Towards Fair and Efficient Public Transportation: A Bus Stop Model

Martin Bullinger
University of Oxford
Oxford, United Kingdom
martin.bullinger@cs.ox.ac.uk

Edith Elkind
Northwestern University
Evanston, United States
edith.elkind@northwestern.edu

Mohamad Latifian
University of Edinburgh
Edinburgh, United Kingdom
mohamad.latifian@ed.ac.uk

ABSTRACT

We consider a stylized formal model of public transportation, where a set of agents need to travel along a given road, and there is a bus that runs the length of this road. Each agent has a left terminal and a right terminal between which they wish to travel; they can walk all the way, or walk to/from the nearest stop and use the bus for the rest of their journey. The bus can make a fixed number of stops, and the planner needs to select locations for these stops. We study notions of efficiency and fairness for this setting. First, we give a polynomial-time algorithm for computing a solution that minimizes the total travel time; our approach can capture further extensions of the base model, such as more general cost functions or existing infrastructure. Second, we develop a polynomial-time algorithm that outputs solutions with provable fairness guarantees (such as a variant of the justified representation axiom or 2-approximate core). Our simulations indicate that our algorithm almost always outputs fair solutions, even for parameter regimes that do not admit theoretical guarantees.

KEYWORDS

Algorithmic game theory; public transportation; proportional fairness

ACM Reference Format:
Martin Bullinger, Edith Elkind, and Mohamad Latifian. 2025. Towards Fair and Efficient Public Transportation: A Bus Stop Model. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

The use of private vehicles is one of the most significant contributors to pollution. For instance, it is responsible for 43% of the greenhouse gas emissions in the European Union [17]. Therefore, providing well-functioning public transport has repeatedly been identified as a key factor in fighting climate change [9, 25, 39].

The need to model and solve problems related to public transport has been under scrutiny from an operations research perspective; see, e.g., [15] for an extensive literature survey. In the optimization literature, the implementation of public transport infrastructure is commonly seen as a two-stage process consisting of a planning phase and an operational phase. The planning phase is concerned with the design of the transportation network as well as with determining optimal operation frequencies [27, 38]. In the operational phase, the cost of operating public transport should be minimized, e.g., by optimally assigning vehicles to routes or drivers to buses [14, 40]. In both phases, the primary metric used to evaluate the solution quality is the social welfare, i.e., the total/average travel time.

While optimizing the social welfare is a natural and intuitively appealing goal, we believe that it is equally important to approach the design of transportation networks from a fairness perspective. That is, the proposed route networks, frequencies and types of vehicles should benefit not just the majority of the population, but also smaller and less powerful groups, providing usable connections between all neighborhoods and serving the needs of all residents.

We propose to tackle this challenge using the conceptual apparatus of group fairness, building on the ideas of justified representation in multi-winner voting [3] and core stability in cooperative game theory [21]. The intuition that we aim to capture is that sufficiently large groups of agents with similar preferences deserve to be represented in the selected solution, or, more ambitiously, that each group should be allocated resources in proportion to its size.

While we believe that this perspective should be taken into account at all stages of transportation planning, we showcase our approach by applying it to a specific and relatively simple task: choosing the locations of the stops for a fixed bus/train route. Specifically, we consider the setting where the trajectory of the vehicle has been exogenously determined, either by topography (e.g., a mountain road or a river) or by existing infrastructure (e.g., train tracks), the number of stops has been fixed in advance due to bounds on the overall travel time, but the designer still has the freedom to decide where to place the stops. Then, to use the public transport option, the user would have to travel to a nearby stop by using private transport (such as walking, cycling, or using an e-scooter), ride the vehicle towards their destination, and then use private transport again for the last-mile travel. Alternatively, they can opt to use private transport for the entire trip; however, we assume that private transport has higher per-mile cost (measured as physical effort, travel time, or monetary cost) than public transport. Crucially, the agents' decision whether to use the public transport at all is influenced by the location of the stops, so the planner's choices made at this stage may have a dramatic effect on the demand for public transport: positioning the stops without taking into account the agents' travel needs may render the system unusable and push the residents towards private transportation solutions.

For readability, when describing the model, we talk about a bus and the agents walking to/from bus stops; however, we emphasise that our model is applicable to inter-urban transportation, such

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

as train routes and long-distance buses (in which case the agents' last-mile transportation solutions may involve cycling or riding a scooter rather than walking).

1.1 Our Contribution

We put forward a stylized model where there is a bus route that travels the length of a given road, and there are n agents who may ride this bus. Each agent wants to travel between two terminal points located along this road; they can walk all the way, or take the bus (in which case they still need to walk to/from suitable stops). The planner has a budget to build a limited number of bus stops and is given a set of possible stop locations; they then decide which stops should be built. A solution, i.e., a set of bus stop locations, is evaluated according to two criteria. First, we measure it in terms of efficiency, defined by the total time the agents spend on traveling between their terminals. Second, we investigate to what extent a solution offers proportional representation to agent groups. We assume that each of the n agents is entitled to the $1/n$ fraction of the available budget. We then want to achieve outcomes that are group-fair, in the sense that there is no set of agents S such that all agents in S can withdraw their shares of the budget and then pool them to build a pair (resp., a set) of stops such that all agents in S prefer the outcome where only these stops are built to the current outcome; we say that solutions with this property provide justified representation (resp., lie in the core).

Our first contribution is a dynamic program that can efficiently compute cost-minimal solutions. This approach is very flexible in that it still works when we add further features to the model, such as travel costs dependent on non-homogeneous road conditions or existing infrastructure. Moreover, while computing the minimum total cost becomes NP-complete when bus stops have variable costs, our dynamic program still runs in pseudo-polynomial time with respect to the budget.

In the second part of the paper, we focus on finding solutions that provide justified representation (JR) or are (approximately) in the core. Unfortunately, efficiency and justified representation turn out to be incompatible. However, we present a polynomial-time algorithm that operates by selecting bus stops at distances proportional to the density of terminal points, and show that this algorithm finds JR solutions whenever the cost of taking the bus is zero. Moreover, this algorithm offers a 2-approximation to the stronger fairness concept of the core, and exhibits excellent empirical performance (on synthetic data). In contrast, there are instances for which no solution can provide a stronger form of JR.

1.2 Related Work

Fairness considerations have a long-standing history in collective decision-making [see, e.g., 35, 36]. In the context of transportation, fairness is often concerned with justice in terms of equity. It is then measured in terms of, e.g., availability to monetarily disadvantaged population [34], distribution of the impact on health caused by pollution [19], or general access to key infrastructure [31].

Fairness in transportation has been studied in the operations research literature, but the existing work is limited to the operational phase of transportation. For instance, Jozefowiez et al. [23] aim at fairly levelling road occupation to avoid congestion, while Matl et al. [29] are concerned with balancing the workload among a fleet of vehicles that have to jointly cover a given set of trips.

In contrast, our approach, i.e., modeling fairness in terms of proportionality, is rooted in the (computational) social choice literature [see e.g., 13, 22, 32]. Our model can be viewed as a special case of multi-winner voting, and our notion of justified representation is an adaptation of a similar concept in multi-winner approval voting [3]. It is also similar to the notion of proportional fairness in fair clustering [12, 30]. In this stream of literature, Li et al. [28] study approximate core stability, where the approximation is with respect to the size of the deviating coalition (which is similar in spirit to our approach) or with respect to the gain by the deviating agents. Kalayci et al. [24] consider a similar approximation in the context of multi-winner voting, and Chaudhury et al. [10] explore similar ideas in federated learning settings.

