Distributed, Private, and Derandomized Allocation of Subsidized Goods

Hamid Reza Nabati Yazdi Zadeh

A Thesis in the Department of

Engineering & Computer Science

Presented in Partial Fulfillment of the Requirements

for the Degree of MASc of Quality Systems Engineering at

Concordia University

Montreal, Quebec, Canada

August 2017

CONCORDIA UNIVERSITY

Faculty of Engineering & Computer Science

This is to certify that the thesis prepared

By: Hamid Reza Nabati Yazdi Zadeh

Entitled: Distributed Allocation of Subsidized Goods in an Efficient and Private Approach

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

Dr. Anjali Awasthi	Chair
Dr. Ivan Contreras	Examiner
Dr. Chun Wang	Examiner
Dr. Jia Yuan Yu	Supervisor

Approved by Dr. Rachida Dssouli

September 6, 2017 Dr. Amir Asif Dean of Faculty

Abstract

Efficient resource allocation is challenging when privacy of users is important. Distributed solution approaches have recently been used extensively to find a solution for such problems. In this work, we study the efficiency of distributed AIMD algorithm for allocation of subsidized goods. To this end, we assign each user a suitable utility function describing the amount of satisfaction that it has from allocated resource. We define the resource allocation as a *total utilitarianism* problem that is an optimization problem of sum of users utility functions subjected to capacity constraint. Recently, a stochastic state-dependent variant of AIMD algorithm is used for allocation of common goods among users with strictly increasing and concave utility functions. We improve this algorithm to allocate subsidized goods to users with concave and nonmonotonous utility functions as well as users with quasi-concave utility functions. We also derandomize the AIMD algorithm and compare its efficiency with the stochastic version. We then model resource allocation problem as a competition game to evaluate the efficiency properties of unique equilibrium when resource allocation parameters change. To illustrate the effectiveness of the proposed solutions, we present simulation results for a public renewable-energy powered charging station in which the electric vehicles (EV) compete to be recharged.

Keywords– Distributed Resource Allocation, Total Utilitarianism, AIMD Algorithm, Game Theory, Electric Vehicle (EV) Charging

Contents

List of Figures

Notation

In this section, we summarize the notations that we use throughout the work. The notations are either standard or defined on related sections. Generally, we denote scalars and abstract objects with lowercase or uppercase letters (e.g., n or Γ), vectors by boldface letters (e.g., $\mathbf{x}$) and sets by uppercase letters (e.g., $\mathcal{N}$). Each user i's value is denoted by subscript letters (e.g., x_i) and each iteration t is located inside parenthesis (e.g., $x_i(t)$).

Symbol	**meaning**
$\mathbb{R}$	the set of real numbers
$\mathbb{R}_+$	the set of non-negative real numbers
$\mathcal{N}$	the set of users
i	each user's indicator
n	number of users
C	capacity constraint
x_i	user i's possible value of allocated resource
x_i^*	user i's optimal allocated resource
u_i	strictly increasing, concave, continuously differentiable utility function
v_i	concave, continuously differentiable utility function
w_i	strictly increasing, quasi-concave, continuously differentiable utility function
t	time steps (iterations)
T	last time steps (iterations)
$s(t)$	capacity constraint signal
$x_i(t)$	user i's possible value of allocated resource in iteration (t)
α	growth factor
β	drop factor
λ_i	the probability for each user i that depends on the long-term average allocated resource
Γ	the parameter that is chosen to ensure that $0 < \lambda_i(\bar{x}_i) < 1$
η_i	the slope of logarithmic as well as Sigmoidal utility functions

χ_i	in kWh is the amount of allocated resource (EV charging) that gives 100 unit utility to the user i
ψ_i	in kWh is the inflection point of Sigmoidal utility functions
Ψ	$\Psi = \sum_{i=1}^{n} \psi_i$
$\mathcal{N}$	a set of players (users)
X_i	non-empty set of available actions for player (user) i
x	a profile of actions
x_i^{ne}	user i's Nash equilibrium allocated resource
log	the natural logarithm

Introduction

In many real-world applications, the goal is allocating limited resources among users in order to achieve maximum total utilization or *total utilitarianism.* It normally leads to solve an optimization problem that the objective function is the sum of users utility functions subjected to capacity and other constraints; that is mathematically defined by

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{maximize}} \quad & \sum_{i=1}^{n} u_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i \leq C\,, \\ & x_i \geq 0\ ,\ i = 1, \ldots, n\,. \end{aligned} \tag{1}$$

The notion of utility function indicates how much each user benefits from the amount of resource that he or she is allocated. To solve this optimization problem, there are two main solution approaches of *centralized* and *distributed.* Centralized solutions are more efficient since users first admit their individual utility functions to a decision maker, which then solves the optimization problem of finding the optimal allocated resources. However, users utility functions are private information and the drawback is that users' privacy protections will be challenged.

Distributed allocation is a key concept to resolve this conundrum, *i.e.*, to allocate resources efficiently while preserving privacy. In distributed resource allocation, a set of users must autonomously assign their resources with respect to certain criteria and the main goal is to reach the *global optimum.* So, the solution to the resource allocation problem does not require any communication of the private utility functions.

Example 1. Imagine a charging station of Electric Vehicles (EVs) whose power supplies from renewable energy (e.g., solar, wind). Recent studies reveal that a fuel-driven vehicle can produce less greenhouse gas emissions than an EV if the recharging energy is entirely produced by coal-fired power plants. Therefore, local stations for charging EVs from renewable energy significantly contributes to achieve real environmental benefits [32]. Intuitively these stations have limited available resources and demand for these finite amounts is increasing. The users are EV owners who connect their vehicles to the charging station. A private utility function determines the level of satisfaction for each EV owner whose EV is connected to the station to be charged. As the demand for the resource overwhelms the capacity, every individual who consumes an additional unit directly harms others who can no longer enjoy the benefits. Since the return of EVs to charging station is non-deterministic, it seems reasonable to assume that EV owners are greedy and prefer to charge their own EVs regardless of others due to avoid range anxiety. We use logarithmic and Sigmoidal utility functions to represent the users' greediness. The logarithmic utility function is chosen because it is increasing and concave and therefore the total utility, that is sum of concave functions, mathematically has a global optimal solution. When the goal is considering the amount of resource that an EV needs for reaching to a predetermined destination, Step function is the ideal utility function. We, therefore, use the continuous Sigmoidal utility function in order to approximate this discontinuous Step function. Sigmoidal utility function is a increasing function with an inflection point that is convex for allocated resource below the inflection point and is concave for allocated resource above the inflection point.

In this work, we propose an extremely computationally efficient and private solution of the resource allocation problems. The solution is applicable for common goods such as clean air and access to public road and land where users do not pay a fee per use, and for subsidized goods where the fee per use is shared with the entire population. Note that subsidized goods generalize common goods, which as fully subsidized. The solution approach is *distributed and iterative*: each user requests an amount of resource that evolves over time steps according to its private utility function and whether the total demand exceed the available capacity. For the case of common goods, in [31] a stochastic and state-dependent variant of AIMD algorithm is used and it is shown that each user's requested resource converges quickly to the optimal allocation. Here, we also show that the derandomized version of such algorithm is efficient . As a concrete example of resource allocation problem that we will follow throughout of this work, we present simulation results for a public renewable-energy powered charging station network in which the electric vehicles (EV) compete to be recharged.

Distributed solution approach for solving the resource allocation problem has some other advantages in comparison to the centralized approach. First, there is no communication among users and therefore it significantly reduces the communication overhead. Second, the distributed approach by decentralizing decisions, is more robust than centralized approach since there is no single point of failure or hack. Third, the efficiency of the network is not dependent to number of users.

When users with private utility functions, greedily try to maximize their own utilities, game theory is appropriate to study interactions among them and their selfish behavior. The joint utilisation of a commonly owned resource, when the scarce resource of interest is easily available to all users, often causes the resource to be overused [10]. This concept, which in game theoretic approach named *the tragedy of the commons*, was popularized by Hardin (1968) in his seminal article [12]. In fact, individuals acting independently and rationally according to each user's self-interest behave contrary to the best interests of the whole group by depleting some common resource. Additionally, an exciting open problem is to improve the proposed approach to make it incentive compatible, thereby ensuring that each user will represent its utility function truthfully. Indeed, by misrepresenting their utility functions strategically, participants can receive a higher allocation, but hurt the utilitarian efficiency of the system as a whole.

Our Contributions

To address this problem, we study the class of distributed resource allocation and we make these following specific contributions:

- We propose AIMD algorithm to allocate subsidized goods to users with concave and nonmonotonous utility functions.
- We propose a derandomized version of AIMD algorithm to allocate common goods to users with strictly increasing, concave utility functions.
- We extend the results to propose a variant of AIMD algorithm to allocate common goods to users with quasi-concave utility functions.
- We then prove that our result of AIMD resource allocation of common goods where users utility functions are strictly increasing, concave is close to the stable result of Nash equilibrium in game theoretic approach.

The rest of this work is organized as follows. In Section , the problem is modeled as a utility-based resource allocation problem for both centralized and distributed solution approaches. We then represent different variants of AIMD resource allocation Algorithm among users with concave and nonmonotonous as well as Quasi-concave utility functions. This section ends by presenting the simulation results for the algorithms that are applied in . In Section , competition on scarce common resource is analyzed by solving resource allocation problem in game theoretic approach, and the numerical results is also presented. Finally, conclusions are summarized in section .

Distributed Resource Allocation

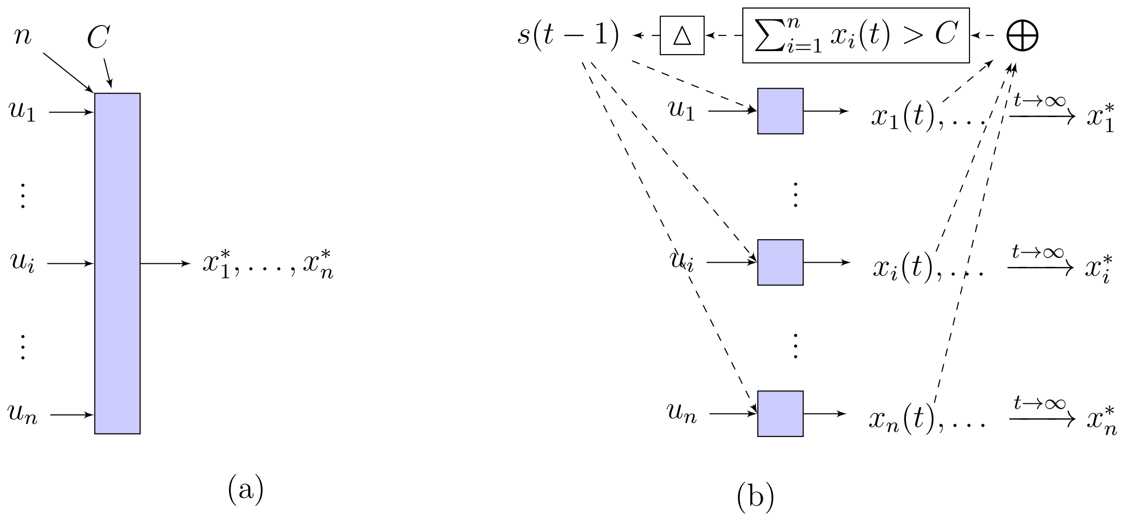
In this section, we formally define resource allocation problem using utility function concept for both centralized and distributed solution approaches. We then propose variants of AIMD distributed algorithm for allocation of subsidized (and common) goods between users based on their specific utility functions. The objective is to determine each user's optimal allocated resource at which maximum total utility is achieved. We also include numerical simulations of total utilitarianism for EV owners who connected their vehicles to a solar-powered charging station to be charged in order to validate the convergence of our solutions.

Remark 1. In real life applications, the resource could be time-slot like energy in kWh or time-varying like power in kW. Although, we defined the optimization problem to be solved in each time-slot, we can use the proposed solution for any time-varying situations without changing the results.

Baseline, Problem Formulation and Objective

Consider n users utilize a limited shared resource $C > 0$, and suppose $x_i \geq 0$ represents the possible amount of resource consumed by each user $i = 1, \ldots, n$. We attribute a *utility*, *i.e.*, a measure of satisfaction, to each user i who takes advantage of the common resource and describe it by means of a *utility function.* The utility function $u_i : \mathbb{R}_+ \to \mathbb{R}_+$, assigns a non-negative real number to each possible value of allocated resource x_i, to represent the level of satisfaction for each user i or quality of service (QoS).

Figure 1a represents a class of centralized resource allocation problems in which a central decision maker calculates the optimal solution $x_1^*, ..., x_n^*$, by collecting all information regarding each user's utility function u_i, capacity constraint C, and number of users n. Therefore, centralized resource allocation problem can be expressed as nonlinear continuous optimization problem (1). Although centralized solution approaches focus to determine efficient resource allocation, in many realistic applications, it is neither applicable nor desirable [22] since it violates users' privacy. Figure 1(b) depicts a class of *distributed (and iterative)* approach to



n
C
u_1
u_i
u_n
$x_1^*, \dots, x_n^*$
(a)
$s(t-1)$
Δ
$\sum_{i=1}^n x_i(t) > C$
$\bigoplus$
$x_1(t), \dots \xrightarrow{t\to\infty} x_1^*$
$x_i(t), \dots \xrightarrow{t\to\infty} x_i^*$
$x_n(t), \dots \xrightarrow{t\to\infty} x_n^*$
(b)

Figure 1: Resource allocation solution approaches, (a) centralized , (b) distributed iterative.

resource allocation problems in which allocations emerge as the result of an iterative of local procedures. In other words, a set of users locally make decisions regarding their resources autonomously. To this end, an algorithm is used to assign each user i an allocated resource $x_i(t)$ in time steps (iterations) $1, \dots, t$. In each iteration, each user's algorithm update user's allocated resource $x_i(t)$ locally by choosing one of these options: increase, decrease or no-change compared with previous iteration $x_i(t-1)$. The increase option continues until receiving one bit signal $s(t-1)$, that notify capacity constraint $\sum_{i=1}^n x_i(t) > C$ is violated and algorithm, based on a certain probability, choose one of the following options : decrease or no-change. When the capacity is available again $\sum_{i=1}^n x_i(t) \leq C$, the increase option of the algorithm restarts immediately. The procedure repeats until the number of iterations is large enough t, and users' allocated resource converge to the optimal allocation $x_1^*, ..., x_n^*$.

