Optimizing Fairness over Time with Homogeneous Workers

Bart van Rossum[1]
Econometric Institute, Erasmus University Rotterdam, The Netherlands

Rui Chen
Cornell Tech, New York City, NY, USA

Andrea Lodi
Cornell Tech, New York City, NY, USA

Abstract

There is growing interest in including fairness in optimization models. In particular, the concept of fairness over time, or, long-term fairness, is gaining attention. In this paper, we focus on fairness over time in online optimization problems involving the assignment of work to multiple homogeneous workers. This encompasses many real-life problems, including variants of the vehicle routing problem and the crew scheduling problem. The online assignment problem with fairness over time is formally defined. We propose a simple and interpretable assignment policy with some desirable properties. In addition, we perform a case study on the capacitated vehicle routing problem. Empirically, we show that the most cost-efficient solution usually results in unfair assignments while much more fair solutions can be attained with minor efficiency loss using our policy.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization

Keywords and phrases Fairness, Online Optimization, Combinatorial Optimization, Vehicle Routing

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.17

Category Short Paper

1 Introduction

While most optimization literature focuses on minimizing costs or maximizing efficiency, there is growing interest in including fairness in optimization models. In particular, fairness over time is desirable for sequential decision making. It relates to the situation where a central decision maker is faced with a multi-period optimization problem involving multiple agents, and the goal is to maximize fairness (minimize unfairness) of the (dis)utilities that agents gain from the solution. A mixed-integer programming framework has been proposed for the offline version of the problem, in which the optimization problems to be solved in all time periods are known in advance [1, 2]. The online version of the problem, where instances are revealed in a dynamic fashion, has been studied for, among other things, resource allocation [5] and railway crew planning [8].

In this paper, we restrict our attention to fairness over time in the context of online assignment problems. We define assignment problems as combinatorial problems that involve dividing the solution into blocks of work to be assigned to workers. Many real-life problems, including vehicle routing and crew scheduling, can be modeled in this way. In each time period, a local problem instance is revealed, and a solution must be determined with a cost that is within a pre-determined factor of the minimum cost. Next, the solution is assigned to a fixed group of workers whose utilities are updated accordingly. The goal is to determine solutions and assignments such that the unfairness of the workers' (dis)utilities is minimized.

[1] Corresponding author

23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).
Editors: Daniele Frigioni and Philine Schiewe; Article No. 17; pp. 17:1–17:6
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 The Online Assignment Problem with Fairness over Time

2.1 Problem Description

We consider a sequential decision making problem with a planning horizon of length T. In each period $t \in T$, the decision maker receives an instance $(c^t(\cdot), \mathcal{X}^t)$ of a combinatorial optimization problem, where $c^t(\cdot)$ and $\mathcal{X}^t$ denote its cost objective and its feasible region, respectively. The decision maker has to choose a solution $\boldsymbol{x}^t \in \mathcal{X}^t_\alpha$. Here, $\mathcal{X}^t_\alpha$ denotes the set of acceptable solutions to the local problem of period t, parameterized by α. In principle, the definition of the set of acceptable solutions can be very general. In this paper, we focus on cost-efficient solutions to avoid pathological cases of perfectly fair but highly inefficient solutions. Specifically, we define $\mathcal{X}^t_\alpha$ to be the set of feasible solutions whose costs are within a factor $(1+\alpha)$ of the minimum cost, i.e.,

$$\mathcal{X}^t_\alpha = \Big\{\boldsymbol{x} \in \mathcal{X}^t : c^t(\boldsymbol{x}) \le (1+\alpha) \min_{\boldsymbol{z} \in \mathcal{X}^t} c^t(\boldsymbol{z})\Big\}. \tag{1}$$

Varying α allows us to characterize the trade-off between efficiency and equity.

We assume that each solution $\boldsymbol{x}^t \in \mathcal{X}^t$ can be partitioned into n pieces of work $(x^t_1, \dots, x^t_n)$ that must be assigned to n homogeneous workers. As such, the work assignment can be freely permuted without changing its cost. With a slight abuse of notation, for each permutation $\pi \in \Pi^n$ of $\{1, \dots, n\}$, we define the mapping $\pi(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ by $\pi(y_1, \dots, y_n) = (y_{\pi_1}, \dots, y_{\pi_n})$. We assume that $\boldsymbol{x}^t \in \mathcal{X}^t$ if and only if $\pi(\boldsymbol{x}^t) \in \mathcal{X}^t$. We do not distinguish between a permutation of subvectors of $(x^t_1, \dots, x^t_n)$ and a permutation of coordinates of an n-dimensional vector as long as they follow the same order.

