Investigating the Orienteering Problem with a capacity constraint

Viet Le
20520093@gm.uit.edu.vn

Vu Hoang Huynh
20520864@gm.uit.edu.vn

Vietnam National University, Ho Chi Minh City - University of Information Technology

Subject: Specialized Project

Instructor: PhD. Ngoc Hoang Luong

July 2023

Table of Contents

Table of Contents

Problem description

Thief orienteering problem (ThOP)

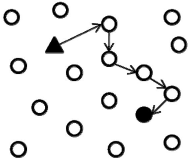
ThOP[1] is a **multi-component optimization problem**, it combines the **Orienteering Problem (OP)** and **Knapsack Problem (KP)**.

[1] André Gustavo dos Santos et al. "The Thief Orienteering Problem: Formulation and Heuristic Approaches". In: *2018 IEEE Congress on Evolutionary Computation (CEC)* (2018), pp. 1–9

Problem description

Orienteering problem (OP)

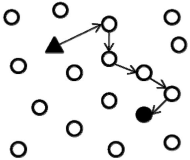
OP is a **routing problem** in which the goal is to determine a path through a given set of points of interest that **maximizes a total score** while **satisfying a given time budget**.



Problem description

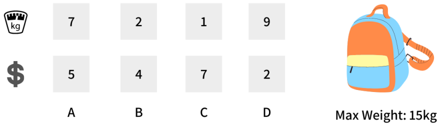
Orienteering problem (OP)

OP is a **routing problem** in which the goal is to determine a path through a given set of points of interest that **maximizes a total score** while **satisfying a given time budget**.

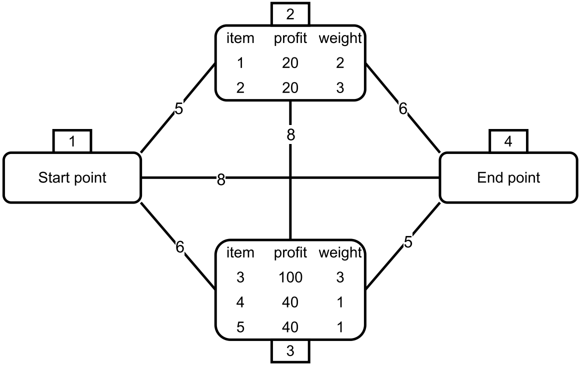


Knapsack problem (KP)

KP is an **optimization problem** in which the goal is to **select a subset of items** from a given set such that **the total value** of the selected items **is maximized**, while the **total weight** of the selected items does **not exceed a given capacity**.



