Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Special Track on AI for Good

Gigs with Guarantees: Achieving Fair Wage for Food Delivery Workers

Ashish Nair, **Rahul Yadav**, **Anjali Gupta**, **Abhijnan Chakraborty**, **Sayan Ranu** and **Amitabha Bagchi**

Dept. of Computer Science and Engineering, Indian Institute of Technology Delhi, India.
{Ashish.R.Nair.cs517, Rahul.Yadav.cs517, anjali, abhijnan, sayanranu, bagchi}@cse.iitd.ac.in

Abstract

With the increasing popularity of food delivery platforms, it has become pertinent to look into the working conditions of the 'gig' workers in these platforms, especially providing them fair wages, reasonable working hours, and transparency on work availability. However, any solution to these problems must not degrade customer experience and be cost-effective to ensure that platforms are willing to adopt them. We propose WORK4FOOD, which provides income guarantees to delivery agents, while minimizing platform costs and ensuring customer satisfaction. WORK4FOOD ensures that the income guarantees are met in such a way that it does not lead to increased working hours or degrade environmental impact. To incorporate these objectives, WORK4FOOD balances supply and demand by controlling the number of agents in the system and providing dynamic payment guarantees to agents based on factors such as agent location, ratings, etc. We evaluate WORK4FOOD on a real-world dataset from a leading food delivery platform and establish its advantages over the state of the art in terms of the multi-dimensional objectives at hand.

1 Introduction and Related Work

Food delivery platforms like Swiggy, Zomato, GrubHub or Deliveroo have become an extremely popular choice among customers to order and get food delivered to them. Alongside offering increased business to the restaurants, they also provide a livelihood to thousands of delivery agents, who pick the ordered food from the restaurants and deliver them at the customers' doorsteps. In developing countries with high unemployment rates, despite the 'gig' nature of delivery jobs, these platforms have become the only source of income for the majority of delivery agents [Fairwork, 2021; Khumalo, 2022]. However, a range of issues are presently plaguing the food delivery industry – poor working conditions of the agents, pressure of on-time delivery while navigating heavy traffic, opaque job assignments, etc. [Zhou and others, 2020]. Specifically, a major concern of the delivery agents is their inadequate income against the backdrop of rising cost of fuel and maintenance[1], forcing many agents to go on repeated strikes to demand better pay [Mitra, 2022; Wilks, 2022]. In fact, a non-profit labor watchdog organization Fairwork (`fair.work`) found that none of the food delivery platforms in India ensures legal minimum wage, even if an agent works for 10+ hours a day [Fairwork, 2021].

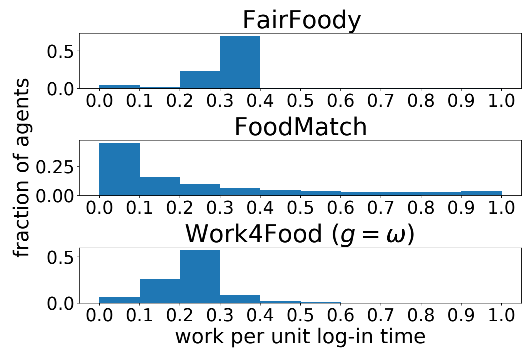


Figure 1: Work distribution in FAIRFOODY, FOODMATCH and our proposal WORK4FOOD (at a specific configuration $g = \omega$ described in Section 3.2) in city B of the dataset. An agent is said to be working if it is servicing an order.

Earlier works in this space have mostly attempted to minimize the order delivery time alone [Kottakki *et al.*, 2020; Ulmer *et al.*, 2021], by pre-placing delivery agents in different parts of the city anticipating demand [Xue *et al.*, 2021], or by minimizing delivery agents' waiting time [Weng and Yu, 2021]. FOODMATCH [Joshi *et al.*, 2022; Joshi *et al.*, 2021] overcame many simplifying assumptions made in prior works (e.g., perfect order arrival information [Yildiz and Savelsbergh, 2019], neglecting the road network [Reyes *et al.*, 2018] and food preparation time [Zeng *et al.*, 2019]), and proposed a realistic and scalable solution. While FOODMATCH minimized the delivery time, FAIRFOODY [Gupta *et al.*, 2022] showed that such one-sided optimization leads to *unfair* work distribution among delivery agents, resulting in unequal incomes. They further proposed a multi-objective algorithm FAIRFOODY to provide *fair distribution of income opportunities* among the agents, while ensuring minimal increase in

[1]A delivery agent typically gets a small delivery fee per order, except occasional tips and incentives [Sodhi, 2021].

