# An Application of K-Set Inequalities in Integer Programming, a Case Study of a Company Based In Accra

[1] **M. K. Ofosu** -The West African Examinations Council, Accra [2] **Professor Samuel Kwame Amponsah**-Lecturer
Department of Mathematics, Kwame Nkrumah University of Science and Technology,
University Post office (PMB) Kumasi, Ghana – West Africa-[3]***Jonathan Annan**-Lecturer
Department of Information Systems and Decision Science -School of Business
Kwame Nkrumah University of Science & Technology Kumasi
University Post office (PMB) Kumasi, Ghana – West Africa
**Corresponding Author: Otchere Fianko Alexander** on behalf of the authors

## ABSTRACT

*Integer programming models are mathematical models that can provide organizations with the ability to optimally obtain their goals through appropriate utilization and allocation of available resources. The knapsack problem is an integer programming problem that has only one constraint and can be used to strengthen cutting planes for general integer programs. In this research paper, we shall model an advert placement and selection slot problem of a company based in Accra as a 0-1 knapsack problem. The k-set inequality algorithm was used to solve the problem by coding it in Fortran 90.*

**Key words**:
Integer Programming, Knapsack Problems, Advert placement, K-Set Inequality

## 1. INTRODUCTION

Organizations face the problem of allocating limited resources among projects to maximize returns from a given investment, by selecting subset of projects, which can be funded within a given budget constraint. The knapsack problem is a general resource allocation problem in which a single resource is assigned to a number of alternatives with the objective of maximizing the total return. The knapsack problem seeks to optimize a set of yes/no decisions, subject to a single non-negative constraint. The problem is a distribution of effort problem that has a linear objective function and a single constraint.

## 2. RELATED WORKS

Jolayemi (2001) presented a model for the scheduling of projects under the condition of inflation and under penalty and reward arrangements. It included the effects of inflation on time-cost trade-off curves and a modified approach to time-cost trade-off analysis. The model revealed that misleading schedules and inaccurate project cost estimates will be produced if the inflation factor is neglected in an environment of high inflation. The authors also showed that award of penalty or bonus is a catalyst for early completion of a project.

Kanniappan et al., (1993) studied the problem of selecting various schemes under the integrated rural development program and to maximize the number of beneficiaries so as to optimize the annual income generated from each scheme. Typical constraints prescribed by the government in the allocation of the funds to several schemes from the budget outlay for integrated rural development each year. Through integer programming model and data from the district rural development agency of Dindigul Anna District, the authors were able to maximize the annual income generated from the schemes. Scogings et al., (1995) presented a model for student enrolment at the University of Natal. The project was part of an effort to find solution to meet the demand for accommodation of lecture rooms, which is to double up on the number of time tabled period so that no lecture room will be idle for the whole day, due to the steady increase in student enrolment over the years.

Hall et al., (1992) developed a mathematical model for a project funding decision facing United State Cancer Institute. The problem was to decide which project to fund given a strict limitation on capital availability. Hamdi et al., (2005) put forward a model that schedules project selection, which was formulated as a binary integer program. The model was applicable in various settings such as selection of engineering projects in corporate planning, or in other environments in which the candidate projects were interdependent. Hospitals need to constantly produce duty rosters for its nursing staff. Appropriate and considerate scheduling of nurses can have an impact on the quality of health care, the recruitment of nurses, the development of budgets, and other nursing functions. The nurse budget problem has been the subject of many academic studies. Cheang et al., (2003) presented an integer programming model to solve this problem.

## 3. MATHEMATICAL FORMULATION OF THE PROBLEM

The formulation of the knapsack problem is the same as a basic integer program with only one constraint and binary variables. Because the knapsack problem is formulated for sets of solutions containing only zeroes and ones, it is ideally suited to model decision systems. The requirement for a knapsack constraint is that it must be a "less than or equal to" constraint and that it must have only positive values for its coefficients and right hand side. Consideration is given to the 0-1 knapsack problem where $\sum_{i-1}^{n} w_i \le b$. Each item has a profit or cost $c_i$ and a weight $w_i$. The problem is to select a subset of the items whose total weight does not exceed the knapsack capacity b, and whose total profit is a maximum.

