NON-DETERMINISTIC POLYNOMIAL OPTIMIZATION PROBLEMS AND THEIR APPROXIMATION

Abridged Version

A. PAZ
Dept. of Computer Science
TECHNION-Israel Institute of Technology

S. MORAN
Dept. of Mathematics
TECHNION-Israel Institute of Technology

1. INTRODUCTION

NP-problems are considered in this paper as recognition problems over some alphabet Σ, i.e. $A \subset \Sigma^*$ is an NP problem if there exists a NDTM (non-deterministic Turing machine) recognizing A in polynomial time. It is easy to show that the following theorem holds true.

<u>Theorem 1</u>: Let A be a set in NP. Then there exists a NDTM M_A which recognizes A such that $M_A = M_{\mu_A} \circ M_{\pi_A} \circ M_{A_1}$, where

1) The operation "o" is defined as follows: $M_1 \circ M_2(x)$ is $M_1(M_2(x))$; M_1, M_2 are Turing machines and x is an input tape.

2) M_{A_1} is a polynomial time deterministic encoding machine. Its task is to encode an input $a \in A$ in some proper way to be denoted by a'.

3) M_{π_A} is a NDTM which choses some permutation $\pi(a')$ out of a possible subgroup of the group of all permutations of the encoded input tape a' in polynomial time.

4) M_{μ_A} is a polynomial time DTM which computes a number $\mu(\pi(a'))$.

5) $a \in A$ iff $\begin{cases} \mu(\pi(a')) \leq K_a & \text{(min problem)} \\ \mu(\pi(a')) \geq K_a & \text{(max problem)} \end{cases}$

where K_a is a number computed in polynomial time by the machine M_{A_1} (K_a is part of the encoding of a).

Thus every NP problem can be represented as an optimization problem and the recognition process can be split into three stages where the non-deterministic stage (the machine M_{π_A}) is separated from the other stages.

Example: Let A be the following (MAX SAT) problem: $a \in A$ iff a is a string of the form $(C_1,\ldots,C_p,K)$ where the C_i are clauses over a set of variables $\{x_1,\bar{x}_1,x_2,\bar{x}_2,\ldots,x_n,\bar{x}_n\}$; K is an integer ($K \leq p$) and, there exists a truth assignment to the variables that satisfies at least K clauses (if $K=p$ then the problem is SAT). M_A a machine that recognizes A can be constructed as $M_A = M_{\mu_A} \circ M_{\pi_A} \circ M_{A_1}$ where: M_{A_1} checks if the input is well formed; if so it does not change it. If not it stops in a rejecting state. M_{π_A} induces a permutation on the clauses and then induces another permutation on the literals within some of the clauses.

M_{μ_A} performs the following algorithm :

(1) Set $\mu \leftarrow 0$;

(2) If all clauses are marked, halt and return μ ;

(3) Let C_i be the first unmarked clause. If C_i is empty then mark C_i and go to (2) ;

(4) Let σ be the first variable appearing in C_i. Set $\mu \leftarrow \mu+1$ mark C_i ;

(5) For each C_j if $\sigma \in C_j$ then mark C_j, Set $\mu \leftarrow \mu+1$;

(6) For each unmarked C_j, $C_j \leftarrow C_j \setminus \{\bar{\sigma}\}$;

(7) Go to (2).

It is easy to see that the above algorithm is polynomial. After the algorithm is completed, the machine checks if $\mu \geqslant K$ and stops in an accepting state if the inequality holds true, and in an rejecting state otherwise. It is easy to see that M_A recognizes the set A as required.

Proof of Theorem 1 follows from the fact that every problem in NP is polynomially reducible to SAT, which is, as mentioned above, a special case of MAX SAT. The reduction is then incorporated into M_{A_1} of the above example.

2. NP OPTIMIZATION PROBLEM (NPOP)

The conjecture that $P \neq NP$ is widely believed to be true. This conjecture prompted many researchers to develop and study polynomial approximations for problems in NP, when considered as optimization problems. See e.g. [Jo 73] or [Sa 76].

