



HAL
open science

The Bi-objective Electric Autonomous Dial-A-Ride Problem

Yue Su, Sophie N Parragh, Jakob Puchinger, Nicolas Dupin

► **To cite this version:**

Yue Su, Sophie N Parragh, Jakob Puchinger, Nicolas Dupin. The Bi-objective Electric Autonomous Dial-A-Ride Problem. *Odysseus*, May 2024, Carmona, Spain. hal-04600496

HAL Id: hal-04600496

<https://hal.science/hal-04600496>

Submitted on 4 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Bi-objective Electric Autonomous Dial-A-Ride Problem

Yue Su

CERMICS, Ecole des Ponts ParisTech, Champs-sur-Marne, France.

Email: yue.su@enpc.fr

Sophie N. Parragh

Institute of Production and Logistics Management,
Johannes Kepler University Linz, Linz, Austria.

Jakob Puchinger

Métis Laboratory, EM Normandie Business School, Clichy, France.

Nicolas Dupin

Univ Angers, LERIA, SFR MATHSTIC, F-49000 Angers, France.

22/11/2023

1 Introduction

The prevalence of ride-sharing services presents a fundamental trade-off between the operational costs and the users' convenience. While ride-sharing operations reduce operational costs, users may experience certain inconveniences, such as longer ride times when sharing their rides with others. Along with the development of ride-sharing services, the emergence of new techniques, such as electric vehicles and autonomous techniques, has drawn academic interest in operations research to apply a more eco-friendly and comfortable mode of transport. The Electric Autonomous Dial-A-Ride Problem (the E-ADARP) was first introduced by [1], which consists in designing a set of minimum-cost routes for a fleet of electric autonomous vehicles (EAVs) by scheduling them to provide ride-sharing services for users specifying their origins and destinations. In this work, we emphasize the conflicting interests of service providers and users in the objective function of the E-ADARP and investigate the Bi-objective E-ADARP (hereafter BO-EADARP), where the two objectives are the total travel time of all vehicles and the total excess user ride time of all users. We generalize a single objective branch-and-price (B&P) algorithm to the bi-objective case, relying on ideas of [2], to solve it. Numerical results and the managerial insights that we observe from the obtained efficient solutions are summarized.

2 The BO-EADARP Description

The problem is defined on a complete directed graph $G = (V, A)$, where V represents the set of vertices and $A = \{(i, j) : i, j \in V, i \neq j\}$ the set of arcs. V can be further partitioned into several subsets, i.e., $V = P \cup D \cup S \cup O \cup F$. P and D represent the set of all pickup and drop-off vertices, S is the set of recharging stations, and O and F denote the set of origin depots and destination depots, respectively. Each user request is a pair $(i, n + i)$ for $i \in P$ and has a maximum user ride time of m_i . The travel time on each arc $(i, j) \in A$ is denoted as $t_{i,j}$. Detailed mixed-integer-linear program (MILP) of the E-ADARP can be found in [1]. We replace the weighted-sum objective function in [1] to separate objective functions, as follows:

$$\min \sum_{i,j \in V} t_{i,j} x_{i,j}^k \quad (1)$$

$$\min \sum_{i \in P} R_i \quad (2)$$

where $x_{i,j}^k$ is a binary decision variable which denotes whether vehicle k travels from node i to j . R_i denotes the excess user ride time of request $i \in P$ and is formulated as the difference between the actual ride time and direct travel time from i to $n + i$.

3 Methodologies

In this section, we first present the ϵ -constraint method to solve the BO-EADARP, which is used to generate benchmark results. Then, we present the framework of the bi-objective branch-and-price (BOBP) algorithm.

3.1 Epsilon-constraint method

The ϵ -constraint method starts by solving two objectives in lexicographical order with the single-objective B&P. To facilitate reading, we denote $z_1(x)$ as the value of the total travel time and $z_2(x)$ the value of the total excess user ride time for the solution x . In other words, we first solve $\text{lex min}_{x \in \mathcal{X}} \{z_1(x), z_2(x)\}$ and then solve $\text{lex min}_{x \in \mathcal{X}} \{z_2(x), z_1(x)\}$, with \mathcal{X} representing the set of all feasible solutions. We use the term $\text{lex min}_{x \in \mathcal{X}} \{z_1(x), z_2(x)\}$ to describe the process in which we find solutions with the smallest values for $z_2(x)$ among solutions in \mathcal{X} that have the smallest values for $z_1(x)$, and similar for $\text{lex min}_{x \in \mathcal{X}} \{z_2(x), z_1(x)\}$. The obtained non-dominated points z^T and z^B define the search area where other non-dominated points are included. The ϵ -constraint method always optimizes one objective (e.g., $z_1(x)$) while the other is bounded by an ϵ value (i.e., $z_2(x) \leq \epsilon$). In each iteration, the ϵ value is updated with the $z_2(x')$, where x' is the newly-found non-dominated solution.

