IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

A Fast and Efficient Genetic Algorithm to Solve 0-1 Knapsack Problem

Megha Gupta
meghaitm@gmail.com

***Abstract*— Knapsack problem is a typical computer algorithm of NP complete (Nondeterministic Polynomial Completeness) problem. The research of solving this problem has great significance not only in theory, but also in application, for example, resource management, investment decisions and so on. For solving this problem, scholars have developed a number of algorithms, however, they are all have some drawbacks. This paper represents a fast Genetic Algorithm to solve the knapsack problem, and also demonstrate its feasibility and effectiveness throng an example.**

***Keywords*— Knapsack Problem, Genetic Algorithm, Computer Simulation.**

I. INTRODUCTION

This The knapsack problem is a traditional problem of combination and optimization [1],[2], and has a variety of applications for capital budgeting, project selection, material incision, cargo loading, cutting stock, bin packing, and economic planning. The knapsack problem is also a NP hard problem, and has been intensively studied, especially in the last decade, attracting both theorists and experimentalists [3], the theoretical interest arises mainly from their structure in which more complex optimization problems can be solved through a series of knapsack-type sub problems. From the practical point of view, these problems can model many industrial situations to find a combination of different objects. Some approaches are presented for this combination problem and achieve all-right results, but not suit for very large scale problem because of their convergence rate. The Genetic Algorithms (GAs) are proposed based on Darwin's principle of survival of the fittest by Professor J. H. Halland in 1975 to solve larger scale combination optimization problem. It can jump out local search space to achieve optimal solutions in global space. In Genetic Algorithms, it process a population of individuals which represent search space solutions, each individual is candidate solution and population including all individuals are exanimated Simultaneously, and quality of population are improved gradually, at last the best solution or secondary solutions are achieved by repeating employing three GA operations: selection, crossover and mutation. GA are theoretically and empirically proven to provide robust search capabilities in complex spaces, offering a valid approach to problems requiring efficient and effective search [4], [5].

The zero-one knapsack problem involves filling a knapsack, which has a weight capacity c, with a number of items numbered 1, 2 ... n one by one. Each item has an associated weight W_i and profit V_i. The aim is to find a combination of items whose weight does not exceed the knapsack's capacity and to maximize the overall profit. Each item has only two choices namely encasing and non-encasing knapsack and item *i* can't be encased repeatedly or partly.

Mathematically, given $C > 0$, $Wi > 0$,$Vi > 0$, $1 \leq i \leq n$, the zero-one knapsack problem can be represented by a vector of binary values $X_1, X_2 \ldots\ldots X_n$, Where $X_i = 0$ or 1 $(1 \leq i \leq n)$.

The aim is to find a vector which satisfies the constraint

$$\sum_{i=1}^{n} W_i X_i \leq C \qquad (1)$$

And maximize the total profit as

$$Maximize \sum_{i=1}^{n} V_i X_i \qquad (2)$$

In recent years, genetic algorithm is widely to solve the knapsack program, but the traditional genetic algorithm is often can't get the satisfactory result, even more, in many cases the result is worse than using the greedy algorithm, because of the large search space and the weakness of local search ability. There is also some improved genetic algorithms to solve the Knapsack problem, however, those algorithms has many shortcomings in the solution speed and the convergence of

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

algorithm. This paper presents an efficient Genetic Algorithm with faster convergence to solve 0-1 knapsack problem.

