Group G
Natural Inspired Computation
ECMM409
29/11/2023
TTP: Group Report

1. Introduction

This project aimed to tackle the Traveling Thief Problem (TTP) using Evolutionary Algorithms (EAs), combining the Traveling Salesman Problem (TSP) with the Knapsack Problem. The primary objective was to optimize a thief's route through cities while maximizing profit by selecting valuable items at each city, adhering to weight constraints.

2. Research Undertaken

Research involved studying existing algorithms for TSP, Knapsack, and their integration into the TTP. A thorough review of Evolutionary Algorithms, specifically Genetic Algorithms (GAs), and Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Reinforcement Learning (RL) was conducted to identify suitable approaches for solving this complex optimization problem.

3. Division of Tasks

The project was divided into key tasks as tabulated below:

	Task Division	Individual
1.	Provide code for parsing and processing documents.	Jiayun Shen
2.	Provide code to optimize the TSP problem.	Yushan Kuerban
3.	Provide code to optimize the KNP problem.	Yushan Kuerban
4.	Research Variable Neighborhood Search and provide a research result	Yushan Kuerban
5.	Provide code to calculate fitness.	Kunyang Ye
6.	Write the main logic and ensure that the program runs.	Tianheng Ying
7.	Provided pseudocode for the ant colony algorithm and analyzed its feasibility.	Kunyang Ye
8.	Write PSO algorithm for comparative study	Prabhat Chauhan
9.	Provided pseudocode for reinforcement learning approach and analyzed its feasibility.	Sai Komal Surisetti
10.	Provide code to analyze the results by the Pareto method.	Prabhat Chauhan
11.	Write group report	All members

Formulation of the TTP model, considering smart assumptions on item values, weights, and city distances. Design and implementation of the EA, exploring GA, PSO, ACO and Neural Network approaches.
Evaluation of algorithm performance based on objective functions that balance travel time and accumulated profit.
Comparison of results from different algorithms.

Section 2: Algorithm Development

TTP Model

Assumptions:

Each city has a set of valuable items with randomly generated profits and weights.
The thief's knapsack has a weight constraint based on a fraction of the total item weights.
Travel times between cities are randomly generated.

1. **Genetic Algorithm (GA) Implementation**

Chromosome representation: Concatenation of TSP and Knapsack genes.
Genetic operators: Crossover and mutation tailored for TTP.
Fitness function: Combines TSP travel time and knapsack profit, with penalty terms for exceeding weight limits.

2. [1, Kennedy] Particle Swarm Optimization (PSO) Implementation

Particle representation: Combination of TSP and Knapsack positions.
PSO dynamics: Velocity and position updates considering personal and global bests.

Objective function: Like GA, balancing travel time and knapsack profit.

3. Ant Colony Optimization (ACO) Implementation

Constructive algorithm: Ants build solutions incrementally, considering both TSP and Knapsack components.
Logic flow:

1.Establish an initial ant colony, with each ant representing a solution.

2.In each iteration, ants decide the next city and item based on the concentration of pheromones and emphasis on heuristic information, considering the constraint of the total backpack weight. Heuristic information is usually related to the distance between each city.

3.After each iteration, pheromone concentration evaporates based on a certain rate. And update the pheromone concentration along the routes travelled by the ants, better solutions receive stronger pheromone concentrations.

4.After multiple iterations, the ant colony tends towards better solutions. The best solution within the colony is considered the final solution.

4. Reinforcement Learning (RL) Implementation:

Created a simulated environment that mimics the TTP, where an agent could iteratively learn to make optimal decisions about the sequence of cities to visit and the items to pick or leave.
Reinforcement Algorithm: Advanced Actor-Critic method
Designed a reward system to guide the agent's learning process.

5. Variable Neighborhood Search (VNS)

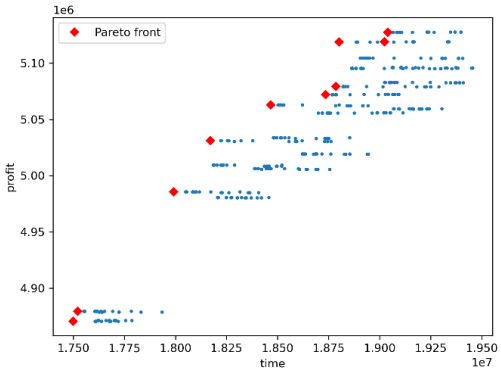
VNS can split items as neighborhoods and try different combinations of items to find the best of them. As part of the TPS part, VNS also can split whole routes to neighborhoods and find the local best. In TTP context, the shaking part can involve changing the order of visited cities or packing plan. This helps escape local optima.
However, while VNS is a powerful and flexible algorithm, its characteristics might not align perfectly with the unique challenges posed by the TTP, making it less suitable compared to other specialized or hybrid algorithms that can better handle the intricacies of this problem.

Section 3: Results

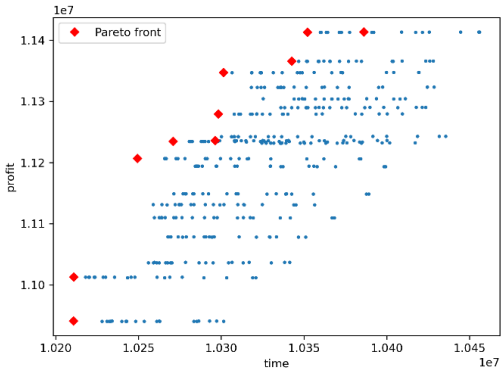
1. Evaluation Metrics

Performance was assessed based on solution quality in terms of travel time, profit, and the [2, E. Zitzler] Pareto front for multi objective optimization. Red points indicate the non-dominated solutions.