By using the value of the other objective function to restrict the search iteratively, all the non-dominated points are obtained. The ϵ -constraint method finishes when z^B is reached.

3.2 The BOBP algorithm

The principle of the BOBP algorithm is extended from the single-objective B&P introduced in [3], which aims to divide the original problem into easier subproblems and store them in the form of “nodes”. We denote each subproblem of the BO-EADARP as $P(\eta)$, where η represents the associated node. However, the BOBP algorithm is different from the single-objective case as lower bound and upper bound sets (instead of single numerical values) are used to decide whether to fathom a node. The main ingredients of the BOBP algorithm are presented as follows:

- **Calculate lower bound set and update upper bound set:** On each branch-and-bound node, we calculate the lower bound set with the dichotomic method. To solve each weighted-sum objective problem, the CG algorithm presented in [3] is applied. Once the lower bound set of the analyzed node η (denoted as $\mathcal{L}(\eta)$) is calculated, we first check if new non-dominated points are obtained. If this is the case, the upper bound set \mathcal{U} is updated.
- **Lower bound filtering and node fathoming:** Then, the lower bounds in the set are filtered with the current upper bound set \mathcal{U} , which stores each candidate point that corresponds to the integer solution that is not dominated by other points in the set. The *filtering* process compares the current $\mathcal{L}(\eta)$ with \mathcal{U} and returns a set of non-dominated portions. If no portion is generated after the filtering process, then the analyzed node η can be fathomed, as it is fully dominated by the current upper bound set \mathcal{U} .
- **Branching procedure:** If the analyzed node cannot be fathomed, branching is applied to generate child nodes. We consider three kinds of branching strategies and apply them to each disjoint non-dominated portion. After branching, a set of child nodes is added to the unprocessed node set \mathcal{T} .

The tree search terminates when there is no unprocessed node remaining in \mathcal{T} , and we have the set of non-dominated points \mathcal{Y}_N equals to \mathcal{U} .

4 Numerical Experiments and Discussion

In this work, we solve the BO-EADARP, where the total travel time and the total excess user ride time are considered as two separate objectives. The BO-EADARP is more difficult to solve than the E-ADARP, as one must fully explore the bi-dimension search

area in order to demonstrate the completeness of the Pareto front. To tackle the BO-EADARP, we introduce one criterion space search algorithm (i.e., the ϵ -constraint) and a decision space search algorithm (i.e., the BOBP algorithm). The BOBP algorithm is based on the generalized branch-and-bound algorithm proposed in [2], where the lower bound set is calculated by the CG algorithm ([3]). In the computational experiments, we solve the BO-EADARP with two different algorithms on small-to-medium-sized instances and we compare the generated efficient solutions and their average computational time from different algorithms. Compared with the classic ϵ -constraint method, the BOBP algorithm seems to be more efficient and generates more efficient solutions in a less average computational time. Then, we analyze the obtained efficient solutions, which offer the following managerial insights for different service providers: (1) for profitable service providers, it is possible to significantly improve service quality while keeping near-optimal operational costs; (2) for non-profitable service providers, there exist efficient solutions of high service quality while at lower operational costs. These efficient solutions are very interesting for this kind of service provider. To sum up, the obtained efficient solutions can help decision-makers select Pareto-optimal transportation plans according to their priorities and preferences.

References

- [1] C. Bongiovanni, M. Kaspi and N. Geroliminis, “The electric autonomous dial-a-ride problem”, *Transportation Research Part B : Methodological* 122, 436-456, 2019.
- [2] S. N. Parragh and F. Tricoire, “Branch-and-bound for bi-objective integer programming”, *INFORMS Journal on Computing* 31(4), 805-822, 2019.
- [3] Y. Su, N. Dupin, S. N. Parragh and J. Puchinger, “A Column Generation Approach for the Electric Autonomous Dial-a-Ride Problem”, arXiv preprint arXiv:2206.13496, 2022.