SCIENTIFIC PAPERS
OF THE UNIVERSITY OF PARDUBICE
Series A
Faculty of Chemical Technology
10 (2004)

AN EFFICIENT SPREADSHEET PROCEDURE FOR SOLVING A 0-1 INTEGER PROGRAMMING PROBLEM

Otakar MACHAČ and Simona BÖHMOVÁ[1]
Department of Economy and Management of Chemical and Food Industry,
The University of Pardubice, CZ–532 10 Pardubice

Received September 29, 2004

The paper describes a new effective procedure for solving a 0-1 integer optimization problem with use of spreadsheet tables. It is pointed out that a preoptimality analysis plays an important role in these types of problems. Next, the algorithm procedure for use of Microsoft Excel was suggested. It is based on a quick elimination of infeasible combinations and on a reduction of feasible combinations which cannot be the optimal solution for sure. The remaining feasible combinations are analysed in detail in a specifically prepared table in which the optimum solution can be found effectively. In the last part, the advantages of suggested procedure, in comparison with heuristic and exact (used by Solver in Excel) methods, are considered.

[1] To whom correspondence should be addressed.

Introduction

The following decision-making problem often appears in the area of managerial or executive activities: there is a group of mutually independent and competitive projects, each of them requires a certain amount of limited resources and yields certain economic contribution (for example Net Present Value of projects - NPV). Very often, resources cannot cover a realization of the whole group of projects. The task is to choose the set of projects that maximizes total economic contribution and stays within budget and other possible constrains at the same time. Projects can be represented by various activities, for example investment projects, research projects, business plan, complex order etc. In literature, such a problem is called "Capital Budgeting Problem".

The crucial assumption is that projects are indivisible, it means a project can be accepted or refused always as a whole, partial investments are not allowed. Therefore this problem is a type of integer programming problem with 0-1 variables (called a 0-1 IP). A 0-1 variable is a decision variable that must equal 0 or 1. If a 0-1 variable corresponding to the activity (project) equals 1, then the activity is undertaken; if it equals 0, the activity is rejected.

Theory

Formulation of a 0-1 Integer Programming Problem

The basic 0-1 Integer Programming Problem can be written as follows

$$\text{MAXIMIZE} \quad Z = \sum_{j=1}^{n} c_j x_j$$

$$\text{SUBJECT TO} \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i \text{ for } i = 1, 2, \ldots m \text{ constraints}$$

$$x_j = 0 \vee 1 \text{ for } j = 1, 2, \ldots n \text{ independent projects}$$

In a matrix form

$$\begin{aligned} &\boldsymbol{Max\ Z} = \boldsymbol{c}^T \boldsymbol{x} \\ &\boldsymbol{Ax} \le \boldsymbol{b} \\ &\boldsymbol{x} = \boldsymbol{0} \vee \boldsymbol{1} \end{aligned} \qquad (1)$$

In some cases (see below), it is better to solve a dual of the original problem. The formulation of the dual problem is then

$$\begin{aligned} \boldsymbol{Ax} &\geq \boldsymbol{d} \\ \boldsymbol{Min\, v} &= \boldsymbol{c}^T\boldsymbol{x} \\ \boldsymbol{x} &= \boldsymbol{0} \vee \boldsymbol{1} \end{aligned} \tag{2}$$

where $d_i = \sum_{j=1}^{n} a_{ij} - b_i$ for $i = 1, 2, ...m$ constraints

Consequently, the easier task is chosen to solve the problem. If the constraints b_i allow using less than 50 % of the sum of all resources required by all projects, it would be better to solve the primal problem. On the contrary, if the values of constraints are close to the sum of resources required by all projects (it means we could realize more projects) it would be better to solve the dual problem. In this case, the goal is to eliminate variables that minimally decreased the amount of all possible contributions. Unlike the primal problem, 0-1 variable equals 0 for the undertaken project and 1 for the not undertaken one.

