An Algorithmic Foundation for Fair, Secure, and Differentially Private Distributed Discrete Optimal Transport

By

Jason Hughes

Thesis

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master's of Science

in the Department of Computer & Information Sciences

at Fordham University

New York, NY

August 2021

Acknowledgements

I would like to thank my advisor, Dr. Juntao Chen for introducing me to the fascinating field of optimal transport and for his guidance over the past year. I would also like to thank my undergraduate advisors Dr. Damian Lyons and Dr. Rolf Ryham for their encouragement and belief in me, the Mathematics and Computer and Information Sciences Departments for their help and resources, and Dr. Szu-Pei Fu and Dr. Ying Mao for taking the time and effort to serve on my review committee.

Finally, I would like to thank my family for their encouragement and belief in me over the past 23 years and for always pushing me to do more. I also want to thank my friends for making the past five years at Fordham feel like home. Without them I would not be where I am today.

August 2021

Contents

Chapter 1

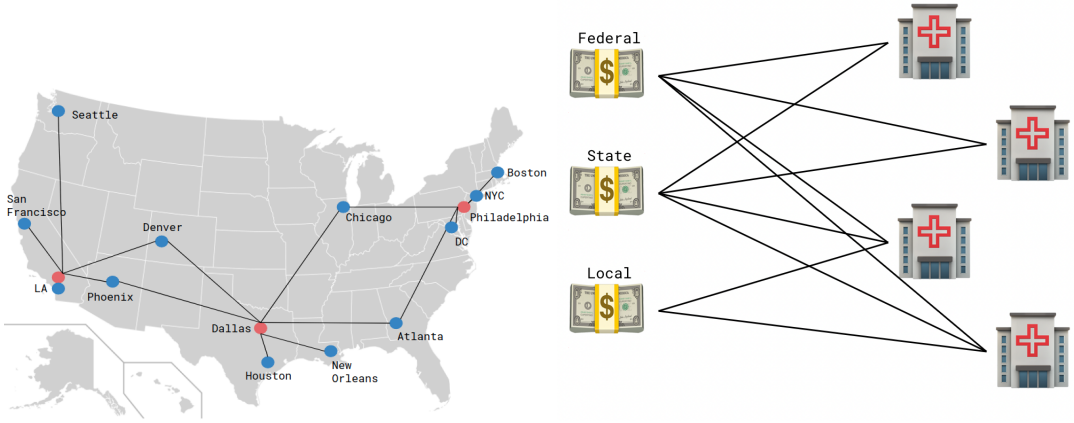
Introduction and Overview

Optimal transport (OT) is designed to find the optimal mapping from a measure on a space to another measure on a different space, which is formalized by the Monge-Kantorovich problem [29]. The discrete formulation can be leveraged to find the most efficient distribution of resources from a set of sources to a set of targets by considering heterogeneous constraints [15, 17]. The traditional framework to solve this problem is centralized but in some cases, a decentralized framework is more advantageous for efficient computation, especially when the transport network gets large [36]. One commonly adopted approach to distribute OT is using alternating direction method of multipliers (ADMM) [5], which will be leveraged in this thesis.

In order to visualize OT's application to resource allocation we employ the traditional mines and factories example. In this example, there is a community with multiple mines and multiple factories that want to take in resources from the mines to manufacture goods. The mine can harvest q amount of resources in a day and the factories can take in p amount of resources to produce goods. Optimal transport will find the optimal way to distribute the resources based on a metric, like utility or cost.

Other examples include, UPS distribution hubs used to transport and deliver packages to major cities, shown in Fig 1.1(a), or the distribution of government funds to fight

Covid-19, depicted in Fig 1.1(b). All of these examples require a fair allocation of resources and use sensitive data to calculate the transport strategy. Hence, it is imperative to investigate how fairness and privacy can be incorporated into the optimal transport algorithm design.



(a) A possible connection scheme of UPS distribution hubs in red to major cities in blue.

(b) A distribution of government funds to hospitals to help fund the fight against the Covid-19 Pandemic.

Figure 1.1: Resource Allocation Examples

1.1 Fairness of Optimal Transport

Under the standard OT paradigm, the resource distribution scheme maximizes the aggregated utilities of all participants in a centralized way, regardless of whether that distribution is fair for the suppliers or receivers [35, 36]. As in the mines and factories example, it may be that the most optimal distribution of the material is to have one factory receive 80% of the material while another receives only 20%. This is not fair to the factory receiving the smaller amount of material, and could lead to that factory shutting down. Thus, it is necessary to incorporate fairness during the transport mechanism design for constrained resource allocation, especially in scenarios that require equity.

To enable a fair allocation of resources, one approach is to include a fairness measure in the objective function of the OT framework. In this way, the resulting transport plan will have a balance between efficiency and fairness. Chapter 3 focuses on the design of distributed OT that considers both fairness and efficiency holistically.

1.2 Security of Optimal Transport

The classic discrete OT framework does not consider that the resource suppliers and receivers could be compromised by an attacker whose goal is to disrupt the efficiency of the resource allocation. This could take the form of a node misrepresenting its parameters, like its utility or amount of resources it can produce or take in. By misrepresenting this information the node can give itself an unfair advantage in the resources allocation plan, which can result in a source not giving out as many resources as it is able to, or a node taking in more resources than it should. The results of the malicious attacks can be disastrous for other nodes in the network.

To this end, our goal is to develop a more robust transport strategy using a game-theoretic framework [4] that captures the interactions between the transport planner and the adversary. Specifically, the planner designs the transport plan that maximizes the social utility by anticipating the compromise of a set of participating nodes by the adversary. In comparison, the attacker's objective is to minimize the aggregated utility of all the nodes under the transport plan. The attacker is stealthy, as it will not modify the node's preference information in an arbitrary manner but considers threshold and magnitude constraints during decision-making. A detailed investigation of the OT security and resiliency under the adversarial environment is the focus of Chapter 4.

1.3 Privacy of Optimal Transport

The previously described distributed algorithms still face threats from an attacker. Specifically, when the nodes need to communicate the computed resource transport preferences with the connected nodes at each update step in the algorithm, the information could be intercepted by an adversary during its transmission over the communication network (e.g., through an eavesdropping attack), after which the attacker can use it to infer the private information at each node (e.g., node's utility parameters used for the design of transport plan).

The privacy concerns of the distributed OT motivate the development of an efficient privacy-preserving mechanism that can protect the nodes' sensitive utility information. To do this, we resort to the powerful differential privacy technique [13, 14]. Specifically, we develop an output variable perturbation-based differentially private distributed OT algorithm, which instead of sharing the authentic transport strategies directly between connected source and target nodes, it perturbs the transport decisions by adding random noise drawn from an appropriate distribution with specified parameters at each step. The proposed distributed algorithm in Chapter 5 prevents a leakage of sensitive information of participants in the network even if the transport strategies negotiated between nodes were captured by an adversary.

1.4 Related Works

Resource allocation has been investigated vastly in various fields with many applications, including communication networks [35], energy systems [3], critical infrastructure [19] and cyber systems [9]. To compute the optimal transport strategy efficiently, a

number of techniques have been developed, such as simultaneous approximation [24], population-based optimization [11], distributed algorithms [26, 36] and linear programming [15]. Additionally, fair resource allocation methods are studied in [2, 10]. Resilient resource allocation methods under adversarial attacks are studied, specifically jamming attacks [16], network topology attacks [31], and data falsification attacks [8]. Differentailly private algorithms are often studied in machine learning. For example, perturbation-based ADMM algorithms were developed to improve privacy in classification learning problems [37, 39]. Differential privacy has also been leveraged to investigate privacy issues in empirical risk minimization [7, 27], support vector machines [40] and deep learning [1]. Additionally, differential privacy has been applied to improve privacy of fog computing [12] and safety of vehicle network [38]. Lastly, a differentially private continuous OT algorithm is developed in [23]. These works lay the foundation for the algorithms developed in the coming chapters.

Previously Published Work: This thesis is based on my publications in the following conferences and journals. The work on the fairness of distributed OT has been published in the IEEE Conference on Information Sciences and Systems (CISS) [20]. The work on security and resiliency of OT will appear in the IEEE Control System Letters (L-CSS) [21]. Lastly, the work on differentially private distributed OT has been submitted to IEEE Global Communications Conference (GLOBECOM) [22].

1.5 Organization of Thesis

This thesis first introduces the network architecture that resources are transported over and the discrete formulation of optimal transport in Chapter 2. Next, the thesis develops a distributed OT algorithm that considers the fairness of the resource allocation in

Chapter 3. The thesis further develops a secure OT scheme under a deceptive adversary that aims to disrupt the resource transport plan in Chapter 4. Lastly, a differentially private distributed OT algorithm to protect the information at the source and target nodes is constructed in Chapter 5. Chapter 6 concludes the thesis.

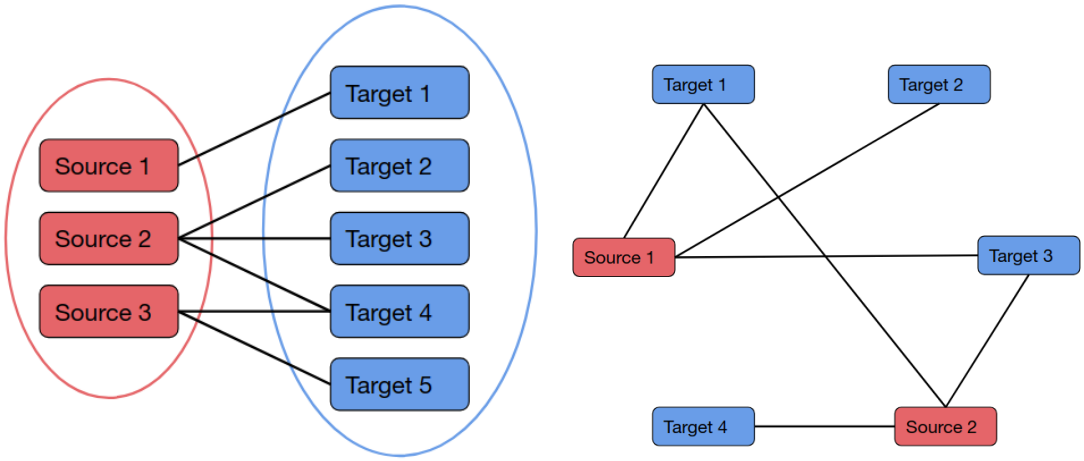
Chapter 2

Background

In order to understand distributed optimal transport, this chapter describes the network architecture that resources are transported over and how a distributed algorithm can be formed from the traditional discrete OT paradigm.

2.1 Network Structure

A transport network is composed of nodes and edges, in the case of resource allocation, the structure can be seen as a bipartite graph where $x \in \mathscr{X} := \{1, ..., |\mathscr{X}|\}$ is a target node taking in resources, and $y \in \mathscr{Y} := \{|\mathscr{X}|+1, ..., |\mathscr{X}|+|\mathscr{Y}|\}$ is a source node supplying resources. Within the network, source nodes are connected to target nodes, although a source node is not required to be connected to all target nodes. For example, in Fig 2.1(a), where there are three source nodes and five target nodes, source 1 is only connected to target 1, source 2 is connected to targets 2, 3 and 4, and source 3 is connected to targets 4 and 5. The set of sources connected to target x is denoted by $\mathscr{Y}_x$, and the set of target nodes connected to source node y is denoted by $\mathscr{X}_y$. Note that $\mathscr{X}_y$, $\forall y$ and $\mathscr{Y}_x$, $\forall x$ are nonempty. Otherwise, the corresponding nodes are isolated in the network and do not play a role in the considered optimal transport strategy design. Additionally, the graphs do not need to have the sources and targets separated as



(a) Separated Groups (b) Non-separated Groups

Figure 2.1: Network Architecture

they are in Fig 2.1(a). The nodes can be intertwined such as the structure in Fig 2.1(b) where the sources and targets are dispersed in a hexagonal shape. Both graphs are bipartite because there are two groups and a connection can only exist between nodes in non-similar groups. The set of feasible transport paths, or edges, connecting targets to sources is denoted by $\{x,y\} \in \mathscr{E}$. The amount of resource that can be transported from a source node or to a target node is bounded from below and above. This is denoted by $\underline{p}_x$ and $\underline{q}_y$ as the lower bounds for the source and target nodes respectively, and $\bar{p}_x$ and $\bar{q}_y$ denotes the upper bounds. The subscripts x and y are used to denote which node the bounds belong to.

2.2 Discrete Optimal Transport

The discrete formulation of optimal transport is useful for data-driven applications specifically for resource allocation, but can also be used for transport between any two finite sets. An example of an optimal distribution can be seen in Fig. 2.2. The traditional

discrete formulation is a discretized version of the primal Monge-Kantorovich problem [15, 28, 32].

$$\begin{aligned} &\max_{\pi_{xy}\geq 0} \sum_{xy} U_{xy}\pi_{xy} \\ \text{s.t.} \quad &\sum_{y\in\mathscr{Y}_x} \pi_{xy} = p \quad \text{and} \quad \sum_{x\in\mathscr{X}_y} \pi_{xy} = q \end{aligned} \tag{2.1}$$

where π_{xy} is the amount of resources allocated by source y for target x, $U_{xy} : \mathbb{R}_+ \to \mathbb{R}$ is the weight of utility between target x and source y, p is the amount of resources that can be taken in by target x, and q is the amount resources that can be given out by source y. This formulation specifically maximizes utility based on the connection between x and y. In order to facilitate the distribution of discrete optimal transport, we offer an equivalent formulation, based on [36], that captures utility from the source and the target nodes'

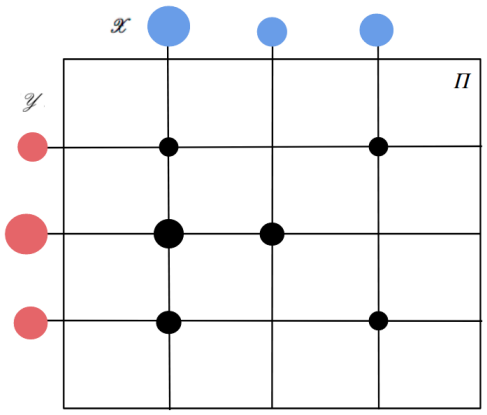


Figure 2.2: The distribution of finite sets $\mathscr{X}$ and $\mathscr{Y}$ after OT assignment. Red indicates a source, blue indicates a target, and black indicates resources transferred from each connection. The size of the circle shows how much resources each node has, can take in or is being transferred at the connection.

perspective. The formulation is as follows:

$$\begin{aligned} \max_{\Pi} \; & \sum_{x\in\mathscr{X}} \sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy}) + \sum_{y\in\mathscr{Y}} \sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy}) \\ \text{s.t.} \quad & \underline{p}_x \le \sum_{y\in\mathscr{Y}_x} \pi_{xy} \le \bar{p}_x, \; \forall x \in \mathscr{X}, \\ & \underline{q}_y \le \sum_{x\in\mathscr{X}_y} \pi_{xy} \le \bar{q}_y, \; \forall y \in \mathscr{Y}, \\ & \pi_{xy} \ge 0, \; \forall \{x,y\} \in \mathscr{E}, \end{aligned} \tag{2.2}$$

where $t_{xy} : \mathbb{R}_+ \to \mathbb{R}$ and $s_{xy} : \mathbb{R}_+ \to \mathbb{R}$ are utility functions for target node x connected to source y and source node y connected to target x, respectively. The method is to maximize the utility of the transport plan $\Pi := \{\pi_{xy}\}_{x\in\mathscr{X}_y, y\in\mathscr{Y}}$ for the given network. Furthermore, $\bar{p}_x \ge \underline{p}_x \ge 0$, $\forall x \in \mathscr{X}$ and $\bar{q}_y \ge \underline{q}_y \ge 0$, $\forall y \in \mathscr{Y}$. The constraints $\underline{p}_x \le \sum_{y\in\mathscr{Y}_x} \pi_{xy} \le \bar{p}_x$ and $\underline{q}_y \le \sum_{x\in\mathscr{X}_y} \pi_{xy} \le \bar{q}_y$ capture the limitations on the amount of requested and transferred resources at the target x and source y, respectively.