In order to quantify efficiency of distributed resource allocation, inspired from the notion of *Price of Anarchy* in game theory, we express the efficiency as the ratio between the output of distributed algorithm and the solution of Equation (1) as follows:

$$\text{efficiency} = \frac{\lim_{t\to\infty} \sum_{i=1}^n u_i(x_i(t))}{\sum_{i=1}^n u_i(x_i^*)} \,. \tag{2}$$

Common Goods

For the sake of retaining simplicity, we first focus on modeling resource allocation of common goods, where users do not pay for their allocated resources. We also consider some further and substantial assumptions, briefly called *concavity assumption*, for users utility functions which provide mathematical tractability of optimization problem (1) however limit its applicability.

Assumption 1. (Concavity Assumption) The utility functions $u_i : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, (i) are strictly increasing functions of x_i with $u_i(0) = 0$, (ii) are concave and continuously diffrentiable with domain $x_i \geq 0$. Where x_i is the amount of resources allocated to user i.

The optimization problem (1) under concavity Assumption 1 for users utility functions, is a convex optimization problem. The objective function, which is the sum of concave functions, is therefore concave and each constraint defines a convex set. Consequently, for a convex optimization problem, there exists a unique tractable global optimal solution [4].

Example 2. In the case of charging EVs, we use normalized logarithmic function as strictly increasing concave utility function that satisfies concavity Assumption 1 in order to model the level of satisfaction of EV owners whose car is connected to the charging station to be charged. Logarithmic utility function u_i means that each EV owner's satisfaction continuously increases when the amount of charging or receiving the resources x_i increase. Therefore, normalized logarithmic utiliy function is expressed as:

$$u_i(x_i) = 100\frac{\log(1 + \eta_i x_i)}{\log(1 + \eta_i \chi_i)}\,, \tag{3}$$

where χ_i in kWh is the amount of allocated resource (EV charging) that gives 100 unit utility to the user i. Moreover, in the lack of resource the utility function value is zero. So, normalized logarithmic utility function satisfies $u_i(0) = 0$ and $u_i(\chi_i) = 100$. The parameter η_i indicates how the charge needed urgently by effecting on the rate of utility percentage that is a function of allocated resource x_i. Intuitively higher values of η_i yield higher utility to user i. Figure 2 represents two normalized logarithmic utility functions u_i with $\eta_i = 0.11$, $\chi_i = 60$ and $\eta_i = 24.9$, $\chi_i = 100$.

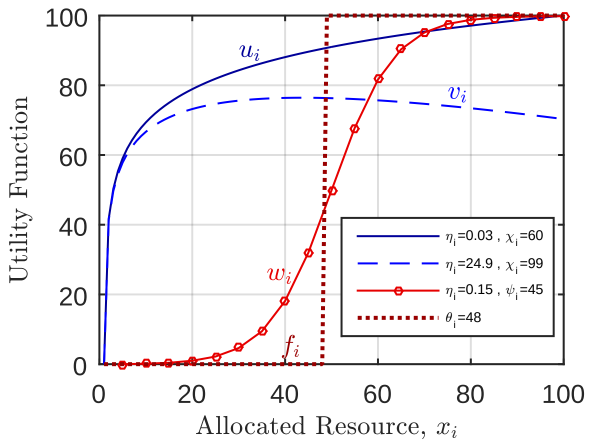


Figure 2: Utility functions: normalized logarithmic (strictly-increasing concave) utility functions u_i, compared with corresponding nonmonotonous payoff functions v_i when $L = 0.3$ as well as discontinuous Step utility function f_i and an approximate continuous Sigmoidal (quasi-concave) utility function w_i .

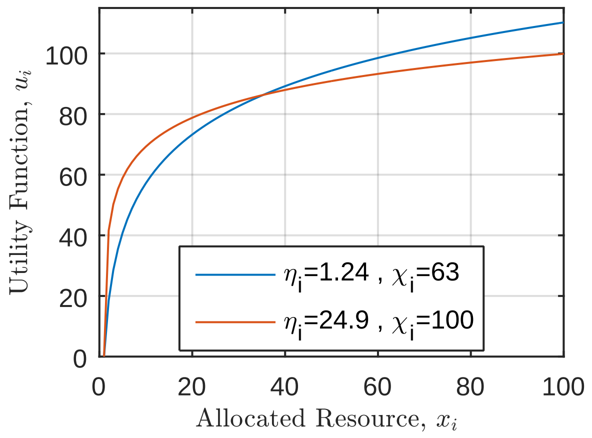


Figure 3: Two normalized logarithmic (strictly-increasing concave) utility functions u_i , with different values of η_i and χ_i .

Remark 2. AIMD *(Additive Increase Multiplicative Decrease)* is a distributed and iterative algorithm that is used widely to control congestion in computer networks. The objective of AIMD is to determine the share of the resource for each user while total demands remains less than the available capacity, and where the limited communication in the network is desired as well as privacy protection is considered. The AIMD algorithm, in its basic version, is composed of two procedures. In the additive increase (AI) phase, users continuously request for more available resource of the network until receiving a notification that the aggregate amount of available resource has been exceeded. Then the multiplicative decrease (MD) phase occurs and users respond to the notification by reducing their share proportionally. The AI phase of the algorithm restarts again immediately and this pattern is repeated by each active user in the network [7].

Herein, we use a *stochastic allocated-dependent* version of AIMD Algorithm to solve optimization problem (1). We shall not describe the theorems and proofs of the algorithm here, rather we refer the interested readers to [31] for details. In AI phase, each active user i continue to update its allocated resource $x_i(t)$ upward by adding an amount of *growth factor* $\alpha \in (0, C)$ to its previous allocated resource $x_i(t-1)$ while $\sum_{i=1}^{n} x_i \leq C$. When the capacity limit has been violated, *i.e.*, $\sum_{i=1}^{n} x_i > C$, users are notified to execute MD phase. Each user will respond to the capacity signal independently with a certain probability λ_i , by multiplying the previous allocated resource $x_i(t-1)$ to a *drop factor* $\beta \in (0, 1)$ to form current allocated resource $x_i(t)$.

The probability λ_i at t-th iteration, for each user i, depends on the long-term average allocated resource $\bar{x}_i(t)$ through the relation $\lambda_i(\bar{x}_i(t)) = \Gamma \frac{u_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$, where the parameter Γ is chosen to ensure that $0 < \lambda_i(\bar{x}_i) < 1$. Therefore, the AIMD stochastic allocated-dependent algorithm used to solve the resource allocation problem (1) is showed as follows.

Algorithm 1 AIMD [31] for user i

```
1: Initialize $x_i(0)$ arbitrary
2: Broadcast the parameter $\Gamma$
3: for time steps $t = 1, 2, 3, \ldots$ do
4:     if $\sum_{j=1}^{n} x_j(t) < C$ then
5:         $x_i(t+1) = x_i(t) + \alpha$;
6:     else
7:         $x_i(t+1) = \beta x_i(t)$ with probability $\lambda_i(\bar{x}_i(t)) = \Gamma \frac{v_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$
8:         $x_i(t+1) = x_i(t)$ otherwise;
9:     end if
10: end for
```

Note that the parameter Γ depends on the worst utility function that is independent of number of users. It must be communicated to all users prior to the algorithms use by a central authority [7].

Derandomized Distributed Resource Allocation

In section , we introduced a stochastic version of AIMD algorithm for an efficient allocation of common goods between users. The probabilistic method can also yield insight into how to construct deterministic algorithms [23]. We now propose a variant of *deterministic* AIMD for the same purpose. We show that the strong convergence of derivative of utility function of long-term average allocated resource $u_i'(\bar{x}_i(t))$, can be used to allocate resource optimally. So, we define $\lambda_i(\bar{x}_i(t)) = \Gamma\frac{u_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$ and we use it in MD phase of the algorithm by $x_i(t+1) = \beta(1-\lambda_i)x_i(t) + \lambda_i x_i(t)$ to build DAIMD Algorithm 2.

Algorithm 2 DAIMD for user i

```
1: Initialize $x_i(0)$ arbitrary
2: Broadcast the parameter $\Gamma$
3: for time steps $t = 1, 2, 3, \ldots$ do
4:     if $\sum_{j=1}^{n} x_j(t) < C$ then
5:         $x_i(t+1) = x_i(t) + \alpha$;
6:     else
7:         $x_i(t+1) = \beta(1-\lambda_i)x_i(t) + \lambda_i x_i(t)$, where $\lambda_i(\bar{x}_i(t)) = \Gamma\frac{u_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$
8:     end if
9: end for
```

Subsidized Goods

We extend resource allocation problem to subsidized goods where the fee per use is shared with the entire population. Suppose if each user i is charged a constant price L per unit of the received resources x_i. Each user payoff function, $v_i : \mathbb{R}_+ \to \mathbb{R}_+$ is defined as utility function minus the cost of received resource as follows:

$$v_i(x_i) = u_i(x_i) - Lx_i\,. \tag{4}$$

Recall each user utility function $u_i(x_i)$ is considered under concavity Assumption 1, therefore, each user payoff function (4) is a concave function but it is not necessarily increasing. The

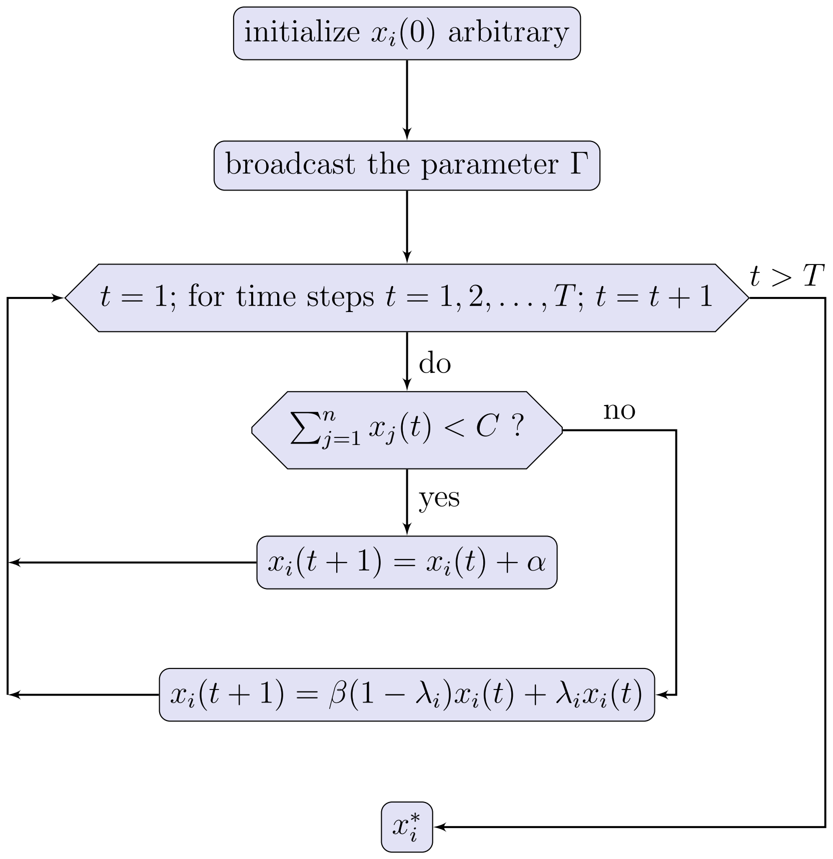

initialize $x_i(0)$ arbitrary
broadcast the parameter Γ
$t = 1$; for time steps $t = 1, 2, \ldots, T$; $t = t + 1$
$t > T$
do
$\sum_{j=1}^{n} x_j(t) < C$?
no
yes
$x_i(t+1) = x_i(t) + \alpha$
$x_i(t+1) = \beta(1 - \lambda_i)x_i(t) + \lambda_i x_i(t)$
x_i^*

Figure 4: Distributed and derandomized algorithm to efficient and private allocation of subsidized goods to each user i, with concave and nonmonotonous utility function. The parameter λ_i at t-th iteration, for each user i, depends on the long-term average allocated resource $\bar{x}_i(t)$ through the relation $\lambda_i(\bar{x}_i(t)) = \Gamma \frac{u_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$, and the parameter Γ is chosen to ensure that $0 < \lambda_i(\bar{x}_i) < 1$. The parameters $0 < \alpha < C$ and $0 < \beta < 1$ are growth factor and drop factors repectively [31].

centralized resource allocation problem can then be formulated as follows:

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{maximize}} \quad & \sum_{i=1}^{n} v_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i \leq C \,, \\ & x_i \geq 0 \,,\; i = 1, \ldots, n \,. \end{aligned} \tag{5}$$

The optimization problem 5 in which the objective function is non-negative sum of concave functions, is concave and there exist a global optimal solution [4].

