Pipeline Interventions

Eshwar Ram Arunachaleswaran
University of Pennsylvania, Philadelphia, PA, USA
eshwar@seas.upenn.edu

Sampath Kannan
University of Pennsylvania, Philadelphia, PA, USA
kannan@cis.upenn.edu

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA
aaroth@cis.upenn.edu

Juba Ziani
University of Pennsylvania, Philadelphia, PA, USA
jziani@seas.upenn.edu

Abstract

We introduce the *pipeline intervention* problem, defined by a layered directed acyclic graph and a set of stochastic matrices governing transitions between successive layers. The graph is a stylized model for how people from different populations are presented opportunities, eventually leading to some reward. In our model, individuals are born into an initial position (i.e. some node in the first layer of the graph) according to a fixed probability distribution, and then stochastically progress through the graph according to the transition matrices, until they reach a node in the final layer of the graph; each node in the final layer has a *reward* associated with it. The pipeline intervention problem asks how to best make costly changes to the transition matrices governing people's stochastic transitions through the graph, subject to a budget constraint. We consider two objectives: social welfare maximization, and a fairness-motivated maximin objective that seeks to maximize the value to the population (starting node) with the *least* expected value. We consider two variants of the maximin objective that turn out to be distinct, depending on whether we demand a deterministic solution or allow randomization. For each objective, we give an efficient approximation algorithm (an additive FPTAS) for constant width networks. We also tightly characterize the "price of fairness" in our setting: the ratio between the highest achievable social welfare and the social welfare consistent with a maximin optimal solution. Finally we show that for polynomial width networks, even approximating the maximin objective to any constant factor is NP hard, even for networks with constant depth. This shows that the restriction on the width in our positive results is essential.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Interventions for fairness, fairness in navigating life paths, social welfare, maximin welfare, budget-constrained optimization, hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.8

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.06592` [1].

Funding *Eshwar Ram Arunachaleswaran*: Supported in part by NSF grant CCF-1763307.
Sampath Kannan: Supported in part by NSF grants CCF-1763307 and CCF-1733794.
Aaron Roth: Supported in part by NSF grant CCF-1763307 and a grant from the Simons Foundation.
Juba Ziani: Supported in part by NSF grant CCF-1763307 and by the Warren Center.

12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 8; pp. 8:1–8:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Inequality can be difficult to correct by the time it manifests itself in consequential domains. For example, faculty in computer science departments are disproportionately male (Way et al. [21]), and although the reasons for this are varied and complex, it seems difficult to correct *only* by intervening in the process of faculty hiring (although the solution likely involves some intervention at this stage). The problem is that interventions at the final stage of a long pipeline may not be enough (or the best way) to address iniquities that compound starting from earlier stages in the pipeline such as graduate school, college, high school, enrichment programs, all the way back to birth circumstances. Because each stage of, for example, employment pipelines feeds into the next, interventions that are isolated to any one stage can have difficulty controlling effects on final outcomes – and although in practice it is difficult to fully understand such a system, we would ideally like to design proposed interventions at a system-wide level, rather than myopically.

Thus motivated, we study an optimization problem within a stylized (and highly simplified) model of such a pipeline. Our model is a layered directed acyclic graph. The vertices in the first layer represent a coarse partitioning of possible birth circumstances into a small number of types – each vertex representing one of these types. There is a probability vector over these vertices and individuals are "born" into some vertex with these probabilities. The graph represents a Markov process that determines how individuals progress through the pipeline. From every vertex there is a stochastic transition matrix specifying the probability that an individual will progress to each vertex in the next layer of the pipeline. We might imagine, for example, that the proportion of children that enroll in each of several elementary schools (the second layer of such a pipeline) varies according to the neighborhood that they are raised in (the first layer). The proportion of children that then go on to enroll in each of several high schools may then vary according to the elementary school they attend, and so on. Finally, vertices at the last layer of the pipeline are associated with payoffs. One may then calculate the expected payoff of an individual as a function of their initial position. These payoffs may vary widely depending on this position.

We are concerned with the problem of how best to invest limited resources so as to *modify* the transition matrices governing different layers of this pipeline to achieve some goal. In the main body of the paper, we focus on a stylized model where the costs of modifying transition matrices are linear, for simplicity of exposition; we extend our results to more complex and realistic cost functions in the Appendix. We consider two goals: the first is simply maximizing social welfare – the expected payoff for an individual chosen according to the given probability vector for the first layer. Although this is a natural objective, it can easily lead to solutions that are "unfair" in the sense that they will prioritize investments that lead to improvements for majority populations over minority populations, simply because majority populations, by their sheer numbers, contribute more to social welfare. The second goal we study is therefore to maximize the *minimum* expected payoff of individuals, where the minimum is taken over all of the initial positions, i.e., layer 1 vertices. This "maximin" objective is a standard fairness-motivated objective in allocation problems [See Barman and Krishnamurthy [2], Procaccia and Wang [19], Budish [5]]. In fact, we study two different variants of this objective, that can be distinguished by the timing with which one wants to evaluate fairness. The *ex-ante* maximin objective asks for a *distribution* over budget-feasible modifications of the transition matrices, that maximize the minimum expected payoff over all initial positions. The *ex-post* maximin objective asks for a single (i.e. deterministic) budget-feasible modification to the transition matrices. Because the problem we study is non-convex, these two goals are distinct – which is preferred depends on *when* one wants to evaluate the fairness of a solution: before or after the randomization.

1.1 Overview of Our Results

Briefly, our main contributions are the following:

1. We define and formalize the *pipeline intervention problem* with the social welfare, ex-ante maximin, and ex-post maximin objectives. We also prove a separation between the ex-post and ex-ante maximin solutions.
2. We give an additive fully polynomial-time approximation scheme (FPTAS) for both the social welfare and ex-post maximin objectives for networks of constant width (but arbitrarily long depth).
3. We give an efficient reduction from the ex-ante maximin objective problem to the ex-post maximin objective problem via equilibrium computation in two-player zero-sum games. Combined with our results from 2, this yields an additive FPTAS for the ex-ante maximin objective problem for constant width networks as well.
4. We define and prove tight bounds on the "price of fairness", which compares the optimal social welfare that can be achieved with a given budget to the social welfare of ex-post maximin optimal solutions.
5. Finally, we show that the pipeline intervention problem is NP hard even to approximate in the general case when the width w is not bounded – and hence that our efficient approximation algorithms cannot be extended to the general case (or even the case of constant depth, polynomial width networks).

1.2 Related Work

There is an enormous literature in "algorithmic fairness" that has emerged over the last several years, that we cannot exhaustively summarize here – but see Chouldechova and Roth [6] for a recent survey. Most of this literature is focused on the myopic effects of a single intervention, but what is more conceptually related to our paper is work focusing on the longer-term effects of algorithmic interventions.

Dwork and Ilvento [9] and Bower et. al. [4] study the effects of imposing fairness constraints on machine learning algorithms that might be composed together in various ways to reach an eventual outcome. They show that generally fairness constraints imposed on constituent algorithms in a pipeline or other composition do not guarantee that the same fairness constraints will hold on the entire mechanism as a whole. (They also study conditions under which fairness guarantees *are* well behaved under composition). Two recent papers (Liu et al. [15], Mouzannar et al. [17]) study parametric models by which classification interventions in an earlier stage can have effects on the data distribution at later stages, and show that for many commonly studied fairness constraints, their effects can either be positive or negative in the long term, depending on the functional form of the relationship between classification decisions and changes in the agent type distribution.

There is also a substantial body of work studying game theoretic models for how interventions affect "fairness" goals. This work dates back to Coate and Loury [7], Foster and Vohra [10] in the economics literature, who propose game theoretic models to rationalize how unequal outcomes might emerge despite two populations being symmetrically situated. More recently, in the computer science literature, several papers consider more complicated models that are similar in spirit to Coate and Loury [7], Foster and Vohra[10]. Hu and Chen [12] propose a two-stage model of a labor market with a "temporary" (i.e. internship) and "permanent" stage, and study the equilibrium effects of imposing a fairness constraint on the temporary stage. Liu et al. [16] consider a model of the labor market with higher dimensional signals, and study equilibrium effects of subsidy interventions which can lessen the cost of exerting effort. Kannan et al. [14] study the effects of admissions policies on a two-stage model of education and employment, in which a downstream employer makes rational decisions. Jung et al. [13] study a model of criminal justice in which crime rates are responsive to the classifiers used to determine criminal guilt, and study which fairness constraints are consistent with the goal of minimizing crime.

