Note

On total functions, existence theorems and computational complexity

Nimrod Megiddo

IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

Christos H. Papadimitriou*

Computer Technology Institute, Patras, Greece, and University of California at San Diego, CA, USA

Communicated by M. Nivat
Received October 1989

Abstract

Megiddo, N. and C.H. Papadimitriou, On total functions, existence theorems and computational complexity (Note), Theoretical Computer Science 81 (1991) 317-324.

Nondeterministic multivalued functions with values that are polynomially verifiable and guaranteed to exist form an interesting complexity class between P and NP. We show that this class, which we call TFNP, contains a host of important problems, whose membership in P is currently not known. These include, besides factoring, local optimization, Brouwer's fixed points, a computational version of Sperner's Lemma, bimatrix equilibria in games, and linear complementarity for P-matrices.

1. The class TFNP

Let Σ be an alphabet with two or more symbols, and suppose that $R \subseteq \Sigma^* \times \Sigma^*$ is a polynomial-time recognizable relation which is *polynomially balanced*, that is, $(x, y) \in R$ implies that $|y| \leq p(|x|)$ for some polynomial p.

* Research supported by an ESPRIT Basic Research Project, a grant to the Universities of Patras and Bonn by the Volkswagen Foundation, and an NSF Grant. Research partially performed while the author was visiting the IBM Almaden Research Center.

The relation R defines the following computational problem Π_R: given an $x \in \Sigma^*$, find any $y \in \Sigma^*$ such that $(x, y) \in R$, if such a y exists, and reply "no" otherwise. The class of all such problems is denoted FNP. The subset of FNP that can be solved in polynomial time is called FP. At present it is not known whether FP = FNP; this question is equivalent to the P = NP question.

We call R *total* if, for every $x \in \Sigma^*$, there is always a $y \in \Sigma^*$ such that $(x, y) \in R$. We let TFNP (for *total functions from* NP) be the class FNP restricted to total relations R. In this note we point out that TFNP contains unexpectedly many natural and diverse problems that are not known to be in FP.

1.1. F(NP$\cap$coNP)

The class TFNP is not unfamiliar. It could be called F(NP$\cap$coNP), as it includes factoring and similar problems that are functional variants of problems in NP$\cap$coNP. Consider two polynomially balanced relations R_1 and R_2 in P such that for each x either there is a y with $(x, 1y) \in R_1$, or there is a z with $(x, 2z) \in R_2$. Here 1 and 2 are symbols used to differentiate between the "certificates" of the two kinds. A typical problem in F(NP$\cap$coNP) asks, given x, to find a y or a z, as appropriate.

Notice that TFNP coincides with F(NP$\cap$coNP). One inclusion takes $R = R_1 \cup R_2$, and the other takes $R_1 = R$ and $R_2 = \emptyset$. Clearly, factoring is in TFNP, as each integer possesses a unique decomposition into primes, each with a certificate à la Pratt [13]. A related problem is that of the *discrete logarithm* modulo a (certified) prime p of a (certified) primitive root x of p. Notice that both of these problems are in fact both in the class TFNP and in the class FUP of unambiguous functions in NP [18] (the subset of FNP that consists of all those "multivalued" functions that are known to have either one solution or none). We should also mention that TFNP is somewhat related to the "one-way function" problem: The inverse of any one-way function is in TFNP *if we restrict our inputs to the range of the one-way function.* This latter condition makes this problem into a sort of "promise problem" and thus removes it from our scope.

Besides these problems, familiar to the complexity community from cryptography theory, TFNP contains many more problems, that do not belong in FUP, and are not known to be in FP. Each member Π_R of TFNP possesses a different kind of guarantee for the totality of R. For some, as with the class PLS [6] recalled below, the guarantee is very simple, based on the existence of local optima. For others, the guarantee is the consequence of some "polynomially nonconstructive" existence proof, such as Sperner's Lemma, Brouwer's Fixed Point Theorem [5, 4], or the convergence of Lemke's algorithm [10, 2] in certain cases. We list some of them below.