We note that placing stops on the line is similar in spirit to facility location [7]; however, our focus in this work is on fairness, whereas much of the facility location literature takes a mechanism design perspective (see, however, [16, 41]). Most related to our paper are models which investigate the same cost function [8, 20]. In particular, the model by Chan and Wang [8] is a special case of our model where $\alpha = 0$ (i.e., taking the bus has no cost), all agents have the same destination, and only two bus stops are built. However, our work differs in two key aspects: we allow for more than two stops to be built and study fairness aspects (rather than strategic manipulation). The facility location literature also considers agents that are interested in more than one location [2, 37], but these works use different cost functions.

2 MODEL

Given a positive integer $k \in \mathbb{N}$, we write $[k] := \{1, \ldots, k\}$. For two numbers $x, y \in \mathbb{Q}$, we denote by $d(x, y) := |x - y|$ the Euclidean distance from x to y. We extend this notation to sets: given a number $x \in \mathbb{Q}$ and a set of numbers $P \subseteq \mathbb{Q}$, we write $d(x, P) := \min_{y \in P} d(x, y)$.

For $\alpha \in [0, 1]$, an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i \in N} \rangle$ of the *α-bus stop problem* (α-BSP) is given by a finite set N of n agents, a finite set $V \subseteq \mathbb{Q}$ of m potential bus stops, a budget $b \in \mathbb{N}$, and, for each agent $i \in N$, their *type* $\theta_i = (\ell_i, r_i)$, where $\ell_i, r_i \in V$ and $\ell_i < r_i$. We refer to the points ℓ_i and r_i as the *terminal points* of i. We denote the set of all terminal points of instance $\mathcal{I}$ by $\mathcal{A}(\mathcal{I}) := \{\ell_i, r_i : i \in N\}$.

A *solution* to an instance of α-BSP is a set of bus stops $S \subseteq V$. A solution S is said to be *feasible* if $|S| \leq b$, i.e., the number of selected bus stops does not exceed the budget.

Our cost function extends models of facility locations in which two stops are build [8, 20]. For each agent $i \in N$, their cost of traveling between two points $\ell, r \in \mathbb{Q}$ is $d(\ell, r)$ if they walk and $\alpha \cdot d(\ell, r)$ if they take the bus. Consequently, the cost of agent i for a solution S to an instance $\mathcal{I}$ is given by

$$c_i^{\mathcal{I}}(S) := \min \begin{cases} d(\ell_i, r_i) \\ \min_{x, y \in S} \left[d(\ell_i, x) + \alpha \cdot d(x, y) + d(r_i, y) \right]. \end{cases} \quad (1)$$

This expression considers two possibilities for i: (1) walking all the way from ℓ_i to r_i, or (2) walking from ℓ_i to a bus stop x, taking the bus to another stop y, and then walking from y to r_i, where x and y are chosen to minimize the overall travel cost.

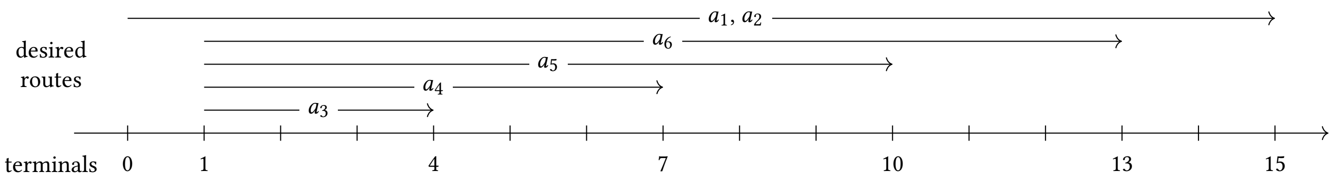

Figure 1: Illustration of Example 2.4. The same instance proves the incompatibility of efficiency and JR in Proposition 2.5.

The *total cost* of a solution $S \subseteq V$ for an instance $\mathcal{I}$ is defined as $c^{\mathcal{I}}(S) := \sum_{i \in N} c_i^{\mathcal{I}}(S)$. Whenever the instance $\mathcal{I}$ is clear from the context, we omit the superscript $\mathcal{I}$. A solution is *efficient* if it minimizes the total cost among feasible solutions.

Apart from efficiency, we are also interested in fairness. Our first concept of fairness builds on ideas from the multi-winner voting and fair clustering literature [3, 4, 12, 28, 30]. Suppose we are given an instance with n agents and budget b. Then, intuitively, each agent is entitled to $\frac{b}{n}$ units of money, so a group of $\lceil \frac{n}{b} \rceil$ agents should be able to dictate the position of one stop. Therefore, one may want to rule out solutions S such that all agents in a group of size at least $\frac{n}{b}$ can lower their costs by abandoning S and building a single stop. However, this condition is too weak, as no agent benefits from a single stop. Hence, we strengthen it by considering groups of agents that are entitled to *two* stops.

Definition 2.1. A solution $S \subseteq V$ is said to provide *justified representation (JR)* if for every set of agents $M \subseteq N$ with $|M| \geq \frac{2n}{b}$ and every pair of stops $T \subseteq V$ there exists an agent $i \in M$ such that $c_i(T) \geq c_i(S)$. Moreover, a solution $S \subseteq V$ is said to provide *strong justified representation* if for every set of agents $M \subseteq N$ with $|M| \geq \frac{2n}{b}$ and every pair of stops $T \subseteq V$ there exists an agent $i \in M$ such that $c_i(T) > c_i(S)$ or for all agents $i \in M$ it holds that $c_i(T) \geq c_i(S)$.

The key distinction between JR and strong JR is that, to define the former, we only consider deviations to pairs of stops that are strictly preferred by each agent in M, whereas to define the latter, we also consider deviations that make no agent in M worse off while making at least one member of M strictly better off. Thus, an outcome that provides strong JR also provides JR, but the converse is not necessarily true.

Justified representation can also be viewed as a notion of stability: a group of at least $\frac{2n}{b}$ agents can deviate by building two stops, and we require that there is no group such that all group members can benefit from a deviation. Note that a budget of 2 is exactly the proportion of the budget that a group of size $\lceil \frac{2n}{b} \rceil$ is entitled to spend. By generalizing this idea to groups of arbitrary size, where a deviating group is allowed to spend a fraction of the budget that is proportional to the group size, we arrive to the concept of the core. We note that the core has been considered as a notion of fairness in a variety of contexts, ranging from participatory budgeting to clustering [1, 5, 10, 11, 18].

Definition 2.2. A subset of agents $M \subseteq N$ is said to *block* a solution $S \subseteq V$ if there exists a subset of stops $T \subseteq V$ such that $|T| \leq |M| \cdot \frac{b}{n}$ and $c_i(T) < c_i(S)$ for all agents $i \in M$. Moreover, a subset of agents $M \subseteq N$ is said to *weakly block* a solution $S \subseteq V$ if there exists a subset of stops $T \subseteq V$ such that $|T| \leq |M| \cdot \frac{b}{n}$, $c_i(T) \leq c_i(S)$ for all agents $i \in M$, and there exists $j \in M$ with $c_j(T) < c_j(S)$. A solution is said to be in the *strong core* if it is not weakly blocked.