2 Model

The *pipeline intervention* problem is defined by a layered directed acyclic graph $G = (V, E)$, where V is the set of vertices (or nodes), and E is the set of edges. The vertices are partitioned into k *layers* $L_1, L_2, \cdots L_k$, each consisting of w vertices. We say that w is the *width* of the graph. For every $t \in [k-1]$, there is a directed edge from every $u \in L_t$ to every $v \in L_{t+1}$; the graph contains no other edge. In turn, every path from layer L_1 to layer L_k must go through exactly one vertex in each layer $L_2, \ldots, L_{k-1}$ in this order. Intuitively, such a layered graph represents a pipeline, in which individuals start at initial positions in layer 1, and transition through the graph to final positions in layer k, stochastically according to transition matrices which we define next. This layered model can be used to abstractly represent real-life pipelines; such a pipeline, that has received attention in previous work (e.g. Kannan et al. [14]), is the education and job market one. Nodes in the initial layer represent a coarse partitioning of the population based on family income levels and educational background. The second layer could represent pre-K experience. For example, one could have 3 nodes in the second layer representing no pre-K, Headstart, and private pre-K. See for example Barnum [3] for a general discussion of as well as pointers to recent studies on the efficacy of Headstart programs. At the next level or two, nodes can represent different qualities of K-12 schools, based on a coarse partitioning of their performance under one of several widely-available metrics, such as the ones provided by U.S. News [20], Niche [18].

The layer after that could be a coarse partitioning where nodes represent, for example, no college, technical or vocational school, and 2 and 4-year colleges coarsely grouped together based on perceived quality according to one of several college rankings. A subsequent layer could encode the details of a student's performance in college, such as their major and GPA, again under a coarse bucketing. The last layer, with numerical rewards could represent different types of employment with rewards determined by starting salaries and prospects for advancement.

In a more accurate model, we might perhaps condition the probability of transition from node u in layer i to node v in layer $i+1$ on the entire path taken by an individual leading up to node u. However, for mathematical tractability, we make the simplifying assumption that the process is Markovian, and this transition probability from u is independent of prior history.

Let $\mathcal{M}$ be the set of left stochastic matrices in $\mathbb{R}^{w \times w}$: i.e., $M \in \mathcal{M}$ if and only if for all $j \in [w]$, $\sum_{i \in [w]} M(i,j) = 1$, and for all $(i,j) \in [w]^2$, $M(i,j) \geq 0$. Let $\mathcal{D} \triangleq \{x \in [0,1]^w : \sum_{k=1}^{w} x(k) = 1\}$ be the set of probability distributions over $[w]$. An instance of the pipeline intervention problem is defined by three elements:

1. A set of initial transition matrices $M_t^0 \in \mathcal{M}$ between layers L_t and L_{t+1}, for all $t \in [k-1]$, such that for all $u \in L_t$, $v \in L_{t+1}$, $M_t^0(v,u)$ denotes the probability of transitioning from node u to node v. Note that we will multiply any input distribution to the right of any transition matrix we use in the paper.

2. An input distribution D_1 over the vertices in layer 1, where $D_1(u)$ denotes the fraction of the population that starts at u in L_1 as their initial position. Without loss of generality we assume $D_1(u) > 0$ for all initial positions $u \in L_1$.
3. Finally, a reward $R(v) \geq 0$ corresponding to each vertex $v \in L_k$ in the final layer. We let $R = (v)_{v \in L_k}^\top$ denote the vector of all rewards on layer k. We assume without loss of generality that the rewards on any two vertices in the final layer are distinct: for all $v, v' \in L_k$, $R(v) \neq R(v')$. We can also assume without loss of generality (up to renaming) that $R(1) > \ldots > R(w)$.

In our model, each vertex u in the *starting layer* L_1 represents the initial position of some population; abusing notation, we refer to this population also as u. An individual in population u transitions to a node in layer L_2, then a node in layer L_3, up until they reach a node v in *destination layer* L_k, and obtains a reward of $R(v)$, with probability given by the transition matrices M_1^0 to M_{k-1}^0. The expected reward of an individual from population u is therefore given by $R^\top M_{k-1} \cdot \ldots \cdot M_1 e_u$, where e_u represents the w-dimensional standard basis vector corresponding to index u. The aim of the pipeline intervention problem is to *modify* the transition matrices between pairs of adjacent layers so as to improve these expected rewards in some way (we study several objectives) given a finite resource constraint.

We will take the point of view of a centralized designer, who can invest money into modifying the transition matrices between layers. We assume some edges can be modified, while some edges cannot; the edges that can be modified are called *malleable*, and the edges that cannot be modified are called *non-malleable*. We denote the set of malleable edges between layers t and $t+1$ by E^t_{mal} and the set of non-malleable edges by $\overline{E^t_{mal}}$ its complement. Further, we assume that modifying these transitions matrices comes at a cost, and that on a given layer t, the cost of transforming M_t^0 to some alternative $M_t \in \mathcal{M}$ is given by:

$$c(M_t, M_t^0) \triangleq \sum_{(i,j) \in [w]^2} \left| M_t(i,j) - M_t^0(i,j) \right|.$$

▶ Remark 1. A critique of such cost functions is that they may not be rich enough to model the cost of improving transitions and opportunities between different stages of, say, the education pipeline.

To address this, we note that while we focus on these simple cost functions in the main body of the paper for simplicity of exposition, our algorithmic results (of Sections 4, 5 and 6) extend to more general and possibly more realistic cost functions – so long as they are convex and increase at least linearly as the distance $\sum_{(i,j) \in [w]^2} \left| M_t(i,j) - M_t^0(i,j) \right|$ between modified transition matrix M_t and initial transition matrix M_t^0 increases. We discuss this extension in more detail in the full version [1]

This extension allows us to model more realistic situations such as those where the cost functions are not linear, but also those where different edges have different costs – as motivated by the fact that real-life interventions often become more expensive the later they happen.

The designer has a total budget of B, and can select target transition matrices $(M_1, \ldots, M_{k-1})$ so long as the cost of modifying the initial transition matrices to his targets does not exceed his budget, and only malleable edges have been modified. That is, he must select target transition matrices subject to the constraint:

$$\sum_{t=1}^{k-1} c(M_t, M_t^0) \leq B.$$

We let

$$\mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right) = \left\{ (M_1, \ldots, M_{k-1}) : \sum_{t=1}^{k-1} c\left(M_t, M_t^0\right) \leq B,\ M_t \in \mathcal{M}\ \forall t,\ M_t(i,j) = M_t^0(i,j)\ \forall (i,j) \in \overline{E_t^{mal}} \right\}$$

be the set of feasible sets of transition matrices, given initial matrices $M_1^0, \ldots, M_{k-1}^0$ and budget B. We will consider several objectives that we may wish to optimize. The first is simply to maximize the overall social welfare (i.e. the expected reward of an individual chosen according to D_1), which is given by

$$W\left(M_1, \ldots, M_{k-1}\right) \triangleq R^\top M_{k-1} \ldots M_1 D_1.$$

The second objective aims to compute a "fair" outcome in the sense that it evaluates a solution according to the expected payoff of the worst-off members of society (here interpreted as individuals starting at the pessimal initial position), rather than according to the average. This is the classic maximin objective. It turns out that there are two distinct variants of this problem, depending on whether one wishes to allow randomized solutions (i.e. distributions over matrices) or not. We will elaborate on this distinction in the next section, but in the deterministic variant we wish to optimize

$$\min_{j \in [w]} R^\top M_{k-1} \ldots M_1 e_j,$$

where $e_j \in \mathbb{R}^w$ is the unit vector with $e_j(j) = 1$, and $e_j(i) = 0$ for all $i \neq j$.

▶ Remark 2. We have assumed that each layer has *exactly* w vertices. In fact, all of our results generalize to the case in which each layer has $\leq w$ vertices.

