TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

Practical Approach Of Genetic Edge

Apurva Tiwari [a] , Khushi Maurya [b] and Mahesh Kumar Tiwari [c]

[a] Scholar, National PG College, Lucknow, India
[b] Scholar, National PG College, Lucknow, India
[c] Assistance Professor, Computer Science Department, National PG College, Lucknow, India
[a]vt1733620@gmail.com, [b]maurya123riya@gmail.com, [c]Maheshyogi@gmail.com

KEYWORD

Genetic Algorithm, Fractional Knapsack Problem, Risk Constraints, Optimization, Resource Allocation

ABSTRACT

The greedy algorithm effectively solves the Fractional Knapsack Problem (FKP) in linear scenarios. However, the problem's complexity increases, and the greedy approach fails to yield optimal solutions when additional constraints, such as weight and risk, are introduced. Our previous theoretical study highlighted the limitations of the greedy method in such multi-constraint environments. This paper presents the real-world application of a Genetic algorithm for these challenges. The FKP is not like 0-1 Knapsack problem, it allows fractional item selection offering greater flexibility in practical applications. The proposed GA optimizes item selection based on value while ensuring that both capacity and risk constraints are met. Experimental results validate the GA's effectiveness, producing optimal solutions with maximum value and controlled risk thresholds.

1. Introduction

Especially the knapsack issue is an important type of optimisation problem in several domains including finance, transportation, and resource allocation. Unlike the classic 0-1 Knapsack Problem, the Fractional Knapsack Problem (FKP) allows fractional inclusion of items, providing more flexible solutions. This study looks at using a genetic algorithm to solve the FKP with extra restrictions, like a risk threshold, to simulate real-world scenarios where adding particular components could have dangers or unknowns.

It becomes essential in this situation to include risk management in the decision-making process. The ability to optimize item selection while meeting capacity and risk limits is critical as organizations deal with more complex settings. A promising answer to this problem is provided by Genetic Algorithms (GAs), which are based on the ideas of evolution. GAs can efficiently explore large solution spaces and converge on ideal configurations that maximize value while minimizing risk by mimicking natural selection processes. Natural selection-inspired optimization methods known as genetic algorithms (GAs) are made up of many essential elements:

Population: A collection of possible remedies (chromosomes).
Chromosomes: A single solution made up of genes is called a chromosome.
Gene: A chromosomal component whose value is known as an allele.
Fitness Value: A metric used to assess the quality of a solution.

Selection (using techniques like Roulette Wheel or Tournament Selection to choose which chromosomes to breed), Crossover (combining genes from parents to make offspring), and Mutation (random modifications to introduce variation) are all steps in the GA process. Until a good solution is found or a maximum number of generations is reached, the cycle of assessing fitness and creating new populations keeps going. In a variety of domains, GAs are useful for resolving intricate optimization issues [9].

Corresponding Author: Khushi Maurya, National Post Graduate College, Lucknow, India
Email: maurya123riya@gmail.com

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

As we now know, the greedy approach—a sort of optimization algorithm that finds the global optimum solution using the local optimal choices—is the best way to solve fractional knapsack. It provides less time and space complexity than other algorithms, but it does so in a linear manner or with a single constraint. In a prior theoretical approach research publication, we explained how a greedy algorithm operates when given numerous constraints. [1]

Research Objective: A unique GA method designed to handle the FKP with capacity and risk restrictions is presented in this research. We show how the suggested algorithm successfully incorporates these elements into the optimization process using a practical methodology. The purpose of the research is to create a genetic algorithm that optimises the value of the items in the knapsack while making sure that the overall weight does not beyond the capacity and that the overall risk stays below a predetermined level [10].