Equivalently, a solution $S \subseteq V$ is in the *core* if, for every set of agents $M \subseteq N$ and every set of stops $T \subseteq V$ with $|T| \leq |M| \cdot \frac{b}{n}$, there exists an agent $i \in M$ such that $c_i(T) \geq c_i(S)$. The core is a demanding solution concept. Therefore, we also define a multiplicative approximation of the core (which we call the β-core), where, for a group of agents to be allowed to deviate by building t stops, the size of the group should be at least β times the number of agents who 'deserve' t stops. Note that the 1-core is identical to the core.

Definition 2.3. Let $\beta \geq 1$. A solution $S \subseteq V$ is said to be in the *β-core* if, for every set of agents $M \subseteq N$ and every set of stops $T \subseteq V$ with $\beta \cdot |T| \leq |M| \cdot \frac{b}{n}$, there exists an agent $i \in M$ such that $c_i(T) \geq c_i(S)$.

We provide an example to illustrate our model.

Example 2.4. Let $\alpha \in [0, 1)$. Consider the instance $\langle N, V, b, (\theta_i)_{i \in N} \rangle$, depicted in Figure 1, with $N = \{a_i : i \in [6]\}$, $V = \{0, 1, 4, 7, 10, 13, 15\}$, and $b = 6 = |N|$. The agents' types are as follows: $\theta_{a_1} = \theta_{a_2} = (0, 15)$, $\theta_{a_3} = (1, 4)$, $\theta_{a_4} = (1, 7)$, $\theta_{a_5} = (1, 10)$, and $\theta_{a_6} = (1, 13)$.

Consider the solution $S^* = \{1, 4, 7, 10, 13, 15\}$. It holds that $c(S^*) = 60\alpha + 2(1 - \alpha)$. However, S^* does not provide JR. To see this, consider $M = \{a_1, a_2\}$ and $T = \{0, 15\}$. Then, $|M| = \frac{2n}{b}$ and $c_a(T) < c_a(S^*)$ for each $a \in M$.

In contrast, any solution $S' = V \setminus \{x\}$ for $x \in \{4, 7, 10, 13\}$ provides JR because then S' contains the terminals of all except possibly one agent, who is only entitled to one stop. ◁

We can use Example 2.4 to prove that providing JR is incompatible with minimizing total cost, apart from the trivial case of $\alpha = 1$ where walking and taking the bus takes the same time.

Proposition 2.5. *For each $\alpha \in [0, 1)$ there exists an instance of α-BSP such that no feasible solution can both minimize the total cost and provide JR.*

Proof. Consider the instance in Example 2.4 and the solution $S^* = \{1, 4, 7, 10, 13, 15\}$. We already know that S^* does not provide JR. To complete the proof, we show that S^* is the unique solution of minimum cost. Recall that $c(S^*) = 60\alpha + 2(1 - \alpha)$.

First, note that the sum of lengths of the agents' routes is $2 \cdot 15 + 3 + 6 + 9 + 12 = 60$. Hence, the cost of every solution is at least 60α.

We now show that every other solution costs more than S^*. Fix a solution $S \subseteq V$ with $|S| = 6$. If $1 \notin S$, then the walking cost of each of the agents in $\{a_3, a_4, a_5, a_6\}$ is at least 1, so $c(S) \geq 60\alpha + 4(1 - \alpha)$. Therefore, we may assume that $1 \in S$. Moreover, if $15 \notin S$, then the walking costs of a_1 and a_2 are at least 2, so the solution is worse than S^*. Hence, we may also assume that $15 \in S$.

Next, assume that $0 \in S$ and hence $\{0, 1, 15\} \subseteq S$. Therefore, we only have 3 stops to cover the right terminals of agents $\{a_3, a_4, a_5, a_6\}$. One of these agents has to walk a distance of at least 3, unless $S = \{0, 1, 4, 7, 10, 15\}$. In the latter case, a_6 has to stay on the bus for 2 units of distance past their right terminal and then walk back. Hence, $0 \in S$ implies $c(S) \geq \min\{60\alpha + 3(1-\alpha), 62\alpha + 2(1-\alpha)\} > c(S^*)$. Thus, we conclude that $0 \notin S$ and therefore $S = V \setminus \{0\} = S^*$. □

In fact, the incompatibility observed in Proposition 2.5 can be strengthened further: it holds for approximate JR and, in case of the 0-BSP, even for approximate minimum cost. We provide the details in the full version of the paper [6]. However, if we replace cost minimality with Pareto optimality, the incompatibility no longer holds: by applying Pareto improvements, we can transform a solution providing JR into a Pareto-optimal solution providing JR.

Definition 2.6. Given an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i \in N} \rangle$, a solution S is said to *Pareto-dominate* another solution S' if $c_i(S) \leq c_i(S')$ for all $i \in N$ and there exists an agent $j \in N$ with $c_j(S) < c_j(S')$. A solution S^* is *Pareto-optimal* for $\mathcal{I}$ if it is not dominated by any other solution.

Proposition 2.7. *Let $\alpha \in [0, 1]$. Then every instance of α-BSP that admits a solution providing JR also admits a solution that is Pareto-optimal and provides JR.*

Proof. Let $\alpha \in [0, 1]$, and consider an instance of α-BSP that admits a solution S providing JR. Suppose that S is not Pareto-optimal. Then it is Pareto-dominated by another solution S_1. We claim that S_1, too, provides JR. To see this, consider a subset of agents $M \subseteq N$ with $|M| \geq \frac{2n}{b}$ and a pair of bus stops $T \subseteq V$. Since S provides JR, there exists an agent $i \in M$ with $c_i(S) \leq c_i(T)$. Hence, $c_i(S_1) \leq c_i(S) \leq c_i(T)$, which establishes that S_1 provides JR. If S_1 is not Pareto-optimal, there exists another solution S_2 that Pareto-dominates it, and our argument shows that S_2 provides JR as well. We can continue in this manner until we reach a Pareto-optimal solution; this will happen after a finite number of steps, as each step reduces the total cost. □

Proposition 2.7 extends to approximate JR solutions; the proof remains the same. We note, however, that Proposition 2.7 does not offer an efficient algorithm to find a Pareto-optimal solution that provides JR, as it is not clear how to compute Pareto improvements in polynomial time.

3 EFFICIENCY

In this section, we show that efficient solutions, i.e., solutions of minimum total cost, can be computed in polynomial time. Our algorithm extends to a more general version of our model, where agents' terminals need not be contained in V.

Our algorithm is based on a dynamic program, which iteratively considers adding new stops to the solution. Capturing our problem by a dynamic program is challenging, because each agent's cost depends on the placement of *two* stops. The crucial observation that enables us to circumvent this difficulty is that, to perform cost *updates* in the dynamic program, it suffices to know the rightmost stop in the current solution. This idea is formalized by Lemma 3.1. All missing proofs can be found in the full version of the paper [6].

Lemma 3.1. *Let $\alpha \in [0, 1]$ and let $\mathcal{I} = \langle N, V, b, (\theta_i)_{i \in N} \rangle$ be an instance of the α-BSP problem. Let $S \subseteq V$, and $h = \max S$. Then for each $k \in V$ with $k > h$ the quantity $c(S) - c(S \cup \{k\})$ is a function of h and k that can be computed in time $O(1)$.*

Lemma 3.1 enables us to set up a two-dimensional dynamic program for computing the minimum total cost of a solution to an α-BSP instance with a given number of stops.