We assume without loss of generality that all input data are positive integers. Introducing the binary decision variable $x_i$ i with

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

The knapsack problem can be modeled as an integer linear programming model:

Maximize $Z = \sum_{i=1}^{n} c_i x_i$

Subject to

$$\sum_{i=1}^{n} w_i x_i \le b$$

**Algorithm**

K-set inequality algorithm (where k is an integer that represents the number of item types in the knapsack problem) was used. Given the knapsack problem with a total of N items, we let the number of item types be represented by the k-partitioned mutually exclusive subsets of N. Let the k-partitioned subsets also have members that represent the number of items that can be selected or not selected from each subset, up to the total number of items in each k-partitioned subset. The steps of the algorithm are:

**Step1:** Input the vector of values, weights and bag capacity.

**Step2:** Input the maximum capacity of the k-partitioned mutually exclusive subsets of item types.

**Step3:** For a feasible solution of each of the k-partitioned mutually exclusive subsets, from zero to the maximum number of each item subset, compute optimal values and weights.

**Step4:** While the computed weight less than knapsack capacity and optimal value less than computed value, optimal value equals computed value and optimal weight equals computed weight.

**Step5:** Repeat Step3 for all feasible solutions.

**Step6:** Stop at the end of final iteration, and report optimal value, optimal weight and the number of each item type selected.

## 4. DATA COLLECTION AND ANALYSIS

The study was undertaken using data collected from acompany, based in Accra, having an advertising budget of One hundred and fourteen thousand Ghana cedis(GH¢114,000.00) for use on advertisements. The company has the following media options: print, television, billboards, and radio. Table1 presents the detailed advert schedule for the company.

Table 1: Cost and Benefits of Placing an advert in a Media

| ITEM | MEDIA | NO. OF TIMES (thousands) | COST (GH¢ '000) | REACH OF PEOPLE |
|---|---|---|---|---|
| **1** | Print | 5 | 18 | 42 |
| **2** | Billboard | 3 | 13 | 26 |
| **3** | Television | 4 | 8 | 51 |
| **4** | Radio | 3 | 7 | 49 |

The problem is to select media types of adverts in such a way that the widest reach of people would be achieved without spending more than the amount allocated for adverts.

In the knapsack model, the holding capacity of the knapsack is the resource limit, given here as the advertising budget. The items to be considered are the different media that can be used, the weight of any item is the cost of placing an advert using that media, and the value of the item is the reach of the media type to the people.

The problem can be modelled mathematically as:

Maximize $R = \sum_{i=1}^{n} r_i m_i$

Subject to

$$\sum_{i=1}^{n} w_1 m_i \leq W$$

Where;

$R$ = Total reach

$r_i$ = Reach of each media or item

$m_i$ = Number of adverts placed using each media

$w_i$ = Cost of placing an advert in each media

$W$ = Total amount available for adverts (resource limit)

Thus, the company's problem is finally modelled as:

Maximize

$$R = 42(m_1 + m_2 + m_3 + m_4 + m_5) + 13(m_6 + m_7 + m_8) + 51(m_9 + m_{10} + m_{11} + m_{12}) + 49(m_{13} + m_{14} + m_{15})$$

Subject to

$$18(m_1 + m_2 + m_3 + m_4 + m_5) + 13(m_6 + m_7 + m_8) + 8(m_9 + m_{10} + m_{11} + m_{12}) + 7(m_{13} + m_{14} + m_{15}) \leq 114$$

## 5. SOFTWARE USED

To carry out the computation of the proposed model, we apply the k-set inequality algorithm coded in Fortran 90. The feature of the software permits the input data to be fixed into the code. The Software displays the final optimal solution for the problem. However, a walkthroughof the algorithm with our model gave the computational iterative values for the various optimal solutions as shown in Table 2.

The Fortran 90 code is shown in Appendix_1.

