**Stochastica XII-1 (1988), 19-31**

# COMPARING NOTIONS OF APPROXIMATION[1]

M. FURNARI AND A. MASSAROTTI

ABSTRACT

*In this note we discuss some drawbacks of some approaches to the classification of NP-complete optimization problems. Then we analyse the Theory of Analytical Computational Complexity to gain some insight about the notions of approximation and approximate algorithms. We stress the different roles played by these notions within the theories of Analytical and Algebraic Complexity. We finally outline a possible strategy to capture a more useful notion of approximation which is inspired by some results on Linear Programming problems.*

## Introduction.

Optimization problems seem to fall naturally into two categories: those with *continuous* variables, and those with *discrete* variables, which are called combinatorial optimization problems.

In the continuous problems, we are generally looking for a set of real numbers or even functions; in the combinatorial problems, we are looking for an object taken from a countable set; tipically: an integer, a set, a permutation set, or a graph.

Generally, to investigate the complexity of combinatorial optimization problems they are reduced to decision problems, then analyzed by means of the notions of the Theory of Algebraic Computational Complexity.

[1]This work was partly supported by Progetto Finalizzato Trasporti, C.N.R.

Unfortunately, many interesting combinatorial optimization problems belong to the class of NP-complete problems; hence it is of some interest to develop a theory capturing efficiently the intuitive notions of approximation.

To this end, many approaches have been proposed, see Johnson, Sahni, Paz and Moran, and Ausiello [1,2,3,4] which, however, all present some significant drawbacks, see Aiello [5].

We propose to analyse the Analytical Theory of Complexity to gain some insight about the notion of approximation, and its use.

In this note we first sketch both approaches (The Analytical and the Algebraic), next we analyse an interesting optimization problem, Linear Programming, that plays a unique role in optimization theory. It can be considered as a continuous optimization problem, and combinatorial in character. Its study is basic for many strict combinatorial problems.

Comparing the results obtained from the classification of Linear Programming, from the point of view of the Analytical and Algebraic theories of complexity, we stress out the different meaning which can be assigned to the notion of approximation in the two approaches. Taking this as a starting point we sketch a new interesting approach for an Algebraic Theory of approximate algorithms.

Basic concepts and terminology.

To provide a formal ground to study the properties of an optimization problem, we give first an abstract notion of optimization problem, next we specialize it in both theories.

An *instance of an optimization problem* is a pair $(X, c)$, where $X$ is any set, called *domain of feasible points;* $c$ is the *cost function,* i.e. a mapping $c : X \to \mathcal{R}$.

The problem is to find an $x \in X$ for which

$$c(x) \leq c(y) \quad \text{for all } y \in X \quad \text{for a minimization problem}$$

or

$$c(x) \geq c(y) \quad \text{for all } y \in X \quad \text{for a maximization problem}$$

Such point $x$ is called a *globally optimal solution* to the given instance or simply an *optimal solution.*

An *optimization problem* is a set $P$ of instances of an optimization problem.

It is important to distinguish between a *problem* and and *instance of a problem.* Informally, an *instance* is described by the input data $\mathcal{I}$ and sufficient information to obtain a solution (i.e., the set of properties to verify, denoted by $p$). A *problem* is a collection of instances, usually generated in a similar way.

The Algebraic Theory of Complexity.

A prototype of computational problem studied by the Algebraic theory of complexity has the following form and is called a *decision problem:*

1) a set of instances $\mathbf{I}_{\Pi_D}$, each of which is given by an input $\mathbf{I}$, and a set of properties $\mathcal{P}$ to verify.
2) a question about the existence of an object that verifies the properties $\mathcal{P}$.

For example:

Minimum Cover:

Instance: A collection of subsets $\{S_i\}$ of finite set $S$, a positive integer $k \leq i$;

Question: Does $\{S_i\}$ contain a cover $S$ of size $m \leq k$?

Hamiltonian Circuit:

Instance: a graph $G = (V, E)$;

Question: Does $G$ contain a Hamiltonian circuit?

Clique:

Instance: a graph $G = (V, E)$, a positive integer $K$;

Question: Does $G$ contain a clique of size $K$ or more; i.e. a subset $V \supseteq V'$ with $|V'| \geq K$ such that every two vertices in $V'$ are joined by an edge in $E$?