Example 3. In the case of charging EVs, we use normalized logarithmic function Equation (3) to build following payoff function:

$$v_i(x_i) = 100\frac{\log(1+\eta_i x_i)}{\log(1+\eta_i \chi_i)} - Lx_i\,. \tag{6}$$

Figure 2 represents normalized logarithmic utility functions u_i with $\eta_i = 24.9$, $\chi_i = 99$ compared with corresponding payoff functions v_i when $L = 0.3$.

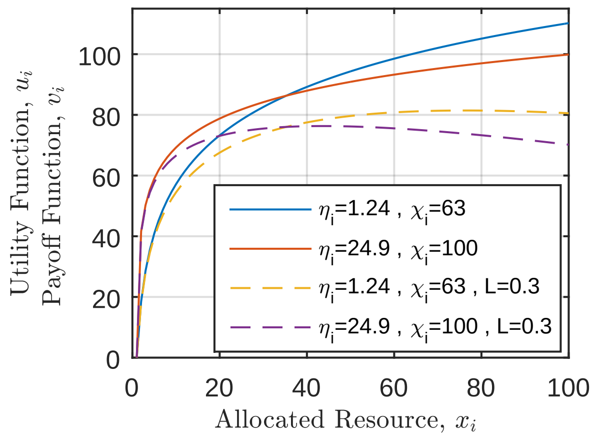


Figure 5: Two normalized logarithmic (strictly-increasing concave) utility functions u_i (solid lines), compared with corresponding non-increasing payoff functions $v_i(x_i) = u_i(x_i) - Lx_i$ (dashed lines) when $L = 0.3$, with different values of η_i and χ_i .

The AIMD algorithm is invented for rate control and in our model it tries to continue allocating whole resource to users without considering maximum users' utility. We, Therefore, conclude that the AIMD Algorithm 1 has no efficient solution among users with non-increasing utility functions. In other words, from a specific point the allocation of resources not only do not increase the utility but also decrease it.

We improve AIMD Algorithm 1 by controlling the allocation do not exceed from maximum payoff of each user and design the PAIMD Algorithm 3. The control is applied locally since each user i calculates the optimal point $x_i^* = \underset{x_i \in \mathbb{R}_+}{\arg\max}\ v_i(x_i)$, $\forall i = 1, \ldots, n$ and then in each iteration, in the (AI) phase of the algorithm compare it toallocated resource $x_i(t) + \alpha$ to choose the minimum allocation.

Algorithm 3 PAIMD for user i

```
1:  Initialize $x_i(0)$ arbitrary
2:  Each user $i$ calculates $x_i^* = \arg\max_{x_i \in \mathbb{R}_+} \; v_i(x_i), \; \forall i = 1, \dots, n$
3:  Broadcast the parameter $\Gamma$
4:  for time steps $t = 1, 2, 3, \dots$ do
5:      if $\sum_{j=1}^{n} x_j(t) < C$ then
6:          $x_i(t+1) = \min(x_i^* \, , \; x_i(t) + \alpha)$
7:      else
8:          $x_i(t+1) = \beta x_i(t)$ with probability $\lambda_i(\bar{x}_i(t)) = \Gamma \frac{v_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$
9:          $x_i(t+1) = x_i(t)$ otherwise;
10:     end if
11: end for
```

Quasi-Concave Utility Functions

In this section, we try to go beyond convex optimization problems, by eliminating concavity from Assumption 1. Therefore, we model resource allocation problem of users with quasi-concave utility functions by choosing versatile Sigmoidal functions. The intuition behind this utility shape is that low values of allocated resource offer very low increase in degree of satisfaction to the user. As the allocated resource continues, user satisfaction increases rapidly until a point where saturation appears and remains sharp after it. User satisfaction again increase slowly when allocated resource continues toward far away saturation point. So, the Sigmoidal function is defined as following:

Definition 1. The utility function of $w_i : \mathbb{R}_+ \to \mathbb{R}_+$ is defined to be Sigmoidal if: (i) The $w_i(0) = 0$ and u_i is strictly increasing function of x_i. (ii) $w_i(x_i)$ is continuously differentiable, with domain $x_i \geq 0$. (iii) $w_i(x_i)$ is convex for $x_i \leq \psi_i$ and is concave for $x_i \geq \psi_i$, which $\psi_i \in \mathbb{R}_+$ is the inflection point.

Example 4. (Why do we need Sigmoidal utility functions?). In some situations, such as charging an electric vehicle with the goal of reaching a predetermined destination (e.g., airport, home, etc.), the user receive negligible (or non) utility until a threshold of resource is reached (e.g., enough electric charge to arrive at the destination). Ideally, in this situations the best description of the utility function is through a discontinuous Step function as follows:

$$f_i(x_i) = \begin{cases} 0 & \text{if } x_i < \theta_i \, ; \\ 100 & \text{if } x_i \geq \theta_i \, , \end{cases} \tag{7}$$

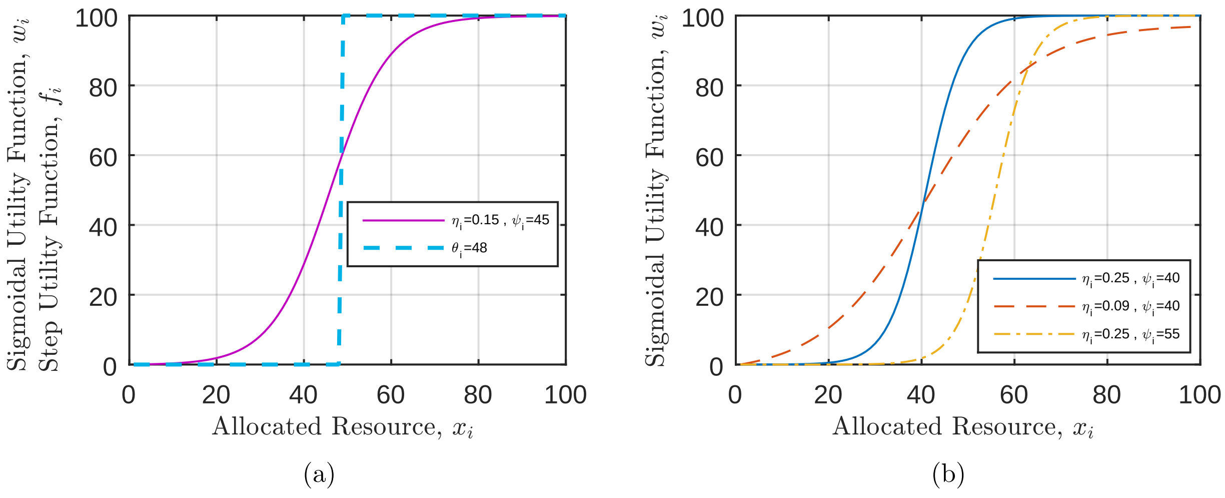


Figure 6: (a) A discontinuous Step utility function f_i and an approximate continuous Sigmoidal utility function w_i , (b) Three Sigmoidal (Quasi-concave) utility fictions $w_i(x_i)$ with different values of η_i and ψ_i .

where θ_i shows the sufficient allocated resource that gives 100 unit utility to user i.

Continious Sigmoidal utility functions may be used to approximate a step utility function to any arbitrary accuracy [30]. Likewise, the user receives negligible additional utility, once a threshold of resource is reached. We now model EV owner satisfaction with Sigmoidal utility function that are expressed by:

$$w_i(x_i) = \frac{100}{1 + e^{-\eta_i(x_i - \psi_i)}} - \frac{100}{1 + e^{\eta_i \psi_i}} , \tag{8}$$

where η_i is the steepness of the curve that indicates how the charge is needed urgently for each user i. The parameter ψ_i in kWh is the inflection point of the function that achieving it satisfies the urgent need of user i to resource. The function satisfies $w_i(0) = 0$ and $\lim_{x_i \to \infty} w_i(x_i) = 100$.

Figure 2 represents a Step utility function f_i with $\theta_i = 48$ and an approximate corresponding Sigmoidal utility function w_i with $\eta_i = 0.15$ and $\psi_i = 45$.

The QAIMD Algorithm 4 represents the procedure of efficient allocation among users with sigmoidal utility functions. The key point is that in each iteration (t), the long-term of allocated resource $\bar{x}_i(t)$ is compared with each user i inflection point ψ_i. If $\bar{x}_i(t) < \psi_i$, the increase phase is built by multiplying the previous state $x_i(t)$ in a growth factor $\frac{1}{\beta} > 1$ to construct current state $x_i(t+1)$ with a probability $\lambda_i(\bar{x}_i(t)) = \Gamma_1 \frac{w_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$. The decrease phase also is made by subtracting α from the previous state. When $\bar{x}_i(t) \geq \psi_i$, the algorithm is work with AIMD Algorithm 1 procedure. Note that there are two parameters Γ_1, Γ_2 to ensure $0 < \lambda_i(\bar{x}_i) < 1$ in each case.

Algorithm 4 QAIMD for user i

```
1: Initialize $x_i(0)$ arbitrary
2: Broadcast the parameters $\Gamma_1$, $\Gamma_2$
3: **for** time steps $t = 1, 2, 3, \dots$ **do**
4:     **if** $\bar{x}_i(t) < \psi_i$ **then**
5:         **if** $\sum_{j=1}^{n} x_j(t) < C$ **then**
6:             $x_i(t+1) = \frac{1}{\beta} x_i(t)$ with probability $\lambda_i(\bar{x}_i(t)) = \Gamma_1 \frac{w_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$
7:             $x_i(t+1) = x_i(t)$ otherwise;
8:         **else**
9:             $x_i(t+1) = \max(0\ ,\ x_i(t) - \alpha)$
10:        **end if**
11:    **else**
12:        do AIMD with $\alpha$, $\beta$ and probability $\lambda_i(\bar{x}_i(t)) = \Gamma_2 \frac{w_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$
13:    **end if**
14: **end for**
```

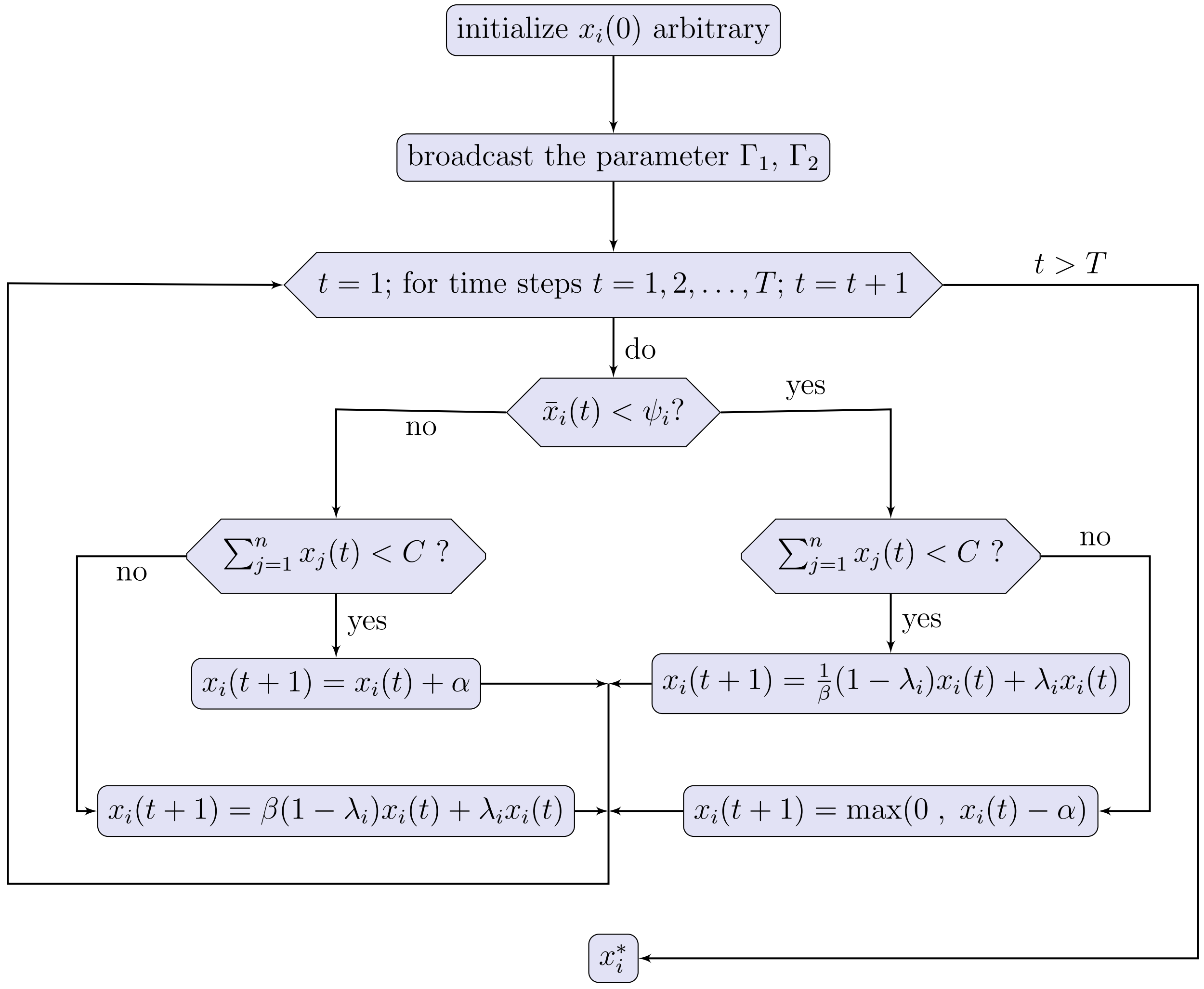

initialize $x_i(0)$ arbitrary
broadcast the parameter Γ_1, Γ_2
$t=1$; for time steps $t=1,2,\ldots,T$; $t=t+1$
$t>T$
do
$\bar{x}_i(t)<\psi_i$?
yes
no
$\sum_{j=1}^{n} x_j(t) < C$?
no
yes
$\sum_{j=1}^{n} x_j(t) < C$?
no
yes
$x_i(t+1)=x_i(t)+\alpha$
$x_i(t+1)=\frac{1}{\beta}(1-\lambda_i)x_i(t)+\lambda_i x_i(t)$
$x_i(t+1)=\beta(1-\lambda_i)x_i(t)+\lambda_i x_i(t)$
$x_i(t+1)=\max(0\ ,\ x_i(t)-\alpha)$
x_i^*

Figure 7: Distributed and derandomized algorithm to efficient and private allocation of common goods to each user i, with quasi-concave utility function. The parameter λ_i at t-th iteration, for each user i, depends on the long-term average allocated resource $\bar{x}_i(t)$ through the relation $\lambda_i(\bar{x}_i(t)) = \Gamma\frac{u_i'(\bar{x}_i(t))}{\bar{x}_i(t)}$, and the parameter Γ is chosen to ensure that $0<\lambda_i(\bar{x}_i)<1$. The parameters $0<\alpha<C$ and $0<\beta<1$ are growth factor and drop factors repectively [31].