1.2. The class PLS

Local optimality is a very rich source of problems in TFNP. The class PLS of polynomial local search problems was introduced in [6]. Any problem in this class, is based on an optimization problem. The input to the optimization problem is a set of data (e.g., for the traveling salesman problem (TSP), the distance matrix). Given such an input x, we can always produce in polynomial time a *feasible solution* (for the TSP, say, the identity permutation of the cities). Also, using the input x, we can decide in polynomial time whether a string $y \in \Sigma^*$ is feasible (in the TSP, a tour), and calculate its integer cost (in the TSP example, the total length of the tour). Finally, we assume that we have defined a *polynomial neighborhood structure*, that is, for some polynomial p, given a feasible solution y and an integer $k \leqslant p(|x|)$, we can produce in polynomial time the *k-th neighbor of y*, a feasible solution z of x. The problem in PLS is this: given an input x, find a feasible solution which has cost no worse than any of its neighbors, that is, a *local optimum*.

It is immediate that all problems in PLS are in TFNP, as the existence of local optima is guaranteed by the finiteness of the solution space. In [6] several PLS problems were shown to be PLS-complete. There are now new, and perhaps more interesting, PLS-complete problems [9, 16], including finding a local optimum under the Lin–Kernighan local search heuristic [14].

As was pointed out in [6], there are problems in PLS that are not the result of compromising for a local instead of a global optimum, but in which any local optimum is the actual desired result. An example, due to Knuth, is the following: given an $m \times n$ matrix $\boldsymbol{A}$, $m < n$, we are looking for an $n \times n$ submatrix $\boldsymbol{B}$ such that $\boldsymbol{B}^{-1}\boldsymbol{A}$ contains elements with absolute values at most 1. This is tantamout to finding a submatrix which is a local maximum, with cost the absolute value of the determinant, if any two such submatrices are considered neighbors whenever they differ by one column.

1.3. Brouwer's fixed points

Brouwer's Theorem states that if D is homeomorphic to a simplex and $f: D \to D$ is continuous, then f has a fixed point, i.e., there exists an $x \in D$ such that $f(x) = x$. An interesting computational problem is as follows. Suppose f is continuous with Lipschitz constant 1, i.e., $|f(x) - f(y)| \leqslant |x - y|$, and let $\varepsilon > 0$ be given. Find an *ε-approximate fixed point*, i.e., an x such that $|f(x) - x| < \varepsilon$. It was shown in [4, 5] that any algorithm which uses f as an oracle takes $\Omega(1/\varepsilon)$ steps to calculate an ε-approximate fixed point.

Needless to say, Brouwer's Theorem guarantees that the following problem is in TFNP: given the description of a Turing machine which computes f and an $\varepsilon > 0$, find an ε-approximate fixed point. The results in [4, 5] do not provide an exponential lower bound for this problem, as an algorithm may examine the structure of the machine in generating a solution.

To guarantee that the Turing machine indeed computes a continuous function with Lipschitz constant 1, we can restrict our functions to such ones that interpolate between values computed at points whose coordinates are integer multiples of ε, and such that no two such values differ by more than ε. The difficulty of the problem is preserved (and the lower bounds of [4, 5] still hold).

1.4. Sperner's Lemma

For simplicity we discuss here only the two-dimensional case. Consider a triangle with vertices labeled 1, 2, 3, and any triangulation of its area (say, the standard $n \times n$ triangulation depicted in Fig. 1). Suppose we label the nodes of the triangulation by 1, 2 or 3, with the only restriction that 1 does not appear on any node on the edge $(2, 3)$ of the original triangle, 2 does not appear on $(1, 3)$, and 3 does not appear on $(1, 2)$. Sperner's Lemma states that there is always a triangle of the triangulation whose vertices are labeled 1, 2, 3. The proof of Sperner's Lemma is constructive, albeit by an algorithm that takes $O(n^2)$ steps. Sperner's Lemma can be used in turn to provide a constructive proof of Brouwer's Fixed Point Theorem (the original proof by Brouwer was nonconstructive in a way that he considered appalling). The associated computational problem, which we call TRICHROMATIC TRIANGLE, is as follows. Given a number n, the nodes of the triangulation correspond to triples (i, j, k) $(i, j, k \geqslant 0, i+j+k=n)$. Given the description of a polynomial-time Turing machine that assigns to any point (i, j, k) in the triangulation a label respecting the restriction on the sides (i.e., a *Sperner* labeling), find a trichromatic triangle (that is, one with all three labels). The problem is obviously in TFNP.

