African Journal of Mathematics and Computer Science Research Vol. 4(4), pp. 170 -176, April 2011
Available online at http://www.academicjournals.org/AJMCSR
ISSN 2006-9731 ©2011 Academic Journals

Full Length Research Paper

Knapsack problem: A case study of garden city radio (GCR), Kumasi, Ghana

Peasah O. K.[1], Amponsah S. K.[2] and Asamoah D.[3]*

[1]Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
[2]Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
[3]Department of Information Systems and Decision Sciences, Kwame Nkrumah University of Science and Technology, School of Business, Kumasi, Ghana.

Accepted 5 October, 2010

Packing or arrangement problems form an integral part in a man's life and cannot be out rightly ignored. Items need to be well arranged and protected to survive the move unscathed. When it is done efficiently, at least to its optimal level, space is saved. A room may become spacious when furniture, bags and other household items are arranged very well. The end result is beauty and attractiveness. Radio stations, need to arrange different types of adverts in order to maximize air time. In this paper, the knapsack problem is employed in selecting adverts to be played on air from a pile of adverts, using the garden city radio, as case study.

Key words: Knapsack, packing, adverts, heuristics scheme.

INTRODUCTION

A great variety of practical problems can be represented by a set of entities, each having an associated value, from which, one or more subsets has to be selected in such a way that the sum of the values of the selected entities is maximized, and some predefined conditions are respected. The most common condition is obtained by associating a size to each entity and establishing that the sum of the entity sizes in each subset does not exceed some prefixed bound. These problems are generally called knapsack problems*,* since they recall the situation of a hitch-hiker, having to fill up his knapsack by selecting from among various possible objects, those which will give him the maximum comfort. The entities will be called *items* and their numbers will be indicated by *n.* The value and size associated with the j^th^ item will be called *profit* and *weight,* respectively, and denoted by p_j and $w_j\,(j = 1,...,n)$.

The majority of problems considered in this paper are single knapsack problems, where one container must be filled with an optimal subset of items. The capacity of such a container will be denoted by c. We shall also consider the more general case where *m* containers, of capacities $c_j\,(i = 1,...,n)$, are available (multiple knapsack problems*).* We shall suppose that profits, weights and capacities are positive integers.

Knapsack problems have been intensively studied because they arise as sub problems in various integer programming problems and, may represent many practical situations. The most typical applications are in capital budgeting and industrial production. Various capital budgeting models have been studied by Weingartner (1963, 1968), Weingartner and Ness (1967), Cord (1964) and Kaplan (1966). Among industrial applications, the classical studies a Cargo Loading Problems (Bellman and Dreyfus, 1962) and on Cutting Stock Problems, Gilmore and Gomory (1963, 1965, 1966) is worth mentioning. More detailed reviews of applications can be found in Salkin (1975); Salkin and de Kluyver (1975) and Martello and Toth (1987).

RELATED WORKS

Pisinger (1999) presented an algorithm for knapsack

*Corresponding author. E-mail: dasamoah.ksb@knust.edu.gh problem where the enumerated core size is minimal, and the computational effort for sorting and reduction is also limited according to hierarchy. The algorithm is based on a dynamic programming approach, where the core size is extended by need, and the sorting and reduction is performed in a similar "lazy" way. Computational experiments are presented for several commonly occurring types of data instances. Experience from these tests indicates that the presented approach outperforms any known algorithm for knapsack problem, having very stable solution times.

Martello and Toth (1988) presented a new algorithm for the optimal solution of the 0 - 1 knapsack problems, which is particularly effective for large-size problems. The algorithm is based on determination of an appropriate small subset of items and the solution of the corresponding "core problem": from this, they derived a heuristic solution for the original problem which, with high probability, can be proved to be optimal. The algorithm incorporates a new method of computation of upper bounds and efficient implementations of reduction procedures. They also reported computational experiments on small-size and large-size random problems, comparing the proposed code with all those available in the literature.

Munapo (2008) presented an approach that enhances the performance of the branch and bound algorithm for the knapsack model. This is achieved by generating and adding new objective function and constraint to knapsack model, which is single constrained. The branch and bound algorithm is then applied and the total numbers of sub-problems are reduced.

