Int. J. Solids Structures Vol. 18, No. 6, pp. 487–495, 1982
Printed in Great Britain.

0020-7683/82/060487-09$03.00/0

THE COMPLEXITY OF PROCEDURES FOR DETERMINING MINIMUM WEIGHT TRUSSES WITH DISCRETE MEMBER SIZES

D. F. YATES,† A. B. TEMPLEMAN† and T. B. BOFFEY‡
University of Liverpool, Liverpool, England

(*Received* 29 *April* 1981)

Abstract—This paper addresses the problem of minimising the weight of a structural truss subject to static constraints and member sizes only being available from a discrete set of commercially available gauges. Substantial theoretical evidence is advanced to suggest that, given current mathematical techniques, it is not possible to solve this problem exactly in all cases. More specifically, the problem under consideration, even in simple cases, is shown to be equivalent to a large number of other problems that are notorious for their computational intractability. In practical terms, such a result justifies the development of methods that approximately solve the problem and consequently, algorithms for solving it to within specified absolute accuracy are considered. It is subsequently shown that such methods are equivalent in complexity to those for solving the original problem.

1. INTRODUCTION

Structural trusses are employed in numerous areas ranging from bridge construction to applications in the aerospace industry. As the weight of a truss is usually related to its cost or is itself of intrinsic importance, the problem of optimising weight subject to the constraints of the particular application is of considerable interest. Bounds on joint deflection, member stress and gauge are frequently encountered and it is the problem of truss-weight minimisation subject to these three types of constraint to which this paper is addressed. Numerous methods have been proposed to tackle this problem both for member sizes chosen from a continuum and from a discrete set of commercially available gauges. Details of many of these can be found, for example in Khot[1] or Fleury and Schmit[2]. All the proposed methods appear to work well on some problems in that they generate an exact or sufficiently good approximate solution, however, none of them claims to work well on all problems. Consequently, it is held that the general problem is "difficult" to solve in the sense of guaranteeing an acceptably close approximate solution using only a "reasonable" amount of computing time. The results of this paper, which deals with the situation where member sizes are available from a discrete set, serve to provide, in this case, substantial theoretical justification for that point of view.

In order to avoid any difficulties that may arise over the terminology used, the following definitions are now introduced. The terminology adopted follows closely that used by Horowitz and Sahni[3], which in turn is based on the standard suggested by Knuth[4].

Definition 1

A *deterministic algorithm* is one in which the result of every operation is uniquely defined.

Definition 2

A *non-deterministic* algorithm is one in which the result of every operation is not uniquely defined but is one of a specified set of possibilities. (This implies that the algorithm will for certain operations have to "choose" an outcome, however, it may be considered that the algorithm evaluates all possible outcomes simultaneously and chooses a correct one, if such exists.)

Definition 3

A *decision problem* is a problem to which there is one of two possible solutions corresponding to the boolean values "true" and "false".

†Department of Civil Engineering.
‡Department of Computational Statistical Science.

Definition 4

A *polynomial-time algorithm* is one whose running time (the number of elementary bit operations performed on an input string of length N) is bounded above by some polynomial $p(N)$. The class of decision problems that can be solved by such an algorithm is denoted by P.

Definition 5

NP is defined to be the class of decision problems that can be solved by a non-deterministic algorithm that runs in polynomial time.

Definition 6

A problem L_1, reduces to another problem L_2, or $L_1 \alpha L_2$, if and only if any instance of L_1 can be solved by a deterministic algorithm for solving L_2 in polynomial time.

Definition 7

A problem is said to be *NP-hard* if every problem in **NP** reduces to it; it is *NP-complete* if it also lies in **NP**.

It is widely held that a problem is not well-solved until a polynomial-time algorithm for it has been devised and correspondingly a problem is considered to be intractable if there is no polynomial-time algorithm that will solve it. An NP-hard problem may be thought of as being at least as difficult to solve as any problem in **NP** and consequently, proving the intractability of one *NP*-complete problem would render all *NP*-hard problems intractable. As yet, however, no *NP*-complete problem has been well-solved or proved to be intractable. Nonetheless, there is considerable evidence to support the conjecture that *NP*-hard problems are intractable in that the class of *NP*-complete problems contains a wealth of problems from the fields of Optimisation, Graph Theory, etc. (see Garey and Johnson [5] for a relatively up-to-date compilation), and over the years each has been the subject of investigation by many researchers, but no polynomial-time algorithm has been devised for even one of them. Irrespective of the truth of the conjecture, it is generally agreed that, at worst, *NP*-hard problems will remain perpetually intractable and, at best, *NP*-complete problems will only be well-solved by a major advance in mathematical techniques.

