Alimentation Deep Multiple Optimal Ant Colony Optimization to solve Vehicle Routing Problem with Time Windows

Ha Minh Hieu, Phan Duc Hung, Tran Duc Chinh, Van Duc Cuong, Dao Van Tung, Huynh Thi Thanh Binh
Hanoi University of Science and Technology
Hanoi, Vietnam
{hieu.hm220026,hung.pd214903,chinh.td224936,cuong.vd220021,tung.dv204932}@sis.hust.edu.vn,
binhht@soict.hust.edu.vn

ABSTRACT

Recently, there has been a noticeable increase in the adoption of deep learning techniques to tackle time window routing issues, which are prevalent in various real-world applications. Our paper introduces Alimentation Deep Multiple Optimal Ant Colony Optimization (AMO-ACO), a new neural-enhanced method for solving the Vehicle Routing Problem with Time Windows. Our model handles the problem as a multi-objective task, using transfer learning across constraints and exploring various paths for the ant colony from starting points.

KEYWORDS

Ant Colony Optimization, Artificial Intelligence, Neural Network

ACM Reference Format:
Ha Minh Hieu, Phan Duc Hung, Tran Duc Chinh, Van Duc Cuong, Dao Van Tung, and Huynh Thi Thanh Binh. 2024. Alimentation Deep Multiple Optimal Ant Colony Optimization to solve Vehicle Routing Problem with Time Windows. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, Australia.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3638530.3664050

1 INTRODUCTION

The Vehicle Routing Problem with Time Windows (VRPTW) extends the traditional Vehicle Routing Problem (VRP) by incorporating time constraints known as time windows. These time windows specify the permissible time frames during customer service, such as delivery or pickup. VRPTW holds great importance in practical scenarios and has garnered considerable attention from the Operations Research community.

In our work, we introduce Alimentation Deep Multiple Optimal Ant Colony Optimization (AMO-ACO), a novel neural-enhanced ACO meta-heuristic designed to tackle the challenges posed by the VRPTW problem. Expanding on the framework of DeepACO [1], initially developed for solving the Capacitated Vehicle Routing Problem (CVRP), we present a more advanced model leveraging Graph Neural Network (GNN) [2] and Attention Mechanism [3]. By amalgamating the strengths of both CVRP and TW problems, we employ suitable hybridization techniques to generate robust heuristic measures. Our goal is to balance vehicle utilization, route optimization, and adherence to time windows through effective combination strategies. Additionally, the model is trained to consider multiple potential starting points, enhancing its ability to explore and exploit promising paths. We introduce a new mechanism, Alimentation, facilitating local search for optimal solutions based on three critical criteria: minimal vehicle usage, shortest total distance traveled, and minimized cost function.

In summary, we outline our contributions as follows:

- We propose AMO-ACO, a novel model by combining both constraints from CVRP and TW problems to generate heuristic measures to strike a balance between vehicle utilization, route optimization, and adherence to time windows through effective combination strategies.
- We utilize mechanism for learning potential starting points, improving its capacity to investigate and capitalize on favorable routes.
- During the inference process, we employ the Alimentation mechanism, to leverage superior solutions.

2 PROPOSED METHOD

Each ant will construct k paths based on k start moves and utilize pheromone trails along with problem-specific heuristic measures, which are outputs from model. If an ant is situated in node i at the t^{th} construction step $(s = i)$ and has built a partial solution $s = \{s\}^{(t-1)}$, the probability of choosing node j as its next destination $(s_t = j)$ is typically determined by:

$$P(s_t|s_{<t}, \rho) = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{il} \in N(s_{<t})} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}}, & \text{if } c_{ij} \in N(s_{<t}), \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

Here, ρ is a Combinatorial Optimization Problem (COP) instance, $N(s_{<t})$ is the set of feasible solution components given the partial solution, and α and β are the control parameters ,τ_{ij} and η_{ij} are the pheromone trial and heuristic measures between node i and j respectively. The probability of generating s can be factorized as:

$$P(s|\rho) = \prod_{t=1}^{n} P(s_t|s_{<t}, \rho) \tag{2}$$

We aim to minimize the expected loss function:

$$L(\theta|\rho) = \mathbb{E}_{s \sim P_{\eta_\theta}(\cdot|\rho)} f(s) \tag{3}$$

GECCO '24 Companion, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06.
https://doi.org/10.1145/3638530.3664050

Here:

$$f = c \times \text{number of vehicles} + \text{total distance} \tag{4}$$

We apply a REINFORCE gradient estimator:

$$\nabla L(\theta|\rho) = E_{s\sim P_{\eta_\theta}(\cdot|\rho)}(f(s) - b(\rho))\, \nabla_\theta\, logP_{\eta_\theta}(s|\rho), \tag{5}$$

where $b(\rho)$ is the average objective value of the constructed solutions.

