ON THE STRUCTURE OF COMBINATORIAL PROBLEMS AND STRUCTURE PRESERVING REDUCTIONS

Giorgio Ausiello - Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del CNR, Via Eudossiana, 18 00184 Roma.

Alessandro D'Atri - Istituto di Automatica dell'Università di Roma, Via Eudossiana, 18 - 00184 Roma.

Marco Protasi - Istituto di Matematica dell'Università di Roma, Piazzale delle Scienze - 00185 Roma.

ABSTRACT

In this paper the concept of combinatorial problem associated to an optimization problem is defined. A partial ordering over an optimization problem is then introduced: all input elements of an optimization problem are classified according to their "structure" (based on the number of approximate solutions of different measure) and then the classes of elements are partially ordered according to the richness of their structures. In this way structure preserving reductions among NP-complete problems can be introduced and some conditions for a reduction to preserve the structure of a combinatorial problem are given.

1. INTRODUCTION

In the recent years, after the basic work of Cook (1971) and Karp (1972) a great deal of research efforts have been devoted to the problem of finding reductions among NP-complete problems and of solving the well known open problem of establishing whether P is equal to NP. Even if this problem still remains open the study of NP-complete problems has brought a new light on some of their most interesting combinatorial properties and this is also the direction of our work.

In Hartmanis and Berman (1976) it is shown that all of the known NP-complete problems are, so to say, the same up to a polynomial time computable isomorphism. In other words all of the known problems can be reduced 1-1 one into the other by a mapping which takes polynomial time in the size of the input on a (deterministic) TM. At the same time both in Simon (1975) and in Hartmanis and Berman (1976) it is shown that these reductions preserve the multiplicity of the solutions.

These results seem to suggest a deep structural similarity among (all?) NP-complete problems but on the other side if we consider an optimization problem related to a combinatorial problem we may notice a differentiation among classes of NP-complete optimization problems. Sahni (1976), Sahni and Gonzales (1974) and Johnson (1976), in fact, show that while in some problems like "job sequencing" we can achieve arbitrarily good approximate solutions in polynomial time, for other families of problems (e.g. graph colouring) no polynomial time good approximation algorithm can exist. This seems to suggest an intrinsic differentiation among optimization problems and among their associated combinatorial problems and also the fact that not necessarily reductions among combinatorial problems can be taken as reductions among optimization problems.

In our work we started from the idea of studying the differences among NP-complete problems and of investigating such issues as

i) under what conditions, given a combinatorial problem we can speak of its subproblems and we can define its approximate solutions;

ii) what reductions among combinatorial problems are such that they map subproblems into subproblems and how these reductions behave with respect to the approximate solutions (do they preserve "good" approximations?);

iii) what ordering can we define among the subproblems of a combinatorial problem and what classes of combinatorial problems can be mapped one into the other by order preserving reductions.

Even if most of the questions are still far from being answered, in this paper we present some promising results and a framework for further research.

First of all we restrict ourselves to considering combinatorial problems which are associated to optimization problems with integer measure . Then, in §3 a partial ordering over an optimization problem is introduced: all input elements (instances) of an optimization problem are classifical according to their "structure" (based on the number of approximate solutions of different measure) and then the classes of elements are partially ordered according to the richness of their structures.

We next define (in §4) reductions which preserve the ordering and, more particularly, reductions which preserve the structure. Some conditions for a reduction to be structure preserving are then given and many examples of combinatorial problems which can be mapped one into the other by structure preserving reductions are shown.

2. OPTIMIZATION AND COMBINATORIAL PROBLEMS

In the following parts of this paper we will restrict ourselves to considering optimization problems which are "strictly related" to NP-complete combinatorial problems. Anyway we will give first some definitions which are more general than required and which are derived from similar definitions given by Johnson (1973). In these definitions we try to capture the basic objects which constitute an optimization problem: essentially *input* objects, *output* objects among which we find the *approximate solutions*, a *measure* over the solutions, an ordering on the values of the measure which characterizes an optimization problem as a *maximization* or a *minimization* problem.