Simulation

In order to figure out the effectiveness of variants of AIMD Algorithm, we simulate them in different cases of charging electric vehicles (EVs). We then calculate the efficiency of each algorithm by comparing the results with an optimal centralized solution, in MATLAB.

Unless otherwise specified, the general setting of simulation is as follows. The resource allocation domain is considered as a charging station whose power supplies from renewable energy (e.g. solar or wind), with constant capacity of C in kWh. We adjust the capacity to be %65 of the sum of users utility functions when each user receives 100 unit satisfaction. The users $n = 50$ are the EV owners who connected their vehicles to the station for charging at the same time. Each EV owner i comprise its own utility function of u_i that corresponds the amount of charge x_i which its vehicle receives. We define the utility u_i, of a greedy EV owner i, to be increasing to the charging status x_i in kWh at which his EV is charged. In other words the EV owner's satisfaction is increasing to the extent that their EV is charged. Therefore, we use both logarithmic and Sigmoidal utility functions.

We also set the parameters $\alpha = 1$ and $\beta = 0.85$ while executing AIMD algorithm. In addition, the parameter Γ is chosen to assure us the condition $\lambda_i(\bar{x}_i) \in (0, 1)$ is satisfied.

Concave Utility Functions

To model the problem, we adopt normalized logarithmic utility function Equation (3) as a strictly increasing concave function which satisfies Concavity Assumption 1. We choose χ_i independent uniformly distributed random number with support $(40, 60)$ and η_i independent uniformly distributed random number with support $(0, 1)$.

We apply deterministic DAIMD Algorithm 2 for allocation of power as a common good (no charging fee) to EVs who connected to station to be charged. Figure 8a shows a rapid convergence for derivative of payoff functions $u_i'(\bar{x}_i(t))$ when iteration t increases. Figure 8b depicts the value of long-term average state $\bar{x}_i(t)$ for six randomly selected users and shows each of them converge to a stable value that is x_i^*. Figure 8c reveals the coincidence of deterministic and stochastic versions of derivative of payoff functions $u_i'(\bar{x}_i(t))$. Figure 8d also represents that deterministic and stochastic version of average state $\bar{x}_i(t)$ fluctuate differently but the long-term averages for each user converge to optimal allocation. The efficiency of deterministic DAIMD Algorithm 2, calculated by Equation (2), in different runs are a real number in the range of $(0.97, 0.99)$.

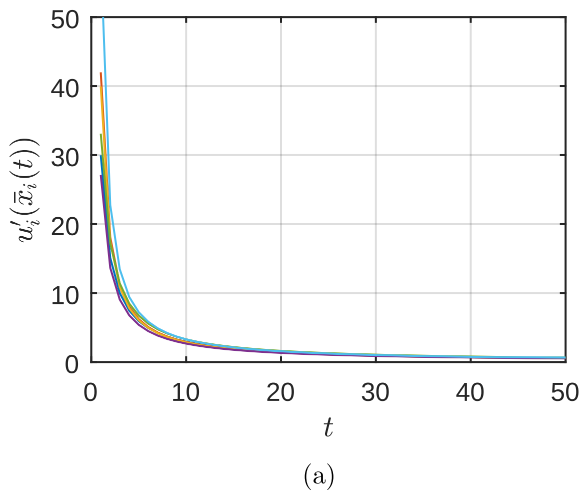
We apply stochastic PAIMD Algorithm 3 for allocation of power as a subsidized good (with charging fee) to EVs who connected to station to be charged. Therefore, we consider the price per unit $L \in \{0.1, 0.2, \ldots, 1\}$ per unit of the power x_i in the payoff function (6). The simulation results in Figure 9a reveals a rapid convergence for derivative of payoff functions $v_i'(\bar{x}_i(t))$ when iteration t increases. Figure 9b represents the value of long-term average state $\bar{x}_i(t)$ for six randomly selected users and shows each of them converge to a stable value that is x_i^*. Figure 9c, depicts the efficiency of AIMD Algorithm 1, PAIMD Algorithm 3 that is calculated by Equation (2). It shows that PAIMD Algorithm 3 has better performance when L increases compared with AIMD Algorithm 1.

Quasi-Concave Utility Functions

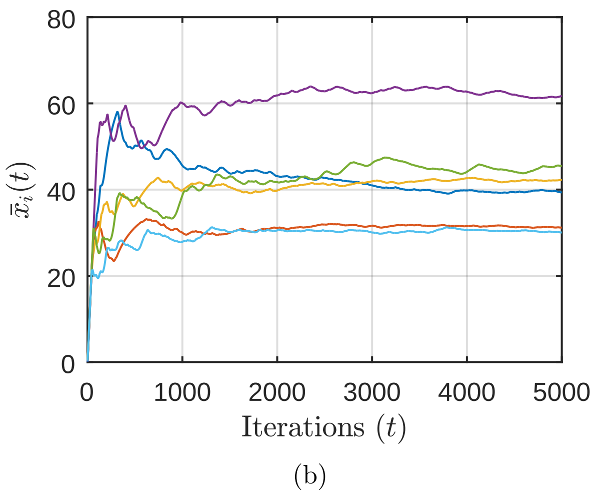
We now model EV owner satisfaction with Sigmoidal utility function that are expressed by Equation (8). We choose ψ_i independent uniformly distributed random number with support $(25, 100)$ and η_i independent uniformly distributed random number with support $(0, 25)$.

Figure 10a depicts the derivative of utility functions $w_i'(\bar{x}_i(t))$ for six randomly selected users. It illustrates that the derivatives approach to zero as t increase but the convergence is slower than the derivatives of logarithmic utility function $v_i'(\bar{x}_i(t)$ in Figure 9a . In Figure 10b the average of allocated resource $\bar{x}_i(t)$ for six randomly selected users is displayed. It shows $\bar{x}_i(t)$ approach to a constant number that is optimal allocated resource x_i^*.

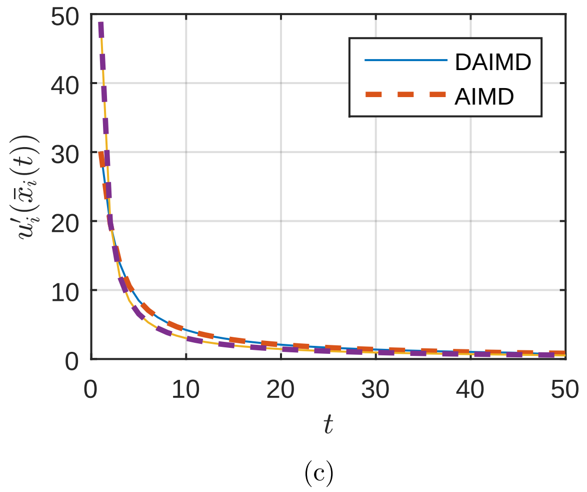
Figure 10c represents the efficiency of QAIMD 4, calculated by Equation (2), for different capacity $C/\Psi = \{0.5, 0.75, \ldots, 3\}$, where $\Psi = \sum_{i=1}^{n} \psi_i$. For each user i the algorithm decides between increasing allocated resource or decreasing it toward zero. The efficiency of the algorithm is better for small values of capacity constraint, but it decrease when capacity is around Ψ. The efficiency improve again when it is large enough.



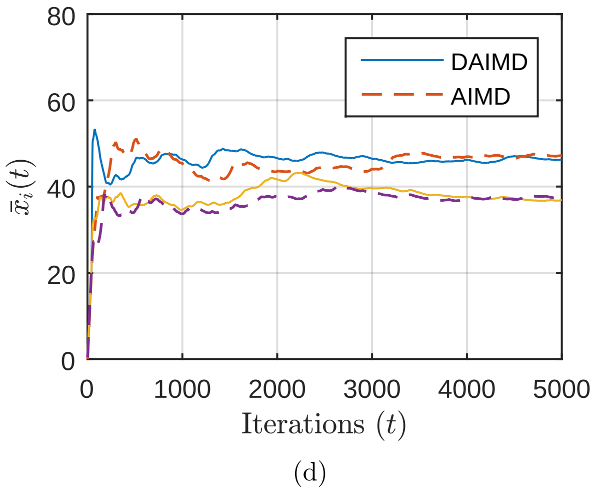
(a)



(b)

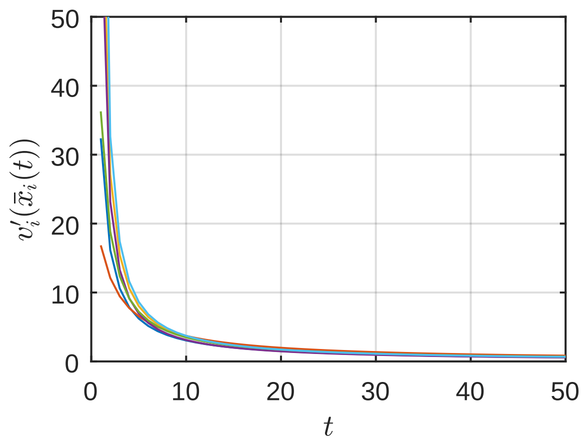


(c)

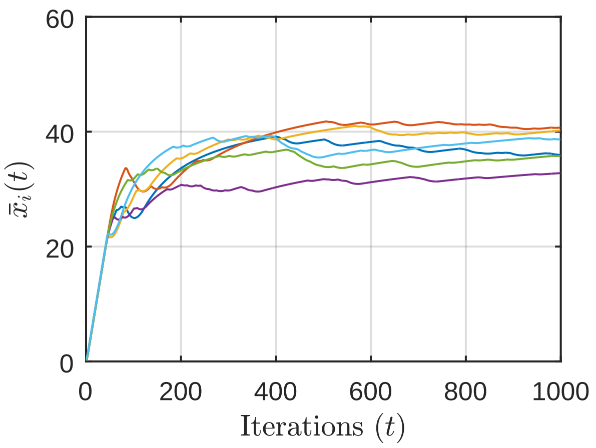


(d)

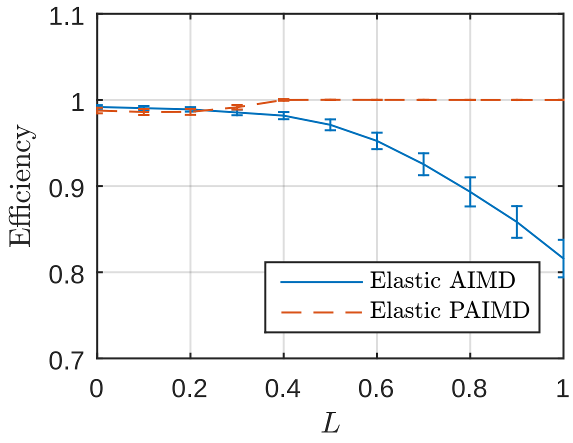
Figure 8: DAIMD Algorithm 2, (a) The deterministic derivative of payoff function $u_i'(\bar{x}_i(t))$ for six randomly selected users, (b) The deterministic average of allocated resource $\bar{x}_i(t)$ to the optimal point for six randomly selected users, (c) the deterministic derivative of payoff function $u_i'(\bar{x}_i(t))$ for two randomly selected users (solid lines) compared with corresponding stochastic ones (dashed lines) , (d) The deterministic average of allocated resource $\bar{x}_i(t)$ to the optimal point for two randomly selected users (solid lines) compared with corresponding stochastic ones (dashed lines).