The previous section points towards the possibility of a new approach to the study of NP problems and NP optimization problems. In what follows, an attempt is made to develop that new approach. The results achieved so far are promising. These results provide some new insight into recently proved approximation results and it is hoped that they will serve as a basis for a more extensive theory of combinatorial approximations.

Definition 1: An NP optimization problem (NPOP) is a subscripted 4-tuple $(A,F,t,\mu)_{EXT}$ where :

$$EXT = MIN \text{ or } EXT = MAX.$$

$A \subset \Sigma^*$ is a polynomial time recognizable recursive set over a finite alphabet Σ (A is the set of all well formed encodings of some given combinatorial entity e.g. graph, family of sets, logical sentence in CNF, etc.). It is assumed that $\lambda \in A$ where λ denotes the empty word.

F is a function $F: A \to P_0(A)$ (the set of all finite subsets of A), where for all $a \in A$, $F(a)$ is a subgroup of the group of all permutations of a, to be called "the set of proper permutation of a". An element in $F(a)$ will be denoted by $\pi(a)$. It is also assumed that the many valued function $a \to \pi(a)$ is computed in polynomial time by a NDTM *("permutation machine")*.

t is a function $t: A \to P_o(Z \cup \{\pm\infty\})$. t is a function intended to specify the property of the elements of A we want to study e.g. the number of clauses which are satisfiable in a given CNF formula, the number of nodes that are pairwise adjacent in a given graph, etc. With regard to the function t we sahll use the following notation

$$op(a) = \text{optimum}(K: K \in t(a)) \quad \text{where}$$

optimum is "max" if EXT = MAX and it is "min" if EXT = MIN. We shall use the value $-\infty$ in connection with MAX problems and the value $+\infty$ in connection with MIN problems.

It is also assumed that F is compatible with t, that is: $a' \in F(a)$ implies that $t(a') = t(a)$, that $t(\lambda) = \{0\}$, and that $t(a) \neq \emptyset$ for all $a \in A$.

μ is a polynomial time function (the measure function) $\mu: \Sigma^* \to Z \cup \{\pm\infty\} \cup \{\alpha\}$ $(\alpha \notin Z)$ satisfying the following properties :

(1) $\mu(w) = \alpha$ iff $w \notin A$;

(2) $(\mu(a) = K) \to K \in t(a)$;

(3) $(\forall a \in A)(\exists \pi^*(a) \in F(a))\ (\mu(\pi^*(a)) = op(a))$.

It should be noticed that the combinatorial properties (and the complexity) of a given NPOP are determined by A, t and the subscript EXT. We shall therefore abbreviate our notation and use the notation $(A,t)_{EXT}$ or $(A,t,\mu)_{EXT}$ whenever the other parameters are not relevant to the context.

Examples: (1) The problem mentioned before MAX SAT can be described in the form $(A,t)_{MAX}$ there A is the set of all CNF formulas and for $a \in A$, $K \in t(a)$ iff there is a truth assignment to the variables occuring in a which satisfies exactly K clauses.

(2) Colorability: $(\mathcal{G},t)_{MIN}$ where $\mathcal{G}$ is the set of all graphs and for $G \in \mathcal{G}$, $K \in t(G)$ iff G is K-colorable.

Remark: When considering NP problems as recognition problems a distinction should be made between *"polynomially constructive"* solutions and *"nonconstructive"* solutions. Consider e.g. the problem of ascertaining whether a given planar graph is 4-colorable. By the 4-color theorem recently proved, every planar graph is 4-colorable and therefore this problem is trivially in P. On the other hand no deterministic P algorithm for coloring a planar graph by 4-colors is known. Some of the definitions in this section and the following sections of our work are intended to deal with both constructive and nonconstructive aspects of NP and NPOP solvability. On the other hand, this version of the paper <u>will not</u> be concerned with the constructive aspects of the problem. Those aspects will be considered in the full paper.

3. REDUCTIONS BETWEEN NPOP's

On the basis of the previous definitions we are able to define and study reducibility and in particular polynomial reducibility between NPOP's.