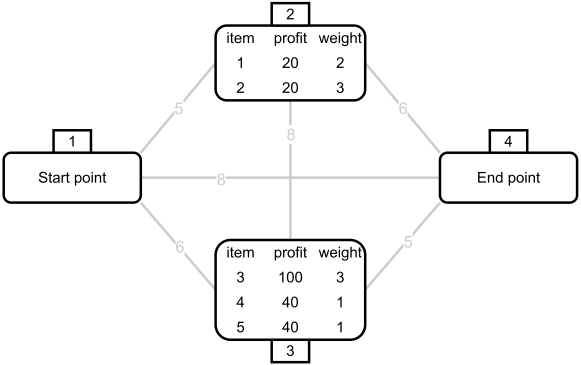
Example



Constraints

- $n = 4, m = 5$
- $v_{min} = 0.1, v_{max} = 1.0, W = 3, T = 75$

Example



Constraints

- $n = 4, m = 5$
- $v_{min} = 0.1, v_{max} = 1.0, W = 3, T = 75$

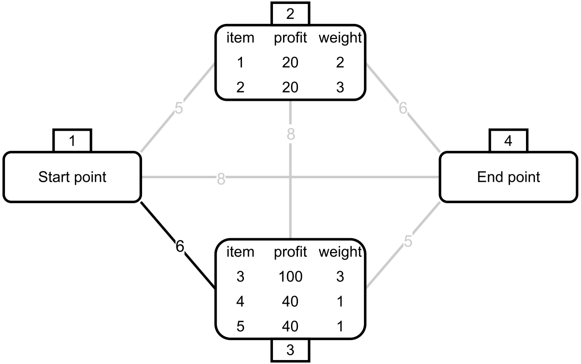
Solution

- $\pi = \langle 1 \rangle$
- $p = \langle 0, 0, 0, 0, 0 \rangle$

Properties

- $p = 0$
- $w = 0$
- $v = v_{max} = 1.0$
- $t = 0$

Example



Constraints

- $n = 4, m = 5$
- $v_{min} = 0.1, v_{max} = 1.0, W = 3, T = 75$

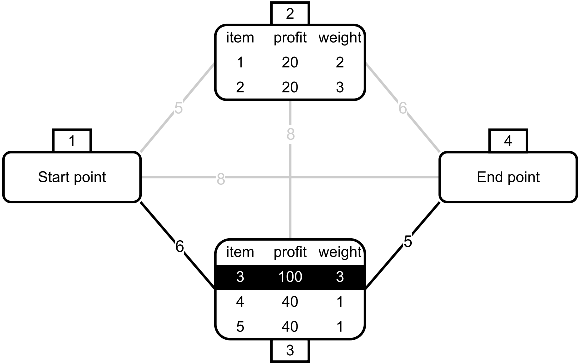
Solution

- $\pi = \langle 1, 3 \rangle$
- $p = \langle 0, 0, 0, 0, 0 \rangle$

Properties

- $p = 0$
- $w = 0$
- $v = v_{max} = 1.0$
- $t = d_{1,3}/v = 6/1.0 = 6$

Example



Constrains

- $n = 4$, $m = 5$
- $v_{min} = 0.1$, $v_{max} = 1.0$, $W = 3$, $T = 75$

Solution

- $\pi = \langle 1, 3, 4 \rangle$
- $p = \langle 0, 0, 1, 0, 0 \rangle$

Properties

- $p = 100$
- $w = 0 + w_3 = 3$
- $v = v_{max} - w(v_{max} - v_{min})/W = 0.1$
- $t = t + d_{3,4}/v = 6 + 5/0.1 = 56$

Motivation

Theoretical Motivation

- ThOP can be a benchmark for evaluating and comparing optimization methods.
- ThOP can contribute to solving its component problems OP and KP, even TTP and TSP.

Motivation

Theoretical Motivation

- ThOP can be a benchmark for evaluating and comparing optimization methods.
- ThOP can contribute to solving its component problems OP and KP, even TTP and TSP.

Practical Motivation

ThOP can be generalized to solve real-world problems:

- Planing a route for a vehicle to collect packages in multiple warehouses with time constraints and capacity limits.
- Planing a route for a rescue team to visit a set of locations to collect supplies and rescue victims.

Table of Contents

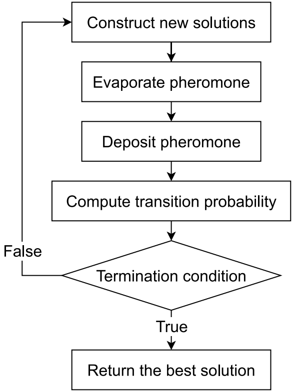
Background

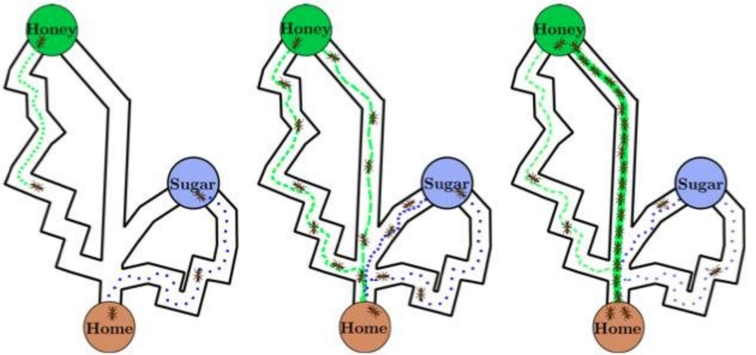
- Ant Colony Optimization (ACO)[2]
- ACO for ThOP[3]
- Randomized Packing Heuristic[3]

[2] Marco Dorigo et al. "Ant colony optimization". In: *IEEE Computational Intelligence Magazine* 1.4 (2006), pp. 28–39