Majority of algorithms for solving knapsack problems typically use implicit enumeration approaches. Different bounds based on the remaining capacity of the knapsack and items not yet included at certain iteration have been proposed for use in these algorithms. Similar methods may be used for a nested knapsack problem as long as there is an established procedure for testing whether an item inserted into a knapsack at one stage can also be inserted at the following stages. Given n different items and a knapsack of capacity, Cáceres and Nishibe (2005) algorithm solves the 0 - 1 knapsack problem using $O(nWp)$ local computation time with $O(p)$ communication rounds. Using dynamic programming, their algorithm solves locally pieces of the knapsack problem in each processor and uses a wave front approach in order to solve the whole problem. The algorithm was implemented in a Beowulf and the obtained times showed good speed-up and scalability.

The binary knapsack problem is a combinatorial optimization problem in which a subset of a given set of elements needs to be chosen in order to maximize profit, given a budget constraint. Das (2003) used a stochastic version of the problem in which the budget is random. They proposed two different formulations of this problem, based on different ways of handling infeasibility, and propose an exact algorithm and a local search-based heuristic to solve the problems represented by these formulations. The results were presented from some computational experiments.

The knapsack problem is believed to be one of the "easier" -hard problems. Not only can it be solved in pseudo-polynomial time, but also decades of algorithmic improvements have made it possible to solve nearly all standard instances from the literature. Pisinger (2005) gave an overview of all recent exact solution approaches, and to show that the knapsack problem is still hard to solve for these algorithms for a variety of new test problems. These problems are constructed either by using standard benchmark instances with larger coefficients, or by introducing new classes of instances for which most upper bounds perform badly. The first group of problems challenges the dynamic programming algorithms while the other groups of problems are focused towards branch-and-bound algorithms. Numerous computational experiments with all recent state-of-the-art codes are used to show that knapsack problem (KP) is still difficult to solve for a wide number of problems. One could say that the previous benchmark tests were limited to a few highly structured instances, which do not show the full characteristics of knapsack problems.

The 0/1 knapsack problem is well-known and it appears in many real domains with practical importance. The problem is NP-complete. The multi-objective 0/1 knapsack problem is a generalization of the 0/1 knapsack problem in which many knapsacks are considered. Many algorithms have been proposed in the past four decades for both single and multi-objective knapsack problem. A new evolutionary algorithm for solving multi-objective 0/1 knapsack problem was proposed by Groan (2003). This algorithm used a ε-dominance relation for direct comparison of two solutions. Some numerical experiments are realized using the best and recent algorithms proposed for this problem. Experimental results show that the new proposed algorithm outperforms the existing evolutionary approaches for this problem.

The knapsack problem is believed to be one of the "easier" NP-hard problems. Not only can it be solved in pseudo-polynomial time by the dynamic programming, but also decades of algorithmic improvements have made it possible to solve nearly all standard instances. The current most successful algorithm for the knapsack problem was presented by Martello, Pisinger and Toth. The algorithm can be seen as a combination of many different concepts and is hence called Combo but they noticed that it is difficult for Combo to solve some instances. Ken'ichi (2004) proposed a new genetic algorithm for knapsack problem. The algorithm can adjust solution spaces in consideration of the stability of each item which can obtain from the greedy algorithm. They applied the proposed method to those difficult instances, and test the effectiveness.

Puchinger (2006) presented a newly developed core concept for the multidimensional knapsack problem (MKP) which is an extension of the classical concept for the one-dimensional case. The core for the multidimensional problem is defined in dependence of a chosen efficiency function of the items, since no single obvious efficiency measure is available for MKP. An empirical study on the cores of widely-used benchmark instances is presented, as well as experiments with different approximate core sizes. Furthermore, they described a memetic algorithm and a relaxation guided variable neighborhood search for the MKP, which are applied to the original and to the core problems. The experimental results show that given a fixed run-time, the different meta-heuristics as well as a general purpose integer linear programming solver yield better solution when applied to approximate core problems of fixed size.

Fontanari (1995) investigated the dependence of the multi-knapsack objective function on the knapsack capacities and on the number of capacity constraints P, in the case when all N objects are assigned the same profit value and the weights are uniformly distributed over the unit interval. A rigorous upper bound to the optimal profit is obtained, employing the annealed approximation and then, compared with the exact value obtained through the Lagrangian relaxation method. The analysis is restricted to the regime where N goes to infinity and P remains finite.

A revenue management model was designed, in which demand is stimulated by moving a number of product units to a promotion space, rather than by price changes. Thus, they addressed the problem of filling a promotion location with limited space to maximize the expected revenue, which we have termed the knapsack problem for perishable Items (KPPI). Examples of the promotion space include shelves close to the cash register, promotion kiosks, or a depot used for selling via the Internet.