Sl.No.	Problem Set	Total Solutions	Non-Dominated Solutions	Estimated Ideal Profit	Estimated Ideal Time
1	a280-n279	498	4	34385.0	16291.666286216343
2	a280-n1395	498	7	333252.0	16270.211487537756
3	a280-n2790	400	15	748619.0	10063.87121407795
4	fnl4461-n4460	400	6	468850.0	17963043.4825153
5	fnl4461-n22300	400	10	5127616.0	17553484.468206145
6	fnl4461-n44600	400	10	11289812.0	10249434.83027215
7	pla33810-n33809	400	5	3518015.0	23703735040.118877
8	pla33810-n169045	800	10	38665370.0	23101463416.539787
9	pla33810-n338090	400	15	85246568.0	12684666441.01839



fnl4461-n22300

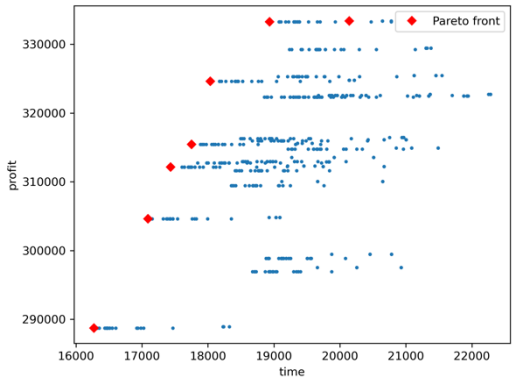


fnl4461-n44600

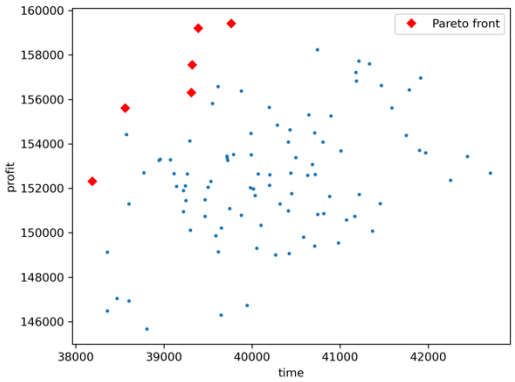
2. Comparative Analysis

Particle Swarm Optimization (PSO):

EA a280_1395 test instance pareto solution



PSO a280_1395 test instance pareto solution



While effective for continuous optimization, PSO might face challenges in handling discrete and combinatorial aspects, which are essential in the TTP with its combination of the Traveling Salesman and Knapsack Problems. PSO's [2] simplicity may be an advantage for certain problems, but the TTP complexity benefits from the adaptability and diversity of EAs. Further, the convergence time linearly increases with no. of particles and no. of dimensions. In the above plot for test instance a280_1395, while the PCO solution are more evenly spaced, EA has emerged as clear winner in finding better profit (twice compared to PSO) and minimum time (half compared to PSO). The PSO was also used to validate the results of EA, utilizing ease of prototyping in PSO.

Ant Colony Optimization (ACO):

ACO mimics the foraging behavior of ants, constructing solutions incrementally through pheromone-based communication. This algorithm has a significant advantage in solving the traveling salesman problem (TSP). However, TTP's combination of TSP and Knapsack introduces continuous aspects that may not align seamlessly with ACO's strengths in discrete spaces. So, we finally gave up on this approach due to the following reasons:

Parameter Sensitivity: It's hard to confirm the best set of parameters including pheromone evaporation rate and the weight of the pheromone concentration and the heuristic function which also referred to as α and β values.

The Limitations of Optimal Solutions: ACO can be stuck local optima, especially in multi-objective optimal problem. It may ignore the globally optimal solutions.

Reinforcement Learning (RL):

Our experimentation focused on training the agent across numerous scenarios to adapt to the dual objectives of maximizing profit while minimizing travel time and knapsack weight.

However, after an extensive period of experimentation and analysis, we have decided not to pursue RL as the primary method for optimizing TTP solutions due to several critical issues:

Computational Complexity: RL algorithms, particularly those involving deep learning, demand substantial computational resources and extended training periods.

Challenges in Reward Function Design: Crafting an effective reward function for TTP proved to be complex, especially in balancing the trade-offs inherent in the problem.
Exploration vs. Exploitation: Efficiently managing the exploration-exploitation trade-off in RL is crucial. However, our experiments showed that this balance is particularly challenging to achieve in the context of TTP.
Generalization Issues: We observed that RL models trained on specific TTP instances often failed to generalize effectively to different problem instances, necessitating frequent retraining or fine-tuning.

Why we choose Evolutionary Algorithms (EAs):
EAs are population-based optimization algorithms inspired by natural selection, suitable for a wide range of optimization problems. Evolutionary Algorithms (EAs) were chosen for solving the Traveling Thief Problem (TTP) due to their versatility, global optimization capabilities, and robustness. EAs can seamlessly handle the integration of Traveling Salesman and Knapsack components, adapt diverse solution spaces, and effectively explore global optima. While Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Reinforcement Learning each have strengths, EAs offered a well-rounded approach suitable for the complex and multi-objective nature of the TTP.

3. Findings

Both GA and PSO showed promise in optimizing the TTP.
GA exhibited better convergence for certain problem instances, while PSO demonstrated resilience in exploring diverse Pareto fronts.

Conclusion
The implemented EA successfully addressed the TTP, offering a flexible framework for optimization. The research and development process illuminated the challenges of integrating TSP and Knapsack components, emphasizing the need for intelligent algorithms like GAs and PSOs in solving complex, real-world optimization problems. Future work may involve fine-tuning algorithm parameters and exploring hybrid approaches to enhance performance.

References:

1. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
2. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," in IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257-271, Nov. 1999, doi: 10.1109/4235.797969.