[3] Jonatas B.C. Chagas et al. "Ants can orienteer a thief in their robbery". In: *Operations Research Letters* 48.6 (Nov. 2020), pp. 708–714

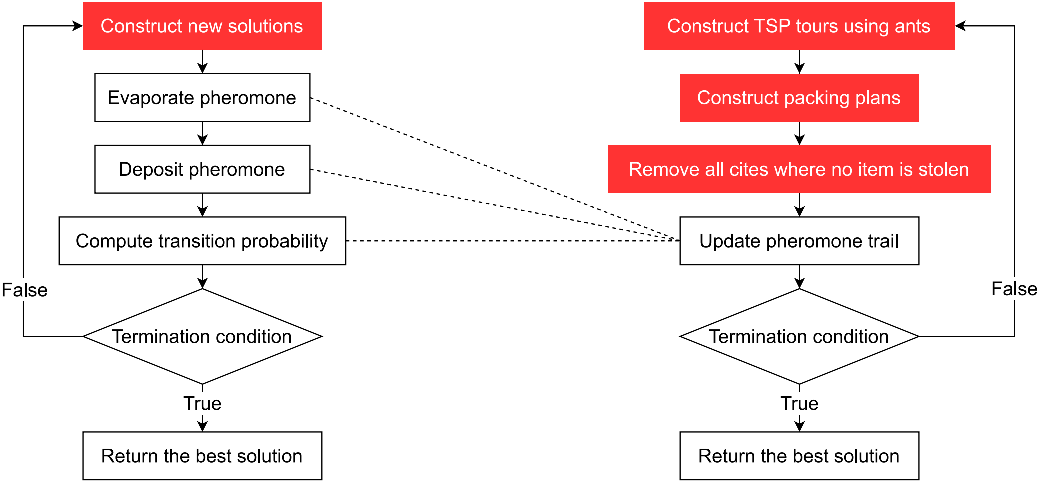
Ant Colony Optimization





(a) Some ants find food sources. (b) Pheromone paths are left in accordance with the food quality. (c) The shortest path to the best food source is followed.

ACO for ThOP



ACO **ACO for ThOP**

Randomized Packing Heuristic

Score

For each item i, there is a score s_i that takes into account a trade-off between the distance that the item needs to be transported, its weight, and its profit.

$$s_i = \frac{p_i^{\theta}}{w_i^{\delta} * d_i^{\gamma}}$$

Randomized Packing Heuristic

$$s_i = \frac{p_i^{\theta}}{w_i^{\delta} * d_i^{\gamma}}$$

Description

- p_i is the profit of item i
- w_i is the weight of item i
- d_i is the distance from the city containing item i to the ending city according to the route
- θ, δ, γ in range $[0, 1]$

Randomized Packing Heuristic

Pseudocode

1. Generate random number θ, δ and γ in the range $[0, 1]$

Randomized Packing Heuristic

Pseudocode

1. Generate random number θ, δ and γ in the range $[0, 1]$
2. Normalize θ, δ, γ such that $\theta + \delta + \gamma = 1$

Randomized Packing Heuristic

Pseudocode

1. Generate random number θ, δ and γ in the range $[0, 1]$
2. Normalize θ, δ, γ such that $\theta + \delta + \gamma = 1$
3. For each item i, compute score s_i

Randomized Packing Heuristic

Pseudocode

1. Generate random number θ, δ and γ in the range $[0, 1]$
2. Normalize θ, δ, γ such that $\theta + \delta + \gamma = 1$
3. For each item i, compute score s_i
4. While not violating constraints, pick an item having next highest score

Randomized Packing Heuristic

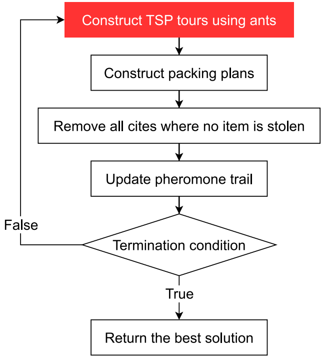
Pseudocode

1. Generate random number θ, δ and γ in the range $[0, 1]$
2. Normalize θ, δ, γ such that $\theta + \delta + \gamma = 1$
3. For each item i, compute score s_i
4. While not violating constraints, pick an item having next highest score
5. Repeat all steps above max_packing_tries times
6. Return the packing plan having best profit