Solving 0-1 IP Model

Even if the 0-1 IP model is a type of the integer linear programming, its solution requires specific methods, different from procedures commonly used to solve LP models. There are two different approaches to solving 0-1 IP problems:

1. *exact solution procedures* — mostly based on a combinatorial approach. The main disadvantage is a high laboriousness (time demanding) — if we consider n variables there are 2^n of possible combinations to be examined (for example in a model with 20 variables there are more than 1 mil. combinations).
2. *heuristic methods* — can lead to a good solution, not necessarily optimal, but can provide useful information to the problem. The advantage is usually its simplicity and clearness.

Some of the methods try to combine both approaches. The group of combinations is reduced using heuristics, followed by combinatorial algorithm that examines selected possible solutions, see [1,4], etc. In reality, most IP problems are solved by the technique called branch-and-bound, see [2,4,8] etc. This method is used by Excel Solver to solve integer linear optimization problems. In the case of 0-1 IP problems, implicit enumeration is usually recommended as being practical to use. Recently, development of spreadsheet techniques has brought new inspirations and possibilities for heuristic and combinatorial solutions methods, see [3] etc.

The solution method for 0-1 IP problem suggested here is also based on a combination of both approaches with use of spreadsheet techniques. The essential part of this method is analysis that precedes the solution itself (preoptimality analysis).

Discussion

Preoptimality Analysis of 0-1 IP model

Postoptimality analysis has been emphasized as a very important step for solving standard problems of linear programming. The analysis is concerned with how changes in the model's parameters affect the optimum solution to the problem. Preoptimality analysis has similar importance for solving the 0-1 IP problems because it reveals some important information on properties of the solution and on the whole effectiveness of the model. Within the preoptimality analysis we suggest to examine and evaluate:

- *Relative value of constraints* b_i

The number of selected projects is always determined by those constraints that are the strictest in view of the sum of resources required if all projects are realized. In the case of constraints:

$\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, the strictest constraints are those with the smallest ratio $p_i = b_i / \sum_{j=1}^{n} a_{ij}$ for $i = 1, 2, ...m$ constraints and $j = 1, 2, ...n$ independent projects. On the other hand, in the case of constraints: $\sum_{j=1}^{n} a_{ij}x_j \geq d_i$ the strictest constraints are those with the highest ratio $p_i = b_i / \sum_{j=1}^{n} a_{ij}$.

The constraints with the ratio close to 0.5 have the highest influence on optimality efficiency. On the other hand, the importance (relevance) of the optimizing procedure decreases as the ratio p_i is closer to 1 or 0.

- *Rank correlation of projects on the basis of project contribution per resource invested, i.e. ratios* c_j / a_{ij}

The higher a direct correlation among effectiveness of invested resources, the higher effect optimization will bring, because projects that dominate the others (from all or most points of view) are selected. Preoptimality analysis and solution algorithm for 0-1 IP model with the use of spreadsheet techniques is shown below on a numerical example.

Example

Let us assume a company is considering 8 mutually independent investment projects. Resources (cash and new employees) required for each investment, limits of both restricted sources and assumed contributions each investment adds to the company are given in the first part (thickly framed) of Table I. But entry data do not contain any information about the problem properties or efficiency of a potential optimalizing procedure. Therefore the results of preoptimality analysis are shown in Table I, too; i.e. ratios $p_i = b_i / \sum_{j=1}^{n} a_{ij}$ for both constraints, ratios $r_{ij} = c_j / a_{ij}$ representing relative efficiency of project j with regard to source i and the order of projects according to the ratio r_{ij}. Based on this order, Spearman coefficient of rank correlation is calculated which allows evaluation of the independence rate between both criteria.