DEFINITION 1. An *optimization problem* A is a 5-tuple
$A \equiv \langle \text{INPUT, OUTPUT, SOL, Q, m} \rangle$ where:
INPUT, OUTPUT are countable domains,
SOL: INPUT $\rightarrow$ $\{A \mid A$ is a finite subset of OUTPUT$\}$
is a recursive mapping that provides approximate solutions for any given element of INPUT
Q : is a totally ordered set
m : OUTPUT $\rightarrow$ Q

DEFINITION 2. i) The *optimal value* $m^*(x)$ of an input x of an optimization problem A is
$m^*(x) = \text{best}\ \{m(y) \mid y \in \text{SOL}(x)\}$

ii) The *optimal solutions* $A^*(x)$ of an input x are
$A^*(x) = \{y \in \text{SOL}(x) \mid m(y) = m^*(x)\}$.

With the notation SOL(x) we denote the set of elements of OUTPUT which are approximate solutions to the input x and the optimization problem is the problem of looking for the best solution in SOL(x).

Note that with the notation "best {...}" we mean the element of Q which is optimal in SOL(x) according to the ordering of Q. Since SOL(x) is finite such an element always exists. In most cases, and in all of our examples, the measure will have values over the integers, ordered under increasing order (maximization problems) or decreasing order (minimization problems). In order to be meaningful, to decide the set OUTPUT has to be "considerably simpler" than to compute SOL of a given x. For example, in the future, since we are interested in NP-complete combinatorial problems, we will ask OUTPUT to be decidable in polynomial time at most. Besides, also to compute the function m should require at most polynomial time.

EXAMPLE 1. MAX-CLIQUE: The problem consists in looking for the maximum complete subgraph in a given graph x.

INPUT : set of finite graphs

OUTPUT : set of complete finite graphs

SOL(x) : set S of complete subgraphs of a given graph x

Q : integers in increasing order

m : number of nodes of a complete graph.

EXAMPLE 2. MAX-SATISFIABILITY: This problem consists in looking for the maximum set of clauses w in a given formula of propositional calculus in CNF over $\{x_1,\ldots,x_r\}$

$$(C_{11} \vee \ldots \vee C_{1n_1}) \vee (C_{21} \vee \ldots \vee C_{2n_2}) \wedge \ldots \wedge (C_{m_1} \vee \ldots \vee C_{mn_m})$$

(where $C_{ij} = x_\ell$ or $\bar{x}_\ell$) such that w is satisfiable, and in exhibiting the satisfying truth assignment.

INPUT : set of formulae of propositional calculus in CNF

OUTPUT : $\{\langle w,f\rangle \mid$ w is a formula in CNF and f is a truth assignment which satisfies w$\}$

SOL(x) : $\{\langle w,f\rangle \mid$ w is a set of clauses of x and f is a satisfying truth assignment$\}$

Q : integers in increasing order

$m(\langle w,f\rangle)$= number of clauses in w.

In Appendix we give a list of optimization problems which we will refer to throughout the paper.

Given an optimization problem A, any total function f such that $(\forall x \in \text{INPUT})[f(x) \in \text{SOL}(x)]$ and that is polynomially computable, can be considered a polynomial approximation of A.

We will be mainly interested in problems that admit at least one polynomial approximation. For example, while MAX-CLIQUE and MAX-SATISFIABILITY admit polynomial approximations, this is not the case with MIN-EXACT-COVER (see Appendix), except, of course, if P = NP.

Strictly related to an optimization problem we have a recognition problem that we will call "associated combinatorial problem". The more natural recognition problem associated to an optimization problem, derives from the question of whether an element x of INPUT has an approximate solution of measure at least (at most) K or not. Most of the combinatorial problems appearing in the literature are of this type.

DEFINITION 3. Let A = $\langle$INPUT, OUTPUT, SOL, Q, m$\rangle$ be an optimization problem. The *combinatorial problem associated* to A is the set $A^c \subseteq \text{INPUT} \times Q$ where $A^c = \{\langle x,K\rangle \mid K \leq m^*(x)$ under the ordering of Q$\}$.