Property	FAIRFOODY	FOODMATCH	WORK4FOOD $g = \omega$
Avg. Delivery Time	16.08 mins	15.91 mins	16.68 mins
Gini Income/log-in Time	0.09	0.63	0.11
Avg. Work per Agent	3.26 hrs	2.31 hrs	2.19 hrs
Cost (in 10E6 p units)	2.46	1.75	1.85 = 1.64 + 0.21

Table 1: Performance of FAIRFOODY, FOODMATCH, and our proposal WORK4FOOD in city B of the dataset. An order's delivery time is the time between the order being placed and delivered. We measure inequality among agent incomes (per unit of log-in time) with Gini scores. The lower the Gini score, the better the fairness. p is the pay of an agent per hour worked. Cost for WORK4FOOD includes handouts = 0.21 10E6 (Def. 5). For FAIRFOODY and FOODMATCH, cost here is the total payment for work.

City	# Restaurants	# Vehicles (avg./day)	# Orders (avg./day)	# Nodes	# Edges	Population
A	2085	2454	23442	39k	97k	>5M
B	6777	13429	159160	116k	299k	>8M
C	8116	10608	112745	183k	460k	>8M

Table 2: Overview of the dataset.

delivery times. To our knowledge, FAIRFOODY is the only prior work looking into the fairness of delivery agents' incomes.

FAIRFOODY tries to ensure that all agents get the same amount of work for every unit of time they are working for the platform. We plot the work distribution in FOODMATCH and FAIRFOODY in Fig. 1 when run on an identical stream of orders. In FAIRFOODY, all agents get concentrated around the same work per unit log-in time (0.3 – 0.4). In sharp contrast, FOODMATCH creates an allocation where some agents get most of the work, and a large number get very little work.

"Work" on a food delivery platform consists of three components: (1) driving to the restaurant (*first-mile*), (2) *waiting* at the restaurant while food is being prepared, and (3) driving from restaurant to the customer's location (*last-mile*). Note that the first-mile time and food preparation time can proceed in parallel. While FOODMATCH pre-dominantly assigns an order to the closest delivery agent from the restaurant's location, FAIRFOODY assigns work to a relatively far-away driver so that the driver reaches the restaurant within the waiting time, and therefore, minimally affect the final delivery time. Through this design, the location of an agent plays a less central role in determining the chances of being allocated an order, and consequently provides FAIRFOODY more flexibility in attaining equitable work distribution.

While the distribution of work certainly looks fairer in FAIRFOODY, there is a downside to this. As shown in Table 1, FAIRFOODY makes agents work significantly more to deliver the same set of orders to ensure fairness. Thus, higher work to deliver the same set of orders increases the platform's cost in terms of delivery payments, fuel costs for agents, and has a negative environmental impact.

In this context, we ask an important question: *Can equitable distribution be achieved without the pitfalls of additional greenhouse emissions, higher cost to the delivery platform, and a more relaxed work distribution among agents?* We design an algorithm called WORK4FOOD, which shows that the above objective can indeed be achieved (See Fig. 1 and Table 1). Our key contributions are as follows:

- **Novel problem:** We propose a more realistic approach to provide fair incomes by ensuring that all agents get income guarantees based on government-mandated minimum wage. In addition, the proposed formulation provides sufficient control to the platform to decide the number of agents to on-board so that the income guarantee can be met, which includes the provision of agent-personalized guarantees (§ 2).
- **Algorithm design:** We design an algorithm called WORK4FOOD, which allows us to set a payment guarantee that is *provably optimal* for minimizing cost for the platform. Powered by a novel combination of *minimum weight bipartite matching* with *Gaussian process regression*, WORK4FOOD analyzes the demand-supply dynamics in the system and generates an allocation that balances the *triple* needs of minimizing delivery time, minimizing cost for the platform, and ensuring payment guarantees for the delivery agents (§ 3).
- **Evaluation:** We evaluate WORK4FOOD on a real food delivery dataset from a large food delivery platform. Our experiments reveal that WORK4FOOD ensures the minimum wage guarantee with high probability, improves cost margin of the platform and achieves fairness without inflating work per unit time. More importantly, WORK4FOOD empowers policymakers with a tool that can be deployed to regulate this business (§ 4).

More detailed version of this work is available in ArXiv [Nair *et al.*, 2022].

2 Preliminaries and Problem Formulation

The food delivery problem involves four parties: customers, restaurants, delivery agents, and the delivery platform (a.k.a. the 'system'). The platform receives a stream of orders, which it then assigns to the agents. While doing so, we keep a few constraints in mind – agents have a fixed capacity to carry orders, and the platform does not assign orders that cannot get delivered within the service level agreement (SLA). The SLA is the maximum delivery time that the platform promises customers. Each agent logs into the platform and informs how long they plan to stay.

2.1 Dataset

We use a licensed food delivery dataset released by [Joshi *et al.*, 2021]. The dataset is sourced from a leading food delivery service in India and consists of eighteen days worth of delivery data from three large metropolitan cities. Table 2 provides a summary of the same. Further details are available in [Nair *et al.*, 2022].