Table I Entry data and basic calculation for a projects group

Project number j =	Cash required mil. CZK a_{1j}	Employees required number a_{2j}	NPV mil. CZK c_j	r_{1j}	Order	r_{2j}	Order	Avr. Order
1	300	65	300	1.000	4	4.615	8	6
2	280	54	300	1.071	3	5.556	2	2.5
3	150	30	180	1.200	1	6.000	1	1
4	450	80	400	0.889	7	5.000	6	6.5
5	320	68	350	1.094	2	5.147	3	2.5
6	220	40	200	0.909	6	5.000	5	5.5
7	350	60	280	0.800	8	4.667	7	7.5
8	330	63	320	0.970	5	5.079	4	4.5
Σ	2400	460	2330					
Constraint b_i	1800	360			correlation coefficient = 0.7381			
Rate p_i, %	75.00 %	78.26 %						

Preoptimality analysis reveals that the rank correlation of project efficiency is medium- strength and limits of both sources can be used up to 3/4 of resources required by all projects. Therefore the optimizing procedure seems to be effective; solving the problem can mean a significant contribution comparing to an empirical or a random selection of projects.

Heuristics

Heuristic solutions are mostly approximate, but fast and simple at the same time. The key principle is mostly based on ranking projects according to a particular viewpoint or a combination of factors and a consequent selection of projects until all limits for individual sources are depleted. Procedures differ from each other in the way that the ranking criteria are chosen. One of the simplest rules is shown by Ragsdale [3]; he recommends to rank projects according to absolute value of contributions (i.e. c_j) and to choose a file of first k projects that still fulfil all the constraints b_i.

For our example, suggested solution is given in Table II. It selects the first 5 projects (i.e. 4, 5, 8, 2, 1) with NPV of 1670. This solution is the best of 5-element combinations but as we will see later, this solution is still far from optimal solution. The main disadvantage of this procedure is that the projects with the highest NPV do not have to be the most efficient and there is no possibility to add projects with lower NPV that are more efficient in using resources.

Table II Order of projects on the basis of c_j

j =	a_{1j}	a_{2j}	c_j	$\sum_j a_{1j}$	$\sum_j a_{2j}$	$\sum_j c_j$
4	450	80	400	450	80	400
5	320	68	350	770	148	750
8	330	63	320	1,100	211	1,070
2	280	54	300	1,380	265	1,370
1	300	65	300	1,680	330	1,670
7	350	60	280	2,030	390	1,950
6	220	40	200	2,250	430	2,150
3	150	30	180	2,400	460	2,330
Sum	2,400	460	2,330			
Constraint	1,800	360				

The other possible approximate heuristic solution procedure lies in ranking projects on the basis of ratios $r_{ij} = c_j/a_{ij}$. For problems with more than one constraint ($i \geq 2$), we can rank projects based on each i separately or based on average order calculating all constraints. In our example, we can find three approximate solutions, on the basis of r_{1j}; r_{2j} and average position. Basic data are given in Table I. Projects order ranked on the basis of r_{1j}, cumulative values of a_{1j}, a_{2j}, and c_j are shown in Table III.

This solution indicates that a company can obtain NPV of 1650 by selecting investments 3, 5, 2, 1, 8, and 6. Similarly, if we rank investments on the basis of r_{2j}, the selected projects are 2, 3, 4, 5, 6, and 8 with NPV of 1750. As we will see later, this solution is already one of the optimal solutions. If we rank investments on the basis of average position, selected projects are the same as solution ranking projects on the basis of r_{1j} with NPV of 1650.

Table III Order of projects on the basis of r_{1j}

j = 1	r_{1j}	$\sum_j a_{1j}$	$\sum_j a_{2j}$	$\sum_j c_j$
3	1.200	150	30	180
5	1.071	470	98	530
2	1.094	750	152	830
1	0.970	1,050	217	1,130
8	0.909	1,380	280	1,450
6	1.000	1,600	320	1,650
4	0.889	2,050	400	2,050
7	0.800	2,400	460	2,330
Constraint	x	1,800	360	x

Solving 0-1 IP problem with Excel's Solver

Our problem can be solved with Solver, too, see [6,7], etc. To solve IP problems with Solver we must use Solver's binary option (the constraint "decision = bin" indicates that variables are 0-1 variables). The solution for our example in Table IV shows that we should undertake projects number 2, 3, 4, 5, 6 and 8 with objective function value of 1,750 mil CZK. The solution is identical to one we have already obtained using heuristics.