Theorem 3.2. *For $\alpha \in [0, 1]$, we can compute a minimum-cost solution for an instance of α-BSP in time $O(nm + m^3)$.*

Proof. Let $\alpha \in [0, 1]$, and consider an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i \in N} \rangle$ of α-BSP where $V = \{v_1, \dots, v_m\}$ with $v_1 \leq v_2 \leq \dots \leq v_m$, i.e., potential stops are sorted from left to right and v_j represents the j-th stop. We will assume $b \leq m$, as otherwise there is an optimal-cost solution that builds a stop at each location. For a solution S with $\max S = v_h$ and $k > h$, let $\Delta(h, k) := c(S) - c(S \cup \{v_k\})$ denote the total reduction in the agents' costs from adding stop v_k to S. By Lemma 3.1, we know that $\Delta(h, k)$ only depends on v_k and v_h, and can be computed in $O(1)$ time. Let us set up a dynamic program $\mathbf{dp}[h, c]$, where

- $h \in \{0, 1, \dots, m\}$ represents the rightmost stop that has been added to the solution, where 0 means that no stop has been added yet, and
- $c \in \{0, \dots, b\}$ represents the budget used so far.

Then, $\mathbf{dp}[h, c]$ is the minimum total cost of a solution that selects at most c stops, with v_h being the rightmost selected stop.

We initialize with

(a) $\mathbf{dp}[0, c] = \mathbf{dp}[1, c] = \sum_{i \in N} (r_i - \ell_i)$ for all $c \in \{0, \dots, b\}$,
(b) $\mathbf{dp}[h, 0] = \infty$ for all $h \in \{1, \dots, m\}$.

Case (b) captures the impossible situation of selecting at least one stop ($h > 0$) while spending no budget. We prevent this case by setting the total cost to ∞. As we assume $b \leq m$, in total the initialization takes $O(nm)$ time.

For updating, we use the cost change function Δ. For $h \in \{1, \dots, m\}$, and $c \in \{1, \dots, b\}$ we update as follows:

$$\mathbf{dp}[h, c] = \min_{h' \in \{0, \dots, h-1\}} \mathbf{dp}[h', c-1] - \Delta(h', h). \tag{2}$$

That is, we consider the position of the stop h' that precedes h in the solution, and evaluate the cost reduction from adding h to a solution that ends with h'. Clearly, the updates can be computed in time $O(m)$, using the update formulas provided by Lemma 3.1. As we assume that $b \leq m$, our table has at most m^2 entries, and can be filled in time $O(m^3)$. Hence, the total running time is $O(nm + m^3)$.

It is not hard to see that our dynamic program is correct; we provide a formal proof of correctness in the full version of the paper. Hence, the minimum cost of a feasible solution for $\mathcal{I}$ is equal to $\min_{h \in \{0, \dots, m\}} \mathbf{dp}[h, b]$. We can efficiently extract an explicit feasible solution of minimum cost by standard techniques. □

We note that we can also assume that $b \leq 2n$ (otherwise, we can build a stop at each terminal), and hence the running time of our algorithm can also be bounded as $O(n^2 + nm^2)$. However, we expect this bound to be less useful than the one stated in Theorem 3.2, since in practical applications it is likely that $m << n$.

Notably, the computations in the dynamic program developed in the proof of Theorem 3.2 are merely updates of the sums of costs for all agents. This allows us to extend Theorem 3.2 to incorporate further features that may be important for some applications.

First, the theorem extends to more general cost functions. Consider a setting where the time to travel between two stops depends on factors other than the distance. For instance, the bus route might encompass intervals with different speed limits, or there may be hilly or curvy roads, where the bus needs to slow down. Hence, the travel costs need not be homogeneous. However, typically the travel cost of a route only depends on the costs of its segments. To capture this, we introduce the notion of separable travel costs, and formally introduce the separable-cost BSP problem, which generalizes the α-BSP problem defined earlier in the paper.

A function $d\colon \mathbb{Q}^2 \to \mathbb{Q}$ is called *separable* if for all $x < y < z$ it holds that $d(x,z) = d(x,y) + d(y,z)$. An instance $\mathcal{I}$ of *separable-cost BSP* is given by a tuple $\langle N, V, b, (\theta_i)_{i\in N}\rangle$ and separable cost functions $d^W : V \times V \to \mathbb{Q}$ and $d_i^B : V \times V \to \mathbb{Q}$ for all $i \in N$ (where the superscripts refer to walking and taking the bus). The agents' costs are then defined as

$$c_i^{\mathcal{I}}(S) := \min \begin{cases} d^W(\ell_i, r_i) \\ \min_{x,y\in S} \left[d^W(\ell_i, x) + d^B(x,y) + d^W(y, r_i) \right]. \end{cases}$$

We can generalize Lemma 3.1 to separable-cost BSP by replacing the costs for walking and taking the bus by the respective separable cost functions in all update formulae. Note that the definition of separable-cost BSP does not require that taking the bus is faster than walking. Our computation can account for this, by allowing the agents to walk rather than take the bus for segments where walking is faster. With the generalized update formulae, we can run the dynamic program from Theorem 3.2 and obtain the following theorem.

Theorem 3.3. *For separable-cost BSP, we can compute the minimum cost in time $O(nm + m^3)$.*

As a second extension, we assume that there already is an existing set of bus stops, but we have a budget to build b additional stops. An instance of α-BSP *with existing bus stops* consists of an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ of the base model together with a set $E \subseteq \mathbb{Q}$ of existing bus stops. The cost of a solution $S \subseteq V$ for agent i is then computed as

$$c_i^{\mathcal{I}}(S) := \min \begin{cases} d(\ell_i, r_i) \\ \min_{x,y\in S\cup E} \left[d(\ell_i, x) + \alpha \cdot d(x,y) + d(y, r_i) \right]. \end{cases}$$

This case can be solved by a simple modification of the dynamic program in Theorem 3.2. Since we can still update a cell in time $O(m)$, the running time is the same as in Theorem 3.2.

Theorem 3.4. *For α-BSP with existing bus stops, we can compute the minimum cost in time $O(nm + m^3)$.*

As a third extension, we consider the case where bus stops do not have identical costs: indeed, construction costs may vary depending on, e.g., ease of access. An instance of α-BSP *with bus stop costs* consists of an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ of the base model together with a *budget function* $\gamma\colon V \to \mathbb{N}$. A solution $S \subseteq V$ is *feasible* if $\sum_{i\in S} \gamma(i) \le b$. Clearly, we can still apply the dynamic program in Theorem 3.2. However, we can no longer assume that $b \le m$. Hence, we obtain a running time of $O(nm^2 + m^3 b)$, which is only pseudo-polynomial due to the dependency on b. If the bus stop costs are represented in unary, the running time remains polynomial. However, for bus stop costs represented in binary, we obtain a computational hardness result, via a reduction from Knapsack.

Proposition 3.5. *Let $\alpha \in [0,1)$. Then, the following decision problem is* NP*-complete: given an instance $\mathcal{I}$ of α-BSP with bus stop costs represented in binary and a rational number $q \in \mathbb{Q}$, decide if there exists a feasible solution S with $c^{\mathcal{I}}(S) \le q$.*

We conclude this section with a structural result regarding minimum-cost solutions. Interestingly, as long as both terminals of all agents belong to the set of potential bus stops, there is a minimum-cost solution that places all stops at the agents' terminals. We remark that this is the only place in this section where the assumption $\mathcal{A}(\mathcal{I}) \subseteq V$ plays a crucial role.