They solved the KPPI using problem decomposition into single product unit sub problems. A natural mathematical setting for the KPPI sub problems is the restless bandit model of a fundamental stochastic model for resolving a trade-off between exploration and exploitation in an optimal fashion. In our model, the bandits (perishable items) are restless, because they can get sold regardless of being in the knapsack or not, the time horizon is finite, and we are to select more than one item for the knapsack, which is allowed to be filled partially, due to the heterogeneity of the items. Each product unit is assigned a promotion priority index, which captures the opportunity cost of promotion, as a function of its price, lifetime, expected demand, and expected promotion power. These indices are then used for each item as objective-function value coefficients in a (classic) knapsack problem, whose solution yields a well-performing heuristic for the KPPI. They thus mix up two models: the restless bandit problem and the knapsack problem.

Optimal dynamic promotions, under time-homogeneous demand, suppose that the item perishes in T time periods, implying a final cost c > 0 if not sold before. Let 1 – p be the probability that a promoted item is sold in one period, and (1 – q) be that of a non-promoted item (q > p). Future costs are discounted by the one-period discount factor β. The next proposition asserts that, optimally, an item with lower probabilities of being sold when not promoted is assigned a higher promotion priority index, and that the index is non-decreasing over time. Hence, once the item is optimally chosen for promotion, then it should remain promoted until it perishes.

PROBLEM FORMULATION

Suppose an FM station wants to play *n* number of adverts, each worth p_i Ghana cedis and a duration (time to broadcast) of w_i seconds, the station wants to broadcast as many valuable adverts as possible, but it can broadcast by not exceeding a total time of c seconds, which adverts should be broadcast to obtain optimal income under the constraints of *c* seconds? This problem can be formulated as a 0 - 1 knapsack problem as explained further.

Knapsack algorithm

The knapsack problem is the classic integer linear programming problem with a single constraint. The 0-1 knapsack problem (KP) is a problem of choosing a subset of the *n* items such that the corresponding profit sum is maximized without having the weight sum to exceed the capacity *b*. This may be formulated as:

Max $p_1x_1+p_2x_2+...+p_nx_n$

That is: $z=\sum_i^n p_ix_i$

Subject to: $w_1x_1+w_2x_2+...+w_nx_n \le c$

That is: $\sum_i^n w_ix_i \le c$

$$x_i = \begin{cases} 1, \; if\ the\ i^{th}\ item\ has\ been\ selected \\ 0, otherwise \end{cases}$$

Since profits and weight are positive, it will be supposed, without loss of generality, that is:

$\sum_{i=1}^{n} w_i > c$ $\qquad w_i \le c$ (i=1,…,n).

Table 1. List of Commercials for prime time for a day collected from garden city radio.

Company name(x_i)	Duration of advert in seconds (w_i)	Value/Cost per advert (c_i)
A001	30	9.00
A001	30	9.00
A002	60	17.00
A003	25	9.00
A003	25	9.00
A003	25	9.00
A004	30	9.00
A004	30	9.00
A004	30	9.00
A004	30	9.00
A004	30	9.00
A005	29	9.00
A005	29	9.00
A005	29	9.00
A006	45	14.00
A006	45	14.00
A007	57	17.00
A008	30	9.00
A008	30	9.00
A008	30	9.00
A008	30	9.00
A010	30	9.00
A010	30	9.00
A010	30	9.00
A010	30	9.00
A011	45	14.00
A011	45	14.00
A011	45	14.00
A012	40	14.00
A012	40	14.00
A012	40	14.00
A013	30	9.00
A013	30	9.00
A013	30	9.00
A014	30	9.00
A014	30	9.00
A015	30	9.00
A015	30	9.00

The c_i and a_i as seen in Table 1 represents the value and weight of selecting item i respectively for inclusion in the knapsack. The constant b represents the maximum weight that the knapsack is permitted to hold.

KP is NP-hard and is a well-known problem and several exact and heuristic algorithms have been proposed for its solution. The exact algorithm can be subdivided into two classes: branch and bound methods (Kolesar, 1980; Greenberg and Hegerich, 1970; Horowitz and Sahni, 1997; Fayard and Plateau, 1975). The performance of both classes largely depends on the size of the problems to be solved. This can generally be decreased by applying reduction procedures (Ingargiola and Korsh, 1973) so as to fix the value of as many variables as possible.

One of the essential ingredients of the implicit enumeration algorithms and the reduction procedures is the quality of the upper bounds used in the computation.

Other types of Knapsack Problems are: Subset-sum problem; the change-making problem; multiple knapsack problem; multi-dimensional knapsack problem.