Here, we must pay attention to the solver Option dialog that includes a Tolerance field for IP models. The default Tolerance value is 5 % and it means that the optimization procedure continues until the solution (its value of objective function) is within 5 % of IP optimum value of objective function. A higher tole-

Table IV Solution of our problem using Excel's Solver.

j =	a_{1j}	a_{2j}	c_j	variables
1	300	65	300	0
2	280	54	300	1
3	150	30	180	1
4	450	80	400	1
5	320	68	350	1
6	220	40	200	1
7	350	60	280	0
8	330	63	320	1
constraint b_i	1800	360		
solution	1750	335	1750	

rance value can speed up the procedure but the solution may not be the optimal one and may be even farther from the real IP optimum. Tolerance of 0 % forces Solver to find the actual IP optimum but is more time-consuming.

The solution in Table 4 was found using default Tolerance value of 5 %. Setting Tolerance to 4 % (and lower %), Solver found solution again but chose projects number 1, 2, 5, 6, 7, 8 to be undertaken with objective function value 1,750 mil CZK. One could think this solution would be better than the one with tolerance value 5 % but this is not true because despite this solution having the same objective function value, it uses more resources (1,800 mil CZK and 350 employees).

The advantage of Solver can be seen in the simplicity of finding the solution but there are disadvantages, too. First, we do not always obtain the true optimal solution and second, Solver does not provide any additional information about the solution.

Efficient Method for 0-1 IP Problem Solving with Microsoft Excel

Next, we will introduce the optimisation method for solving a 0-1 IP problem combining both the heuristic and combinatorial procedures with use of some of the Excel tools. The whole procedure can be summarised in following steps:

1. Formulation of 0-1 integer programming problem
2. Preoptimality analysis, see procedure in Table I
3. Selection of primal or dual problem for the optimisation
4. Separation of k-elements subsets for $k = 1, 2, \dots n$ into 2 groups: k_p subsets containing feasible solutions and k_n subsets with infeasible solutions.
5. Reduction of k_p subsets to k_{p+} subsets with possible optimal solutions and to k_{p-} subsets that cannot contain optimal solution.
6. Complete examination of all combinations in k_{p+} subsets and determination of the optimal solution.

The algorithm of suggested method is shown on our example. Model formulation and preoptimality analysis is discussed afore. Since $p_1 = 75.00$ % and $p_2 = 78.26$ %, the dual problem is chosen (see Table I). The formulation of dual problem is:

minimize $300x_1 + 3000x_2 + 180x_3 + 400x_4 + 350x_5 + 200x_6 + 280x_7 + 320x_8 = z$

subject to:

$$300x_1 + 280x_2 + 150x_3 + 450x_4 + 320x_5 + 220x_6 + 350x_7 + 330x_8 \geq 600$$

$$65x_1 + 54x_2 + 30x_3 + 80x_4 + 68x_5 + 40x_6 + 60x_7 + 63x_8 \geq 100$$

$$x_j = 0 \vee 1 \quad \text{for } j = 1, 2, \dots 8$$

The separation of k-elements subsets (defined in step 4) can be efficiently carried out in Excel using the following algorithm:

Projects are sorted according to above-mentioned constraints either in ascending or descending order. Projects rearrangement according to a given criteria can be done in Excel very simply by means of "Data\Sort" option. Project rearrangement according to total resource requirements is shown in Table V. Hence we can easily find out two boundary combinations for each $k = 1, 2, \ldots n$, with the highest and the lowest value. If both boundary combinations for k-elements subset comply with a given constraint, all the rest combinations comply, too. On the contrary, if both boundary combinations do not comply, none of the other possible k-elements combinations comply either. Finally, if the lowest combination fulfills the conditions and the highest does not (or contrariwise), such a k-elements subset contains some combinations that comply with constraint and other ones that do not. In maximizing model (1), infeasible combinations k_n will be determined by the highest value of k_{pi}, in dual - minimizing model (2), infeasible combinations will be, on the contrary, determined by the lowest value k_{pi}.