2.2 Formulation

We now define some terms useful for our proposal.

Definition 1 (Delivery Times – EDT, SDT, XDT)**.** *The* expected delivery time $EDT(o, v)$ *of an order o is the expected time it would take to be delivered if it is assigned to agent v.*

The shortest delivery time $SDT(o)$ *of order o is the fastest it can be delivered if some vehicle could serve it with no wait time or other detour delays. It is the sum of the food preparation time and the shortest travel time between the restaurant and customer's location.*

The excess delivery time $XDT(o, v)$ *of an order o with respect to an agent v is the difference between* $EDT(o, v)$ *and* $SDT(o)$*. We compute these delivery times using standard graph algorithms as in [Joshi* et al.*, 2021].*

Problem 1 (Minimizing delivery time)**.** *Given a set of agents* V^t *and unallocated orders* O^t *at timestamp t, find an allocation of orders to agents so that* $\sum_{\forall o \in O^t} XDT(o, A(o))$ *is minimized.* $A(o)$ *denotes the agent allocated to order o.*

FOODMATCH [Joshi *et al.*, 2021] studies the above problem and proposes an effective solution.

Definition 2 (Work and Active Times)**.** Work time W_v^t *is the time spent by a delivery agent v till time t servicing orders, i.e., either traveling or waiting for assigned orders. The* active time A_v^t *(in hours) of the agent is the time for which they log in to the system till time t.* W_v *and* A_v *are the total* work time *and total* active time *of agent v over their entire shift.*

In all of the models we compare, the agent incomes are proportional to their work times. Therefore, *to guarantee income to agents, our new proposed algorithm guarantees that they get enough work.*

Definition 3 (Work Guarantee Ratio)**.** *The work guarantee ratio* g_v *of a delivery agent v is the amount of work time guaranteed by the system to agent v per unit active time. Note a work guarantee naturally translates into an income guarantee since the payment is a function of the work.*

Definition 4 (Work Guarantee)**.** *The* work guarantee G_v^t *for a delivery agent v is the amount of work (in hours) the platform guarantees to the agent till time t, i.e.,* $G_v^t = g_v \times A_v^t$*. The total* work guarantee G_v *of agent v is given by* $G_v = g_v \times A_v$*.*

Definition 5 (Work Payment, Handout and Platform Cost)**.** *In our models, every agent is paid the same rate p for every hour they worked. So, the work payment* P_v *for any agent v is given by* $P_v = p \times W_v$*.*

We envision a system where the platform hands out money to agents in lieu of any unmet work guarantees. So, the handout H_v *for any agent v is given by:*

$$H_v = p \times \max(0, G_v - W_v) \tag{1}$$

The platform cost C *is the total money the platform has to pay the agents for their work and any unsatisfied work guarantee. It is given by:*

$$C = \sum_v pW_v + \sum_v H_v \tag{2}$$

Problem 2 (Minimize Platform Cost)**.** *Find order to vehicle allocation such that C is minimized. Note that due to the handout component in C (Eq. 2), all agents are guaranteed to get their promised income.*

In this work, we propose to not only minimize food delivery time (Prob 1), but also the cost to platform (Prob 2).

Problem 3 (Work Guarantee Problem)**.** *Find order to vehicle allocation such that both platform cost C and total excess delivery time are minimized while ensuring guaranteed income to all agents.*

The proposed problem is therefore a *multi-objective optimization* problem. Moreover, Prob 3 is NP-hard since Prob 1 is NP-hard [Joshi *et al.*, 2021]. Hence, we explore heuristics in the form of bipartite matching.

3 WORK4FOOD: Proposed Algorithm

WORK4FOOD aggregates all incoming orders in a window of size $\Delta = 3$ minutes and any unassigned orders from past windows. As in FOODMATCH, we batch orders together if they can be picked and delivered efficiently by the same agent. We use the same batching algorithm as in FOODMATCH. We then assign each batch to an active agent. The pseudocode of WORK4FOOD is provided in Alg.1 in [Nair *et al.*, 2022].

3.1 Order Matching

To assign orders to agents, we perform matching on a *weighted bipartite graph* between order batches B^t and active vehicles V^t in the current window using the *Hungarian* algorithm [Kuhn, 1955; Munkres, 1957].

Edge weights in matching graph: The weight of the edges in the bipartite matching graph at time t between order batches $b \in B^t$ and vehicles $v \in V^t$ is given by

$$e(v, b) = \begin{cases} max\left\{W_v^t + w_v^b - G_v^t, 0\right\} & G_v^t > W_v^t \\ w_v^b & G_v^t \leq W_v^t \end{cases} \tag{3}$$

w_v^b denotes the extra work agent v will do to deliver batch b. The edge weight captures the additional cost to deliver batch b using agent v given the work guarantee. Any work below the work guarantee comes at no extra cost to the platform, as it needs to pay unmet guarantees through handouts. Thus, the edge weights equal the extra work beyond the guarantee.