The formulation in (2.2) is equivalent to (2.1) because, rather than summing over the connections between targets and sources, $\{x,y\} \in \mathscr{E}$, the objective in (2.2) sums over the connections of each individual node, captured by $\sum_{x\in\mathscr{X}_y}$ and $\sum_{y\in\mathscr{Y}_x}$, and then aggregate those summations with $\sum_{x\in\mathscr{X}}$ and $\sum_{y\in\mathscr{Y}}$ to get an aggregated utility for both the target and source sides respectively. Thus the problem in (2.2) captures the utility from the perspective of the source and target nodes individually rather than the connection between them as in problem (2.1).

In order to solve the maximization problem in (2.2), the following assumption needs to be made.

Assumption 1. *The utility functions t_{xy} and s_{xy} are concave and monotonically increasing, $\forall x \in \mathscr{X}$, $\forall y \in \mathscr{Y}$.*

There are a number of functions of interest that satisfy the properties in Assumption 1. For example, the utility functions t_{xy} and s_{xy} can adopt a linear form, indicating a linear growth of payoff on the amount of transferred and consumed resources. t_{xy} and s_{xy} can also take a logarithmic form on the argument, representing the marginal utility decreases with the amount of transported resources. From this assumption a convex optimization problem is formulated below.

$$\begin{aligned} \min_{\Pi} \sum_{x\in\mathscr{X}} \sum_{y\in\mathscr{Y}_x} -t_{xy}(\pi_{xy}) + \sum_{y\in\mathscr{Y}} \sum_{x\in\mathscr{X}_y} -s_{xy}(\pi_{xy}) \\ \text{s.t.} \quad \underline{p}_x \leq \sum_{y\in\mathscr{Y}_x} \pi_{xy} \leq \bar{p}_x,\ \forall x \in \mathscr{X}, \\ \underline{q}_y \leq \sum_{x\in\mathscr{X}_y} \pi_{xy} \leq \bar{q}_y,\ \forall y \in \mathscr{Y}, \\ \pi_{xy} \geq 0,\ \forall \{x,y\} \in \mathscr{E}, \end{aligned} \tag{2.3}$$

By taking the opposite of the concave utility functions they become convex and monotonically decreasing. Now, the functions are minimized to obtain the optimal value. Doing this allows for the many advantages of convex optimization [6].

In order to facilitate the development of distributed algorithms in the coming chapters, we introduce ancillary variables $\pi_{xy,t}$ and $\pi_{xy,s}$. The subscripts t and s indicate that the corresponding parameters are associated with a target node or source node, respectively. We then set $\pi_{xy} = \pi_{xy,t}$ and $\pi_{xy} = \pi_{xy,s}$, indicating the solutions proposed by the targets and sources are consistent with the ones proposed by the central planner. With

this we can reformulate (2.3) to the following,

$$\begin{aligned}
\min_{\Pi_t \in \mathscr{F}_t, \Pi_s \in \mathscr{F}_s, \Pi} \; & -\sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} s_{xy}(\pi_{xy,s}) \\
\text{s.t.} \quad & \pi_{xy,s} = \pi_{xy}, \ \forall (x,y) \in \mathscr{E}, \\
& \pi_{xy,t} = \pi_{xy}, \ \forall (x,y) \in \mathscr{E}, \\
& \underline{p}_x \leq \sum_{y \in \mathscr{Y}_x} \pi_{xy,t} \leq \bar{p}_x, \ \forall x \in \mathscr{X}, \\
& \underline{q}_y \leq \sum_{x \in \mathscr{X}_y} \pi_{xy,s} \leq \bar{q}_y, \ \forall y \in \mathscr{Y}, \\
& \pi_{xy} \geq 0, \ \forall \{x,y\} \in \mathscr{E},
\end{aligned} \tag{2.4}$$

with the sets defined as,

$$\Pi_t := \{\pi_{xy,t}\}_{x \in \mathscr{X}_y, y \in \mathscr{Y}}, \qquad \Pi_s := \{\pi_{xy,s}\}_{x \in \mathscr{X}, y \in \mathscr{Y}_x}, \tag{2.5}$$

$$\begin{aligned}
\mathscr{F}_t &:= \{\Pi_t | \pi_{xy,t} \geq 0, \underline{p}_x \leq \sum_{y \in \mathscr{Y}_x} \pi_{xy,t} \leq \bar{p}_x, \ (x,y) \in \mathscr{E}\}, \\
\mathscr{F}_s &:= \{\Pi_s | \pi_{xy,s} \geq 0, \underline{q}_y \leq \sum_{x \in \mathscr{X}_y} \pi_{xy,s} \leq \bar{q}_y, \ (x,y) \in \mathscr{E}\}.
\end{aligned} \tag{2.6}$$

This formulation will be considered when developing distributed algorithms with fairness, security and privacy in the following chapters.

Chapter 3

Fair and Efficient Distributed Optimal Transport

A fair and efficient transport scheme can be achieved by incorporating a fairness metric into the discrete OT formulation. In the formulation in (2.2) the central planner devises an optimal transport strategy by maximizing the utility. In practice, some target nodes receive more resources because of the inherent nature of the optimization problem. This efficient resource allocation plan yields a larger objective value. However, it is not fair for some nodes if their requests for resources are ignored. For example, in energy systems, the resilience planing should take into account these generally under considered communities which are hit heavily by natural disasters [3]. Though, from the central planner's perspective, the resilience planning in these areas may not contribute as significantly as other areas to the system's utility by cost-benefit analysis.

In this chapter, we describe how such a metric can be incorporated into the optimal transport paradigm and then distributed using alternating direction method of multipliers (ADMM) to compute the transport plan more efficiently in large-scale networks.

3.1 Problem Formulation

Recall the network structure outlined in section 2.1 where $\mathscr{X}_y$ is the set of target nodes connected to source node y and $\mathscr{Y}_x$ is the set of source nodes connected to target node x. Moreover, recall the discrete formulation in (2.3), where there is no consideration of fairness in the resource allocation objective function. One possible way to achieve a fairer allocation scheme is to introduce a fairness measure to the objective function in the optimal transport framework which admits the following formulation:

$$\begin{aligned}\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy}) + \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy}) \\ + \sum_{x\in\mathscr{X}} \omega_x f_x\Big(\sum_{y\in\mathscr{Y}_x}\pi_{xy}\Big),\end{aligned} \tag{3.1}$$

where $\omega_x \geq 0$ is a weighting constant for fairness, and $f_x : \mathbb{R}_+ \to \mathbb{R}$ is a fairness function. Note that $\sum_{y\in\mathscr{Y}_x}\pi_{xy}$ is the total amount of resources received for target node x. Thus, $f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy})$ quantifies the level of fairness by allocating $\sum_{y\in\mathscr{Y}_x}\pi_{xy}$ resources to each target x. To facilitate a fair transport strategy f_x needs to be chosen strategically. One consideration is that the marginal utility of the fairness term f_x should decrease. Otherwise, it will lead to an unfair distribution of resources, i.e., some target nodes receive most of the resources in the network as the central planner aims to maximize $\sum_{x\in\mathscr{X}}\omega_x f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy})$. We have the following assumption on the properties of the fairness function.

Assumption 2. *The fairness function f_x, $\forall x \in \mathscr{X}$ is concave and monotonically increasing.*

There can be various choices for the fairness function. One possible choice is a

proportional fairness function [2]:

$$f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy}) = \log(\sum_{y\in\mathscr{Y}_x}\pi_{xy}+1),\ \forall x\in\mathscr{X}. \tag{3.2}$$

To this end, the central planner's goal is to devise a fair and efficient transport strategy that maximizes the objective function (3.1) while taking into account the same set of constraints on resources capacity in (2.3).

3.2 Distributed Algorithm for Fair and Efficient Transport Strategy Design

The optimal allocation of resources can be computed in a central manner using discrete OT, but one primal concern is the computational feasibility. It can be computationally expensive to obtain a fair and efficient resource distribution plan when the number of sources and targets becomes enormous, as can be observed in a large-scale network for resource allocation. Therefore, the next objective is to devise a fair and efficient transport strategy from a centralized way to a fully distributed fashion.

3.2.1 Feasibility and Optimality

Before developing the distributed algorithm, we first analyze the feasibility of the formulated optimization problem.

Lemma 1. *It is feasible to find a fair transport plan* Π *if the following conditions are*

satisfied:

$$\sum_{y \in \mathscr{Y}_x} \bar{q}_y \geq \underline{p}_x, \quad \forall x \in \mathscr{X}, \tag{3.3}$$

$$\sum_{y \in \mathscr{Y}} \bar{q}_y \geq \sum_{x \in \mathscr{X}} \underline{p}_x. \tag{3.4}$$

The two inequalities in Lemma 1 have natural interpretations. Inequality (3.3) ensures that all the target nodes' requests can be fulfilled, while (3.4) indicates that the total demand of resources is less than the total supply.

We next characterize the existence of optimal solution to the formulated problem.

Lemma 2. *Under Assumptions 1 and 2, and the inequalities* (3.3) *and* (3.4)*, there exists a fair and efficient transport strategy that maximizes the objective* (3.1) *while satisfying the constraints in* (2.2)*.*

The existence of the optimal solution is guaranteed by the concavity of t_{xy}, s_{xy} and f_x, as well as the feasibility of the problem resulting from (3.3) and (3.4). In order to formulate the problem, the first step is to rewrite the optimization problem from (2.4), introducing the fairness constraint. To this end, the reformulated optimal transport prob-

lem under fairness consideration is presented as follows:

$$
\begin{aligned}
\min_{\Pi_t \in \mathscr{F}_t, \Pi_s \in \mathscr{F}_s, \Pi} \; & -\sum_{x\in\mathscr{X}} \sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \sum_{y\in\mathscr{Y}} \sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) \\
& -\sum_{x\in\mathscr{X}} \omega_x f_x\Big(\sum_{y\in\mathscr{Y}_x} \pi_{xy,t}\Big) \\
\text{s.t.} \quad & \pi_{xy,s} = \pi_{xy},\ \forall (x,y) \in \mathscr{E}, \\
& \pi_{xy,t} = \pi_{xy},\ \forall (x,y) \in \mathscr{E}, \\
& \underline{p}_x \le \sum_{y\in\mathscr{Y}_x} \pi_{xy,t} \le \bar{p}_x,\ \forall x \in \mathscr{X}, \\
& \underline{q}_y \le \sum_{x\in\mathscr{X}_y} \pi_{xy,s} \le \bar{q}_y,\ \forall y \in \mathscr{Y}, \\
& \pi_{xy} \ge 0,\ \forall \{x,y\} \in \mathscr{E},
\end{aligned}
\tag{3.5}
$$

with the sets defined in (2.5) and (2.6). Note that due to the constraints, the optimal solutions, Π_t, Π_s and Π of (3.5) are all equivalent.

3.2.2 Distributed Algorithm

The next focus is to develop a distributed algorithm to solve the problem (3.5). Let $\alpha_{xy,s}$ and $\alpha_{xy,t}$ be the Lagrangian multipliers associated with the constraint $\pi_{xy,s} = \pi_{xy}$ and $\pi_{xy,t} = \pi_{xy}$, respectively. The Lagrangian then facilitates the application of ADMM in the distributed algorithm design. Specifically, the Lagrangian associated with the

optimization problem (3.5) can then be written as follows:

$$
\begin{aligned}
L(\Pi_t, \Pi_s, \Pi, \alpha_{xy,t}, \alpha_{xy,s}) = & -\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) \\
& - \sum_{x\in\mathscr{X}} \omega_x f_x(\sum_{y\in\mathscr{Y}_x} \pi_{xy,t}) + \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x} \alpha_{xy,t}(\pi_{xy,t} - \pi_{xy}) \\
& + \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} \alpha_{xy,s}(\pi_{xy} - \pi_{xy,s}) + \frac{\eta}{2}\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy})^2 \\
& + \frac{\eta}{2}\sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} (\pi_{xy} - \pi_{xy,s})^2,
\end{aligned} \tag{3.6}
$$

where $\eta > 0$ is a positive scalar constant controlling the convergence rate in the algorithm. In (3.6), the last two terms $\frac{\eta}{2}\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x}(\pi_{xy,t} - \pi_{xy})^2$ and $\frac{\eta}{2}\sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}(\pi_{xy} - \pi_{xy,s})^2$, act as penalization and are quadratic. Hence, the Lagrangian function L is strictly convex, ensuring the existence of a unique optimal solution.

Continuing with the steps of ADMM, we use the Lagrangian to develop the distributed algorithm and is presented in the following proposition.

Proposition 1. *The iterative steps of ADMM to* (3.5) *are summarized as follows:*

$$
\begin{aligned}
\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} & -\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \omega_x f_x(\sum_{y\in\mathscr{Y}_x} \pi_{xy,t}) \\
& + \sum_{y\in\mathscr{Y}_x} \alpha_{xy,t}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy}(k))^2,
\end{aligned} \tag{3.7}
$$

$$
\begin{aligned}
\Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} & -\sum_{x\in\mathscr{X}_y} (s_{xy}(\pi_{xy,s}) - c_{xy}(\pi_{xy,s})) \\
& - \sum_{x\in\mathscr{X}_y} \alpha_{xy,s}(k)\pi_{xy,s} + \frac{\eta}{2}\sum_{x\in\mathscr{X}_y} (\pi_{xy}(k) - \pi_{xy,s})^2,
\end{aligned} \tag{3.8}
$$

$$\pi_{xy}(k+1) = \arg\min_{\pi_{xy}} -\alpha_{xy,t}(k)\pi_{xy} + \alpha_{xy,s}(k)\pi_{xy} + \frac{\eta}{2}(\pi_{xy,t}(k+1) - \pi_{xy})^2 + \frac{\eta}{2}(\pi_{xy} - \pi_{xy,s}(k+1))^2, \tag{3.9}$$

$$\alpha_{xy,t}(k+1) = \alpha_{xy,t}(k) + \eta(\pi_{xy,t}(k+1) - \pi_{xy}(k+1))^2, \tag{3.10}$$

$$\alpha_{xy,s}(k+1) = \alpha_{xy,s}(k) + \eta(\pi_{xy}(k+1) - \pi_{xy,s}(k+1))^2, \tag{3.11}$$

where $\Pi_{\tilde{x},t} := \{\pi_{xy,t}\}_{y\in\mathscr{Y}_x, x=\tilde{x}}$ *represents the solution at target node* $\tilde{x} \in \mathscr{X}$*, and* $\Pi_{\tilde{y},s} := \{\pi_{xy,s}\}_{x\in\mathscr{X}_y, y=\tilde{y}}$ *represents the proposed solution at source node* $\tilde{y} \in \mathscr{Y}$*. In addition,* $\mathscr{F}_{x,t} := \{\Pi_{x,t} | \pi_{xy,t} \geq 0, y \in \mathscr{Y}_x, \underline{p}_x \leq \sum_{y\in\mathscr{Y}_x} \pi_{xy,t} \leq \bar{p}_x\}$*, and* $\mathscr{F}_{y,s} := \{\Pi_{y,s} | \pi_{xy,s} \geq 0, x \in \mathscr{X}_y, \underline{q}_y \leq \sum_{x\in\mathscr{X}_y} \pi_{xy,s} \leq \bar{q}_y\}$.

Proof. See Appendix A. □

Iterations (3.7)-(3.11) can be simplified down to four steps, and the results are summarized below.

Proposition 2. *The iterations* (3.7)-(3.11) *can be simplified as follows:*

$$\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \omega_x f_x\Big(\sum_{y\in\mathscr{Y}_x} \pi_{xy,t}\Big) + \sum_{y\in\mathscr{Y}_x} \alpha_{xy}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy}(k))^2, \tag{3.12}$$

$$\Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} -\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) - \sum_{x\in\mathscr{X}_y} \alpha_{xy}(k)\pi_{xy,s} + \frac{\eta}{2}\sum_{x\in\mathscr{X}_y} (\pi_{xy}(k) - \pi_{xy,s})^2, \tag{3.13}$$

$$\pi_{xy}(k+1) = \frac{1}{2}(\pi_{xy,t}(k+1) + \pi_{xy,s}(k+1)), \tag{3.14}$$

$$\alpha_{xy}(k+1) = \alpha_{xy}(k) + \frac{\eta}{2}(\pi_{xy,t}(k+1) - \pi_{xy,s}(k+1)). \tag{3.15}$$

Proof. See Appendix B □

Iterations (3.7)-(3.11) can be iterated through to obtain a fair and efficient resource transport strategy whose convergence is guaranteed [5]. The first step, (3.12) updates the transport plan on the target nodes. Note that the fairness is explicitly considered during the solution updates, in (3.8). Step (3.13) updates the transport plan on the source nodes. Next, step (3.14) is the negotiation between the target and source nodes. Finally, (3.15) updates the dual variable, α_{xy}. For convenience, we summarize the iterations from Proposition 2 in Algorithm 1.