Table V Projects ranked according to a_{ij}

Project	according to a_{ij}			Project	according to a_{ij}		
j =	a_{1j}	a_{2j}	c_j	j =	a_{1j}	a_{2j}	c_j
3	150	30	180	3	150	30	180
6	220	40	200	6	220	40	200
2	280	54	300	2	280	54	300
1	300	65	300	7	350	60	280
5	320	68	350	8	330	63	320
8	330	63	320	1	300	65	300
7	350	60	280	5	320	68	350
4	450	80	400	4	450	80	400
Sum	2,400	460	2,330	Sum	2,400	460	2,330
b_i primal	1,800	360	x	b_i primal	1,800	360	x
d_i dual	600	100	x	d_i dual	600	100	x

We start the examination for the dual model in accordance to step 4 with one-element subsets, that is for $k = 1$. It is obvious from Table V that none of 1-element combination complies with both constraint conditions because even the project with the highest values does not comply either (it is not true that $a_{14} \geq 600$ and it is not true that $a_{24} \geq 100$ either).

Let us now start with examination of 2-elements combinations ($k = 2$). The total number of 2-elements combinations is 28. It can be deduced from Table V that combination of projects number 3 and 6 with the lowest requirements do not comply with both given conditions, whereas combination of projects with the highest requirements, number 7 and 4 or 5 and 4, comply with both conditions already. Hence two elements subsets can contain feasible and thus optimal solution.

Next, we will continue with 3-elements combinations, $k = 3$. Using Table V, we can deduce that combination of three projects with the lowest values, projects 3, 6 and 2, is a solution feasible for both restricted conditions (150 + 220 + 280) = 650 and (30 + 40 + 54) = 124. If this combination is a feasible solution, all the rest of possible combinations will be feasible, too. We do not need to examine following combinations for $k = 4, 5, 6, 7$ and 8 elements as they all are evidently feasible.

Elimination of Combinations that Cannot Contain Optimal Solution

In next step, we will reduce feasible k_p combinations to those (k_{p+}) with expected optimal solutions according to point 5 and refuse combinations (k_{p-}) for which the optimal solution cannot be expected. The practical procedure will be shown in our example.

First, we examine 3-elements combinations that are all feasible. The best solution in this group of solutions can easily be found in Table II with projects ranked on the basis of c_j. The minimal value of objective function for 3-elements combinations has the value of 660 for projects combination of 3, 6, 7 ($c_3 + c_6 + c_7 = 660$). Logically, we can exclude possibility that 4- and more elements combinations can have a lower value of objective function. In the next step, we must focus on 2-elements combinations containing feasible and infeasible solutions.

Complete Examination of k_{p+} Combinations

As we excluded infeasible combinations and reduced some of feasible solutions, it is obvious that the optimal solution is in the group of 2-elements combinations. Therefore we have to examine this group completely. Even this step can be solved effectively by reducing the number of combinations. For this purpose, the table with 2-elements combinations is prepared.

Combinations are made systematically — from the highest values of a_{1j} to the lowest. At the same time, the combinations are verified whether they comply with the first constraint. Selection stops if the next combinations cannot comply with this constraint. The next step is to verify if selected combinations also comply with the second constraint. Those which do not comply are eliminated — second reduction. In the case of selected combinations complying with both constraints, the values of objective function are calculated and the minimum — optimal solution is found (there can be more than one). The solution for our example is demonstrated in Table VI.