2. Literature Review

This is a survey of the literature based on the most recent findings in the field of genetic algorithms applied to the knapsack problem, taking risk restrictions into consideration:

1. **Optimisation of Multidimensional Knapsack Problem**: Genetic algorithms have been studied recently to handle the complexity of MKPs. A revised evolutionary algorithm was put forth that combined local improvement techniques and repair operators to preserve workable solutions and increase productivity. [2], [3] This methodology guarantees the efficient management of even the largest and most intricate datasets while maintaining the diversity of solutions.
2. **Hybrid Genetic Algorithms for Risk Management**: To solve knapsack problems containing risk restrictions, hybrid models that combine GAs with other optimisation approaches, such as the harmony search algorithm, have been created. In real-world applications like as portfolio management and scheduling, these techniques produce more robust and dependable results by adaptively adjusting tactics based on stochastic aspects. [4]

   Customised variation operators are included in recent work by Srivastava et al. (2023) to increase the effectiveness of genetic algorithms for the FKP. This method particularly tailors crossover and mutation processes to preserve diversity and prevent early convergence, guaranteeing more resilient solutions in a variety of settings. [5]

3. **Adaptive Memory Techniques in Dynamic Environments:** The application of adaptive memory genetic algorithms (AMGA), which use memory techniques to retain and recall optimal solutions as environmental variables change, is highlighted in a work published in the WCECS 2013. When tackling multi-dimensional knapsack issues, where capacities and constraints may change dynamically over time, these solutions become advantageous. [6]
4. **Elitism and Multi-Objective Optimization**: Researchers looked at the effect of elitism on preserving superior solutions over generations in a different recent study. Multi-objective optimisation frameworks combined with genetic algorithms not only maximise the value of the knapsack but also minimise other aspects such as risk or cost, demonstrating the adaptability of GAs to a wide range of problem constraints. [7]

These works demonstrate how genetic algorithms can be continuously improved to solve knapsack problems, especially where risk and other limitations play a major role. This body of literature supports the usefulness of genetic algorithms in addressing challenging optimisation issues under restrictions, and it offers a strong basis for additional investigation and improvement in our work.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

3. Methodology

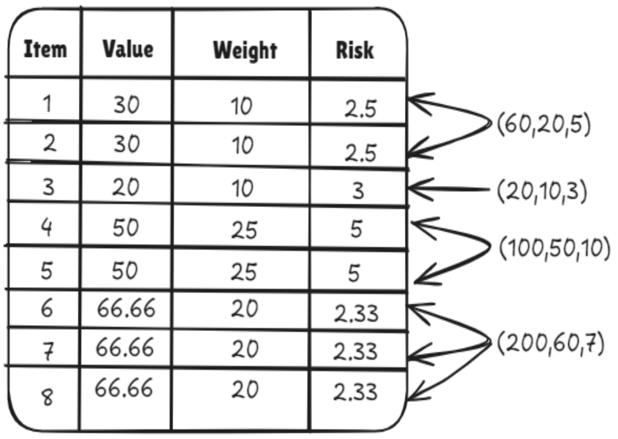
3.1. Problem Formulation

Items Definition: Every item has a value, weight, and risk associated with it. The four items in the dataset that were used have the following specifications:

TABLE 1: EXAMPLE

Item	Value	Weight	Risk
1	60	20	5
2	20	10	3
3	100	50	10
4	200	60	7

Now, after doing partition we have: -
Values: [30, 30, 20, 50, 50, 66.66, 66.66, 66.66]
Weights: [10, 10, 10, 25, 25, 20, 20, 20]
Risks: [2.5, 2.5, 3, 5, 5, 2.33, 2.33, 2.33]



Item	Value	Weight	Risk
1	30	10	2.5
2	30	10	2.5
3	20	10	3
4	50	25	5
5	50	25	5
6	66.66	20	2.33
7	66.66	20	2.33
8	66.66	20	2.33

(60,20,5)
(20,10,3)
(100,50,10)
(200,60,7)

Constraints:
Capacity of the knapsack: 90 units
Maximum risk threshold: 15 units

3.2. Genetic Algorithm Parameters

- **Population Size:** 50 people for a wide-ranging search area.
- **Iterations:** 300 in order to facilitate convergence.
- **Mutation Rate:** 0.1 to prevent too rapid convergence and add genetic diversity.
- **Selection Method:** Fitter individuals have a higher chance of being selected because the roulette wheel is selected based on fitness.
- **Crossover:** The combining of genes from two parents at a single point.
- **Mutation:** To provide variation to the progeny, use bit-flip mutation.
- **Elitism:** Holds onto the best answer for every generation in order to maintain ideal solutions.