Proposition 3.6. *For every $\alpha \in [0,1]$ and every instance $\mathcal{I}$ of α-BSP there exists a minimum-cost feasible solution S^* with $S^* \subseteq \mathcal{A}(\mathcal{I})$.*

An interesting consequence of Proposition 3.6 is that it enables us to deal with yet another variant of the base model, where we allow infinitely large sets of potential bus stops (e.g., intervals of $\mathbb{Q}$). Indeed, we can then transform an instance $\mathcal{I}$ by setting $V := \mathcal{A}(\mathcal{I})$ and apply Theorem 3.2. In particular, this covers the case where $V = \mathbb{Q}$, which can be viewed as a continuous version of our model.

4 FAIRNESS

We now turn to the consideration of fairness. Our main contribution is an algorithm that efficiently computes outcomes that provide JR if $\alpha = 0$, i.e., if taking the bus has zero cost. Moreover, the solutions computed by this algorithm lie in the 2-approximate core (and the bound of 2 is tight). Besides these theoretical guarantees, we establish that our algorithm has good empirical performance: in more than 99.9% of our (synthetically generated) instances, the algorithm finds an outcome in the core, even for $\alpha > 0$.

4.1 Theoretical Possibilities and Limitations

We first show that JR solutions exist if taking the bus has zero cost. For this, we consider Algorithm 1. Its key idea is to order all terminal points and select them iteratively from left to right whenever we have passed sufficiently many agent terminals (counted with multiplicities). This approach is similar to the CommitteeCore algorithm by Pierczyński and Skowron [33], which is used to find outcomes in the core of 1-dimensional multi-winner elections.

Given an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ of 0-BSP, Algorithm 1 starts by sorting $V = \{v_1, \dots, v_m\}$ so that $v_1 \le v_2 \le \dots \le v_m$. Then, for each $j \in [m]$, the algorithm computes x_j as the number of agent terminals at or to the left of v_j. Next, for $k \in [b]$, it computes s_k as the leftmost element $v_j \in V$ with $x_j \ge k \left\lfloor \frac{2n}{b} \right\rfloor$; it then returns $S = \{s_k : k \in [b]\}$. Since $|S| \le b$, S is a feasible solution.

The proof of correctness of Algorithm 1 is based on the following technical lemma.

Lemma 4.1. *Let $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ be an instance of 0-BSP, and let S be a solution for $\mathcal{I}$. Consider a coalition $M \subseteq N$ that prefers*

Algorithm 1 Fair solutions for α-BSP.

Input: Instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ of α-BSP
Output: A solution S

```
Sort V = {v_1, ..., v_m} so that v_1 ≤ v_2 ≤ ··· ≤ v_m
for j = 1, ..., m do
    x_j ← |{i ∈ N : ℓ_i ≤ v_j}| + |{i ∈ N : r_i ≤ v_j}|
for k = 1, ..., b do
    s_k ← min{v_j : x_j ≥ k ⌊2n/b⌋}
S ← {s_k : k ∈ [b]}
return S
```

$T \subseteq V$ to S, and an agent $i \in M$. Suppose that, when T is built, i walks from their left terminal ℓ_i to $\ell' \in T$, then takes the bus from ℓ' to $r' \in T$, and then walks from r' to their right terminal r_i. If there exists an $\ell \in S$ with $\ell_i \le \ell \le \ell'$ or $\ell' \le \ell \le \ell_i$, then for every stop $r \in S$ it holds that $d(\ell, \ell') < d(r, r')$. Similarly, if there exists a stop $r \in S$ with $r_i \le r \le r'$ or $r' \le r \le r_i$, then for every stop $\ell \in S$ it holds that $d(\ell, \ell') > d(r, r')$.

Proof. Assume that M, S, T, i, ℓ' and r' are as in the statement of the lemma and that there exists an $\ell \in S$ between ℓ_i and ℓ'. Fix an r^* in $\arg\min_{x\in S} d(r_i, x)$. By assumption, we have $c_i(T) < c_i(S)$. Hence, for every $r \in S$, we conclude that

$$d(\ell_i, \ell') + d(r_i, r') < d(\ell_i, \ell) + d(r_i, r^*) \qquad \text{and hence}$$
$$d(\ell_i, \ell') - d(\ell_i, \ell) < d(r_i, r^*) - d(r_i, r'). \tag{3}$$

Since ℓ lies between ℓ_i and ℓ', we have $d(\ell, \ell') = d(\ell_i, \ell') - d(\ell_i, \ell)$; substituting this into (3), we obtain

$$d(\ell, \ell') < d(r_i, r^*) - d(r_i, r') \le d(r_i, r) - d(r_i, r') \le d(r, r'),$$

where the first transition is by the choice of r^* and the second transition uses the triangle inequality. The proof for the second statement of the lemma is analogous. □

Theorem 4.2. *Algorithm 1 runs in polynomial time, and for 0-BSP it computes feasible solutions that provide JR.*

Proof. Clearly, Algorithm 1 runs in polynomial time. We claim that the solution S computed by Algorithm 1 provides JR.

Indeed, assume for contradiction that there exists a coalition $M \subseteq N$ of size $|M| \ge 2 \cdot \frac{n}{b}$ and a pair of stops $T = \{\ell', r'\} \subseteq V$ such that $c_i(T) < c_i(S)$ for all $i \in M$.

Let $L = \{\ell_i : i \in M\}$. Since there are at most $\lfloor \frac{2n}{b} \rfloor$ terminals between every two consecutive stops in S, there exists an $\ell^* \in S$ such that at least one of the terminals in L is before ℓ^* or exactly at ℓ^*, and at least one of the terminals in L is after ℓ^* or exactly at ℓ^*. Hence, if $\ell^* \le \ell'$, then there exists an agent $i \in M$ with $\ell_i \le \ell^* \le \ell'$ and if $\ell' \le \ell^*$, then there exists an agent $i \in M$ with $\ell' \le \ell^* \le \ell_i$. By Lemma 4.1 for every $r \in S$ we have $d(\ell^*, \ell') < d(r, r')$.

By a similar argument, there exists an agent $j \in M$ and a stop $r^* \in S$ where $r_j \le r^* \le r'$ or $r' \le r^* \le r_j$. By Lemma 4.1 for every $\ell \in S$ we have $d(r^*, r') < d(\ell, \ell')$. Setting $r = r^*$ and $\ell = \ell^*$, we obtain a contradiction. □

Example 4.3. Consider the execution of Algorithm 1 on the instance from Example 2.4. There, we had $V = \{0, 1, 4, 7, 10, 13, 15\}$, leading to the values $(x_j)_{j=1}^7 = (2, 6, 7, 8, 9, 10, 12)$. This leads to $(s_k)_{k=1}^6 = (0, 1, 1, 7, 13, 15)$. Hence, Algorithm 1 returns $S = \{0, 1, 7, 13, 15\}$ for this instance. ◁

The example shows that, while Algorithm 1 always returns a feasible solution that provides JR, the obtained solution may fail to exhaust the budget. In such cases, having achieved JR via Algorithm 1, we can distribute the remaining budget to accomplish other goals. For instance, as per Theorem 3.4, we can extend the solution to lower the total cost as much as possible. Note that, after we add stops, the solution continues to provide JR.