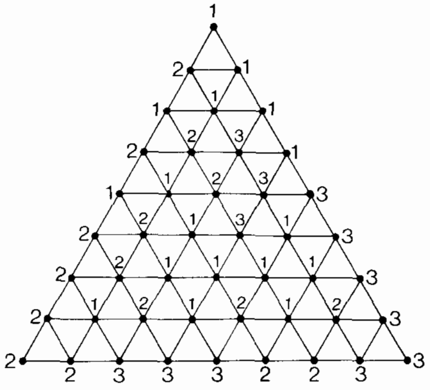

Fig. 1. Sperner's Lemma.

1.5. Bimatrix equilibrium points

A two-person game is played by two players, A and B. Player A has m (pure) strategies $1, \ldots, m$, and player B has n (pure) strategies $1, \ldots, n$. For each strategy i of player A and j of player B, we are given the *payoffs* A_{ij} and B_{ij} to A and B, respectively.

A *mixed* (or randomized) strategy for player A is a vector $\boldsymbol{x} = (x_1, \ldots, x_m)^T (\boldsymbol{x} \geqslant 0)$, $\sum x_i = 1)$ of probabilities assigned to the pure strategies of A. Similarly, a mixed strategy for B is a vector $\boldsymbol{y} = (y_1, \ldots, y_m)^T$ $(\boldsymbol{y} \geqslant 0, \sum y_j = 1)$. The expected payoffs to A and B, associated with a pair of mixed strategies $(\boldsymbol{x}, \boldsymbol{y})$, are $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}$ and $\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{y}$, respectively. A pair (x, y) is called a *Nash-equilibrium* if $x^{T}Ay \geqslant (x')^{T}Ay$ and $x^{T}By \geqslant x^{T}By'$ for all other mixed strategies x' and y'. Brouwer's Fixed Point Theorem implies Nash's celebrated result that every game has a mixed strategy equilibrium. This equilibrium is in fact a "basic feasible solution", see [10], and thus it has a rational number representation of acceptable length.[1]

In our opinion, an important open problem in complexity theory is whether there exists a polynomial-time algorithm for computing a mixed strategy equilibrium for a two-person game. The problem is obviously in TFNP.

1.6. Linear complementarity

Suppose that we are given an $n \times n$ integer matrix M and an integer n-vector q. We are asking for vectors x and y such that

$$Mx - y = q, \qquad x, y \geqslant 0, \quad x^{T}y = 0.$$

Notice that the latter condition requires that x and y be *complementary*, i.e., for every j, if $x_j \neq 0$, then $y_j = 0$. Hence, this problem is called the *Linear Complementarity Problem* (LCP). The LCP generalizes the convex quadratic programming problem (including the linear programming problem) and (0, 1)-programming. Thus, it is NP-complete in general [12]. However, it can be solved in polynomial time by the ellipsoid method if M is positive semidefinite [12].

There is an important case of the LCP whose complexity is unknown: Suppose that M is a P-matrix, i.e., all of its principal minors[2] are positive. In this case, the LCP is *guaranteed* to have a unique solution [15]. Moreover, this solution can be found in exponential time by Lemke's algorithm [10] or simply by a complete enumeration of bases; it follows that it too has a reasonably long rational representation. Other algorithms for this problem were recently developed in [7, 8], with complexity bounds which depend on certain condition numbers of the matrix M.