3.3. Fitness Function

The fitness function ensures that limitations (risk and capacity) are not crossed by evaluating each person according to the overall value. The fitness value is set to zero if the limitations are broken, hence penalising the person.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

3.4. Algorithm Overview

- **Start:** Randomly generate an initial population.
- **Evaluate Fitness:** For each individual, calculate the total value, weight, and risk, and assign a fitness score based on constraints.
- **Selection:** Select individuals based on their fitness scores using roulette wheel selection.
- **Crossover:** Pair up individuals and exchange genetic material at a random crossover point.
- **Mutation:** Randomly mutate a small portion of the population by flipping bits.
- **Elitism:** Retain the best individual from the current generation.
- **Repeat:** Go to step 2 and repeat the process for a specified number of generations.
- **Result:** Output the best individual (solution) after all generations.

3.5. Algorithm

Here is the Genetic Algorithm approach:

a. **Initialize Parameters:**
   - Set num_items as the number of items.
   - Set population_size, num_generations, and mutation_rate for the genetic algorithm.
   - Define capacity as the knapsack weight limit and risk_threshold as the risk constraint.

b. **Define Fitness Function:**
   - For each individual:
     - Compute total_weight, total_value, and total_risk.
     - If total_weight > capacity or total_risk > risk_threshold, set fitness to 0 (constraint violation).
     - Otherwise, set fitness to total_value.

c. **Initialize Population:**
   - Create a random binary matrix population of size [population_size, num_items].

d. **Begin Genetic Algorithm Loop:**
   - For generation from 1 to num_generations.

e. **Evaluate Fitness for Each Individual:**
   - For each individual in the population:
     - Calculate fitness using the calculate_fitness function.

f. **Track Best Solution:**
   - Find the highest fitness (max_value) in fitness array.
   - If max_value > best_value, update best_value and best_individual.

g. **Selection using Roulette Wheel:**
   - Compute selection probabilities as fitness / sum(fitness).
   - Select population_size individuals with replacement based on these probabilities to form selected_population.

h. **Perform Crossover on Selected Population:**
   - For each i from 1 to population_size - 1 in steps of 2:
     - Generate a random crossover_point in the range [1, num_items-1].
     - Swap genes of individuals i and i + 1 from crossover_point to the end.

i. **Mutation:**
   - Select mutation_count indices in selected_population randomly (calculated as population_size * mutation_rate).
   - For each idx in mutation_indices:

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

- Generate a random mutation_point within [0, num_items-1].
- Flip the bit at mutation_point in selected_population[idx].

j. **Elitism (Retain the Best Individual):**
   - Copy elite_individual (best individual from previous generation) to the last position in population.

k. **Update Population:**
   - Set population to selected_population.

l. **End of Generations:**
   - Print best_individual and best_value

4. Results

4.1. Execution Overview

The specified parameters were used to run the genetic algorithm. The program reached an ideal solution after 300 generations [11].

4.2. Best Solution

Chromosome Representation: The chromosome proved to be the most effective remedy. configuration [1, 1, 1, 0, 0, 0, 1, 1, 1].

- **Items Included**: 1, 2, 3, 6, 7, and 8.
- **Total Value**: 279.98
- **Total Weight**: 90 units (precisely matching the capacity of knapsack).
- **Total Risk**: 14.99 units (below the risk threshold).

This answer shows how the GA can identify the best configuration to maximise value while respecting weight and risk limitations.