In this paper it is proved that the problem under consideration is a member of the class of *NP*-hard problems. As a half-way stage, the simpler problem of truss-weight minimisation subject only to deflection constraints is examined and it is shown that even without the added complications of stress and gauge constraints, the problem is *NP*-hard. Such results theoretically justify the development of algorithms that approximately solve these problems—an expedient that is almost universally adopted in practive. Correspondingly, algorithms that guarantee an absolute bound on the error are considered and it is then shown that the problem of devising such an algorithm is also *NP*-hard.

1.1 *Truss weight minimisation with deflection constraints*

Let D be a structural truss with M members and J joints, then the weight of D is taken to be:

$$W(\mathbf{A}) = \sum_{i=1}^{M} \rho_i l_i A_i \tag{1.1}$$

where ρ_i is the mass density and l_i the length of member i which has cross-sectional area A_i, $i = 1, 2, \ldots, M$. The minimum weight problem considered requires minimisation of $W(\mathbf{A})$ subject to:

$$\left.\begin{array}{ll} \delta_j \leq \delta_j\dagger & j = 1, 2, \ldots, J \\ \text{and} \quad A_i \in S & i = 1, 2, \ldots, M. \end{array}\right\} \tag{1.2}$$

Here δ_j is the deflection of joint j in direction θ_j due to a loading F and S is a finite subset of R^+. Only a single load case is considered since multiple-load cases do not alter the nature of the problem but do complicate notation. Without loss of generality, the set S of available member sizes is taken to be well ordered, i.e. $i < j$ implies $s_i < s_j$ and vice-versa, $\forall\, i, j$.

Using the Principle of Virtual Work and the substitutions:

$$\gamma_i = \rho_i l_i,$$

$$\beta_{ij} = \frac{l_i T_i T'_{ij}}{E}$$

where T_i is the force in member i due to loading F, T'_{ij} the virtual force in member i due to unit load applied in direction θ_j at joint j and E is Youngs' modulus, the minimisation of (1.1) subject to (1.2) can be transformed to:

$$\left.\begin{aligned} &\underset{\mathbf{A}}{\text{minimise}}\ w(\mathbf{A}) = \sum_{i=1}^{M} \gamma_i A_i \\ &\text{subject to:} \\ &\sum_{i=1}^{M} \frac{\beta_{ij}}{A_i} \le \delta_j\dagger \quad j = 1, 2, \ldots, J \\ &A_i \in S \quad i = 1, 2, \ldots, M. \end{aligned}\right\} \tag{1.3}$$

For convenience, problem (1.3) will be referred to as DMTD (Discrete member-size Minimum weight Truss problem with Deflection constraints). In the succeeding section it is shown that DMTD is *NP*-hard.

2. TO SHOW THAT DMTD IS *NP*-HARD

In order to show that a given problem, L_2 say, is *NP*-hard, it is sufficient to select a problem, L_1, already known to be *NP*-hard and show that L_1 reduces to L_2, i.e. $L_1 \alpha L_2$. Consequently, it must be shown how to obtain an instance of L_2 from any instance of L_1 in polynomial deterministic time in such a way that the solution of the instance of L_1 can be determined from the solution of the instance of L_2 in polynomial deterministic time. Having done this, it may be readily concluded that since the relation is transitive, L_2 is *NP*-hard. (Details of the transitivity of α can be found, e.g. in Karp[6], and examples of using this technique in Horowitz and Sahni[3]).

For current purposes the problem selected as being known to be *NP*-hard is the 0–1 Knapsack problem (KP) (see Horowitz and Sahni[3]). It is stated here as:

$$\left.\begin{aligned} &\underset{\mathbf{Y}}{\text{maximise}}\quad G(\mathbf{Y}) = \sum_{i=1}^{M} a_i Y_i \\ &\text{subject to:} \\ &\sum_{i=1}^{M} b_i Y_i \le R \\ &Y_i = 0 \text{ or } 1 \qquad i = 1, 2, \ldots, M \\ &a_i \in Z^+, b_i \in Z^+ \quad \forall i. \end{aligned}\right\} \tag{2.1}$$

It is noted that some texts quote alternative formulations of the 0–1 Knapsack problem, however, no difficulties arise as all of the alternatives are related to (2.1) via α.