Definition 2: Let $(A_1,t_1)_{EXT_1}$ and $(A_2,t_2)_{EXT_2}$ be two NPOP. Then $g\colon \Sigma^* \to \Sigma^*$ is a (polynomial) reduction of the first NPOP into the second iff g (is a polynomial function which) satisfies the following conditions :

(1) $a_1 \in A_1$ iff $g(a) \in A_2$;

(2) There exists a (polynomial time) function $f\colon A_1 \times Z \to Z$ such that: $\forall a_1 \in A_1$, $f(a_1, op(g(a_1))) = op(a_1)$ (that is, one can compute $op(a_1)$ if $op(g(a_1))$ is known). The reduction is *order preserving* if the above function f satisfies the following additional conditions :

Let $a_1 \in A_1$ and $a_2 = g(a_1) \in A_2$, then

(2.1) $\forall K \in t_2(a_2)$, $f(a_1,K) \in t_1(a_1)$;

(2.2) $\forall K_1,K_2 \in t_2(a_2)$ it is true that

$$K_1 < K_2 \leftrightarrow \begin{cases} f(a_1,K_1) < f(a_1,K_2) & \text{if } EXT_1 = EXT_2 \\ f(a_1,K_1) > f(a_1,K_2) & \text{else .} \end{cases}$$

That is: one can derive better approximations to $op(a_1)$ from better approximations to $op(a_2)$. An order preserving reduction is *measure preserving* if the function f satisfies also the property;

(2.3) $(\forall a_1 \in A_1)(\forall K \in Z)$, $f(a_1,K) = K$.

The measure preserving reductions have the property that any measure μ_2 on $(A_2,t_2)_{EXT_2}$ induces a measure μ_1 on $(A_1,t_1)_{EXT_1}$ such that $\mu_1(a_1) = \mu_2(a_2)$ ($a_1 \in A_1$ and $a_2 = g(a_1) \in A_2$). It is easy to show that measure preserving reductions can exist only between NPOP's such that $EXT_1 = EXT_2$. The importance of measure preserving reductions will be illustrated in Section 4, Lemma 3.

The notation "$(A,t_1)_{EXT_1} \leqslant (B,t_2)_{EXT_2}$" will be used to denote reducibility, where "$\underset{p}{\leqslant}$" denotes polynomial reducibility and "$\overset{g}{\underset{p}{\leqslant}}$" denotes polynomial reducibility with corresponding function g.

As in the case of NP problems, many NPOP's are reducible one to another by a measure preserving reduction. For example, let MAX CLIQUE be the following NPOP: (G,t_{MC}) where G is the set of all graphs, and for $g \in G$, $t_{MC}(g)=\{K \mid G$ contains a complete subgraph with K nodes$\}$. We shall show now that MAX SAT $\underset{p}{\leqslant}$ MAX CLIQUE: Let: $\{C_1,\dots,C_p\}$ be an input for MAX SAT, where each C_i contains literals from $X = \{x_1,\bar{x}_1,\dots,x_n,\bar{x}_n\}$. We reduce it to the following graph $G(N,A)$, where :

$$N = \{V_{\sigma,i} \mid \sigma \in X, \quad \text{and} \quad \sigma \in C_i\}$$
$$A = \{(V_{\sigma,i}, V_{\tau,j}) \mid \tau \neq \bar{\sigma} \quad \text{and} \quad i \neq j\} .$$

(This is, in fact, the reduction mentioned by Karp ([Ka 72]), extended to NPOP's.)

It will be shown now that the class of NPOP's can be divided into two subclasses, such that no problem in one class can be reduced by a measure preserving reduction to a problem in the second class (unless P = NP).

Definition 3: Let $(A,t)_{EXT}$ be a NPOP. Then for each $K \in Z$, $(A,t)_{EXT,K} = \{a \mid a \in A \text{ and } OP(a) \leqslant K\}$.

Definition 4: $(A,t)_{EXT}$ is a *"simple NPOP"* iff for all $K \in Z$, $(A,t)_{EXT,K}$ is a set in P. It is a *"rigid NPOP"* if it is not simple (i.e. for some K, $(A,t)_{EXT,K}$ is in NP\P, where the notation NP\P stands for the sets which are in NP and are not in P provided that $P \neq NP$).