It is not known at present how to solve the LCP with a P-matrix in polynomial time. Neither is it known how to tell in subexponential time whether a matrix is a P-matrix.

The following problem (see [11]), which we call P-LCP, is in TFNP: given an instance M, q of the LCP, produce either a nonpositive principal minor of M *or* a solution of the LCP. It would be extremely interesting if the problem P-LCP were in FP.

This concludes our list of important and most intriguing problems that are in TFNP, and for which membership in FP would be very interesting, but is open. However, there are more examples one could list. An interesting graph-theoretic one was pointed out to us by S. Poljak. It is well-known (see [1]) that in any cubic graph, there is an even number of Hamiltonian circuits through each edge. The

[1] This result is true for any finite number of players. However, with more than two players we need a different model of computation to discuss the interesting computational question.

[2] A principal minor of M is a determinant of a submatrix M_{SS} of M obtained by deleting from M all the rows and columns with indices not in a certain subset $S \subseteq \{1, \ldots, n\}$.

proof is constructive, but goes through a possible exponential number of Hamiltonian paths to find the "mate" of a given Hamiltonian circuit. The interesting problem is this: given a cubic graph, an edge and a Hamiltonian circuit through the edge, find another Hamiltonian circuit through this edge.

We should point out here the difference between the problems we discuss in this note and some other problems where solutions are not only guaranteed to exist, but are *in abundance*, placing those problems in random polynomial time (example: concentrators with n nodes, n in unary). The existence of solutions in our problems above is not established by a probabilistic argument, and solutions are not, generally, in abundance.

2. On completeness

It is worth examining whether there are notions of completeness that may shed light on the complexity of these problems. A *reduction* from problem Π_R to problem Π_S is a pair of polynomially computable functions f and g such that, for any $x \in \Sigma^*$, $(x, g(y)) \in R$ iff $(f(x), y) \in S$. We call a problem complete for class FC if it is in FC, and all problems in FC reduce to it. As expected, we can preclude using the notion of NP-completeness for showing that such problems are intractable, as expressed in the following theorem.

Theorem 2.1. *There is an* FNP-*complete problem in* TFNP *if and only if* NP = coNP.

Proof. The "if" part is trivial: if NP = coNP, then any FNP-complete problem is in TFNP (recall that TFNP = F(NP ∩ coNP)). Suppose now that an FNP-complete problem Π_R is in TFNP, and that the functional version of SATISFIABILITY, denoted FSAT, reduces to Π_R via f and g. Then any unsatisfiable Boolean formula B has a certificate of unsatisfiability, namely, the string y (guaranteed to exist by the fact that $\Pi_R \in$ TFNP) such that $(f(B), y) \in R$ and $g(y) =$ "no". □

Several specialized cases of this argument have been used elsewhere [6, 11].

It is quite interesting, in view of Theorem 2.1, that in some cases of problems in TFNP, if together with the input we are also given one of the solutions that are guaranteed to exist, then the problem indeed becomes NP-complete. Consider, for example, the following variant of the computational problem associated with Sperner's Lemma (see Subsection 1.4 above), which we call SECOND TRICHROMATIC TRIANGLE: given an integer n, a polynomial-time Turing machine which computes a Sperner label for each vertex (i, j, k), and a trichromatic triangle, recognize whether there exists another trichromatic triangle.

Theorem 2.2. *The problem* SECOND TRICHROMATIC TRIANGLE *is* NP-*complete.*

Proof. We only sketch the proof here. Given a Boolean formula B with m variables, we construct a Sperner labeling of the triangle, where $n = 2^{m+4}$. It will be apparent

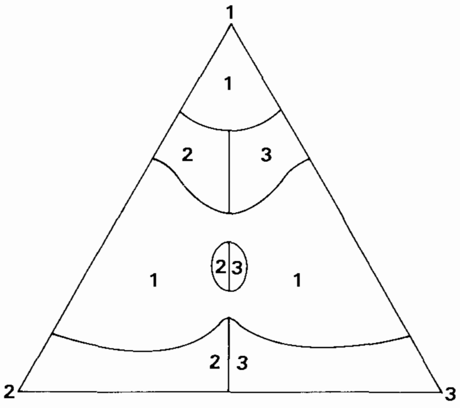

Fig. 2. The construction for Theorem 2.2.