2.1 Optimization Problems of Interest

In this paper, we will provide algorithms to solve the following three optimization problems. We note at the outset that these optimization problems are non-convex, due to the fact that our objective values are not convex for $k \geq 2$. Hence we should not expect efficient algorithms in the fully general setting; we will give efficient algorithms for networks of constant width w (i.e. algorithms whose running time is polynomial in the depth of the network k), and show that outside of this class, the problem is NP hard even to approximate.

- **Social welfare maximization** The first optimization problem we aim to solve is that of maximizing the social welfare of our network, under our budget constraint:

$$\begin{aligned} OPT_{SW} = \max_{M_1, \ldots, M_{k-1}} \quad & R^\top M_{k-1} \ldots M_1 D_1^0 \\ \text{s.t.} \quad & (M_1, \ldots, M_{k-1}) \in \mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right) \end{aligned} \tag{1}$$

- **Ex-post maximin problem** The second optimization problem aims to maximize the minimum expected reward that a population can obtain, where the minimum is taken over all initial positions:

$$\begin{aligned} OPT_{MM} = \max_{M_1, \ldots, M_{k-1}} \quad & \min_{j \in [w]} R^\top M_{k-1} \ldots M_1 e_j \\ \text{s.t.} \quad & (M_1, \ldots, M_{k-1}) \in \mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right) \end{aligned} \tag{2}$$

- **Ex-ante maximin problem** The third optimization problem has the same objective as Program 2, but allows randomization over sets of transition matrices that satisfy the budget constraint. Note that the budget constraint must be satisfied *ex-post*, for *any realization* of the set of transition matrices. To define this optimization problem, we let $\Delta\mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right)$ the set of probability distributions with support $\mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right)$. The optimization program is given by:

$$OPT_{RMM} = \max_{\Delta M} \quad \min_{j \in [w]} R^\top \mathbb{E}_{M \sim \Delta M}\left[M_{k-1} \ldots M_1\right] e_j$$
$$\text{s.t.} \quad \Delta M \in \Delta\mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right), \tag{3}$$

where the expectation is taken over the randomness of distribution ΔM. Note that where Program 3 can be viewed as optimizing an *ex-ante* notion of fairness, in which we are evaluated on the minimum expected value of individuals starting at any initial position, *before the coins of* ΔM *are flipped.* In contrast, Program 2 evaluates the minimum expected value of individuals starting at any initial position for an already established set of transition matrices.

▶ Remark 3. Programs (1), (2) and (3) all have solutions, and as such the use of maxima instead of suprema is well defined. To see this, first note that the feasible sets are non-empty since $\left(M_1^0, \ldots, M_{k-1}^0\right) \in \mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right)$ for all $B \geq 0$. For Program (1), the existence of a maximum is an immediate consequence of the fact that the objective function is continuous in $(M_1, \ldots, M_{k-1})$ and $\mathcal{F}$ and $\mathcal{M}$ are compact sets. For Program (2), note that no solution can have $R^\top M_{k-1} \ldots M_1 e_j \geq \|R\|_\infty$ for any j, as $M_{k-1} \ldots M_1 e_j$ is a probability distribution. Hence, we can rewrite the program as

$$\max_{v, M_1, \ldots, M_{k-1}} \quad v$$
$$\text{s.t.} \quad 0 \leq v \leq \|R\|_\infty,$$
$$R^\top M_{k-1} \ldots M_1 e_j \geq v \; \forall j \in [w],$$
$$(M_1, \ldots, M_{k-1}) \in \mathcal{F}\left(B, M_1^0, \ldots, M_{k-1}^0\right).$$

This is an optimization problem with a continuous objective function over a compact set, so it admits a solution. A similar argument follows for Program (3).

3 Algorithmic Preliminaries

Our paper uses a dynamic programming approach for solving programs (1) and (2). (Our solution to program (3) is a game-theoretic reduction to our solution to program (2)). Our algorithms will search over possible input distributions in $\mathcal{D}$ starting from layer L_t for all $t \in \{2, \ldots, k-2\}$, and over possible ways of splitting the total budget B and allocating budget B_t to the transition from layer L_t to layer L_{t+1}, for all $t \in [k-1]$. To do so, we will need to discretize both the budget space $[0, B]$ and the probability space $\mathcal{D}$.

3.1 Cost of Discretizing the Budget

To discretize the budget space, we define $\mathcal{B}(\varepsilon) = \{k\varepsilon, \; \forall k \in \mathcal{N}\}$ to be the set of numbers on the real line that are multiples of ε. We consider the following discretized version of Program 1:

$$
\begin{aligned}
OPT^{\varepsilon}_{SW} = \max_{M_1,\dots,M_{k-1}} \quad & R^\top M_{k-1}\dots M_1 D_1 \\
\text{s.t.} \quad & c(M_t, M^0_t) \leq B_t \ \forall t \in [k-1] \\
& B_t \in \mathcal{B}(\varepsilon) \ \forall t \in [k-1], \ \sum_{t=1}^{k-1} B_t \leq B \\
& M_t(i,j) = M^0_t(i,j) \ \forall (i,j) \in \overline{E^{mal}_t}, \ M_t \in \mathcal{M} \ \forall t
\end{aligned} \tag{4}
$$

Program 2 needs an analogous modification. (We do not need to explicitly consider Program (3), since our solution for this one will be a reduction to our solution to Program (2).)

We show that this discretization does not affect the optimal value of our problems by much:

▷ Claim 4. There exists a feasible solution $\left(M^{\varepsilon}_1, \dots, M^{\varepsilon}_{k-1}\right)$ to Program (4) (resp. for the analogous modification of Program) with objective value at least $OPT_{SW} - (k-1)\varepsilon \left\|R\right\|_\infty$ (resp. $OPT_{MM} - (k-1)\varepsilon \left\|R\right\|_\infty$).

We provide a brief proof sketch below, and defer the full proof to the full version [1].

Proof Sketch. We prove this result by constructing transition matrices M^{ε}_t that use roughly ε budget less than M^*_t. We show that we can do so so as to only lose welfare of the order of ϵ in each of the $k-1$ layer transitions we consider, and that this loss composes additively. ◁

▶ **Definition 5.** *Let $K \subseteq [0,1]^w$. We call a subset S of K an ε-net for K with respect to the ℓ_1-norm if and only if for every $D \in K$, there exists $D' \in S$ such that*

$$\|D - D'\|_1 \leq \varepsilon.$$

▷ Claim 6 (ε-nets in ℓ_1-distance for $\mathcal{D}$). Take $\varepsilon > 0$. There exists an ε-net $\mathcal{D}(\varepsilon)$ of $\mathcal{D}$ with respect to the ℓ_1-norm that has size $\left(\frac{1}{\varepsilon}\right)^w$.

This is a standard proof, which can be found in the full version of this paper [1].

4 Social Welfare Maximization

We want to solve the following optimization problem:

$$
\begin{aligned}
\max_{M_1,\dots,M_{k-1}} \quad & R^\top M_{k-1}\dots M_1 D_1 \\
\text{s.t.} \quad & \sum_{t=1}^{k-1} c\left(M_t, M^0_t\right) \leq B, \\
& M_t \in \mathcal{M} \ \forall t \in [k-1],
\end{aligned} \tag{5}
$$

4.1 A Dynamic Programming Algorithm for Social Welfare Maximization

In this section, we describe a dynamic programming algorithm for approximately solving the problem above on long skinny networks. The algorithm will run in polynomial time when the width w of the network is small; its running time is polynomial in the depth k of the network, but exponential in the width w. The formal description can be found in the full version. Our algorithm works backwards, starting from the final transition matrix from layer L_{k-1} to L_k. It builds up the solutions to sub-problems parameterized by three parameters – a layer t, a starting distribution over the vertices in layer t, and a budget $B_{\geq t}$ that can be used at layers $\geq t$. For each sub-problem, it computes an approximately welfare-optimal solution. Once all of these sub-problems have been solved, the optimal solution to the original problem can be read off from the "sub-problem" in which $t = 1$, the starting distribution is the distribution on initial positions, and $B_{\geq 1} = B$. Here is the informal description of the algorithm:

1. For t going backwards from $k-1$ to 1, the algorithm does the following exploration over budget splits and probability distributions $D_t, D_{t+1} \in \mathcal{D}(\varepsilon)$ (an ε-net for the w-dimensional simplex in ℓ_1 norm) on L_t:
   a. The algorithm explores all discretized splits of a budget $B_{\geq t}$ to be used for layers t to $k-1$ into a budget B_t to expend on layer t and a budget $B_{\geq t+1}$ to expend on the remaining layers $t+1$ to $k-1$, as well as all choices of target output probability distribution $D_{t+1} \in \mathcal{D}(\varepsilon)$ on layer L_{t+1} and the starting probability distribution $D_t \in \mathcal{D}(\varepsilon)$. Informally, we can think of these "target" and "initial" probability distributions as guesses for what the distribution on vertices in layer $t+1$ and layer t look like in the optimal solution. Recall that for each D_{t+1} and $B_{\geq t+1}$, our algorithm has already computed a near-optimal solution for a smaller sub-problem, which we will utilize in the next step.
   b. The algorithm then finds a transition matrix from L_t to L_k that maximizes welfare when the starting distribution on layer t is D_t and the remaining transition matrices are fixed as in the solution to the corresponding sub-problem. Although the overall welfare-maximization problem is non-convex, this sub-problem can be solved as a linear program (Program 6) because all transition matrices except for one have been fixed as the solution to our sub-problem.
   c. Finally, the algorithm picks and stores the recovered transition matrices from layer L_t to L_k that yield the highest reward, among all the transition matrices recovered from step 1b.

We remark that while (for notational simplicity) our algorithm is written as if all layers have size exactly w, it can easily be extended to the case in which all layers have size *at most* w.

We briefly note why Program (6) is a linear program. The objective is linear because only the matrix M_t represents variables. Thus we simply need to verify that the constraint on the cost is linear.

▶ **Definition 7.** *We say that a transition matrix $M_t \in \mathcal{M}$ is feasible with respect to a budget split $B_{\geq t}, B_{\geq t+1}$ if and only if*

$$c\left(M_t, M_t^0\right) \leq B_{\geq t+1} - B_{\geq t}.$$

and $M_t(i,j) = M_t^0(i,j)$ for every non-malleable edge (i,j).

Note that saying that M_t feasible with respect to $B_{\geq t}, B_{\geq t+1}$ is equivalent to saying that M_t is a feasible solution to Program (6) with parameters $B_{\geq t}, B_{\geq t+1}, D_t, D_{t+1}$ for any $D_t, D_{t+1} \in \mathcal{D}(\varepsilon)$. The constraint $c\left(M_t, M_t^0\right) \leq B_{\geq t+1} - B_{\geq t}$ can be equivalently replaced by $2w^2 + 1$ linear constraints. To do so, we introduce w^2 variables - $a_1, a_2, \cdots a_{w^2}$. The constraint can then be rewritten in the form $\sum_{i=1}^{w^2} |f_i| \leq B_{\geq t+1} - B_{\geq t}$, where each f_i is a linear combination of the variables. We can thus express the budget constraint of Program 6 by the following set of linear constraints:

Algorithm 1 Dynamic Program for (Approximate) Social Welfare Maximization.

Input: Input distribution D_1, reward vector R, initial transition matrices $M_1^0, \dots, M_{k-1}^0$, budget B, discretization parameter ε.
Output: Transition $M(B^\varepsilon, D_1^0)$ from L_1 to L_k.
Initialization: Let $B_{\geq k} = 0$, $M(B_{\geq k}, D_k) = I$, $B^\varepsilon = \max\{x \in \mathcal{B}(\varepsilon) :\ x \leq B\}$.
for *layer* $t = k-1, \dots, 1$ **do**
  for *all distributions* $D_t \in \mathcal{D}(\varepsilon)$ *if* $t \neq 1$ *(*$D_t = D_1^0$ *if* $t = 1$*) and budgets* $B_{\geq t} \in \mathcal{B}(\varepsilon)$ *with* $B_{\geq t} \leq B$ **do**
    for *all distributions* $D_{t+1} \in \mathcal{D}(\varepsilon)$ *and budgets* $B_{\geq t+1} \leq B_{\geq t}$ *such that* $B_{\geq t+1} \in \mathcal{B}(\varepsilon)$ **do**
      Solve linear program

$$\begin{aligned} M_t(B_{\geq t}, B_{\geq t+1}, D_t, D_{t+1}) = \arg\max_{M_t} \quad & R^\top M(B_{\geq t+1}, D_{t+1}) M_t D_t \\ \text{s.t.} \quad & c\left(M_t, M_t^0\right) \leq B_{\geq t} - B_{\geq t+1}, \\ & M_t(i,j) = M_t^0(i,j)\ \forall (i,j) \in \overline{E_t^{mal}} \\ & M_t \in \mathcal{M} \end{aligned} \tag{6}$$

    end
    Pick $B_{\geq t+1}, D_{t+1}$ leading to the highest objective value in Program 6, and set $M(B_{\geq t}, D_t) = M(B_{\geq t+1}, D_{t+1}) M_t(B_{\geq t}, B_{\geq t+1}, D_t, D_{t+1})$.
  end
end
Return $M(B^\varepsilon, D_1)$.

1. $f_i \leq a_i \quad \forall i \in [w^2]$
2. $-f_i \leq a_i \quad \forall i \in [w^2]$
3. $\sum_{i=1}^{w^2} a_i \leq B_{\geq t+1} - B_{\geq t}$.

Thus, Program 6 can be written as a linear program with the number of constraints and variables being polynomial in w.

4.2 Running Time and Social Welfare Guarantees

We provide the running time and social welfare guarantees of Algorithm 8 below.

▶ **Theorem 8.** *Algorithm 1 instantiated with discretization parameter ε yields a solution achieving social welfare at least $OPT - 3(k-1)\varepsilon\|R\|_\infty$, and has running time $O\left(k\frac{B}{\varepsilon}\left(\frac{1}{\varepsilon}\right)^{w^2} f(w)\right)$, where $f(w)$ is any upper-bound on the running time for solving linear Program 6, which is always polynomial in w.*

This immediately yields the following corollary:

▶ **Corollary 9.** *Algorithm 1 with discretization parameter $\varepsilon' = \frac{\varepsilon}{3(k-1)}$ yields social welfare at least $OPT - \varepsilon\|R\|_\infty$, and has running time $O\left(k^2\frac{B}{\varepsilon}\left(\frac{k}{\varepsilon}\right)^{w^2} f(w)\right)$, where $f(w)$ is any upper-bound on the running time for solving linear Program 6, which is always polynomial in w.*

We observe that this running time is polynomial in k (the depth of the network) and $1/\varepsilon$ (the inverse additive error tolerance), but exponential in w (the width of the network). Hence our algorithm runs in polynomial time for the class of constant width networks.

▶ Remark 10. We note that our additive near-optimality guarantee can be translated into a multiplicative guarantee. In the case where *all edges are malleable*, this follows from noting that given budget B, $OPT \geq \frac{B}{2w}\|R\|_\infty$: this can be reached by investing the totality of the budget into transitioning every node in the second-to-last layer to the highest reward node in the last layer, with probability $\frac{B}{2w}$ for each such node. Taking $\varepsilon = \delta \cdot \frac{B}{6(k-1)w}$ for some constant $\delta < 1$ gives a multiplicative approximation to the optimal social welfare with approximation factor $1 - \delta$.

For the case in which non-malleable edges are allowed, a lower bound on OPT is given by $OPT \geq W_0$. Taking $\varepsilon = \delta \cdot \frac{W_0}{3(k-1)\|R\|_\infty}$ yields a multiplicative $1 - \delta$ approximation still.

