DOCUMENT RESUME

ED 308 248 TM 013 668

AUTHOR	Adema, Jos J.
TITLE	A Note on Solving Large-Scale Zero-One Programming Problems. Research Report 88-4.
INSTITUTION	Twente Univ., Enschede (Netherlands). Dept. of Education.
PUB DATE	Apr 88
NOTE	17p.
AVAILABLE FROM	Bibliotheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Achievement Tests; *Computer Assisted Testing; *Heuristics; Item Banks; *Problem Solving; *Programing; *Test Construction; Testing Problems; Test Reliability
IDENTIFIERS	*Zero One Programming

ABSTRACT

A heuristic for solving large-scale zero-one programming problems is provided. The heuristic is based on the modifications made by H. Crowder et al. (1983) to the standard branch-and-bound strategy. First, the initialization is modified. The modification is only useful if the objective function values for the continuous and the zero-one programming problems are close to each other. Given the initialization, the branch-and-bound method is stopped when a feasible solution to the problem is found. The heuristic also uses the reduced costs to fix non-basic variables to 1 or 0. An example taken from achievement test construction illustrates the efficiency of the proposed heuristic. Several test construction problems were implemented and solved by the proposed heuristic for item banks with 400 items. Modifications were introduced in the LANDO computer program. A table illustrates that the central processing unit times for solving the zero-one programming problem were close to the times needed to solve the continuous problem. (SLD)

A Note on Solving Large-Scale Zero-One Programming Problems

Research Report 88-4

Jos J. Adema

department of EDUCATION

Division of Educational Measurement and Data Analysis

University of Twente

Colofon:
Typing: Mevr. L.A.M. Padberg
Cover design: Audiovisuele Sectie TOLAB Toegepaste
Onderwijskunde
Printed by: Centrale Reproductie-afdeling

A Note on Solving Large-Scale Zero-One Programming Problems

Jos J. Adema

A note on solving large-scale zero-one programming problems / Jos J. Adema - Enschede : University of Twente, Department of Education, April, 1988. - 10 pages

Abstract

A heuristic for solving large-scale zero-one programming problems is given. The heuristic is based on Crowder, Johnson and Padberg's (1983) modifications of the standard branch-and-bound strategy. First, the initialization is modified. The modification is only useful if the objective function value for the continuous and the zero-one programming problem are close to each other. Given the initialization, the branch-and-bound method is stopped when a feasible solution to the problem is found. The reduced costs are used to fix variables. As an example, the heuristic is applied to achievement test construction problems with good results and excellent CPU-times.

A Note on Solving Large-Scale Zero-one Programming Problems

Recently, Crowder, Johnson and Padberg (1983) gave some methodological advances which, combined with clever branch-and-bound strategies, solve sparse large-scale zero-one programming problems in feasible time. This method may replace standard branch-and-bound approaches such as in Dakin (1965) which for many applications take too much time (Papadimitriou & Steiglitz, 1982). In this note a heuristic will be given which is based on some ideas proposed by Crowder et al. in their branch-and-bound strategy. The heuristic is useful for solving large-scale zero-one linear programming problems with small differences between the coefficients in the objective function. A practical example from the area of achievement test construction shows that the heuristic solves zero-one programming problems in CPU-times close to those for solving the continuous relaxations.

Notation

Zero-one programming problems of the following form are considered:

(P). Maximize $\{c'x \mid Ax \leq b,\ x_j = 0 \text{ or } 1 \text{ for } j = 1,\ldots, n\}$

where A is an mxn-matrix and b and c are vectors of lenghts m and n.

The continuous optimal objective function will be denoted by Z_{LP} and the true lower bound on the zero-one optimal objective function by Z_+. In the heuristic, the continuous optimal reduced costs, d_j, corresponding to variable x_j are used to fix variables at the value 0 or 1.

Modifications in Standard Branch-and-bound

The modifications in the branch-and-bound strategy given ın this section can be applied when (1) the optimal value of the objective function for the zero-one programming problem P is close to Z_{LP}, and (2) problem P is a large--scale pıogramming problem. It is assumed that Z_{LP} is not equal to 0.