that a polynomial-time Turing machine that assigns the appropriate labels can be constructed. The labeling is shown schematically in Fig. 2, in terms of the areas of the triangle labeled 1, 2, and 3. All "narrow areas" have width two units. There is an obvious point where the three colors meet, called x in the figure. All other trichromatic points will be of the form (i, j, j) (i.e., lying on the height of the triangle from vertex 1), such that *the binary description of* $i/16$ *or* $(i-8)/16$ *is a satisfying truth assignment.* □

A similar result can be shown for the variant of the Brouwer fixed-point problem, where we are asking for an approximate fixed point not in the neighborhood of the given one.

Related results with regard to the Nash-equilibrium problem can be found in [3].

Finally, can we hope to have TFNP-complete problems? As with other classes (such as NP∩coNP, R, BPP, etc.) whose machine-based definition is *semantic* instead of syntactic (i.e., depends on a property that the machine exhibits when computing on any input), we do not expect such problems to exist.

References

[1] C. Berge, *Graphs and Hypergraphs* (North-Holland, Amsterdam, 1973).

[2] R.W. Cottle and G.B. Dantzig, Complementary pivot theory of mathematical programming, *Linear Alg. Appl.* **1** (1968) 103-125.

[3] I. Gilboa and E. Zemel, Nash and correlated equilibria: Some complexity considerations, *Games and Economic Behavior* **1** (1989) 80-93.

[4] M. Hirsch, C.H. Papadimitriou and S. Vavasis, Exponential lower bounds for finding Brouwer fixed points," *J. Complexity*, to appear.

[5] M.D. Hirsch and S. Vavasis, Exponential lower bounds for finding Brouwer fixed points, in: *Proc. 28th Ann. IEEE Symp. on Foundations of Computer Science* (1987) (IEEE Computer Society Press, Los Angeles, 1987) 401-410.

[6] D.S. Johnson, C.H. Papadimitriou and M. Yannakakis, How easy is local search?," *J. Comput. System Sci.* **37** (1988) 79-100.

[7] M. Kojima, N. Megiddo and Y. Ye, An interior point potential reduction algorithm for the linear complementarity problem, *Mathematical Programming*, to appear.

[8] M. Kojima, N. Megiddo and T. Noma, Homotopy continuation methods for complementarity problems, *Mathematical Programming*, to appear.

[9] M. Krentel, Structure in locally optimal solutions, in: *Proc. FOCS 1989.*

[10] C.E. Lemke, Bimatrix equilibrium points and mathematical programming, *Management Sci.* **11** (1965) 681-689.

[11] N. Megiddo, A note on the complexity of P-matrix LCP and computing an equilibrium, Research Report RJ 6439, IBM Almaden Research Center, San Jose, CA 95120, 1988.

[12] K.G. Murty, *Linear Complementarity, Linear and Nonlinear Programming* (Heldermann Verlag, Berlin, 1988).

[13] V.R. Pratt, Every prime has a succinct certificate, *SIAM J. Comput.* **A** (1975) 214-220.

[14] C.H. Papadimitriou, The complexity of the Lin-Kernighan heuristic for the travelling salesman problem, submitted.

[15] H. Samelson, R.M. Thrall and O. Wesler, A partition theorem for Euclidean n-space, *Proc. Amer. Math. Soc.* **9** (1958) 805-807.

[16] A.A. Schäffer and M. Yannakakis, Simple local search problems that are hard to solve, *SIAM J. Comput.*, to appear.

[17] D. Solow, R. Stone and C.A. Tovey, Solving LCP on P-matrices is probably not NP-hard, Unpublished note, November 1987.

[18] L.G. Valiant, Relative complexity of checking and evaluating, *Inform. Process. Lett.* **5** (1976) 20-23.