Using heuristic scheme in solving knapsack problem

According to Amponsah and Darkwa (2009), heuristic scheme may be used to solve the knapsack problem instead of the method of branch and bound using the steps as:

Step 1: Input the vector of weight $\{w_j\}$ and items value $\{c_j\}$

Step 2: Input random initial solution $S_o \in \{0,1\}^n$ and check for feasibility of S_o by using the equation $\sum_{j=1}^{n} w_j x_j \leq b$. If S_o is not feasible, discard it and choose another S_o.

Step 3: Find a feasible solution and compute the objective function value $f(S_o)$ by using the objective function $Max \sum_{j=1}^{n} c_j x_j$

Step 4: Obtain a new solution $S_1 \in \{0,1\}^n$ from S_0 by flip operation and check for feasibility, continue flip operations until the solution S_1 so obtained is feasible. Compute the objective function value $f(s_1)$. If $f(S_1) > f(S_0)$ then put $S_0 = S_1$ else retain S_0 and descend S_1.

Step 5: Repeat step 3 for all feasible S_i

Data collection

Garden city radio (GCR) has various ways of generating income. These include sponsorship of programmes, social and funeral announcements, advertisements, among others. We focused on advertisements which are slotted before, during or after programmes.

GCR has three categories of time for adverts: Prime time (6am, 1pm, 6pm news); adjacencies (five minutes before and after news prime time); break in programmes (adverts within a programme). Each of these categories have different rates and for that matter, constitutes a source of income to the radio station. The rate also depends on the duration of the advert.

Clearly, the pile up of commercials at the FM-station at any point in time is a typical case of a knapsack problem.

Software developed (resource optimizer)

We used Visual Basic dot Net to write codes that was developed into software using Heuristic method. The features of software developed enables: Data entry, which can be entered manually, loaded from previously saved file or generated; data verification; specify extra options for the algorithm; generation of initial solution and flip to generate other solutions to obtain an optimal solution; allows any number of Iterations; report generated can be viewed or saved for future used.

RESULTS

Prime time

The software generated an initial data structure of

1. S_0= {0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 0 0 1 1 1 0}

2. The objective function value corresponding to S_0 is:

F(S_0) = GH□ 168.00

3. Number of loops to obtain optimal solution = 115
2. Optimal Solution: S_{115} ={0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1}
4. Total number of adverts selected = 26 out of 38
5. F(S_{115}) = GH □ 282.00
6. Total time available for adverts = 900 sec
7. Total time for selected adverts = 890 sec

Selected adverts is as seen in Table 2.

News Adjacencies:

1.S_0={010000111011000001101001001101101101011010 0001011011010101110111 1 000011}
2. The objective function value corresponding to S_0 is F(S_0) = GH□ 294.00
3. Number of loops to obtain optimal solution = 332
4. Optimal Solution: S_{332} = {0001100111111011 1o110111 1111101110 011001110101101 110111100110010010111}
5. Total time available for adverts = 1800 sec (30min)
6. Total number of adverts = 46 out of 68
7. Total time for selected adverts = 1795 sec
8. F(S_{332}) = GH □ 382.00

Selected adverts for news adjacencies is as seen in Table 3.

Conclusion and future Work

We have described the pile of adverts at GCR as a clear case of knapsack problem. Adverts were selected from a pile of adverts based on their value but now, effective, efficient and more scientific means can now be used. Piling of adverts is reduced significantly since more

Table 2. Selected adverts for primetime news (06:00 to 06:30 and 013:300 to 18:30 hours GMT). Total time available is 900 s (15 min).

Adverts	Time (s)	Number of adverts	Total time	Amount (Ghana cedis)
A003	25	2	50	18.00
A004	30	4	120	36.00
A005	29	2	58	18.00
A006	45	2	90	28.00
A007	57	1	57	17.00
A008	30	2	60	18.00
A010	30	2	60	18.00
A011	45	3	135	42.00
A012	40	3	120	42.00
A013	30	3	90	27.00
A015	25	2	50	18.00
Total		26	890	282.00

Table 3. Selected adverts for news adjacencies (1800 s; 30 min).

Advert	Time (s)	Number of adverts	Total time	Amount (Ghana cedis)
B001	45	2	90	18
B002	30	2	60	14
B007	30	2	60	14
B008	30	2	60	14
B009	50	3	150	33
B010	60	3	180	33
B011	45	3	135	27
B012	30	5	150	35
B013	30	2	60	14
B014	30	6	180	42
B015	29	5	145	35
B016	30	1	30	7
B017	45	3	135	27
B018	30	2	60	14
B020	60	5	300	55
Total		46	1795	382

adverts are played within a given period of time. Income generated from playing advert shot up. Software developed can be used to solve knapsack problem using HS to select commercials from a pile, given limited time constraint.