(a)

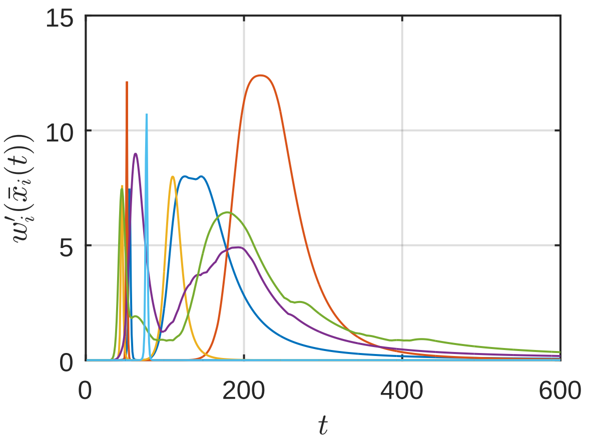


(b)

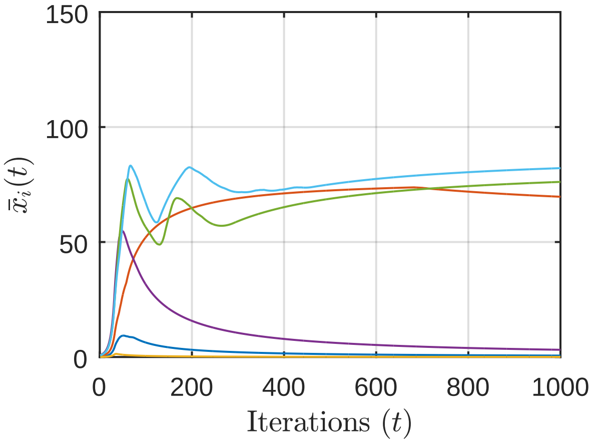


(c)

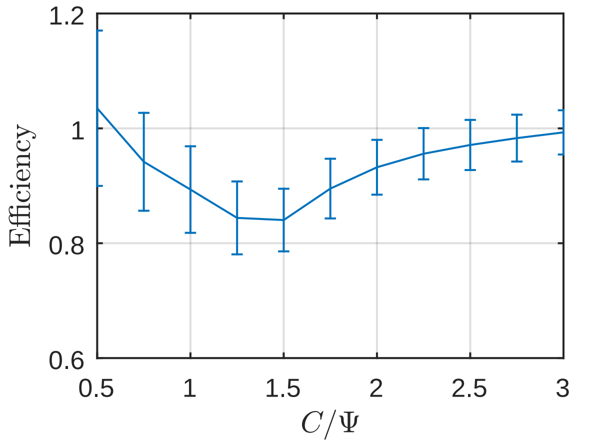
Figure 9: (a) The derivative of payoff function $v_i'(\bar{x}_i(t))$ for six randomly selected users when $L = 0.3$, (b) the average of allocated resource $\bar{x}_i(t)$ to the optimal point for six randomly selected users when $L = 0.3$, (c) the efficiency of AIMD Algorithm 1 and PAIMD Algorithm 3 for $L \in \{0, 0.1, \dots, 1\}$ calculated by Equation (2).



(a)



(b)



(c)

Figure 10: (a) The derivative of utility functions $w_i'(\bar{x}_i(t))$ for six randomly selected users when $C/\Psi = 1.5$, (b) the average of allocated resource $\bar{x}_i(t)$ to the optimal point for six randomly selected users when $C/\Psi = 1.5$, (c) the efficiency of QAIMD Algorithm 4 calculated by Equation (2).

Loss of Efficiency Due to Competition

In this section, we allow the individual users to act strategically as in a game. We consider a game in strategic form, where all users' utility functions are common knowledge. The resulting competition over a scarce resource is reminiscent of the tragedy of the commons []. An user may deviate from the AIMD algorithm and strategically request more resource in order to improve its payoff. Alternatively, an user may follow the AIMD algorithm but mispresent its utility function. However, we show that, in some situations, the AIMD outcome and the game's Nash equilibrium are close to each other.

Resource Allocation as a Strategic Game

Imagine a resource allocation problem in which there are n users, competing to utilize a scarce fixed common resource of $C > 0$. Each user i chooses his own consumption of resources x_i from a set of action space $X_i = \{x_i \in \mathbb{R} \mid 0 \leq x_i \leq C\}$. A profile of actions $x = (x_i, x_{-i})$ describe a particular combination of actions chosen by all users, where $x_{-i} \in X_{-i}$ is a particular possible of actions for all players who are not i.

Consuming an amount $x_i \geq 0$ gives user i a benefit equal to $u_i(x_i)$ when $\sum_{j=1}^{n} x_j \leq C$ and intuitively no other users benefits from i's choice. When x_i increases or other users consume more resources so that $\sum_{j=1}^{n} x_j > C$, the user get nothing $u_i(x_i) = 0$ because additional requested resources are not provided. Then we define the payoff function $\tilde{u}_i(x_i, x_{-i})$ of a user i from a profile of actions x as

$$\tilde{u}_i(x_i, x_{-i}) = \begin{cases} u_i(x_i) & \text{if } \sum_{j=1}^{n} x_j \leq C\,; \\ 0 & \text{if } \sum_{j=1}^{n} x_j > C\,. \end{cases} \tag{9}$$

Where the utility function $u_i(x_i)$ is considered to be concave, strictly increasing, and continuously differentiable ,*i.e.*, follows assumption 1.

The strategic game $(\mathcal{N}, X_i, \tilde{u}_i)_{i \in \mathcal{N}}$, that have infinitely many pure strategies but utility functions are not continuous, is discontinuous infinite strategic games. This problem should be consider precisely because it may lead to problem of nonexistence of unique Nash equilibrium.

Nash Equilibrium

To cut to the chase, the key notion to solve the *strategic game* $(\mathcal{N}, X_i, \tilde{u}_i)_{i \in \mathcal{N}}$, is the *Nash equilibrium*, that is an outcome (a decision made by each player) such that no player can improve his individual payoff through an unilateral move. As stable situations, Nash equilibrium are often considered to be the expected outcomes from interactions. To solve for a Nash equilibrium we compute the best-response function correspondence for each player and then find an action profile for which all best-response functions are satisfied together.

To find a solution for Equation (9), we first write out each player i's best-response correspondence and we consider that given x_{-i}, player i will want to choose an element in $BR_i(x_{-i})$. Given $x_{-i} \in X_{-i}$ each player i's best response is the difference between C and $\sum_{j \neq i}^{n} x_j$. If user i asks for more, then all users get nothing while if asks for less then he is leaving some resources unclaimed and therefore

$$BR_i(x_{-i}) = C - \sum_{j \neq i}^{n} x_j \,. \tag{10}$$

It is easy to see from the best response correspondence that any profile of demands $x_i \in [0, C]$ that add up to C will be a Nash equilibrium. Hence, each player i is indifferent between all of his requests $x_i \in [0, C]$ and the game is just not blessed with a unique equilibrium and has an infinite number of equilibria. The obvious problem with multiple equilibria is that the players may not know which equilibrium will prevail. Hence, it is entirely possible that a non-equilibrium outcome results because one player plays one equilibrium strategy while a second player chooses a strategy associated with another equilibrium [5].

It turns out that resource allocation encounters conflict over scare resources that results from the tension between individual selfish interests and common good. As Hardin stated in his article [12], "freedom in a commons brings ruin to all," that here means, social utility of an uncontrolled use of the common resources that each user have the freedom to make choices, is worse than if those choices were regulated. This results in the occurrence of the phenomenon called *tragedy of the commons.* In fact, individual users acting independently according to their own self-interest behave contrary to the common good of all users by depleting that resource through their collective actions.

To solve this problem, we first need to bring back the continuity to the payoff function Equation (9). Thus, we apply the resource allocation back-off condition $\sum_{j=1}^{n} x_j > C$ directly to the payoff function for each user i. We define a concave penalty function $\tau : \mathbb{R}_+ \to \mathbb{R}_+$

so that $\tau(0) = 1$ and $\tau(C) = 0$ and multiply it to the payoff function (9). To generalize, we also consider each unit of resource costs L and we have

$$\tilde{v}_i(x_i, x_{-i}) = u_i(x_i)\,\tau\Big(\sum_{j=1}^{n} x_j\Big) - Lx_i \quad \text{for all} \quad x_i, x_j \in [0, \infty)\,. \tag{11}$$

Example 5. Consider, for example, the concave penalty function $\tau(z)$ as follows:

$$\tau\Big(z\Big) = \sqrt{1 - \frac{z^p}{C^p}}\ , \tag{12}$$

where $z = \sum_{j=1}^{n} x_j$ and $p \in \mathbb{N}$. Intuitively, $\tau(0) = 1$ and $\tau(C) = 0$.

Figure 11 represents some examples of concave penalty functions Equation (12) for $p \in \{1, 2, 4, 8\}$. Although the larger values of p reduce inefficiency of Nash equilibrium, however make calculations more complex. In realistic situation of EV charging, this function can be programmed to the charger and it works when the demand exceeds from capacity C.

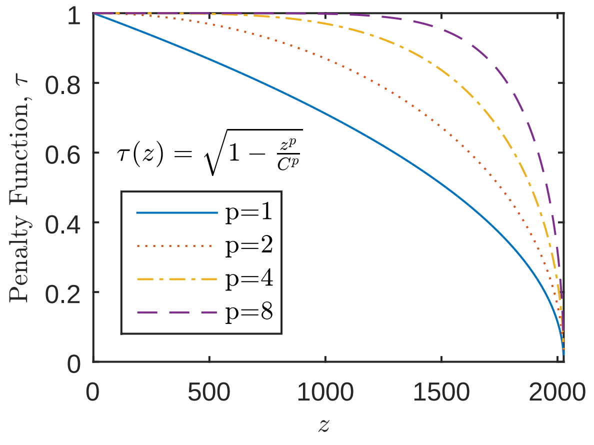


Figure 11: Penalty Function

Since the payoff functions are continuous there is a strong result on existence of the pure Nash equilibrium that is stated by Theorem 1 [8].

Theorem 1. *(Debreu, Glicksberg, Fan) An infinite strategic form game $\mathcal{G} = (\mathcal{N}, X_i, f_i)_{i\in\mathcal{N}}$ such that for each $i \in \mathcal{G}$*

i) X_i is compact and convex;

ii) $f_i(x_i, x_{-i})$ is continuous in x_{-i};

iii) $f_i(x_i, x_{-i})$ is continuous and concave [1] in x_i .

[1]in fact quasi-concavity suffices.

Then a pure strategy Nash equilibrium exists.

Another important question that arises in the analysis of strategic form games is whether the Nash equilibrium is unique. Theorem 2, provides sufficient conditions for uniqueness of an equilibrium in games with infinite strategy sets.

Theorem 2 (Theorem 1, [25])**.** *Consider a strategic form game $\mathcal{G} = (\mathcal{N}, X_i, f_i)_{i\in\mathcal{N}}$. For all $i \in \mathcal{G}$, assume that the action sets $X_i = \{x_i \in \mathbb{R}^{m_i} | h_i(x_i) \geq 0\}$, where h_i is a concave function, and there exists some $\tilde{x} \in \mathbb{R}^{m_i}$ such that $h_i(\tilde{x}_i) > 0$. Assume also that the payoff functions $(f_i)_{i\in\mathcal{N}}$ are diagonally strictly concave for $x \in X$. Then the game has a unique pure strategy Nash equilibrium. Where payoff functions $(f_i)_{i\in\mathcal{N}}$ are diagonally strictly concave for $x \in X$, if for every $x^{ne}, \bar{x} \in X$, we have $(\bar{x} - x^{ne})^\top \nabla f(x^{ne}) + (x^{ne} - (\bar{x})^\top \nabla f(\bar{x}) > 0$.*

The game $(\mathcal{N}, X_i, \tilde{v}_i)_{i\in\mathcal{N}}$ has unique Nash equilibrium that is calculated by maximizing user i's payoff function $\tilde{v}_i(x_i, x_{-i})$ and finding the solution to the first order conditions. So, we write down the first-order condition of user i's payoff function as follows

$$x_i^{ne} = \frac{\partial \tilde{v}_i(x_i, x_{-i})}{\partial x_i} = 0\,. \tag{13}$$

We therefore have n such equations, one for each player, and the unique Nash equilibrium is the strategy profile x^{ne} for which all users in the network, the Equation 13 are satisfied together, so that

$$x^{ne} = (x_i^{ne}, x_{-i}^{ne}),\ x_i^{ne} = \underset{x_i\in[0,C]}{\arg\max}\ \tilde{v}_i(x_i, x_{-i}),\ \forall i \in \mathcal{N}\,. \tag{14}$$

When resource allocation problem form as a result of selfish competition among users, the resulting stable solution may not, in fact, be system optimal [20]. In this circumstance, we would like to measure inefficincy constituted due to decentralized control. This is very important to decide whether a decentralized mechanism can be applied, regarding the loss of efficiency in comparison with the performance that would be obtained with a central authority. *Price of anarchy* (PoA) [17], is a concept that quantifies this inefficiency and is measured as the ratio between the worst equilibrium and the centralized solution. In the problem considered here, this notion will be slightly different and defined as the efficiency of the unique Nash equilibrium of the game $\mathcal{G} = (\mathcal{N}, X_i, f_i)_{i\in\mathcal{N}}$ and the optimal centralized solution of (6) as follows:

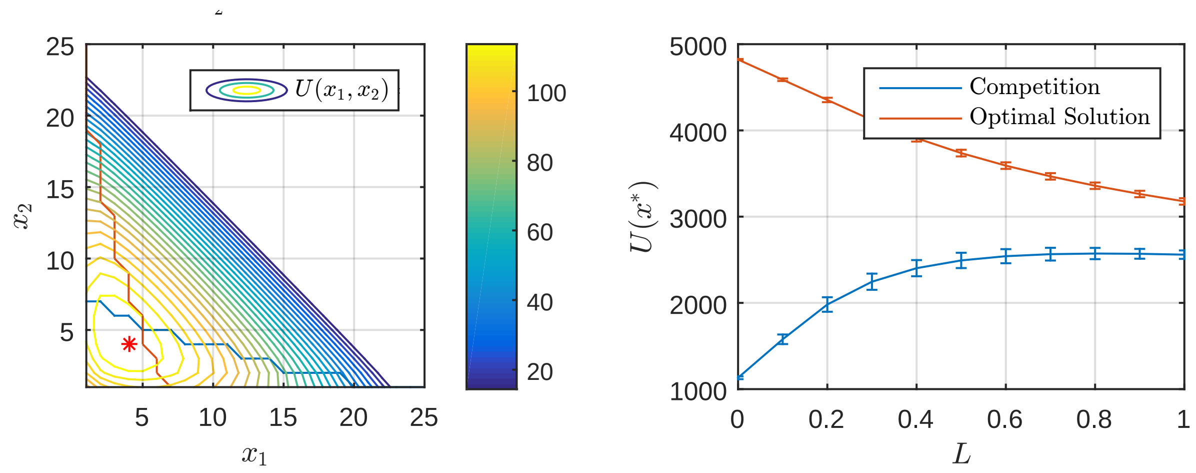
$$PoA = \frac{\sum_{i=1}^n \tilde{v}_i(x_i^{ne})}{\sum_{i=1}^n v_i(x_i^*)}\,, \tag{15}$$

where x^{ne} is the unique Nash equilibrium given by (14) and x^* is the solution of (6) .