Algorithm 1 Fair Distributed Algorithm

1: **while** $\Pi_{x,t}$ and $\Pi_{y,s}$ not converging **do**
2: Compute $\Pi_{x,t}(k+1)$ using (3.12), for all $x \in \mathscr{X}_y$
3: Compute $\Pi_{y,s}(k+1)$ using (3.13), for all $y \in \mathscr{Y}_x$
4: Compute $\pi_{xy}(k+1)$ using (3.14), for all $\{x,y\} \in \mathscr{E}$
5: Compute $\alpha_{xy}(k+1)$ using (3.15), for all $\{x,y\} \in \mathscr{E}$
6: **end while**
7: **return** $\pi_{xy}(k+1)$, for all $\{x,y\} \in \mathscr{E}$

3.3 Discussions on the Fair Distributed Algorithm

In this section, we discuss several crucial aspects of the proposed distributed algorithm for fair and efficient resource allocation mechanisms including the effects and implementation of fairness.

3.3.1 Fairness and Efficiency Trade-off

The fairness of the transport scheme is ensured during the updates of solutions. As shown in (3.12), the level of fairness is regulated by the parameter ω_x, $x \in \mathscr{X}$.

Specifically, ω_x trades off between the efficiency and fairness of the transport strategy. With a larger ω_x, the fairness term has a more significant impact on the solution, yielding a fairer resource allocation plan. For every target x, it maximizes $f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy,t})$ at each step. The concavity of f_x guarantees that it is impossible for a single target in the network to receive all the resources. Together with the penalization terms $\sum_{y\in\mathscr{Y}_x}\alpha_{xy}(k)\pi_{xy,t}+\frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}(\pi_{xy,t}-\pi_{xy}(k))^2$, it also ensures that the request for resources from each target x will not be arbitrarily large.

Another interesting observation is that after both the target node x and source node y proposing their strategy (based on (3.12) and (3.13)), the central manager will mediate both requests by taking an average of the fair solution $\pi_{xy,t}$ and the efficient solution $\pi_{xy,s}$ as shown in (3.14). Hence, the final solution yielded by Algorithm 1 will be both fair and efficient.

3.3.2 Implementation of Fairness

In the reformulated problem (3.5), we associated the fairness function, f_x, $\forall x\in\mathscr{X}$, with the corresponding target node. This leads to natural interpretations that when proposing the transport strategy, each target needs to be aware of the fairness of the resource allocation over networks. In a resource distribution market, the supplier (source node) may not care where its resources are finally allocated. However, a target cares whether it gets more or less resources than another target. For example, if a large company is distributing resources to customers, the company (the source) does not care where their product goes as long as they sell the product, while consumers care if only a few people are able to buy the product. This observation is consistent with the iteration steps (3.12) and (3.13), where each target $x\in\mathscr{X}$, aims to maximize the fairness term $\omega_x f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy,t})$, while each source $y\in\mathscr{Y}$, merely maximize its own utility.

Note that during the problem reformulation, the fairness term could also be applied to the source (supplier side). Then, the objective function in (3.5) becomes the following: $-\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) - \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) - \sum_{x\in\mathscr{X}} \omega_x f_x(\sum_{y\in\mathscr{Y}_x}\pi_{xy,s})$. A distributed algorithm can be designed to solve this reformulated problem by using similar techniques as those in Section 3.2. When the fairness term is associated with the source side, it means that all the suppliers need to inherently consider fairness when distributing resources. It also can be interpreted that if a source enters the market, it needs to comply with the agreed fairness rules in the resource allocations.

3.4 Case Studies

In this section, we corroborate the algorithm for distributed optimal transport with the fairness consideration with numerical case studies. We consider a scenario with five target nodes and two source nodes and a transport network structure connecting all source nodes to both target nodes. Figure 2.1(a) shows the network structure of resource transportation. We define the upper bound, $\bar{p}_x$ for target node $x \in \mathscr{X} = \{1,2\}$ and $\bar{q}_y$ for source nodes $y \in \mathscr{Y} = \{3,4,5,6,7\}$. The lower bounds, $\underline{q}_y$ and $\underline{p}_x$ are 0 for all nodes. There is utility associated with each for both the target and source nodes. We adopt linear utility functions as follows: $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$, and $s_{xy}(\pi_{xy}) = \sigma_{xy}\pi_{xy} - \zeta_{xy}\pi_{xy}$,

$\forall\{x,y\} \in \mathscr{E}$. The corresponding parameters are selected as:

$$[\delta_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 1 & 3 & 1 & 3 & 2 \\ 2 & 2 & 4 & 1 & 2 \end{bmatrix}$$

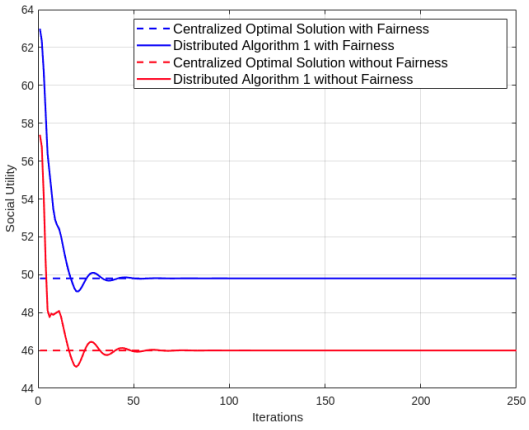
$$[\sigma_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 3 & 3 & 5 & 4 & 5 \\ 4 & 3 & 5 & 3 & 6 \end{bmatrix}$$

$$[\zeta_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 2 & 1 & 3 & 1 & 2 \end{bmatrix}$$

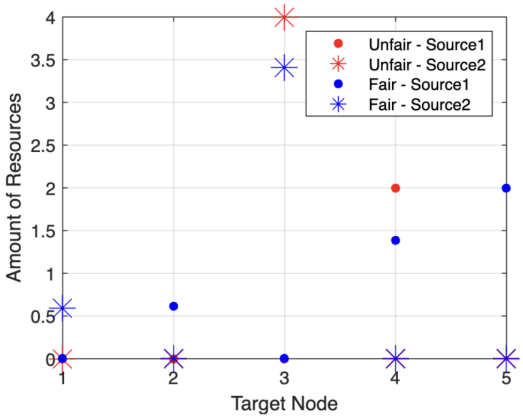
We consider proportional fairness in the resource allocation, i.e., the fairness function admits the form shown in (3.2).

3.4.1 Fair and Distributed Resource Allocation

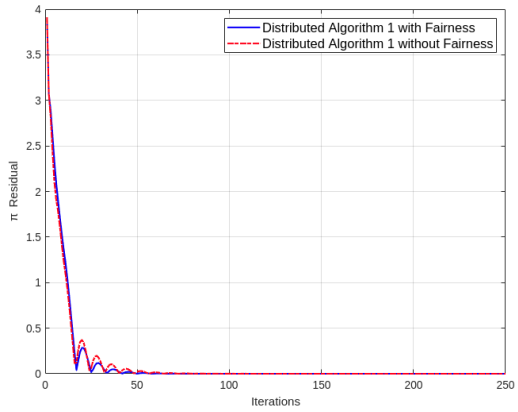
We first show the effectiveness of Algorithm 1. Specifically, we compare the optimal transport strategies with and without fairness considerations using Algorithm 1. For the algorithm with fairness, we set the weighting factor $\omega_x = 3$. We focus on comparing the algorithms induced social utility. The social utility is the aggregate of the payoffs of the sources and targets and the benefits of fairness in the resource allocation. The results are shown in Fig. 3.1. Fig. 3.1(a) shows the distributed algorithm (both with and without fairness consideration) converges to the corresponding centralized optimal solution π^o_{xy} (i.e., problem (3.5) is solved directly). Note that in Fig. 3.1(a) that the algorithm with fairness converges to a higher social utility. The increase in the social utility is due to the addition of fairness when designing the resource transport scheme. It also shows that the fairness has little effect on the convergence of the algorithm. In Fig. 3.1(b), the fairer transport scheme is compared to one without fairness. Without fairness most of



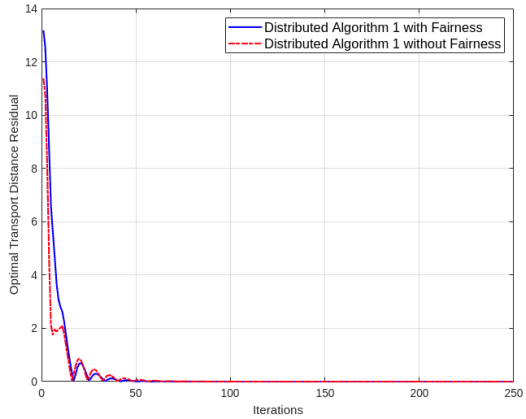
(a) Social utility under transport strategies



(b) Fair vs. unfair allocation scheme



(c) $\sqrt{\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x}(\pi_{xy}(k)-\pi^o_{xy})^2}$



(d) Distance residual

Figure 3.1: Impact of fairness consideration on the transport strategy design using Algorithm 1.

the resources go to target nodes 3, 4 and 5. When fairness is considered all of the targets get resources and targets 3, 4 and 5 still get the most resources. Fig. 3.1(c) and 3.1(d) shows the residual of transport strategy. The residual measures the difference between the strategy at the current update and the centralized optimal solution. It also highlights that the residual goes to 0 around $k = 50$, which demonstrates the effectiveness and convergence of the designed distributed algorithm.

3.5 Conclusion

This chapter investigated fair and efficient transport of a limited amount of resources in a network of participants with various preferences. The designed distributed algorithm can successfully yield the identical transport plan designed under the centralized manner, making our algorithm applicable to large-scale networks. The fairness is explicitly promoted in the algorithm, through bargaining and negotiations between each pair of resource supplier (source) and resource receiver (target). Throughout the negotiation steps, the sources maximize their revenue but need to consider the fairness requests. Similarly, the targets optimize the fairness but should take into account the efficiency of resource allocation as well. The negotiation/algorithm terminates when the two parties reach a consensus.

Chapter 4

Secure and Resilient Distributed Discrete Optimal Transport

In this chapter, we formulate a discrete distributed optimal transport algorithm while considering a deceptive adversary. An adversary could modify a node's preference information or alter threshold and magnitude constraints during the decision-making process. To counteract the adversarial manipulation, this chapter develops a game-theoretic approach for secure and resilient OT design.

4.1 Problem Formulation

The formulation of adversarial optimal transport is based on (2.3) while incorporating the attacker's behavior.

4.1.1 Adversarial Optimal Transport

The attacker's goal is to minimize the aggregated transport utility by compromising the preference coefficients in the target's utility functions (which can happen at the information exchange stage). Specifically, the parameters in the utility function t_{xy} are com-

promised, for $x \in \mathscr{X}_a$, $y \in \mathscr{Y}_x$, where $\mathscr{X}_a$ denotes a subset of adversarial receiver nodes. Then, $\mathscr{X}_o := \mathscr{X} \setminus \mathscr{X}_a$ is the set of uncompromised targets. We denote by $\tilde{t}_{xy,\xi_{xy}}$ the modified utility under the attack, where ξ_{xy} represents the magnitude of the adversarial modifications on the corresponding parameters. For example, when the utility function admits a linear form as $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$, where $\delta_{xy} > 0$ is a utility parameter, the compromised utility form under the deception attack becomes $\tilde{t}_{xy,\xi_{xy}}(\pi_{xy}) = (\delta_{xy} + \xi_{xy})\pi_{xy}$. Additionally, if t_{xy} takes a form of $t_{xy}(\pi_{xy}) = \delta_{xy}\min(\zeta_x, \pi_{xy})$, where ζ_x denotes a threshold after which the benefit of consuming more resources for target x does not increase, the compromised utility form can be constructed as $\tilde{t}_{xy,\xi_{xy}}(\pi_{xy}) = (\delta_{xy} + \xi_{xy,1})\min\{\zeta_x + \xi_{xy,2}, \pi_{xy}\}$. As another example, when t_{xy} takes a form of $t_{xy}(\pi_{xy}) = \delta_{xy}\min(\zeta_{xy}, \pi_{xy})$, where ζ_{xy} denotes a threshold after which the benefit of consuming more resources for target x from source y does not increase, the compromised utility form can be constructed as $\tilde{t}_{xy,\xi_{xy}}(\pi_{xy}) = (\delta_{xy} + \xi_{xy,1})\min\{\zeta_{xy} + \xi_{xy,2}, \pi_{xy}\}$. In this scenario, the attacker's action includes both $\xi_{xy,1}$ and $\xi_{xy,2}$, $\forall x \in \mathscr{X}_a$, $y \in \mathscr{Y}_x$. For a general scenario, we denote by $\Xi := \{\xi_{xy}\}_{x\in\mathscr{X}_a, y\in\mathscr{Y}_x}$ the attacker's deceptive strategy. Then, the adversarial optimal transport can be formulated as follows.

$$
\begin{aligned}
\max_{\Pi}\min_{\Xi} \quad & \sum_{x\in\mathscr{X}_o}\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy}) + \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy}) \\
& + \sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x} \tilde{t}_{xy,\xi_{xy}}(\pi_{xy}) + \sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x} l(\xi_{xy}) \\
\text{s.t.} \quad & \underline{p}_x \le \sum_{y\in\mathscr{Y}_x} \pi_{xy} \le \bar{p}_x,\ \forall x \in \mathscr{X}, \\
& \underline{q}_y \le \sum_{x\in\mathscr{X}_y} \pi_{xy} \le \bar{q}_y,\ \forall y \in \mathscr{Y}, \\
& \pi_{xy} \ge 0,\ \forall \{x,y\} \in \mathscr{E}, \\
& \boldsymbol{\xi}_x \in \mathscr{A}_x,\ \forall x \in \mathscr{X}_a,
\end{aligned}
\tag{4.1}
$$

where $\boldsymbol{\xi}_x := [\xi_{x1}, \xi_{x2}, ..., \xi_{x|\mathscr{Y}_x|}]$, for $x \in \mathscr{X}_a$; and $\mathscr{A}_x$ is the attacker's feasible action set on the target node $x \in \mathscr{X}_a$ and $l : \mathbb{R} \to \mathbb{R}_+$ is a function capturing the cost of the attack.

Remark: The solution to the adversarial OT problem is related to the robust OT design. Robust OT also admits a minimax formulation but its goal is to find an optimal solution in the presence of structural and known uncertainties. Comparatively, in the adversarial OT, such uncertainty is replaced by strategic attacks, and the designed transport plan should be resistant to adversarial manipulations.

4.2 Adversarial Optimal Transport under Linear Utilities

In this section, we consider utility functions admitting a linear form for both the sender and receiver. Specifically, $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$ and $s_{xy}(\pi_{xy}) = \gamma_{xy}\pi_{xy}$, where $\delta_{xy}, \gamma_{xy} \in \mathbb{R}_+$. To design the optimal transport plan, the transport planner needs to know the utility parameters including δ_{xy}, γ_{xy}, $\forall x \in \mathscr{X}, y \in \mathscr{Y}_x$. Thus, the source and target nodes need to report their parameters and one way to achieve this is through communications. The wireless channel enabling the communication is vulnerable to cyber attacks. The attacker can disrupt the communication by various techniques, such as jamming and distributed denial of service attacks. Therefore, it is imperative for the central planner to develop resilient transport strategies under the adversarial environment. In the considered scenario, we assume that the attacker is able to compromise a subset of receiver nodes in the network, denoted by $\mathscr{X}_a$. One interpretation is that the nodes in $\mathscr{X}_a$ do not have a secure communication protocol with the central planner. In comparison, the nodes in the set $\mathscr{X}_o = \mathscr{X} \setminus \mathscr{X}_a$ are able to set up high-confidence communication channels and hence are secure from adversarial attacks.