Table 2: Optimal Solutions for the various iterative stages

| ITERATION | ITEM SELECTED | OPTIMAL VALUE | OPTIMAL WEIGHT |
|---|---|---|---|
| **1** | {0,0,0,3} | 147 | 21 |
| **2** | {0,01,3} | 198 | 29 |
| **3** | {0,0,2,3} | 249 | 37 |
| **4** | {0,0,3,3} | 300 | 45 |
| **5** | {0,0,4,3} | 351 | 53 |
| **6** | {0,1,0,3} | 173 | 34 |
| **7** | {0,1,1,3} | 224 | 42 |
| **8** | {0,1,2,3} | 275 | 50 |
| **9** | {0,1,3,3} | 326 | 58 |
| **10** | {0,1,4,3} | 377 | 66 |
| **11** | {0,2,0,3} | 199 | 47 |
| **12** | {0,2,1,3} | 250 | 55 |
| **13** | {0, 2, 2, 3} | 301 | 63 |
| **14** | {0,2,3,3} | 352 | 71 |
| **15** | {0,2,4,3} | 403 | 79 |
| **16** | {0,3,0,3} | 225 | 60 |
| **17** | {0,3,1,3} | 276 | 68 |
| **18** | {0,3,2,3} | 327 | 76 |
| **19** | {0,3,3,3} | 378 | 84 |
| **20** | {0,3,4,3} | 429 | 92 |
| **21** | {1,0,0,3,} | 189 | 39 |
| **22** | {1,0,1,3} | 240 | 47 |
| **23** | {1,0,2,3} | 291 | 55 |
| **24** | {1,0,3,3} | 342 | 63 |
| **25** | {1,0,4,3} | 393 | 71 |
| **26** | {1,1,0,3} | 215 | 52 |
| **27** | {1,1,1,3} | 266 | 60 |
| **28** | {1,1,2,3} | 317 | 68 |
| **29** | {1,1,3,3} | 368 | 76 |
| **30** | {1,1,4,3} | 419 | 84 |
| **31** | {1,2,0,3} | 241 | 65 |
| **32** | {1,2,1,3} | 292 | 78 |
| **33** | {1,2,2,3} | 343 | 81 |
| **34** | {1,2,3,3} | 394 | 89 |
| **35** | {1,2,4,3} | 445 | 97 |
| **36** | {1,3,0,3} | 267 | 78 |
| **37** | {1,3,1,3} | 318 | 86 |
| **38** | {1,3,2,3} | 369 | 94 |
| **39** | {1,3,3,3} | 420 | 102 |
| **40** | {1,3,4,3} | 471 | 110 |
| **41** | {2,0,0,3} | 231 | 57 |
| **42** | {2,0,1,3} | 282 | 65 |
| **43** | {2,0,2,3} | 333 | 73 |

| | | | |
|---|---|---|---|
| **44** | {2,0,3,3} | 384 | 81 |
| **45** | {2,0,4,3} | 435 | 89 |
| **46** | {2,1,0,3} | 257 | 70 |
| **47** | {2,1,1,3} | 308 | 78 |
| **48** | {2,1,2,3} | 359 | 86 |
| **49** | {2,1,3,3} | 410 | 94 |
| **50** | {2,1,4,3} | 461 | 102 |
| **51** | {2,2,0,3} | 283 | 83 |
| **52** | {2,2,1,3} | 334 | 91 |
| **53** | {2,2,2,3} | 385 | 99 |
| **54** | {2,2,3,3} | 436 | 107 |
| **55** | {2,2,4,3} | 438 | 108 |
| **56** | {2,3,0,3} | 309 | 96 |
| **57** | {2,3,1,3} | 360 | 104 |
| **58** | {2,3,2,3} | 411 | 112 |
| **59** | {2,3,3,2} | 413 | 113 |
| **60** | {2,3,4,1} | 415 | 114 |
| **61** | {3,0,0,3} | 273 | 75 |
| **62** | {3,0,1,3} | 324 | 83 |
| **63** | {3,0,2,3} | 375 | 91 |
| **64** | {3,0,3,3} | 426 | 99 |
| **65** | {3,0,4,3} | 477 | 107 |
| **66** | {3,1,0,3} | 299 | 88 |
| **67** | {3,1,1,3} | 349 | 96 |
| **68** | {3,1,2,3} | 400 | 104 |
| **69** | {3,1,3,3} | 451 | 112 |
| **70** | {3,1,4,2} | 453 | 113 |
| **71** | {3,2,0,3} | 325 | 101 |
| **72** | {3,2,1,3} | 376 | 109 |
| **73** | {3,2,2,2} | 378 | 110 |
| **74** | {3,2,3,1} | 380 | 111 |
| **75** | {3,2,4,0} | 382 | 112 |
| **76** | {3,3,0,3} | 351 | 114 |
| **77** | {3,3,1,1} | 304 | 108 |
| **78** | {3,3,2,0} | 306 | 109 |
| **79** | {4,0,0,3} | 315 | 93 |
| **80** | {4,0,1,3} | 366 | 101 |
| **81** | {4,0,2,3} | 417 | 109 |
| **82** | {4,0,3,2} | 419 | 110 |
| **83** | {4,0,4,1} | 421 | 111 |
| **84** | {4,1,0,3} | 341 | 106 |
| **85** | {4,1,1,3} | 392 | 114 |
| **86** | {4,1,2,1} | 345 | 108 |
| **87** | {4,1,3,0} | 347 | 109 |
| **88** | {4,2,0,2} | 318 | 112 |
| **89** | {4,2,1,1} | 320 | 113 |
| **90** | {4,2,2,0} | 322 | 114 |
| **91** | {4,3,0,0} | 246 | 111 |
| **92** | {5,0,0,3} | 357 | 111 |

| | | | |
|---|---|---|---|
| **93** | {5,0,1,2} | 359 | 112 |
| **94** | {5,0,2,1} | 361 | 113 |
| **95** | {5,0,3,0} | 363 | 114 |
| **96** | {5,1,0,1} | 385 | 110 |
| **97** | {5,1,1,0} | 287 | 111 |

**6 RESULTS**

The various feasible combinations of media types to be selected to achieve optimal reach of people at minimum cost can be seen from Table 2.