A natural follow-up question is whether Theorem 4.2 can be extended to arbitrary $\alpha \in (0, 1)$. Unfortunately, our next result shows that this is not the case.

Proposition 4.4. *Let $\alpha \in (0, 1)$. Then, for α-BSP, Algorithm 1 may return a solution that does not provide JR.*

We can show further fairness guarantees for Algorithm 1, namely that it computes solutions in the 2-core. However, the approximation guarantee of 2 is tight.

Theorem 4.5. *Let $\alpha \in [0, 1)$. Then, for α-BSP, Algorithm 1 computes solutions in the 2-core. However, for each $0 < \epsilon \le 1$, it may output solutions that are not in the $(2 - \epsilon)$-core.*

Proof. Let $\alpha \in [0, 1)$. We start by proving that the output of Algorithm 1 is in the 2-core. Consider an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i\in N}\rangle$ and a solution S computed by Algorithm 1 on $\mathcal{I}$. Assume for contradiction that there is a set of agents $M \subseteq N$ and a set of stops $T \subseteq V$ with $|T| \le \frac{b}{2n} \cdot |M|$ such that $c_i(T) < c_i(S)$ for all $i \in M$.

For each $t \in T$, define $\ell^t := \max(\{s \in S : s \le t\} \cup \{-\infty\})$ and $r^t := \min(\{s \in S : s \ge t\} \cup \{\infty\})$. These are the closest bus stops in S to the left and to the right of t; if there is no such stop, we set this variable to $-\infty$ or ∞, respectively. Let $C_t := \{i \in N : \ell^t < \ell_i < r^t \text{ or } \ell^t < r_i < r^t\}$, i.e., C_t is the set of agents that have at least one terminal strictly between ℓ^t and r^t. By design of Algorithm 1, we have

$$|C_t| \le \frac{2n}{b} - 1. \tag{4}$$

We claim that for every $i \in M$ there exists $t \in T$ with $i \in C_t$. Indeed, since $c_i(T) < c_i(S)$, we know that the cost of i with respect to T comes from a route in which they take the bus, i.e., there exist stops $t_1, t_2 \in T$ such that i walks from ℓ_i to t_1, takes the bus to t_2, and then walks to r_i. If $i \notin C_{t_1} \cup C_{t_2}$, then there exist $s_1, s_2 \in S$ such that s_1 is between ℓ_i and t_1, and s_2 is between t_2 and r_i. It is easy to see that $c_i(T) = c_i(\{t_1, t_2\}) \ge c_i(\{s_1, s_2\}) \ge c_i(S)$, contradicting the cost improvement of i according to T. Hence, it follows that $i \in C_{t_1} \cup C_{t_2}$.

We conclude that

$$|M| \le \left|\bigcup_{t\in T} C_t\right| \le \sum_{t\in T} |C_t| \overset{\text{Eq. (4)}}{\le} |T| \left(\frac{2n}{b} - 1\right) < |M|.$$

This is a contradiction, and hence, S is in the 2-core.

To prove that the bound is tight, let $0 < \epsilon \le 1$. We define an instance that is parameterized by an integer k. First, observe that $\frac{k+1}{2k}(2 - \epsilon)$ converges to $1 - \frac{\epsilon}{2}$ as k tends to infinity. Hence, we can choose k large enough so that $\frac{k+1}{2k}(2 - \epsilon) \le 1 - \frac{\epsilon}{4}$. Moreover, fix an integer $x > \frac{4}{\epsilon}$. We define an instance with $b = 2k$ and k groups of agents $N^1, \ldots, N^k$ of size x each, so $n = kx$. We want the agents' left and right terminals to be separated by a sufficiently long part of the path with no agent terminals. To this end, for each $j \in [k]$, we let $\ell^j = j$, $r^j = 2k + (2j-1)k$, and set

- $\ell_i := \ell^j$ for each $i \in N^j$;
- $r_i := r^j$ for $x-1$ agents in N^j and $r_i = r^j + k$ for the remaining agent in N^j.

An illustration is given in Figure 2. For $V = \cup_{i \in N}\{\ell_i, r_i\}$, the algorithm outputs the $2k$ bus stops in $S = [k] \cup \{2k + 2jk : j \in [k]\}$.

Now, consider the coalition M consisting of all agents except those whose right terminals are at $r^j + k$, $j \in [k]$. We have

$$|M| = n - k = k(x-1) > k\left(x - \frac{\epsilon}{4}x\right) = \left(1 - \frac{\epsilon}{4}\right)kx$$
$$\geq \frac{k+1}{2k}\,(2-\epsilon)\,kx = (k+1)\,(2-\epsilon)\,\frac{n}{b}.$$

Here, the first inequality holds by our choice of x, and the second inequality holds by our choice of k. Consider the set of bus stops

$$T = \{k\} \cup \{2k + (2j-1)k : j \in [k]\},$$

with $|T| = k+1$. To show that M is a blocking coalition for the $(2-\epsilon)$-core, it remains to argue that for every agent $i \in M$ it holds that $c_i(T) < c_i(S)$. To see this, fix a $j \in [k]$ and $i \in M \cap N^j$. Then, given solution T, i can walk to k and take the bus all the way to r_i. However, in S, i can board the bus at ℓ_i, but then she needs to walk at least k to her destination. Since the overall distance she travels in the latter case is at least as large as in the former case, and involves strictly more walking (k instead of $k-j$), agent i prefers T to S. □

We note that α-BSP can be shown to be a special case of committee selection with monotonic preferences and uniform costs. Hence, the results of Jiang et al. [22] (Theorem 1) imply that the 16-core of α-BSP is always non-empty. However, Theorem 4.5 offers a much better approximation guarantee.

It remains an open problem how to construct solutions in the core; in fact, we do not even know if the core is always non-empty. This seems to be a challenging question. For instance, by Proposition 2.5, the cost-minimal solution does not even provide JR. Similarly, the solution that selects the most popular terminal points may not provide JR (an example is given in the full version [6]).

Finally, we observe that solutions that provide strong JR (and therefore lie in the strong core) need not exist.

Proposition 4.6. *For every $\alpha \in [0,1)$ there exists an instance of α-BSP such that no feasible solution provides strong JR.*

Proof. Consider an instance $\mathcal{I} = \langle N, V, b, (\theta_i)_{i \in N}\rangle$ with agent set $N = [8]$ and budget $b = 4$. We set $V = [16]$, and for each $i \in N$ we set $\ell_i = 2i - 1$ and $r_i = 2i$. Hence, we have an instance with 8 agents, where all agents have different terminals.

Then, any feasible solution S has an empty intersection with the set of terminals of at least 4 agents. Hence, there exists a set $M \subseteq N$ with $|M| \geq 4$ such that $c_i(S) = 1$ for all $i \in M$, i.e., no agent in M can do any better than to walk. Note that $|M| \geq \frac{2n}{b}$. Hence, the agents in M are entitled to two bus stops.

Now, let $i \in M$ and consider $T = \{\ell_i, r_i\}$. Then $c_i(T) = \alpha < 1 = c_i(S)$, and for $j \in M \setminus \{i\}$ it holds that $c_j(T) = 1 = c_j(S)$. Hence, T is strictly better for agent i and at least as good for all other members of M. Therefore, S does not provide strong JR. □

4.2 Computation of Outcomes in the Core on Synthetic Data

While it remains open whether the core of α-BSP is always non-empty, our simulations indicate that Algorithm 1 can frequently find outcomes in the core. This is despite the fact that, according to Proposition 4.4 and Theorem 4.5, in the worst case this algorithm may fail to find JR outcomes for $\alpha \in (0,1)$ and outcomes in the β-core for $\beta < 2$. All computations were performed using an Apple M2 CPU with 24 GB of RAM.