Table VI 2-elements combinations — finding optimum solution

j_p	j_{p+h}	$a_{1,jp}$	$a_{1,jp+h}$	$\sum a_{1(p+h)}$	$\sum a_{2(p+h)}$	$\sum c_{(p+h)}$
4	7	450	350	800	140	680
4	8	450	330	780	143	720
4	5	450	320	770	148	750
4	1	450	300	750	145	700
4	2	450	280	730	134	700
4	6	450	220	670	120	600
4	3	450	150	600	110	580
7	8	350	330	680	123	600
7	5	350	320	670	128	630
7	1	350	300	650	125	580
7	2	350	280	630	114	580
8	5	330	320	650	131	670
8	1	330	300	630	128	620
8	2	330	280	610	117	620
5	1	320	300	620	133	650
5	2	320	280	600	122	650

Since all feasible solutions chosen according to first constraint (a_{1j}) comply with the second constraint (a_{2j}) we do not need to proceed with the second reduction. There are 16 feasible solutions in Table VI and 3 of them are optimum solutions of dual problem — combinations of projects 3, 4; 1, 7 and next 2, 7. Objective function has a same value of 580 for all the three combinations. Since all 1-element combinations are infeasible and minimal value of objective function for 3-elements combinations is 660, all the three combinations are optimal solution to our problem.

Transformation of Dual into Primal Problem

Transformation of dual into primal problem for our example is in Table VII. Results from the Table show that the dual and primal problems have three optimal solutions with the same value of objective function. At the same time, we can see that these solutions are not completely the same in point of resource requirements. The worst seems to be solution eliminating projects 3 and 4 with the highest requirements on resources. The best appears to be the third solution with the lowest requirements on resources. It should be pointed out that there is no such information if we solve the problem with Excel's Solver.

Table VII Optimal solution of dual and primal problems

Dual problem				Primal problem			
Projects	Sum a_{1j}	Sum a_{2j}	Sum c_j	Projects	Sum a_{1j}	Sum a_{2j}	Sum c_j
3; 4	600	110	580	1;2;5;6;7;8	1,800	350	1,750
2; 7	630	114	580	1;3;4;5;6;8	1,770	346	1,750
1; 7	650	125	580	2;3;4;5;6;8	1,750	335	1,750

Conclusion

The main advantage of the method suggested is effective connection of heuristic and exact methods together with use of spreadsheet techniques of Excel. The method includes preoptimality analysis and provides more possibilities for postoptimality analysis in which the entry values are changed and influence on feasibility and optimality can be observed. At the same time, the solution procedure is highly effective, particularly when the spreadsheet technique with database and logic functions is used. The method presented is applicable even to more extensive models with more variables where a pure combinatorial solving procedure has its limitations even if a more powerful computer is used. Another benefit can be seen in easy interpretation and understandability of the whole procedure. Therefore we believe that the method suggested can mean a valuable contribution to the solving of this type of 0-1 integer problems even in practice.

References

[1] Balas E., Martin C.H.: Management Science, **26**, 86 (1980).
[2] Dennis T.L., Dennis L.B.: *Management Science*, West Publishing Company, 1991.
[3] Ragsdale C.T.: *Spreadsheet Modelling and Decision Analysis*, South-Western College Publishing, 2001.
[4] Winston W. L.: *Operations Research: Applications and Algorithms*, 2nd edition, PWS-KENT Publishing Company, Boston, 1991.
[5] Bell P.C.: *Management Science/Operations Research: A Strategic Perspective*, South-Western College Publishing, Cincinnati, Ohio, 1999.
[6] Winston W.L., Albright S.C.: *Practical Management Science*, 2nd edition, Duxbury/ Thomson Learning, 2001.
[7] More J.H., Weatherford L.R.: *Decision Modeling with Microsoft Excel*, 6th

edition, Prentice Hall, 2001.
[8] Jablonský J.: *Operational Research: Quantitative Models for Economic Decision-Making*, Professional Publishing, 2002 (in Czech).
[9] Albright S.C., Winston W.L.: *Spreadsheet Modeling and Applications – Essentials of Practical Management Science,* Thomson Learning-Brooks/Cole, 2005.
[10] Gros I.: *Quantitative Methods in Managerial Decision-Making*, Grada Publishing, a.s., Prague, 2003 (in Czech).