Proof of Theorem 8

The proof of Theorem 8 relies on the following lemma, and its corollary:

▶ **Lemma 11.** *Let $M \in \mathbb{R}^{w \times w}$ be a left stochastic matrix, and let $D, D' \in \mathcal{D}$ be probability distributions.*

$$\|MD - MD'\|_1 \leq \|D - D'\|_1.$$

Proof. Note that

$$\begin{aligned}
\|M(D - D')\|_1 = \sum_{i=1}^{w} |(M(D - D'))\,(i)| &= \sum_{i=1}^{w} \left| \sum_{j=1}^{w} M(i,j)(D(j) - D'(j)) \right| \\
&\leq \sum_{i=1}^{w} \sum_{j=1}^{w} |M(i,j)(D(j) - D'(j))| \\
&= \sum_{j=1}^{w} |D(j) - D'(j)| \sum_{i=1}^{w} |M(i,j)| \\
&= \sum_{j=1}^{w} |D(j) - D'(j)| \\
&= \|D - D'\|_1,
\end{aligned}$$

where the inequality follows from the triangle inequality, and the second-to-last equality from the fact that

$$\sum_{i=1}^{w} |M(i,j)| = \sum_{i=1}^{w} M(i,j) = 1 \quad \forall j \in [w]$$

as M is a left stochastic matrix. ◀

▶ **Corollary 12.** *Let $R \in \mathbb{R}^w$ be a real vector and $D, D' \in \mathcal{D}$ be probability distributions such that $\|D - D'\|_1 \leq \varepsilon$, and $M \in \mathbb{R}^{w \times w}$ a left stochastic matrix. Then*

$$R^\top MD \geq R^\top MD' - \|R\|_\infty \cdot \varepsilon.$$

Proof of Corollary 12. $\|R^\top M(D' - D)\|_1 \leq \|R\|_\infty \|M(D' - D)\|_1 \leq \|R\|_\infty \|D' - D\|_1 \leq \|R\|_\infty \cdot \varepsilon$, where the first step follows from Holder's inequality. ◀

We are now ready to prove Theorem 8:

Proof of Theorem 8. Let us denote by $B^\varepsilon_1, \ldots, B^\varepsilon_{k-1}$ a split of the budget for the discretized problem with $B^\varepsilon_{\geq t} = B^\varepsilon_t + \ldots + B^\varepsilon_{k-1}$. Let $M^\varepsilon_1, \ldots, M^\varepsilon_{k-1}$ a set of transition matrices achieving welfare $R^\top M^\varepsilon_{k-1} \ldots M^\varepsilon_1 D^0_1 \geq OPT^\varepsilon \triangleq OPT - (k-1)\varepsilon \left\| R \right\|_\infty$ that is feasible with respect to budget split $B^\varepsilon_1, \ldots, B^\varepsilon_{k-1}$. Note that such a budget split and matrices exist by Claim 4. Let D^ε_t the probability distribution on layer t defined by these transition matrices, i.e.

$$D^\varepsilon_t = M^\varepsilon_{t-1} \ldots M^\varepsilon_1 D^0_1.$$

To prove the result, we will show by induction that for all $B_{\geq t} \geq B^\varepsilon_{\geq t}$, and for $D_t \in \mathcal{D}(\varepsilon)$ such that $\|D_t - D^\varepsilon_t\|_1 \leq \varepsilon$,

$$R^\top M(B_{\geq t}, D_t) D_t \geq OPT^\varepsilon - 2(k-t)\varepsilon \|R\|_\infty.$$

This will directly imply that as B^ε is one of the possible values of $B_{\geq 1}$,

$$R^\top M(B^\varepsilon, D_1) D_1 \geq OPT^\varepsilon - 2(k-1)\varepsilon \left\| R \right\|_\infty.$$

Combined with Claim 4 that states $OPT_\varepsilon \geq OPT - (k-1)\varepsilon \left\| R \right\|_\infty$, we will obtain the result.

Let us now provide our inductive proof. First, consider the transition from layer L_{k-1} to layer L_k. Note that

$$OPT^\varepsilon \leq R^\top M^\varepsilon_{k-1} \ldots M^\varepsilon_1 D^0_1 = R^\top M^\varepsilon_{k-1} D^\varepsilon_{k-1}.$$

Let $D_{k-1} \in \mathcal{D}(\varepsilon)$ be such that $\|D_{k-1} - D^\varepsilon_{k-1}\| \leq \varepsilon$. Note then that by Corollary 12,

$$R^\top M^\varepsilon_{k-1} D_{k-1} \geq R^\top M^\varepsilon_{k-1} D^\varepsilon_{k-1} - \varepsilon \|R\|_\infty.$$

Further, M^ε_{k-1} is feasible for Program (6) with respect to $B_{\geq k-1}, B_{\geq k} = 0$, given $B_{\geq k-1} \geq B^\varepsilon_{\geq k-1}$. As such, for $B_{\geq k-1} \geq B^\varepsilon_{\geq k-1}$, we have that

$$R^\top M(B_{\geq k-1}, D_{k-1}) D_{k-1} \geq R^\top M^\varepsilon_{k-1} D_{k-1},$$

and in turn

$$R^\top M(B_{\geq k-1}, D_{k-1}) D_{k-1} \geq OPT^\varepsilon - \varepsilon \|R\|_\infty.$$

Now, suppose the induction hypothesis holds at layer $t+1$. I.e., for all $B_{\geq t+1} \geq B^\varepsilon_{\geq t+1}$, for $D_{t+1} \in \mathcal{D}(\varepsilon)$ such that $\|D_{t+1} - D^\varepsilon_{t+1}\|_1 \leq \varepsilon$,

$$R^\top M(B_{\geq t+1}, D_{t+1}) D_{t+1} \geq OPT^\varepsilon - 2(k-t-1)\varepsilon \|R\|_\infty.$$

For any $B_{\geq t} \geq B^\varepsilon_{\geq t}$, note that one can set $B_{\geq t+1} = B^\varepsilon_{\geq t+1}$ and $B_t \geq B^\varepsilon_t$; hence, M^ε_t is feasible for Program (6) with respect to $B_t \geq B^\varepsilon_t, B^\varepsilon_{\geq t+1}$. Since $\|D_t - D^\varepsilon_t\|_1 \leq \varepsilon$ and $\|D_{t+1} - M^\varepsilon_t D^\varepsilon_t\|_1 \leq \varepsilon$, we have that by Corollary 12,

$$\begin{aligned} R^\top M(B^\varepsilon_{\geq t+1}, D_{t+1}) M^\varepsilon_t D_t &\geq R^\top M(B^\varepsilon_{\geq t+1}, D_{t+1}) M^\varepsilon_t D^\varepsilon_t - \varepsilon \|R\|_\infty \\ &\geq R^\top M(B^\varepsilon_{\geq t+1}, D_{t+1}) D_{t+1} - 2\varepsilon \|R\|_\infty. \end{aligned}$$

Using the induction hypothesis, we obtain that $R^\top M(B^\varepsilon_{\geq t+1}, D_{t+1}) M^\varepsilon_t D_t \geq OPT^\varepsilon - 2(k-t)\varepsilon \|R\|_\infty$. In particular, we get

$$R^\top M(B_{\geq t}, D_t) D_t \geq OPT^\varepsilon - 2(k-t)\varepsilon \|R\|_\infty,$$

which concludes the proof of the social welfare guarantee. For the running time, we note that at each time step t, we solve one instance of Program 6 for each of the (at most) $\frac{B}{\varepsilon}$ possible budget splits of $B_{\geq t}$ and for each of the $\left(\frac{1}{\varepsilon}\right)^w$ (by Claim 6) probability distributions in $\mathcal{D}(\varepsilon)$ in layer L_t and layer L_{t+1}; i.e., for each t, the algorithm solves $O\left(\frac{B}{\varepsilon}\left(\frac{1}{\varepsilon}\right)^{w^2}\right)$ optimization programs. Then, the algorithm finds the solution of all of these programs with the best objective value, which can be done in time linear in the number of such solutions, i.e. $O\left(\frac{B}{\varepsilon}\left(\frac{1}{\varepsilon}\right)^{w^2}\right)$. This is repeated for $k-1$ values of t. ◀

5 (Ex-post) Maximin Value Maximization

Although social welfare maximization is a natural objective, it is well-known that it can be "unfair" in the sense that it explicitly prioritizes the welfare of larger populations (here represented as initial positions that have larger probability mass) over smaller populations. We can alternately evaluate a solution according to the welfare of the *least-well-off* population (here represented by the initial position with the smallest expected value) and ask to optimize *that* objective. We show how to optimize this objective in this section, when one demands a deterministic solution.