Theorem 2: If $(A,t_1)_{EXT}$ is a rigid NPOP and $(B,t_2)_{EXT}$ is a simple NPOP, then

$$(A,t_1)_{EXT} \not\leqslant_p (B,t_2)_{EXT} .$$

Proof. Let $K_o \in Z$ be such that $(A,t_1)_{EXT,K_o} \in NP\backslash P$. Assume that $(A,t_1)_{EXT} \overset{g}{\leqslant}_p (B,t_2)_{EXT}$. The following polynomial algorithm will check for each $w \in \Sigma^*$ if $w \in (A,t_1)_{EXT,K_o}$:

(a) check if $w \in A$, if not reject ;

(b) reduce w by g to $b \in B$;

(c) check whether $b \in (B,t_2)_{EXT,K_o}$. If so accept else reject.

(Clearly, $b \in (B,t_2)_{EXT,K_o} \Leftrightarrow w \in (A,t_1)_{EXT,K_o}$) .

All three steps of the algorithm are polynomial, so that the algorithm is polynomial as a whole. It follows that $(A,t_1)_{EXT,K_o} \in P$, which is impossible. The theorem is thus proved.

If $P \neq NP$, then a set A is NP complete implies that $A \in NP\backslash P$. Combining this with the known NP completeness results, all known NPOP's can be shown to be either rigid or simple. Some examples are given below :

RIGID NPOP's

(a) Colorability (see [S 73]). *Planar* colorability is a special type of rigid NPOP as there is only one K(=3), for which $(A,t)_{EXT,K}$ is NP complete .

(b) Bin Packing = $(IS, t_{BP})_{MIN}$

$IS = \{(a_1,\dots,a_n,a_{n+1}) \mid \forall i \quad a_i \in Z\}$

$t_{BP}((a_1,\dots,a_n,a_{n+1})) = \{K \mid$ the set $\{a_1,\dots,a_n\}$ can be divided into K subsets, the sum of numbers in each of them $\leqslant a_{n+1}\}$.

SIMPLE NPOP's

(a) MAX SAT

(b) MAX CLIQUE.

Some additional properties of reducibility will be studied and proved in the full paper. In particular, a theorem similar to the Cook theorem ([Co 71]) will be shown to hold true, i.e. we shall show that the NPOP described below has the property that any other NPOP can be reduced to it by a measure preserving reduction. The NPOP is $OSAT = (OCNF, t_{OS})_{EXT}$ where:

$OCNF = \{(C_1,\dots,C_t,\ y_o,\dots,y_\ell \mid$ the C_i's are clauses over a set of variables X, and $\{y_o,\dots,y_\ell\} \subset X\}$. The y_i's are the *"Measure variables"*.

For $a \in OCNF$, $t_{OS}(a)$ will be defined as follows: Let X_a be the set of variables appearing in a, and let:

$$\mathcal{B}_a = \{B \mid B\colon X_a \to \{0,1\} \text{ is a Boolean function}\}.$$

Then $\pm\infty \in t_{OS}(a)$ iff there is a $B_\infty \in \mathcal{B}_a$ such that B_∞ does not satisfy the Boolean formula appearing in a. For $K \in Z$, $K \in t_{OS}(a)$ iff there is a $B_K \in \mathcal{B}_a$ such that B_K satisfies the Boolean formula appearing in a, and

$$K = \sum_{i=0}^{\ell} B_K(y_i)2^i .$$

The proof that every NPOP can be reduced to the above NPOP is derived by an *"adjustment"* of Cook's theorem for recognition problems, to NPOP's. The details of the proof will be given in the full paper.

4. P-APPROXIMATION FOR NPOP

The last section of the paper will deal with the problem of approximating NPOP in polynomial time.

