A study report on solving 0–1 knapsack problem with imprecise data

Conference Paper · January 2017

DOI: 10.1109/ICCCI.2017.8117698

2 authors, including:

Jayashree Padmanabhan
Anna University, Chennai
60 PUBLICATIONS **363** CITATIONS

SEE PROFILE

A study Report on Solving 0-1 Knapsack Problem with Imprecise Data

Jayashree Padmanabhan, Swagath S
Madras Institute of Technology, Anna University
Chennai , Tamil Nadu
pjshree12@gmail.com

Abstract—**Knapsack problem has been widely studied due to its broad applications in many fields of science. Dealing with imprecise data arises in many real world applications. A study report is presented to comprehend the problem and its many variants for suitable applicability. A revised method considering possibility and necessity factors to work with imprecise data and to support different levels of optimization along with repetitiveness in solution is proposed. A fuzzy model is proposed to deal with imprecise parameters whose de-fuzzification gives a standard constraint optimization problem which is solved using genetic algorithm. The work proposes to extend the standard solution to find whether accepted level of profit can be achieved and for allowing multiple instances of the same object in the solution. The results of genetic optimization over two different solution paths are analyzed and better performance is achieved.**

Keywords—Combinatorial problem; Knapsack; optimization; Fuzzy; Genetic algorithm

I. INTRODUCTION

One of the classic combinatorial optimization problems is the knapsack problem which finds many applications in the computing domain. From a given a set of items (each with a weight and a value), the objective is to find number of items that can be included so that the total weight is less than or equal to a given limit and also the total value is maximum. The scalability of knapsack problem is narrow, i.e., the existing dynamic programming solution cannot be scaled for high limit of the weights. Also the classical knapsack problem is the one which handles crisp values of weights and profits. However there may arise situations in real time where the profit and weight values may not be crisp and may be imprecise. Hence solving the problem with these imprecise characteristics cannot be done by the classical dynamic programming solution. Thus in this work we propose a fuzzy model to deal with imprecise parameters whose de-fuzzification gives a standard constraint problem which is solved using genetic algorithm.

A. Motivation

The main motivation is to extend the classical Knapsack to real time situations like capital budgeting, network planning, etc. where data is not so accurate. A classic example of such real time situation is variable budgeting in a developing firm.

Being NP-Hard, only pseudo polynomial algorithms like dynamic Programming algorithms are available. It can only be applied to a small capacity or weight limit C. Some existing approaches handled fuzzy data, to convert them to precise ones. But extending the same by applying genetic algorithm will tend to scale up the capacity limit C that can be handled by the system.

II. RELATED WORKS

A. Traditional knapsack

The traditional knapsack problem has finite set of items say S, whose cardinality is n, i.e., |S|=n, with every item in S having a weight w>0 and profit p>0 associated with it, and the task is to compute the maximum profit that be achieved by choosing a subset of items H where H⊆S such that the total weight of items in H does not exceed C. This can be formulated as

$$maximize \sum_{i=1}^{n} p_i x_i$$

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

$$max \sum x_i \in \{0,1\}, \quad i = 1, \dots n$$

where xi=1, indicates the ith item is included in the solution.

In [1] the authors have proposed an effective method for solving small to medium sized knapsack problems using a modified discrete shuffled frog leaping algorithm to solve 0-1 knapsack problems. They have used the local search of the 'particle swarm optimization' technique and the competitiveness mixing of information of the 'shuffled complex evolution' technique for solving tightly constrained 0-1 knapsack problems. A hybrid algorithm to solve the 0-1 Knapsack Problem using the Genetic Algorithm combined with Rough Set Theory was proposed in [2]. In [3] a modified scatter search methodology for different sizes of 0-1 Knapsack Problem (KP) was presented. Rough set theory was used to improve the initial features of scatter search. Rough Set Theory was adopted to reduce attributes for finding the important genes, hence reducing the search space. An artificial chemical reaction optimization algorithm (ACROA) with a greedy strategy was proposed in [4] to solve 0-1 knapsack problem. ACROA was used to implement the local and global search along with a greedy based solution repairing strategy. Cohort Intelligence, ability to learn from each other was tested with Knapsack Problem for several cases of the

0–1 KP [5] and the effect of various parameters on the solution quality has been discussed. A binary particle swarm optimization with a greedy strategy was devised in [6] to solve 0-1 knapsack problem considering penalty function and greedy measure. In [7] the authors presented Gravitational Search Algorithm (GSA) with the cognitive component of finding the best position of the particles and the particle with the best position in the system as well in the neighborhood.

