Discussion Paper No. 455

ON SEARCH OVER RATIONALS

by

Eitan Zemel
Northwestern University

December 1980

Supported, in part, by NSF Grant Eng. 7902896 and by the J. L. Kellogg Center for Advanced Studies in Managerial Economics.

ABSTRACT

Papadimitriou and Reiss have independently shown how a rational $x = p/q$, p,q positive integers with $p, q \leq N$ can be found in $O(\log N)$ queries of the form is $x \leq y$? We examine some of the algorithmic implications of their findings.

I. Introduction.

It is sometimes possible to solve a mathematical programming problem of the form:

$$s^* = \max f(X)$$

$$\text{(P)} \qquad \text{subject to } X \varepsilon F \quad R^n$$

using the following strategy. First, a finite set S, which is known to contain the optimal objective value s^*, is identified. Then, one searches for s^* in S using a test whose role is to decide for each $s \varepsilon S$, whether or not $s^* \leqslant s$. This approach is especially viable in cases where an efficient subroutine for performing the test is available and where the set S is easily generated and is rather small or contains enough structure to allow efficient search within it. Examples of problems which where successfully solved using this technique include the maximization of ratio functions, [3], [10], [12], [16], minimax location problems on trees, [4], [5], [9], [17], the linear knapsack and multiple choice knapsack problems, [1], [2], [7] [8], [18], etc.

The cardinality of the set S plays a crucial role in the overall complexity of such algorithms. Thus, there is a natural tendency towards preferring smaller sets to larger ones. In fact, the polynomial nature of several algorithms of this type depend crucially on the fact that an ingeneous reasoning allows one to consider a set S whose cardinality is polynomial (e.g. [13]). Naturally, in order to obtain a small set S, one usually exploits to the maximum degree possible the specific details of the problem at hand. Thus, there is often a close relationship between the structure of the problem (P) and the structure of the set S. For instance, in [4], [5], [9], [14], [17], one uses a set S, which is derived from the set of inter-nodal distances on a tree. Much of the efficiency of the algorithm depends on this particular structure.

The strategy considered in this note is the opposite of the one described in the paragraph above. Rather than search within a small and problem specific set S, we consider a set $R \supseteq S$, which is typically much larger than S. Specifically, we consider sets of the form

$$R_N = \{x/y : x \geq 0,\ y > 0,\ x,y \leq N,\ x,y \text{ integer}\}$$

where N is sufficiently large to ensure that $R_N \supseteq S$. As will be demonstrated below, such a strategy allows one to derive polynomial algorithms for several problems (P) for which no such algorithms existed before. For other problems, the method can be used to attain a better bound on the computational complexity. As a bonus, we typically also get a very significant simplification in the structure of the algorithms for (P).

At the heart of the proposed strategy is the following interesting theorem, due independently to Papadimitriou, [15], and to Reiss [16]:

Theorem 1, [15], [16]: Let $s^* \epsilon R_N$ be given, but unknown. Then, s^* can be identified within $O(\log N)$ querries of the form is $s \geq s^*$?

Papadimitriou's method is based on Farey's series; that of Reiss on continued fractions. We note that both methods, but in particular the one of Reiss, [16], lent themselves easily to practical implementation.

In the following two sections we examine two cases where such a strategy seems advantageous. The first is the problem of minimization of a ratio function over a combinatorial set, the second is the weighted min max location problem on a tree.

II. Minimization of Ratio Functions

Consider the problem:

(PR) $$s^* = \min\ (c_o + cx)/(d_o + dx) \quad \text{subject to } x \ \epsilon\ F$$

where the coefficients appearing in the objective function are integers and where $d_0 + dx > 0$, $x \ \epsilon\ F$. For ease of exposition, we assume throughout this section that F is a set of 0-1 vectors in R^n. However, the results remain virtually unchanged for any bounded Polyhedral set with rational extreme points whose numerator and denominator can be bounded apriori. Consider the Linear version of (PR):