We recommend to GCR to use the software in selecting adverts to be played on air. There are situations where too many adverts spoil the beauty of programmes and make it boring. In this case, it is considered as more than one constraint (that is, both adverts limit and time limit, where the adverts number and limited time are not related); we get the multiply-constrained knapsack problem. We recommend that in future such situation should be considered.

REFERENCES

Amponsah SK, Darkwah FK (2009). Lecture notes on Operational Research. Kwame Nkrumah University of Science and Technology, Kumasi.

Bellman R, Dreyfus SE (1962). «Applied Dynamic Programming», Princeton University Press, Princeton, N.J.

Cáceres EN, Nishibe C (2005). «0-1 Knapsack Problem: BSP/CGM Algorithm and Implementation», IASTED PDCS: 331-335.

Cord J (1964). «A Method for Allocating Funds to Investment Projects when Returns are Subject to Uncertainty», Manage. Sci.10: 335-341.

Das S, Ghosh D (2003). «Binary knapsack problems with random budgets», J. Oper. Res. Soc., 54(14): 970-983.

Fayard D, Plateau G (1975). «Resolution of the 0 – 1 Knapsack Problem: Comparison of Methods» Math. Programming 8, 272 – 307.

Fontanari JF (1995). «A statistical analysis of the knapsack problem» J. Phys. A: Math. Gen., 28 4751-4759

Gilmore PC, Gomory RE (1963). «A Linear Programming Approach to the Cutting Stock Problem II» Oper. Res., 11: 863-888.
Gilmore PC, Gomory RE (1965). «Multi-Stage Cutting Stock Problems of Two and More Dimensions» Oper. Res., 13, 94-120.
Gilmore PC, Gomory RE (1966). «The Theory and Computation of Knapsack Functions» Oper. Res., 14: 1045-1074.
Greenbrg H, Hagerich RL (1970). «A Branch Search Algorithm for Knapsack Problem» Manage. Sci., 16: 327-332.
Groan C, Oltean M, Dumitrescu D (2003). «A new evolutionary algorithm for the multiobjective 0/1 knapsack problem» Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics – ICTAMI, Alba Iulia.
Ingargiola GP, Korsh JFA (1973). «Reduction Algorithm for Zero-One Single Knapsack Problems» Manage. Sci., 20, 460-463.
Kaplan S (1966). «Solution of the Lorie-Savage and Similar Integer Programming Problems by the Generalized Lagrange Multiplier Method», Oper. Res., 14, 1130-1136.
Ken'ichi I, Ryohei S, Mitsuo G (2004). «Adjustment Type GA for Hard Knapsack Problems» Faji Shisutemu Shinpojiumu Koen Ronbunshu (CD-ROM), 20: 717-722.
Kolesar PJ (1997). «A Branch and Bound Algorithm for Knapsack Problems» Math. Oper. Res., 4: 339-356.
Martello S. and Toth P. (1987). Algorithms for Knapsack Problems, Annals of Discrete Mathematics pp213 – 258, Elsevier Science Publishers B.V. (North-Holland).
Martello S, Toth P (1988). «A New Algorithm for the 0-1 Knapsack Problem» Manage. Sci., 34(5): 633-644.
Munapo E (2008). «The efficiency enhanced branch and bound algorithm for the knapsack model» Adv. Appl. Math. Anal. ISSN 0973-5313, 3(1): 81-89
Pisinger, D., 1999. An exact algorithm for large multiple knapsack problems. Eur. J. Oper. Res., 114: 528-541.
Pisinger D (2005). «Where are the hard knapsack problems?» Computers and OR 32: 2271-2284.
Puchinger J, Raidl GR, Pferschy U (2006). «The Core Concept for the Multidimensional Knapsack Problem» in evolutionary computation in combinatorial optimization – evocop.
Salkin HM (1975). «Integer Programming» Addison-Wesley, New York.
Salkin HM, De Kluyver CA (1975). «The Knapsack Problem: A Survey» Naval Res. Logistics Q., 22: 127-144, 0244730666.
Weingartner HM (1968). «Capital Budgeting and Interrelated Projects: Survey and Synthesisu. Manage. Sci., 12: 485-516.
Weingartner HM, Ness DN (1967). «Methods for the Solution of the Multi-Dimensional 0-1 KnapsackProblem» Oper. Res., 15: 83-103.