II. Literature Survey

A number of researchers have reported applying GAs to solve 0/1 knapsack problems. Anagun and Sarac [6] developed a genetic algorithm for solving 0/1 knapsack problems in which GA's performance was optimized using [7]. Bhatia and Basu [8] proposed a gene induction approach for genetic algorithms. The approach was applied to 0/1 knapsack problem, and found near optimal results in all the representative problem instances in the literature. Pawlak [9] introduced RS that can process data with uncertainty, reduce the size of data sets, and generate decision rules from the data sets. In these data sets, some attributes may be redundant and can be eliminated without reducing the original classification quality. In RS, the process of finding a smaller set of attributes that ensures the same classification quality is called attribution reduction and the underlying set is referred to as a deduct. This paper presents an efficient genetic algorithm with fast convergence towards results to solve knapsack problem. In multiple knapsack problems (MKP), there are a set of containers of various capacity. A Nominated grouping genetic algorithm has been proposed in [10]. If Fl and F2 are two feasible solutions for a bin packing problem then Fl *dominates F2 if the value of the optimal solution which can* be obtained by assigning FI to a bin is no worse than the value of the optimal solution that can be obtained by assigning F2 to the same bin. UGGA and genetic operators are guaranteed to generate chromosomes consisting only of un-dominated bin assignments. It also showed that UGGA significantly outperformed other algorithms for strongly correlated and multiple subset-sum problem but poorly performed on the weakly correlated instances. Similarly, the single-objective genetic algorithm (SOGAs) is compared with multi-objective genetic algorithms in the applications to multi-objective knapsack problems [11]. It has shown that MOGAs outperform SOGAs even when they are evaluated with respect to a scalar fitness function used in SOGAs. It also verifies that the search ability of MOGAs in inversely proportional to the number of objectives. Knapsack problem used as a class of compositional design problems is shown in [12]. The problem with complicated constraints is formulated as a set of local sub problems with simple constraints and a supervising problem. Every problem is solved by GA to generate a set of suboptimal solutions and in the supervising problem, the elements of every set are optimally combined by GA to yield the optimal solution for the original problem. The method is a self-learning method where the empirical knowledge obtained by solving the problem is effectively utilized to solve similar problems efficiently. A genetic algorithm in which the number of individuals changes to show increase in accuracy of solution is shown in [13]. The number of population is doubled initially against the accuracy of solution. First stage increases the searching ability. Then, accuracy is improved at second stage by reducing the number of population. Wei, beibel and jiang derived an improved solution for 0-1 knapsack problem based on the dual population genetic algorithm, which can overcome the find of precocious and local convergence in iterative processes. The performance valuation shows that the solution is better than the traditional genetic algorithm [14]. Yanqin Ma and Jianchen Wan solves the 0-1 knapsack problem with the hybrid adaptive genetic algorithm which combined with greedy algorithm. It presents a method for optimal design of an improved adaptive algorithm and repairs the infeasible solution with greedy algorithm [15] while Basima and Moutaz applied several mutation methods to different non-deterministic polynomial (NP) hard problems [16]. Chen Lin presented heuristic strategies that takes into account the characteristics of 0-1 knapsack problem. Here, a heuristic Genetic Algorithms (GA) is proposed to solve the 0-1 knapsack problem, in every generation, populations are divided into two sections: superior clan and inferior clan, and in superior clan are pick up to replace the chromosomes in inferior clan. This approach of schema replacement promotes individual evolution effectively and achieve best solution of the problem [17].

III. Proposed Genetic Algorithm

GAs start with selecting an initial population, iteratively apply operators to reproduce new populations, evaluate these populations, and decide whether or not the algorithms should continue to execute. GAs differ from classical optimization algorithms mainly in that Gas operate on a population of individuals instead of parameters in classical algorithms. Compared to the optimization algorithms, each individual in a population is encoded into a chromosome that represents a candidate solution. A chromosome is composed of genes that are usually of binary form. The evaluation of an individual is determined by the fitness function value corresponding to the objective

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

function value. Typical GAs include the following steps:

1. Generate an initial random population of chromosomes.
2. Evaluate the population of chromosomes using an appropriate fitness function.
3. Select the subset of chromosomes with better fitness value as parents.
4. Crossover the pairs of parents with given probability (Pc) to produce offspring.
5. Mutate the chromosomes of offspring with probability (Pm) to avoid early trap into local solutions.
6. Re-evaluate the fitness values of offspring.
7. Terminate algorithms if the stopping criteria are satisfied.

Knapsack Problem has a constraint as given in equation (2), we need two functions to solve it, one is fitness function and second one is constraint function. When GA is in step one as stated above it follows both fitness and constraint function to generate population which is fully feasible to solution. Till this GA is efficient but when it comes to step 4 and 5 its population is went infeasibility of solution, how, as discussed below:

Assume a knapsack problem of four items has a full feasible random population. When it performs step four using two feasible solution as parents, it generates to children, it could happen that one of it or both are not feasible as shown below:

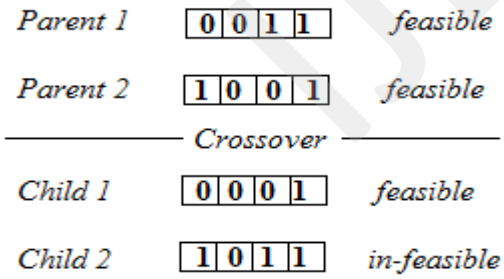

Figure 1: Crossover Operation

Also when step 5 of doing mutation is performed, it also may lead next solution to infeasibility as shown below:

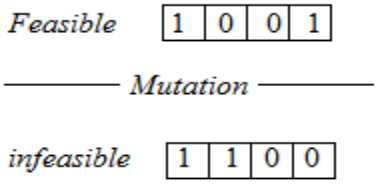

Figure 2: Mutation Operation

With concerning two figures above, we come to the conclusion that in conventional genetic algorithm, due to the crossover and mutation operations leads the next population to be inefficient and that's why it takes a much time to be converged to best solution. So to make the new population we adding a small function to conventional GA such that: after the end of mutation operator we adding a function named as "fcheck", which will check the population and find all the infeasible solutions and it will replace all infeasible solutions to randomly generated new solutions thus the new population is fully efficient in terms feasibility. It simply works as described below:

For i = 1 to population Size
Check the current solution;
If (infeasible)
Replace by random feasible solution
Else
Pass to next population
End
End

So the modified GA is now works as given below:

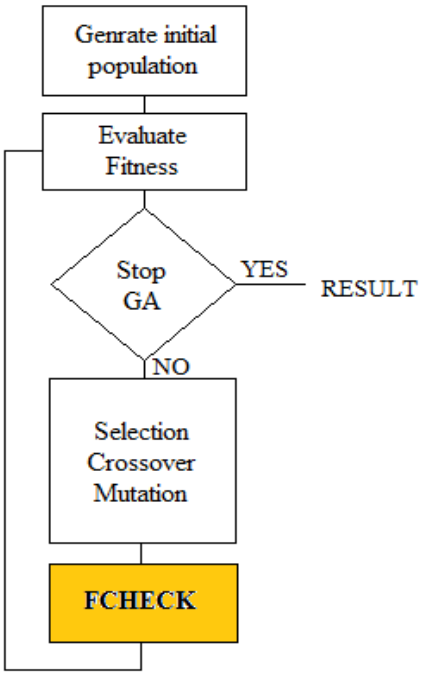

Figure 3: Proposed GA

Figure above showing proposed genetic algorithm, its impact has been shown in next section where results are compared with conventional GA.

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

IV. Simulation and Results

Proposed work has been implemented in MATLAB R2009b and simulation is performed many times and results were analyzed. Here we showing one of the experiment and its result.
The considered knapsack problem is:

Item	Value	weight
A	42	7
B	12	3
C	40	4
D	25	5

Capacity of given knapsack is 10.
The comparison of results is shown in table below:

Point of Comparison	Conventional GA	Proposed GA
Set	[0 0 1 1]	[0 0 1 1]
Value of Set	65	65
Weight of Set	9	9
Computation Time	2.91 Sec	0.20 Sec
Iterations	11	2

Figure below showing the convergence of both GAs:

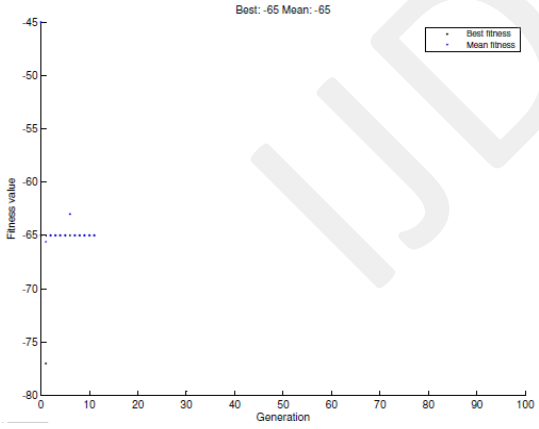

Figure 4: Convergence of Conventional GA

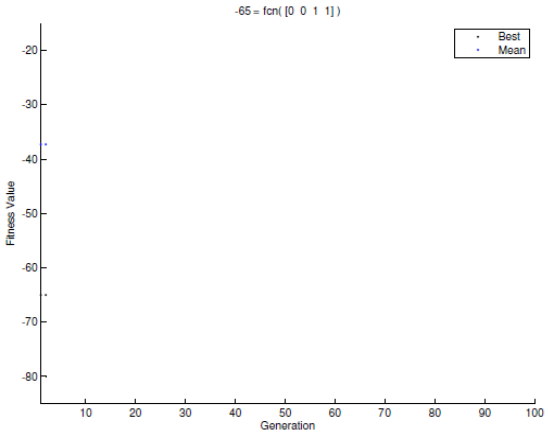

Figure 5: Convergence of Proposed GA

IV. Conclusions

In this paper, we solve the Knapsack Problem using Hybrid Genetic Algorithm. Compared to traditional genetic algorithms, Hybrid genetic algorithm has a great improvement not only in the quality of solution, but also in the solution. New algorithm shows its efficiency also in terms of convergence towards best solution, it due the efficient population which we creating using "fcheck" function. This work can also be applied to multi objective knapsack problems in future to improve speed of operation.

References

[1] Han K H, Kim J H. Genetic quantum algorithm and its application to combinatorial optimization problem[C]. Proceedings of the 2000 Congress on Evolutionary Computation, Piscataway, 2000, 2: 1354 -1360.

[2] Han K H, Park K H, Lee C H, et al. Parallel quantum-inspired genetic algorithm for combinatorial optimization problem[C]// Proceedings of the 2001 IEEE Congress on Evolutionary Computation, 2001: 1422-1429.

[3] S. Martello, D. Pisinger, P. Toth, New Trend in exact algorithms for the 0-1 knapsack problem, European Journal of Operational Research 123 (2000) 325–332.

[4] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. (Addison-Wesley, New York, 1989).

[5] J.H.Holland, Adaptation in Natural and Artificial Systems (University of Michigan Press, 1975; MIT, London, 1992).

[6] A. S. Anagun, and T. Sarac, ''Optimization performance of genetic algorithm for 0-1 knapsack problems using Taguchi method,'' Lect. Notes Comput. Sc., vol. 3982, pp. 678-687, 2006.

[7] G., Taguchi, S. Chowdhury, and Y. Wu, "Taguchi's Quality Engineering Handbook," Wiley-Interscience, New York, NY, 2004.

[8] A. K. Bhatia, and S. K. Basu, "Tackling 0/1 knapsack problem with gene induction," Soft Comput., vol. 8, no. 1, pp. 1-9, 2003.

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

[9] Z. Pawlak, "Rough set," Int. J. Inform. Comput. Sci., vol.11, pp. 341-356, 1982.

[10] Alex S. Fukunaga, "A new grouping genetic algorithm for the multiple knapsack problem", IEEE congress on evolutionary computation, 2008, pp. 2225-2232.

[11] Hisao Ishibuchi, Yusuke Nojima, Tsutomu Doi, "Comparison between single-objective and multi-objective genetic algorithms: performance comparison and performance measure", 2006 IEEE congress on Evolutionary Computation, 2006, pp. 1143-1150.

[12] Masakazu Suzuki, Yuki Hiyama and Hideki Yamada, "An efficient solution for compositional design problems by multi-stage genetic algorithm", 22nd IEEE International Symposium on Intelligent Control, 2007, pp. 626-633

[13] Akihiko Tsukahara and Akinori Kanasugi, "Genetic algorithm that can dynamically change number of individuals and accuracy", IEEE Frontiers in the Convergence of Bioscience and Infromation Technologies, 2007, pp. 785-789.

[14] Wei Shen, Beibei Xu, Jiang-ping Huang, "An Improved Genetic Algorithm for 0-1 Knapsack Problems", Second International Conference on Networking and Distributed Computing 2011.

[15] Yanqin Ma and Jianchen Wan, "Improved Hybrid Adaptive Genetic Algorithm for Solving Knapsack Problem", The 2nd International conference on intelligent control and information processing 2011.

[16] Basima Hani Hasan and Moutaz Saleh Mustafa, "Comparative Study of Mutation Operators on the Behavior of Genetic Algorithms Applied to Non-deterministic Polynomial (NP) Problems", Second International Conference on Intelligent Systems, Modelling and Simulation 2011.

[17] Chen Lin, "A Heuristic Genetic Algorithm Based on Schema Replacement for 0-1 knapsack Problem", Fourth International Conference on Genetic and Evolutionary Computing 2010.