The above edge weight does not explicitly optimize delivery times, so we add an extra term corresponding to FOODMATCH's edge weight [Joshi *et al.*, 2021].

If O_v is the batch (set) of orders already being carried by agent v, then the edge weight in FOODMATCH is given by

$$e_{fm}(v, b) = \sum_{o \in b} XDT(o, v) + \sum_{o' \in O_v} \Delta_{XDT}(v, o', b) \tag{4}$$

where $\Delta_{XDT}(v, o', b)$ is the change in XDT of order $o' \in O_v$ if batch b is assigned to agent v.

Theorem 1 (Equivalence to FOODMATCH)**.** *When batching is turned off, i.e., each agent has a capacity of only one order, and the platform provides no work guarantee to the agents, then* FOODMATCH *and* WORK4FOOD *compute the same batch-agent assignments.*

Proof. Provided in [Nair *et al.*, 2022]. □

3.2 Setting the Work Guarantee Ratio

A key component in determining the edge weights of the bipartite graph is the work guarantee to be provided to delivery agents (Eq. 3). This guarantee needs to be determined based on the demand-supply dynamics. To determine this guarantee, we initiate our analysis under the equitability assumption that the same guarantee is provided to all agents, i.e., *the guarantee is a function of only the demand-supply and does not depend on the attributes of the agent*. We discuss agent-personalized guarantee in the subsequent section.

Let us denote the fixed work guarantee ratio by g (Def. 3), pay per active hour by p, and hourly minimum wage guarantee by P_{min}. Thus, we want $p \times g \geq P_{min}$. For every g, platform can set an appropriate hourly pay $p = p(g) = P_{min}/g$ such that the wage guarantee is met with minimum cost.

Theorem 2 (Optimal value of g)**.** *If $\omega = \sum_v W_v / \sum_v A_v$ is the ratio of the total work time to the total active time of the system with the current set of orders and agents, then we claim that $g = \omega$ minimizes the platform cost C.*

Proof. This choice of g is explained by analysing the platform cost C for different g values. We can rewrite Eq. 2 as:

$$C(g) = \frac{P_{min}}{g} \sum_v W_v + \frac{P_{min}}{g} \sum_v \max(0, gA_v - W_v) \quad (5)$$

We consider two cases, $g \leq \omega$ and $g > \omega$ assuming WORK4FOOD works ideally, i.e., (i) the total work in the system $\sum_v W_v$ does not depend on g; (ii) WORK4FOOD meets guarantees whenever there is sufficient work; (iii) when work is scarce, agents are not assigned more work than the guarantee as agents with unmet guarantees are preferred by Eq. 3.

Case 1: $g \leq \omega$**.** Here, there is enough work available for all the agents to be able to meet the fixed work guarantee since $\sum_v gA_v \leq \omega \sum_v A_v = \sum_v W_v$. So, the second term in Eq. 5 would be equal to zero and we have $C(g) \propto 1/g$. Hence, $C(g)$ is minimized when $g = \omega$.

Case 2: $g > \omega$**.** As there is not enough work for all guarantees to be met, agents will not cross their guarantees, and Eq. 5 would reduce to:

$$C(g)_{|g>\omega} = \frac{P_{min}}{g} \sum_v W_v + \frac{P_{min}}{g} \sum_v (gA_v - W_v) \quad (6)$$

$$= P_{min} \sum_v A_v \quad (7)$$

So, if $g > \omega$ then, C is independent of g. However, in this case, the distribution of work for the agents may be unequal even though they end up earning equally (proportional to their active times). This is not desirable. Hence, we set $g = \omega$. □

An empirical substantiation of the above theorem is provided in [Nair *et al.*, 2022].

3.3 Estimating ω_v for Each Agent

From Theorem 2, setting $g = \omega$ is the optimal choice when all agents get the same guarantee. However, the total work time to total active time varies with time of the day (See Fig. 2),

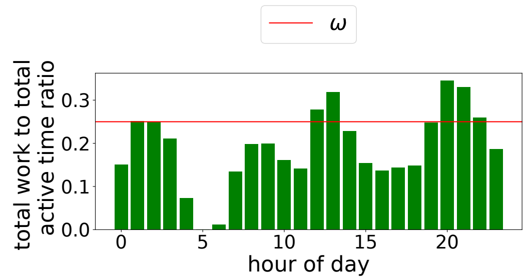


Figure 2: Total work to total active time at different hours in city B.

since it is a function of several variable factors such as the location of agents, number of agents in the system, and order-density. Hence, it is important to *predict* ω_v, i.e., the ratio of expected work to active time when an agent v wants to on-board. Towards that end, we use *Gaussian Process Regression* [Rasmussen and Williams, 2006] to model available work W_v of agent v. As input, the agent provides its active time A_v on joining the system. So, we have $\omega_v = W_v/A_v$ and the work guarantee ratio is set as $g_v = \omega_v$ to provide a dynamic guarantee.

