr{P}}_{\mathscr{C}}| \tag{3}$$

So for each input I_{PR} of PROGRAMS REALLOCATING problem we can point out the following relation among the approximate and optimal solutions of related problems relative to inputs all corresponding to the same input I_{PR}:

$$|\hat{\mathscr{P}}_{\mathscr{S}\mathscr{C}}| - |\mathscr{P}^*_{\mathscr{S}\mathscr{C}}| = |\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}| - |\mathscr{P}^*_{\mathscr{N}\mathscr{C}}| = |\mathscr{P}^*_{\mathscr{C}}| - |\hat{\mathscr{P}}_{\mathscr{C}}|. \tag{4}$$

Let us suppose there exists a heuristic algorithm A which approximates the optimal solutions of the NODE-COVER problem in a rather "good" way. If the NODE-COVER problem is related to a PROGRAM REALLOCATING problem, to each solution $\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}$ of $\mathscr{A}$ corresponds a solution $\hat{\mathscr{P}}$ of the reallocating problem.

Consider now the algorithm $\overline{\mathscr{A}}$ obtained adding to $\mathscr{A}$ the following conclusive step:

$$- \text{ compute the complementation } \{1, \ldots, n\} - \hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}} \tag{5}$$

since the graph $G(N, A)$ is the complement of the graph $R(N, B)$, each complementation to $\{1, \ldots, n\}$ of a NODE-COVER of the graph $R(N, B)$ is a CLIQUE of the graph $G(N, A)$. So, the algorithm $\overline{\mathscr{A}}$ really gives the solution $\hat{\mathscr{P}}_{\mathscr{C}}$, of the CLIQUE related to the PROGRAMS REALLOCATING problem. Since both $\hat{\mathscr{P}}_{\mathscr{C}}$ and $\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}$ imply the same concrete solution $\hat{\mathscr{P}}$ of the considered concrete problem (the former gives the programs which do not need reallocation, the latter those which do need reallocation), all the solution-measures should agree with the fact that $\hat{\mathscr{P}}, \hat{\mathscr{P}}_{\mathscr{C}}$ and $\hat{\mathscr{P}}_{\mathscr{N}\mathscr{C}}$ must be considered equally "good" as well as every evaluation function of approximate algorithms should consider both $\mathscr{A}$ and $\overline{\mathscr{A}}$ equally "good" approximate algorithms. From this consideration it follows that, according to equality (4), the absolute difference between the cardinality of an approximate solution and the cardinality of the optimal solution relative to the same input, represents an absolute measure for approximate solutions of CLIQUE and NODE-COVER which is quite satisfactory as regards invariance of « goodness" of solutions of the PROGRAMS REALLOCATING problem under changes of the solving process which they come out from.

4. INVARIANCE OF THE EVALUATING FUNCTIONS AND DUALITY

Besides any references to concrete problems, we can more generally say that NODE-COVER and CLIQUE are really dual aspects of the same combinatorial problem. This duality arises just from the fact that the same information contained in an undirected unweighted graph can be likewise contained in the complemented graph, apart from a simple inversion into the code which assigns a given meaning to the existence of an arc between two nodes and the opposite meaning to its non-existence. Therefore, we can assert that if there is a "good" approximate algorithm for NODE-COVER there must be a "good" one for CLIQUE and vice versa.

Moreover an evaluating function cannot be quite arbitrary but it must be invariant under complementation of graphs.

Consider now two problems whose statements share the following beginning part:

⟨⟨ Given the undirected graph $G(N, A)$ and the weighting function

$W: A \to Z^+$, find k disjoint subsets $S_1, \ldots, S_k$ of N such that:

$$\bigcup_{i=1}^{k} S_i = N$$

and...⟩⟩

One of them, that called K-MINIMUM-CLUSTER, concludes adding the following request:
⟨⟨... such to minimize

$$m_{KC} = \sum_{i=1}^{k} \sum_{\substack{\{u, v\} \in A \\ u, v \in S_i}} w(\{u, v\}) \rangle\rangle.$$

The other, called K-MAX-CUT, concludes as following:
⟨⟨...such to maximize

$$m_{KMC} = \sum_{\substack{\{u, v\} \in A \\ u \in S_i \\ v \in S_j \\ i<j}} w(\{u, v\}) \rangle\rangle.$$

In other words it is possible to divide the set N in k parts either minimizing the summation of the weights inside k subsets (K-MINIMUM-CLUSTER), or maximizing the summation of the weights outside k subsets (K-MAXIMUM-CUT).

But, since

$$m_{KC} + m_{KMC} = \text{Const.} \tag{6}$$

where

$$\text{Const.} = \sum_{\{u, v\} \in A} w(\{u, v\})$$

is the summation of all the weights, it is beyond doubt that the K-MINIMUM-CLUSTER and K-MAXIMUM-CUT are dual formalizations of just the same problem. So it is quite contradictory to think that there could exist a "good" approximate algorithm for the former which is "not good" for the latter.