The attacker compromises the sensitive data δ_{xy}, $x \in \mathscr{X}_a, y \in \mathscr{Y}_x$, reported by the vulnerable target nodes and stealthily modifies them to new values aiming to decrease the social utility of the resource allocation. The adversarial disruption can be regarded as a data poisoning attack, under which the data point δ_{xy} is changed to $\tilde{\delta}_{xy} := \delta_{xy} + \xi_{xy}$, for $x \in \mathscr{X}_a, y \in \mathscr{Y}_x$. Here, ξ_{xy} denotes the action of the attacker, representing the magnitude of data modification to the particular data point δ_{xy}. For convenience, we follow the notations in (4.1), where Ξ denotes the attacker's malicious manipulations on the data points and $\boldsymbol{\xi}_x$ is the attackers action on the target node $x \in \mathscr{X}_a$.

To this end, the adversarial OT can be formulated in the following max-min format:

$$
\begin{aligned}
\max_{\Pi} \min_{\Xi} U(\Pi, \Xi) = & \sum_{x \in \mathscr{X}_o} \sum_{y \in \mathscr{Y}_x} \delta_{xy} \pi_{xy} + \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} \gamma_{xy} \pi_{xy} \\
& + \sum_{x \in \mathscr{X}_a} \sum_{y \in \mathscr{Y}_x} (\delta_{xy} + \xi_{xy}) \pi_{xy} + c_a \sum_{x \in \mathscr{X}_a} \|\boldsymbol{\xi}_x\|_1 \\
\text{s.t.} \quad & \underline{p}_x \leq \sum_{y \in \mathscr{Y}_x} \pi_{xy} \leq \bar{p}_x, \ \forall x \in \mathscr{X}, \\
& \underline{q}_y \leq \sum_{x \in \mathscr{X}_y} \pi_{xy} \leq \bar{q}_y, \ \forall y \in \mathscr{Y}, \\
& \pi_{xy} \geq 0, \ \forall \{x, y\} \in \mathscr{E}, \\
& \boldsymbol{\xi}_x \in \mathscr{A}_x, \ \forall x \in \mathscr{X}_a,
\end{aligned}
\tag{4.2}
$$

where $c_a \in \mathbb{R}_+$ is a non-negative cost coefficient and $\mathscr{A}_x$ is the feasible action set of the attacker on target node x, $x \in \mathscr{X}_a$. U is the objective value under strategies Π and Ξ. The term $c_a \sum_{x \in \mathscr{X}_a} \|\boldsymbol{\xi}_x\|_1$ captures the cost of the attack. The sparsity induced by the l_1 norm is a convex approximation of the l_0 norm [5, Chapter 6] and indicates that the attacker has constraints on the number of compromise of utility parameters at a particular node $x \in \mathscr{X}_a$. The attacker is a minimizer of (4.2) as its goal is to minimize the aggregated transport utility reflected by the first three terms in the objective function U while using

the least costly attack scheme captured by the last term in U.

If the attacker modifies all the data parameters significantly, it is easy for the planner to detect such adversarial perturbations. Also, the data $\tilde{\delta}_{xy}$ after compromise should still be non-negative, otherwise, the deception can be identified straightforwardly. Thus, the action set $\mathscr{A}_x$ needs to be carefully modeled to capture the attacker's deceptive behavior. One form of $\mathscr{A}_x$ can be chosen as follows:

$$\mathscr{A}_x = \{\boldsymbol{\xi}_x | \|\boldsymbol{\xi}_x\|_2^2 \leq \kappa_x, \boldsymbol{\xi}_x + \boldsymbol{\delta}_x \geq \mathbf{0}\},\ x \in \mathscr{X}_a, \tag{4.3}$$

where $\kappa_x \in \mathbb{R}_+$ denotes the upper limit of the standard norm of adversarial modifications at the target node $x \in \mathscr{X}_a$ by the attacker; $\boldsymbol{\delta}_x := [\delta_{x1}; \delta_{x2}; ...; \delta_{x|\mathscr{D}_x|}]$; and $\mathbf{0}$ is a zero vector with appropriate dimension.

Problem (4.2) can be seen as a two-person zero-sum game denoted by G, where the transport planner is a maximizer and the attacker is a minimizer. The solution to the game G is characterized by the Nash equilibrium which predicts the outcome of the optimal transport strategy under adversarial environment. The formal definition of the Nash equilibrium strategy [4] is presented as follows.

Definition 1 (Nash Equilibrium)**.** *The strategy pair* $\{\Pi^*, \Xi^*\}$ *is a saddle-point Nash equilibrium of game G if*

$$U(\Pi, \Xi^*) \leq U(\Pi^*, \Xi^*) \leq U(\Pi^*, \Xi),\ \forall\, \Pi, \Xi \tag{4.4}$$

where U is the objective function in (4.2)*.*

Solving game G requires addressing the formulated max-min problem (4.2). Specifically, both the central planner and the attacker need to compute their solutions holis-

tically. This centralized computation paradigm does not scale well as the number of nodes in the transport network becomes large. Furthermore, to compute the solution Π, the central planner is required to have a complete information on the transport network, including the sensitive parameters of all participants' preferences. Thus, it is imperative to design a computationally efficient mechanism to solve game G. Our subsequent goal is to develop a distributed algorithm to compute the equilibrium transport strategy which also preserves the privacy of the participants to some extent.

4.3 Analysis and Distributed Algorithm

This section aims to design a holistic and fully distributed algorithm to compute the optimal strategies of the attacker and the participants in the transport network.

4.3.1 Equivalence between Max-Min and Minimax Problems

Before designing the algorithm, we prove that the formulated max-min problem (4.2) is equivalent to its minimax counterpart and hence show the existence of Nash equilibrium to game G. Specifically, the following results are shown.

Proposition 3. *The max-min problem* (4.2) *yields the same solution as its minimax counterpart, i.e.,* $\min_{\Xi} \max_{\Pi} U(\Pi, \Xi)$ *subject to the same set of the constraints as in* (4.2). *Thus, there exists saddle point Nash equilibrium to game G. However, such equilibrium is not necessarily unique.*

Proof. The equivalence between max-min and minimax problems directly follows from the von Neumann's minimax theorem [25]. As the objective function U is not strictly concave in Π and not strictly convex in Ξ, the Nash equilibrium is not necessarily unique [4, Chapter 4]. □

Note that Proposition 3 facilitates a convenient design of efficient mechanisms called best-response dynamics in finding the equilibrium strategies. We will describe this approach in detail in the ensuing sections.

4.3.2 Distributed Updates on the Deception Strategy

The attacker deceives the transport planner by compromising δ_{xy}, $x \in \mathscr{X}_a, y \in \mathscr{Y}_x$, strategically. As the attacker's goal is to minimize U, a smaller $\tilde{\delta}_{xy}$ (hence a smaller δ_{xy}) will decrease the utility at the corresponding target node as indicated by the term $\sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}(\delta_{xy}+\xi_{xy})\pi_{xy}$. However, simply modifying the values of all δ_{xy}, $\forall x \in \mathscr{X}_a, y \in \mathscr{Y}_x$, to their minimum does not guarantee to minimize U. One reason is that the transport strategy will be changed under the attack. Though the value of the term $\sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}(\delta_{xy}+\xi_{xy})\pi_{xy}$ decreases, the other terms such as $\sum_{x\in\mathscr{X}_o}\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy}$ and $\sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy}$ may increase under the attack. Thus, the attacker's deceptive strategy is nontrivial to devise.

In the following, we describe how to leverage best-response dynamics to compute the strategy. Specifically, the attacker updates its decision Ξ by fixing the transport planner's strategy $\Pi' = \{\pi'_{xy}\}_{x\in\mathscr{X}_y, y\in\mathscr{Y}}$. In this regard, the first two terms in the objective function $U(\Pi,\Xi)$ and the first three constraints in (4.2) can be safely ignored as they are irrelevant with the attacker's deceptive strategy design. Thus, the attacker solves the following optimization problem:

$$\begin{aligned} \min_{\Xi} \quad & \sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}\xi_{xy}\pi'_{xy} + c_a\sum_{x\in\mathscr{X}_a}\|\boldsymbol{\xi}_x\|_1 \\ \text{s.t.} \quad & \boldsymbol{\xi}_x \in \mathscr{A}_x,\ \forall x \in \mathscr{X}_a. \end{aligned} \tag{4.5}$$

The attacker can design the optimal deceptive strategy Ξ^* in a distributed fashion. First,

we observe that the cost function in (4.5) is decoupled across vulnerable target nodes. Then, the optimal $\boldsymbol{\xi}_x^*$, $\forall x \in \mathscr{X}_a$, can be obtained separately. Solving (4.5) is thus equivalent to addressing $|\mathscr{X}_a|$ sub-problems as follows, for $x \in \mathscr{X}_a$,

$$\begin{aligned} \min_{\boldsymbol{\xi}_x} \quad & \sum_{y \in \mathscr{Y}_x} \xi_{xy} \pi'_{xy} + c_a \|\boldsymbol{\xi}_x\|_1 \\ \text{s.t.} \quad & \boldsymbol{\xi}_x \in \mathscr{A}_x. \end{aligned} \tag{4.6}$$

(4.6) can be reformulated as the following, for $x \in \mathscr{X}_a$:

$$\begin{aligned} \min_{\boldsymbol{\xi}_x, \boldsymbol{\chi}_x} \quad & \sum_{y \in \mathscr{Y}_x} \xi_{xy} \pi'_{xy} + \mathbf{1}^\mathsf{T} \boldsymbol{\chi}_x \\ \text{s.t.} \quad & \boldsymbol{\xi}_x \in \mathscr{A}_x, \\ & c_a \boldsymbol{\xi}_x \leq \boldsymbol{\chi}_x, \\ & c_a \boldsymbol{\xi}_x \geq -\boldsymbol{\chi}_x, \end{aligned} \tag{4.7}$$

where $\mathbf{1}$ is a vector of appropriate dimension with all ones; T denotes the transpose operator; and $\boldsymbol{\chi}_x$ is an auxiliary $|\mathscr{Y}_x|$-dimensional decision variable. Note that the objective function in (4.7) is linear and the constraints are convex, and thus (4.7) can be solved efficiently.

Equivalence between problems (4.6) *and* (4.7): First, we can rewrite $c_a\|\boldsymbol{\xi}_x\|_1$ as the following: $\sum_{i=1}^{|\boldsymbol{\xi}_x|} \text{abs}(c_a \boldsymbol{\xi}_{x,i})$, where $\boldsymbol{\xi}_{x,i}$ is the i-th element of $\boldsymbol{\xi}_x$ and $\text{abs}(\cdot)$ denotes an operator of taking the absolute value. Thus, the objective function of (7) can be recast as $\sum_{y \in \mathscr{Y}_x} \xi_{xy} \pi'_{xy} + \sum_{i=1}^{|\boldsymbol{\xi}_x|} \text{abs}(c_a \boldsymbol{\xi}_{x,i})$. We then introduce an auxiliary variable $\boldsymbol{\chi}_x$ with a same dimension as $\boldsymbol{\xi}_x$ that satisfies the condition $\text{abs}(c_a \boldsymbol{\xi}_{x,i}) \leq \boldsymbol{\chi}_{x,i}$, $\forall i$. Then the optimization

problem

$$\min_{\boldsymbol{\xi}_x} \sum_{y\in\mathscr{Y}_x} \xi_{xy}\pi'_{xy} + \sum_{i=1}^{|\boldsymbol{\xi}_x|} \text{abs}(c_a\boldsymbol{\xi}_{x,i})$$
$$\text{s.t.} \quad \boldsymbol{\xi}_x \in \mathscr{A}_x,$$

can be reformulated as

$$\min_{\boldsymbol{\xi}_x,\boldsymbol{\chi}_x} \sum_{y\in\mathscr{Y}_x} \xi_{xy}\pi'_{xy} + \sum_{i=1}^{|\boldsymbol{\chi}_x|} \boldsymbol{\chi}_{x,i}$$
$$\text{s.t.} \quad \boldsymbol{\xi}_x \in \mathscr{A}_x,$$
$$\text{abs}(c_a\boldsymbol{\xi}_{x,i}) \leq \boldsymbol{\chi}_{x,i},\ \forall i = 1,...,|\boldsymbol{\chi}_x|.$$

Note that $\sum_{i=1}^{|\boldsymbol{\chi}_x|} \boldsymbol{\chi}_{x,i}$ is equivalent to $\mathbf{1}^\top\boldsymbol{\chi}_x$. In addition, $\text{abs}(c_a\boldsymbol{\xi}_{x,i}) \leq \boldsymbol{\chi}_{x,i}$ can be written as $-\boldsymbol{\chi}_{x,i} \leq c_a\boldsymbol{\xi}_{x,i} \leq \boldsymbol{\chi}_{x,i}, \forall i$. Putting it in a vector form yields $-\boldsymbol{\chi}_x \leq c_a\boldsymbol{\xi}_x \leq \boldsymbol{\chi}_x$. Thus, we obtain the formulation of (4.7).

4.3.3 Distributed Updates on the Transport Strategy

Under the best-response mechanism, the transport planner determines the transport strategy by regarding the deceptive strategy $\Xi' = \{\xi'_{xy}\}_{x\in\mathscr{X}_a, y\in\mathscr{Y}_x}$ as fixed. Thus, the planner can omit the last term in the objective function $U(\Pi,\Xi)$ and the last constraint

in (4.2) when making the decision. The planner's problem can be formulated as follows.

$$\begin{aligned}
\max_{\Pi} \; & \sum_{x\in\mathscr{X}_o}\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy} + \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy} \\
& + \sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}(\delta_{xy}+\xi'_{xy})\pi_{xy} \\
\text{s.t.} \quad & \underline{p}_x \le \sum_{y\in\mathscr{Y}_x}\pi_{xy} \le \bar{p}_x, \; \forall x \in \mathscr{X}, \\
& \underline{q}_y \le \sum_{x\in\mathscr{X}_y}\pi_{xy} \le \bar{q}_y, \; \forall y \in \mathscr{Y}, \\
& \pi_{xy} \ge 0, \; \forall \{x,y\} \in \mathscr{E}.
\end{aligned} \tag{4.8}$$

Solving (4.8) in a centralized manner requires the transport planner to know all parameters including δ_{xy} and γ_{xy}, $\forall\{x,y\} \in \mathscr{E}$. The next goal is to design a distributed method to compute the optimal Π in (4.8).

First, auxiliary variables $\pi_{xy,t}$ and $\pi_{xy,s}$ from (2.4) are introduced to (4.8). These two transport plans should be equal to each other to reach a consensus. Thus, additional constraints $\pi_{xy,t} = \pi_{xy}$ and $\pi_{xy} = \pi_{xy,s}$, $\forall\{x,y\} \in \mathscr{E}$ are included. Then, (4.8) can be reformulated as follows:

$$\begin{aligned}
\min_{\Pi_t\in\mathscr{F}_t,\Pi_s\in\mathscr{F}_s} \; & -\sum_{x\in\mathscr{X}_o}\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy,t} - \sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy,s} \\
& - \sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}(\delta_{xy}+\xi'_{xy})\pi_{xy,t} \\
\text{s.t.} \quad & \pi_{xy,t} = \pi_{xy}, \forall\{x,y\} \in \mathscr{E}, \\
& \pi_{xy} = \pi_{xy,s}, \forall\{x,y\} \in \mathscr{E},
\end{aligned} \tag{4.9}$$

where the sets Π_t and Π_s are defined in (2.5) and $\mathscr{F}_t$ and $\mathscr{F}_s$ are defined in (2.6).

The Lagrangian associated with (4.9) is:

$$\begin{aligned} L(\Pi_t,\Pi_s,\Pi,\alpha_{xy,t},\alpha_{xy,s}) = &-\sum_{x\in\mathscr{X}_o}\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy,t}-\sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy,s} \\ &-\sum_{x\in\mathscr{X}_a}\sum_{y\in\mathscr{Y}_x}\left(\delta_{xy}+\xi'_{xy}\right)\pi_{xy,t}+\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x}\alpha_{xy,t}\left(\pi_{xy,t}-\pi_{xy}\right) \\ &+\sum_{y\in\mathscr{Y}}\sum_{x\in\mathscr{X}_y}\alpha_{xy,s}\left(\pi_{xy}-\pi_{xy,s}\right)+\frac{\eta}{2}\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy,t}-\pi_{xy}\right)^2 \\ &+\frac{\eta}{2}\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy}-\pi_{xy,s}\right)^2. \end{aligned} \tag{4.10}$$

Here, $\alpha_{xy,t}$ and $\alpha_{xy,s}$ are Lagrangian multipliers associated with the constraints, and η is a positive constant.