Gaussian Process Regression

Gaussian process regression (GPR) is a *Bayesian regression model* used in geo-spatial and time series interpolation tasks. It makes predictions based on the similarity between known points and unknown points.

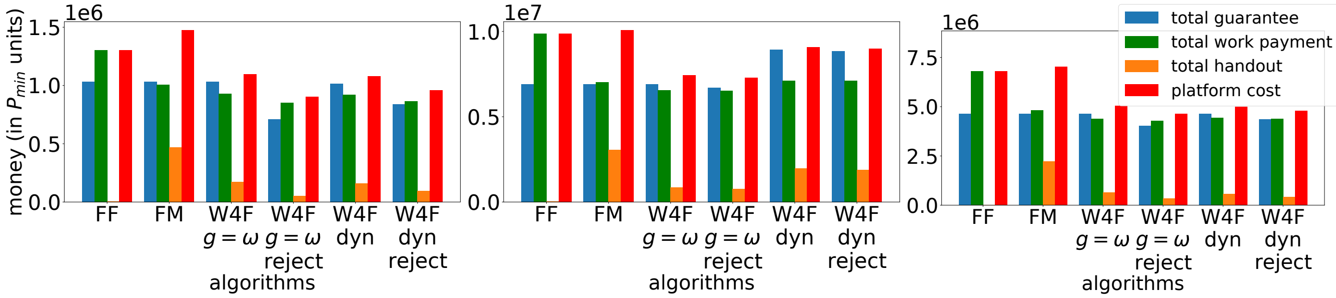
For a given train dataset $(\mathbf{X}, Y)$ and test points $\mathbf{X}_*$, GPR predicts the underlying function f. In the basic GPR model, we have:

$$\begin{bmatrix} f(\mathbf{X}_*) \\ f(\mathbf{X}) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\mathbf{X}_*, \mathbf{X}_*) & k(\mathbf{X}_*, \mathbf{X}) \\ k(\mathbf{X}_*, \mathbf{X})^T & k(\mathbf{X}, \mathbf{X}) \end{bmatrix}\right) \quad \text{(prior)}$$

$$y(\mathbf{X}) \sim \mathcal{N}\left(f(\mathbf{X}), \eta^2 I\right) \quad \text{(likelihood)}$$

$y(\mathbf{X})$ and $f(\mathbf{X})$ are random variable vectors. Training data Y is modeled as samples of $y(\mathbf{X})$. $k(x, x') = \sigma^2 \cdot exp(-(x - x')^2/2l^2)$ is the *kernel function* and $\eta^2 I$ is the noise covariance. σ, l and η are model parameters *learned* using *gradient descent* to best explain the training data $(\mathbf{X}, Y)$. The predictions of the GPR model are made by computing $p(f(\mathbf{X}_*) \mid y)$ by marginalizing $f(\mathbf{X})$ from $p(f(\mathbf{X}_*), f(\mathbf{X}) \mid y)$. We use the *Sparse Variational* version of the GPR (with *Cholesky Variational Distribution*) for better running times on the large dataset. Variational inference uses a new distribution $q(f(\mathbf{X}_*), f(\mathbf{X}))$, called the *variational distribution*, to approximate $p(f(\mathbf{X}_*), f(\mathbf{X}) \mid y))$. In Sparse GPR, inducing locations X_s are found to summarize the training data.

In our setting, the GPR predicts the total work $W_v = \mathbf{X}_*$ an agent v will get during its active time, which is used for getting ω_v. We characterize each agent $v \in V^t$ with the feature set $\mathbf{X} = \{\mathbf{X}^v \mid v \in V^t\}$, where $\mathbf{X}^v$ is a vector containing log-in/log-off time, log-in location coordinates in terms of latitude and longitude, number of agents currently in the system, and the number of orders per window. Other features may also be used based on the factors influencing allocation (ex. driver rating).



(a) City A (b) City B (c) City C

Figure 3: Cost analysis of different algorithms.

We note that we only need an estimate of the available work to provide a guarantee that can be met. Getting the prediction exactly correct is not required. Hence, GPR is an appropriate choice. In addition, GPR does not require us to set the functional form of the prediction model or to tune any hyper-parameters.

Balancing agent-order dynamics: FOODMATCH [Joshi *et al.*, 2021] shows that platforms tend to engage more agents than required particularly, on slots that do not correspond to lunch and dinner hours. Over-provisioning drivers results in lower guarantees and consequently may violate the minimum wage guarantee as mandated by law. The prediction from GPR enables platforms to on-board the optimal number of drivers. Specifically, the system would on-board drivers as long as the expected pay per hour, i.e., $\omega_v \times p$, is higher than the minimum pay per hour.