4.2.1 Experimental Setup. For our experiments, we consider the following parameter ranges:

- Number of agents $n \in \{5, \ldots, 25\}$.
- Number of bus stops $m \in \{5, \ldots, 15\}$.
- Budget $b \in \{3, \ldots, m-1\}$, i.e., we consider all budgets that facilitate taking the bus at all and do not enable building all possible bus stops.
- Cost parameter $\alpha \in \{0, 0.1, \ldots, 0.9\}$.

For each combination, we generated 400 random instances.

Generating Random Instances. Our set of bus stops is a subset of the integers between 1 and 100. From this range, we choose m potential locations for bus stops uniformly at random. Moreover, we determine the types of agents by independently selecting two terminals from the set of possible stops uniformly at random.

Algorithmic Benchmark. We measure the frequency of instances on which Algorithm 1 computes solutions that provide JR/are in the core. For comparison, we benchmark our algorithm against the naive algorithm, which selects the stops as if the agents were distributed uniformly on the line. More precisely, if V is the set of possible stops, the naive algorithm ignores the actual agents' types and assumes instead that for each point $p \in \{1, \ldots, 100\}$, $n/50$ agents have one of their endpoints at p and report a point in $\arg\min d(p, V)$ as the respective terminal; it then runs Algorithm 1 under this assumption on the agents' types. In other words, Algorithm 1 can be seen as a weighted version of the naive algorithm, which takes actual user demands into account. In the full version of the paper, we consider a second benchmark algorithm, which is based on the idea of selecting the most demanded bus stops.

Verification of Fair Solutions. An integral part of performing simulations is an efficient algorithm for verifying whether a given solution provides JR or is in the core. While JR can be checked by a simple polynomial-time algorithm (by checking if enough agents benefit from using each pair of stops), we set up an integer program to verify whether a solution is in the core. The theory for this part of the simulations is described in the full version of the paper [6].

4.2.2 Experimental Results. The primary goal of our experiments is to measure the performance of Algorithm 1 in terms of achieving outcomes in the core or providing JR beyond the guarantee of Theorem 4.2. Figure 3 shows an aggregated view of our results. We see that, for a fixed α, Algorithm 1 computes a solution in the core in more than 99.9% of the cases. Even for the parameter combinations with the highest frequency of fairness violations, the failure rate of Algorithm 1 with respect to computing solutions in the core does not exceed 3%. Figure 4 gives a glimpse at a more nuanced

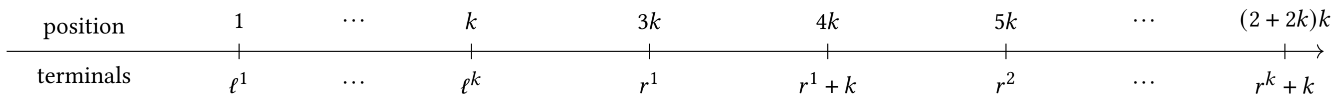

Figure 2: Algorithm 1 fails to output solutions in the $(2-\epsilon)$-core for $\epsilon \in (0,1]$.

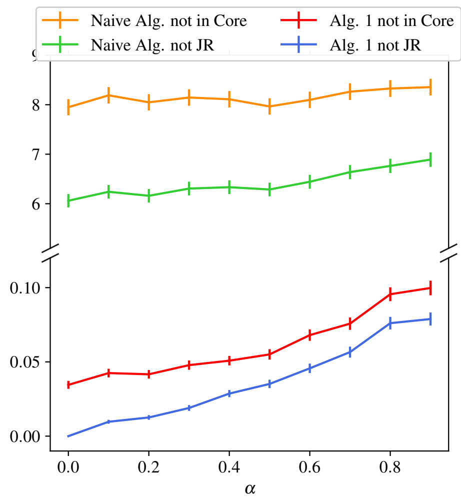

Figure 3: Aggregated frequency of fairness violations of the solutions computed by Algorithm 1 and the naive algorithm along with the standard error. The x-axis shows our range for the cost parameter α and the y-axis shows the percentage of the instances in which the desired property is not satisfied.

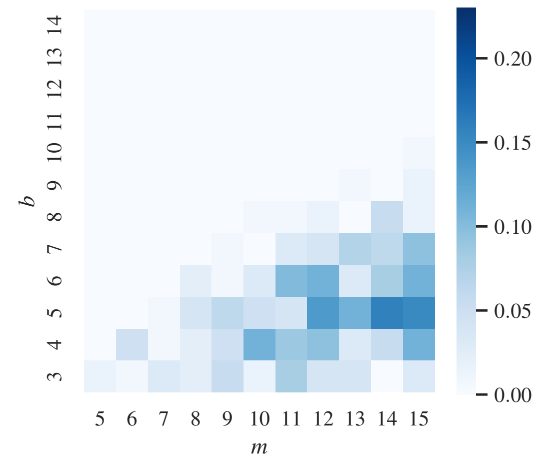

Figure 4: A heat map showing the frequency of core violations (as a percentage, i.e., 0.05 means 0.05%) of the solutions computed by Algorithm 1 for pairs of m and b in instances of 0-BSP. Each cell is averaged over all values of n.

distribution of the failures of computing solutions in the core. We defer a more detailed analysis to the full version of the paper. We remark that the upper left triangle of the figure cannot contain any core violations because then the budget exceeds the number of stops. One trend that can be observed from Figure 4 and that is confirmed in our analysis for any fixed number of agents is that the frequency of core violations is the highest for a comparatively high number of potential bus stops and a smaller budget.

Another interesting observation is that, for a significant fraction of the instances, if the solution computed by Algorithm 1 is not in the core, it fails JR as well; indeed, this fraction tends to increase as the cost parameter α increases. Moreover, Algorithm 1 performs much better than the naive algorithm. On average it performs up to 230.2 better for smaller values of α and still 83.7 times better for $\alpha = 0.9$. A detailed comparison of the performance of the two algorithms is provided in the full version of the paper [6]. Our interpretation of this finding is that, while placing bus stops uniformly is a simple and appealing approach, taking into account the actual user demands results in much fairer solutions.

5 CONCLUSION

We proposed a stylized model for planning a bus route. Our model can capture efficiency in terms of travel costs as well as fairness in terms of proportional representation of the agents. We have developed a dynamic program that minimizes the total travel cost for the agents. This approach turned out to be extremely versatile, in that it also applies to many variants of the base model. Concerning fairness, our main contribution is an algorithm that constructs JR solutions under the assumption that taking the bus has no cost. This algorithm is also a 2-approximation for the core.

Our work suggests two natural avenues for further research. First, it remains open how to compute JR solutions for instances of α-BSP where $\alpha \neq 0$; indeed, we do not even know if such solutions always exist. An even harder question is whether the core is always non-empty. This resembles the situation in approval-based committee voting, where the same question is famously open [see, e.g., 26]. Another promising direction is to extend our model to more complex topologies. For instance, one can consider the setting where the set of potential stops (and terminals) is a subset of $\mathbb{Q}^2$ or the vertex set of a (planar) graph. Further ahead, an important research challenge is to develop a richer framework to reason about fairness in more realistic models of public transport.

