Bi-Objective Optimization for the Traveling Thief Problem using a Co-evolutionary Strategy

Radu Solca, Gabriel Ardeleanu

November 2023

Abstract

In this paper, we will present The Traveling Thief Problem, a unique challenge by combining two classical problems: the Traveling Salesman Problem and the Knapsack Problem. To address the bi-objective component, we employ a co-evolutionary approach, leveraging the power of concurrent optimization. This allows us to evolve solutions for both objectives in parallel to achieve optimal solutions for the TTP. Our research presents the results of applying this co-evolutionary algorithm to various data inputs, concluding its effectiveness while proposing ways to improve the results in future experiments.

1 Introduction

In real-world optimization, problems often present themselves as tricky combinations of NP-hard optimization problems that interact with one another. These multi-component optimization problems represent significant challenges, not only due to the nature of each component but also the interaction that binds them into one problem. Bi-objective problems introduce the idea of a trade-off between two optimizations, which makes finding a satisfactory solution particularly tricky.

The Traveling Thief Problem stands as a prime example of a bi-objective optimization problem. In TTP, the goal is to determine the optimal route for a set of thefts, minimizing the total cost of travel between locations and maximizing the value of stolen goods. The conflicting nature of these objectives, as picking up objects will add to the total carried weight, thus slowing the thief which is equal to a higher travel cost, represents our main focus in solving this problem. The Traveling Thief Problem has been the focus of various competitions at international conferences, such as GECCO (Genetic and Evolutionary Computation Conference), held over multiple years, including 2017, 2019 [1] and 2023. These competitions have encouraged the development of new algorithms and strategies to address the complexities of problems like TTP.

To address the complex nature of bi-objective problems, a co-evolutionary approach has emerged as a promising solution. It's aim is to divide the task into the individual aspects of the problem, allowing them to evolve simultaneously. Their evolution will still be based on the Fitness Function that considers the whole problem, meaning that the individuals who work best together are more likely to participate in evolution. This ensures that our search is towards the Pareto front[2], in other words, the combinations that offer the best trade-offs between the conflicting objectives. As this experiment is only aimed at providing results, we will discuss the Pareto front when considering ways to improve.

In this work, we present our results in solving the Traveling Thief Problem, highlighting the performance of our co-evolutionary approach. We will also provide an analysis of the achieved results and discuss potential strategies for further improvements.

2 Methodology

Our approach to addressing the Traveling Thief Problem unfolds within an intricate co-evolutionary structure, designed to optimize the simultaneous Traveling Salesman Problem and Knapsack Problem. Our methodology starts with the initialization of two separate populations: one dedicated to the TSP and the other to the KS, their members being initially randomized to provide a diverse

[1] Source: https://gecco-2019.sigevo.org/index.html/Competitions

[2] Source: https://en.wikipedia.org/wiki/Pareto$_{f}ront$

starting point.

The essential part of our method is found in the evaluation of each individual using a modified fitness function in the evaluation process. Each TSP solution is subjected to an in-depth evaluation, achieved through its pairing with every KS solution from the opposing population. This allows us to approximate the overall compatibility between TSP and KS for each individual. The resulting fitness values are not fixed, they actually correspond to the mean value of all combinations where one individual stays the same. Simply put, it is a measure of compatibility with the other population during evolution and finally represents a corresponding TSP solution. In our algorithm, this evaluation process cascades through the entire TSP population, with each individual evaluated accordingly. Equally, our methodology calls for the mirrored evaluation, as every KS solution is used to go through a similar process. After which, we once again, calculate the average fitness values.

These mean values are the foundation of our evaluation and evolution methods. They aren't just tools for looking back; they guide us throughout our co-evolutionary journey. In the parallel evolution process, we use one-point crossover and a range of mutations to introduce genetic diversity into our solution space, improving the TSP and KS populations.

Our method is based on the separation of the TTP into its TSP and KS subproblems. It aims to create a dynamic synergy between these co-evolving populations through well-organized processes. This approach seeks to reveal the potential for optimal solutions to the complex Traveling Thief Problem.

For this experiment, we also implemented a "good crossover" which ensures that at least one of the parents is above a fixed fitness percentage. It will be decided at every crossover event with a given probability and a given top percentage that the used parent must be a part of. This way we can encourage both the improvement of generations using good parents and not loosing potentially good information in overall bad parents.