5.1 A Dynamic Programming Algorithm for Computing an Ex-post Maximin Allocation

In this subsection, we adapt the dynamic programming approach in Section 4.1 to give an approximation algorithm for the problem of maximizing the minimum expected reward over all initial positions. Recall that $\mathcal{D}$, the probability simplex, denotes the set of all possible probability distributions on a layer. Intuitively, our algorithm for maximizing social welfare kept track of a single probability distribution in each subproblem: the overall probability of arriving at each vertex in the layer *over both the randomness of an individual's initial position, and the randomness of the transition matrix*. In order to optimize the *minimum* expected value over all initial positions, we will need to keep track of more state. At every layer L_t, we will keep track of the probability of reaching each vertex in that layer *from each initial position in the starting layer*. So, we will now keep track of collections of w probability distributions in $\mathcal{D}^w$, one for each starting position. We call the elements of $\mathcal{D}^w$ *population-wise distributions*.

We introduce a discretization $\mathcal{A}(\varepsilon)$ of $\mathcal{D}^w$, as follows: $\mathcal{A}(\varepsilon) \triangleq (\mathcal{D}(\varepsilon))^w$, where $\mathcal{D}(\varepsilon)$ denotes a ε-net of $\mathcal{D}$ (of size $\left(\frac{1}{\varepsilon}\right)^w$). Given a population-wise probability distribution $A_t \in \mathcal{A}(\varepsilon)$ at layer t, we write A_t^j for the probability distribution corresponding to population j. The algorithm works as follows, just as before, running backwards from the final layer to the first layer. We describe the algorithm below informally, a formal presentation may be found in the full version [1].

5.1.1 Algorithm

1. For t going backwards from $k-1$ to 1, the algorithm does the following, for every population-wise distribution $A_t \in \mathcal{A}(\varepsilon)$ on L_t:
   a. The algorithm explores all splits of the budget $B_{\geq t}$ for layers t to k into a budget B_t for the transition from L_t to L_{t+1} and a budget $B_{\geq t+1}$ for L_{t+1} to L_k, as well as all choices of output population-wise probability distributions $A_{t+1} \in \mathcal{A}(\varepsilon)$ on layer L_{t+1}.
   b. The algorithm then finds a near-optimal transition matrix from L_t to L_k for every budget decomposition, by using the previously computed near-optimal solution for layers L_{t+1} to L_k, and solving a program similar to Program 6. The program maximizes the *minimum reward obtained from any initial position*, assuming the population-wise distribution of individuals at layer L_t is given by $\mathcal{A}_t$.
   c. Finally, the algorithm picks and stores the best recovered transition matrices from layer L_t to L_k that yield the highest reward, among all the transition matrices recovered from step 1b.

The input population-wise probability distribution $A_1 \in \mathcal{D}^w$ on the first layer is defined in the following manner, $A_1^j := e_j$ (the j-th basis vector in the usual orthonormal basis of $\mathbb{R}^w$) for all $j \in [w]$.

Note that the Program in step 1b can be written as a linear program of size polynomial in w, using the same method that was employed to write Program (6) as a linear program.

5.2 Running Time and Ex-Post Maximin Value Guarantees

Remember that we let OPT_{MM} denote the maximin value value of the given network. The running time and accuracy guarantees of the described Algorithm are provided below:

▶ **Theorem 13.** *The Algorithm described in Section 5.1.1 with discretization parameter ε yields maximin value at least $OPT_{MM} - 3(k-1)\varepsilon\|R\|_\infty$, and has running time $O\left(k\frac{B}{\varepsilon}\left(\frac{1}{\varepsilon}\right)^{w^4} g(w)\right)$, where $g(w)$ is any upper-bound on the running time for solving the linear Program in Step 1b, which is always polynomial in w.*

This immediately implies the following corollary:

▶ **Corollary 14.** *The algorithm described in 5.1.1 with discretization parameter $\varepsilon' = \frac{\varepsilon}{3(k-1)}$ yields maximin value at least $OPT_{MM} - \varepsilon\|R\|_\infty$, and has running time $O\left(k^2\frac{B}{\varepsilon}\left(\frac{k}{\varepsilon}\right)^{w^4} g(w)\right)$, where $g(w)$ is a polynomial upper-bound on the running time of the linear Program in Step 1b.*

The proof of Theorem 13 is almost identical to that of Theorem 8. We provide a complete proof in the full version [1].

6 (Ex-ante) Maximin Value Maximization

In this section, we consider the problem of optimizing the *ex-ante* minimum expected value over all initial positions: in other words, we allow ourselves to find a *distribution* over solutions, and take expectations over the randomness of this distribution, solving:

$$\begin{aligned} \max_{\Delta M} \quad & \min_{j\in[w]} R^\top \mathbb{E}_{M\sim\Delta M}\left[M_{k-1}\dots M_1\right] e_j \\ \text{s.t.} \quad & \Delta M \in \Delta\mathcal{F}\left(B, M_1^0, \dots, M_{k-1}^0\right) \end{aligned} \tag{7}$$

We show in the full version [1] that this can yield strictly higher utility than optimizing the ex-post minimum value. We then give an algorithm for solving the ex-ante problem by exhibiting a game theoretic reduction to the ex-post problem.

6.1 Solving the Ex-ante Maximization Problem Using Algorithm 1

Because Program 3 is a max min problem over a polytope, we can view it as a zero-sum game, and the solution that we want corresponds to a maxmin equilibrium strategy of this game. As first shown by Freund and Schapire[11], it is possible to compute an approximate equilibrium of a zero-sum game if we can implement a *no-regret* learning algorithm for one of the players, and an approximate best-response algorithm for the other player: if we simply simulate repeated play of the game between a no-regret player and a best-response player, then the empirical average of player actions in this simulation converges to the Nash equilibrium of the game.

This forms the basis of our algorithm. One player plays the "multiplicative weights" algorithm over the initial positions in layer 1 of the graph. This induces at every round a distribution over initial positions. The best response problem, which must be solved by the other player, corresponds to solving a welfare-maximization problem given the distribution over initial positions represented by the multiplicative weights distribution. Fortunately, this is exactly the problem that we have already given a dynamic programming solution for. The solution in the end corresponds to the uniform distribution over the solutions computed by the best-response player over the course of the dynamics. The algorithm is described formally in the full version [1].

Algorithm 2 2-Player Dynamics for the Ex-Ante Maximin Problem.

Input: Time horizon T, reward vector R on layer L_k, initial transition matrices $M_1^0, \dots, M_{k-1}^0$, budget B, discretization parameter ε.
Output: $M^1, \dots, M^T \in \mathcal{F}\left(B, M_1^0, \dots, M_{k-1}^0\right)$.
Initialization: The no-regret player picks $D^1 = \left(\frac{1}{w}, \dots, \frac{1}{w}\right) \in \mathcal{D}$, the uniform distribution over $[w]$.
for $t = 1, \dots, T$ **do**

The no-regret player plays distribution $D^t \in \mathcal{D}$.
The best-response player chooses $M^t \in \mathcal{F}\left(B, M_1^0, \dots, M_{k-1}^0\right)$ such that

$$R^\top M_{k-1}^t \dots M_1^t D^t \geq \max_{M \in \mathcal{F}} R^\top M_{k-1} \dots M_1 D^t - \varepsilon \left\|R\right\|_\infty ,$$

using Algorithm 1.
The no-regret player observes $u_i^t = \frac{R^\top M_{k-1}^t \dots M_1^t e_i}{\|R\|_\infty}$ for all $i \in [w]$, and picks D^{t+1} via multiplicative weight update, as follows:

$$D^{t+1}(i) = \frac{D^t(i)\beta^{u_i^t}}{\sum_{j=1}^w D^t(j)\beta^{u_j^t}} \ \forall i \in [w],$$

with $\beta = \frac{1}{1+\sqrt{2\frac{\ln w}{T}}} \in [0, 1)$.

end

▶ **Lemma 15.** *Let $T > 0$, $\overline{\Delta M}$ be the probability distribution that picks $(M_1, \dots, M_{k-1}) \in \mathcal{F}(B, M_1^0, \dots, M_k^0)$ with probability*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{1}\left\{(M_1, \dots, M_{k-1}) = \left(M_1^t, \dots, M_{k-1}^t\right)\right\},$$

where $M^1, \dots, M^T$ are the outputs of Algorithm 2. Then $\overline{\Delta M}$ $\left(\varepsilon + \sqrt{2\frac{\ln w}{T}} + \frac{\ln w}{T}\right) \|R\|_\infty$-approximately optimizes Program 3.