Definition 5: A function $h\colon \Sigma^* \to Z \cup \{\pm\infty\} \cup \{\alpha\}$ is a p-approximation for an NPOP $(A,t)_{EXT}$ iff h is a polynomial (in the length of a) time function satisfying the following properties :

(1) $h(w) = \alpha$ iff $w \notin A$;

(2) $h(a) \geqslant op(a)$ if $EXT = MIN$ and $h(a) \leqslant op(a)$ if $EXT = MAX$.

The performance of a p-approximation h can be defined (see [Sa 76]) as follows :

$$(\forall a,\ a \neq \lambda)\ P_{h,(A,t)_{EXT}}(a) = \max_{\pi(a)\in F(a)} \left\{ \frac{|h(\pi(a))-op(a)|}{\min(h(\pi(a)),op(a))} \right\}.$$

And as a function of the length of the input the performance is defined as :

$$(\forall n \in Z)\ P_{h(A,t)_{EXT}}(n) = \max\{P_{h(a,t)_{EXT}}(a) \mid \ell(a) \leqslant n\}. \quad (\ell(a) = \text{length of } a.)$$

Definition 6: An NPOP $(A,t)_{EXT}$ is p-approximable iff for any $\varepsilon > 0$ there is a p-approximating function h for $(A,t)_{EXT}$ such that $P_{h,(A,t)}(a) \leqslant \varepsilon$ for all $a \in A$. $(A,t)_{EXT}$ is fully p-approximable iff for any $\varepsilon > 0$ there is a p-approximating function h as above with the additional property that h can be computed in polynomial time Q where $Q = Q(\ell(a)), \frac{1}{\varepsilon})$ i.e. Q is a polynomial in both the length of a and the value $\frac{1}{\varepsilon}$.

The importance of measure preserving reductions follows from the following :

Lemma 3. If $(A,t_1)_{EXT} \overset{g}{\underset{p}{\leqslant}} (B,t_2)_{EXT}$ then the following holds true : If $(B,t_2)_{EXT}$ is (fully) p-approximable then so is $(A,t_1)_{EXT}$ provided that g is measure preserving.

Proof: Let the time complexity of g be $P_o(n)$, for some polynomial P_o. Then, by definition for all $a \in A$, $\ell(g(a)) \leqslant P_o(\ell(a))$

Assume that (B,t_2) is fully p-approximable in $P(\ell(a), \frac{1}{\varepsilon})$ time for some polynomial P. One can assume that P is nondecreasing in both its variables, otherwise the negative terms may be omitted. We must show that $(A,t_1)_{EXT}$ is fully p-approximable in $P'(\ell(a), \frac{1}{\varepsilon})$ time for some (other) polynomial P'.

Let $a \in A$ and $\varepsilon > 0$ be given. Then we can find an ε approximation to op(a) using the following algorithm :

(1a) Reduce a by g to $b \in B$;

(1b) Find an ε-approximation to op(b) = op(a) (g is measure preserving).

Due to the fact that for measure preserving reductions $EXT_1 = EXT_2$, every ε-approximation to op(b) is also an ε-approximation to op(a) (both approximations will have the same value and op(a) = op(b)). The time required by Step (1a) of this algorithm is bounded by $P_o(\ell(a))$ and the time required by Step (1b) of the algorithm is bounded therefore by $P(\ell(b), \frac{1}{\varepsilon}) \leqslant P(P_o(\ell(a)), \frac{1}{\varepsilon}) = P_1'(\ell(a), \frac{1}{\varepsilon})$ (P was assumed to be nondecreasing). $P'(\ell(a), \frac{1}{\varepsilon}) = P_o(\ell(a)) + P_1'(\ell(a), \frac{1}{\varepsilon})$ is the required polynomial.

Remark: This proof will fit also the p-approximation case with some minor changes which are left to the reader.

Some results concerning p-approximable, and in particular fully p-approximable NPOP's, are represented below.

4.1 Necessary Condition for p-Approximability

Theorem 4: $(A,t)_{EXT}$ is p-approximable then $(A,t)_{EXT}$ is simple.