Simulation

In this section, we proceed to simulate resource allocation in competition game to investigate in more details the inefficiency of Nash equilibrium. For this purpose, suppose the EV charging station settings of the section . Each user's utility function is considered as the normalized logarithmic function (3) with uniformly distributed random parameters of $\eta_i \in (0, 25)$ and $\chi_i \in (25, 100)$. We also consider the concave penalty function Equation 12 with $p = 1$ for executing the simulation. We start to simulate the problem for two players. Consider the charging station with the limited resource of $C = 25$ kWh and two EV owners $i \in \{1, 2\}$ which their EVs are connected to the station for charging. Both players have normalized logarithmic utility function Equation (3) with parameters $\eta_1 = 15$, $\chi_1 = 30$ and $\eta_2 = 38$, $\chi_2 = 70$ respectively. Figure 12a depicts inefficiency of distributed competitive resource allocation in two-player game ,*i.e.*, best response functions lines intersection, compared with optimal solution $U(x_1^*, x_2^*)$.

Now consider the same setting for a charging station with $n = 50$ users. Figure 12b plots social optimum of Nash equilibrium $\sum_{i=1}^{n} \tilde{v}_i(x_i^{ne})$, compared with optimal centralized solution $\sum_{i \in \mathcal{N}} v_i(x_i^*)$ for diffrent $L = \{0, 0.1, \ldots, 1\}$. Figure 13 represents the price of anarchy in competition against two parameters of price and number of users. The PoA is so sensitive to number of users in the competition such that increasing number of users negatively affect on PoA. Moreover, if the selfish behavior of users in competition do not control by pricing, inefficiency increase and consequently the PoA decrease. Note that the price of anarchy is independent of the competition topology [26].



(a) (b)

Figure 12: Inefficiency of distributed competitive resource allocation, (a) In two-player game. Nash equilibrium, *i.e.*, best response functions lines intersection, compared with optimal solution $\sum_{i=1}^{n} \tilde{v}_i(x_i^{ne})$, (b) In n-player ($n = 50$) game compared with optimal solution.

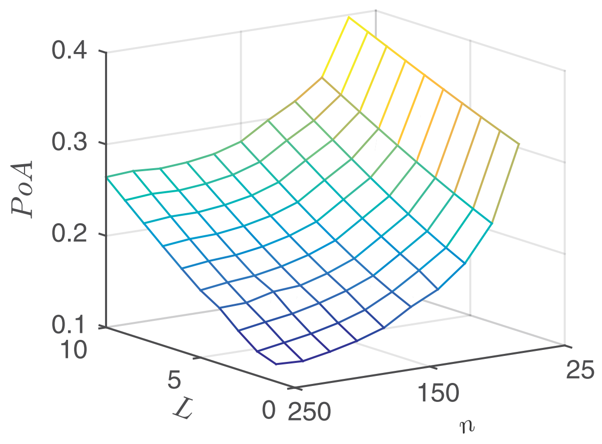


Figure 13: Price of Anarchy (PoA)

Related Work

Both centralized and distributed solution approaches for the generic problem of resource allocation were studied widely in various fields of expertise and a full review is impossible here. Figure 15 represents the conceptual resource allocation framework that is used in this work.

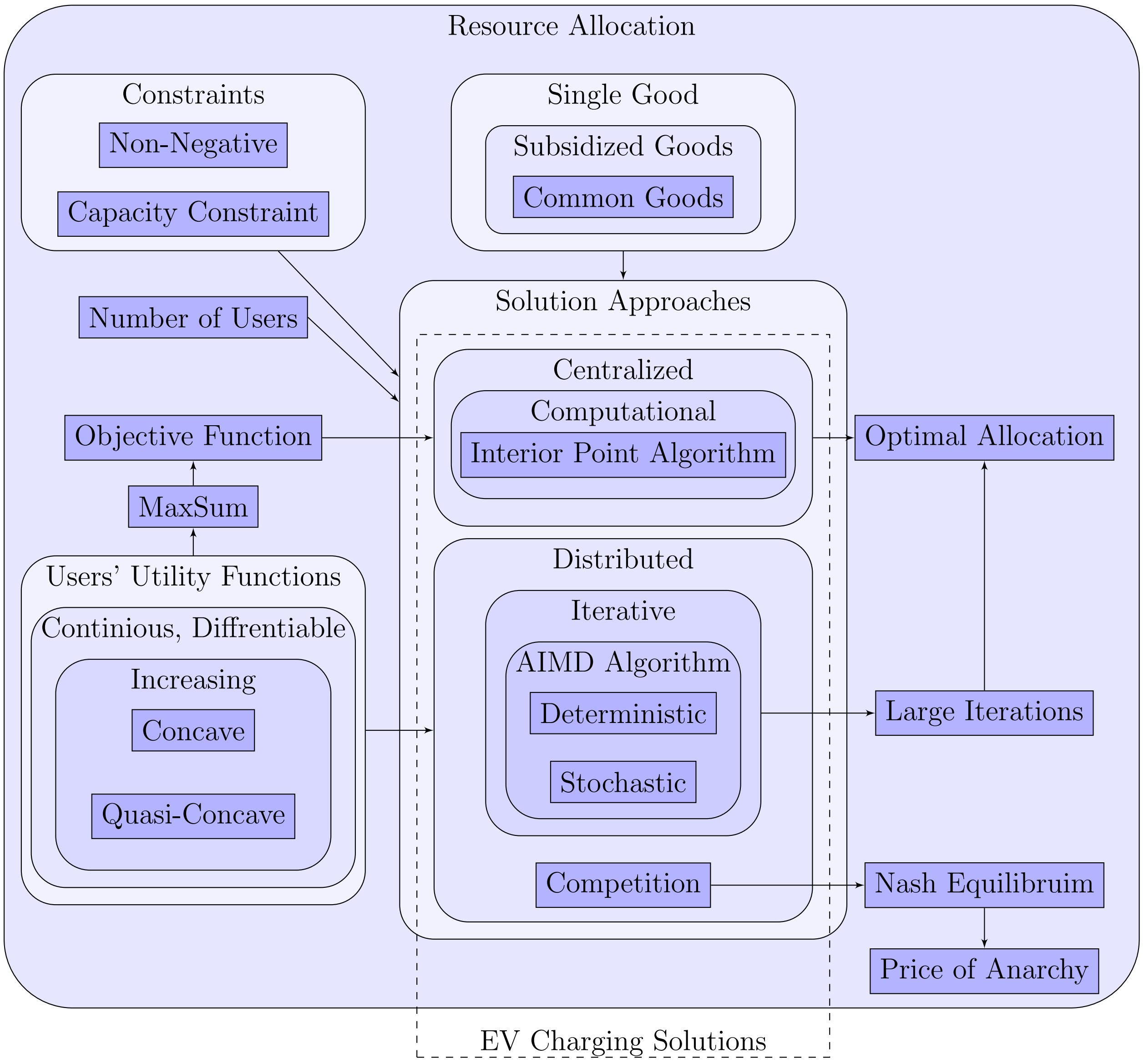


Figure 14: Conceptual diagram of resource allocation area that is studied in this work.

In many recent applications, in data (or communication) networks the area of resource allocation optimization has received a surge attention. In such networks, each user adopt an admissible utility function to quantify its benefit from achieving resource and the problem, that is called *Network Utility Maximization (NUM)*, defines as a constrained maximization of some utility functions [24], [13]. Users utility functions are commonly considered to be concave, continuous and strictly increasing functions modeling *elastic networks* [16], [27], which are more mathematically tractable [4], but limits applicability. On the other hand, many applications require *inelastic network* models where non-concave utility functions or discontinious utility functions need to be maximized [9]. Kelly (1997), in his seminal paper [16], proposed an algorithm to achieve proportional fairness of rate allocation in elastic networks. Inelastic networks that are more challenging, studied in [19], [9], [11] and specificely Sigmoidal programming algorithm is proposed in [30]. In [1], using utility proportional fairness policy, both elastic and inelastic utility functions compared. In large-scale networks, distributed solutions are particularly attractive where a centralized solution is not feasible [24].

There is also substantial literature on AIMD, the algorithm proposed by Chiu and Jain in [6] and applied experimentally by Jacobson in [14], as the most efficient-fair rate control in Internet applications. The efficiency and fairness of the AIMD algorithm also investigated in [18] and a comprehensive review of the AIMD algorithm and its applications is collected in [7]. This work uses the result of [31] that used AIMD algorithm in stochastic framework for common goods resource allocation.

EV charging has been the most widely studied as an application of distributed resource allocation. In [2], Ardakanian et al. (2013) proposed a distributed control algorithm that adapts the charging rate of EVs to the available capacity of the network ensuring that network resources are used efficiently and each EV charger receives a fair share of these resources. Inspired by the design of the Internet, which offers best effort services to elastic applications that back off in case of congestion [19], our approach is to quickly adapt EV charging rates to the condition of the network [4]. Specifically, we propose a distributed control algorithm so that every charger can independently set its charging rate based on congestion signals it receives from measurement nodes installed on its path to the sub-transmission substation. This algorithm ensures that EV chargers receive a proportionally fair [11] share of the available capacity of the distribution network, and lines and transformers are not overloaded. In [3], a static non-cooperative game formulation of the problem of distributed charging in electrical vehicle (EV) networks is proposed.

In [29], Stuedli et al. (2012) proposed a distributed AIMD based algorithm to allocate available power among connected EVs in order to maximize the utilization of EV owners in a range of situations. In [28], they (2012) also used the same formalization framework to expand the modifications of the basic AIMD algorithm to charge EVs. In both articles they considered a fairness policy as a constraint. The effectiveness of AIMD at mitigating the impact of domestic charging of EVs on low-voltage distribution networks is investigated [21] by Liu and McLoone (2015).

In [22], Marden and Wierman (2013) introduced a class of games termed distributed welfare games, which represents a game theoretic model for resource allocation problems with separable objective/welfare functions. It is the closest line of work to ours in game theoretic approach, however it is considered a finite strategic-form game where each player has a finite action set and a discrete utility function. Hardin (1968), in his seminal article [12], popularized the concept of *the tragedy of the commons* as an economic theory of a situation within a shared-resource system where individual users acting independently according to their own self-interest behave contrary to the common good of all users by depleting or spoiling that resource through their collective action. In [17], Koutsoupias and Papadimitriou (1999) introduced the concept of *price of anarchy* that is the idea of quantifying the inefficiency of selfish solutions using the framework of approximation. In network resource allocation, the notion of price of anarchy is introduced to quantify efficiency loss by Johari and Tsitsiklis (2004) in [15].

Conclusions & Future Work

In this work, we introduced the problem of resource allocation for subsidized goods among large number of users. We proposed variants of AIMD distributed algorithm for an efficient and private allocation. To this end, we first defined resource allocation problems as a class of NUM problems. We improved AIMD algorithm,both stochastic and deterministic, to allocate subsidized goods where users have concave and nonmonotonous utility functions. We extended the results to propose a variant of AIMD algorithm to allocate common resource where users have quasi-concave utility functions.