(PL) $$\max\ cx \quad \text{subject to } x \ \epsilon\ F$$

where c is a vector of integers. Obviously, any algorithm for (PR) can solve (PL) as well. A connection in the reverse direction is established by the following observation whose proof is trivial:

Proposition 1 Let F be such that $d_o + d_x > 0$ for every $x \in F$. For a scalar λ let:

$$Z(\lambda) = \max\ (c - \lambda d)X \quad \text{subject to } x \in F$$

then: $$\lambda \geq s^* \text{ iff } Z(\lambda) \geq c_o - \lambda d_o.$$

This approach was utilized by Chandrasekaran, [3], for the minimal ratio spanning tree problems and by Karp, [10], and Lawler, [12], for the minimal ratio cycle problem. More generally, we have the following interesting result due to Megiddo, [13]:

Theorem 2 [13]: If problem (PL) can be solved within O(p(n)) comprisons and O(q(n)) additions than problem (PR) can be solved in time O(p(n)(q(n) + p(n))).

Megiddo's result is based on the strategy referred to in the introduction where the algorithm for (PL) serves a guide for a search through S. The main point in his result is a clever way to identify an appropriate set S whose cordinality is bounded by O(p(n)). Obviously, if p(•) and q(•) are polynomially bounded functions than so is the complexity of the algorithm for (PR). Nevertheless, we note that theorem 2 falls short of asserting that (PR) is solved in polynomial time if (PL) is. This is due to the limitation on the type of operations allowed by the algorithm for (PL), namely additions and comparisons only. Although these limitations can be somewhat relaxed, they cannot be entirely removed. Thus, a polynomial algorithm for (PL) which is based, say, on multiplications of cost coefficients, cannot be used to support a polynomially bounded algorithm for (PR). Examples of problems for which the only known polynomial algorithms are of this type will be given below. But, first we use Theorem 1 to close the gap left open by Theorem 2:

Theorem 3 (PR) is solvable in polynomial time iff (PL) is.

Proof The only if part is trivial since (PL) is a special case of (PR). For the other direction, let:

$$d = \max_{i = 0 \ldots n} \max \{|c_i|, |d_i|\}$$

and let:

$$s^* = (c_0 + cx^*)/(d_0 + dx^*) \equiv p/q$$

Obviously, p,q are integers whose absolute value is bounded by, say, N = 2nd. Thus, we can identify s* within R_N using O(logN) = O(log(n • d)) tests, each involving a run of the algorithm for (PL). To conclude the proof we note that the cost entries in each of these runs (expressed as integers) are bounded from above by Nd = $O(n^2d^2)$.

We now apply theorem 3 to several ratio problems which do not fall within the scope of theorem 2. All the examples given are consequences of the extremely powerful technique developed in a recent paper by Gröetchel, Lovász and Schrijver, [6]. The method, which is based on Khachians polynomial algorithm for Linear Programming [11], is used in [6] to construct polynomial algorithms for numerous combinatorial programming problems. The first two examples are based on algorithms for (PL) taken from [6]; the third is taken from [19].

Example I Let G be a perfect graph. It is shown in [6] how to find, in polynomial time, a subset of vertices of G which is independent (i.e. no two vertices in the subset are connected by an edge) and which maximizes a linear objective function. Thus, we can solve in polynomial time the problem:

$$s^* = \max (c_0 + cx)/(d_0 + dx)$$

where x is the incidence vector of an independent set in G, and where c, d are vectors of integers with $d_0 + dx > 0$ for every feasible x.

Example II Let F be a family of subsets of{1...n}, which is closed under unions and intersections, and let f be an integral valued supermodular function over F. Then, it is shown in [6] how max f(x) over x ε F can be found in polynomial time (subject to some mild assumptions on the way F is given). Using Theorem 3, we can solve, in polynomial time, the problem:

$$\max (f_0 + f(x)/(g_0 + g(x))$$

for g(x) integral valued submodular function over F and such that $g_0 + g(x) > 0$, x ε F.