If no feasible solution to P is known, the branch-and-bound method is initialized by assuming $Z_+ = -\infty$. But if it is known that the optimal value of the zero-one objective function is close to Z_{LP}, the branch-and-bound method can, after solving the relaxation of P, be initialized by $Z_+ = K_1 Z_{LP}$, where K_1 is a constant $(0 << K_1 <1)$.

Given the above initialization it is known that every zero-one solution found during the search process has a value of the objective function between $K_1 Z_{LP}$ and Z_{LP}. So if K_1 is close to 1 every solution is a good solution. This means that the branch-and-bound method can be stopped when the first feasible solution for P is found. In this way a good solution, but not necessarily the best one, is obtained. In most applications this is no problem because the coefficients in the model are estimates and the differences between the exact solution and the one found can be made arbitrarily small.

The heuristic also uses the reduced costs to fix nonbasic variables to 1 or 0:

1) Fix x_j to 0 if $x_j = 0$ in the continuous solution and $Z_{LP} - K_2 Z_{LP} < d_j$

2) Fix x_j to 1 if $x_j = 1$ in the continuous solution and $Z_{LP} - K_2 Z_{LP} < -d_j$

where $K_2 < 1$. The above rules are applied after the continuous solution of the relaxation of P is found. The value of K_1 cannot be chosen as high as the value of K_2. because when choosing K_1 it must be certain that the value of the objective function for the solution of P is larger than $K_1 Z_{LP}$. If the value of K_1 or K_2 is too large, the decision tree is small. Then it does not take much time before it is clear that no solution to P can be found for the chosen values of K_1 and K_2. In such a case the values of K_1 and/or K_2 can be adjusted and the procedure is started anew. The following example illustrates the efficiency of the above heuristic.

Example

In achievement test construction a high value for the reliability coefficient of the test is wanted and this goal can be achieved by selecting items from a test item bank with large contribution to the reliability (van der Linden &

Adema, 1987). However, other test construction goals are also possible (Theunissen, 1985; van der Linden & Boekkooi-Timminga, 1988). In practice, test item banks usually consist of hundreds of items and practical constraints have to be imposed on selection of items.

The heuristic has been applied to zero-one programming models with the goal as mentioned above in the objective function. The (0,1)-variables x_j were defined as follow:

$$x_j = \begin{cases} 0 \text{ item j not in the test} \\ 1 \text{ item j in the test.} \end{cases}$$

The total time of test administration and the mean of the difficulties of the items in the test were restricted. Except for these constraints all the coefficients in the constraints were -1, 0 or 1. One constraint was introduced to fix the number of items in the test Also the numbers of items to be selected from different subdomains of the item bank were restricted. Sometimes we want two items to be simultaneously included in or excluded from the test. Constraints to satisfy this wish were included in the model. In all, 14 constraints were imposed.

A number of test construction problems were implemented and solved by the proposed heuristic for item banks with 400 items. This was done on a DEC2060 computer. The modifications were introduced in the program LANDO. It was assumed that the items in the item bank satisfy an item response model. An item response model specifies a relationship between the observable examinee test performance and the unobservable trait or ability assumed to underlie performance on the test. The relationship between the "observable" and the "unobservable" quantities is described by a mathematical function. For this reason, item response models are mathematical models based on assumptions about the test data (Hambleton & Swaminathan, 1985). In the implementations, the Rasch model and the 3-parameter logistic model were used. The probability that an item i is answered correctly by a person with ability θ under the Rasch model is

$$P(+|i,\theta) = [1 + \exp(b_i - \theta)]^{-1}$$

where b_i is the difficulty of item i and repesents the point on the ability scale at which an examinee has a 50 percent probability of answering item i correctly. Under the 3-parameter model this probability is

$$P(+|i,\theta) = c_i + [(1 - c_i) / (1 + \exp(-a_i(\theta - b_i))]$$

where the parameter c_i represents the probability of examinees with low ability correctly answering an item and the parameter a_i, called item discrimination, is proportional to the slope of $P(+|i,\theta)$ at the point $\theta = b_i$.

The results are shown in Table 1. To see how important the choice of K_1 is, two values, .99 and .995, were chosen for K_1. Parameter K_2 was set equal to .998 if 20 items were selected and equal to .999 if 40 items were selected. It is possible to choose K_2 higher when 40 items are selected, because the difference between Z_+ and Z_{LP} in percents is smaller for 40 items selected than for 20 item selected.