For example the combinatorial problem CLIQUE associated to the optimiza tion problem MAX-CLIQUE consists in deciding whether a graph x has a complete subgraph of (at least) K nodes or not. Besides, the combinatorial problem PARTIAL-SATISFIABILITY which is the set $\{\langle w,K\rangle | w$ has at least K satisfiable clauses$\}$ is clearly the recognition problem associated to MAX-SATISFIABILITY.

3. EQUIVALENCE CLASSES AND ORDERINGS OVER COMBINATORIAL PROBLEMS.

As we pointed out in the introduction our purpose is to find what are the relations among different NP-complete combinatorial problems not only in terms of mappings of one problem into the other, but also by taking into account the approximate solutions. For example we would like to find a way of mapping approximate solutions of one problem into corresponding approximate solutions of another problem.

The first intuitive approach would be to introduce a relation over a combinatorial problem such that any input is greater or equal than all of its subproblems, but it is easy to see that this approach would not allow to achieve a partial order.

It is instead necessary to introduce a classification of the elements of an optimization problem on the base of the combinatorial structure of the elements and then to introduce an ordering among the equivalence classes of elements. For a wide family of combinatorial problems we noticed that it is sufficiently meaningful to characterize the combinatorial structure of an element by the number of approximate solutions at any level of approximation and to classify the elements of a combinatorial problem according to this concept of combinatorial structure.

Note that this characterization is stronger than considering only the number of optimal solutions but weaker than considering the complete and-or graph of the subproblems of an input of a combinatorial problem.

DEFINITION 4. Let $A \equiv \langle$ INPUT, OUTPUT, SOL, Q, m$\rangle$ be an optimization problem and let $x \in$ INPUT; we define *structure of* x (denoted ℓ_x) the list of the cardinalities of the approximate solutions of x. Formally $\ell_x = \langle a_1,\dots,a_n\rangle$, $n \geq 0$ such that there exist $u_1,\dots,u_n \in Q$ with $u_i < u_{i+1}$ (for every $i < n$) with respect to the ordering of Q and

i) $(\forall i)\left[a_i = |A_i(x)|\right]$

ii) $A_i(x) = \{y | y \in \mathrm{SOL}(x) \text{ and } m(y) = u_i\}$ (approximate solutions at level u_i)

iii) $\bigcup_i A_i(x) = \mathrm{SOL}(x)$

Obviously $A_n(x) = A^*(x)$ as defined in definition 2.

DEFINITION 5. An optimization problem is said to be *convex* if for every $x \in$ INPUT, for every element a_i in ℓ_x (corresponding to a value u_i of the measure) the set of approximate solutions of x of measure q is non empty for every q such that $u_i \leq q \leq m^*(x)$.
For example, such problems as MAX-CLIQUE, MAX-SATISFIABILITY, MIN-NODE-COVER are convex while MAX-SUBSET-SUM is a typical non convex problem.

At this point we may define equivalence classes over INPUT:

DEFINITION 6. Let A be an optimization problem and let $x, y \in$ INPUT; we say that x *is equivalent to* y *in the problem* A ($x \equiv_A y$) if and only if they have the same structure (i.e. $\ell_x = \ell_y$). We denote $[x]_A$ the equivalence class of x under $\equiv_A$.

As an example of the fact that this concept of structure captures the intrinsic combinatorial property of a problem also in the case of non convex problem we may consider the problem MAX-SUBSET-SUM. If we consider the following inputs

$$x = \langle [3,5,5,7,2],10 \rangle \qquad y = \langle [6,10,10,14,4],20 \rangle$$

(where y is obtained from x by simply doubling all elements of the multiset and the constraint)
we have $$\ell_x = \ell_y = \langle 1,1,2,2,1,1,3 \rangle$$

and the list ℓ_y turns out to be equal to ℓ_x even if the corresponding lists of measures happen to be different.
In the following definition we begin to introduce a concept of ordering among classes of elements of INPUT and, hence, indirectly a more formal concept of subproblem.

DEFINITION 7. Let ℓ_r and ℓ_s be the structures of $r, s \in$ INPUT we say $\ell_r \leq \ell_s$ if $\ell_r = \langle a_1, \ldots, a_n \rangle$ and $\ell_s = \langle b_1, \ldots, b_m \rangle$ and there exist $i, \ldots, i_n$ with $i_1 < \ldots < i_n \leq m$ and $(\forall j < n)[a_j \leq b_{i_j}]$.
Clearly this relation is a partial order and induces a partial order over INPUT/$_{\equiv_A}$

DEFINITION 8. Let $r, s \in$ INPUT; we say that r is a subproblem of s in A (denoted $r \leq_A s$) if the structure of r is a substructure of s, formally $\ell_r \leq \ell_s$.

DEFINITION 9. Let $[r]_A, [s]_A \in$ INPUT/$_{\equiv_A}$; we say that $[r]_A \leq_A [s]_A$ if $r \leq_A s$.
Note that the concept of subproblem, instead, does not give rise to a

partial order over INPUT because the antisymmetric property does not hold in that case.

For example if we again consider the problem MAX-SUBSET-SUM and beside the already considered elements x and y we examine the structure of the input

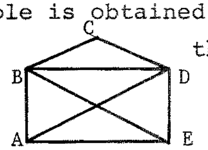
$$z = \langle [3,5,5,7,2,11],12\rangle \qquad \ell_z = \langle 1,1,2,2,1,1,3,1,3\rangle$$

we find out the following relations:

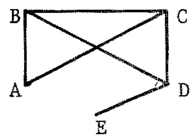
$$[x] = [y]$$

$$[x] \leq [z]$$

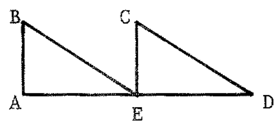
Another example is obtained if we consider the problem MAX-CLIQUE

if x is the graph 

the list is $\langle 1,5,8,5,1\rangle$

if y is the graph 

the list is $\langle 1,5,6,2\rangle$

and if z is the graph 

the list is $\langle 1,5,6,2\rangle$

so that clearly $[y] = [z] \leq [x]$.

4. ORDER PRESERVING REDUCTIONS

In Karp (1972) the concept of (many-one) reduction among combinatorial problems was first extensively studied.

In this paragraph we will consider reductions among combinatorial problems (associated to optimization problems) and we will be concerned of those reductions which preserve the "combinatorial structure" or at least the ordering defined according to it. Since we are interested in considering both constituents of an element of a combinatorial problem (i.e. the input and the value of the measure) we give a slightly modified definition of a reduction.

DEFINITION 10. Let A^c and B^c be two combinatorial problems associated to the optimization problems A and B; we say that $A^c \leq B^c$ (*A^c is poly-*

nomial by reducible to B^c) if there are two recursive functions $f_1 : INPUT_A \to INPUT_B$ and $f_2 : INPUT_A \times Q_A \to Q_B$ such that

$$\langle x,K \rangle \in A^c \text{ iff } \langle f_1(x),\ f_2(x,K) \rangle \in B^c$$

and f_1 and f_2 are polynomially computable on a Turing machine.

Obviously this definition implies Karp's definition. On the other side we have that most of Karp's reductions can be expressed in our formulation.

Since we are interested in studying what NP-complete problems can be reduced one to the other in such a way to preserve the structure of subproblems we first characterize a larger class of reductions that we call "order preserving" reductions because they preserve the ordering among the classes of inputs.

DEFINITION 11. Let A^c and B^c be two combinatorial problems and A and B the relative optimization problems; let $\langle f_1, f_2 \rangle$ be a polynomial reduction of A^c to B^c. We say that the reduction is order preserving ($\leq_{op}$) if $[x]_A \leq [z]_A$ implies $[f_1(x)]_B \leq [f_1(z)]_B$.

A very important class of reductions that, as we will see, can be used among many combinatorial problems are the reductions that map from the input element of one problem to an input element of another problem provided with the same structure.

DEFINITION 12. Let A^c and B^c be two combinatorial problems and A and B the relative optimization problems; the reduction $\langle f_1, f_2 \rangle$ of A^c to B^c is said to be *structure preserving* ($\leq_{sp}$) if for every $x \in INPUT_A$ and $y \in INPUT_B$, $f_1(x) = y$ implies $\ell_x = \ell_y$ where ℓ_x and ℓ_y are the structures of x in A and of y in B.

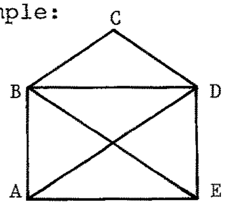
FACT. If a reduction is structure preserving it is also order preserving.

EXAMPLES i) CLIQUE $\leq_{sp}$ SET-PACKING
Let x be the graph $\langle N,A \rangle$

$f_1(x) = \{S_1, \ldots, S_n\}$ where $n = |N|$

$f_2(x,K) = K$ $\qquad S_i = \Big\{\{i,j\} \,\big|\, \{i,j\} \notin A\Big\}$

For example:



$S_A = \Big\{\{A,A\},\{A,C\}\Big\}$

$S_B = \Big\{\{B,B\}\Big\}$

$S_C = \Big\{\{C,C\},\{A,C\},\{E,C\}\Big\}$

$$S_D = \left\{\{D,D\}\right\}$$

$$S_E = \left\{\{E,E\},\{E,C\}\right\}$$

$$\ell_x = \ell_{f_1(x)} = \langle 1,5,8,5,1 \rangle$$

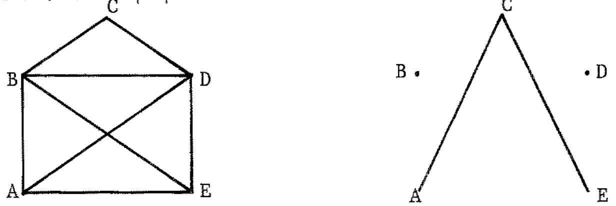
The only way the set S_U and the set S_V cannot be disjoint is if they both contain the element $\{U,V\}$. Hence, for every subgraph G' in x, for every arc $\{U,V\}$ in G' the corresponding sets S_U and S_V are disjoint and viceversa. This proves that for every K-clique in x there are K disjoint sets among $S_1, \dots, S_n$. Besides, if two K-clique differ for at least one node, also the two corresponding families of K disjoint sets are different and viceversa. This proves that the number of solutions of measure K in the problem CLIQUE is the same as the number of solutions of measure K in the problem SET-PACKING. Since $f_2(x,K) = K$ we have that if in the list of x there are b solutions of measure u and c solutions of measure $v > u$ in the list of $f_1(x)$ b will still precede c. Hence the reduction is shown to be structure preserving.

ii) CLIQUE $\leq_{sp}$ NODE-COVER

Let x be the graph $\langle N,A \rangle$

$f_1(x) = \bar{x}$ = the complement of the graph x

$f_2(x,K) = |N| - K$



The node covers of measure

1 are $\{C\}$

2 are $\left\{\{A,E\},\{A,C\},\{B,C\},\{D,C\},\{E,C\}\right\}$

3 are all triples except $\{A,B,D\},\{B,D,E\}$

4 are all fourtuples

5 are $\{A,B,C,D,E\}$

Since Q are the integers in decreasing order the list of $f_1(x)$ is $\langle 1,5,8,5,1 \rangle = \ell_x$. The proof that this reduction is structure preserving is similar to the preceding one.

Note that, in the example, the proofs make use of the following facts in order to prove that the reductions are structure preserving:

i) the reductions preserve the number of solutions at corresponding levels of the measures,

ii) the measures are related via monotonous functions.

A generalization of this observation is possible and brings the following definitions and results. First of all we translate into our terminology the concept of parsimonious reduction which is originally due to Simon (1975) and is studied in Berman and Hartmanis (1976):

DEFINITION 13. A reduction $f = \langle f_1, f_2 \rangle$ from A^c to B^c is said to be *parsimonious* if

$$(\forall x \in INPUT_A)(\forall K \in Q_A)$$

$$\left|\{y \in SOL_A(x) \mid m_A(y) = K\}\right| = \left|\{y \in SOL_B(f_1(x)) \mid m_B(y) = f_2(x,K)\}\right|$$

Since in our case we are interested in considering the number of approximate solutions of an element x for various values of the measure, we give the following definition