Each assignment of $\boldsymbol{x}^t$ yields a payoff vector $\boldsymbol{p}(\boldsymbol{x}^t) \in \mathbb{R}^n$, representing the payoffs to the n workers. Since the workers are homogeneous, we assume that the same piece of work x^t_i yields an identical payoff to each worker, i.e., $\boldsymbol{p}(\pi(\boldsymbol{x}^t)) = \pi(\boldsymbol{p}(\boldsymbol{x}^t))$ for all permutations π. Payoffs are aggregated in a linear fashion, such that the current utility vector equals the sum of all previous payoff vectors, i.e., $\boldsymbol{u}^t = \sum_{\tau=1}^t \boldsymbol{p}(\boldsymbol{x}^\tau)$. We let $\phi(\cdot)$ denote an unfairness measure of the worker's utilities satisfying the definition of the inequity measure in [6], e.g., the difference between the largest and smallest utilities. Our goal is to determine online a sequence of solutions $(\boldsymbol{x}^1, \dots, \boldsymbol{x}^T) \in \prod_{t=1}^T \mathcal{X}^t_\alpha$ such that $\phi(\boldsymbol{u}^T)$ is minimized. The offline version of the online assignment problem with fairness over time (OAPFoT) reads as

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} \quad & \phi(\boldsymbol{u}^T) \\ \text{s.t.} \quad & \boldsymbol{u}^t = \boldsymbol{u}^{t-1} + \boldsymbol{p}(\boldsymbol{x}^t),\ \boldsymbol{x}^t \in \mathcal{X}^t_\alpha, \quad t = 1, \dots, T, \\ & \boldsymbol{u}^0 = \boldsymbol{0}. \end{aligned}$$

In brief, in each time period t, the problem can be split into the following three steps:

1. *Defining the set the acceptable solutions.* In the case of $\mathcal{X}^t_\alpha$ being cost-efficient solutions as defined in (1), we must compute the optimal cost $\min_{\boldsymbol{y} \in \mathcal{X}^t} c^t(\boldsymbol{y})$ to explicitly define $\mathcal{X}^t_\alpha$. We assume that a suitable algorithm is available for this problem, and consider this step to be outside the scope of this paper.
2. *Picking an acceptable solution.* A criterion, which may or may not depend on $\boldsymbol{u}^{t-1}$, has to be decided in order to pick a solution from $\mathcal{X}^t_\alpha$. Then, potentially we need to solve an optimization problem associated with that criterion, which we refer to as the *α-subproblem*, to obtain a particular acceptable solution $\boldsymbol{z}^t$.
3. *Assigning the solution to workers.* If the criterion we use in Step 2 has not yet taken $\boldsymbol{u}^{t-1}$ into account, we may need to determine how to assign the solution $\boldsymbol{z}^t$ obtained in Step 2 to workers to obtain $\boldsymbol{x}^t = \pi(\boldsymbol{z}^t)$. In other words, we decide upon a permutation π.

The latter two steps can be used to minimize ϕ. Proposing good strategies for picking and assigning solutions is the main goal of this paper. In the remainder of this work, we analyze the performance of the following combination of strategies. In Step 2, we propose to pick the solution whose payoffs minimize the inequity function, i.e., we pick $\boldsymbol{z}^t \in \arg\min_{\boldsymbol{z}\in\mathcal{X}_\alpha^t} \phi(\boldsymbol{p}(\boldsymbol{z}))$. The rationale behind this strategy is that solutions with equitable payoffs, when properly assigned, lead to equitable utilities. In Step 3, we make use of a simple policy that assigns work, in increasing order of payoffs, to workers, in decreasing order of current utility. Some theoretical justifications for this assignment policy are provided in Section 2.3.

2.2 Complexity of the Problem

We first present some results regarding the complexity of OAPFoT. A simple reduction from the partition problem yields the following proposition.

▶ **Proposition 1** ($\mathcal{NP}$-hardness of the Offline OAPFoT). *The offline version of OAPFoT is $\mathcal{NP}$-hard even if the original cost minimization problem is solvable in polynomial time and $T = 1$, or even if each $\mathcal{X}_\alpha^t$ contains only solutions identical up to a permutation and $n = 2$.*

The above results indicate that even the offline assignment problem is hard to solve. We now turn our attention to online assignment policies.