B. *Knapsack with imprecise data*

In areas where the knapsack takes in imprecise parameters as input, several fuzzy models have been proposed so far. One such modeling is done in [8]. This work takes in the profit and weight as a range of values and are modeled using a trapezoidal function defined by (1).

$$\mu_X(u)\begin{cases} 1, & if\ X \in [x_1, x_2] \\ \dfrac{(a - x_1 + x)}{a}, & if X \in [x_1 - a, x_1] \\ \dfrac{(b - x + x_2)}{b}, & if\ X \in [x_2, x_2 + b] \\ 0, & elsewhere \end{cases} \qquad (1)$$

where x1<=x2 and a, b>0Here the fuzzy interval is denoted by the quadruple (x_1,x_2,a,b).

Consider a finite set of items A, where |A|=n. Let Pi (p_{i1}, p_{i2}, a_i, b_i) be a trapezoidal fuzzy interval which models the imprecise profit of the ith item and let Wi (w_{i1} ,w_{i2}, c_i, d_i) be a trapezoidal fuzzy interval which models the imprecise weight of the ith item. Let C be the crisp capacity of the knapsack and D denoting a given accepted level of profit. Let H⊆I be a given, nonempty subset of items. Then W(H),the fuzzy weight of H and P(H), the fuzzy profit of H is given as:

$$P(H) = (\textstyle\sum_{i=1}^{n} p_{i1}, \sum_{i=1}^{n} p_{i2}, \sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i) \quad where\ i \in H$$

$$W(H) = (\textstyle\sum_{i=1}^{n} w_{i1}, \sum_{i=1}^{n} w_{i2}, \sum_{i=1}^{n} c_i, \sum_{i=1}^{n} d_i) \quad where i \in H \qquad (2)$$

By using the denotations in (2), the knapsack problem is formulated with possibility and necessity measures as in (3).

$$\max_{A \subseteq I} F(A) = \max_{A \subseteq I}(\min\{Pos(F(A) \geq D), Nec(W(A) \leq C)\}) \qquad (3)$$

so as to achieve the maximum possible profit without violating the capacity constraint. After this, the problem of knapsack is solved by converting to a mixed integer programming model and is solved using linear programming techniques. The above formulation is transformed to an integer programming model as cited in (4):

$$\max\ \lambda,$$
$$\textstyle\sum_{i=1}^{n}((p_i + b_i)x_i - b_i \lambda x_i) \geq D,$$
$$\textstyle\sum_{i=1}^{n}(w_i x_i + \sum_{i=1}^{n} d_i \lambda x_i) \leq C,$$
$$0 \leq \lambda \leq 1, x_i \in \{0,1\}.$$

This linear programming model when solved gives the desired solution which has the following limitations. It cannot be extended to higher weight limits and solving unbounded item knapsack is infeasible. (An item can be chosen more than once).A novel ant colony optimization algorithm was adopted in [9] to solve binary knapsack problem. Fuzzy possibility and necessity approaches are used to obtain optimal decision by the proposed ant colony algorithm.

For solving the multi objective 0–1 knapsack problem quantum-inspired artificial immune algorithm for exploration of the search space was presented in [10] and system was able to find better spread of solutions and better convergence compared to a quantum-inspired evolutionary algorithms. The multidimensional knapsack problem has been solved using a Boltzmann machine[11]. Alternative neural network approaches, including a Hopfield network with asymmetric weights and the Dual-Mode Dynamics Neural Network, have been proposed to handle inequality constraints, and tested using the knapsack problem as detailed in [12].

III. PROPOSED WORK

The work on solving 0-1knapsack with imprecise weights and profits includes the following steps:

1) Model input data using trapezoidal functions
2) Formulate the problem using fuzzy model
3) Convert it to a linear programming problem formulation.
4) Apply Genetic algorithm techniques on the problem formulated
5) The linear programming problem formulated above need correction factor to converge at the right profit, so search through the state space for the right point and perform step 4 for each state.

The steps 1 and 2 are implemented using the models proposed in [8]. The output from the step 2 is converted to a linear programming problem by using Zadeh extension principle. The workflow is depicted in Fig 1.