Proof. Let $(A,t)_{EXT}$ be p-approximable, and let $K \in Z$ be given. Then $(A,t)_{EXT,K}$ is in P: by definition, for each $\varepsilon \geqslant 0$ there is a polynomial (time) function $h_\varepsilon : \Sigma^* \to Z \cup \{t\}$, such that $\forall a \in A$, $\frac{|h_\varepsilon(a)-op(a)|}{\min\{h_\varepsilon(a),op(a)\}} < \varepsilon$.

Let EXT = MAX. (The other case is similar and is omitted.) $h_\varepsilon(a)$ and $op(a)$ are integeres by definition and $h_\varepsilon(a) \leqslant op(a)$. Thus, $h_\varepsilon(a) > K$ implies that $op(a) > K$. On the other hand, choosing $\varepsilon = \frac{1}{K}$, the inequality

$$\frac{|h_\varepsilon(a)-op(a)|}{\min\{h_\varepsilon(a),op(a)\}} < \frac{1}{K} \text{ implies that } \frac{op(a)-h_\varepsilon(a)}{h_\varepsilon(a)} < \frac{1}{K} \text{ or } \frac{op(a)}{h_\varepsilon(a)} - 1 < \frac{1}{K}$$

and for $h_\varepsilon(a) \leqslant K$ this inequality is impossible unless $op(a) = h_\varepsilon(a)$.
It follows that

$$[h_{\frac{1}{K}}(a) \leqslant K] \Leftrightarrow [op(a) \leqslant K].$$

In other words $h_{\frac{1}{K}}$ is polynomial function that recognizes $(A,t)_{EXT,K}$.

Q.E.D.

It can be shown that the converse of Theorem 4 is not true, and that there are some simple NPOP's which are not p-approximable (the TSP(*) problem [PS 76] is an example), assuming $P \neq NP$.

4.2 Necessary Condition for Fully p-Approximability

Definition 7: $(A,t)_{EXT}$ is p-simple iff there is some polynomial $Q(x,y)$ such that $\forall K \in Z$, $(A,t)_{EXT,K}$ is recognizable in $Q(\ell(a),K)$ time.

Theorem 5: $(A,t)_{EXT}$ is fully p-approximable implies that $(A,t)_{EXT}$ is p-simple.

Proof. Let $(A,t)_{EXT}$ be fully p-approximable and let $K \in Z$ be given. Then $(A,t)_{EXT,K}$ is recognizable in $Q(\ell(a),K)$ time for some polynomial $Q(x,y)$: by definition there is some polynomial $Q'(x,y)$ such that $(A,t)_{EXT}$ is ε p-approximable in $Q'(\ell(a),\frac{1}{\varepsilon})$ time, choosing $\varepsilon = \frac{1}{K}$, $(A,t)_{EXT}$ is $\frac{1}{K}$ p-approximable in $Q'(\ell(a),K)$ time, and applying the same argument as in Theorem 4 we see that $(A,t)_{EXT,K}$ is recognizable in $Q'(\ell(a),K)$ time (that is: $Q = Q'$).

Q.E.D.

(*) See Appendix

Definition 8: Let $f: Z \to Z$ be a (recursive) function and let $(A,t)_{EXT}$ be a NPOP. Then :

$$(A,t)_{EXT,f(n)} = \{a \in A \mid op(a) \leq f(\ell(a))\} \quad .$$

The following Lemma 4 introduces a useful tool for recognizing p-simple NPOP's:

Lemma 6 : $(A,t)_{EXT}$ is p-simple implies that $(A,t)_{EXT,p_1(n)} \in P$ for any polynomial $p_1(n)$.

Proof: $(A,t)_{EXT}$ is p-simple implies that there exists a polynomial $Q(x,z)$, such that for all $K \in Z$, $(A,t)_{EXT,K}$ is recognizable in $Q(\ell(a),K)$ time. This implies that $(A,t)_{EXT,p_1(n)}$ is recognizable in $Q(\ell(a),p_1[(\ell(a))]) = p(\ell(a))$ time, where p is a polynomial. Q.E.D.