▶ **Definition 2** (Online assignment policy). *An online assignment policy is a function $\tau(\cdot,\cdot)$ that maps any combination of a utitlity vector $\boldsymbol{u} \in \mathbb{R}^n$ and a payoff vector $\boldsymbol{p} \in \mathbb{R}^n$ to a permutation $\pi \in \Pi^n$.*

Informally, each policy determines how the next payoffs should be assigned to workers based on their current utilities and the next payoffs. Due to the online nature of the problem, it is easy to see that no such policy can always attain minimum unfairness.

▶ **Proposition 3** (Non-existence). *For $n \geq 2$ and $t \geq 3$, there does not exist an online assignment policy that attains perfect fairness on all instances that admit perfect fairness.*

2.3 A Simple Online Assignment Policy

We close this section with some positive results for one particular policy that is rather intuitive and simple to implement. This policy, which we refer to as the *best-to-worst policy* (BTW) and denote by τ^{BTW}, assigns payoffs, in increasing order, to workers, in decreasing order of current utility. More formally, $\pi^{\text{BTW}} := \tau^{\text{BTW}}(\boldsymbol{u},\boldsymbol{p})$ satisfies $\pi_i^{\text{BTW}}(\boldsymbol{p}) < \pi_j^{\text{BTW}}(\boldsymbol{p})$ if $\boldsymbol{u}_i > \boldsymbol{u}_j$ and only if $\boldsymbol{u}_i \geq \boldsymbol{u}_j$ with ties broken arbitrarily. We next show that this policy is always locally optimal in the unfairness measure ϕ.

▶ **Proposition 4** (Local optimality of BTW). *Let $\boldsymbol{u},\boldsymbol{p} \in \mathbb{R}^n$ and ϕ be an inequity measure. It holds that $\pi^{BTW} = \tau^{BTW}(\boldsymbol{u},\boldsymbol{p}) \in \arg\min_{\pi\in\Pi^n} \phi(\boldsymbol{u} + \pi(\boldsymbol{p}))$.*

Proof. Assume without loss of generality that $p_1 \leq p_2 \leq \ldots \leq p_n$, $u_1 \leq u_2 \leq \ldots \leq u_n$, and $\pi_i^{\text{BTW}} = n - i + 1$ for $i = 1,\ldots,n$. Let $\pi \in \arg\min_{\pi\in\Pi^n} \phi(\boldsymbol{u} + \pi(\boldsymbol{p}))$, and let i be the smallest index for which $\pi_i < n - i + 1$, i.e., for which the assignments of π and π^{BTW} differ. This implies that there exists a $j > i$ for which $\pi_j = n - i + 1 > \pi_i$. Recall that, by construction, $p_{\pi_i} \leq p_{\pi_j}$. If $p_{\pi_i} = p_{\pi_j}$, then we can reverse the assignments of i and j without affecting the resulting utilities. Otherwise, we have $p_{\pi_i} < p_{\pi_j}$, and reversing the assignments of i and j constitutes a Pigou-Dalton transfer that can only decrease unfairness [6]. Note that the smallest index satisfying our condition is now increased by at least one.

Hence, in at most $n-1$ of such transfers we can convert the assignment determined by π into that determined by π^{BTW}. Since each transfer can only decrease unfairness, it holds that $\phi(\boldsymbol{u}+\pi^{BTW}(\boldsymbol{p})) \leq \phi(\boldsymbol{u}+\pi(\boldsymbol{p}))$. Therefore, $\pi^{BTW} \in \arg\min_{\pi\in\Pi^n} \phi(\boldsymbol{u}+\pi(\boldsymbol{p}))$. ◀

In addition, under the BTW policy the unfairness at any stage is bounded by the maximum unfairness of any set of payoffs. We present a simplified version of this result for range unfairness, defined as the largest difference between payoffs/utilities.

▶ **Proposition 5** (Bounded range unfairness). *Under the best-to-worst policy and with* $\phi(\boldsymbol{u}) = \max_{i=1,\ldots,n} u_i - \min_{i=1,\ldots,n} u_i$, *it holds that* $\phi(\boldsymbol{u}^t) \leq \max_{\tau=1,\ldots,t} \phi(\boldsymbol{p}^\tau)$ *for all* $t = 1,\ldots,T$.

The above result is the main motivation for minimizing the unfairness of the payoffs in our proposed strategy, as it further minimizes an upper bound of $\phi(\boldsymbol{u}^t)$ for all t.