From the instance (2.1) of KP define the instance of DMTD by:

$$\left.\begin{aligned} &\underset{\mathbf{A}}{\text{minimise}}\ w(\mathbf{A}) = \sum_{i=1}^{M} \gamma_i A_i \\ &\text{subject to:} \\ &\sum_{i=1}^{M} \frac{\beta_i}{A_i} \le C \\ &A_i = A^{(1)} \text{ or } A^{(2)} \quad i = 1, 2, \ldots, M \\ &A^{(1)} < A^{(2)} \end{aligned}\right\} \tag{2.2}$$

where

$$\left.\begin{aligned} \gamma_i &= a_i \\ \beta_i &= b_i Q(Q+1) \end{aligned}\right\} \quad i = 1, 2, \ldots, M$$

$$C = 1$$

$$A^{(1)} = (R + \sigma Q)Q$$

$$A^{(2)} = (R + \sigma Q)(Q+1)$$

$$\sigma = \sum_{i=1}^{M} b_i$$

and

$$Q \in Z^+.$$

Let I denote the instance (2.1) of KP and I' the instance (2.2) of DMTD, then:

Result 1

$\tilde{\mathbf{Y}}$ is an optimal solution to I if and only if $\tilde{\mathbf{A}}$ is an optimal solution to I' where:

$$\tilde{Y}_i = \begin{cases} 1 & \text{if} \quad \tilde{A}_i = A^{(1)} \\ 0 & \text{if} \quad \tilde{A}_i = A^{(2)}. \end{cases}$$

Proof (*Necessity*)

Suppose $\mathbf{Y}\dagger$ is a feasible solution of I where:

$$Y_i\dagger = \begin{cases} 1 & i \in V \\ 0 & i \in V_c \end{cases}$$

then

$$\sum_{i \in V} b_i \le R$$

whence

$$\sum_{i \in V} b_i(Q+1) + \sum_{j \in V_c} b_j Q \le R + Q \sum_{k=1}^{M} b_k$$

$$= R + \sigma Q$$

or

$$\sum_{i\in V}\frac{b_i(Q+1)}{(R+\sigma Q)}+\sum_{j\in V_c}\frac{b_jQ}{(R+\sigma Q)}\leq 1$$

or

$$\sum_{i\in V}\frac{\beta_i}{A^{(1)}}+\sum_{j\in V_c}\frac{\beta_j}{A^{(2)}}\leq 1.$$

Thus if Y† is a vector that satisfies the constraints of I, then the vector A† defined by:

$$A_i\dagger=\begin{cases}A^{(1)} & i\in V\\ A^{(2)} & i\in V_c\end{cases}$$

satisfies the constraints of I'.

Let $\tilde{\mathbf{Y}}$ be an optimal solution to I and $\mathbf{Y}^*$ any other feasible but suboptimal vector, then:

$$\sum_{i=1}^{M}a_iY_i^*<\sum_{i=1}^{M}a_i\tilde{Y}_i$$

and thus

$$\sum_{k\in V}a_kY_k^\dagger=\sum_{k\in V}a_k<\sum_{i\in T}a_i\tilde{Y}_i=\sum_{i\in T}a_i$$

where $\mathbf{Y}^*$ is defined by the partition V, V_c, and $\tilde{\mathbf{Y}}$ by the partition T, T_c.

Now since

$$\sum_{k\in V}a_k+\sum_{l\in V_c}a_l=\sum_{i\in T}a_i+\sum_{j\in T_c}a_j=\sum_{i=1}^{M}a_i$$

then

$$\sum_{l\in V_c}a_l>\sum_{j\in T_c}a_j$$

and thus

$$\sum_{l\in V_c}a_l(A^{(2)}-A^{(1)})>\sum_{j\in T_c}a_j(A^{(2)}-A^{(1)}).$$

Hence:

$$\sum_{k=1}^{M}a_kA^{(1)}+\sum_{l\in V_c}a_l(A^{(2)}-A^{(1)})>\sum_{i=1}^{M}a_iA^{(1)}+\sum_{j\in T_c}a_j(A^{(2)}-A^{(1)})$$

and therefore

$$\sum_{k\in V}a_kA^{(1)}+\sum_{l\in V_c}a_lA^{(2)}>\sum_{i\in T}a_iA^{(1)}+\sum_{j\in T_c}a_jA^{(2)}$$

i.e.