Insert Table 1 here

For the 3-parameter model the dispersion in the coefficients of the objective function was greater. Therefore, more variables were fixed after applying the rules with the reduced costs. As a conseqence, the CPU-times for the 3-parameter model were better.

Because of the modification in the initialization, more branches are fathomed without finding a feasible solution of P for $K_1 = .995$ then for $K_1 = .99$. Therefore it is possible that more CPU-time is needed to solve P for $K_1 = .995$.

Conclusions

A heuristic for solving large-scale zero-one programming problems is proposed. This heuristic is useful in particular when the optimal objective function value of P is close to Z_{LP}. The heuristic was used to solve test construction problems. As shown in Table 1, the CPU-times for solving the zero-one programming problem were close to the CPU-times needed to solve the continuous problem.

References

Crowder, H., Johnson E.L., and Padberg, M. (1983). Solving large scale zero-one programming problems. Operations Research, 31, 803-834.

Dakin R.J. (1965). A tree-search algorithm for mixed integer programming problems. Comp. J., 8, 250-255.

Hambleton, R.K. & Swaminathan, H. (1985). Item response theory: Principles and applications. Boston: Kluwer-Nijhoff publishing.

Papadimitriou, C.H. & Steiglitz, K. (1982). Combinatorial optimization: Algorithms and complexity. Englewood Cliffs: Prentice Hall.

Theunissen, T.J.J.M. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

van der Linden, W.J. & Adema, J.J. (1988). Algoritmic test design using classical parameters. Submitted for publication.

van der Linden, W.J. & Boekkooi-Timminga, E. (1988). A maximin model for test design with practical constraints. Psychometrika, in press.

Table 1

CPU-times and objective function values for different values of K_1 and K_2

			Zero-one		Continuous	
ni	K_1	K_2	Objective Function Value	CPU (sec)	Objective Function Value	CPU (sec)
			Rasch model			
20	.990	.998	8.80	22.17	8.8533	18.04
20	.995	.998	8.82	24.65	8.8533	18.04
40	.990	.999	17.27	34.82	17.3911	24.63
40	.995	.999	17.32	35.96	17.3911	24.63
			3-parameter model			
20	.990	.998	9.43	18.35	9.4468	16.68
20	.995	.998	9.43	18.93	9.4468	16.68
40	.990	.999	18.35	25.46	18.3500	25.42
40	.995	.999	18.35	25.47	18.3500	25.42

Note. ni = number of items in the test

Titles of recent Research Reports from the Division of Educational Measurement and Data Analysis, University of Twente, Enschede, The Netherlands.

RR-87-1 R. Engelen, Semiparametric estimation in the Rasch model

RR-87-2 W.J. van der Linden (Ed.), IRT-based test construction

RR-87-3 R. Engelen, P. Thommassen, & W. Vervaat, Ignatov's theorem: A new and short proof

RR-87-4 E. van der Burg, & J. de Leeuw, Use of the multinomial jackknife and bootstrap in generalized nonlinear canonical correlation analysis

RR-87-5 H. Kelderman, Estimating a quasi-loglinear models for the Rasch table if the number of items is large

RR-87-6 R. Engelen, A review of different estimation procedures in the Rasch model

RR-87-7 D L. Knol & J.M.F. ten Berge, Least-squares approximation of an improper by a proper correlation matrix using a semi-infinite convex program

RR-87-8 E. van der Burg & J. de Leeuw, Nonlinear canonical correlation analysis with k sets of variables

RR-87-9 W.J. van der Linden, Applications of decision theory to test-based decision making

RR-87-10 W.J. van der Linden & E. Boekkooi-Timminga, A maximin model for test design with practical constraints

RR-88-1 E. van der Burg & J. de Leeuw, Nonlinear redundancy analysis

RR-88-2 W.J. van der Linden & J.J. Adema, Algorithmic test design using classical item parameters

RR-88-3 E. Boekkooi-Timminga, A cluster-based method for test construction

RR-88-4 J.J. Adema, A note on solving large-scale zero-one programming problems

<u>Research Reports</u> can be obtained at costs from Bibliotheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

Department of
EDUCATION

A publication by
the Department of Education
of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands