Applied Mathematics and Computational Intelligence
Volume 10, No.1, Dec 2021 [78-86]

Comparison Between Exact Optimization and Heuristics Approaches for Maximizing Benefit of Point Redemption: A Knapsack Problem

Ahmad Afiq Iqmal Ahmad Saidi[1], Lai Jing Yee[1], Ivy Lee Xin Zhen[1] and Syariza Abdul-Rahman[1]*

[1]School of Quantitative Science, Universiti Utara Malaysia, Malaysia.

* Corresponding author : syariza@uum.edu.my

Received: 6 January 2021; Accepted: 5 April 2021; Available online: 12 April 2021

ABSTRACT

Knapsack problem (KP) is a well-known NP-hard combinatorial optimization problem which seeks to assign several items into a limited capacity knapsack with the objective to maximize the total profit. KP has been used to model many real-life problems. In this study, a point redemption problem based on a membership Clubcard is considered. Customers with certain amount of points sometimes having problem to redeem a selection of coupons to maximize the value of their money. Thus, this problem can be modelled as a knapsack problem and solved by using optimization or heuristics approaches. The objective of this study is to evaluate the solutions produced by greedy techniques compared to optimal solution. This study presents solution comparison between optimal solution and solutions produced by three greedy heuristics which are greedy by profit, greedy by weight and greedy by profit density. Two different total points of customers were considered with fifteen different coupons for selection. Comparison for two total points of customers found that the greedy by profit density is the best greedy method so far for maximizing the value of money because the total value obtained is the nearest to the optimal solution. This is because this greedy method considers two criteria i.e. coupon values and points in calculating the density which give benefit for identifying suitable coupons to be selected. It can be concluded that this point redemption problem can be modelled as a knapsack problem and heuristic approaches can be used to find reasonable solution which is comparable to the optimal solution.

Keywords: Greedy heuristics, Knapsack problem, Maximum benefit, Optimization, Point redemption.

1 INTRODUCTION

Knapsack problem can be defined as a situation when there are few items with known sizes and profits that need to be loaded into a knapsack with limited capacity. The objective of this problem is to maximize the total profit, with the capacity as the constraints. This problem is static and deterministic, since all items that is considered at the time, with the size and the profit are known a priori [1]. In this problem, given a set of n items, and each item j, has an integer profit p_j and an integer weight w_j. In this case, the problem is to choose a subset of items such that their overall profit is maximized, while the overall weight does not exceed a given capacity c. The binary decision variables x_j are being used in order to present whether the item j is included in the knapsack or not. It is assumed that all the profits and weights are positive, where all the weights are smaller than the capacity c and the overall weight of the items exceeds c, without loss of generality [2]. The knapsack problem is classified as NP-hard [3] which means that unknown polynomial time algorithm for searching an optimum solution in complexity theory. In other words, no algorithm can solve the problem in a logical manner or within an acceptable time limit.

Knapsack problem has been used to solve real life problems in several areas such as project selection [4], cutting stock [5], resource allocation [6], and capital budgeting [7]. Project portfolio selection problem was presented in [4] where analytical network process was used to find the weight of criteria in the problem. The project portfolio selection is based on a knapsack problem where a list of projects needs to be selected based on limited budget. The cutting stock problem in [5] is to determine how to cut the rolls into smaller widths to fulfil the orders in such a way to minimize the amount of waste. There are many possible combinations of cut to be considered, known as pattern. The knapsack problem is incorporated to identify new patterns to add. Allocation of a fixed amount of resources to activities to minimize the cost incurred by the allocation is called as resource allocation problem [6] and it is also considered as knapsack problem. In a capital budgeting problem, a set of investment projects are selected which can maximize shareholder values, subject to limitations on budgets or other resources consumed [7].

There are various methods to solve knapsack problem such as optimization, heuristics and metaheuristics. [8] solved penalized knapsack problems, a variant of a knapsack problem which associate penalty in the objective function. The problem was solved by using an optimization algorithm to obtain optimal solution for several generated datasets. The multi-dimensional knapsack problem was solved in [9] by using heuristic approach based on Linear Programming relaxation with the aim to find a subset of items with maximum value that satisfies several knapsack constraints. Results tested on benchmark datasets show that the proposed approach is comparable with the best results presented in the literature. A heuristic approach namely as Recursive Core Heuristic was proposed to solve multiple knapsack problem [10]. The proposed heuristic is a recursive method that performs computation on the core of knapsacks and was tested on random generated data. The results were compared to Martello and Toth Heuristic Method and showed better performance [10].

Tabu search approach was used to solve capital budgeting problem at a semiconductor company where the problem was modelled as a knapsack problem [7]. The study found that the best solutions obtained by tabu search approach are comparable to the global optimum solution generated by using optimization approach. The memory structure in tabu search approach that can forbid certain attribute gives advantage to the solution search in finding good solution quality. Study by [11] solved a knapsack problem by using several metaheuristic algorithms which are tabu search, scatter search and local search algorithms. Comparisons were made based on execution time, solution quality and relative difference to best known quality. It is found that scatter search used the least execution time while tabu search yields the least deviation compared to other metaheuristics.

The quadratic multiple knapsack problem was investigated in [12] based on theoretical properties and dominances among linear models. Surrogate and Lagrangian relaxations were successfully applied to a quadratic formulation and to a Level 1 reformulation linearization that leads to a decomposable structure. A bilevel knapsack problem arising from a Stackelberg game setting was proposed in [13] which consist of a follower and a leader to decide on the optimal utilization of a bounded resource. The trade-off between incentives and shared profits between follower and leader are required. Natural Greedy-type heuristics are investigated for a lower-level player with bounded computation power. The complexity of the resulting problems deriving several inapproximability results and pseudopolynomial algorithms and integer linear programming models were also discussed.

A hybrid algorithm based on k-nearest neighbor (KNN) quantum and cuckoo search algorithm was proposed to solve a multidimensional knapsack problem in [14]. KNN technique was integrated to quantum cuckoo search algorithm in improving the solution quality. Experimentation on well-known benchmark dataset of the problem shows that the proposed hybrid algorithm is superior. A variant of multi-knapsack and multidimensional knapsack problems namely as multiple multidimensional knapsack problem with family-split penalties was proposed in [15] which arises in several real problem such as resource management of distributed computing contexts and service-oriented architecture environments. The new problem considers selection of families of items to limited resources to maximize profit. Split penalties are acquired whenever items in the families are assigned to different knapsacks. The study proposed an exact algorithm based on Benders' cuts approach and tested on benchmark problems. The results show the effectiveness of the proposed algorithm compared to other solvers. Fuzzy environment for allocation optimization problems was investigated in [16] by developing a combination of penalty function method and Nelder and Mend's algorithm. The study presents numerical solutions of crisp reliability optimization problems of nonlinear programming problem and shows its effectiveness compared to fuzzy solutions.

In this study, a point redemption problem based on a membership Clubcard is considered. Customers with certain amount of points sometimes having problem to redeem on a selection of various coupons which can maximize the value of their money. Thus, this problem can be modelled as a knapsack problem where the objective is to maximize the value of the money and the selection is based on the total amount of points owned by customers. The problem can be solved by using optimization or heuristics approaches depending on the problem size and complexity. The objective of this study is to evaluate the solutions produced by greedy techniques compared to optimal solution to see how good the solutions produced by the greedy heuristics compared to the optimal solution.

This paper is organized as follows. In Section 2, we briefly explained on the point redemption problem as knapsack problem and its formulation. Section 3 presents the greedy heuristics approaches. Section 4 describes the experimental results and discussion. Finally, the conclusion is provided in Section 5.

2 PROBLEM DESCRIPTION

We consider a point redemption problem based on a membership and reward programme of a company. By using membership card, it enables customers to save and earn more rewards when they shop with the membership card at participating stores. In each transaction, the membership card gives points to the customers that can be converted to cash vouchers for future use. Every RM1 spent, the customers will be given one point that can be accumulated. In this problem, customers have a certain amount of points to be redeemed on a selection of various coupons. In each coupon, it requires different redemption points. The problem can be formulated as a Knapsack Problem and the mathematical model is presented as below,

$$Max \sum_{i=1}^{n} v_i x_i \tag{1}$$

Subject to

$$\sum_{i=1}^{n} w_i x_i \leq C \tag{2}$$

$$x_i \in \{0,1\}$$

x_i is a 0|1 decision variable of coupon i where 1 is if coupon i is selected and 0 otherwise, $v_i$the value (in Ringgit Malaysia (RM)) of coupon i, w_i is the points required for coupon i, n is the number of coupons considered for selection and C is the total amount of points available, where $i = \{1, 2, 3, \dots n\}$. The objective is to maximize the monetary value of selecting the coupons as presented in Equation 1. Equation 2 is the restriction of the total amount of points available. Since the total amount of points is limited, thus only coupons that can maximize the value of the money will be chosen.

3 METHODOLOGY

In this study, list of coupons and points required were obtained from a membership and reward programme of a company, while the value in RM of each coupon is based on the price of the item in the market. This study extracted only fifteen coupons as for experimentation. The data of the fifteen coupons ($n = 15$), coupon value (RM) (v_i) and coupon point (w_i) are presented in Table 1. This problem is modelled as an integer programming model based on Equations 1 and 2. There are two total amounts of available points considered in this study which are $C = 22{,}000$ and $C = 19{,}000$ for testing the sensitivity of the proposed model. These values are determined by chance. The integer programming model was solved by using Excel Solver and produced optimal solution.

Table 1 : List of coupons, points and value (RM).

Coupon	Value (RM)	Points required
Sushi King Discount Coupon	6.60	600
RM 20 Accessories & Shoes off	20.00	2,000
RM 4 Cash Voucher	4.00	770
Secret Recipe	10.00	1,000
Online RM 10 eVoucher	10.00	1,500
Online RM 20 eVoucher	20.00	3,800
Moms & Kids Products RM 20 off	20.00	2,000
Mobile & Gadgets RM 60 off	60.00	2,500
Skincare Products RM 20 off	20.00	2,000
Home & Living RM 12 off	12.00	2,300
Electronic Devices RM 40 off	40.00	7,800
Kit Kat Products RM 13 off	13.00	600

Online RM 40 eVoucher	40.00	3,000
Digi RM 20 Prepaid Top Up	20.00	2,000
RM 20 Cash Voucher	20.00	3,500

Three greedy heuristics which are greedy by profit, greedy by weight and greedy by profit density were used to construct the solution and were compared with the optimal solution of integer programming model. The algorithm for the three greedy heuristics is presented in Figure 1. The coupons are ranked based on the chosen greedy approach. Greedy by profit will rank the coupons from the highest profit to the lowest, greedy by weight will rank the coupons from the lowest points to the highest, while greedy by profit density will rank the coupons from the highest profit density to the lowest. The profit density is calculated as the ratio of coupon value and number of points for each coupon i. Then, based on the rank each coupon is assessed whether they can be selected or not based on the remaining capacity C i.e. total amount of available points, else the remaining capacity will be reset as previous. The next coupon in the rank will be examined whether it can be selected or not. The process is repeated until all coupons have been examined whether they can be selected based on the limited amount of available points. Finally, the *Zval* or the objective function value and total point used are returned. The heuristics approaches were coded in Matlab 2017.

```
Coupon i = {1, 2, 3, ..n}, v_i = {v_1, v_2, v_3, ... v_n}, w_i = {w_1, w_2, w_3, ... w_n}, Capacity C
totalCap = 0, Zval = 0, h = {1, 2, 3, ..n}
Rank all items according to the chosen greedy approach:
    Greedy by Profit: Rank the variables from highest to lowest profit.
    Greedy by Weight: Rank the variables from lowest to highest weight.
    Greedy by Profit Density: Rank the variables from highest to lowest profit
    density.
Set the rank as h = 1 to n
For each h
    i = h
    totalCap = totalCap + w_i
    if totalCap ≤ C
        Zval = Zval + v_i
    Else
        totalCap = totalCap - w_i
```

Figure 1: The algorithm for the greedy heuristics.

4 RESULTS AND DISCUSSION

Tables 2 and 3 present the comparison of results of the proposed problem solved by using optimization and greedy heuristics for C=22,000 and C=19,000. The bold font presents the best solution from all methods, while the bold and italic presents the best solution from the greedy methods. For each column method, value 1 means the coupon is chosen and 0 is otherwise. Based on the results presented in Table 2, the greedy by profit density is the best among the three greedy methods where greedy by profit density obtained the total value of RM243.60 and 21,470 of the total point used. These values are the same as presented by the optimal solution. Based on the total value, the second best greedy method is goes to greedy by weight with the total cost of RM235.60 and total point used is 20,270 while, greedy by profit presents the worst total value which is RM213 and total point used is 21,700.

Table 2: Comparison of results for C=22,000.

Coupon	Optimal Solution	Greedy by Profit	Greedy by Weight	Greedy by Profit Density
Sushi King Discount Coupon	1	0	1	1
RM 20 Accessories & Shoes off	1	1	1	1
RM 4 Cash Voucher	1	0	1	1
Secret Recipe	1	0	1	1
Online RM 10 eVoucher	1	0	1	1
Online RM 20 eVoucher	0	1	0	0
Moms & Kids Products RM 20 off	1	1	1	1
Mobile & Gadgets RM 60 off	1	1	1	1
Skincare Products RM 20 off	1	0	1	1
Home & Living RM 12 off	0	0	1	0
Electronic Devices RM 40 off	0	1	0	0
Kit Kat Products RM 13 off	1	1	1	1
Online RM 40 eVoucher	1	1	1	1
Digi RM 20 Prepaid Top Up	1	0	1	1
RM 20 Cash Voucher	1	0	0	1
Objective function, total value (RM)	**243.60**	213.00	235.60	***243.60***
Total point used	**21,470**	21,700	20,270	***21,470***
C = 22,000				

The solutions for C=19,000 by using greedy methods shows a different pattern. Based on the results presented in Table 3, the best solution for the greedy methods is obtained by the greedy by profit density with the total value of RM223.60 and the total point used is 17,970. It is followed by greedy by profit and greedy by weight with the total values of RM203 and RM195.60, and the total point used are 18,900 and 17,270, respectively. The optimal solution obtained for C=19,000 is RM225.60 for the total value and 18,770 for the total point used.

Table 3: Comparison of results for C=19,000.

Coupon	Optimal Solution	Greedy by Profit	Greedy by Weight	Greedy by Profit Density
Sushi King Discount Coupon	1	0	1	1
RM 20 Accessories & Shoes off	1	1	1	1
RM 4 Cash Voucher	1	0	1	1
Secret Recipe	1	1	1	1
Online RM 10 eVoucher	0	0	1	1

Online RM 20 eVoucher	0	0	0	0
Moms & Kids Products RM 20 off	1	1	1	1
Mobile & Gadgets RM 60 off	1	1	1	1
Skincare Products RM 20 off	1	0	1	1
Home & Living RM 12 off	1	0	1	0
Electronic Devices RM 40 off	0	1	0	0
Kit Kat Products RM 13 off	1	1	1	1
Online RM 40 eVoucher	1	1	0	1
Digi RM 20 Prepaid Top Up	1	0	1	1
RM 20 Cash Voucher	0	0	0	0
Objective function, total value (RM)	**225.6**	203	195.6	***223.6***
Total point used	**18,770**	18,900	17,270	***17,970***
C = 19,000				

Based on the proposed solution for the two different total amounts of available points, it shows that greedy by profit density is the best among the greedy methods. Greedy by profit density considers the ratio of value and point for item *i.* This density value provides the value per unit size which is more accurate calculation compared to greedy by profit and greedy by weight. Since this problem involved a minimum number of coupons, thus optimal solution by using optimization can be obtained and compared. For the case of large problem size, optimization has limitation for problem solving therefore only heuristics or metaheuristics can be used. For future planning, large number of coupons can be considered, and heuristics or metaheuristics could be employed as solution methodology.

5 CONCLUSIONS

This study presents a point redemption problem where this problem can be considered as a knapsack problem. This problem considers a list of coupons that need to be selected while considering limited amount of available total points, with the aim to maximize the total value in RM in selecting the coupons. The objective of this study is to evaluate the solutions produced by greedy techniques compared to optimal solution. Two different total points of customers were considered with fifteen different coupons for selection. The problem was modelled as Integer Programming and solved by using Excel Solver to produce optimal solution. Three greedy heuristics which are greedy by profit, greedy by weight and greedy by profit density were used to construct the solution and were compared with the optimal solution of integer programming model. Comparison for two total points of customers found that the greedy by profit density is the best greedy method so far for maximizing the value of money because the total value obtained is the nearest to the optimal solution. This is because this greedy method considers two criteria i.e. coupon values and points in calculating the density which give benefit for identifying suitable coupons to be selected. This study found that this problem can be modelled as a knapsack problem and heuristic approaches can be used to find reasonable solution which is comparable to the optimal solution. For future study, large data size can be considered and metaheuristics such as simulated annealing and tabu search can be used as solution methodologies.