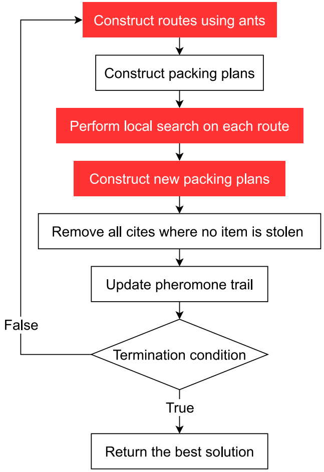
ACO++

- ACO++[4] algorithm is a combination of a heuristic approach based on Ant Colony Optimization with a randomized packing heuristic and with local searches.
- ACO++ outperformed all other algorithms (ACO, BRKGA, ILS, GA) by more than 96% of the total of test cases.[4]

[4] Jonatas B. C. Chagas et al. "Efficiently solving the thief orienteering problem with a max–min ant colony optimization approach". In: *Optimization Letters* 16.8 (Nov. 2021), pp. 2313–2331



ACO for ThOP



ACO++

ACO++ - Drawbacks

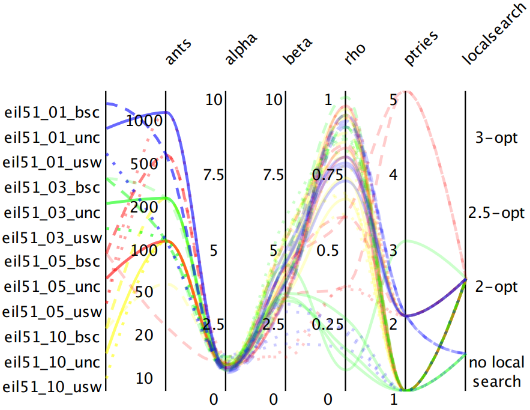
- The packing algorithm relies heavily on randomness.

Packing Algorithm

1. **Generate random number θ, δ and γ in the range $[0, 1]$**
2. Normalize θ, δ, γ such that $\theta + \delta + \gamma = 1$
3. For each item i, compute score s_i
4. While not violating constraints, pick an item having next highest score
5. Repeat all steps above max_packing_tries times
6. Return the packing plan having best profit

ACO++ - Drawbacks

- The hyperparameters are sensitive.



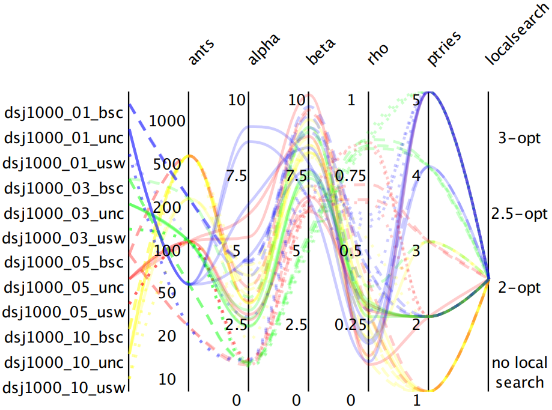
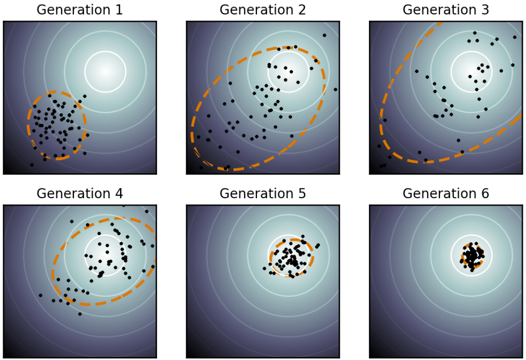