As discussed earlier, the given range of weight and profit values are taken as trapezoidal functions shown in Fig 2 and as given in (1). Thus we have a set of trapezoids using which the knapsack problem is formulated with the constraint that sum of weights defined value to yield a minimum threshold profit as in (3).

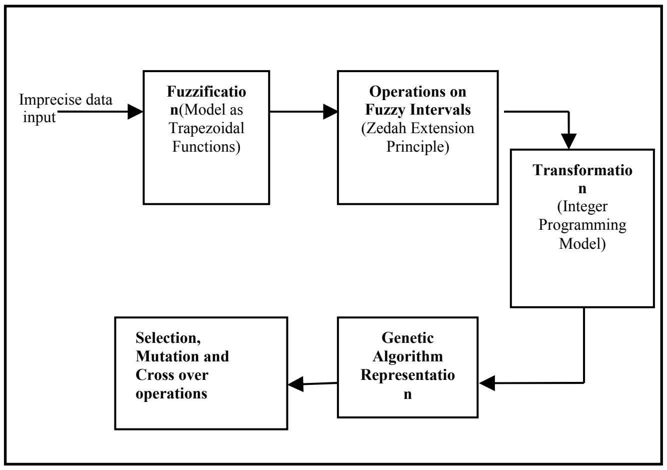
Imprecise data input
Fuzzification (Model as Trapezoidal Functions)
Operations on Fuzzy Intervals (Zedah Extension Principle)
Transformation (Integer Programming Model)
Genetic Algorithm Representation
Selection, Mutation and Cross over operations

Figure 1. System Workflow Diagram

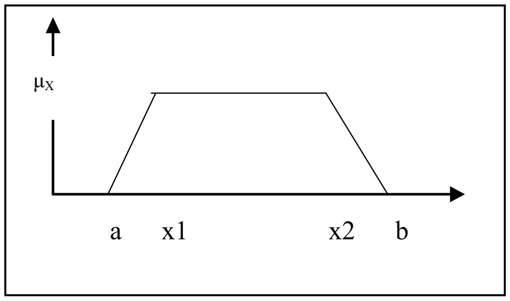

Figure 2. Trapezoidal representation of Input

Equation (4) represents the linear programming problem of Knapsack. Here p and w represent fuzzy quantities with an error factor say 't'. which replaces 'λ', (ie) λ=t/(powers of 10). Now the formulation becomes as in (5).

$$\sum_{i=1}^{n}\left(\left(p_i+(1-t10^{-k})b_i\right)x_i\right)\geq D$$
$$\sum_{i=1}^{n}\left(\left(w_i+t10^{-k}d_i\right)x_i\right)\leq C, \qquad (5)$$
$$x_i\in\{0,1\},\ i=1,2\ldots n.$$

and is solved applying Genetic algorithm. If t increases, the required objective function is maximized. But beyond a level T, feasible solution cannot be obtained. Thus the goal is to find maximum t that gives a feasible solution. It is shown that once for a given T feasible solution is not obtained for all t >T the feasible solution cannot be obtained. Hence to find this t we can use the bisection approach.

A. Genetic Algorithm Representations

Every item is represented using a pair of weight and profit values. A chromosome 'A' is defined as the array of size equal to the number of elements, where each element in the array is either '0' or '1' corresponding to exclusion or inclusion of ith item in the solution. Each cell has a weight part, which is given by . Similarly the profit part of each cell is given by .

The sum of profits of those items that are included in the solution A is defined as the fitness function as shown in (6).

$$Fitness = \sum cell[i][0] * A[i]$$
$$for\ i = 1,2,3\ \ldots.. \qquad (6)$$

Valid chromosomes are those whose weight is less than the upper bound. Only valid chromosomes are selected for crossover.

B. Population Selection

Two trials with varying initial population, crossover and selection operations are carried out and observed results are recorded for comparative analysis.

Method 1: The population size is set to twice the number of genes in the chromosome. Each chromosome of the population is first generated randomly. For each chromosome, its weight is calculated using the following.

$$Total\ weight\ T = \sum cell[i][1] * A[i]$$

where cell[i][[1] is the weight of the ith item given. The error is calculated as,

$$Error\ e = |C - T|,$$

where C is the capacity of the knapsack. The chromosome is then made to converge to the nearest feasible solution (i.e) the weight of the chromosome should be less than or equal to the capacity C of the knapsack.

Method 2: The initial population is taken as an identity matrix of order equal to the size of the chromosome.