The best solution among them was 477 reach of people at a cost of 107 consisting of advertising on 3 prints, 0 billboards, 4 TV, and 3 radios, thus iteration 65 in Table 2.

**7. CONCLUSIONS**

We have described the advert placement and selection problem of a company as a 0-1 knapsack programming problem. The k-set inequality algorithm was used to solve the company's advert placement and selection problem. It was observed that the solution that gave maximum achievable value was {3, 0, 4, 3}. This means that the company should spend a total cost of one hundred and seven thousand Ghana cedis (GH¢107,000) to obtain an optimal reach of four hundred and seventy seven thousand (477,000) people, consisting of placing three prints, no billboards, four TV, and three radio advertisement slots. Currently there is no such method for determining what media types to be used and in what quantity to be placed by the company. The media are chosen using guess work and by the discretion of the people in charge.

For the data used for our analysis, the company using their crude approach arrived at the following conclusion; placed a total of one print, three billboards, four TV, and three radio adverts, thus {1, 3, 4, 3}. Total reach achieved was four hundred and seventy one thousand (471,000) people at the total cost of one hundred and ten thousand Ghana cedis (GH¢110,000).

Using the more scientific Knapsack problem model for the placement and selection of the company's advert slot gives a better result. Management may benefit from the proposed approach for placement and selection of adverts to guarantee optimal reach of people. We therefore recommend that the Knapsack problem model should be adopted by the company for advert placement and media planning.

**REFERENCES**


1. Cheang B., Li H, Lim A, and Rodrigues B (2003). Nurse Rostering problems- a bibliographic survey. *European Journal of Operations Research* 151, pp 447.
2. Hall N., Hershey J, Dessler L, and Stotts R (1992). A model for making project funding decisions at the national cancer institute. *Journal of Operations Research,* volume 40, no 6, November-December 1992.
3. HamdiB. and Omer S B (2005). Capital budgeting with risk adjustment for large-scale interdependent projects. Review of social, economic and business studies, vol. 7/8, pp71-82.
4. Jolayemi J. K. (2001). A model for scheduling projects under the condition of inflation, and under penalty and reward arrangements. *Journal of Operations Research Society of South Africa*, Vol. 17, No. 2, P81-99.
5. Kanniappan P. and Thangavel K (1993). An optimization model for selection of integrated rural development program. *Journal of Operations Research Society of South Africa*, volume 9(1), pp 13-20.
6. Scorgings C. J. and UysP W (1995). Optimization of resource utilization at a university- an allocation problem. *Journal of Operations Research Society of South Afric*a, volume 11(1/2), pp 4-18.

**Appendix_1**

```
Program Knapsack
IMPLICIT NONE
Real::TotalWeight
Integer::MaxPrint, MaxBillboard, MaxTV, MaxRadio, MaxValue = 0
Integer:: i, j, k, l, n (4)
Type Bounty
   Integer:: Val
   Real::Wht
End Type Bounty
Type (Bounty) :: Print, BillBoard, TV, Radio, Sack, Current
Print = Bounty (42, 18)
BillBoard = Bounty (26, 13)
TV = Bounty (51, 8)
Radio = Bounty (49, 7)
Sack = Bounty (0, 114)
MaxPrint = 5, MaxBillBoard = 3, MaxTV = 4, MaxRadio = 3
Do I = 0, MaxPrint
     Do j = 0, MaxBillBoard
         Do k = 0, MaxTV
           Do l = 0, MaxRadio
Current% Val = l*Radio% Val + k*TV% Val + j*Billboard% Val + i*Print% Val
Current% Wht = l*Radio% Wht + k*TV% Wht + j*Billboard% Wht + i*Print% Wht
If (Current% Wht< Sack% Wht) Then
      If (MaxValue< Current% Val) Then
         MaxValue = Current% Val
         TotalWeight = Current% Wht
         N (1) = I, n (2) = j, n (3) = k, n(4) = l
      End If
End If
           End Do
       End Do
    End Do
End Do
WRITE (*, "(A, I0) ") "Optimum Value achievable is ", MaxValue
WRITE (*, "( 4(A, I0), A)" ) "This is achieved by ", n(1), "Print", n(2), "BillBoard", n(3), "TV and ", n(4), "Radio"
WRITE (*, "(A, F6.2, A, F6.2)") "The Optimum Weight is", TotalWeight
End Program Knapsack
```