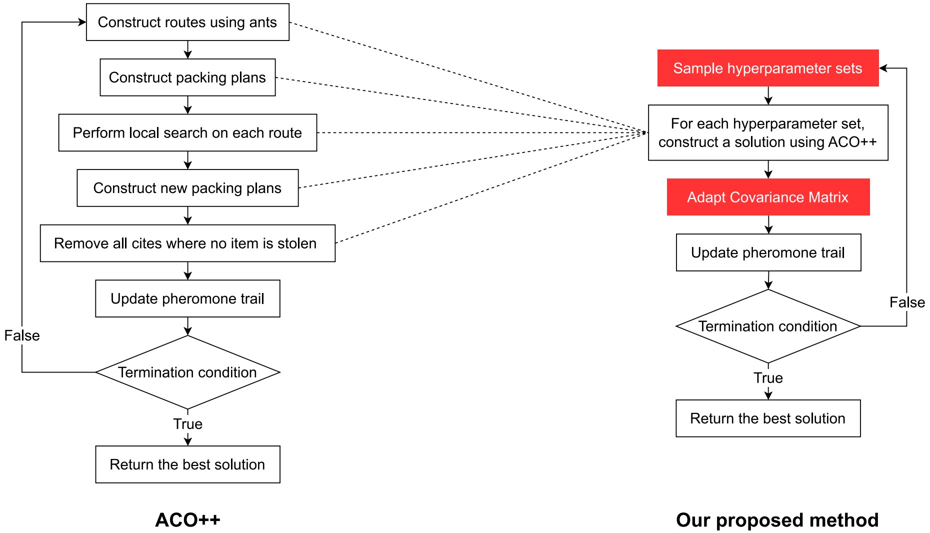
Table of Contents

Introduction

CMA-ES[5] stands for **Covariance Matrix Adaptation Evolution Strategy**, which is a **stochastic**, **derivative-free** method for numerical optimization of **non-linear** or **non-convex continuous** optimization problems.



[5] Nikolaus Hansen et al. "Completely Derandomized Self-Adaptation in Evolution Strategies". In: *Evolutionary Computation* 9.2 (June 2001), pp. 159–195


Construct routes using ants
Construct packing plans
Perform local search on each route
Construct new packing plans
Remove all cites where no item is stolen
Update pheromone trail
Termination condition
False
True
Return the best solution
ACO++
Sample hyperparameter sets
For each hyperparameter set, construct a solution using ACO++
Adapt Covariance Matrix
Update pheromone trail
Termination condition
False
True
Return the best solution
Our proposed method

Experiment - Our proposed method

- Adaptation Mechanism from AACO-NC[6]
- Hierarchical Clustering

[6] Petr Stodola et al. "Adaptive Ant Colony Optimization with node clustering applied to the Travelling Salesman Problem". In: *Swarm and Evolutionary Computation* 70 (2022), p. 101056. ISSN: 2210-6502

Experiment - Result

Result

- **59.03%** of test cases, **our proposed algorithm** is **better 1.11%** on profit on average.
- **40.97%** of test cases, **ACO++** is **better 1.29%** on profit on average.

Table of Contents

Future Works

- Run the experiment with more random seed.

Future Works

- Run the experiment with more random seed.
- Utilize crossover operations in the genetic algorithms during the local search phase.

Future Works

- Run the experiment with more random seed.
- Utilize crossover operations in the genetic algorithms during the local search phase.
- Improve packing strategy.

Thank you for listening

References

[1] André Gustavo dos Santos et al. "The Thief Orienteering Problem: Formulation and Heuristic Approaches". In: *2018 IEEE Congress on Evolutionary Computation (CEC)* (2018), pp. 1–9

[2] Marco Dorigo et al. "Ant colony optimization". In: *IEEE Computational Intelligence Magazine* 1.4 (2006), pp. 28–39

[3] Jonatas B.C. Chagas et al. "Ants can orienteer a thief in their robbery". In: *Operations Research Letters* 48.6 (Nov. 2020), pp. 708–714

[4] Jonatas B. C. Chagas et al. "Efficiently solving the thief orienteering problem with a max–min ant colony optimization approach". In: *Optimization Letters* 16.8 (Nov. 2021), pp. 2313–2331

[5] Nikolaus Hansen et al. "Completely Derandomized Self-Adaptation in Evolution Strategies". In: *Evolutionary Computation* 9.2 (June 2001), pp. 159–195

[6] Petr Stodola et al. "Adaptive Ant Colony Optimization with node clustering applied to the Travelling Salesman Problem". In: *Swarm and Evolutionary Computation* 70 (2022), p. 101056. ISSN: 2210-6502