Node Cover:

Instance: a graph $G = (V, E)$, a positive integer $K \leq |V|$;

Question: Is there a node cover of size $K$ or less for $G$; i.e., a subset $V \supseteq V'$ with $|V'| \leq K$ such that for every edge $[u, v] \in E$ at least one of $u$ or $v$ belongs to $V'$?

The foundations of the Algebraic Complexity Theory were laid in the Stephen Cook paper [6]. In it many important facts are proved. First, Cook emphasized the significance of *polinomial time reductibility*; secondly, he focussed attention on the class NP problems, i.e. the class of decision problems that can be solved in polynomial time by a nondeterministic machine; thirdly, he proved that one particular problem in NP, called *satisfiability problem*, has the property that every other problem in NP can be reduced to it. Hence, if the satisfiability problem can be solved with a polynomial time algorithm, then so can every problem in NP. Karp in [7] proved that the decision version of many well known combinatorial problems share the same property.

Since a wide variety of other problems have been proved equivalent in difficulty to these problems (see Garey [8]). This equivalence class has been called *class of NP-complete problems,* it consists of the *hardest* problem in NP. Every problem belonging to this class shows the following interesting properties:

1. No NP-complete problem has been solved by an known polynomial algorithm.
2. If there is a polynomial algorithm for any NP-complete problem, then there are polynomial algorithms for all NP-complete problems.

To study the optimization problem from the point of view of the Algebraic Theory of Complexity, we shall assume that $X$ (the set of feasible solutions) and $c$ (the cost function) are given implicitly in terms of two algorithms $A_X$ and $A_c$. The algorithm $A_X$, given a combinatorial object $x$ and a set $S$ of parameters, will decide whether $x$ is an element of $X$, the set of feasible solutions specified by the given parameters. On the other hand $A_c$, given a feasible solution $x$ and another set of parameter $Q$, returns the value of $c(x)$.

Then the combinatorial optimization problem can be defined in the following way:

1. a set of instances $P_{\Pi_D}$, each given by an input $\mathcal{I}$, and a set of properties $\mathcal{P}$ to verify.
2. for each $\mathcal{I} \in P_\Pi$ there exists a finite set $S_\Pi(\mathcal{I})$, called the set of feasible solutions;
3. a function $m_\Pi(\omega)$ which computes the value of the cost function for every $\omega \in S_\Pi$.

The optimal solution will be the $\omega^* \in S_\Pi(\mathcal{I})$ such that $m_\Pi(\omega^*) \leq m_\Pi(\omega) \forall \omega \in S_P(\mathcal{I})$ if $P$ is a minimization problem (respectively; $m_P(\omega^*) \geq m_P(\omega) \forall \omega \in S_p(\mathcal{I})$ if $P$ is a maximization problem).

Within this framework any $\omega_\mathcal{I} \in S_\Pi(\mathcal{I})$ such that $\omega_\mathcal{I} \neq \omega^*_\mathcal{I}$ denotes an approximate solution for $P$, and an algorithm $A_\Pi$ which computes $\omega_\mathcal{I} \in S_\Pi(\mathcal{I})$ for every $\mathcal{I}$ is called an approximate algorithm for $P$.

It must be pointed out that, although it is possible to give a decision version of any optimization problem, it is not so simple and straightforward to give an optimization version for any decision problem. That is, there exist some decision problems that must be considered intrinsic decision problems. For example, consider the Hamiltonian circuit problem. In this case, to transform this decision problem into an equivalent optimization problem it is necessary to modify the set of properties to be verified, for example, relaxing the constraint that requires to visit each node of the graph. Hence a possible choice is given by:

find the maximum subgraph of $G = (V, E)$ which has an Hamiltonian circuit.