We also modeled the same problem as a strategic game in order to figure out the unique Nash equilibrium. We represented inefficiency of the solution by calculating *price of anarchy* that is the result of the tragedy of the commons in a network in which users compete for scarce resources. We then proved that our result of AIMD resource allocation of common goods where users utility functions are concave increasing, is better than stable result of

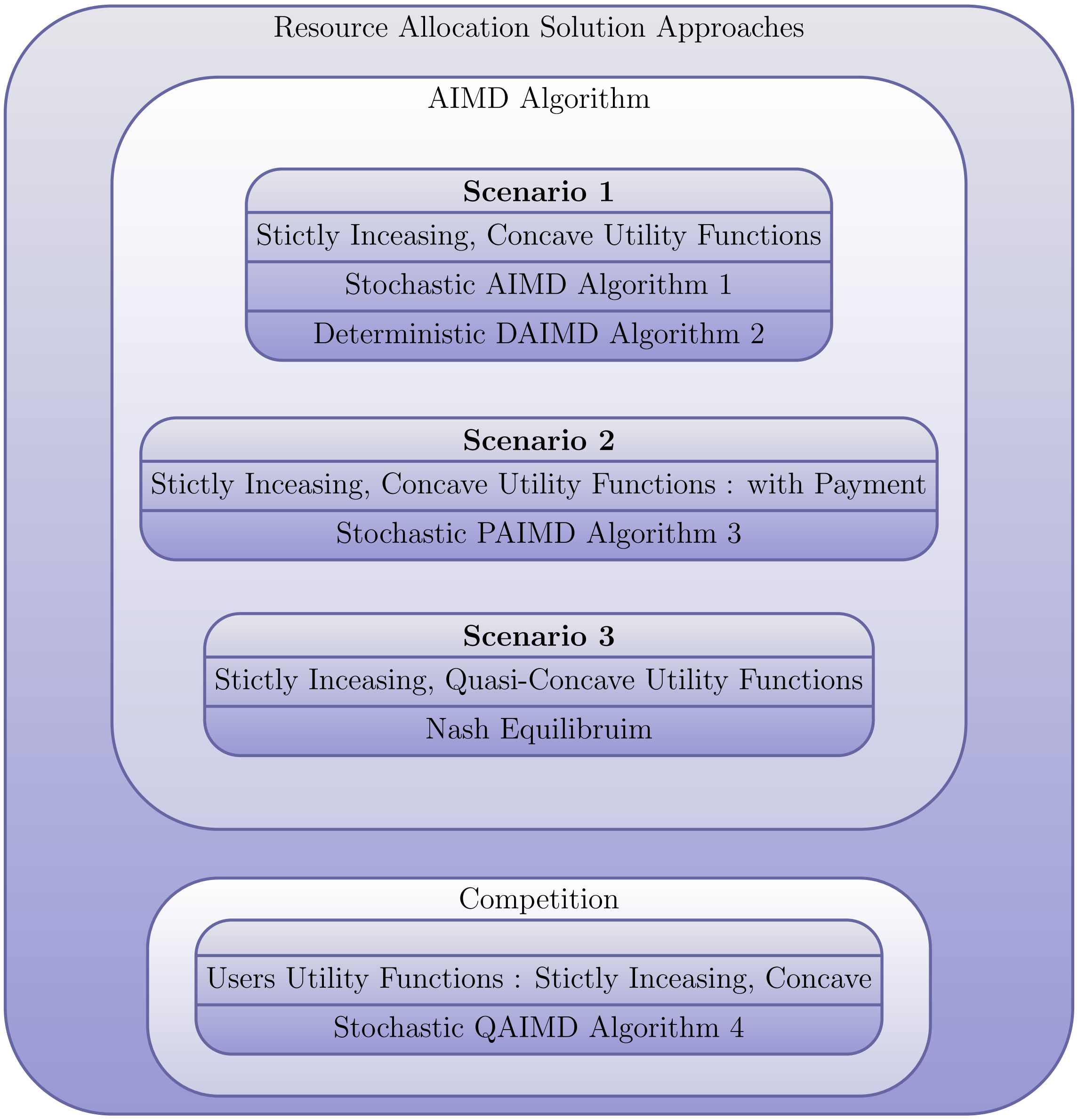

Resource Allocation Solution Approaches
AIMD Algorithm
Scenario 1
Stictly Inceasing, Concave Utility Functions
Stochastic AIMD Algorithm 1
Deterministic DAIMD Algorithm 2
Scenario 2
Stictly Inceasing, Concave Utility Functions : with Payment
Stochastic PAIMD Algorithm 3
Scenario 3
Stictly Inceasing, Quasi-Concave Utility Functions
Nash Equilibruim
Competition
Users Utility Functions : Stictly Inceasing, Concave
Stochastic QAIMD Algorithm 4

Figure 15: Conceptual diagram of Resource Allocation Solution Approaches: AIMD and Competition.

Nash equilibrium in game theoretic approach.

We simulated the results in a different networks of renewable-energy powered charging station in which EVs connected to be charged and we showed through simulations that our algorithms converge to the optimal solution.

Given that distributed resource allocation for inelastic networks using NUM is NP-hard, it is not surprising that many issues remain open on this challenging topic. However, AIMD Algorithm efficiency remains unknown when any condition of Assumption 1 is violated, more precisely either a generic non-concave or a discontinuous utility function is considered.

Another exciting open problem is to improve the proposed approach to make it incentive compatible, thereby ensuring that each user, *e.g.*, electric vehicle will represent its utility function truthfully.

Acknowledgment

This work was supported by grant T27346 from *CN Center for Studies in Sustainable Supply Chain Management*. The author wish to thank Dr. Ahmet Satir for useful helps relating to the material in this work.

Appendices

AIMD Preliminaries[2]

The additive-increase/multiplicative-decrease (AIMD) algorithm is a feedback control algorithm best known for its use in TCP congestion control. AIMD combines linear growth of the congestion window with an exponential reduction when a congestion takes place. Multiple flows using AIMD congestion control will eventually converge to use equal amounts of a contended link [6]. The related schemes of multiplicative-increase/multiplicative-decrease (MIMD) and additive-increase/additive-decrease (AIAD) do not converge. The approach taken is to increase the transmission rate (window size), probing for usable bandwidth, until loss occurs. The policy of additive increase may, for instance, increase the congestion window by a fixed amount every round trip time. When congestion is detected, the transmitter decreases the transmission rate by a multiplicative factor; for example, cut the congestion window in half after loss. The result is a saw-tooth behavior that represents the probe for bandwidth.

To make AIMD algorithm more tangible, we can compare it with Economic Order Quantity (EOQ) in inventory management. The EOQ is used as part of a continuous review inventory system in which the level of inventory is monitored at all times and a fixed quantity is ordered each time the inventory level reaches a specific reorder point.

AIMD requires a binary signal of congestion. Most frequently, packet loss serves as the signal; the multiplicative decrease is triggered when a timeout or acknowledgement message indicates a packet was lost. It is also possible for in-network mechanisms to mark congestion (without discarding packets) as in Explicit Congestion Notification (ECN).

Let $\omega(t)$ be the sending rate (e.g. the congestion window) during time slot t, $(a > 0)$ be the additive increase parameter, and $(0 < b < 1)$ be the multiplicative decrease factor.

$$\omega^{(t+1)} = \begin{cases} \omega^{(t)} + a, & \text{if congestion is not detected} \\ \omega^{(t)} + b, & \text{if congestion is detected} \end{cases}$$

[2]https://en.wikipedia.org/wiki/Additive_increase/multiplicative_decrease

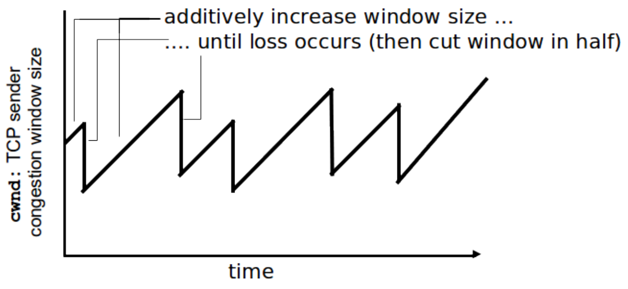


Figure 16: Saw-tooth AIMD, TCP congestion control

In TCP, after slow start, the additive increase parameter a is typically one MSS (maximum segment size) per round-trip time, and the multiplicative decrease factor b is typically 1/2.

MATLAB Code

In the following the simulation procedure written in MATLAB code is represented. We first define concave and quasi-concave utility functions and their derivatives in section . in section , AIMD for concave and nonmonotonous utility functions and in , AIMD for quasi-concave utility functions is represented.

Utility Functions

```
function [ u ] = UtilFunc(x , eta , x_max )
zeta=100;
u = 1 * log( 1 + eta * x ) / log( 1 + eta * x_max  );
end

function [ dU ] = dUtilFunc(x , eta , x_max )
zeta=100;
dU =(zeta * eta) / ( ( 1 + eta * x) * log( 1 + eta * x_max) );
end

function [ w_ ] = w(x , etaa , x_inf )
zeta=100;
w_ = zeta/(1+exp(-etaa*(x-x_inf))) - zeta/(1+exp(etaa*x_inf));
end

function [ d_w ] = dw(x , etaa , x_inf )
zeta=100;
d_w = zeta * etaa * exp(-etaa*(x-x_inf))  / (1+exp(-etaa*(x-x_inf)))^2 ;
end
```

AIMD for Concave and Nonmonotonous Utility Functions

```
% Stochastic and Detrministic AIMD for Subsidized (and common) Goods.
% Concave and nonmonotonous utility functions
clc; clear all;
for m=1:1000

% Define parameters
alpha=1; beta=0.65; n=50;
for i=1:n
    x_0(i)=rand; eta(i)= rand * 25; x_max(i)= randi([25,100]); x_u(i) = 0;
end
C=0.65*sum(x_max); gamma=50; L_Vec = 0:0.1:1;

% AIMD Algorithm
for l=1:11
    L=L_Vec(l); x=x_0; sum_x= zeros(1,50); t=1;
    while t<1000
        for i=1:50
            sum_x(i) = sum_x(i) + x(i);
            xbar(i) =sum_x(i)/(t+1);
            du_xbar(i) =dUtilFunc(xbar(i) ,eta(i),x_max(i))-L;
            if sum(x)<C & x(i)<x_max(i)
                u_i_before(i)=UtilFunc(x(i), eta(i), x_max(i))-(L*x(i));
                x(i)=x(i)+alpha;
                u_i_after(i)=UtilFunc(x(i), eta(i), x_max(i))-(L*x(i));
                if u_i_before(i) <  u_i_after(i)
                    x(i)=x(i);
                else
                    x(i)=x(i)-alpha;
                end
            elseif sum(x)<C & x(i)>=x_max(i)
                x(i)=x(i);
            else
                R=rand;
                lambda = gamma*du_xbar(i) / xbar(i);
                if R > lambda
                    x(i)=beta*x(i);
                else
                    x(i)=x(i);
                end
            end
        mat_du_xbar((l-1)*n+i,t)= du_xbar(i);
```

```
            mat_xbar(t, (l-1)*n+i)= xbar(i);
            end
            t=t+1;
        end
        for i=1:50
            u_i_modified(i)=UtilFunc(x(i), eta(i), x_max(i))-(L*x(i));
        end
        U_modified(l)= sum(u_i_modified);
        Total_Allocated_x_modified(:,l)= sum(x);
    end
    U_mat(m,:)=U;
    x_stoch=x;

    % Deterministic AIMD Algorithm
    for l=1:1
        L=L_Vec(l); x=x_0; sum_x= zeros(1,50); t=1
        while t<10000
            for i=1:50
                sum_x(i) = sum_x(i) + x(i);
                xbar(i) =sum_x(i)/(t+1);
                du_xbar(i) =dUtilFunc(xbar(i) ,eta(i),x_max(i))-L;
                if sum(x)<C & x(i)<x_max(i)
                    x(i)=x(i)+alpha;
                elseif sum(x)<C & x(i)>x_max(i)
                    x(i)=x(i);
                else
                    lambda = gamma*du_xbar(i) / xbar(i);
                    x(i)=beta*(1-lambda)*x(i) + (lambda)*x(i);
                    if x(i)<x_0(i)
                        x(i)=x_0(i);
                    else
                        x(i)=x(i);
                    end
                end
            mat_du_xbar((l-1)*n+i,t)= du_xbar(i);
            mat_xbar(t, (l-1)*n+i)= xbar(i);
            end
            t=t+1;
        end
        for i=1:50
            u_i_deter(i)=UtilFunc(x(i), eta(i), x_max(i))-(L*x(i));
        end
        U_deter(l)= sum(u_i_deter);
        Total_Allocated_x_deter(:,l)= sum(x);
    end
    U_mat_deter(m,:) = U_deter;
    x_deter=x;

    % Nonlinear programming solver (fmincon)
    for l=1:11
        L=L_Vec(l);
    fun = @(x) -( ( 100*log( 1 + eta(1) * x(1) ) / log( 1 + eta(1) * x_max(1) ))-L
        *x(1) + ...
                 ( 100*log( 1 + eta(2) * x(2) ) / log( 1 + eta(2) * x_max(2) ))-L
                     *x(2) + ...
                    % ... All utility functions 3 to 48.
                 ( 100*log( 1 + eta(49) * x(49) ) / log( 1 + eta(49) * x_max(49)
                    ))- L* x(49) + ...
                 ( 100*log( 1 + eta(50) * x(50) ) / log( 1 + eta(50) * x_max(50))
                     )- L* x(50) );
    x0(1:50) = 10;
    A = ones(1,50);
    b = C;
    Aeq = [];
    beq = [];
    lb=zeros(1,50);
    ub(1:50) = x_max;
    [x,fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub);
    u_fmincon(l)= -fval;
    if l==1
    y_fmincon=x;
    end
    Total_Allocated_x_fmincon(:,l)= sum(x);
    end
    u_fmincon_mat(m,:)=u_fmincon;

    end

    % Plot Utility Functions (Concave and nonmonotonous)
    h = figure;
    h1=plot(mat_xxx_u(:,1),mat_u(:,1)); hold on;
    h3=plot(mat_xxx_u(:,2),mat_u(:,2)); hold on;
    h2=plot(mat_xxx_u_l(:,1),mat_u_l(:,1),'--'); hold on;
    h4=plot(mat_xxx_u_l(:,2),mat_u_l(:,2),'--');
    grid on;
    xlabel('Allocated Resource, $x_i$ ','FontSize',10, 'Interpreter','Latex')
```

```
ylabel({'$$\textnormal {\hspace{0.1mm} Utility Function, $u_i$ }$$', '$$\
    textnormal {Payoff Function, $v_i$}$$'},'FontSize',10, 'Interpreter', '
    Latex')
h_legend = legend('Location','best', strcat('\eta_i= ', num2str(eta_graph_1),
    ' , \chi_i= ', num2str(x_max_graph_1)), strcat('\eta_i= ', num2str(round(
    eta_graph_n, 1)), ' , \chi_i= ', num2str(x_max_graph_n)), strcat('\eta_i=
    ', num2str(eta_graph_1), ' , \chi_i= ', num2str(x_max_graph_1), ' , L=0.3'
    ), strcat('\eta_i= ', num2str(round(eta_graph_n, 1)), ' , \chi_i= ',
    num2str(x_max_graph_n), ' , L=0.3'))
set(h_legend,'FontSize',5);
set(h,'Units','Inches');
axis([0, 100 , 0, 105])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'uv','-dpdf','-r0')

% Plot: the derivative of payoff function $v'_i(\bar{x}_i{(t)})$ for six
% randomly selected users when $L=0.3$.
h = figure; l=4;
for i=1:9:50
    plot(mat_du_xbar((l-1)*n+i,1:50))
    hold on
end
xlabel('$t$','Interpreter','Latex')
ylabel('$v^{\prime}_i(\bar{x}_i(t))$', 'Interpreter', 'Latex')
grid on
set(h,'Units','Inches'); axis([0, 50 , 0, 50]);
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'uprime_xbar_t','-dpdf','-r0')

% Plot: the average of allocated resource $\bar{x}_i{(t)}$ to the optimal
% point for six randomly selected users when $L=0.3$.
h = figure; l=4;
plot(mat_xbar(:,(l-1)*n+1));  hold on;
plot(mat_xbar(:,(l-1)*n+10)); hold on;
plot(mat_xbar(:,(l-1)*n+19)); hold on;
plot(mat_xbar(:,(l-1)*n+28)); hold on;
plot(mat_xbar(:,(l-1)*n+37)); hold on;
plot(mat_xbar(:,(l-1)*n+46)); hold off
xlabel('Iterations $(t)$','Interpreter','Latex')
ylabel('$\bar{x}_i(t)$', 'Interpreter', 'Latex')
grid on
set(h,'Units','Inches'); axis([0, 1000 , 0, 60]);
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
```

```
print(h,'xbar_t','-dpdf','-r0')

% Plot: the deterministic derivative of payoff function
% $u'_i(\bar{x}_i{(t)})$ for two randomly selected users (solid
% lines),compared with corresponding stochastic ones (dashed lines.
h = figure; l=1;
    plot(mat_du_xbar_deter((l-1)*n+1,1:50));  hold on
    plot(mat_du_xbar((l-1)*n+1,1:50),'--', 'LineWidth', 2);  hold on
    plot(mat_du_xbar_deter((l-1)*n+20,1:50)); hold on
    plot(mat_du_xbar((l-1)*n+20,1:50),'--', 'LineWidth', 2); hold on
xlabel('$t$','Interpreter','Latex')
ylabel('$u^{\prime}_i(\bar{x}_i(t))$', 'Interpreter', 'Latex')
legend('DAIMD','AIMD')
grid on
set(h,'Units','Inches');
axis([0, 50 , 0, 50])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'Duprime_xbar_t_compare','-dpdf','-r0')

% Plot: The deterministic average of allocated resource $\bar{x}_i{(t)}$ to
% the optimal point for two randomly selected users (solid lines) compared
% with corresponding stochastic ones (dashed lines).
h = figure; l=1;
plot(mat_xbar_deter(:,(l-1)*n+1)); hold on;
plot(mat_xbar(:,(l-1)*n+1) , '--', 'LineWidth', 1); hold on
plot(mat_xbar_deter(:,(l-1)*n+35)); hold on
plot(mat_xbar(:,(l-1)*n+35), '--', 'LineWidth', 1); hold off
xlabel('Iterations $(t)$','Interpreter','Latex')
ylabel('$\bar{x}_i(t)$', 'Interpreter', 'Latex')
legend('DAIMD','AIMD')
grid on
set(h,'Units','Inches'); axis([0, 5000 , 0, 80])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'Dxbar_t_compare','-dpdf','-r0')

% Plot: the efficiency of stochastic AIMD Algorithm and PAIMD Algorithm for
% $L\in\{0, 0.1, \dots, 1\}$.
h = figure;
PoA = U_mat ./ u_fmincon_mat;
y3 = mean(PoA); e3 = std(PoA);
PoA_modified = U_mat_modified ./ u_fmincon_mat;
y4 = mean(PoA_modified); e4 = std(PoA_modified);
errorbar(L_Vec, y3, e3); hold on
errorbar(L_Vec, y4, e4,'--');
grid on
xlabel('$L$','Interpreter','Latex')
ylabel('Efficiency', 'Interpreter', 'Latex')
legend({'Elastic AIMD','Elastic PAIMD'},'Interpreter','Latex', 'Location','
   southeast')
set(h,'Units','Inches');
axis([0, 1 , 0.7, 1.1])
set(h, 'PaperPosition', [0 0 3.20 2.4]);
set(h, 'PaperSize', [3.20 2.4]);
print(h,'efficiency','-dpdf','-r0')

% Plot: the efficiency of deterministic AIMD Algorithm and DAIMD Algorithm
% for $L\in\{0, 0.1, \dots, 1\}$.
h = figure;
PoA_deter = U_mat_deter ./ u_fmincon_mat;
y4 = mean(PoA_deter);
e4 = std(PoA_deter);
errorbar(L_Vec, y4, e4);
grid on
xlabel('$L$','Interpreter','Latex')
ylabel('Efficiency', 'Interpreter', 'Latex')
set(h,'Units','Inches');
axis([0, 1 , 0, 1.1])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'Defficiency','-dpdf','-r0')
```

