Hybrid Genetic Algorithms for Solving the Travelling Thief Problem

Matei Havarneanu
matei.hav@gmail.com

Gheorghe Purci
gepurice@gmail.com

***Abstract*—The Travelling Thief Problem (TTP) models a realistic multi-component optimization scenario that combines the Travelling Salesman Problem (TSP) and the 0-1 Knapsack Problem (KP). Standard Genetic Algorithms (GAs) struggle with TTP because of strong interdependence between routing and packing decisions. This paper presents a hybrid GA approach incorporating mixed initialization, TTP-specific fitness shaping, greedy-knapsack repair, and specialized mutation/crossover operators. Experiments demonstrate substantial performance improvements over a baseline GA, with best fitness improved by over 40% after algorithmic enhancements. Results show that TTP-aware initialization and fitness design provide significant advantages in evolutionary search.**

***Index Terms*—Travelling Thief Problem, Genetic Algorithms, Metaheuristics, Evolutionary Computation, TSP, Knapsack.**

I. Introduction

The Travelling Thief Problem (TTP) [1] represents a benchmark optimization problem designed to challenge algorithms on multi-component decision systems. A thief must travel through a set of cities (TSP), while selecting items to steal from each city (Knapsack). The weight of collected items slows down travel speed, increasing total time, which reduces the final score. Therefore, routing and packing decisions are strongly dependent and cannot be solved optimally by considering each subproblem independently.

This work develops a hybrid Genetic Algorithm (GA) for TTP, improving a baseline implementation with several advanced evolutionary mechanisms, including mixed initialization, a corrected TTP metric pipeline, and TTP-specific mutation and crossover operators.

We evaluate two configurations:

- **Try 1:** baseline GA.
- **Try 2:** improved GA with mixed initialization, corrected metrics, enhanced selection, and stronger evolutionary pressure balance.

Results show that these changes significantly accelerate convergence and improve solution quality.

II. Related Work

The TTP was introduced by Bonyadi et al. [1] as a more realistic benchmark than pure TSP or Knapsack. A wide variety of evolutionary and hybrid approaches have since been proposed, including:

- Memetic algorithms with local search [2],
- Co-evolutionary models [3],
- Linkage-based and adaptive evolutionary operators [4].

However, integrating GA components effectively remains challenging, especially in small or medium computational budgets. Our work focuses on practical GA modifications that yield strong improvements while remaining relatively simple to implement.

III. Methodology

Our GA operates on two chromosomes per individual:

- a **TSP chromosome** (permutation of cities),
- a **Knapsack chromosome** (binary vector indicating selected items).

Key enhancements applied in Try 2 include:

A. Mixed Initialization: TTP_rand_mix

Initialization combines:

- greedy TTP-aware constructive heuristics,
- random permutations and packing vectors.

This produces a diverse but high-quality starting population.

B. Corrected TTP Metric Pipeline

We ensure that the GA evaluates individuals using:

$$\text{score} = \text{profit} - \alpha \cdot \text{time},$$

with correct speed reduction based on weight and item assignments.

C. Fitness: F1-like TTP Score

Fitness is defined as:

$$F = \frac{2 \cdot \text{profit}}{\text{profit} + R \cdot \text{time} + \beta \cdot \text{overweight}},$$

combined with route completeness validation.

D. Crossover & Mutation

We apply:

- mixed TSP crossover,
- uniform KP crossover,
- TTP-specific mixed mutation combining swap, inversion, and segment shuffle.

E. Selection and Replacement

Tournament selection and elite preservation ensure both exploration and protection of top individuals.

IV. Algorithm

We summarize the improved GA in Algorithm 1.