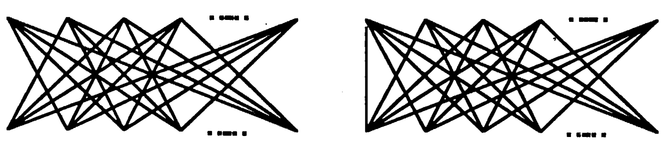
To capture the intuitive notion of closeness between solutions, one has to introduce the concept of performances measure of an approximate algorithm. Johnson and Sahni

[1,2] defined the following measure of performance in the worst case analysis:

$$R_{A_\Pi} = \begin{cases} \max\limits_{\mathcal{I}\in S_\Pi} m_\Pi(\omega_{\mathcal{I}^*})/m_\Pi(\omega^{A_\Pi}) & \text{for a maximization problem} \\ \max\limits_{\mathcal{I}\in S_\Pi} m_\Pi(\omega^{A_\Pi})/m_\Pi(\omega_{\mathcal{I}^*}) & \text{for a minimization problem} \end{cases}$$

Concerning this notion of measure of performance we may notice that:

1. it may be considered **natural**, i.e. captures the intuitive notion of closeness. However, there is no formal relation between its definition and the problem input data. Hence the choice is totally arbitrary.
2. it is not invariant under linear transformations of the measure of the feasible solutions. As immediate consequence we have that equivalent problems do not share the same properties with respect to the approximation class (for example Clique-Node cover) [5];
3. it is not continuous with respect to simple modifications of the problem input data. Generally, this behaviour is attributed to the specific heuristics used by the approximate algorithm. For example, it is well known that the problem of the chromatic number of a non planar graph belongs to the NP-complete class; however many approximate algorithms have been proposed: the most famous was proposed by Welsh & Powell. This algorithm utilizes the *priority to the bigger set* heuristics. If we give as inputs to this algorithm the bipartite graphs showed in fig. 1, we obtain that the Welsh & Powell algorithm gives a chromatic number $N/2$ for $G_a$ and 2 for $G_b$. For both graphs the chromatic number is 2.



a) $G_a = (V_1, E_1)$ b) $G_a = (V_2, E_2)$

Fig. 1. *Bipartite graphs. Where* $V_1 = V_2$, *and* $E_2 = E_1 \cup \{e_{1,2}\}$

As a result of these remarks, it seems natural to pose the following questions:

Why does the same heuristics give results more satisfactory for a given problem than for problems which exhibit similar structure?

Is the heuristics choice more strictly connected to problem structural characteristics or is it strictly connected to the approximate algorithms measure performance?

In [5] it was shown that, if the measure of performance for an approximate algorithm is invariant under linear transformations, then the heuristics of *priority to the bigger set* is unable to give, in the worst case analysis, results which are better than worst solution with respect to the optimal solution.

We can assert that the notions of approximation and approximate algorithms are weakly connected to the Algebraic Theory of Complexity. The meanings assumed by these notions are strictly related to the feasible solutions of a problem and show a deep connection with the choice of performance measure. Therefore these concepts appear superimposed on the framework of the Algebraic Theory of Complexity in an arbitrary manner.

The Analytical Theory of Complexity.

A prototype of problem considered into this theory has the following form:

$$\min_{x \in g} f_o(x) \qquad g = \{x \in G\, f_j(x) \leq 0,\ 1 \leq j \leq m\}$$

where $G$ is a closed subset of a Banach space $E$, $f_j \in \mathcal{F}\ 0 \leq j \leq m$ are continuous real functions defined on $G$ and called constraints, $f_o$ is the objective functional for the problem and $G$ is its domain.

Characterizing $G$ and $f_j$ it is possible to obtain an important class of problems, indeed if we suppose $G$ convex, $E = \mathcal{R}^n$ and $f_j$ convex functions on $G$, then the family of mathematical problems with fixed $m$, $G$ and $E$ represent the class of convex problems with $m$ constraints and domain $\mathcal{R}^n \supseteq G$.

In this theory [10,12] a numerical method of solution for a mathematical programming problem is represented as a set of rules to accrue the necessary information to solve the problem. The information source is fixed in advance and is called *oracle.*

Formally, for a given class of problems $P(\mathcal{F}, G, m, E)$, an oracle $O$ is given by an observation function $c(x, f) : G \times \mathcal{F} \rightarrow I$, where $I$ is a set called information space. For example, if the information space is $I = \mathcal{R}^{n+1}$ and $c(x, f) = f(x)$, then the oracle gives for each point of the domain the values of all the functionals of the problem at given point. A such type of oracle will be called a zeroth order oracle.