C. Selection

The work adopts two different selection procedures as described below for comparative analysis.

Method 1: This work uses Roulette wheel selection, according to which each chromosome is given a slice of a circular roulette wheel equal in area to its fitness. The individual chromosomes are mapped uniquely to a number between 0 and n. From 0 to n-1 a number is chosen randomly. This is same as rotating a wheel and selecting a roulette. Doing it N times lets one choose n different samples of parent chromosomes for cross over. The fitness of the chromosomes is not taken care of in this method.

Method 2: This work incorporated the selection algorithm discussed in [13]. It combines the advantageous parts of roulette wheel selection and rank selection i.e., exploitation and exploration traits are combined here. This is a generation dependent selection algorithm, i.e., the initial generations have a low selection pressure, hence the algorithm explores the search space, in contrast when the selection pressure is high in the later generations, that the algorithm exploits.

D. Crossover

Two crossover operations are utilized to generate better offspring.

Method 1: Multi parent crossover[14] is used here. The number of parents to be used is selected randomly (say n). Then each parent is split into n sections. The offspring is constructed by selecting the sections in the diagonal way respectively one section from each parent.

Method 2: Single Point crossover is used here. It is defined as swapping the segments of two different chromosomes selected i.e., swapping the sub arrays between two chromosomes. Here single point crossover is used. The crossover point is determined randomly by generating a random number between 0 and number of items- 1.

E. Mutation

Mutation is made to prevent genetic algorithms from falling into a local extreme. Mutation is performed here on each bit position of the chromosome with 0.1 % probability.

IV. SIMULATION AND RESULT ANALYSIS

Ga toolkit available in Global Optimization Toolbox in MATLAB was used for realizing the problem. The ga operators were customized for the problem. Given the input for the system as the range of values for weights and profits, the maximum profit that is achievable without exceeding the capacity is computed and fitness plots for the two trials are plotted as shown in Fig.3 and Fig.4 respectively. Here the plots are in negative axes values since Ga toolkit in MATLAB basically minimizes the objective function.

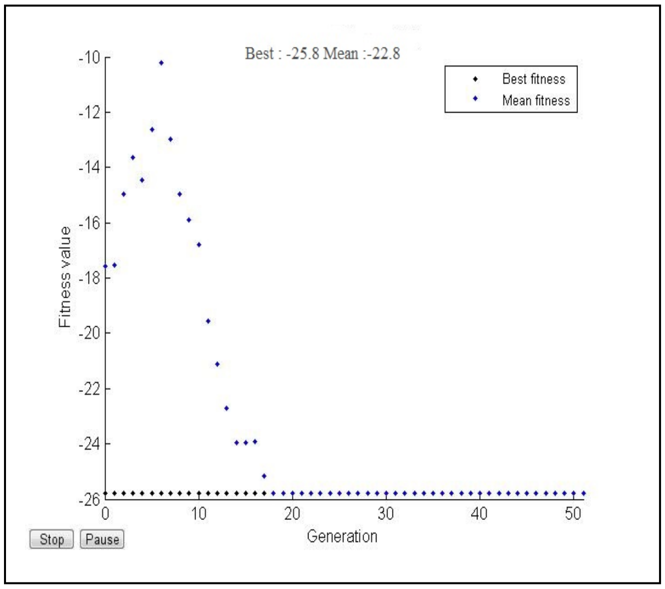

Figure 3. Trial 1 Plot for Best Fitness

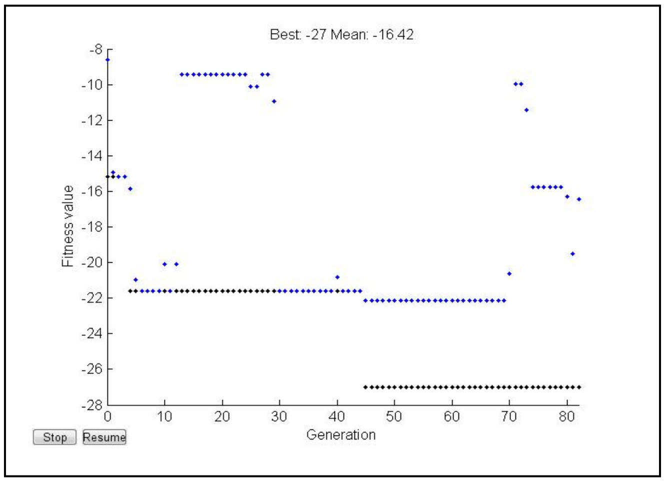

Figure 4. Trial 2 Plot for Best Fitness

It is inferred that method 1 converges faster than method 2 due to the abundant feasible population in the initial state while method 2 explores widely and so converges slowly. Considering the trade off in time, method 2 on the other hand converges at global maximum.