By the above lemma one can show that many of the simple NPOP's are not p-simple if $P \neq NP$ (e.g. MAX SAT, MAX CLIQUE, MAX CUT ([GJS 74]),), by showing that for some polynomial $p(n)$ (in general $p(n) = n$ will do), $(A,t)_{EXT,p(n)}$ is NP complete. MAX SUBSET SUM and JOB SEQUENCING WITH DEADLINE (JSD) ([Sa 76]) are p-simple.

In the full paper there will also be given sufficient conditions for fully p-approximability, and it will be shown that among most known NPOP's, all those problem that satisfy the necessary conditions for fully p-approximability also satisfy the sufficient conditions, while most known NPOP's which are not known to the fully p-approximable do not satisfy the necessary conditions and therefore cannot be fully p-approximable (unless P = NP). We shall also give sufficient conditions for p-approximability and discuss the relation between them and the necessary conditions.

BIBLIOGRAPHY

[Co 71] S.A. Cook: "The Complexity of Theorem Proving Procedures", 3-rd STOC (1971), pp. 151-158.

[GJS 74] M.R. Garey, D.S. Johnson, L. Stockmeyer: "Some Simplified NP-Complete Problems", 6-th STOC (1974), pp. 47-63.

[Jo 73] D.S. Johnson: "Approximation Algorithms for Combinatorial Problems", 5-th STOC (1973), pp. 38-49.

[Ka 72] R.M. Karp: "Reducibility among Combinational Problems", Complexity of Computer Computations, R.E. Miller & J.W. Thatcher (eds.) Plenum Press, N.Y. (1972), pp. 85-104.

[PS 76] C.H. Papadimitriou and K. Steiglitz: "Some Complexity Results for the TSP", 8th STOC (1976), pp. 1-9.

[Sa 76] S. Sahni: "General Techniques for Combinatorial Approximation". TR 76-6, University of Minnesota, Dept. of Computer Sciences.

[S 73] L. Stockmayer: "Planar 3-Colorability is NP Complete", SIGACT News, 5, No.3, (1973), pp. 19-26.

APPENDIX

The following NPOP's were mentioned in the paper, but were not formally defined:

(1) TSP (Travelling Salesman Problem): $= (W(G), t_{TSP})_{MIN}$, where $W(G)$ is the set of all weighted graphs $W(G)$, (that is graphs combined with a weight function $W: A \to Z$), and for a given weighted graph $W[G(N,A)]$, $t_{TSP}(W[G(N,A)]) = \{K \mid$ there exists a Hamiltonian cycle in the graph whose weight is $K\} \cup \{\pm\infty\}$ (we add $\pm\infty$ to $t_{TSP}(W[G(N,A)])$ to make sure that it is not empty).

(2) MAX CUT: $= (W(G), t_{CUT})_{MAX}$, where $W(G)$ is as above and $t_{CUT}(W[G(N,A)]) = \{K \mid$ A contains a cutset of weight $K\}$.

(3) MAX SUBSET SUM $= (IS, t_{SS})_{MAX}$ where $IS = \{(a_1, \ldots, a_n, a_{n+1})\}$ is the set of all finite integer sequences, and $t_{SS}((a_1, \ldots, a_n, a_{n+1})) = \{K \mid K \leqslant a_{n+1}$ and there are $1 \leqslant i_1 < \ldots < i_s \leqslant n$, $\sum_{j=1}^{s} a_{i_j} = K\}$.

(4) JSD (Job Sequencing with Deadlines) $= (IS^3, t_{JS})_{MAX}$ where:

$$IS^3 = \{(T_1, D_1, P_1, \ldots, T_n, D_n, P_n) \mid \{T_i, D_i, P_i\} \subset Z \quad \text{for } i = 1, \ldots, n\},$$

and

$t_{JS}((T_1, D_1, P_1, \ldots, T_n, D_n, P_n)) = \{K \mid$ there is a permutation σ of $(1, 2, \ldots, n)$ such that

$$\sum_{i=1}^{n} \delta_\sigma(i) P_\sigma(i) = K,$$

where

$$\delta_\sigma(i) = [\text{if } T_\sigma(1) + T_\sigma(2) + \ldots + T_\sigma(i) \geqslant D_\sigma(i) \text{ then } 1 \text{ else } 0].$$