Since duality can more generally come out whenever two optimization problems concern respectively minimization and maximization of bound quantities, an evaluating function of approximate algorithms must be at least invariant under alternation of two optimizing quantities which linearly depend each other. Formally, given two dual optimization problems (representation of the same real problem) Q_1 and Q_2, let $q_1(u)$ and $q_2(u)$ be the parameters to be optimized relative to the same input u and let $q_1^*(u)$ and $q_2^*(u)$ be the corresponding optimal values.

Let a linear dependence be stated between q_1 and q_2:

$$q_2(u) = \alpha q_1(u) + C,$$

where C is a constant.

An evaluating function $\mathscr{F}$ must satisfy the following condition of invariance under linear transformation:

$$\mathscr{F}(q_1(u)) = \mathscr{F}(\alpha q_1(u) + C),$$

while both $\mathscr{F}(q_1^*(u))$ and $\mathscr{F}(q_2^*(u)$, referring to optimal solutions, must assume the minimum value of $\mathscr{F}$ (marginally, this implies that correctly defined evaluating functions should always be lowerly bounded functions of the input u).

6. RELATIVITY AND INVARIANCE OF THE EVALUATING FUNCTION

In the optimization problems, an absolute measure of "goodness" of approximate solutions can be given just by the absolute difference between the value $\hat{q}$ obtained for the parameter which had to be optimized and the optimal value q^* obtainable for the same input. We already hinted at this possibility at the end of section (3). Since absolute measures allow only to compare "goodness" of solutions relating to the same problems and to the same input, it is necessary to introduce "relative" measures for achieving more general comparison. But, to save invariance, reference values must be chosen with caution.

To illustrate certain contradictions into which one can fall when invariance is neglected, we have to analyze some results given in the literature. For example, Sahni [6] has proposed the following relative measure of approximate solutions:

$$\frac{|q^* - \hat{q}|}{q^*} \tag{7}$$

trying to evaluate approximate algorithms by looking for absolute upper bounds of the function (7). In this way he found a fast approximate algorithm (the computing time is proportional to the "size" input) for the K-MAX-CUT and he demonstrated that it was an $1/k$-approximate algorithm, that is, according to the formalism introduced in section (4), the relation

$$\frac{|m_{KMC}^* - \hat{m}_{KMC}|}{m_{KMC}^*} < \frac{1}{k} \tag{8}$$

is always satisfied, for all inputs.

But, from (6), putting C in the place of "constant", we have

$$m_{LMC}=C-m_{KC}$$

and, from (8), by simple substitutions and manipulations, we get for the K-MINIMUM-CLUSTER the following evaluation:

$$\frac{|m_{KC}^{*}-\hat{m}_{KC}|}{m_{KC}^{*}}<\frac{1}{k}\,\frac{C-m_{KC}^{*}}{m_{KC}^{*}}. \tag{9}$$

So the same partition of the set N of nodes has two different worst case evaluation according to whether it is considered as a solution of K-MAX-CUT or as a solution of K-MINIMUM-CLUSTER in spite of the fact that, as it was shown in section (4), K-MAX-CUT and K-MINIMUM-CLUSTER are really dual aspects of the same problem. It is easy to deduce that this contradiction comes out just from the non-invariance of the relative measure (7) under linear transformations depending on the inadequacy of the choice of the reference value (for which has been taken the optimal value q^*).

From a merely formal point of view it is evident that the invariance of (7) under general linear transformation can be saved only by introducing into the denominator a second term T transforming just like the other involved parameters. In practice it needs to look for a suitable second reference value. We propose to assume T as given by the value that the parameter to optimize assumes in correspondence of the trivial solution of the optimization problem for the given input. Here for "trivial" solutions we mean that approximate solutions which are obtainable just without any computation. For example, the trivial solution of SET-COVERING is given by all the sets of the assigned family, that of NODE-COVER is the set of nodes N, that of K-MINIMUM-CLUSTER and K-MAX-CUT is the partition of N in which one set is equal to N and all the other are empty sets, that of GRAPH-COLORING is the number of the nodes, that of the KNAPSACK is the empty set, etc.

Moreover, besides the necessity of saving invariance we think that a good relative measure must refer the absolute measure $|\hat{q}-q^*|$ of an approximate solution to that $|T-q^*|$ of the trivial solutions. In other words we formally propose to adopt the following relative measure of approximate solutions

$$r=\left|\frac{\hat{q}-q^*}{T-q^*}\right|, \tag{10}$$

which assumes all the values into the interval [0, 1]. 0 and 1 corresponding, respectively, to the optimal solution and to the trivial solution.

As an example of application we want briefly to show what happens for K-MAX-CUT and K-MINIMUM-CLUSTER when the measure (10) is used in place of (7). We have in correspondence of the trivial solutions, respectively, the following values of the optimizing parameters:

$$T_{KMC}=0, \tag{11}$$

$$T_{KC}=C. \tag{12}$$

In fact, both for a trivial solution of K-MAX-CUT and for one of K-MINIMUM-CLUSTER (which are bijectively mapped one into the other by the identity) all the nodes are put in only one set, so that (11) and (12) come out respectively from the two perfectly equivalent implied consequences:

– there is not any arc between two different sets in a trivial solution of K-MAX-CUT;

– all the arcs do connect nodes belonging to the same set in a trivial solution of K-MINIMUM-CLUSTER.