Applied to a given problem, a method of solutions gives a sequence of points $\{x_1, \cdots, x_r, x^o\}$. The sequence $\{x_1, \cdots, x_r\}$ represents the questions asked to the oracle looking for the solution, and the points $x^o$ represents the result of the application of the method. The sequence $\{x_1, \cdots, x_r, x^o\}$ will be called the *trajector of the method* on the given problem.

Generally, the problems approached with the Analytical theory are very hard, thus the solution method will guarantee only approximate solutions, not exact solutions.

In order to analyse and compare these solution methods, it is necessary to have some means to measure the error, i.e. the proximity level of the results given by a method. The *absolute error* is defined as the difference between the values of the objective functional computed, respectively, on the approximate and optimal solutions. The absolute error will be denoted by $\epsilon(x, f)$. To obtain a deeper insight, it is necessary define the *relative error.* This notion is defined by:

$$\nu(x, f) = \frac{\epsilon(x, f)}{\rho(f)}$$

where $\rho(f)$ represents a normalizing map. The normalizing map for the class of convex programming problems, universally chosen, is given by:

$$\rho(f) = \sup_{x \in G} - \inf_{x \in G} f(x)$$

In this theory the notion of approximation must be interpreted as even closer bringing to the optimal solution. In this approach the notion of approximation is strictly connected to the increasing of information on the problem furnished by the oracle. Therefore the notion of approximation represents an intrinsic feature of the problem rather than the set of feasible solution for it, as in the Algebraic Theory of Complexity. The statement *$x$ is an approximate solution for $f$ with a relative error $\nu$ must be interpreted as $x$ is an approximate solution for $f$ $1/\nu$ times better than the worst solution given by a trivial search.*

In Aiello & al. [5] a similar measure of the performances for approximate algorithms of the combinatorial optimization problems has been proposed. This type of measure is invariant under linear transformations of the measure of feasible solutions.

From this brief description of both theories, we can conclude that not only the mathematical methods used in the theories but also the meanings of similar notions are different. Furthermore, we can point out conflicting views in the search of methods to solve a given problem. In the Algebraic Theory of Complexity, complete information about the problem (its code) must be given as input for the method, and a limitation on the nature of the method is determined only by the type of process which is applied to the code of the problem in order to obtain the code of the solution (the method must be algorithmic in the sense of the computability theory). By contrast, in the Analytical Theory of Complexity, the picture is reversed: the methods that can be applied are in no way limited, but on the other hand the initial information about the problem is incomplete, and its acquisition has to be paid for.

Linear Programming.

An interesting comparison between these theories can be obtained by applying them to the problem of Linear Programming. This problem can be considered the easiest problem of convex organization, indeed the functions used are linear. Furthermore, its geometric structure can be used to bring out its combinatorial nature.

Within the framework of the theory of the Algebraic Complexity the Linear Programming problem has the folowing form:

Linear Programming:

Instance: An integer $n \times d$ matrix $\underline{A}$, an integer n-vector $\underline{b}$, an integer d-vector $\underline{c}$;

Question: Find a rational d-vector $\underline{x}$ such that $Ax \leq b$ and $c^T x$ is maximized.

The most popular algorithm that solves the Linear Programming problem, within the framework of the Algebraic Theory of Complexity, is the Simplex algorithm. This algorithm solves a linear programming problem by finding an initial feasible solution and then, if it does not maximize the objective function, a new feasible solution is found and the check for maximizing the objective function is made again. Empirical evidence for the complexity of the Simplex algorithm shows it to be quite efficient. However it can be proved that its complexity is exponential, under the worst case analysis [9].

Khachian, using the algorithm of Modified Method of Centres of Gravity by Shor & Yudin [11,12], showed that the Linear Programming problem belongs to the class of the polynomial time solvable problems on a deterministic Turing machine. The Shor algorithm can be viewed as an N-dimensional generalization of the well known search by bisection on a line segment.

Khachian [13] provided a computing time analysis of the ellipsoid algorithm to test a system of linear inequalities for satisfiability.

The ellipsoid algorithm for a solution to $Ax = b$, $x \geq 0$, where the input consists of $m \times n$, $m \leq n$, integer matrix $\underline{A}$ with columns $a_i$, $1 \leq i \leq n$, and an integer vector $\underline{b}$, either outputs a vector $\underline{p}$ such that $Ap = b$ or returns "no solution".