3 Case Study: Capacitated Vehicle Routing Problem

We perform a case study on the capacitated vehicle routing problem (CVRP) to test the effectiveness of our proposed strategies and to analyze the role of the budget parameter α. We define the payoff of a route in terms of either its distance or its load, i.e., we use both variable-sum and constant-sum payoffs [3]. As the unfairness measure ϕ, we use the range, defined as the largest difference in payoffs between any two routes. Solving the α-subproblem, i.e., $\boldsymbol{z}^t \in \arg\min_{\boldsymbol{z}\in\mathcal{X}_\alpha^t} \phi(\boldsymbol{p}(\boldsymbol{z}))$, now corresponds to selecting a set of routes for which the range, in terms of either distance or load, is minimized, subject to the α-budget constraint. This problem strongly relates to the CVRP with route balancing, for which no efficient exact algorithms are known.

We now introduce the necessary notations for the α-subproblem, omitting time indices t for brevity. Let N denote the set of customers, K denote the number of vehicles, i.e., workers, and B denote the cost of the most cost-efficient solution. We denote by R the set of all feasible routes and we introduce a binary variable x_r, $\forall r \in R$ that takes value 1 if route r is selected. The cost and payoff of route r are denoted by c_r and p_r, respectively. Binary parameter a_{ir} indicates whether customer $i \in N$ is visited by route r. We model the range using variables η and γ representing the maximum payoff and the minimum payoff, respectively. We model the maximum and minimum based on the last customer of the route.[2] Binary parameter b_{ir} indicates whether customer i is the last on route r. Finally, let M be an upper bound on the minimum payoff of any route. Our formulation for the α-subproblem now reads as

$$
\begin{aligned}
\min \quad & \eta - \gamma \\
\text{s.t.} \quad & \sum_{r\in R} a_{ir} x_r = 1, && \forall i \in N, \\
& \sum_{r\in R} p_r b_{ir} x_r \leq \eta, && \forall i \in N, \\
& M\left(1 - \sum_{r\in R} b_{ir} x_r\right) + \sum_{r\in R} p_r b_{ir} x_r \geq \gamma, && \forall i \in N, \\
& \sum_{r\in R} c_r x_r \leq B(1+\alpha), \; \sum_{r\in R} x_r = K, \; \boldsymbol{x} \in \{0,1\}^R.
\end{aligned}
$$

[2] We build on the formulation for the min-max multiple traveling salesman as presented by N. Bianchessi, C. Tilk, and S. Irnich at Column Generation 2023 in Montréal.

We solve the above program using branch and price, in which we solve a pricing problem for each possible last customer at each node of the search tree. The pricing problems are solved using bidirectional labeling and the ng-route relaxation [4]. Since we consider symmetric instances of the CVRP, we strengthen the formulation by enforcing that the index of the last customer along a route is at least that of the first customer. We branch on the last customer first, followed by arcs, and separate rounded capacity inequalities at the root node of the branching tree.

The set-up of our experiments is as follows. Similar to [3], we generate a sequence of 20 daily instances of $N = 15$ customers by taking disjoint subsets of instance X-n641-k35 [7]. This yields a single online instance of $T = 20$ time periods. We use $K = 5$ vehicles (one for each worker, i.e., $n = 5$), and set the vehicle capacity of each instance to $Q = \lceil \frac{1}{K-1} \sum_{i=1}^{N} q_i - 1 \rceil$. We consider values of α in $\{0, 1\%, \ldots, 10\%\}$, and use the best-to-worst policy to assign routes to workers. We solve the LP-relaxation of the restricted master problem using CPLEX 22.1.0.