4 Empirical Evaluation

In this section, we benchmark WORK4FOOD against the baselines of FOODMATCH [Joshi *et al.*, 2021] and FAIRFOODY [Gupta *et al.*, 2022] and establish that:

- **Practicality:** Compared to FOODMATCH and FAIRFOODY, WORK4FOOD provides a more practical balance between delivery times and cost-efficiency of the platform.
- **Fairness and sustainability:** WORK4FOOD generates fair allocations without the pitfalls of strenuous work distribution among agents or additional greenhouse emissions.

More experiments on running time efficiency and ratings-based work guarantee can be found in [Nair *et al.*, 2022]. Our codebase is available at https://github.com/idea-iitd/Work4Food. Setup details is provided in [Nair *et al.*, 2022].

4.1 Cost to Platform

Fig. 3 analyses the monetary cost of each baseline and WORK4FOOD under several payment guarantees across all three cities. The work guarantees (and therefore payment guarantee) used to evaluate FAIRFOODY (FF) and FOODMATCH (FM) are set to $g_v = \omega$ (Recall Theorem 2). Hence, the "total guarantee" bars for the first three algorithms are the same in Fig. 3. Recall, any unmet payment guarantees are compensated through handouts.

The platform cost (red bar) is up to 25% lower with WORK4FOOD (with $g = \omega$) compared to FOODMATCH and FAIRFOODY. This highlights the efficacy of WORK4FOOD in keeping platform cost low through explicit modeling of pay gap in edge weights (Eq. 3). The cost is highest in FOODMATCH since its allocations are skewed towards a minority of agents resulting in large handouts to the remaining agents.

We next focus on the variations of WORK4FOOD where agents are rejected if payment guarantees are predicted to not be met (reject) as per GPR and setting ω dynamically through GPR. We note that when driver rejection is allowed, the handout component decreases by up to 70%, and thereby making the system even more cost-efficient. When g is set dynamically based on demand and supply, the income for agents increases. While this naturally leads to an increase in cost to platform (although remains substantially lower than FOODMATCH and FAIRFOODY), dynamic guarantees provide a more transparent working environment for agents.

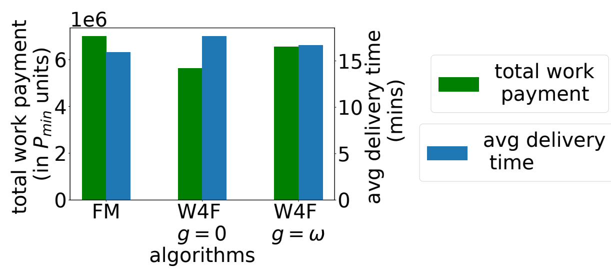


Figure 4: Comparing WORK4FOOD with no guarantee in city B.

Property	FF	W4F $g = 4\omega/3$
Avg. Delivery Time	15.98 mins	16.14 mins
Gini Income/log-in Time	0.07	0.02
Avg. Work per Agent	2.36 hrs	1.89 hrs

Table 4: Performance of FAIRFOODY v/s WORK4FOOD in city A.

4.2 Delivery Times, Equitability and Environmental Impact

Table 3 evaluates WORK4FOOD and baselines on the above mentioned metrics. The below analysis establishes that WORK4FOOD provides the best balance between cost-efficiency, delivery time, equitability of work and income, and environmental impact.

City	Property	FF	FM	W4F			
				$g = \omega$	$g = \omega$ reject	dynamic	dynamic reject
A	Avg. Delivery Time	15.98 mins	15.65 mins	16.45 mins	17.1 mins	16.48 mins	16.97 mins
	SLA Violations (%)	0.12	0.02	0.15	0.34	0.13	0.28
	Gini Income/Active Time	0.07	0.6	0.1	0.24	0.25	0.36
	Gini Work for Min. Wage	0	0.31	0.11	0.05	0.11	0.04
	Avg. Work per Agent	2.36 hrs	1.83 hrs	1.7 hrs	1.94 hrs	1.68 hrs	1.98 hrs
	CO_2 Emission	68 tonnes	46 tonnes	46 tonnes	41 tonnes	46 tonnes	42 tonnes
	Running Time per Window	3.3 s	0.7 s	2.6 s	2.5 s	2.2 s	2.1 s
B	Avg. Delivery Time	16.08 mins	15.91 mins	16.68 mins	16.7 mins	16.51 mins	16.51 mins
	SLA Violations (%)	0.19	0.15	0.26	0.27	0.27	0.27
	Gini Income/Active Time	0.09	0.63	0.11	0.12	0.33	0.33
	Gini Work for Min. Wage	0.01	0.29	0.09	0.08	0.08	0.07
	Avg. Work per Agent	3.26 hrs	2.31 hrs	2.19 hrs	2.21 hrs	2.37 hrs	2.40 hrs
	CO_2 Emission	364 tonnes	245 tonnes	239 tonnes	238 tonnes	256 tonnes	255 tonnes
	Running Time per Window	25.5 s	11.4 s	19.8 s	19.5 s	19.2 s	19.1 s
C	Avg. Delivery Time	16.82 mins	16.53 mins	17.35 mins	17.47 mins	17.28 mins	17.34 mins
	SLA Violations (%)	0.39	0.28	0.56	0.60	0.55	0.57
	Gini Income/Active Time	0.1	0.65	0.1	0.17	0.25	0.31
	Gini Work for Min. Wage	0	0.32	0.1	0.06	0.1	0.06
	Avg. Work per Agent	2.90 hrs	2.06 hrs	1.90 hrs	1.92 hrs	1.92 hrs	1.96 hrs
	CO_2 Emission	293 tonnes	186 tonnes	183 tonnes	178 tonnes	184 tonnes	182 tonnes
	Running Time per Window	31.9 s	8.9 s	23.1 s	23.1 s	22.2 s	21.6 s