$$\sum_{i=1}^{M}a_iA_i^\dagger>\sum_{i=1}^{M}a_i\tilde{A}_i$$

whence

$$\sum_{i=1}^{M} \gamma_i A_i^* > \sum_{i=1}^{M} \gamma_i \tilde{A}_i.$$

Hence necessity is proven.

Sufficiency can likewise be proved by noting that all implications are reversable.

Theorem 1

DMTD is *NP*-hard.

Proof

Since Result 1 holds and instance I' of DMTD can be obtained in polynomial deterministic time from instance I of KP, then KPαDMTD. Further, since KP is *NP*-hard, then DMTD is *NP*-hard also.

3. TRUSS WEIGHT MINIMISATION WITH GAUGE, STRESS AND DEFLECTION CONSTRAINTS

In considering the problem of minimising the weight of a truss subject to these three types of constraint it is only necessary to examine the problem defined by (1.1) and (1.2) subject to the additional constraints:

$$\left.\begin{array}{l} A_i \geq A_i\dagger \\ \sigma_i \leq \sigma_i\dagger \end{array}\right\} \quad i = 1, 2, \ldots, M$$

where σ_i is the stress in member i due to the external loading F.

As before the problem can be transformed, in addition to the manipulations detailed in Section 1.1, by the substitutions:

$$\bar{A}_i = \max\left(A_i\dagger, \frac{l_i T_i}{E\sigma_i\dagger}\right),$$

and

$$A_i^* = \min_{s_k \geq \bar{A}_i} (s_k)$$

to the problem:

$$\left.\begin{array}{l} \underset{\mathbf{A}}{\text{minimise}}\ w(\mathbf{A}) = \sum_{i=1}^{M} \gamma_i A_i \\ \text{subject to:} \\ \sum_{i=1}^{M} \frac{\beta_{ij}}{A_i} \leq \delta_j\dagger \quad j = 1, 2, \ldots, J \\ \left.\begin{array}{l} A_i \geq A_i^* \in S \\ A_i \in S \end{array}\right\} \quad i = 1, 2, \ldots, M. \end{array}\right\} \tag{3.1}$$

For the sake of convenience, this problem will be referred to as DMT (Discrete member-size Minimum weight Truss problem). It is now a simple task to show that DMT is *NP*-hard.

As before KP is used as the problem known to be *NP*-hard and (2.1) taken as the instance of KP. Define an instance of DMT by:

$$\underset{\mathbf{A}}{\text{minimise}}\ w(\mathbf{A}) = \sum_{i=1}^{M} \gamma_i A_i$$

subject to:

$$\sum_{i=1}^{M} \frac{\beta_i}{A_i} \leq C \tag{3.2}$$

$$\left.\begin{matrix} A_i \geq A_i^{\dagger} \\ A_i \in S \end{matrix}\right\} \quad i = 1, 2, \ldots, M$$

where

$$\left.\begin{matrix} \gamma_i = a_i \\ \beta_i = b_i Q(Q+1) \end{matrix}\right\} \quad i = 1, 2, \ldots, M$$

$$C = 1$$

$$S = \{A^{(0)}, A^{(1)}, A^{(2)}\}$$

$$A^{(0)} = (R + \sigma Q)(Q - 1)$$

$$A^{(1)} = (R + \sigma Q)Q$$

$$A^{(2)} = (R + \sigma Q)(Q + 1)$$

$$\sigma = \sum_{i=1}^{M} b_i$$

$$Q \geq 2 \in Z^{+}$$

and

$$A_i^{\dagger} = A^{(1)} \quad \forall i.$$

A result analagous to Result 1 can now be proved; the details are omitted as its proof is identical to that of Result 1. This then leads to:

Theorem 2

DMT is *NP*-hard.

Proof

This follows the proof of Theorem 1 with, in this case, DMT replacing DMTD.

In essence, Theorems 1 and 2 prove not only that DMTD and DMT respectively are *NP*-hard, but also the stronger results that DMTD and DMT are *NP*-hard even in the simple cases which involve a single deflection constraint and a choice of only two possible member sizes.