ACKNOWLEDGMENTS

Most of this work was done while Edith Elkind was at University of Oxford. Martin Bullinger was supported by the AI Programme of The Alan Turing Institute. Edith Elkind and Mohamad Latifian were supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grants EP/X038548/1 and EP/Y003624/1. We would like to thank the anonymous reviewers.

REFERENCES

[1] Pankaj K Agarwal, Shao-Heng Ko, Kamesh Munagala, and Erin Taylor. 2022. Locally fair partitioning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*. 4752–4759.

[2] Eleftherios Anastasiadis and Argyrios Deligkas. 2018. Heterogeneous facility location games. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 623–631.

[3] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. 2017. Justified representation in approval-based committee voting. *Social Choice and Welfare* 48, 2 (2017), 461–485.

[4] Haris Aziz, Barton E. Lee, Sean Morota Chu, and Jeremy Vollen. 2024. Proportionally representative clustering. In *Proceedings of the 20th International Conference on Web and Internet Economics (WINE)*.

[5] Haris Aziz, Evi Micha, and Nisarg Shah. 2023. Group fairness in peer review. In *Proceedings of the 36th Annual Conference on Neural Information Processing (NeurIPS)*.

[6] Martin Bullinger, Edith Elkind, and Mohamad Latifan. 2024. *Towards fair and efficient public transportation: A bus stop model*. Technical Report. https://arxiv.org/abs/2411.08784.

[7] Hau Chan, Aris Filos-Ratsikas, Bo Li, Minming Li, and Chenhao Wang. 2021. Mechanism design for facility location problems: A survey. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*. 4356–4365.

[8] Hau Chan and Chenhao Wang. 2023. Mechanism design for improving accessibility to public facilities. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2116–2124.

[9] Lee Chapman. 2007. Transport and climate change: a review. *Journal of transport geography* 15, 5 (2007), 354–367.

[10] Bhaskar Ray Chaudhury, Linyi Li, Mintong Kang, Bo Li, and Ruta Mehta. 2022. Fairness in federated learning via core-stability. In *Proceedings of the 35th Annual Conference on Neural Information Processing (NeurIPS)*.

[11] Bhaskar Ray Chaudhury, Aniket Murhekar, Zhuowen Yuan, Bo Li, Ruta Mehta, and Ariel D. Procaccia. 2024. Fair federated learning via the proportional veto core. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

[12] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. 2019. Proportionally fair clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 1032–1041.

[13] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. 2017. Fair public decision making. In *Proceedings of the 18th ACM Conference on Economics and Computation (ACM EC)*. 629–646.

[14] Joachim R. Daduna and José M. Pinto Paixão. 1995. Vehicle scheduling for public mass transit—an overview. In *Computer-Aided Transit Scheduling: Proceedings of the 6th International Workshop on Computer-Aided Scheduling of Public Transport*. 76–90.

[15] Guy Desaulniers and Mark D. Hickman. 2007. Public transit. *Handbooks in operations research and management science* 14 (2007), 69–127.

[16] Edith Elkind, Minming Li, and Houyu Zhou. 2022. Facility location with approval preferences: strategyproofness and fairness. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 391–399.

[17] European Parliament. 2019. CO2 emissions from cars: facts and figures. https://www.europarl.europa.eu/news/en/headlines/society/20190313STO31218/co2-emissions-from-cars-facts-and-figures-infographics. [Online; accessed 16-January-2024].

[18] Brandon Fain, Ashish Goel, and Kamesh Munagala. 2016. The core of the participatory budgeting problem. In *Proceedings of the 12th International Conference on Web and Internet Economics (WINE)*. 384–399.

[19] David J. Forkenbrock and Lisa A. Schweitzer. 1999. Environmental justice in transportation planning. *Journal of the American Planning Association* 65, 1 (1999), 96–112.

[20] Yuhei Fukui, Aleksandar Shurbevski, and Hiroshi Nagamochi. 2020. Group strategy-proof mechanisms for shuttle facility games. *Journal of Information Processing* 28 (2020), 976–986.

[21] Donald B Gillies. 1959. Solutions to general non-zero-sum games. *Contributions to the Theory of Games* 4, 40 (1959), 47–85.

[22] Zhihao Jiang, Kamesh Munagala, and Kangning Wang. 2020. Approximately stable committee selection. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. 463–472.

[23] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research* 195, 3 (2009), 761–769.

[24] Yusuf Hakan Kalayci, David Kempe, and Vikram Kher. 2024. Proportional representation in metric spaces and low-distortion committee selection. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*. 9815–9823.

[25] Soo Chen Kwan and Jamal Hisham Hashim. 2016. A review on co-benefits of mass public transportation in climate change mitigation. *Sustainable Cities and Society* 22 (2016), 11–18.

[26] Martin Lackner and Piotr Skowron. 2023. *Multi-Winner Voting with Approval Preferences*. Springer-Verlag.

[27] William Lampkin and P. D. Saalmans. 1967. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *Journal of the Operational Research Society* 18, 4 (1967), 375–397.

[28] Bo Li, Lijun Li, Ankang Sun, Chenhao Wang, and Yingfan Wang. 2021. Approximate group fairness for clustering. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 6381–6391.

[29] Piotr Matl, Richard F. Hartl, and Thibaut Vidal. 2018. Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science* 52, 2 (2018), 239–260.

[30] Evi Micha and Nisarg Shah. 2020. Proportionally fair clustering revisited. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*.

[31] Rafael H. M. Pereira, Tim Schwanen, and David Banister. 2017. Distributive justice and equity in transportation. *Transport reviews* 37, 2 (2017), 170–191.

[32] Dominik Peters and Piotr Skowron. 2020. Proportionality and the limits of welfarism. In *Proceedings of the 21st ACM Conference on Economics and Computation (ACM EC)*. 793–794.

[33] Grzegorz Pierczyński and Piotr Skowron. 2022. Core-stable committees under restricted domains. In *Proceedings of the 18th International Conference on Web and Internet Economics (WINE)*. 311–329.

[34] John Pucher. 1982. Discrimination in mass transit. *Journal of the American Planning Association* 48, 3 (1982), 315–326.

[35] John Rawls. 1971. *A Theory of Justice*. Harvard University Press.

[36] Amartya Sen. 2009. *The Idea of Justice*. Belknap Press of Harvard University Press.

[37] Paolo Serafino and Carmine Ventre. 2014. Heterogeneous facility location without money on the line. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*. 807–812.

[38] Lionel Adrian Silman, Zeev Barzily, and Ury Passy. 1974. Planning the route system for urban buses. *Computers & operations research* 1, 2 (1974), 201–211.

[39] Ben J. Waterson, Bipul Rajbhandari, and Nick B. Hounsell. 2003. Simulating the impacts of strong bus priority measures. *Journal of Transportation Engineering* 129, 6 (2003), 642–647.

[40] Anthony Wren and Jean-Marc Rousseau. 1995. Bus driver scheduling—an overview. In *Computer-Aided Transit Scheduling: Proceedings of the 6th International Workshop on Computer-Aided Scheduling of Public Transport*. 173–187.

[41] Houyu Zhou, Minming Li, and Hau Chan. 2022. Strategyproof mechanisms for group-fair facility location problems. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*. 613–619.