DEFINITION 14. A reduction $f = \langle f_1, f_2 \rangle$ from A^c to B^c is said to be *strongly parsimonious* if it is parsimonious and

$$(\forall z \in f_1(INPUT_A))(\forall h \in Q_B)$$

$$\left|\{y \in SOL_B(z) \mid m_B(y) = h\}\right| \neq 0 \rightarrow (\exists x \in INPUT_A)(\exists K \in Q_A)$$

$$\left[\left|\{y \in SOL_A(x) \mid m_A(y) = K\}\right| \neq 0 \text{ and } z = f_1(x) \text{ and } h = f_2(x,K)\right]$$

DEFINITION 15. A reduction of A^c to B^c is said to be *strictly monotonous* if for every x, $K_1 < K_2$ in Q_A implies $f_2(x,K_1) < f_2(x,K_2)$ in Q_B.

THEOREM 1. If a reduction is strongly parsimonious and strictly monotonous then it is structure preserving.

PROOF. If the reduction is strongly parsimonious then for every a_i in the structure of x there exists b_j in the structure of $f_1(x)$ such that $a_i = b_j$ and the length of the structure of x is greater or equal then the length of the structure of $f_1(x)$. Besides, since the reduction is strictly monotonous, given two elements of the list of x, a_{i_1}, and a_{i_2}, and the corresponding elements of the list of $f_1(x), a_{j_1}$ and a_{j_2}, if $i_1 < i_2$ then $j_1 < j_2$. This implies that for every i, $a_i = b_i$ and hence the two lists ℓ_x and $\ell_{f_1(x)}$ coincide. QED

In order to characterize a class of reductions among NP-complete problems which can be shown to be structure preserving, we notice that many known reductions are easily proved to be structure preserving by the following.

LEMMA. Let A be a convex optimization problem; let f be a reduction from A^c to B^c such that

i) f is separable into f_1 and f_2 (according to definition 10),

ii) f is parsimonious,

iii)
$$f_2(x,K)=\begin{cases} a(x)+k & \text{if } A \text{ and } B \text{ are both maximization (minimization) problems} \\ a(x)-k & \text{otherwise} \end{cases}$$

iv) f_2 is such that $f_2(x,\min\{m_A(y)\mid y\in SOL(x)\})=\min\{m_B(z)\mid z\in SOL(f_1(x))\}$

then f is structure preserving.

PROOF. First of all notice that condition iii) implies the strict monotonicity of f; besides condition ii), iii) and iv) and the fact A^c is convex imply that definition 14 is satisfied.
Now we are ready to state the next results which show that some well known reductions among NP-complete problems are indeed structure preserving.

THEOREM 2. The following sets can be reduced by structure preserving reductions in the following way:

i) CLIQUE $\leq$ SET-PACKING (*)

ii) CLIQUE $\leq$ NODE-COVER (*)

iii) NODE-COVER $\leq$ SET-COVERING-I

iv) NODE-COVER $\leq$ FEEDBACK-NODE-SET

v) SET-COVERING I $\leq$ HITTING-SET

vi) CHROMATIC-NUMBER $\leq$ EXACT-CLIQUE-COVER

PROOF. Clearly CLIQUE, NODE-COVER, SET-COVERING I, CHROMATIC-NUMBER are combinatorial problems associated to convex optimization problems. Besides the reductions appearing in Karp (1972) or obtained by simple variations, are parsimonious (see also Simon (1975)) and satisfy conditions iii) and iv) of the lemma. QED

In Theorem 2, we have only considered convex problems, but structure preserving reductions among non convex problems can be found. As an example we have:

THEOREM 3. PARTITION $\leq_{SP}$ SUBSET-SUM

(*) i) and ii) have already been proved by a direct argument.

PROOF. Let us consider the following reduction

$$f_1(S) = \langle S, \left\lfloor \frac{1}{2} \sum_{x \in S} x \right\rfloor \rangle$$

$$f_2(S,k) = \left\lfloor \frac{1}{2}\left(\sum_{x \in S} x - k\right) \right\rfloor$$

It is easy to see that the reduction is structure preserving QED.

As a last result it is possible to show an example of the fact that there are NP-complete problems which are not reducible each other in any direction via structure preserving reductions.