It follows that for K-MAX-CUT:

$$r_{KMC}=\left|\frac{\hat{m}_{KMC}-m^*_{KMC}}{T_{KMC}-m^*_{KMC}}\right|,$$

which for (11) reduces itself to that given by (7) and consequentely (8) holds again. For what concerns K-MINIMUM-CLUSTER we obtain from (10) and (12) the following relative measure of solutions

$$r_{KC}=\left|\frac{\hat{m}_{KC}-m^*_{KC}}{C-m^*_{KC}}\right|.$$

But, from (6), being

$$m_{KC}=C-m_{KMC},$$

by substitution we obtain the equality

$$r_{KC}=\left|\frac{\hat{m}_{KC}-m^*_{KC}}{C-m^*_{KC}}\right|=\left|\frac{\hat{m}_{KMC}-m^*_{KMC}}{m^*_{KMC}}\right|=r_{KMC}.$$

That is, for a given partition of nodes there is actually one and only one relative measure both when it is considered as a solution of K-MAX-CUT and when it is considered as a solution of K-MINIMUM-CLUSTER. So it follows that every $1/k$-approximate algorithm solving the former must came out to be a $1/k$-approximate algorithm for the latter and vice versa.

7. RELATED RESULTS

In this final section we want to conclude the present paper by giving a brief account of some other results which can be obtained adopting the relative measure (10).

SET-COVERING. For every input size there exists an input in which the set is $U_k = \{u_1, \ldots, u_{2k}\}$ and the given family of sets $F = \{S_1, \ldots, S_k, S_{k+1}, \ldots, S_{3k/2}\}$ is such that:

$$S_i = \{u_{k+2i-1}, u_{k+2i}\} \quad \text{for} \quad i = 1, \ldots, k/2,$$

$$S_i = \{u_{i-k/2}, u_{i+k/2}\} \quad \text{for} \quad i = k/2+1, \ldots, 3k/2.$$

For this kind of input the optimal solution is evidentely given by the family $F' = \{S_{k/2+1}, \ldots, S_{3k/2}\}$ and the optimizating parameter (the cardinality of the solution) assumes its optimal value

$$m^*_{SC} = k.$$

An algorithm like that proposed in [4], founded on the principle of "priority of the largest set", admits as approximate solution the trivial solution $F = \{S_1, \ldots, S_{3k/2}\}$. So in the worst case

$$\hat{m}_{SC} = T_{SC} = 3k/2.$$

From this we can deduce that the worst case relative measure given by (10) is always well-defined and equal to 1. It means that the "priority of the largest set" principle does not guarantee better solutions than those obtainable by trivial algorithms, from a worst case analysis point of view.

CLIQUE. No satisfactory approximate algorithm for CLIQUE is presently known. Since CLIQUE is dual of NODE-COVER and NODE-COVER is reducible to SET-COVERING also when they are considered optimization problems, the previous fact seemed in contrast with the alleged existence of a "good" algorithm for SET-COVERING. The contradiction disappears in the light of the critical analysis that has been done in the previous sections.

GRAPH-COLORING. The existence of ε-approximate algorithms for K-MINIMUM-CLUSTER does not imply at all the existence of polynomial-bounded optimal algorithms for GRAPH-COLORING, when the function (10) is used.

HAMILTONIAN CYCLE. The existence of ε-approximate algorithms for TRAVELLING-SALESMAN does not imply at all the existence of polynomial-bounded optimal algorithms for HAMILTONIAN CYCLE when the function (10) is used.

For this two last problems, the adoption of the relative measure (7) has induced Sahni [6] to achieve just the opposite results.

REFERENCES

1. A. AIELLO, E. BURATTINI, A. MASSAROTTI et F. VENTRIGLIA, *A New Evaluation Function for Approximation Algorithms*, Proc. Informatica 77, Bled Yugoslavia, 1977.
2. A. AIELLO, E. BURATTINI, A. MASSAROTTI et F. VENTRIGLIA, *On the Optimization Problems Approximation*, Proc. Informatica 78, Bled Yugoslavia, 1978.
3. A. AIELLO, E. BURATTINI, A. MASSAROTTI et F. VENTRIGLIA, *Sulla struttura dei problemi combinatoriali*, Convegno "Complessita' di calcolo, modelli gestionali e territorio", Arco Felice (NA), 14-15 Giugno 1977.
4. D. S. JOHNSON, *Approximation Algorithms for Combinatorial Problems*, J. of Computer and System Science, Vol. 9, 1974, pp. 256-278.
5. R. M. KARP, *Reducibility Among Combinatorial Problems*. In *Complexity of Computer Computations*. R. E. Miller and J. W. Thatcher, Eds., Plenum Press, New York, 1972, pp. 85-104.
6. S. SAHNI et T. GONZALEZ, *P-Complete Approximation Problems*, J. Assoc. Comput. Mach., Vol. 23, 1976, pp. 555-565.