Proposition 4. *We obtain the following steps applying the ADMM algorithm to* (4.9)*:*

$$\begin{aligned} \Pi_{x,t}(k+1)\in\arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} &-\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy,t}+\sum_{y\in\mathscr{Y}_x}\alpha_{xy,t}(k)\pi_{xy,t} \\ &+\frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy,t}-\pi_{xy}(k)\right)^2, \end{aligned} \tag{4.11}$$

$$\begin{aligned} \Pi_{x,t}(k+1)\in\arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} &-\sum_{y\in\mathscr{Y}_x}\left(\delta_{xy}+\xi'_{xy}\right)\pi_{xy,t} \\ &+\sum_{y\in\mathscr{Y}_x}\alpha_{xy,t}(k)\pi_{xy,t}+\frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy,t}-\pi_{xy}(k)\right)^2, \end{aligned} \tag{4.12}$$

where (4.11) *is used for* $x\in\mathscr{X}_o$ *and* (4.12) *for* $x\in\mathscr{X}_a$*.*

$$\begin{aligned} \Pi_{y,s}(k+1)\in\arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} &-\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy,s}+\sum_{x\in\mathscr{X}_y}\alpha_{xy,s}(k)\pi_{xy,s} \\ &+\frac{\eta}{2}\sum_{x\in\mathscr{X}_y}\left(\pi_{xy}(k)-\pi_{xy,s}\right), \end{aligned} \tag{4.13}$$

$$\begin{aligned}\pi_{xy}(k+1) \in \arg\min_{\pi_{xy}} \alpha_{xy,t}(k)\pi_{xy} + \alpha_{xy,s}(k)\pi_{xy} \\ +\frac{\eta}{2}(\pi_{xy,t}(k+1)-\pi_{xy})^2 + \frac{\eta}{2}(\pi_{xy}-\pi_{xy,s}(k+1))^2,\end{aligned} \tag{4.14}$$

$$\alpha_{xy,t}(k+1) = \alpha_{xy,t}(k) + \eta(\pi_{xy,t}(k+1)-\pi_{xy}(k+1))^2, \tag{4.15}$$

$$\alpha_{xy,s}(k+1) = \alpha_{xy,s}(k) + \eta(\pi_{xy}(k+1)-\pi_{xy,s}(k+1))^2, \tag{4.16}$$

where $\Pi_{\tilde{x},t} = \{\pi_{xy,t}\}_{y\in\mathscr{Y}_x, x=\tilde{x}}$ *and* $\Pi_{\tilde{y},s} = \{\pi_{xy,s}\}_{x\in\mathscr{X}_y, y=\tilde{y}}$ *denote the transport strategy computed by target node* $\tilde{x}$ *and source node* $\tilde{y}$*, respectively. Additionally, we define* $\mathscr{F}_{x,t} := \{\Pi_{x,t} | \pi_{xy,t} \geq 0, y \in \mathscr{Y}_x, \underline{p}_x \leq \sum_{y\in\mathscr{Y}_x}\pi_{xy,t} \leq \bar{p}_x\}$ *and* $\mathscr{F}_{y,s} := \{\Pi_{y,s} | \pi_{xy,s} \geq 0, x \in \mathscr{X}_y, \underline{q}_y \leq \sum_{x\in\mathscr{X}_y}\pi_{xy,s} \leq \bar{q}_x\}$.

Proof. See Appendix A. □

Proposition 5. *Iterations* (4.11)-(4.16) *can be simplified to five steps resulting in:*

$$\begin{aligned}\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy,t} + \sum_{y\in\mathscr{Y}_x}\alpha_{xy,t}(k)\pi_{xy,t} \\ +\frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}(\pi_{xy,t}-\pi_{xy}(k))^2,\end{aligned} \tag{4.17}$$

$$\begin{aligned}\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x}\left(\delta_{xy}+\xi'_{xy}\right)\pi_{xy,t} \\ +\sum_{y\in\mathscr{Y}_x}\alpha_{xy,t}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}(\pi_{xy,t}-\pi_{xy}(k))^2,\end{aligned} \tag{4.18}$$

where we use (4.17) *for* $x \in \mathscr{X}_o$ *and* (4.18) *for* $x \in \mathscr{X}_a$.

$$\begin{aligned}\Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} -\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy,s} + \sum_{x\in\mathscr{X}_y}\alpha_{xy,s}(k)\pi_{xy,s} \\ +\frac{\eta}{2}\sum_{x\in\mathscr{X}_y}(\pi_{xy}(k)-\pi_{xy,s}),\end{aligned} \tag{4.19}$$

$$\pi_{xy}(k+1) = \frac{1}{2}\left(\pi_{xy,t}(k+1) + \pi_{xy,s}(k+1)\right), \tag{4.20}$$

$$\alpha_{xy}(k+1) = \alpha_{xy}(k) + \frac{\eta}{2}\left(\pi_{xy,t}(k+1) - \pi_{xy,s}(k+1)\right). \tag{4.21}$$

Proof. See Appendix B. □

Theorem 1. *The algorithm described in Proposition 5 converges to an optimal solution.*

Proof. As (4.17)-(4.21) are equivalent to (4.11)-(4.16), so it is sufficient to show that (4.11)-(4.16) converge to the optimal solution. The convergence of (4.11)-(4.16) directly follows from the general arguments in [5, Section 3.2]. Therefore, the iterations (4.17)-(4.21) converge to the optimal solution of (4.9). □

In the above proposed distributed algorithm, each node computes its transport strategy based on the local information, i.e., information of connected nodes rather than all the nodes. The nodes update their strategies iteratively by communicating with connected neighbors. This is different from the centralized computation where the central planner needs to know all nodes' information to design the transport plan and then broadcasts the decision to the nodes.

4.3.4 Integrated Distributed Algorithm

We combine the algorithms for the attacker and the participants into one distributed algorithm. The integrated algorithm follows the updates below.

$$\begin{aligned} \boldsymbol{\xi}_x(k+1) \in \arg\min_{\boldsymbol{\xi}_x, \boldsymbol{\chi}_x} & \sum_{y \in \mathscr{Y}_x} \xi_{xy}\pi_{xy}(k) + \mathbf{1}^\top \boldsymbol{\chi}_x \\ \text{s.t.} \quad & \boldsymbol{\xi}_x \in \mathscr{A}_x,\ c_a\boldsymbol{\xi}_x \le \boldsymbol{\chi}_x,\ c_a\boldsymbol{\xi}_x \ge -\boldsymbol{\chi}_x. \end{aligned} \tag{4.22}$$

$$\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x}\delta_{xy}\pi_{xy,t} + \sum_{y\in\mathscr{Y}_x}\alpha_{xy}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy,t}-\pi_{xy}(k)\right)^2, \text{ for } x\in\mathscr{X}_o, \tag{4.23}$$

$$\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x}\left(\delta_{xy}+\xi_{xy}(k)\right)\pi_{xy,t} + \sum_{y\in\mathscr{Y}_x}\alpha_{xy}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x}\left(\pi_{xy,t}-\pi_{xy}(k)\right)^2, \text{ for } x\in\mathscr{X}_a, \tag{4.24}$$

$$\Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} -\sum_{x\in\mathscr{X}_y}\gamma_{xy}\pi_{xy,s} + \sum_{x\in\mathscr{X}_y}\alpha_{xy}(k)\pi_{xy,s} + \frac{\eta}{2}\sum_{x\in\mathscr{X}_y}\left(\pi_{xy}(k)-\pi_{xy,s}\right), \tag{4.25}$$

$$\pi_{xy}(k+1) = \frac{1}{2}\left(\pi_{xy,t}(k+1)+\pi_{xy,s}(k+1)\right), \tag{4.26}$$

$$\alpha_{xy}(k+1) = \alpha_{xy}(k) + \frac{\eta}{2}\left(\pi_{xy,t}(k+1)-\pi_{xy,s}(k+1)\right). \tag{4.27}$$

The convergence of the integrated distributed algorithm is worth investigation. We have the following result.

Theorem 2. *The designed integrated distributed algorithm* (4.22)-(4.27) *converges to a saddle-point equilibrium.*

Proof. Based on Proposition 3, we know that there exists an equilibrium with $\{\boldsymbol{\xi}_x^*\}_{x\in\mathscr{X}_a}$ and Π^* to the minimax game G. Theorem 1 further shows that the max-problem (4.8) converges to the best response of the min-problem (4.7). Note that the trajectory of best response dynamics for continuous concave-convex zero-sum games always converges to saddle points [18]. Thus, the developed integrated distributed algorithm (4.22)-(4.27) converges to $\{\boldsymbol{\xi}_x^*\}_{x\in\mathscr{X}_a}$ and Π^*. □

For convenience, we summarize the integrated distributed algorithm in Algorithm 2.

Algorithm 2 Integrated Distributed Algorithm With Deceptive Adversary

while $\boldsymbol{\xi}_x$, $\Pi_{x,t}$ and $\Pi_{y,s}$ not converged **do**
  Compute $\boldsymbol{\xi}_x(k+1)$ using (4.22), $\forall x \in \mathscr{X}_a$
  Compute $\Pi_{x,t}(k+1)$ using (4.23), $\forall x \in \mathscr{X}_o$
  Compute $\Pi_{x,t}(k+1)$ using (4.24), $\forall x \in \mathscr{X}_a$
  Compute $\Pi_{y,s}(k+1)$ using (4.25), $\forall y \in \mathscr{Y}$
  Compute $\pi_{xy}(k+1)$ using (4.26), $\forall \{x,y\} \in \mathscr{E}$
  Compute $\alpha_{xy}(k+1)$ using (4.27), $\forall \{x,y\} \in \mathscr{E}$
end while
return $\boldsymbol{\xi}_x(k+1)$, $\forall x \in \mathscr{X}_a$ and $\pi_{xy}(k+1)$, $\forall \{x,y\} \in \mathscr{E}$

4.4 Case Studies

This section corroborates Algorithm 2 for distributed OT while considering adversarial opponents. We consider the first case with five target nodes and two source nodes with a network structure connecting every source node to every target node. The network structure follows similarly from Fig. 2.1(a) except the network is complete in this case study. The upper bounds for the source nodes are $\bar{p}_1 = 2$, $\bar{p}_2 = 3$, $\bar{p}_3 = 4$, $\bar{p}_4 = 3$, $\bar{p}_5 = 2$, $\bar{q}_6 = 5$, and $\bar{q}_7 = 5.5$. The lower bound for all nodes are set to 0. Additionally, we consider linear utility functions $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$, and $s_{xy}(\pi_{xy}) = \gamma_{xy}\pi_{xy}, \forall \{x,y\} \in \mathscr{E}$. The corresponding parameters in the linear functions are selected as follows:

$$[\delta_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 4 & 12 & 4 & 12 & 8 \\ 8 & 8 & 16 & 4 & 4 \end{bmatrix},$$

$$[\gamma_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 6 & 4.5 & 12 & 6 & 9 \\ 3 & 6 & 7.5 & 9 & 12 \end{bmatrix}.$$

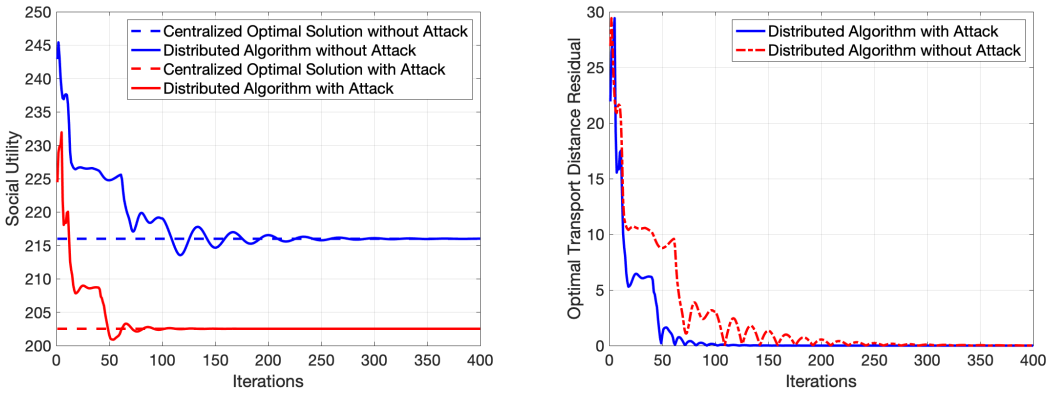
Furthermore, adversary's parameters are $c_a = 0.5$ and $\kappa_x = 15$, $\forall x \in \mathscr{X}_a$, and the deceptive targets include nodes 2 and 5. We next design the resilient transport strategy using the proposed distributed Algorithm 2.

First, we show that the algorithm works and converges to the same value obtained by the centralized method. We also compare the transport strategies when the network has and does not have adversaries. When there is an adversary, we use a combination of (4.23) (for benign targets) and (4.24) (for deceptive targets) to calculate $\Pi_{x,t}(k+1)$. When there is no adversary, meaning none of the nodes are compromised, we only use (4.24) to compute $\Pi_{x,t}$.

The results of the case studies are shown in Fig. 4.1. Specifically, Fig. 4.1(a) shows the social utility which is the aggregated payoff of all nodes. Fig. 4.1(a) confirms that the algorithm converges to the centralized solution both with and without attacks. Note that when considering attacks, the algorithm converges to a lower social utility. This is due to the fact that the algorithm accounts for the adversarial impacts which decreases the desired utility between the source node and the compromised target node. Fig. 4.1(b) highlights the distance residual of the transport strategy, which measures the difference between the strategy at each step and the equilibrium solution.

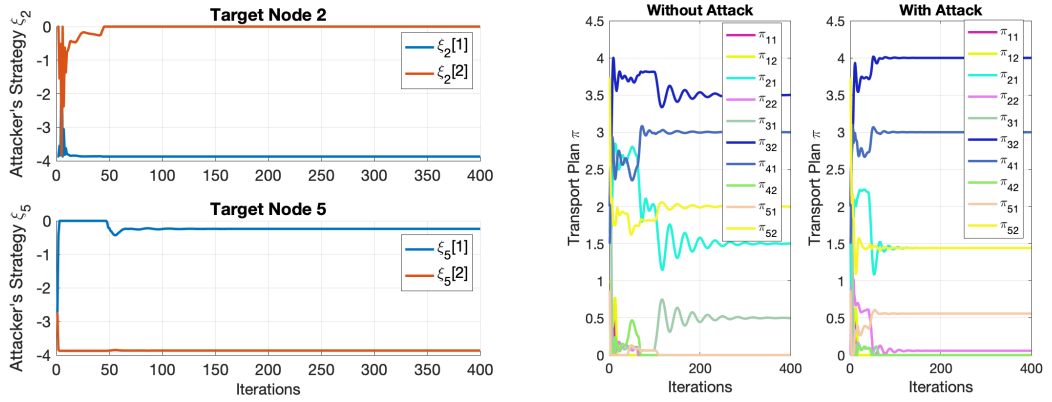
The attacker's strategy $\boldsymbol{\xi}_x$ is shown in Fig. 4.2(a). For both compromised nodes, the deceptive strategies $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_5$ converge to a nonzero values, indicating that the attacker is actively affecting the transport plan. Fig. 4.2(b) further illustrates this phenomenon as the resource allocation strategies are different in the two investigated cases.