AIMD for Quasi-Concave Utility Functions

```
% Stochastic AIMD for common Goods.
% Quasi-concave utility functions
clc; clear all;
 for m=1:1000

% Define parameters
    ii_vec=0.5:0.25:3; n=50;
    for ii=1:11
        cc=ii_vec(ii);
    for i=1:n
            x_0(i)=rand;
            etaa(i)=randi([11,50])*0.01;
            x_inf(i)=randi([40,60]);
    end
    alpha=1; beta=0.85; gamma1=5; gamma2= 10; C=sum(x_inf)*cc;
    sum_x= zeros(1,n); x=x_0; L=0; t=1; t1=0; t2=0;

% AIMD Algorithm
    while t<100000
        for i=1:n
            sum_x(i) = sum_x(i) + x(i);
            x_bar(i) = sum_x(i) / t;
            du_xbar_Sigmoid(i)= dw(x_bar(i), etaa(i), x_inf(i));
            R = rand;
            if sum(x)<C
                if x_bar(i)<x_inf(i)
                    lambda1 = gamma1*du_xbar_Sigmoid(i) / x_bar(i);
                    if lambda1<R
                        x(i)=x(i) * 1/beta;
                    else
                        x(i)=x(i);
                    end
                else
                    x(i)= x(i)+ alpha;
                    if x(i)>1.5*x_inf(i)
                        x(i)= x(i)-alpha;
                    else
                        x(i)=x(i);
                    end
                end
            else
                if x_bar(i)<x_inf(i)
                        x(i) = x(i)-alpha;
                        if x(i)<0
                            x(i)= 0;
                        else
                            x(i)= x(i);
                        end
                else
                     lambda2 = gamma2*du_xbar_Sigmoid(i) / x_bar(i);
                     if lambda2<R;
                         x(i) = x(i)*beta;
                     else
                         x(i)=x(i);
                     end
                end
            end
            end
            t=t+1;
        end
        y=x;
            for i=1:n
                u_i(i)=w(y(i), etaa(i),x_inf(i));
            end
        sum_u=sum(u_i);
        sum_y=sum(y);
        cum_u_cc(ii)=sum_u;
        end
        cum_u_cc_mat(m,:) = cum_u_cc;
        x=x_0;
        for ii=1:11
            cc=ii_vec(ii);
            C=sum(x_inf)*cc;
        end

% Nonlinear programming solver (fmincon); As in Previous
 end

% Plot: the derivative of utility functions $w'_i(\bar{x}_i{(t)})$ for six
% randomly selected users when $C / \Psi=1.5$.
h = figure; ii=5;
for i=1:9:50
    plot(mat_du_xbar((ii-1)*n+i,1:600))
    hold on
end
xlabel('$t$','Interpreter','Latex')
ylabel('$w^{\prime}_i(\bar{x}_i(t))$', 'Interpreter', 'Latex')
grid on
set(h,'Units','Inches'); axis([0, 600 , 0, 15])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'wprime_xbar_t','-dpdf','-r0')

% Plot: the average of allocated resource $\bar{x}_i{(t)}$ to the optimal
% point for six randomly selected users  when $C / \Psi=1.5$.
h = figure; l=4;
plot(mat_xbar(:,(ii-1)*n+1));  hold on;
plot(mat_xbar(:,(ii-1)*n+10)); hold on
plot(mat_xbar(:,(ii-1)*n+19)); hold on;
plot(mat_xbar(:,(ii-1)*n+28)); hold on;
plot(mat_xbar(:,(ii-1)*n+37)); hold on;
plot(mat_xbar(:,(ii -1)*n+46)); hold off;
xlabel('Iterations $(t)$','Interpreter','Latex')
ylabel('$\bar{x}_i(t)$', 'Interpreter', 'Latex')
grid on
set(h,'Units','Inches'); axis([0, 1000 , 0, 100])
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'wxbar_t','-dpdf','-r0')

% Plot: the efficiency of QAIMD Algorithm.
poa= cum_u_cc_mat ./ u_fmincon_mat;
h = figure;
y1 = mean(poa); e1 = std(poa);
xx=ii_vec; errorbar(xx, y1, e1);
grid on
xlabel('$C/\Psi$','Interpreter','Latex')
ylabel('Efficiency', 'Interpreter', 'Latex')
set(h,'Units','Inches'); axis([0.5, 3 , 0.6, 1.2]);
set(h, 'PaperPosition', [0 0 3.20 2.4]); set(h, 'PaperSize', [3.20 2.4]);
print(h,'efficiencyS','-dpdf','-r0');
```

References

[1] A. Abdel-Hadi and C. Clancy. A utility proportional fairness approach for resource allocation in 4g-lte. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pages 1034–1040. IEEE, 2014.

[2] O. Ardakanian, C. Rosenberg, and S. Keshav. Distributed control of electric vehicle charging. In *Proceedings of the fourth international conference on Future energy systems*, pages 101–112. ACM, 2013.

[3] O. Beaude, S. Lasaulce, and M. Hennebel. Charging games in networks of electrical vehicles. In *Network Games, Control and Optimization (NetGCooP), 2012 6th International Conference on*, pages 96–103. IEEE, 2012.

[4] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[5] G. P. Cachon and S. Netessine. Game theory in supply chain analysis. In *Models, Methods, and Applications for Innovative Decision Making*, pages 200–233. INFORMS, 2006.

[6] D. M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 17(1):1–14, 1989.

[7] M. Corless, C. King, R. Shorten, and F. Wirth. *AIMD dynamics and distributed resource allocation*. SIAM, 2016.

[8] G. Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10):886–893, 1952.

[9] M. Fazel and M. Chiang. Network utility maximization with nonconcave utilities using sum-of-squares method. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 1867–1874. IEEE, 2005.

[10] J. Garcia and F. Nino. Coevolutionary learning in the tragedy of the commons. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 2202–2209. IEEE, 2003.

[11] P. Hande, S. Zhang, and M. Chiang. Distributed rate allocation for inelastic flows. *IEEE/ACM Transactions on Networking*, 15(6):1240–1253, 2007.

[12] G. Hardin. The tragedy of the commons. *Journal of Natural Resources Policy Research*, 1(3):243–253, 2009.

[13] T. Harks. Utility proportional fair bandwidth allocation: An optimization oriented approach. In *International Workshop on Quality of Service in Multiservice IP Networks*, pages 61–74. Springer, 2004.

[14] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM computer communication review*, volume 18, pages 314–329. ACM, 1988.

[15] R. Johari and J. N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[16] F. Kelly. Charging and rate control for elastic traffic. *European transactions on Telecommunications*, 8(1):33–37, 1997.

[17] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413. Springer, 1999.

[18] A. Lahanas and V. Tsaoussidis. Exploiting the efficiency and fairness potential of aimd-based congestion avoidance and control. *Computer Networks*, 43(2):227–245, 2003.

[19] J. W. Lee, R.R. Mazumdar, and N.B. Shroff. Nonconvexity issues for internet rate control with multiclass services: stability and optimality. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1. IEEE, 2004.

[20] S. Lichter, C. Griffin, and T. Friesz. The calculation and simulation of the price of anarchy for network formation games. *arXiv preprint arXiv:1108.4115*, 2011.

[21] M. Liu and S. McLoone. Enhanced aimd-based decentralized residential charging of evs. *Transactions of the Institute of Measurement and Control*, 37(7):853–867, 2015.

[22] J. R. Marden and A. Wierman. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.

[23] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge university press, 2005.

[24] D. P. Palomar and M. Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, 2006.

[25] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.

[26] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[27] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on selected areas in communications*, 13(7):1176–1188, 1995.

[28] S. Stüdli, E. Crisostomi, R. Middleton, and R. Shorten. Aimd-like algorithms for charging electric and plug-in hybrid vehicles. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–8. IEEE, 2012.

[29] S. Stüdli, E. Crisostomi, R. Middleton, and R. Shorten. A flexible distributed framework for realising electric and plug-in hybrid vehicle charging policies. *International Journal of Control*, 85(8):1130–1145, 2012.

[30] M. Udell and S. Boyd. Maximizing a sum of sigmoids. *Optimization and Engineering*, 2013.

[31] F. Wirth, S. Stüdli, J. Y. Yu, M. Corless, and R. Shorten. Nonhomogeneous place-dependent markov chains, unsynchronised aimd, and network utility maximization. *arXiv preprint arXiv:1404.5064*, 2014.

[32] E. Xydas, C. Marmaras, and L.M. Cipcigan. A multi-agent based scheduling algorithm for adaptive electric vehicles charging. *Applied Energy*, 177:354–365, 2016.