THEOREM 4. SIMPLE-CUT $\underset{SP}{\not\lessgtr}$ CUT-INTO-EQUAL-SIZE-SUBSETS

PROOF. Given any graph G of n nodes considered as an input to SIMPLE-CUT , if $\langle a_1,\dots,a_k\rangle$ is its structure, we have that the property $\sum_{i=1}^{k} a_i = 2^{n-1}$ must be satisfied. On the other side if G' is a graph of 2m nodes considered as an input to the problem CUT-INTO-EQUAL-SIZE-SUBSETS we have that the property to be satisfied by the structure $\langle b_1,\dots,b_{k'}\rangle$ is $\sum_{i=1}^{k'} b_i = \frac{1}{2}\binom{2m}{m}$. Hence no structure preserving reduction can be found neither from SIMPLE-CUT to CUT-INTO-EQUAL-SIZE-SUBSETS, nor viceversa. QED

5. CONCLUSIONS

In this paper we have made a first attempt to introduce a relation among classes of elements of an optimization problem so that such concepts as those of "subproblem", "approximate solution", "combinatorial structure" could be formally defined. On one side the main purpose was to find what reductions among what combinatorial problems would preserve the ordering. In this direction it has been shown that many NP-complete combinatorial problems can be mapped by structure preserving reductions, that is by reductions that preserve the "pattern" (so to say) of approximate solutions of the combinatorial problem, while, on the other side, there are NP-complete problems which are not reducible via structure preserving mappings.

On the other hand, this framework has been proposed also with the aim of finding a differentiation among various classes of NP-complete problems. A refinement of the class of NP-complete problems seems to be necessary to explain a different behaviour of these problems with respect to approximation algorithms. From this point of view this paper is just a starting point and research in this direction is still in progress.

AKNOWLEDGEMENTS

We wish to thank Allan Borodin and Marco Gaudiano for many valuable suggestions.

6. BIBLIOGRAPHY

[1] S.COOK: *The complexity of theorems proving procedures.* Proc 3rd ACM Symposium Theory of Computing (1971).

[2] M.R.GAREY, D.S.JOHNSON: *The complexity of near-optimal graph coloring.* Journal of ACM (1976), 23-1.

[3] J.HARTMANIS, L.BERMAN: *On isomorphisms and density of NP and other complete sets.* Proc 8th ACM Symposium Theory of Computing (1976).

[4] D.JOHNSON: *Approximation algorithms for combinatorial problems.* JCSS (1974), 9.

[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).

[6] S.SAHNI: *Algorithms for scheduling independent tasks.* Journal of ACM (1976), 23-1.

[7] S.SAHNI, T.GONZALES: *P-Complete problems and approximate solutions.* Proc IEEE 15th annual Symposium on Switching and Automata Theory (1974).

[8] J.SIMON: *On some central problems in computational complexity.* Cornell University Department of Computer Science TR 75-224 (1975).

APPENDIX

We give a list of optimization problems whose definition is given in our terminology. Note that the corresponding combinatorial problems in the text are denoted by erasing the prefix MAX or MIN Besides Q will be the set of integers in increasing order (MAXimization problems) or in decreasing order (MINimization problems) .

MIN-NODE-COVER:

- INPUT : set of finite graphs.
- OUTPUT: set of pairs $\langle A,N \rangle$ where A is a finite set of pairs of integers such that for any $\langle z,y \rangle \in A$. $z \in N$ or $y \in N$.
- SOL(x): set of pairs $\langle A',N' \rangle$ where A' is the set of arcs of x and N' is a subset of nodes of x which covers A'.
- m : number of nodes of the set N'.

MAX-SET-PACKING:

- INPUT : set of families of finite sets.
- OUTPUT: set of families of mutually disjoint finite sets.
- SOL(x): families S' of mutually disjoint elements of x.
- m : number of sets in the family S'.

MIN-FEEDBACK-NODE-SET:

- INPUT : set of finite digraphs.
- OUTPUT: set of pairs $\langle C',N \rangle$ where C' is a set of cycles over the integers and N is a set of integers such that any cycle contains an integer of N.
- SOL(x): set of pairs $\langle C,N \rangle$ where C is the set of all cycles of x and N is a set of nodes of x such that any cycle contains a node of N.
- m : number of nodes in N.