We further investigate a larger scale network with 3 source nodes and 30 target nodes and every target is connected to every source node. The parameters are generated randomly following uniform distributions: $\delta_{xy} \sim U(6,11)$, $\gamma_{xy} \sim U(7,12)$, $\bar{p}_x \sim U(5,10)$, and $\bar{q}_y \sim U(67,75)$. Nodes 8, 15, and 25 are considered to be possibly compromised



(a) Social Utility (b) Distance Residual

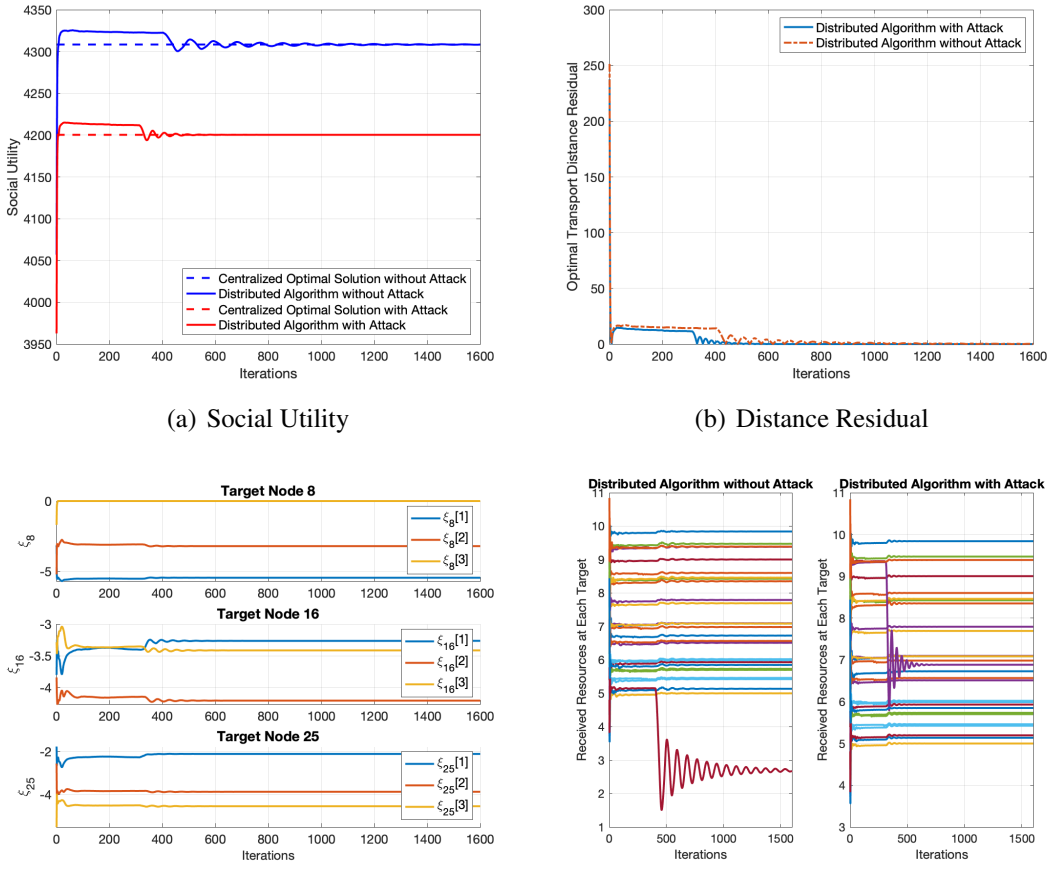
Figure 4.1: Impact of the adversarial attacks on the transport strategy design using Algorithm 2. (a) and (b) depict the trajectories of social utility and residual of transport strategy, respectively.



(a) Attacker's Strategy (b) Transport Plan

Figure 4.2: The strategy of the attacker and resulting transport plan. (a) shows the attacker's strategy at the target nodes 2 and 5. (b) shows the corresponding transport plan under two scenarios.

with $c_a = 0.5$ and $\kappa_x = 40$. The obtained results are shown in Fig. 4.3. The results also converge to the centralized solutions. We can conclude that the designed algorithm is applicable to large-scale networks.



(a) Social Utility

(b) Distance Residual

(c) Attacker's Strategy

(d) Transport Plan

Figure 4.3: Example of a larger-scale network. (a) and (b) depict the trajectories of social utility and residual of transport strategy, respectively. (c) and (d) show the attacker's strategy and the corresponding transport plans, respectively.

4.5 Conclusion

In this chapter, an adversarial discrete optimal transport framework for resource matching in which the participating nodes could be malicious by reporting untruthful preference parameters was formulated. We have developed a distributed algorithm for computing the strategic resource allocation strategies which are resilient to such attacks. The designed algorithm converges to the same solution as the one designed by a central-

ized planner, and it is applicable to large scale networks susceptible to deceptive attacks. The adversarial behavior is specifically acknowledged in the algorithm when a participating node is compromised. Each connected pair of target and source nodes negotiate on the their proposed transport plans, and thus the compromised node's actions is taken into account in the final allocation schemes. The algorithm terminates when the sources and targets reach a consensus.

Chapter 5

Differentially Private Distributed Optimal Transport

When transport information is shared between connected nodes during strategy updates, an attacker can use that information to infer private data at each node, which raises significant privacy concerns. This chapter investigates how differential privacy can be leveraged to protect source and target node data. To achieve this goal we first describe the distributed optimal transport algorithm, and then discuss where and how differential privacy can be added to the algorithm.

5.1 Non-Private Distributed Algorithm

We start from working directly from the discrete OT formulation in (2.4) as the privacy measures are not considered directly in the objective function of the minimization problem. It is necessary to add the following assumption to Assumption 1 about the utility functions.

Assumption 3. *The utility functions t_{xy} and s_{xy} are continuously differentiable with $t'_{xy} \leq \rho$ and $s'_{xy} \leq \rho$, where ρ is a positive constant.*

The Lagrangian from (2.4) is formulated as follows:

$$
\begin{aligned}
L(\Pi_t, \Pi_s, \Pi, \alpha_{xy,t}, \alpha_{xy,s}) = & - \sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}_x} t_{xy}(\pi_{xy,t}) \\
& - \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} s_{xy}(\pi_{xy,s}) + \sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}_x} \alpha_{xy,t}(\pi_{xy,t} - \pi_{xy}) \\
& + \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} \alpha_{xy,s}(\pi_{xy} - \pi_{xy,s}) + \frac{\eta}{2} \sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy})^2 \\
& + \frac{\eta}{2} \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} (\pi_{xy} - \pi_{xy,s})^2, \qquad (5.1)
\end{aligned}
$$

where $\eta > 0$ is a positive scalar constant controlling the convergence rate in the algorithm designed below.

Note that in (5.1), the last two terms of the Lagrangian $\frac{\eta}{2} \sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy})^2$ and $\frac{\eta}{2} \sum_{y \in \mathscr{Y}} \sum_{x \in \mathscr{X}_y} (\pi_{xy} - \pi_{xy,s})^2$, acting as penalization, are quadratic. Hence, the Lagrangian function L is strictly convex, ensuring the existence of a unique optimal solution.

We then can apply ADMM to the minimization problem in (2.4).

Proposition 6. *The iterative steps of applying ADMM to problem* (2.4) *are summarized as follows:*

$$
\begin{aligned}
\Pi_{x,t}(k+1) \in \arg \min_{\Pi_{x,t} \in \mathscr{F}_{x,t}} & - \sum_{y \in \mathscr{Y}_x} t_{xy}(\pi_{xy,t}) \\
& + \sum_{y \in \mathscr{Y}_x} \alpha_{xy,t}(k) \pi_{xy,t} + \frac{\eta}{2} \sum_{y \in \mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy}(k))^2, \qquad (5.2)
\end{aligned}
$$

$$
\begin{aligned}
\Pi_{y,s}(k+1) \in \arg \min_{\Pi_{y,s} \in \mathscr{F}_{y,s}} & - \sum_{x \in \mathscr{X}_y} s_{xy}(\pi_{xy,s}) \\
& - \sum_{x \in \mathscr{X}_y} \alpha_{xy,s}(k) \pi_{xy,s} + \frac{\eta}{2} \sum_{x \in \mathscr{X}_y} (\pi_{xy}(k) - \pi_{xy,s})^2, \qquad (5.3)
\end{aligned}
$$

$$\pi_{xy}(k+1) = \arg\min_{\pi_{xy}} -\alpha_{xy,t}(k)\pi_{xy} + \alpha_{xy,s}(k)\pi_{xy} + \frac{\eta}{2}(\pi_{xy,t}(k+1) - \pi_{xy})^2 + \frac{\eta}{2}(\pi_{xy} - \pi_{xy,s}(k+1))^2, \tag{5.4}$$

$$\alpha_{xy,t}(k+1) = \alpha_{xy,t}(k) + \eta(\pi_{xy,t}(k+1) - \pi_{xy}(k+1))^2, \tag{5.5}$$

$$\alpha_{xy,s}(k+1) = \alpha_{xy,s}(k) + \eta(\pi_{xy}(k+1) - \pi_{xy,s}(k+1))^2, \tag{5.6}$$

where $\Pi_{\tilde{x},t} := \{\pi_{xy,t}\}_{y\in\mathscr{Y}_x, x=\tilde{x}}$ *represents the solution at target node* $\tilde{x} \in \mathscr{X}$*, and* $\Pi_{\tilde{y},s} := \{\pi_{xy,s}\}_{x\in\mathscr{X}_y, y=\tilde{y}}$ *represents the proposed solution at source node* $\tilde{y} \in \mathscr{Y}$*. In addition,* $\mathscr{F}_{x,t} := \{\Pi_{x,t} | \pi_{xy,t} \geq 0, y \in \mathscr{Y}_x, \underline{p}_x \leq \sum_{y\in\mathscr{Y}_x} \pi_{xy,t} \leq \bar{p}_x\}$*, and* $\mathscr{F}_{y,s} := \{\Pi_{y,s} | \pi_{xy,s} \geq 0, x \in \mathscr{X}_y, \underline{q}_y \leq \sum_{x\in\mathscr{X}_y} \pi_{xy,s} \leq \bar{q}_y\}$.

Proof. See Appendix A. □

Again the steps can be simplified down to four steps, and the results are summarized below.

Proposition 7. *The iterations* (5.2)-(5.6) *can be simplified as follows:*

$$\Pi_{x,t}(k+1) \in \arg\min_{\Pi_{x,t}\in\mathscr{F}_{x,t}} -\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) + \sum_{y\in\mathscr{Y}_x} \alpha_{xy}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x} (\pi_{xy,t} - \pi_{xy}(k))^2, \tag{5.7}$$

$$\Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s}\in\mathscr{F}_{y,s}} -\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) - \sum_{x\in\mathscr{X}_y} \alpha_{xy}(k)\pi_{xy,s} + \frac{\eta}{2}\sum_{x\in\mathscr{X}_y} (\pi_{xy}(k) - \pi_{xy,s})^2, \tag{5.8}$$

$$\pi_{xy}(k+1) = \frac{1}{2}(\pi_{xy,t}(k+1) + \pi_{xy,s}(k+1)), \tag{5.9}$$

$$\alpha_{xy}(k+1) = \alpha_{xy}(k) + \frac{\eta}{2}\left(\pi_{xy,t}(k+1) - \pi_{xy,s}(k+1)\right). \tag{5.10}$$

Proof. See Appendix B. □

For convenience, we summarize the distributed OT algorithm into Algorithm 3.

Algorithm 3 Distributed OT Algorithm

1: **while** $\Pi_{x,t}$ and $\Pi_{y,s}$ not converging **do**
2: Compute $\Pi_{x,t}(k+1)$ using (5.7), for all $x \in \mathscr{X}_y$
3: Compute $\Pi_{y,s}(k+1)$ using (5.8), for all $y \in \mathscr{Y}_x$
4: Compute $\pi_{xy}(k+1)$ using (5.9), for all $\{x,y\} \in \mathscr{E}$
5: Compute $\alpha_{xy}(k+1)$ using (5.10), for all $\{x,y\} \in \mathscr{E}$
6: **end while**
7: **return** $\pi_{xy}(k+1)$, for all $\{x,y\} \in \mathscr{E}$

5.2 Differentially Private Algorithm

In this section, we first present the privacy concerns in the developed distributed OT in Section 5.1. We then develop a differentially private distributed OT algorithm which promotes nodes' privacy explicitly during decision updates.

5.2.1 Privacy Concerns in the Distributed OT

In the distributed OT paradigm highlighted in Algorithm 3, the intermediate results are shared between connected nodes during updates. This sharing mechanism raises privacy concerns as an adversary that can access this result (e.g, through eavesdropping attack) has the ability to infer the participants' private information. Specifically, the adversary could leverage the compromised information $\Pi_{x,t}(k)$ and $\Pi_{y,s}(k)$ at each update step, k, to infer the node's private information including the sensitive preference

parameters in the utility functions t_{xy} and s_{xy}. We denote the set of private preference information at node p by D_p, $p \in \mathscr{P}$.

Next, we use an example to further illustrate the node's private information set. Specifically, we consider utility functions admitting a linear form for both the sender and receiver: $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$ and $s_{xy}(\pi_{xy}) = \gamma_{xy}\pi_{xy}$, where $\delta_{xy}, \gamma_{xy} \in \mathbb{R}_+$. Then, for a target node $x \in \mathscr{X}$, we have the set $D_x = \{\delta_{xy} : \forall y \in \mathscr{Y}_x\}$. Similarly for a source node $y \in \mathscr{Y}$, we have the set $D_y = \{\gamma_{xy} : \forall x \in \mathscr{X}_y\}$. The information contained in D_p is crucial for developing optimal transport plans. Leakage of such private information is undesired in many resource allocation scenarios, especially those with societal impacts. For example, in the distribution of scarce vaccine resources, these preference parameters could indicate the severity of epidemics in different neighborhoods (modeled by nodes). It is obvious that each participant does not want to leak this piece of information to other unauthorized parties.

To this end, we aim to protect the privacy of each node in the transport network using differential privacy [14]. Specifically, we propose adding randomness to the transport decisions communicated between each pair of source-target nodes during updates, and hence prevent the adversary from learning the sensitive utility parameters of the nodes simply based on the transport decisions. To achieve this goal, first, let D_p and D'_p be two information/data sets differ by one data point (utility parameter). In other words, their *Hamming Distance* is equal to 1, denoted by $H(D_p, D'_p) = 1$. Here, $H(D_p, D'_p) = \sum_{i=1}^{|D_p|} \mathbf{1}\{i : d_i \neq d'_i\}$, where d_i and d'_i denote the ith data point in the information sets D_p and D'_p, respectively. Recall that the date points in these sets refer to the nodes' utility parameters which we aim to protect from leakage under the condition that the adversary intercepts the transport plans. The formal definition of differential privacy is presented below.

Definition 2 ($\beta_p(k)$-Differential Privacy)**.** *Consider the transport network $\mathscr{G} = \{\mathscr{P}, \mathscr{E}\}$, where $\mathscr{P}$ is composed of both source nodes and target nodes, and $\mathscr{E}$ is the set of edges connecting the nodes. At each node $p \in \mathscr{P}$, there is an information set D_p which is used to compute the resource transport plan. Let R be a randomized counterpart of Algorithm 3. Further, let $\beta(k) = \big(\beta_1(k), \beta_2(k), ..., \beta_{|\mathscr{P}|}(k)\big) \in \mathbb{R}_+^{|\mathscr{P}|}$, where $\beta_p(k) \in \mathbb{R}_+$ is the privacy parameter of node p at iteration k. Consider the outputs $\Pi_{x,t}(k)$ and $\Pi_{y,s}(k)$ at iteration k of Algorithm 3. Let D'_p be any information set such that $H(D'_p, D_p) = 1$ and $\widetilde{\Pi}_{x,t}(k)$ and $\widetilde{\Pi}_{y,s}(k)$ be the corresponding outputs of Algorithm 3 while using the information set D'_p. The algorithm R is $\beta_p(k)$-differentially private for any D'_p for all nodes $p \in \mathscr{P}$ and for all possible sets of outcome solutions S, if the following condition is satisfied at every iteration k:*

$$\Pr[\Pi_p(k) \in S] \leq \exp\left(\beta_p(k)\right) \cdot \Pr[\widetilde{\Pi}_p \in S], \tag{5.11}$$

$$\textit{where } \Pi_p(k) = \begin{cases} \Pi_{p,t}(k), \text{ if } p \in \mathscr{X}, \\ \Pi_{p,s}(k), \text{ if } p \in \mathscr{Y}, \end{cases} \quad \textit{and } \widetilde{\Pi}_p(k) = \begin{cases} \widetilde{\Pi}_{p,t}(k), \text{ if } p \in \mathscr{X}, \\ \widetilde{\Pi}_{p,s}(k), \text{ if } p \in \mathscr{Y}. \end{cases}$$

5.2.2 Output Variable Perturbation

In order to ensure that the sensitive preference information at each node remains private when transport plans are published over the network, we develop a differentially private algorithm based on output variable perturbation. This algorithm involves adding random noise to the output decision variables $\Pi_{x,t}(k+1)$ and $\Pi_{y,s}(k+1)$ during updates. More specifically, the random noise vectors, $\varepsilon_x(k+1) \in \mathbb{R}^{|\mathscr{Y}_x|}$ and $\varepsilon_y(k+1) \in \mathbb{R}^{|\mathscr{X}_y|}$ are added to the variables $\Pi_{x,t}(k+1)$ and $\Pi_{y,s}(k+1)$ obtained by (5.7) and (5.8), respectively.