3 Results

Using the process outlined in the preceding chapter, our methodology has yielded promising results in our pursuit of solving the Traveling Thief Problem (TTP). Through the co-evolutionary approach, we have used the synergy between the Traveling Salesman Problem (TSP) and the Knapsack Problem (KS) to optimize the challenge, marking a significant step forward in the quest for effective problem-solving strategies.

The following tables showcase the obtained values from various sets of input data, compared with the suggested "decent" values provided as benchmarks by the originator of the problem. [3]

[3]Source: https://cs.adelaide.edu.au/ optlog/TTP2017Comp/

	Benchmark values	Our values
Case 1	18.049	14.849
Case 2	109.221	97.261
Case 3	428.117	408.017

Table 1: Input data for 280 cities

	Benchmark values	Our values
Case 1	256.591	242.721
Case 2	1.611.342	1.600.532
Case 3	6.536.729	6.498.059

Table 2: Input data for 4461 cities

	Benchmark values	Our values
Case 1	1.810.973	14.849
Case 2	1.536E7	15.248.719
Case 3	5.732E7	56.872.620

Table 3: Input data for 33810 cities

For reference, the values we used for each variable are the following: 100 generations, 30 individuals per population, the probability of a gene mutation occurring 0,1. For the good crossover method, the probability of it occurring is 0,5 and we choose to select a parent from the top 80 percent of the population. In the making of a new generation we also copy a random number of individuals between 5 and 10 percent of the population.

Case 1 = 1 item per city, low knapsack capacity, strongly correlated item weight/profit

Case 2 = 5 items per city, medium capacity, uncorrelated but similar item weights/profits

Case 3 = 10 items per city, high capacity, uncorrelated item weights/profits

It is crucial to remember that the presented results should be viewed as preliminary. These findings are not the final outcomes of our research but rather serve as a Proof of Concept, illustrating the feasibility of our approach in addressing the TTP problem.

4 Conclusions

4.1 Possible Improvements

To enhance the effectiveness of our approach in tackling the Traveling Thief Problem (TTP), we can explore several avenues for improvement:

Parameter Exploration: One promising solution is to experiment with various problem values, such as population size, mutation rates, the number of generations, and other algorithmic parameters. By systematically adjusting these values and applying the algorithm on a smaller scale, we can fine-tune the algorithm's performance and potentially uncover configurations that lead to more efficient solutions.

Initialization Strategies: Introducing reasonably good, though not necessarily optimal, solutions to the initial populations is another opportunity for improvement. For instance, we can kickstart the algorithm by incorporating fast but suboptimal solutions, like greedy evaluations, which can guide the search in the right direction.

Pareto Front Analysis: It's worthwhile to explore the concept of the Pareto front, which represents a set of solutions where no solution is superior in all objectives. Analyzing the Pareto front can provide valuable insights into trade-offs between competing objectives. By delving into this analysis, we can better understand the balance between the TSP and KS objectives and refine our approach for future experiments, potentially yielding more robust results.

These potential improvements offer exciting opportunities for further enhancing the capabilities of our co-evolutionary approach when dealing with the complex and multifaceted Traveling Thief Problem.

4.2 Final Conclusions

After gaining a comprehensive understanding of the co-evolutionary process and meticulously analyzing the results, it becomes evident that the genetic algorithm employed in our approach is, indeed, capable of yielding satisfactory results.

Furthermore, our research indicates there's plenty of room for improvement. By fine-tuning parameters, introducing effective initialization strategies, and drawing insights from Pareto front analysis, we have the potential to achieve better results in future experiments.

In summary, our findings not only validate the effectiveness of the co-evolutionary approach but also provide motivation for improving the current algorithm resulting in overall better solutions for the presented Traveling Thief Problem.

5 Bibliography

Traveling Salesman Problem:
https://en.wikipedia.org/wiki/Travelling$_salesman_problem$

KnapSack Problem:
https://en.wikipedia.org/wiki/Knapsack$_problem$

GECCO competition source:
https://gecco-2019.sigevo.org/index.html/Competitions
https://www.egr.msu.edu/coinlab/blankjul/gecco19-thief/

Co-Evolution:
https://ieeexplore.ieee.org/document/8454482

Used Library:
https://deap.readthedocs.io/en/master/examples/coev$_coop.html$

Pareto front:
https://en.wikipedia.org/wiki/Pareto$_front$
https://pymoo.org/algorithms/moo/nsga2.html