REFERENCES

[1] A. J. Kleywegt and J. D. Papastavrou, "The Dynamic and Stochastic Knapsack Problem," *Operations Research*, vol. 46, no. 1, pp. 17-35, 1998.

[2] D. Pisinger, "Where are the Hard Knapsack Problems?," *Computers & Operations Research*, vol. 32, no. 9, pp. 2271-2284, 2005.

[3] A. Fréville, "The Multidimensional 0–1 Knapsack Problem, An Overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1-21, 2004.

[4] A. K. Haddadh, S. H. Yakhchali and Z. Jalili bal, "MCDM Techniques and Knapsack Approach for Project Selection Problem: A Case Study," *International Journal of Humanities and Management Sciences*, vol. 4, no. 4, pp. 397-400, 2016.

[5] Z. Degraeve and M. Peeters, "Optimal Integer Solutions to Industrial Cutting-Stock Problems: Part 2, Benchmark Results," *INFORMS Journal on Computing*, vol. 15, no. 1, pp. 58-81, 2003.

[6] N. Katoh and T. Ibaraki, *Resource Allocation Problems*, in Handbook of Combinatorial Optimization, D. Z. Du. & P. M. Pardalos, Ed. Boston, MA: Springer, pp. 905-1006, 1998.

[7] S. Abdul-Rahman and V. Chun Yung, *Comparison of Optimization and Tabu Search Approach for Solving Capital Budgeting Based on Knapsack Problem*, in Recent Applications in Quantitative Methods and Information Technology, N. Aziz, S. Abdul-Rahman & N. Zainal Abidin, Ed. Sintok: UUM Press, pp. 1-24, 2019.

[8] A. Ceselli and G. Righini, "An Optimization Algorithm for a Penalized Knapsack Problem," *Operations Research Letters*, vol. 34, no. 4, pp. 394-404, 2006.

[9] K. Fleszar and K. S. Hindi, "Fast, Effective Heuristics for the 0–1 Multi-Dimensional Knapsack Problem," *Computers & Operations Research*, vol. 36, no. 5, pp. 1602-1607, 2009.

[10] M. E. Lalami, M. Elkihel, D. E. Baz and V. Boyer, "A Procedure-Based Heuristic for 0-1 Multiple Knapsack Problems," *International Journal of Mathematics in Operational Research*, vol. 4, no. 3, pp. 214-224, 2012.

[11] D. Sapra, R. Sharma and A. P. Agarwal, "Comparative Study of Metaheuristic Algorithms Using Knapsack Problem," in *Proc. 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, Noida*, 2017, pp. 134-137.

[12] L. Galli, S. Martello, C. Rey and P. Toth, "Polynomial-size Formulations and Relaxations for the Quadratic Multiple Knapsack Problem," *European Journal of Operational Research*, vol. 291, no. 3, pp. 871-882, 2021.

[13] U. Pferschy, G. Nicosia, A. Pacifici and J. Schauer, "On the Stackelberg Knapsack Game," *European Journal of Operational Research*, vol. 291, no. 1, pp. 18-31, 2021.

[14] J. García and C. Maureira, "A KNN Quantum Cuckoo Search Algorithm Applied to the Multidimensional Knapsack Problem," *Applied Soft Computing*, vol. 102, no. 107077, 2021.

[15] S. Mancini, M. Ciavotta and C. Meloni, "The Multiple Multidimensional Knapsack with Family-Split Penalties," *European Journal of Operational Research*, vol. 289, no. 3, pp. 987-998, 2021.

[16] F. Jameel, R. A. Zaboon and H. A. Hashim, "Solution of Fuzzy Reliability Allocation Optimization Problems," *Journal of Applied Mathematics and Computational Intelligence*, vol. 4, no. 1, pp. 325–340, 2015.