Recall that $p \in \mathscr{P} = \mathscr{X} \cup \mathscr{Y}$ and thus $p = x$, $\forall x \in \mathscr{X}$, and $p = y$, $\forall y \in \mathscr{Y}$. The random noise vector $\varepsilon_p(k)$ is generated according to a distribution with density function $F_p(\varepsilon) \sim e^{-\xi_p(k)||\varepsilon||}$. Here, $\xi_p(k) = \frac{\rho}{\eta}\beta_p(k)$, where β_p is a privacy term at each node p.

Thus, the proposed solutions at the target node x and the source node y at step $k+1$ admit

$$\begin{aligned} \Pi^*_{x,t}(k+1) &= \Pi_{x,t}(k+1) + \varepsilon_x(k+1), \\ \Pi^*_{y,s}(k+1) &= \Pi_{y,s}(k+1) + \varepsilon_y(k+1), \end{aligned} \tag{5.12}$$

where $\Pi^*_{x,t}$ and $\Pi^*_{x,t}$ are perturbed solutions of Π^t_x and Π^t_x, respectively. The distributed OT algorithm with output perturbation includes the following steps:

$$\begin{aligned} \Pi_{x,t}(k+1) \in \arg\min_{\Pi^t_x \in \mathscr{F}^t_x} & -\sum_{y\in\mathscr{Y}_x} t_{xy}(\pi_{xy,t}) \\ & + \sum_{y\in\mathscr{Y}_x} \alpha_{xy}(k)\pi_{xy,t} + \frac{\eta}{2}\sum_{y\in\mathscr{Y}_x} \left(\pi_{xy,t} - \pi_{xy}(k)\right)^2, \end{aligned} \tag{5.13}$$

$$\Pi^*_{x,t}(k+1) = \Pi_{x,t}(k+1) + \varepsilon_x(k+1), \tag{5.14}$$

$$\begin{aligned} \Pi_{y,s}(k+1) \in \arg\min_{\Pi_{y,s} \in \mathscr{F}_{y,s}} & -\sum_{x\in\mathscr{X}_y} s_{xy}(\pi_{xy,s}) \\ & - \sum_{x\in\mathscr{X}_y} \alpha_{xy}(k)\pi_{xy,s} + \frac{\eta}{2}\sum_{x\in\mathscr{X}_y} \left(\pi_{xy}(k) - \pi_{xy,s}\right)^2, \end{aligned} \tag{5.15}$$

$$\Pi^*_{y,s}(k+1) = \Pi_{y,s}(k+1) + \varepsilon_y(k+1), \tag{5.16}$$

$$\pi^*_{xy}(k+1) = \frac{1}{2}\left(\pi^*_{xy,t}(k+1) + \pi^*_{xy,s}(k+1)\right), \tag{5.17}$$

$$\alpha_{xy}(k+1) = \alpha_{xy}(k) + \frac{\eta}{2}\left(\pi^*_{xy,t}(k+1) - \pi^*_{xy,s}(k+1)\right). \tag{5.18}$$

As a result of the perturbation in (5.14) and (5.16), $\Pi^*_{x,t}(k)$ and $\Pi^*_{y,s}(k)$ are random-

ized. Specifically, within each iteration, the node perturbs the output variable $\Pi_{x,t}(k)$ or $\Pi_{y,s}(k)$ respectively in order to obtain $\Pi^*_{x,t}(k)$ or $\Pi^*_{y,s}(k)$. The proposed scheme is further illustrated in Fig. 5.1. It is important to note that the information sets at each node, i.e., D_p containing sensitive utility parameters, remains untouched and is not perturbed. For convenience, the differentially private distributed OT algorithm based on the output variable perturbation is summarized in Algorithm 4. We further have the following theorem which guarantees the privacy-preserving property of Algorithm 4.

Theorem 3. *The proposed Algorithm 4 is β_p-differentially private with $\beta_p(k)$ for node p at iteration k. Let $Q(\Pi^*_{x,t}|D_x)$ and $Q(\Pi^*_{x,t}|D'_x)$ be the probability density functions for $\Pi^*_{x,t}$ given the information sets D_x and D'_x such that $H(D_x, D'_x) = 1$. The ratio of*

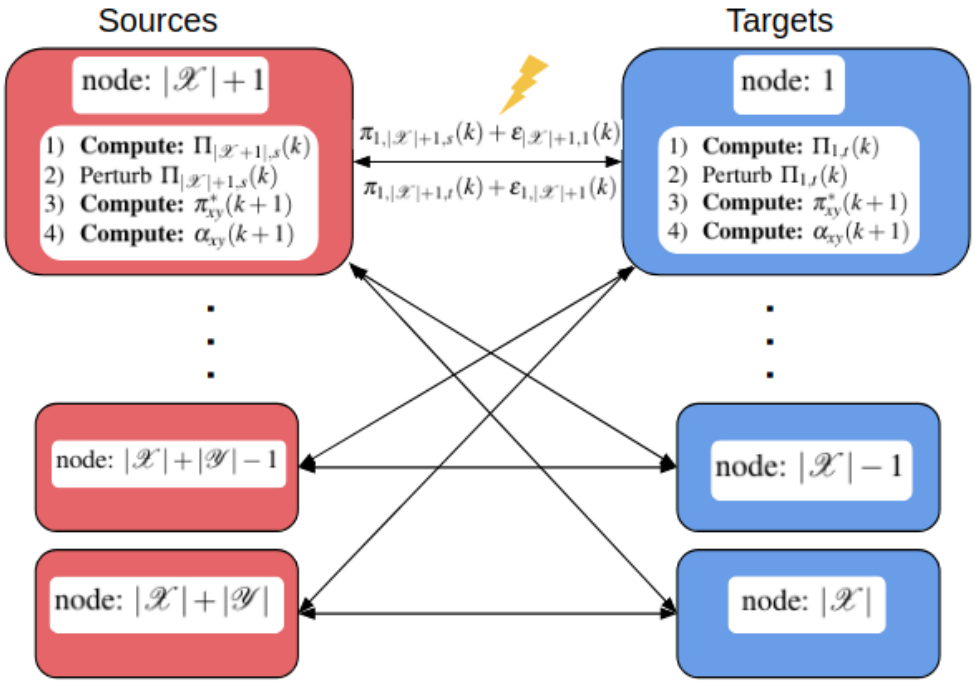


Figure 5.1: Illustration of the differentially private distributed OT scheme. The information exchanged between nodes is susceptible to be intercepted by the adversary (e.g., by eavesdropping attack to the wireless channel). Hence, an appropriate random noise is added to the outputs at each update step.

Algorithm 4 Differentially Private Distributed OT Algorithm With Output Variable Perturbation

for $k = 0, 1, 2, \ldots$ **do**
  for $x \in \mathscr{X}_y$ **do**
    Compute $\Pi_{x,t}(k+1)$ using (5.13)
    Compute $\Pi^*_{x,t}(k+1)$ using (5.14)
  end for
  for $y \in \mathscr{Y}_x$ **do**
    Compute $\Pi_{y,s}(k+1)$ using (5.15)
    Compute $\Pi^*_{y,s}(k+1)$ using (5.16)
  end for
  Compute $\pi^*_{xy}(k+1)$ using (5.17), for all $\{x,y\} \in \mathscr{E}$
  Compute $\alpha_{xy}(k+1)$ using (5.18), for all $\{x,y\} \in \mathscr{E}$
end for
return $\pi^*_{xy}(k+1)$, for all $\{x,y\} \in \mathscr{E}$

probability density of $\Pi^*_{x,t}$ *is bounded:*

$$\frac{Q(\Pi^*_{x,t}(k)|D_x)}{Q(\Pi^*_{x,t}(k)|D'_x)} \leq e^{\beta_x(k)}. \tag{5.19}$$

It follows similarly for the probability density of $\Pi^*_{y,s}$*, i.e.,*

$$\frac{Q(\Pi^*_{y,s}(k)|D_y)}{Q(\Pi^*_{y,s}(k)|D'_y)} \leq e^{\beta_y(k)}. \tag{5.20}$$

Note that (5.19) *and* (5.20) *directly imply* $\frac{\Pr(\Pi^*_{x,t}(k)|D_x)}{\Pr(\Pi^*_{x,t}(k)|D'_x)} \leq e^{\beta_x(k)}$ *and* $\frac{\Pr(\Pi^*_{y,s}(k)|D_y)}{\Pr(\Pi^*_{y,s}(k)|D'_y)} \leq e^{\beta_y(k)}$*, respectively.*

Proof. See Appendix C. □

In summary, the proposed Algorithm 4 guarantees the privacy of all participating nodes during their decision sharing.

5.3 Case Studies

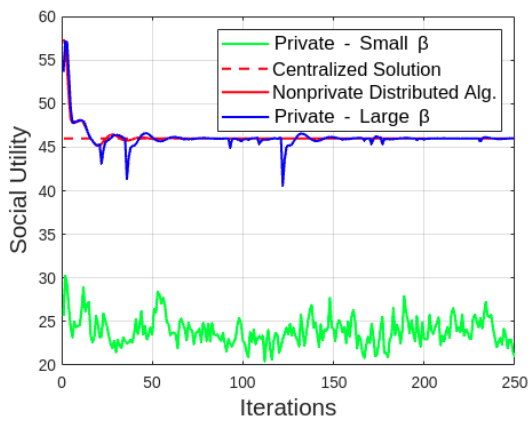
In this section, we corroborate the effectiveness of the developed differentially private algorithm and show how the added privacy affects the transport plan and the efficiency. We first study a network with two source nodes and five target nodes where every source is connected to every target. The target nodes are labeled from 1 to 5 and the source nodes have indices 6 and 7. The upper bounds for the target nodes are $\bar{p}_1 = 2$, $\bar{p}_2 = 3$, $\bar{p}_3 = 4$, $\bar{p}_4 = 3$, $\bar{p}_5 = 2$, $\bar{q}_6 = 4$ and $\bar{q}_7 = 4$ for the source nodes. The lower bound for all nodes are set to 0. Additionally, we consider linear utility functions $t_{xy}(\pi_{xy}) = \delta_{xy}\pi_{xy}$, and $s_{xy}(\pi_{xy}) = \gamma_{xy}\pi_{xy}, \forall\{x,y\} \in \mathscr{E}$. The corresponding parameters in the functions are selected as follows:

$$[\delta_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 0.25 & 1.5 & 0.25 & 1.5 & 0.75 \\ 0.75 & 0.75 & 1.75 & 0.25 & 0.25 \end{bmatrix},$$

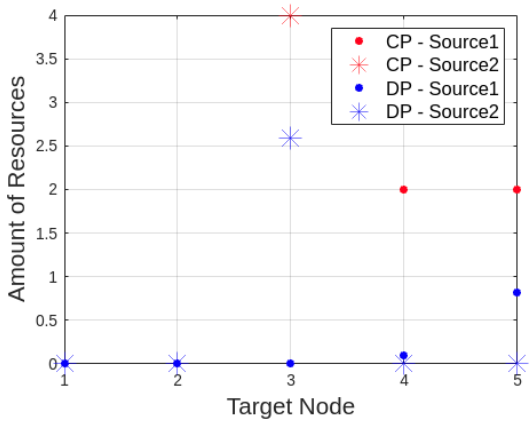
$$[\gamma_{xy}]_{x\in\mathscr{X},y\in\mathscr{Y}} = \begin{bmatrix} 0.5 & 0.3 & 1.5 & 0.5 & 0.8 \\ 0.2 & 0.5 & 0.6 & 0.4 & 1.5 \end{bmatrix}.$$

In the following study, we investigate the impact of β_p which captures the level of privacy. According to the definition, a smaller β_p yields a higher level privacy. For small β_p, we choose $\beta_1 = 0.2$, $\beta_2 = 0.1$, $\beta_3 = 0.3$, $\beta_4 = 0.1$, $\beta_5 = 0.2$, $\beta_6 = 0.1$, and $\beta_7 = 0.1$. For larger values of β_p, we increase these numbers $1,000$-fold. Additionally, we define $\eta = 1$ and $\rho = 2$.

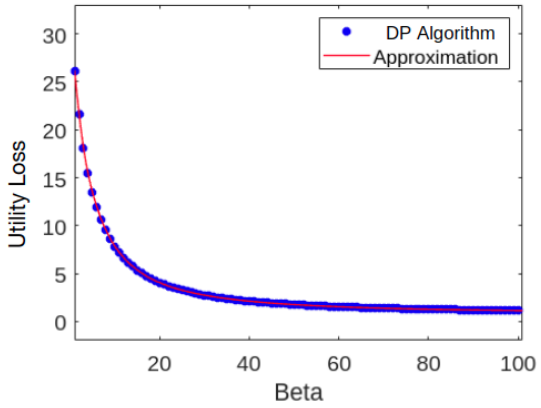
We leverage the developed algorithms, Algorithms 3 and 4, to compute the transport plans. The results are shown in Fig. 5.2. First, in Fig. 5.2(a), the trajectory of transport plan yielded by the differentially private algorithm oscillates around some point. The



(a) Social utility



(b) Transport plan



(c) Privacy and transport efficiency trade-off

Figure 5.2: The first graph (a) shows the performance of the proposed algorithms. (b) depicts the transport plans designed using centralized algorithm by the central planner (CP) and using the differentially private (DP) algorithm. (c) shows an increase of the privacy level (smaller β_p) decreases the transport utility, reflecting the trade-off between privacy and transport efficiency.

oscillation is because of the random noise added to the decision at each output perturbation step. We can also see that when β_p is small, the resulting social utility (i.e., transport efficiency), which is an aggregation of the utilities of all participating nodes, is relatively small. In comparison, when β_p is large, the social utility is close to the one returned by Algorithm 3 where differential privacy is not incorporated, and thus the transport so-

lution is efficient. Fig. 5.2(c) further shows this phenomenon and reveals the inherent trade-off between the amount of added privacy and the transport efficiency. Fig. 5.2(b) illustrates how the privacy factor affects the overall transport plan. The decreased optimality due to the privacy promotion indicates that the resource allocation is no longer taking full advantage of how much source nodes can provide or how much target nodes can request. For example, target node 3 can request at most 4 units of resources, and does so when privacy is not added to the algorithm. When privacy is concerned, it only requests 2.5 units of resources. This study shows how private and efficient transport scheme can be achieved.

5.4 Conclusion

This chapter developed a differentially private distributed optimal transport algorithm which has a theoretical guarantee of achieved privacy. The algorithm protects the sensitive information at each node by perturbing the output of the transport schemes shared between connected nodes during updates. Under the designed mechanism, even if the transport decision is intercepted during its transmission, the adversary still cannot discover the underlying sensitive information used in the transport strategy design. The privacy level for each node can be determined appropriately by considering its trade-off with the resulting transport efficiency. Future work includes extending the current model-based distributed optimal transport framework to data-driven learning-based optimal transport while considering data privacy in the learning process.

Chapter 6

Conclusion

This thesis has laid a foundation for fair, secure, and privacy-preserving distributed discrete optimal transport design. When a fairness metric is incorporated, resources are distributed more evenly across the target nodes. The fairness is explicitly promoted in the algorithm, through negotiations between each pair of source and target. Throughout the negotiation steps, the sources maximize their revenue but need to consider the fairness requests. Similarly, the targets optimize the fairness but should take into account the efficiency of resource allocation as well. The algorithm terminates when the two parties reach a consensus. Using a game-theoretic approach, the second part of this thesis developed secure and resilient transport strategies to counteract the adversarial attacks to a set of source nodes in the resource allocation planning. The adversarial behavior is specifically acknowledged in the developed best-response type of algorithm when each connected pair of target and source nodes negotiate on the their proposed transport plans, and the compromised node's actions are inherently taken into account in the final allocation schemes. The third part of the thesis proposed a differential privacy based mechanism that perturbs information published over the network in the distributed OT algorithm. Under the designed mechanism, even if the transport decision is intercepted during its transmission, the adversary still cannot discover the underlying sensitive in-

formation used in the transport strategy design. The privacy level for each node can be determined appropriately by considering its trade-off with the resulting transport efficiency.