Example III Let G be an undirected graph whose arcs are subject to probabilistic failure. Denote the probability that arc i will be operative throughout a certain time interval by p_i. Assume we are given only the information:

$$a_i \leqslant p_i \leqslant b_i$$

where i runs over all the edges of G. It is shown in [19] how we can calculate in polynomial time the best possible upper and lower bound on the probability of events of the type:

(E_1): every two nodes of G are connected by a path.

(E_2): two specific nodes of G, say s and t, are connected by a path.

etc. Using theorem 3, we can calculate, in polynomial time, the best bounds on the conditional probability of E_1 given E_2, provided that the probability of the later does not vanish. Similar results can be obtained for directed graphs and for other types of conditional events.

In spite of examples I - III, we note that almost all polynomial algorithms available to date fall within the stipulations of Theorem 2. For such problems, one is naturally interested in comparing the algorithm of this Theorem versus the one of Theorem 3. We note at the outset one advantage of the later algorithm. The set R_N used by this algorithm is a standard one and does not depend on the problem (PL) (except the obvious dependence on the parameter N). Thus, we can use a general master problem for (PR) which accepts any algorithm for (PL) as a subroutine. Moreover, this master program is extremely simple requiring little more than a binary search over a continuous region, [16]. This is in sharp contrast to the algorithm of [13], where the structure of the problems (PL) bears close relation to S so that the master problem for (PL) is problem specific. As for the computational complexity of the two methods, they, of course, depend on the specific problem one is considering. In table I below, we examine several typical examples. In the first two, an accelerated version of Megiddo's algorithm is used, as in [13]. The other two are not specifically analyzed in [13], so the complexity is given by the expression given in theorem 2.

The examples considered are the following:

(i) The minimum ratio cycle problem.

(ii) The minimum ratio spanning tree problem.

(iii) The maximum ratio matching problem.

(iv) The minimum ratio flow problem.

For all four problems, let m and n denote the cardinality of the set of edges and nodes of G respectively, and let d denote an upper bound on the coefficients appearing in the objective row, expressed as integers. For the fourth problem we let u denote the maximal capacity of an arc.

Problem	Algorithms of Theorem 2	Algorithms of Theorem 3
(i)	$0\ (m\ n^2 \log(n))$	$0\ (mn \log(nd))$
(ii)	$0\ (m \log^2(n) \log\log(n))$	$0\ (m \log(nd) \log\log(n))$
(iii)	$0\ (n^6)$	$0\ (n^3 \log(dn))$
(iv)	$0\ (m^4 \log^2(u))$	$0\ (m^2 \log(u) \log(d \cdot m))$

TABLE I

Naturally, the difference between the bounds implied by theorems 2 and 3 increases with the complexity of (PL) and decreases with log d. Thus, in general, the algorithm based on Theorem (3) is preferable for problems where the linear version is of high complexity and where the coefficients in the objective row are not extraordinarilly large.

III. The Weighted P center problem on a Tree

We consider, in this section, another problem, for which theorem 1 provides a polynomial algorithm. However, this problem is not of the ratio type problem discussed in the previous section. Consider a tree network T = (V,E) which is embedded in the Euclidean plane so that edges correspond to line segments meeting each other at the nodes. Let A be the union of all the edges of T, each considered as the subset of points which make out the respective line segment. For each pair of points $x \in A$, $y \in A$, let $d(x,y)$ denote the distance between them, measured along the edges of T.

Let $\Sigma = \bigcup_{i=1}^{k} \Sigma_i$, and $\Delta = \bigcup_{j=1}^{r} \Delta_j$ be two subsets of A. We assume that each set Σ_i, Δ_j corresponds to a closed and connected subtree of T, possibly containing just one point. Without loss of generality, we can assume that all the end points of the sets Σ_i, $i=1\ldots k$, Δ_j, $j=1\ldots r$ are considered vertices of T, i.e., are members of V. Denote by n the cardinality of this set.