MIN-CHROMATIC-NUMBER:

- INPUT : set of finite graphs.
- OUTPUT: set of pairs $\langle N,\Pi \rangle$ where N is a finite set and Π is a partition of N.
- SOL(x): $\langle N,\Pi \rangle$ where N is the set of nodes of x and (y,z) arc of x implies that y and z have to be in two different classes of the partition Π.
- m : number of classes of Π.

MIN-SET-COVERING(I,II):

- INPUT : set of finite families of finite sets.
- OUTPUT: set of pairs $\langle F,S \rangle$ where F is a family of finite sets and $S = \bigcup_{S' \in F} S'$.

SOL(x): set of pairs $\langle F,S\rangle$ where $S = \bigcup_{S_i \in x} S_i = \bigcup_{S_i \in F} S_i$ and $F \subseteq x$.

(I)m : $|F|$

(II)m : $\sum_{S_i \in F} |S_i|$

MIN-EXACT-COVER: as MIN-SET-COVERING I where F is a family of disjoint sets.

MIN-EXACT-CLIQUE-COVER:

INPUT : set of finite graphs.

OUTPUT: set of families of complete disjoint finite graphs.

SOL(x): if $x = \langle N,A\rangle$ then SOL(x) is the set of families of complete disjoint finite graphs $F = \{G_1,\ldots,G_N\}$ such that $\bigcup_{1 \leq i \leq n} N_i = N$

m : $|F|$

MIN-HITTING-SET:

INPUT : set of finite families of finite sets.

OUTPUT: set of pairs $\langle F,S\rangle$ where F is a family of sets and S is a set such that $(\forall S_i \in F)\left[|S \cap S_i| \geq 1\right]$.

SOL(x): set of pairs $\langle x,S\rangle$ where $S \subseteq \bigcup_{S_i \in x} S_i$ and $(\forall S_i \in x)\left[|S \cap S_i| \geq 1\right]$.

m : $|S|$.

MAX-SUBSET-SUM:

INPUT : set of pairs $\langle T,b\rangle$ where T is a finite multiset of integers and b is an integer.

OUTPUT: set of pairs $\langle S,c\rangle$ where S is a finite multiset of integers, c is an integer and $\sum_{x_i \in S} x_i \leq c$.

SOL(x): if $x = \langle T,b\rangle$ then SOL(x) is the set of pairs $\langle S,b\rangle$ such that $S \subseteq T$.

m : $\sum_{x_i \in S} x_i$.

MAX-SIMPLE-CUT:

INPUT : set of finite graphs.

OUTPUT: set of triples $\langle R,N,M\rangle$ where $M \subseteq N$, $R \subseteq M\times(N-M)$.

SOL(x): if $x = \langle N,A\rangle$ then SOL(x) is the set of triples $\langle R,N,M\rangle$ where $R = \{\langle a,b\rangle \in A \mid a \in M \text{ and } b \in (N-M)\}$

m : $|R|$.

MIN-CUT-INTO-EQUAL-SIZE-SUBSETS:

INPUT : set of triples $\langle G,a,b\rangle$ where G is a finite graph, a and b are nodes of G.

OUTPUT: set of 5-tuples $\langle R,N,M,c,d\rangle$ where $R \subseteq M\times(N-M)$, $|M| = |N-M|$,

$c \in M$, $d \in N-M$.

SOL(x): 5-tuples $\langle R,N,M,a,b \rangle$ if $x = \langle G,a,b \rangle$, N are the nodes of G and $R = \{\langle u,v \rangle \in A \mid u \in N,\ v \in N-M\}$.

m : $|R|$.

MIN-PARTITION:

INPUT : set of multisets S.

OUTPUT: set of pairs of multisets $\langle A,B \rangle$ such that

$$\sum_{x \in A} x \leq \sum_{x \in B} x .$$

SOL(x): set of pairs $\langle A,B \rangle$ corresponding to dichotomies of S.

m : $\sum_{x \in B} x - \sum_{x \in A} x$.