Optimal transport has become increasingly more popular over recent years because of its applications to fields such as machine learning, image processing, computer graphics, economics and more. Its popularity will continue to grow and thus the need for incorporated fairness, security, and privacy in the algorithm such as the ones proposed and tested in this thesis will be increasingly in demand.

Appendix A

Proof of Proposition 1, 4 and 6

Proof. Let $\vec{x} = [\vec{\Pi}_{x,t}^T, \vec{\Pi}^T]^T$, $\vec{y} = [\vec{\Pi}^T, \vec{\Pi}_{y,s}^T]^T$, and $\alpha = [\{\alpha_{xy,t}\}^T, \{\alpha_{xy,s}\}^T]^T$, where $\vec{}$ denotes the vectorization operator. We note that these vectors are all $2N \times 1$ where N is the number of connections between targets and sources. This is also the size of $\mathscr{E}$. Now we can write the constraints in matrix form such that $A\vec{x} = \vec{y}$ where $A = [\mathbf{I}, \mathbf{0}, \mathbf{I}, \mathbf{0}]$. Here $\mathbf{I}$ and $\mathbf{0}$ denote the identity and zero matrices respectively, both of which are $N \times N$. Next, we note that $\vec{x} \in \mathscr{F}_{\vec{x},t}$ and $\vec{y} \in \mathscr{F}_{\vec{y},s}$, where $\mathscr{F}_{\vec{x},t} = \{\vec{x} | \pi_{xy,t} \geq 0, \underline{p}_x \leq \sum_{y \in \mathscr{Y}_x} \pi_{xy,t} \leq \bar{p}_x, \{x,y\} \in \mathscr{E}\}$, $\mathscr{F}_{\vec{y},s} := \{\vec{y} | \pi_{xy,s} \geq 0, \underline{q}_y \leq \sum_{x \in \mathscr{X}_y} \pi_{xy,s} \leq \bar{q}_y, xy \in \mathscr{E}\}$. In turn we can solve the minimization in (3.5) with the iterations: 1) $\vec{x}(k+1) \in \arg\min_{\vec{x} \in \mathscr{F}_{x,t}} L(\vec{x}, \vec{y}(k), \alpha(k))$; 2) $\vec{y}(k+1) \in \arg\min_{\vec{y} \in \mathscr{F}_{y,s}} L(\vec{x}(k), \vec{y}, \alpha(k))$; 3) $\alpha(k+1) = \alpha(k) + \eta(A\vec{x}(k+1) - \vec{y}(k+1))$, whose convergence is proved [5]. Because we have no coupling among $\Pi_{x,t}, \Pi_{y,s}, \pi_{xy}, \alpha_{xy,t}$, and $\alpha_{xy,s}$ the above iterations can be decomposed to equations (3.7)-(3.11)(or (4.11)-(4.16), (5.2)-(5.6)). □

Appendix B

Proof of Proposition 2, 5 and 7

Proof. Note that (3.9), (4.13) and (5.4) are equivalent. As are (3.10), (4.14), (5.5), and (3.11), (4.15), (5.6), and (3.15), (4.20), (5.10). Thus they can be substituted respectively.

As (3.9) is strictly concave, we can solve it by first-order condition:

$$\pi_{xy}(k+1) = \frac{1}{2\eta}(\alpha_{xy,t}(k) - \alpha_{xy,s}(k)) + \frac{1}{2}(\pi_{xy,t}(k+1) + \pi_{xy,s}(k+1)).$$

By substituting the above equation into (3.10) and (3.11) we get:

$$\alpha_{xy,t}(k+1) = \frac{1}{2}(\alpha_{xy,t}(k) + \alpha_{xy,s}(k)) + \frac{\eta}{2}(\pi_{xy,t}(k+1) - \pi_{xy,s}(k+1)),$$

$$\alpha_{xy,s}(k+1) = \frac{1}{2}(\alpha_{xy,t}(k) + \alpha_{xy,s}(k)) + \frac{\eta}{2}(\pi_{xy,t}(k+1) - \pi_{xy,s}(k+1)).$$

We can see that $\alpha_{xy,t} = \alpha_{xy,s}$ during each update. Hence, $\pi_{xy}(k+1)$ can be further simplified as $\pi_{xy}(k+1) = \frac{1}{2}(\pi_{xy,t}(k+1) + \pi_{xy,s}(k+1))$. In addition, we can achieve (3.10) and (3.11) from $\alpha_{xy,t} = \alpha_{xy,s} = \alpha_{xy}$ represented in (3.15). □

Appendix C

Proof of Theorem 3

Proof. We first show the bounded ratio in (5.19). We have $\frac{Q(\Pi^*_{x,t}(k)|D_x)}{Q(\Pi^*_{x,t}(k)|D'_x)} = \frac{F_x(\varepsilon_x(k))}{F_x(\varepsilon'_x(k))} = \frac{e^{-\xi_x(k)||\varepsilon_x(k)||}}{e^{-\xi_x(k)||\varepsilon'_x(k)||}}$. Our goal is to find a $\xi_x(k)$ such that the following inequality holds true: $\xi_x(k)(||\varepsilon_x(k)|| - ||\varepsilon'_x(k)||) \leq \beta_p(k)$. Let $W = \arg\min_{\Pi_{x,t}} f_x(k|D_x)$ and the associated $W' = \arg\min_{\Pi_{x,t}} f_x(k|D'_x)$, where $f_x(k)$ is the objective function for the target node $x \in \mathscr{X}$ at iteration k, shown in (5.13). Also, let g and h be defined at each node $x \in \mathscr{X}$ such that $g(\Pi^*_{x,t}(k)) = f_x(k|D_x)$ and $h(\Pi^*_{x,t}(k)) = f_x(k|D'_x) - f_x(k|D_x)$.

Therefore, $h(\Pi^*_{x,t}(k)) = -\tilde{t}_{xy}(\pi_{xy,t}) + t_{xy}(\pi_{xy,t})$, where $\tilde{t}_{xy}$ refers to the altered utility function due to the difference between D'_x and D_x. Assumption 1 implies that $f_x(k|D_p) = g(\Pi^*_{x,t}(k))$ and $f_x(k|D'_x) = g(\Pi^*_{x,t}(k)) + h(\Pi^*_{x,t}(k))$ are both convex. We differentiate $h(\Pi^*_{x,t}(k))$ with respect to $\Pi^*_{x,t}(k)$ and get:

$$\nabla h(\Pi^*_{x,t}(k)) = -\tilde{t}'_{xy}(\pi_{xy,t}) + t'_{xy}(\pi_{xy,t}).$$

Assumption 3 further implies that $0 \leq t'_{xy} \leq \rho$. Thus, $||\nabla h(\Pi^*_{x,t})|| \leq \rho$. From the definitions of W and W', we have $\nabla g(W) = \nabla g(W') + \nabla h(W') = 0$. Based on Lemma 14 in [30] and knowing that $g(\cdot)$ is η-strongly convex, the following inequality holds: $\langle \nabla g(W) - g(W'), W - W' \rangle \geq \eta ||W - W'||^2$. Thus, by the Cauchy-Schwartz inequality,

we obtain

$$||W - W'|| \cdot ||\nabla h(W')|| \geq (W - W')^T \nabla h(W') =$$
$$\langle \nabla g(W) - g(W'), W - W' \rangle \geq \eta ||W - W'||^2.$$

Dividing both sides by $\eta||W - W'||$ yields $||W - W'|| \leq \frac{1}{\eta}||\nabla h(W')|| \leq \frac{\rho}{\eta}$. From (5.13), we have $||W - W'|| = ||\varepsilon_x(k) - \varepsilon'_x(k)|| \leq \frac{1}{\eta}||\nabla h(W')||$. Thus, we obtain

$$\xi_x(k)(||\varepsilon_x(k)|| - ||\varepsilon'_x(k)||) \leq \xi_x(k)(||\varepsilon_x(k) - \varepsilon'_x(k)||) \leq \frac{\rho}{\eta}\xi_x(k).$$

Therefore, by choosing $\xi_x(k) = \frac{\eta}{\rho}\beta_p(k)$, the inequality $\xi_x(k)(||\varepsilon_x(k) - \varepsilon'_x(k)||) \leq \beta_p(k)$ holds. Thus, the output variable perturbation is β_p-differentially private for target node $x \in \mathscr{X}$. The proof follows identically for the perturbed output variable $\Pi^*_{y,s}(k)$ at the source node $y \in \mathscr{Y}$ and hence omitted. □

Bibliography

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] A. Abdel-Hadi and C. Clancy. A utility proportional fairness approach for resource allocation in 4G-LTE. In *International Conference on Computing, Networking and Communications*, pages 1034–1040, 2014.

[3] A. S. Awad, M. F. Shaaban, T. H. El-Fouly, E. F. El-Saadany, and M. M. Salama. Optimal resource allocation and charging prices for benefit maximization in smart PEV-parking lots. *IEEE Transactions on Sustainable Energy*, 8(3):906–915, 2016.

[4] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1998.

[5] S. Boyd, N. Parikh, and E. Chu. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers, 2011.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(29):1069–1109, 2011.

[8] H. Chen, M. Zhou, L. Xie, K. Wang, and J. Li. Joint spectrum sensing and resource allocation scheme in cognitive radio networks with spectrum sensing data falsification attack. *IEEE Transactions on Vehicular Technology*, 65(11):9181–9191, 2016.

[9] J. Chen and Q. Zhu. Security investment under cognitive constraints: A gestalt Nash equilibrium approach. In *52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, 2018.

[10] P. Coucheney, C. Touati, and B. Gaujal. Fair and efficient user-network association algorithm for multi-technology wireless networks. In *Proceedings of IEEE INFOCOM*, pages 2811–2815, 2009.

[11] K. Deb and C. Myburgh. A population-based fast algorithm for a billion-dimensional resource allocation problem with integer variables. *European Journal of Operational Research*, 261(2):460–474, 2017.

[12] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang. A differential privacy-based query model for sustainable fog data centers. *IEEE Transactions on Sustainable computing*, 4(2):145–155, 2017.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[14] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9. Foundations and Trends in Theoretical Computer Science, 2014.

[15] A. Galichon. *Optimal Transport Methods in Economics*. Princeton University Press, 2018.

[16] A. Garnaev and W. Trappe. Fair resource allocation under an unknown jamming attack: a Bayesian game. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 227–232, 2014.

[17] N. E. Gretsky, J. M. Ostroy, and W. R. Zame. The nonatomic assignment model. *Economic Theory*, 2(1):103–127, 1992.

[18] J. Hofbauer and S. Sorin. Best response dynamics for continuous zero–sum games. *Discrete & Continuous Dynamical Systems-B*, 6(1):215, 2006.

[19] L. Huang, J. Chen, and Q. Zhu. Distributed and optimal resilient planning of large-scale interdependent critical infrastructures. In *2018 Winter Simulation Conference (WSC)*, pages 1096–1107, 2018.

[20] J. Hughes and J. Chen. Fair and distributed dynamic optimal transport for resource allocation over networks. In *55th Annual Conference on Information Sciences and Systems (CISS)*, 2021.

[21] J. Hughes and J. Chen. Resilient and distributed discrete optimal transport with deceptive adversary: A game-theoretic approach. In *IEEE Control System Letters*, pages 1166–1171, 2022.

[22] J. Hughes and J. Chen. Differentially private admm-based distributed discrete optimal transport for resource allocation. In *IEEE Global Communications Conference: Communication & Information Systems Security*, Submitted, 2021.

[23] N. LeTien, A. Habrard, and M. Sebban. Differentially private optimal transport: Application to domain adaptation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2852–2858, 2019.

[24] V. S. Mirrokni, S. O. Gharan, and M. Zadimoghaddam. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1690–1701. SIAM, 2012.

[25] H. Nikaidô. On von Neumann's minimax theorem. *Pacific Journal of Mathematics*, 4(1):65–72, 1954.

[26] D. Niu and B. Li. An efficient distributed algorithm for resource allocation in large-scale coupled systems. In *Proceedings of IEEE INFOCOM*, pages 1501–1509, 2013.

[27] J. Pawlick and Q. Zhu. A mean-field stackelberg game approach for obfuscation adoption in empirical risk minimization. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 518–522, 2017.

[28] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[29] F. Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 55. Springer, 2015.

[30] S. Shalev-Shwartz. Online learning: Theory, algorithms, and applications. 2007.

[31] G. Shao, R. Wang, X.-F. Wang, and K.-Z. Liu. Distributed algorithm for resource allocation problems under persistent attacks. *Journal of the Franklin Institute*, 357(10):6241–6256, 2020.

[32] J. Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018.

[33] J. Solomon and A. Vaxman. Optimal transport-based polar interpolation of directional fields. *ACM Trans. Graph.*, 38(4), July 2019.

[34] C. Villani. Optimal transport – old and new. *Grundlehren der mathematischen Wissenschaften*, 338:xxii+973, 2008.

[35] C. You, K. Huang, H. Chae, and B.-H. Kim. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411, 2016.

[36] R. Zhang and Q. Zhu. Consensus-based distributed discrete optimal transport for decentralized resource matching. *IEEE Transactions on Signal and Information Processing over Networks*, 5(3):511–524, 2019.

[37] T. Zhang and Q. Zhu. Dynamic differential privacy for admm-based distributed classification learning. *IEEE Transactions on Information Forensics and Security*, 12(1):172–187, 2017.

[38] T. Zhang and Q. Zhu. Distributed privacy-preserving collaborative intrusion detection systems for VANETs. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):148–161, 2018.

[39] X. Zhang, M. M. Khalili, and M. Liu. Improving the privacy and accuracy of admm-based distributed algorithms. In *International Conference on Machine Learning*, pages 5796–5805. PMLR, 2018.

[40] Y. Zhang, Z. Hao, and S. Wang. A differential privacy support vector machine classifier based on dual variable perturbation. *IEEE Access*, 7:98238–98251, 2019.

Abstract

Jason Hughes

An Algorithmic Foundation for Fair, Secure, and Differentially Private Distributed Discrete Optimal Transport

Thesis Advised by Juntao Chen, Ph.D.

Optimal transport (OT) is a framework that can be used to facilitate the optimal allocation of resources in a network with multiple source and target nodes. To ease the computational complexity encountered by large-scale networks with a massive number of nodes, a distributed algorithm, based on the alternating direction method of multipliers (ADMM), is developed for computing the optimal transport strategy. However such a formulation lacks fairness, robustness and privacy considerations. Thus, there is an imperative need to develop distributed OT algorithms that allow for a more fair allocation of resources, accounts for possible deception attacks to the transport nodes and keeps nodes' sensitive information private during transport strategy updates. To achieve this goal, this thesis first incorporates a fairness metric into the objective function of the discrete OT problem and then leverages ADMM to develop a distributed algorithm. It then establishes a game-theoretic approach to counteract a deception attack where an attacker aims to compromise the transport plan. This formulation results in a min-max problem, and it can be solved in a distributed fashion to obtain a secure and resilient transport scheme. The distributed algorithms formed require communications on strategies between nodes during updates, which could potentially be intercepted and leveraged by an adversary, leading to private information being leaked. By incorporating differential privacy, the developed distributed algorithm guarantees the privacy of the sensitive information at each source and target node. All of the proposed algorithms are corrob-

orated through case studies. The developed algorithmic foundation for fair, secure, and privacy-preserving discrete OT has broad applications to economics, machine learning and more.

Vita

Jason Hughes is a graduate student in the Fordham University Graduate School of Arts and Sciences where he is studying Data Science and plans to obtain his Master's of Science in August 2021. His primary research interest is the applications of optimal transport to data driven tasks like machine learning, computer graphics and more. In 2020 he received his Bachelor's of Science in Mathematics from Fordham University. He held positions in the Fordham University Robotics and Computer Vision lab for almost two years where he researched applications of IMU data classification for obstacle detection in autonomous drones. In his spare time, he enjoys running, exploring New York City and being outside.