We can think of the set Σ as the set of prospective supply points on T. Similarly, we refer to Δ as the demand set. For each individual demand region, Δ_j, $j = 1\ldots r$, let w_j be a positive integer which represents the "importance" or "weight" of this set. Similarly, for each supply region, Σ_i, $i = 1\ldots k$, let v_j be a positive integer which is inversely related to the "speed" of a server dispatched from this region. For every point $x \varepsilon \Sigma$, let $v_x = v_i$ where i is the (unique) index such that $x \varepsilon \Sigma_i$. Similarly, for $y \varepsilon \Delta$, let $w_y = w_j$ for j such that $y \varepsilon \Delta_j$.

Consider a "customer" (demand point) $y \varepsilon \Delta$ which is being "serviced" by a facility established at a point $x \varepsilon \Sigma$. Let

$$s(x, y) = v_x\, w_y d(x, y)$$

be the objective function, as viewed from the perspective of this customer. Naturally, if we establish centers at locations $S = \{x_1,\ldots,x_p\}$, the one most preferred by y is the point $x^* \varepsilon S$ such that

$$s(x^*,y) = \min_{x \varepsilon S} s(x, y)$$

Thus, the weighted p - Center Problem (WPCP) can be defined as the problem of finding a set $S = \{x_1,\ldots,x_p\}$, $x_i \varepsilon \Sigma$, $i = 1,\ldots,p$ which minimizes the expression:

$$\max_{y \varepsilon \Delta} \; \min_{x \varepsilon S} \; s(x, y).$$

We say that (WPCP) is <u>discrete</u> if both Σ and Δ are finite sets i.e. if $\Sigma \cup \Delta \subseteq V$. We say that the problem is <u>semi-discrete</u> if at least one of the sets is finite. A problem which is not semi-discrete will be referred to as <u>continuous.</u> Also, a problem will be referred to as <u>semi-weighted</u> if only one of the sets Σ or Δ is weighted i.e. if $v_x = 1$, $x \varepsilon \Sigma$ or $w_y = 1$, $y \varepsilon \Delta$. The problem is <u>un-weighted</u> if neither set is weighted. We will refer to the unweighted version of (WPCP) as (PCP).

It is well known that (WPCP) defined on a general graph is np-hard even for the unweighted case and even when the sets Σ and Δ are simple, [12]. On the other hand, the unweighted problem on a tree is well behaved. Polynomial algorithms for this problem are given in [3], [4], [5], [9], [14], [17]. The current best bound for the discrete and semidiscrete case is $O(n \log(n))$, [5]. Putting together the results of [5] and [14] we also get an $O(nq \log(pn/q^2))$ algorithm for the continuous problem where $q = \min\{n, p\}$. The weighted case of (WPCP) is not treated in the literature, except for few special cases [9], [14].

All the above mentioned algorithms for (PCP) are based on the strategy presented in the introduction. First, a set S is constructed which is intimately related to the set of internodal distances on T. In addition, a test is designed, [4], [9], [17], whose role is to decide, for each $r \in S$, whether or not $s \geq s^*$. In all cases, the test requires $O(n)$ steps and the cardinality of the set S is polynomially bounded. Thus, the overall algorithm for (PCP) is plynomial.

In order to handle the weighted case using theorem 1 we need to establish an appropriate set R_N which contains s^* and to establish a test routine which can guide the search over R_N. These two issues are handled in the following two propositions given here without proofs. The first is a generalization of proposition 1 in [14] and can be proved by the same methods. The second, which is applicape to the semiweighted cases only, can be obtained by combining the ideas of [4], and [9] and [17]. The question of whether a similar polynomially bounded test can be devised for the fully weighted case is open.

Proposition 2 The optimal value to (WPCP), is contained in the set

$$S_t = \left\{ d(x, y) / \sum_{i=1}^{k} \frac{1}{w_{x_i} \cdot v_{y_i}} \right\} \; x, y \in V, \; k \leq t, \; x_i \in \Sigma, \; y_i \in \Delta, \; i=1,\ldots,k.$$

where

$$t = \begin{cases} 1 & \text{if the problem is discrete} \\ 2 & \text{if the problem is semi discrete} \\ 2p & \text{if the problem is continuous} \end{cases}$$

Proposition 3. If (WPCP) is semiweighted i.e. if $v_x = 1, x \in \Sigma$ or $W_y = 1, y \in \Delta$) there exists an $O(n)$ procedure for deciding whether a given scalar s satisfies $s \geq s^*$.