Table 3: Performance analysis.

Delivery time: FOODMATCH has the lowest delivery times since it specifically minimizes this factor. However, WORK4FOOD does not perform poorly. Specifically, all of its variations across all cities are within 1.5 mins of FOODMATCH. The agent reject versions of WORK4FOOD have only slightly higher delivery times, even with a reduced set of agents available for delivery. It points to how the platform keeps under-worked agents in the system even when there is no real impact on customer experience.

Typically, the food delivery platform provides service level agreements (SLAs), which mandate the time within which each order must be delivered. In our dataset, the SLA was 45 minutes. We measure the number of SLA violations by each algorithm. As visible in Table 3, SLA violation is not a concern for any algorithm since it is always below 1%.

Equitability: Inequality among agents is measured using the Gini score. Lower the Gini, better the equality among agents. We calculate Gini across two component distributions: **(1)** agents' incomes per unit active time and **(2)** work done by agents to get to the minimum payment guarantee. The results reveal several interesting insights. First, FOODMATCH has a highly unequal distribution of work and pay, as seen from the Gini Scores. The Gini income per active time scores of WORK4FOOD with a fixed guarantee is close to that of FAIRFOODY, which specifically optimizes the income distribution. However, as well see later, while the excellent Gini of FAIRFOODY comes at the cost high CO_2 emissions and strenuous workload, WORK4FOOD does not suffer from these issues. Finally, we note that rejecting agents predicted to get low work leads to better equality among agents in terms of the work they do to earn minimum wage.

Environmental Impact: Even as FAIRFOODY fairly distributes work, it forces agents to work 30-40% more on average than FOODMATCH, thus making the system highly inefficient. It also leads to higher fuel costs for agents and causes a negative environmental impact. We measure the environmental impact using CO_2 emissions calculated from the total distance traveled by agents [Moran, 2018]. WORK4FOOD, on the other hand, does not lead to higher work times or CO_2 emissions to achieve better wage fairness and guarantees.

4.3 Flexibility of WORK4FOOD

Fine-tuning the work guarantee parameter g allows customization of WORK4FOOD towards various needs. In the next discussion, we highlight some of these aspects.

Fairness: The work to active time ratio on average in FAIRFOODY is $4/3$ times that of WORK4FOOD. We run WORK4FOOD at $g = 4\omega/3$ and compare the results with FAIRFOODY in Table 4 (Recall ω from Theorem 2). We observe that WORK4FOOD at $g = 4\omega/3$ not only provides better income equality, but also more relaxed agent workload. These benefits of WORK4FOOD do not come at the cost of delivery time as it only increases by a minuscule 9 seconds.

Platform Cost: What happens if we provide no work guarantee, i.e., $g = 0$. Fig. 4 answers the question. With no guarantees, WORK4FOOD is 20% cheaper than FOODMATCH but has 1 min 45 s higher average delivery time. This variation of the algorithm can reduce the platform cost and is the cheapest of all algorithms compared.

5 Conclusion

WORK4FOOD provides a mechanism for food delivery platforms to control and guarantee agent wages while at the same time optimizing their costs and maintaining reasonable delivery times. Optimizing platform cost also leads to a more efficient system with agents not being forced to work more to earn fair wages. It leads to savings in fuel costs and avoids negative environmental impact in order to provide better wages to agents. We also show that by tuning the work guarantee parameter of WORK4FOOD, it can be customized to address various objectives including minimizing cost, optimizing delivery times, or providing guarantees based on agent performance. Most importantly, WORK4FOOD injects transparency into the system where agents know the income promised to them and the service provider is assured that cost would be minimized without hampering delivery times.