In proving these results no attempt was made to distinguish between determinate and indeterminate trusses, in fact such a distinction was unnecessary. That DMT is *NP*-hard does, however, introduce certain issues which have a bearing on practical applications, and since it has become almost a fact of life that most practicable structures are indeterminate, it will be these that are highlighted in the following.

Only a single analysis of a statically determinate truss is necessary to define the single instance of DMT which, if it can be solved exactly, will generate member sizes that correspond to minimum weight. On the other hand, despite the fact that DMT is algebraically well defined for a statically indeterminate truss, this is not so numerically and current procedural dictates require an iterative solution technique which performs an analysis and generates a numerically well defined instance of DMT which must be solved at each iteration. In view of the computationally intractable nature of *NP*-hard problems, the result of Theorem 2 predicts that the solution of an instance of DMT for a determinate truss may require an unacceptably large amount of time and that the weight optimisation and consequent solution of several instances of DMT for an indeterminate truss may be even more time critical. That this is so is borne out by

the practical experience of workers in the field who have long since realised that the number of iteration steps (and therefore the number of DMT's requiring solution) significantly affects the computer time necessary to reach a solution, and consequently, that the number should be minimised whenever possible. Now Theorem 2 must be considered as a prime theoretical reason for justifying the expedient which is now ubiquitous in practice: that of seeking only an approximate solution to DMT. Consequently, the remainder of this paper is concerned with examining a particular class of approximation algorithms for DMT where:

Definition 8

Let L be an optimisation problem, then an *approximation algorithm* Λ for L is an algorithm that always generates a feasible solution to L whose value is "sufficiently close" to the value of the optimum.

4. ABSOLUTELY-BOUNDED APPROXIMATION ALGORITHMS FOR DMT

The class considered is that of absolutely-bounded approximation algorithms, where:

Definition 9

Λ is an *absolutely-bounded* approximation algorithm for a problem L if and only if for every instance I of L:

$$|\Omega\dagger(I) - \Omega^*(I)| \leq \kappa$$

where $\Omega^*(I)$ is the optimum value of the instance I, $\Omega\dagger(I)$ is the value of the feasible solution to I that is generated by Λ and κ is a constant. (It is noted that some authors, see e.g. Horowitz and Sahni[3], use the term "absolute" instead of "absolutely-bounded").

In the present case DMT is a minimisation problem and the above condition can be rewritten as:

$$\Omega(I) \leq \kappa + \Omega^*(I) \tag{4.1}$$

for all instances. Since approximate solutions to DMT are so frequently sought in practice an absolutely-bounded approximation algorithm would clearly be most desirable and of great value. For many *NP*-hard problems, algorithms of this type can be shown to exist only if the class of problems solvable in polynomial deterministic time is equivalent to the class of problems solvable in polynomial non-deterministic time viz. the approximation algorithms are themselves *NP*-hard; unfortunately this is the case with absolutely-bounded approximate algorithms for DMT. It is not difficult to conjecture that this is true in view of the fact that DMT can be scaled. In order to substantiate this conjecture, a formal proof that deriving an absolutely-bounded approximate algorithm for DMT is *NP*-hard is given below, but first, the following definition is provided:

Definition 10

The absolutely-bounded approximate DMT problem is the problem of finding any feasible solution of DMT which satisfies (4.1).

Theorem 3

The absolutely-bounded approximate DMT problem is *NP*-hard.

Proof

This follows the techniques used to prove that the absolutely-bounded approximate 0–1 Knapsack problem is *NP*-hard, see [8].

The problem adopted as being known to be *NP*-hard is DMT. Let (3.1) define instance I of DMT and assume that $\gamma_i \in Z^+$, $\beta_{ij} \in Z^+$, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, J$, and $s_k \in Z^+$, $k = 1, 2, \ldots, K$. Further assume that Λ is a polynomial time algorithm that guarantees feasible solutions $\Omega\dagger(I)$ such that $\Omega\dagger(I) \leq \kappa + \Omega^*(I)$ for every instance I and fixed finite κ.