V. CONCLUSION

The work on revised knapsack with imprecise data adopted fuzzy theory to deal with the imprecision in the input and genetic algorithm to solve the transformed model of the base fuzzy model. It is seen that using genetic algorithm shows better performance for large inputs. This work can be further incorporated with additional methods such as including the 't' value as a gene in the chromosome itself rather than generating population at each 't' and mutation function can be modified with the weighted probability based on their relative weight and profit values.

REFERENCES

[1].Kaushik Kumar Bhattacharjee, and S.P.Sarmah,"Shuffled frog leaping algorithm and its application to 0/1 knapsack problem,"APPL SOFT COMPUT, vol. 19, pp. 252-263, June 2014.
[2].TribikramPradhani, AkashIsrani, and Manish Sharma,"Solving the 0-1 Knapsack Problem Using Genetic Algorithm and Rough Set Theory,"IEEE Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on, Department of Information and Communication Technology (ICT), Manipal University, Karnataka, India, 2014.
[3]. Hassan Rezazade,"A Rough Sets based modified Scatter Search algorithm for solving 0-1 Knapsack problem," Decision Science Letters,vol 4, issue 3, pp. 425-440, 2015.
[4].Truong, Tung Khac, Li, KenliXu, Yuming, Ouyang, Aijia, Nguyen, and TienTrong, "Solving 0 - 1 knapsack problem by artificial chemical reaction optimization algorithm with a greedy strategy" in J INTELL FUZZY SYST, vol. 28, no. 5, pp. 2179-2186, 2015.
[5].Anand J. Kulkarni, and HinnaShabir, "Solving 0–1 Knapsack Problem using Cohort Intelligence Algorithm,"INT J MACH LECT CYBERN, vol. 7, Issue 3, pp 427–441, 2016.

[6]. Phuong HoaiNguyena,b, Dong Wanga , and Tung Khac Truong, "A New Hybrid Particle Swarm Optimization and Greedy for 0-1 Knapsack Problem. INDONES JELECTR ENG COMPUT SC, vol. 1, No. 3, pp 411 ~ 418, 2016.
[7].Razavi, SeyedehFatemeh,Sajedi, and Hedieh," Cognitive discrete gravitational search algorithm for solving 0-1 knapsack problem," in J INTELL FUZZY SYST, vol. 29, no. 5, pp. 2247-2258, 2015.
[8].Adam H.Kasperski and Michal Kulej,"The 0-1 Knapsack problem with fuzzy data,"FUZZY OPTIM DECIS MA, Springer, vol.6, Issue.2, pp.163-172,2007.
[9].C. Changdar , G.S. Mahapatra , and R.K. Pal,"An Ant colony optimization approach for binary knapsack problem under fuzziness,"APPL MATHCOMPUTAT, vol.223, pp.243–253, Oct 2013.
[10].JiaquanGao ,Guixia He, Ronghua Liang, and ZhilinFeng, "A quantum-inspired artificial immune system for the multiobjective 0–1 knapsack problem,"APPL MATHCOMPUTAT, vol. 230, no.1, pp.120–137, mar 2014.
[11].Vaithyanathan, S., Ogmen, H., and Ignizio, J, "Generalized Boltzmann Machines for Multidimensional Knapsack Problems," Intelligence Engineering System through Artificial Neural Network," vol.4, pp.1079-1084, 1994.
[12]. Lust T, and TeghemJ,"The multi-objective multidimensional knapsack problem: A survey and a new approach," INTTRANS OPER RES,vol. 19, no.4, pp.495-520, 2012.
[13]. Rakesh Kumar and Jyotishree, "Blending Roulette Wheel Selection and Rank Selection in Genetic Algorithms,"INT J MACH LECT COMP,vol 2, No. 4, pp. 365-370,2012.
[14].Yang Wang, Zhipeng Lu and Jin-Kao Hao, "A study of Multi-parent Crossover Operators in a Memetic Algorithm,"11thLECT NOTES COMPUT SC,Springer, pp.556-565, 2010.