The proof of Lemma 15 follows from interpreting Program 3 as zero-sum game, noting that the best response problem for the maximization player corresponds to the welfare-maximization problem for which we have an efficient algorithm, and then applying the no-regret dynamic analysis from Freund and Schapire [11]. The details are provided in the full version [1].

7 Price of Fairness

In this section, we compute lower bounds on a notion of "price of fairness", and we show these lower bounds are tight when restricting attention to pipelines whose edges are *all* malleable. Specifically, we compare the optimal welfare achievable with the welfare that is achievable if we instead use our budget to maximize the *minimum* value over initial positions – i.e. if we solve the maximin problem. We focus on the ex-post maximin problem – i.e. we prove our bounds with respect to *deterministic* solutions. We note that there may be many different maximin optimal solutions that differ in their overall welfare, and so we consider two variants of the price of fairness in our setting – comparing with both the *maximum* welfare consistent with a maximin optimal solution, and the *minimum* welfare consistent with a maximin optimal solution.

Let OPT_{SW} the optimal value of Program (1) (the optimal social welfare). Let S^f be the set of solutions to Program (2) (the deterministic maximin problem). Further, define

$$W(M_1, \ldots, M_{k-1}) \triangleq R^\top M_{k-1} \ldots M_1 D_1^0$$

to be the social welfare achieved by transition matrices $M_1, \ldots, M_{k-1}$, and

$$W_{fair}^+ \triangleq \max_{(M_1,\ldots,M_{k-1}) \in S^f} W(M_1, \ldots, M_{k-1}),$$

$$W_{fair}^- \triangleq \min_{(M_1,\ldots,M_{k-1}) \in S^f} W(M_1, \ldots, M_{k-1})$$

to be the maximum and minimum social welfare respectively that are consistent with maximin optimal solutions. We define two variants of "the price of fairness" in our setting as: $P_f^+ \triangleq \frac{OPT_{SW}}{W_{fair}^+} \geq 1$, and

$P_f^- \triangleq \frac{OPT_{SW}}{W_{fair}^-} \geq 1$.

Note that $P_f^+ \leq P_f^-$ always, as P_f^+ compares the optimal social welfare with the solution of Program 2 with highest social welfare , while P_f^- considers the solution that has the lowest social welfare. We provide matching lower bounds on P_f^+ and upper bounds on P_f^-. This, in turn, provides tight bounds on the price of fairness with respect to *any* choice of maximin solution.

7.1 Lower Bounds on P_f^+

Our lower bounds are based on the following construction:

▶ **Example 16.** Consider a network with only two layers, L_1 and L_2, such that L_1 has w nodes and L_2 has 2 nodes. Suppose the starting distribution is given by $D_1^0 = (1-(w-1)\varepsilon, \varepsilon, \ldots, \varepsilon)^\top$ for $\varepsilon > 0$ small enough, the reward vector is given by $R = (1,0)^\top$, and the initial transition matrix M_1^0 is given by

$$M_1^0 = \begin{pmatrix} 0 & \ldots & 0 \\ 1 & \ldots & 1 \end{pmatrix}.$$

I.e., in the initial transition matrix, every starting node transitions to the destination node that has reward 0, and the welfare of the initial network is 0. We assume all edges are malleable.

▶ **Theorem 17.** *For all $w \in \mathbb{N}$, for any $\delta > 0$, there exists a network with $k = 2$ with price of fairness*

$$P_f \geq \begin{cases} w - \delta & \text{if } 0 < B \leq 2 \\ \frac{2w}{B} - \delta & \text{if } 2 < B \leq 2w \\ 1 & \text{if } B \geq 2w \end{cases}.$$

The proof follows from solving the social welfare maximization problem and the maximin value problem on Example 16. The complete proof is provided in the full version [1].

7.2 Upper Bounds on P_f^-

Importantly, in this section, we restrict ourselves to pipelines such that *all* edges are malleable. In this case, we show upper bounds that tightly match the lower bounds of Section 7.1.

Our upper bounds will make use of the following claim, which bounds the maximum social welfare that can be achieved under budget B.

▶ **Lemma 18.**

$$OPT_{SW} \leq \|R\|_\infty \text{ and } OPT_{SW} \leq W^0 + \frac{B}{2}\|R\|_\infty,$$

where $W^0 = R^\top M_{k-1}^0 \dots M_1^0 D_1^0$ *is the initial welfare.*

The proof of this lemma is straightforward and is deferred to the full version [1].We will also need lower bounds on the social welfare achieved by any optimal solution to the maximin program. The first lower bound is a function of B and w, but is independent of W^0.

▶ **Lemma 19.** *When all edges are malleable, for any* $\left(M_1^f, \dots, M_{k-1}^f\right) \in S^f$*,*

$$W\left(M_1^f, \dots, M_{k-1}^f\right) \geq \min\left(1, \frac{B}{2w}\right)\|R\|_\infty.$$

The proof of Lemma 19 is deferred to the full version [1]. The second lower bound we need shows that the social welfare achieved by a solution to Program (2) is lower-bounded by the initial social welfare $W^0 = R^\top M_{k-1}^0 \dots M_1^0 D_1^0$.

▶ **Lemma 20.** *When all edges are malleable, for any* $\left(M_1^f, \dots, M_{k-1}^f\right) \in S^f$*,*

$$W\left(M_1^f, \dots, M_{k-1}^f\right) \geq W^0.$$

We defer the full proof of Lemma 20 to the full version [1]. We can now use Lemmas 18, 19 and 20 to derive nearly tight upper bounds on the price of fairness with respect to the worst maximin solution:

▶ **Theorem 21.** *For every instance of the problem in which edges are malleable, we have that*

$$P_f^- \leq \begin{cases} w+1 & \text{if } 0 < B \leq 2 \\ \frac{2w}{B} & \text{if } 2 < B \leq 2w \\ 1 & \text{if } B \geq 2w \end{cases}.$$

Proof. We divide the proof in three cases:

1. $B \geq 2w$. By Lemma 19, it must be the case that any optimal solution to Program (2) has welfare at least $\min\left(1, \frac{B}{2w}\right)\|R\|_\infty = \|R\|_\infty$. It is then immediately the case that $OPT_{SW} = \|R\|_\infty$ by Lemma 18 and $P_f = 1$.
2. $2 < B \leq 2w$. By Lemma 18, we have $OPT_{SW} \leq \|R\|_\infty$. Further, by Lemma 19, we have that any solution to Program (2) has welfare at least $\frac{B}{2w}\|R\|_\infty$. This immediately yields the result.

3. $0 < B \le 2$. By Lemma 18, we have $OPT_{SW} \le W^0 + \frac{B}{2}\|R\|_\infty$. By Lemmas 19 and 20, we have that the social welfare of any maximin solution is at least W^0 and at least $\frac{B}{2w}\|R\|_\infty$. Therefore, the price of fairness is upper-bounded on the one hand by

$$P_f^- \le \frac{W^0 + \frac{B}{2}\|R\|_\infty}{W^0} = 1 + \frac{\frac{B}{2}\|R\|_\infty}{W^0}$$

and on the other hand by

$$P_f^- \le \frac{W^0 + \frac{B}{2}\|R\|_\infty}{\frac{B}{2w}\|R\|_\infty} = w + \frac{W^0}{\frac{B}{2w}\|R\|_\infty}.$$

When $W^0 \ge \frac{B}{2w}\|R\|_\infty$, the first bound gives $P_f^- \le 1 + \frac{\frac{B}{2}\|R\|_\infty}{\frac{B}{2w}\|R\|_\infty} = w + 1$, and when $W^0 \le \frac{B}{2w}\|R\|_\infty$, the second bound yields $P_f^- \le w + \frac{\frac{B}{2w}\|R\|_\infty}{\frac{B}{2w}\|R\|_\infty} = w + 1$, which concludes the proof. ◀

8 Hardness of Approximation

In this section, we show that the problem of finding the ex-post maximin value of a pipeline intervention problem instance within an approximation factor of 2 is NP-hard in the general case, where the width w of the network is not bounded. More specifically, we show that no algorithm that has a time bound polynomial in w, k and B can give a 2-approximation to the maximin value unless $P = NP$. This hardness result holds for k as small as 17. We remark that our result and proof can be immediately extended to show hardness of C-approximation, for any constant C, for an appropriate choice of constant depth k.