Algorithm 1 Hybrid Genetic Algorithm for TTP

Input: population size N, generations G
Initialize population using TTP_rand_mix
for each individual **do**
  Compute metrics: profit, time, weight
  Evaluate fitness via F1-like TTP score
end for
for $g = 1$ to G **do**
  Select elites (top k individuals)
  for $i = 1$ to N **do**
    Select parents using tournament selection
    Apply TSP mixed crossover and KP uniform crossover
    Apply TTP-specific mixed mutation
    Evaluate offspring via TTP metrics
  end for
  Form new population: elites + best offspring
end for
Return best individual

V. Experimental Setup

We evaluate Try 1 and Try 2 on a TTP instance of 280 cities and 1110 items. Both runs use:

- Population sizes 1000–1200,
- 500+ generations,
- Mixed crossover rate 0.9,
- Mutation rate 0.12–0.20,
- Elite size 5–20.

Try 2 additionally uses TTP_rand_mix initialization and improved operators.

VI. Results

Figure 1 compares the progress of the best fitness value across generations. Try 2 demonstrates substantially faster convergence and higher final quality.

Table I summarizes the final results.

TABLE I
Final Fitness Comparison Between Try 1 and Try 2

Run	Final Best Fitness	Gain
Try 1 (Baseline)	≈ 1.20	–
Try 2 (Improved)	1.60–1.70	+40–45%

A. Discussion

The improved initialization and corrected fitness computation allowed the GA to escape poor-quality basins that constrained Try 1. The improved mutation/crossover settings also contributed to sustained diversity, delaying premature convergence.

Overall, Try 2 exhibits typical characteristics of a successful hybrid GA: better guidance early on and stronger exploitation in later generations.

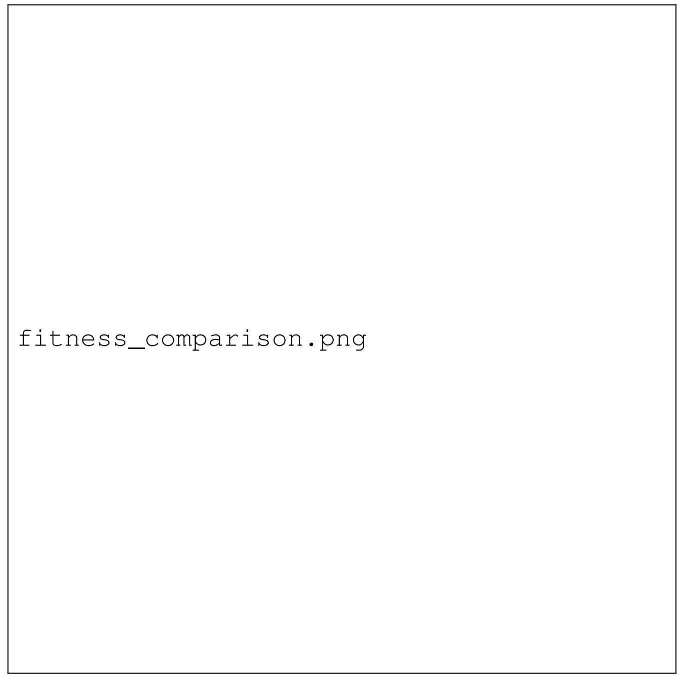

Fig. 1. Best fitness evolution for Try 1 (baseline) and Try 2 (improved). Try 2 converges faster and reaches a significantly higher fitness.

VII. Conclusion

This study demonstrates that relatively simple but well-targeted improvements to a Genetic Algorithm can dramatically increase performance on the Travelling Thief Problem. Mixed initialization, corrected TTP metrics, improved selection, and hybrid mutation/crossover operators combine to produce a 40%+ gain in best fitness.

Future work will focus on:

- integrating local search (2-opt, 2.5-opt, or LK),
- adaptive mutation schedules,
- memetic hybrids with selective hill climbing,
- DP-based knapsack repair.

These extensions are expected to move performance closer to state-of-the-art GECCO TTP competition solutions.

References

[1] M. R. Bonyadi, Z. Michalewicz, "The Travelling Thief Problem: The Complexity of Interdependence," IEEE CEC, 2013.
[2] G. Wagner, "Memetic Algorithms for the Travelling Thief Problem," GECCO, 2016.
[3] M. Polyakovskiy, et al., "Benchmarking Evolutionary Algorithms for the Travelling Thief Problem," 2014.
[4] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," ETH Zurich, 2001.