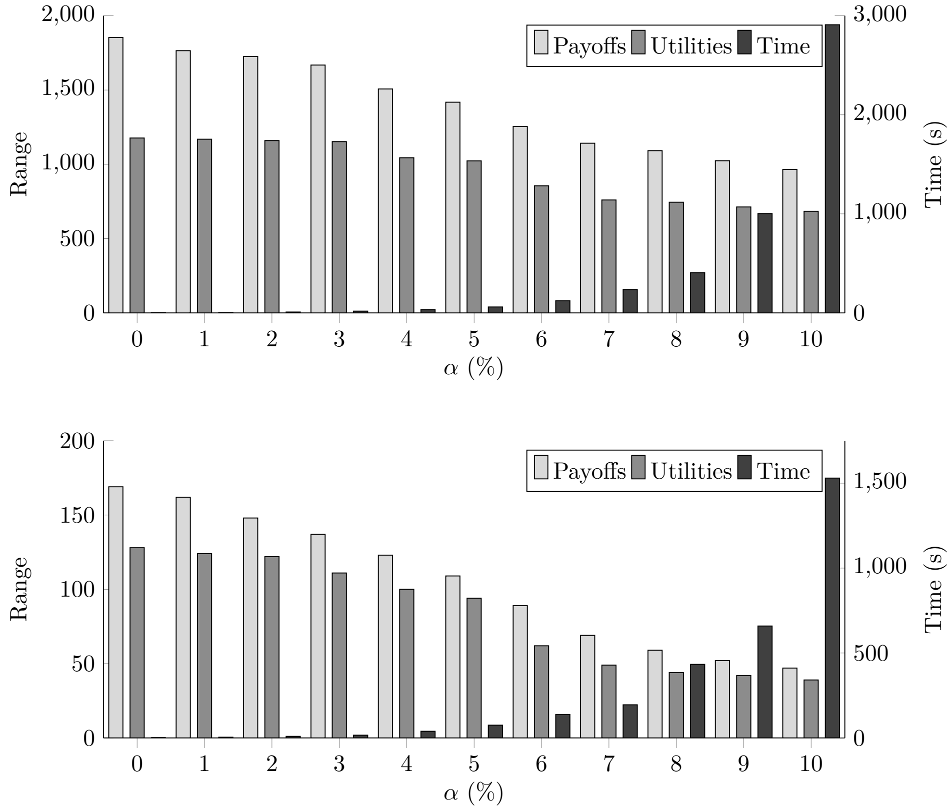

Figure 1 Results for the distance (top) and load (bottom) resources for different values of α.

Results of our case study for both the distance and load resource are summarized in Figure 1. For each value of α, we present the range of the payoffs ($\boldsymbol{p}$), utilities ($\boldsymbol{u}$), and the computing time. All results are averaged over the periods in our planning horizon. In line with Proposition 5, we find that the unfairness of the utilities is generally well below that of the payoffs, showing the effectiveness of the best-to-worst policy. Both ranges of payoffs and utilities decrease as α grows, though the effect is more pronounced for the payoffs. While we observe similar patterns for distance and load, we note that the range of the load displays a stronger reduction as a function of α. In addition, the range of the utilities is closer to that of the payoffs for load than for distance.

The computing times grow rapidly in α. We remark that the α-subproblem appeared to be a lot harder to solve than the cost minimization problem. While an optimal solution to the latter was always obtained within seconds, the former could take multiple hours for $\alpha = 10\%$. This discrepancy can be attributed to the large integrality gap of our formulation, resulting in large branch-and-bound trees, which can have more than 100,000 nodes.

4 Future Research

We consider several directions for future research. First, we aim to study the performance of using a different criterion for picking acceptable solutions. In particular, we will base the selection on current utilities $\boldsymbol{u}^{t-1}$ and select the solution $\boldsymbol{z}^t = \arg\min_{\boldsymbol{z}^t \in \mathcal{X}_\alpha^t} \phi(\boldsymbol{p}(\boldsymbol{z}^t) + \boldsymbol{u}^{t-1})$. This would effectively eliminate the need for an assignment policy, and would also require a slight reformulation of the α-subproblem. Surprisingly, preliminary experiments indicate that this approach might be counterproductive in some cases. Second, we aim to further explore why the α-subproblem appears to be much harder than the original cost-efficient problem. Hopefully, the resulting insights can be used towards the development of more efficient solution methods. We experimented with the use of cutting planes from the vehicle routing literature but these attempts proved ineffective, indicating the need for developing other techniques. Finally, we plan to perform a case study on a large-scale crew scheduling problem, to analyze the performance of our approach in settings with a larger number of workers.

References

1. Andrea Lodi, Philippe Olivier, Gilles Pesant, and Sriram Sankaranarayanan. Fairness over time in dynamic resource allocation with an application in healthcare. *Mathematical Programming*, 2022.
2. Andrea Lodi, Sriram Sankaranarayanan, and Guanyi Wang. A framework for fair decision-making over time with time-invariant utilities, 2022. URL: `https://arxiv.org/abs/2212.10070`.
3. Piotr Matl, Richard F Hartl, and Thibaut Vidal. Workload equity in vehicle routing: The impact of alternative workload resources. *Computers & Operations Research*, 110:116–129, 2019.
4. Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9:61–100, 2017.
5. Tareq Si Salem, Georgios Iosifidis, and Giovanni Neglia. Enabling long-term fairness in dynamic resource allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–36, 2022.
6. Man Yiu Tsang and Karmel S. Shehadeh. A unified framework for analyzing and optimizing a class of convex inequity measures, 2022. URL: `https://arxiv.org/abs/2211.13427`.
7. Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.
8. Bart van Rossum, Twan Dollevoet, and Dennis Huisman. Dynamic railway crew planning with fairness over time. Technical Report EI 2022-10, Econometric Institute, Erasmus University Rotterdam, 2022.