References

[Fairwork, 2021] Fairwork. Fairwork India ratings 2020: Labour standards in the platform economy. https://fair.work/wp-content/uploads/sites/131/2021/01/Fairwork_India_2020_report.pdf, 2021. Accessed: 2022-06-05.

[Gupta *et al.*, 2022] Anjali Gupta, Rahul Yadav, Ashish Nair, Abhijnan Chakraborty, Sayan Ranu, and Amitabha Bagchi. Fairfoody: Bringing in fairness in food delivery. In *Proc. AAAI*, 2022.

[Joshi *et al.*, 2021] Manas Joshi, Arshdeep Singh, Sayan Ranu, Amitabha Bagchi, Priyank Karia, and Puneet Kala. Batching and matching for food delivery in dynamic road networks. In *Proc. ICDE*, 2021.

[Joshi *et al.*, 2022] Manas Joshi, Arshdeep Singh, Sayan Ranu, Amitabha Bagchi, Priyank Karia, and Puneet Kala. Foodmatch: Batching and matching for food delivery in dynamic road networks. *ACM Trans. Spatial Algorithms Syst.*, 8(1), mar 2022.

[Khumalo, 2022] Sibongile Khumalo. Food delivery apps bring new hope to SA's jobless, but it's not always a piece of cake. https://www.news24.com/fin24/companies/food-delivery-apps-bring-new-hope-to-sas-jobless-but-its-not-always-a-piece-of-cake-20220205, 2022. Accessed: 2022-06-05.

[Kottakki *et al.*, 2020] Krishna Kumar Kottakki, Sunil Rathee, Kranthi Mitra Adusumilli, Jose Mathew, Bharath Nayak, and Saket Ahuja. Customer experience driven assignment logic for online food delivery. In *Proc. IEEE IEEM*, 2020.

[Kuhn, 1955] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[Mitra, 2022] Debraj Mitra. Food delivery chain partners strike in Kolkata, seek higher pay and bar on outsourcing. https://www.telegraphindia.com/my-kolkata/news/food-delivery-chain-partners-strike-in-parts-of-kolkata-seek-higher-pay-and-bar-on-outsourcing/cid/1848505, 2022. Accessed: 2022-06-05.

[Moran, 2018] Greg Moran. Emissions 101: A two-wheeler takedown. https://auto.economictimes.indiatimes.com/autologue/emissions-101-a-two-wheeler-takedown/2826, 2018. Accessed: 2022-06-05.

[Munkres, 1957] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[Nair *et al.*, 2022] Ashish Nair, Rahul Yadav, Anjali Gupta, Abhijnan Chakraborty, Sayan Ranu, and Amitabha Bagchi. Gigs with guarantees: Achieving fair wage for food delivery workers. *arXiv preprint arXiv:2205.03530*, 2022.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* Adaptive computation and machine learning. MIT Press, 2006.

[Reyes *et al.*, 2018] Damián Reyes, Alan L. Erera, Martin W. P. Savelsbergh, Sagar Sahasrabudhe, and Ryan J. O'Neil. The meal delivery routing problem. Optimization Online, 2018.

[Sodhi, 2021] Tanishka Sodhi. We are slaves to them: Zomato, Swiggy delivery workers speak up against unfair practices. https://www.newslaundry.com/2021/08/14/we-are-slaves-to-them-zomato-swiggy-delivery-workers-speak-up-against-unfair-practices, 2021. Accessed: 2022-06-05.

[Ulmer *et al.*, 2021] Marlin W. Ulmer, Barrett W. Thomas, Ann Melissa Campbell, and Nicholas Woyak. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transp. Sci.*, 55:75–100, 2021.

[Weng and Yu, 2021] Wentao Weng and Yang Yu. Labor-right protecting dispatch of meal delivery platforms. In *Proc. IEEE CDC*, 2021.

[Wilks, 2022] Andrew Wilks. Turkey's food delivery couriers latest to strike amid economic crisis. https://www.al-monitor.com/originals/2022/02/turkeys-food-delivery-couriers-latest-strike-amid-economic-crisis, 2022. Accessed: 2022-06-05.

[Xue *et al.*, 2021] Guiqin Xue, Zheng Wang, and Guangwei Wang. Optimization of rider scheduling for a food delivery service in o2o business. *Journ. of Adv. Trans.*, 2021.

[Yildiz and Savelsbergh, 2019] Baris Yildiz and Martin Savelsbergh. Provably high-quality solutions for the meal delivery routing problem. *Transportation Science*, 53, 07 2019.

[Zeng *et al.*, 2019] Yuxiang Zeng, Yongxin Tong, and Lei Chen. Last-mile delivery made practical: An efficient route planning framework with theoretical guarantees. *Proc. VLDB Endow.*, 13(3):320–333, 2019.

[Zhou and others, 2020] Irene Zhou et al. Digital labour platforms and labour protection in China. Technical report, International Labour Organization, 2020.