Define an instance I', of the absolutely-bounded approximate DMT problem, by:

$$\underset{\mathbf{A}}{\text{minimise}}\ w(\mathbf{A}) = \sum_{i=1}^{M} (\kappa + 1)\gamma_i A_i$$

subject to:

$$\sum_{i=1}^{M} \frac{\beta_{ij}}{A_i} \leq \delta_j \dagger \quad j = 1, 2, \ldots, J$$

$$\left.\begin{matrix} A_i \geq A_i^* \in S \\ A_i \in S \end{matrix}\right\} \quad i = 1, 2, \ldots, M$$

where the γ_i, the β_{ij} and the s_k are as defined for I.

It is clear that I and I' have the same set of feasible vectors $\mathbf{A}$, that the solution vector $\bar{\mathbf{A}}$ to I is also the solution vector to I', and that $\Omega^*(I') = (\kappa + 1)\Omega^*(I)$.

Further, since the γ_i and s_k are integer, all feasible solutions to I' have value $\Omega^*(I')$ or at least $\bar{\Omega}(I') = \Omega^*(I') + (\kappa + 1)$. Consequently, if $\Omega\dagger(I')$ is the value of the solution generated by Λ for instance I', then either $\Omega\dagger(I') - \Omega^*(I') = 0$ or $\Omega\dagger(I') - \Omega^*(I') \geq \kappa + 1$. However, Λ guarantees $\Omega\dagger(I') - \Omega^*(I') \leq \kappa$, hence $\Omega\dagger(I') = \Omega^*(I')$. Consequently, Λ can be used to obtain an optimal solution for I' and hence for I, and the result follows from the fact that DMT is *NP*-hard.

This result clearly shows that finding a solution to the absolutely bounded approximate DMT problem is as difficult as finding a solution to the original DMT problem. In practical terms this result implies that any acceptably fast algorithm will only guarantee a condition on $\Omega\dagger(I)$ that is weaker and therefore clearly less desirable than (4.1).

5. SUMMARY AND DISCUSSION

The problem of minimising the weight of structural trusses subject to constraints on joint deflection and member stress and gauge has been considered. In particular, the case DMT where member sizes are available from a discrete set has been examined and shown to be *NP*-hard. As an intermediate step, the problem DMTD where only joint deflection constraints apply, was also shown to be of equivalent difficulty. Consequent upon these results, the problem of devising an approximation algorithm for DMT that guaranteed a bound on the absolute error was considered, but unfortunately, this too proved to be *NP*-hard.

That these problems are *NP*-hard can be viewed from two directions. On the one hand, *NP*-hardness does not incontravertibly prove that these problems are intractable since it may be true that **P = NP**, however this result has so far eluded proof. Furthermore, there exist *NP*-hard problems, for example: integer linear programming and the Knapsack problem, for which there are algorithms that work well in practice despite their having exponential time complexity. On the other hand problems that are *NP*-complete and/or *NP*-hard are, in general, *de facto* intractable since currently available techniques cannot "adequately" solve them and examples that work well in practice, such as those quoted above are rare. Despite these two opposing viewpoints it is more reasonable, in the light of the current state of knowledge, to operate under the assumption that **P ≠ NP** with all its attendant implications, than to adopt the alternative stance and commit considerable time and manpower to proving the contrary with, on the basis of experience, only a minimal expectation of success. It is the belief of the authors that such should be the operational considerations for DMT especially when account is taken of the accumulated practical experience gained in attempting to solve it.

REFERENCES

1. N. S. Khot, Algorithms based on optimality criteria to design minimum weight structures. *Engng Optimisation* **5**, 2 (1981).
2. C. Fleury and L. A. Schmit, Primal and dual methods in structural optimisation. *J. Structural Div., ASCE* **106**, 5 (1980).
3. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*. Computer Science Press, Potomac (1978).
4. D. E. Knuth, A terminological proposal. *SIGACT News* **6**, 1 (1974).
5. M. R. Garey and D. S. Johnson, *Computers and Intractability*. Freeman, San Francisco (1979).
6. R. Karp, Reducibility among combinatorial problems. In: *Complexity of Computer Computations* (Edited by R. E. Miller and J. W. Thatcher). Plenum Press, New York (1972).
7. S. A. Cook, The complexity of theorem proving procedures. *Proc. 3rd. ACM Symp. on Theory of Computing* (1971).
8. S. Sahni, Approximate algorithms for the 0–1 Knapsack problem. *JACM* **22** (1975).