5. Discussion

The Fractional Knapsack Problem can be successfully solved using genetic algorithms when risk tolerance is met. GA's versatility combined with mechanisms like crossover, mutation, and elitism enable it to effectively search a wide search field [10].

The algorithm's success depends on striking a balance between exploitation and exploration:

Exploration: The algorithm is able to investigate several solutions because of the mutation rate and the diversity of the original population.

Exploitation: The greatest solutions are honed over centuries to arrive at the best outcome through selection and elitism.

Challenges: The primary difficulties are making sure there is enough population diversity to prevent local optima and fine-tuning the parameters. Revisions to crossover processes, mutation rate, and population size were essential to keeping the algorithm robust.

6. Conclusion

In this research, we show how to optimise the Fractional Knapsack Problem with risk restrictions using a genetic algorithm. The technique successfully strikes a compromise between maximising item value and making sure weight and risk constraints are adhered to. Because of its great adaptability, the strategy can be used in different real-world situations where managing risks and resource constraints are necessary.

Future Work: Subsequent investigations could delve into the realm of multi-objective optimisation by including supplementary limitations like time or availability, or broadening the genetic algorithm to incorporate

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.03, I.04 (2024)
https://www.tejasjournals.com/
https://doi.org/10.5281/ zenodo.14122669

sophisticated methods like adaptive mutation rates or hybrid strategies combining various optimisation algorithms.

7. Reference

[1]. Apurva Tiwari, Mahesh K. Tiwari. (2024) “The Genetic Edge: Revolutionizing Multi-Constraint Fractional Knapsack Solutions”.

[2]. Shah, Shalin. "Genetic algorithm for a class of knapsack problems." arXiv preprint arXiv:1903.03494 (2019).

[3]. W. Shen, B. Xu and J. -p. Huang, "An Improved Genetic Algorithm for 0-1 Knapsack Problems," *2011 Second International Conference on Networking and Distributed Computing*, Beijing, China, 2011, pp. 32-35, doi: 10.1109/ICNDC.2011.14.

[4]. Rezoug, A., Bader-El-Den, M. & Boughaci, D. Guided genetic algorithm for the multidimensional knapsack problem. Memetic Comp. 10, 29–42 (2018). https://doi.org/10.1007/s12293-017-0232-7

[5]. Patel, P.S., Singh, A. A diversity preserving genetic algorithm with tailor-made variation operators for the quadratic bottleneck knapsack problem. Evol. Intel. 17, 1953–1965 (2024). https://doi.org/10.1007/s12065-023-00875-7

[6]. Ünal, Ali Nadi. “A Genetic Algorithm for the Multiple Knapsack Problem in Dynamic Environment.” (2013).

[7]. Kilincli, Taskiran, Gamze, "An Improved Genetic Algorithm for Knapsack Problems" (2010).

[8]. SB Verma, Brijesh P., and BK Gupta, Containerization and its Architectures: A Study, ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, Vol. 11 N. 4 (2022), 395-409, eISSN: 2255-2863, DOI: https://doi.org/10.14201/adcaij.28351

[9]. Anamika Agarwal, S. B. V., B. K. Gupta, A Review of Cloud Security Issues and Challenges, ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, Issue, Vol. 12 N. 1 (2023), pp 1-22, eISSN: 2255-2863, 2023 https://doi.org/10.14201/adcaij.31459

[10]. Ms Gaurvi Shukla, E-Commerce As a Tool for Economic Empowerment and Poverty Reduction,TEJAS Journal of Technologies and Humanitarian Science, ISSN-2583-5599, Vol.02, I.02(2023)

[11]. Vd. Sandeep Aggarwal1, Dr. Balbir Singh, A Case Study on Spinal Canal Stenosis – An Ayurvedic Prospective, TEJAS Journal of Technologies and Humanitarian Science, ISSN-2583-5599, Vol.02, I.03(2023)