The method choses the initial ellipsoid $E_o$ in n-space, centred at the origin, which is guarantied theoretically to contain a feasible solution, if any exists. The method then recursively generates a sequence of ellipsoids $E_0, E_1, \cdots$, each one has a geometrically decreasing volume and containes all the feasible solutions of its precedessors. In the limit, a feasible solution will be approximated with arbitrary precision. Here the trajector of the method is given by the centres of the ellipsoids.

Khachian proves that if there is no solution in the first polynomially bounded sequence of elements of the trajector of the method, then no solution exists at all.

Wolfe [14] reported a long bibliography of results related to this algorithm, showing its impact on current research efforts. Grotschel et al. [15] and Karp and Papadimitriou [16] developed some interesting consequences of the ellipsoid algorithm in combinatorial problems and provided additional reasons for its study.

Final Remarks.

We can observe that in the framework of the Algebraic Theory of the Complexity:

1. the notions of approximation and approximate algorithms are scarcely correlated to the structural properties of the given problem, at least in the case of combinatorial optimization problems.
2. a lot of results seem strictly related to the choice of a measure of performance rather than to the effective *goodness* of the choosen method;

In the Analytical Theory of Complexity the same notions are more strictly related to the intrinsic features of the problem. The notion of approximation can than be considered arising naturally from the given problem and the solution methods are measured using more objective criteria.

We suggest that these results must not be considered as a simple and occasional, although very interesting, results, but they are intrinsically related to the approach of the

Analytical Complexity Theory. We think, in particular, that they are specifically related to the interpretation of the notions of approximation and approximate algorithm which are used in this theory.

References.

[1] Johnson, D.S. (1974) "Approximation algorithms for combinatorial problems". *J. Comput. System. Sci. 9*, 256-278.

[2] Sahni, S., Gonzalez, T. (1976) "P-complete approximation problems", *J. ACM 23,* 555-565.

[3] Paz A., Moran, S. (1977) "NP-optimization problems and their approximation". *Proc. 4th International Colloquium on Automata Languages and Programming,* Turku.

[4] Ausiello, G., Marchetti-Spaccamela, A. (1980) "Toward a unified approach for the classification of NP-complete optimization problems". *Theor. Computer Sci. 12*, 83-96.

[5] Aiello, A., Burattini, E., Massarotti, A., Ventriglia, F. (1979) "Towards a general principle of evaluation for approximate algorithms". *RAIRO, 13,* 227-239.

[6] Cook, S. (1971) "The complexity of theorem-proving procedures". *Proc. 3rd Ann. ACM Symp. on Theory of Computing,* New York, 151-158

[7] Karp, R. (1972) *Reducibility among combinatorial problems.* In Miller R.E. and Thatcher J.W. Eds. Complexity of Computer Computation, Plenum Press, New York.

[8] Garey, M.R., Johnson, D.S. (1979) *Computer and Intractability.* Freeman Press, San Francisco.

[9] Klee, V., Minty, G.J. (1972) *How good is the Simplex Method* In Sisda O. Inequalities III, Academic Pres, New York.

[10] Traub, J.F., Wozniakowski, H. (1980) *A general Theory of Optimal Algorithm.* Academic Press, New York.

[11] Shor, N.Z. (1977) "Cut-off method with space extension in convex programming problems", *Cybernetics 13,* 94-96.

[12] Yudin, D.B., Nemirovsky, A.S. (1983) *Problem complexity and method efficiency in optimization.* John Wiley & Sons.

[13] Khachian, L.G. (1979) "A polynomial algorithm in linear programming". *Soviet Mathematics Doklady 20,* 191-194.

[14] Wolfe, P. (1980) A bibliography for the ellipsoid algorithm. I.B.M. Research Center, Yorktown heights, New York.

[15] Grotschel, M., Lovasz, L., Schrijver, A. (1980) The ellipsoid method and its consequences in combinatorial optimization. Report 80151-OR, University of Bonn, W. Germany.

[16] Karp, R.M., Papadimitriou, C.H. (1980) "On linear characterization of combinatorial optimization problems". *Proc. Twenty- First Annual Symp. on Foundations of Comp. Sci., IEEE.*

Istituto di Cibernetica del C.N.R.

80072 Arco Felice

Napoli. ITALY