2.1 Learning a problem-specific mapping from an instance to its heuristic measures

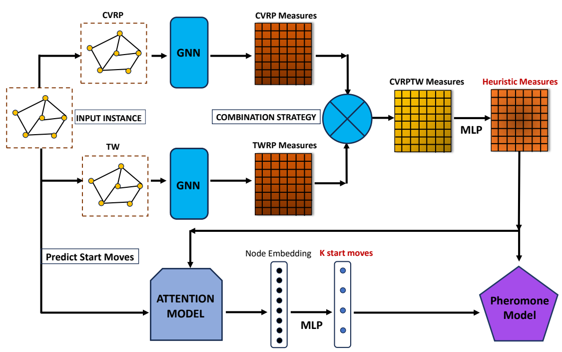

Figure 1: AMO-ACO Model

To parameterize the heuristic space, we introduce a heuristic learner, defined by a GNN with trainable parameters θ. The heuristic learner maps an input COP instance ρ to its heuristic measures and potential start moves.

Specifically, we employ two GNN backbones models [2], which use recursive message passing and edge gating mechanism from graph embedding. One takes input to solve the problem constrained by capacity factors, while the other takes input to address cases constrained by time factors. We train to generate two matrices representing each problem, denoted as M_1 and M_2. These matrices have the same dimensions $((n+1)\times(n+1), d)$, where $n+1$ is the number of nodes excluding depot and d is the hidden dim, as shown in Figure 1.

We utilize two parallel problems to leverage the advantages of each constraint condition. After generating two matrices corresponding to the two conditions, we employ a combination strategy to merge their outputs, resulting in a final matrix that harnesses the strengths of both problems. This combined matrix is fed into a Multi-Layer Perceptron (MLP) decoder, which maps the extracted edge features into heuristic measures.

We experiment and propose various strategies to combine the two problems, such as matrix multiplication and using a different GNN network. Especially, inspired by Evolutionary Multitasking (EMT) [4], we introduce and test a novel method where we treat the two input matrices as an EMT problem, leveraging knowledge transfer techniques to address it [5].

Additionally, we concurrently train an Attention Mechanism based [3] model to work alongside the newly generated heuristic measures.

In numerous instances, a solution to a VRP problem can manifest in various forms when depicted as a sequence of nodes. The initial action significantly impacts the subsequent actions of the agent. This model produces k positions of nodes, inspired by POMO [6], where ants can potentially start moving, as illustrated in Figure 1.

2.2 Alimentation mechanism

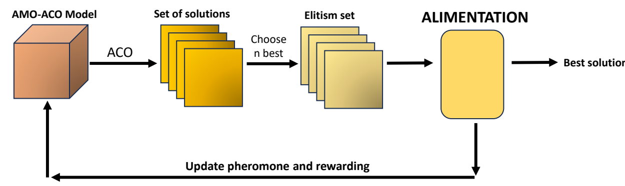

Figure 2: Inference Method

Local search is optionally applied to enhance the constructed solutions. Hence, we utilize the Alimentation mechanism to conduct separate Local Search processes for promising solutions.

More specifically, after constructing solutions in one iteration, we evaluate these solutions against the best solutions thus far, considering three criteria: minimal vehicle usage, shortest total distance traveled, and minimized cost function. Subsequently, specialized local searches are performed, each focused on optimizing one of these criteria, aiming to find better solutions.

If the solution fails to improve after several local search attempts, we proceed to update the reward for these final solutions, considering them as a reward for the search effort. Specifically, for each final solution obtained after t iterations, we update the pheromone according to the formula:

$$\tau_{ij} = \tau_{ij} + \frac{t \times \alpha}{\text{total distance}} \tag{6}$$

Here, α is a predetermined constant that depends on the number of input nodes in the problem.

REFERENCES

[1] Haoran Ye et al. "DeepACO: Neural-enhanced Ant Systems for Combinatorial Optimization". In: (2023). DOI: https://doi.org/10.48550/arXiv.2309.14032.
[2] Chaitanya K. Joshi et al. "Learning the Travelling Salesperson Problem Requires Rethinking Generalization". In: (2022). DOI: https://doi.org/10.48550/arXiv.2006.07054.
[3] Wouter Kool, Herke van Hoof, and Max Welling. "Attention, Learn to Solve Routing Problems!" In: (2019). DOI: https://doi.org/10.48550/arXiv.1803.08475.
[4] Abhishek Gupta et al. "Half a Dozen Real-World Applications of Evolutionary Multitasking and More". In: (2021). DOI: https://doi.org/10.48550/arXiv.2109.13101.
[5] Liang Feng et al. "Explicit Evolutionary Multitasking for Combinatorial Optimization: A Case Study on Capacitated Vehicle Routing Problem". In: (2020). DOI: 10.1109/TCYB.2019.2962865.
[6] Yeong-Dae Kwon et al. "POMO: Policy Optimization with Multiple Optima for Reinforcement Learning". In: (2020). DOI: https://doi.org/10.48550/arXiv.2010.16011.

ACKNOWLEDGMENTS

This research is sponsored by the Office of Naval Research (ONR) Global (ONRG) and the U.S. Army Combat Capabilities Development Command Indo-Pacific (DEVCOM Indo-Pacific), under grant number N62909-23-1-2018.