Let u be an upper bound on the weights appearing in (WPCP) and let d be an upper bound on a length of an arch in T. It follows from Proposition 3 that $s^* \in R_N$ for $N = ndu^{2t}$. Thus, we can solve the semiweighted (WPCP) in $O(n \log(ndu))$ steps in the discrete or semidiscrete use and in $O(np \log(ndu))$ steps in the continuous case.

REFERENCES

[1] Balas E, and E. Zemel "An Algorithm for large Knapsack Problems" Opns Res 28 pp. 1130-1155, 1980.

[2] Bentley, J. L. and M. I. Shamos "A Linear Time Weighted Median Algorithm and Applications" Annouced in Shamos, M. I. "Geometry and Statistics" in Algorithms and Complexity: New Directions and Recent Results" J. F. Traub (Ed.), Academic Press, 1976.

[3] Chandrasebaran, R. "Minimum Ratio Spanning Trees" Networks Z. pp. 335-342, 1977

[4] Chandrasebaran, R. and A. Tamir "An $0(n \log p)^2$ Algorithm for the continuous p-center Problem on a Tree". Discussion Paper No. 367, The Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1978.

[5] Frederickson, G. N. and D. B. Johnson, "Generating and Searching Sets Induced by Networks" to appear. Proc 1980 ICALP Conf.

[6] Groetschel, M. L. Lovasz and A. Schrijver, "The Elipsoid Method and its Consequences in Combinational Optimization" Report No. 80151-OR, Institut fur Oronometrie and Operations Research, Universitat Bonn W. Germany, 1980.

[7] Ibaraki, T. "Approximate Algorithms for the Multiple Choice Continuous Knapsack Problems", Department of Applied Mathematics and Physics, Kyoto University, Japan, 1978.

[8] Johnson, D. B. and T. Mizoguchi" Selecting th K-th Element in X + Y and $x_1 + x_2 + \ldots x_m$" SIAM. J. on Computing Vol. 7 No. 2 1978.

[9] Kariv, O. and S. L. Hakimi "An algorithmic Approach to Network Location Problems; Part I: P-Centers" SIAM. J. Appl. Math. 37 pp. 513-538, 1979.

[10] Karp, R. M. "A Characterization of the Minimum Cycle Mean in a Digraph" Memorandum No. UCB/ERL/47, Electronic Research Laboratory. College of Engineering, University of California at Berkeley, 1977.

[11] Khachian, L. G. "A polynomial Algorithm In Linear Programming" English Translation Soviet Math Dobl. 20 pp. 191-194, 1979.

[12] Lawler, E. L. "Combinational Optimization: Networks and Matroids" Holt, Rinehart and Winston, New York, 1976.

[13] Megiddo, N. "Combinational Optmization With Rational Objective Functions" Math. of O.R. 4 pp. 414-424, 1979.

[14] Megiddo, N., A. Tamir, E. Zemel and R. Chandrasekaran "An $0(n\log^2 n)$ Algorithm for the k-th Longest Path in a Tree With Applications to Location Problems" To Appear in SIAM. J. On Computing.

[15] Papadimitriou, C. H. "Efficient Search for Rationals" Inf. Proc. let 8 pp. 1-4, 1979.

[16] Reiss, S. P. "Rational Search" Inf. Proc. let 8 pp. 89-90, 1979.

[17] A. Tamir and E. Zemel "Locating Centers on a Tree With Discontinuous Supply and Demand Regions" Discussion Paper No. 397, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1979.

[18] Zemel, E. "The Linear Multiple Choice Knapsack Problem" To appear in Opns. Res.

[19] Zemel, E. "Polynomial Algorithms for Estimating Network Reliability" Discussion Paper No. 449. The Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1980.