We show this hardness result via a reduction from a gap version of the vertex cover problem. The result of Dinur and Safra [8] shows that it is NP-hard to approximate the minimum vertex cover by a factor smaller than 1.306. In particular, their result shows that the following gap version of vertex cover is NP-hard: given $(\mathcal{G}, \kappa)$, we wish to either know if the graph $\mathcal{G}$ has a vertex cover of size κ, or has no vertex cover smaller than size 1.306κ.

We provide a description of out reduction below, and defer a formal statement along with a complete proof to the full version [1].

8.1 The Reduction

Our reduction works as follows: we construct a pipeline intervention instance of constant width (17 layers) from the given graph. The first layer has a node corresponding to each edge (u, v) of the original graph, and is connected by edges to nodes corresponding to vertices u and v on the second layer. We set up the instance so that positive probability mass is only ever added to a set of edge disjoint paths, where each path corresponds to a vertex in the original graph. These paths are shown by the dark, solid lines in Figure 1. The main idea behind the reduction is the following - by observing how allocations finding the maximin value split the budget over these edge disjoint paths, we can find out which vertices would form a small vertex cover of the original graph.

Formally, let the given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we reduce from have n vertices ($|\mathcal{V}| = n$) and m edges ($|\mathcal{E}| = m$). We construct of a pipeline intervention problem instance $\mathcal{I}'$ with $k+2$ layers and width w, where $k = 15$ and w is polynomial in n. The instance $\mathcal{I}'$ has an associated budget $B(\kappa, \varepsilon) = 2k\kappa\varepsilon$ where $\varepsilon < \frac{1}{2}$. For the sake of clarity, we refer to the set of vertices $\mathcal{V}$ in the vertex cover instance as "vertices" and the vertices in the instance $\mathcal{I}'$ as "nodes". A complete description of instance $\mathcal{I}'$ is as follows:

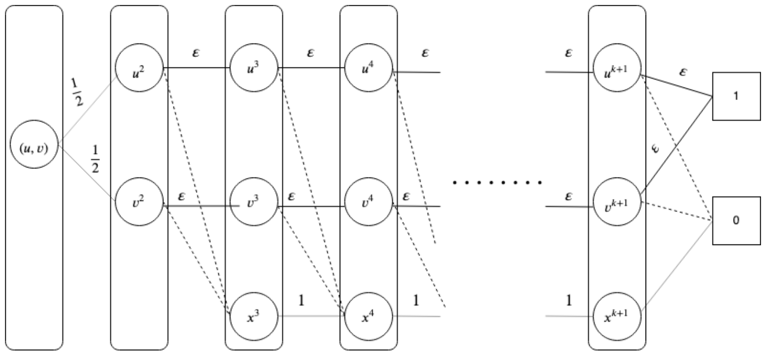

Figure 1 Constructed Instance of the Pipeline Intervention problem.

1. The first layer, L_1, has exactly m nodes, with each edge (u, v) in graph $\mathcal{G}$ having a unique corresponding node of the same label in layer L_1.
2. The second layer has exactly n nodes, with each vertex v in graph $\mathcal{G}$, having a unique corresponding node in layer L_2 with label v^2.
3. The next $k-1$ layers are of the following form - layer L_i, for $i = 3$ to $k+1$, has $n+1$ nodes. The first n nodes have labels from the set $\{v^i\}_{v \in V}$, i..e, each vertex v in the original graph $\mathcal{G}$ has a corresponding node v^i in layer L_i. The last node is indexed by x^i and exists to capture the "leftover" outward probability from the nodes $\{v^{i-1}\}_{v \in V}$ in layer L_{i-1}.
4. The final layer L_{k+2} has two reward nodes - y, of reward 1 and z, of reward 0.

We now describe the initial transition matrices.

1. From layer L_1 to layer L_2: for every node (u, v) in layer L_1, the outgoing probability is equally split between edges to nodes u^1 and v^1 in layer L_2, i.e., edges $((u, v), u^1)$ and $((u, v), v^1)$ each have probability $\frac{1}{2}$.
2. From layer L_i to layer L_{i+1} for $i = 2$ to k: For all vertices $v \in \mathcal{V}$ (i.e., the original graph), the corresponding edge (v^i, v^{i+1}) (in our construction) has probability ε. The remaining outgoing probability out of node v^i goes to the leakage node x^{i+1}. We call edges of the form (v^i, x^{i+1}) "leakage" edges. For $i \geq 3$, the edge (x^i, x^{i+1}) has all the outward probability, i.e., 1, from node x^i.
3. From layer L_{k+1} to layer L_{k+2}: each node in layer L_{k+1} is connected to z, the zero reward node, with probability 1.

We let P_v be the path going through nodes $v^2, v^3 \cdots v^{k+1}, y$ in our construction. We will refer to $\{P_v\}_{v \in \mathcal{V}}$ as vertex paths. Let E' be the set of edges found on paths $\{P_v\}_{v \in V}$. Let E'' contain of all the "leakage" edges in the instance $\mathcal{I}'$,i.e., edges of the form (v^i, x^{i+1}) as well as all edges of the form (v^{k+1}, z). We stipulate, as part of the description of the instance, that $E' \cup E''$ is the set of malleable edges in $\mathcal{I}'$ and that the probability mass on any other edge cannot be changed. This completes the description of instance $\mathcal{I}'$.

References

1 Eshwar Ram Arunachaleswaran, Sampath Kannan, Aaron Roth, and Juba Ziani. Pipeline interventions, 2020. `arXiv:2002.06592`.

2 Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664, 2017.

3 Matt Barnum. *A new study questions whether Head Start still produces long-run gains seen in past research*, 2019. URL: https://www.chalkbeat.org/2019/8/8/21108602/a-new-study-questions-whether-head-start-still-produces-long-run-gains-seen-in-past-research.

4 Amanda Bower, Sarah N Kitchen, Laura Niss, Martin J Strauss, Alexander Vargas, and Suresh Venkatasubramanian. Fair pipelines. *arXiv preprint*, 2017. arXiv:1707.00391.

5 Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

6 Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint*, 2018. arXiv:1810.08810.

7 Stephen Coate and Glenn C Loury. Will affirmative-action policies eliminate negative stereotypes? *The American Economic Review*, pages 1220–1240, 1993.

8 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.

9 Cynthia Dwork and Christina Ilvento. Fairness under composition. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

10 Dean P Foster and Rakesh V Vohra. An economic argument for affirmative action. *Rationality and Society*, 4(2):176–188, 1992.

11 Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332, 1996.

12 Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the 2018 World Wide Web Conference*, pages 1389–1398, 2018.

13 Christopher Jung, Sampath Kannan, Changwa Lee, Mallesh M. Pai, Aaron Roth, and Rakesh Vohra. Fair prediction with endogenous behavior. *Manuscript*, 2020.

14 Sampath Kannan, Aaron Roth, and Juba Ziani. Downstream effects of affirmative action. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 240–248, 2019.

15 Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6196–6200. AAAI Press, 2019.

16 Lydia T Liu, Ashia Wilson, Nika Haghtalab, Adam Tauman Kalai, Christian Borgs, and Jennifer Chayes. The disparate equilibria of algorithmic decision making when individuals invest rationally. *arXiv preprint*, 2019. arXiv:1910.04123.

17 Hussein Mouzannar, Mesrob I Ohannessian, and Nathan Srebro. From fair decision making to social equality. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 359–368, 2019.

18 Niche. *Best Public Elementary Schools in America*, 2020. URL: https://www.niche.com/k12/search/best-public-elementary-schools/.

19 Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692, 2014.

20 U.S. News. *Best High Schools Rankings*, 2020. URL: https://www.usnews.com/education/best-high-schools.

21 Samuel F Way, Daniel B Larremore, and Aaron Clauset. Gender, productivity, and prestige in computer science faculty hiring networks. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1169–1179, 2016.