# AN ABSTRACT OF THE THESIS OF

Béatrice Moissinac for the degree of Master of Science in Computer Science presented on November 22, 2013.

Title: Reinforcement Learning-based Off-Equilibrium Incentives to Approximate the VCG Mechanism

Abstract approved: _____

Kagan Tumer

Auctions are used to solve resource allocation problem between many agents and many items in real-world settings. Unfortunately, in most cases, it is possible for selfish agents to manipulate the system for their own interest at the expense of the social welfare. Such manipulation can be prevented using the Vickrey-Clarke-Groves mechanism, which guarantees complete truthfulness from the agents, and therefore, preserve optimal social welfare. However, the Vickrey-Clarke-Groves mechanism is computationally expensive, mainly due to the search for the optimal allocation of items (the "Winner Determination Problem").

In this work, we propose the use of off-equilibrium incentives to approximate the VCG mechanism, where the agents use reinforcement learning using "difference rewards" to compute those incentives. In one round of the reinforcement learning the agents: (i) declare their preferences in terms of allocation; (ii) compute their reward using the difference reward; (iii) and update their Q-table and move toward system efficiency. We demonstrate theoretically the equivalence of the off-equilibrium incentives and the VCG mechanism, and empirically show that this approximation of VCG mechanism leads to desirable outcomes in a congestion game.

# Reinforcement Learning-based Off-Equilibrium Incentives to Approximate the VCG Mechanism

by

Béatrice Moissinac

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented November 22, 2013
Commencement June 2014

Master of Science thesis of Béatrice Moissinac presented on November 22, 2013.

APPROVED:

---

Major Professor, representing Computer Science

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

---

Béatrice Moissinac, Author

# ACKNOWLEDGEMENTS

This work has been submitted for review at the 13th International Conference on Autonomous Agents and Multiagent Systems 2014.

# TABLE OF CONTENTS

# TABLE OF CONTENTS (Continued)

Page

# LIST OF FIGURES

# LIST OF ALGORITHMS

# Chapter 1: Introduction

The optimal allocation of resources is a problem in many aspects of human life. From the individual time management to the social choice problems, resource allocation problems are costly to solve but even more costly not to address.

The most general case of resource allocation - which is also the most complex - is allocating many resources to many agents. How can we allocate resources such that no resource is wasted, and those who needed it the most obtain it? This particular problem is practically solved everyday using auctions [16, 29]. The premise of an auction is that the auctioned resource is bought by the participant who valued this resource the most (and can afford it). And in a perfect world, this would maximize the social welfare. Furthermore, the common belief that traditional auctions can achieve an efficient outcome is perpetuated at the highest level of the political system. In 1994, then Vice President Al Gore mandated the Federal Communications Commission (FCC) to conduct an auction to assign the limited electromagnetic spectrum. In his opening speech of the FCC's fourth auction, he stated that "Now we're using the auctions to put licenses in the hands of those who value them the most." [31, 32].

Nevertheless, Game Theory has extensively studied common situations where bidders manipulate the auction to obtain their desired outcome, no matter the potential harm done to the social welfare[1]. The selfish behavior of bidders has examples in electricity market [27, 49], airwaves attribution [30, 31], transportation [3, 5], pollution rights [34], mining and oil rights [13], or airport landing slots [40].

Human agents or automated agents, the issue remains the same: finding an efficient allocation of resource is a difficult problem. Nevertheless, multiagent systems offer undeniable advantages: simulation and easy demonstration of a learning behavior. More precisely, Reinforcement Learning (RL) has been investigated as a means to improve the resource allocation of large systems [19]. Simultaneously, auctions and Game Theory were also investigated, separately from RL, to improve resource allocation. For instance, in the work by Bredin et al. [7], the authors setup a market place where agents can

[1] A detailed introduction to Game Theory can be found in the work by Camerer [8].

bid for computational priority on servers. They assumed a list of tasks to be executed sequentially on different type of servers, with a fixed budget constraint. It is shown that if the agent has a perfect information on this game, then the agent can compute a strategy (a bid) that maximizes the agent's utility (i.e. minimize its execution time). In addition, they proved that under perfect information of this constraint game, there exists a Nash equilibrium.

A multiagent system using Reinforcement Learning usually assumes imperfect information. Such system, based on independent agents, will encounter the same problem of agents trying to manipulate the system toward their own interest, often at the expense of the social welfare of the system [25, 39].

For more than half a century, economists and game theorists have been seeking ways to incentivize the bidders not to be selfish, and try to produce an efficient outcome from the point of view of the social welfare. In 1961, William Vickrey published his work on second-price sealed auctions, which gives the bidders the incentive to bid their true valuation of an item[2] [51]. Later on, this work, combined with the Clarke Pivot [10] and the Groves payment [21], formed the Vickrey-Clarke-Groves mechanism (VCG) [11]. The VCG mechanism is a well-known result in Mechanism Design, a field of Game Theory which focuses on defining solution concepts and equilibrium for private information games (Games where bidders keep some information for themselves). The peculiar aspect of VCG auction is that it produces a social-optimal outcome, and guarantees that the bidders bid their true valuation of an item: VCG is incentive-compatible[3]. The idea is based on computing the correct payment for a bidder to obtain an item. Through the individual payment, the VCG mechanism internalizes the impact of a bidder on the system by setting the proper computation of a bidder's payment.

The VCG mechanism is a combinatorial auction based on a key step called the Winner Determination Problem (WDP). The WDP requires the exhaustive enumeration of all possible allocation to find the one which maximize the system's utility. Unfortunately, this search is computationally intractable (NP-Hard), and is difficult to implement when the number of agents and items increases [12]. A dynamic programming approach has

[2]Also called incentive-compatibility, this property states that agents' individual welfare is better when they reveal all private information truthfully

[3]This property states that agents' individual welfare is better when they reveal all private information truthfully

been proposed in the work by Rothkopf et al. [42]. For each possible set of allocation, the algorithm computes the system's utility. The algorithm goes from smallest sets to biggest sets adding one item at a time, and guaranteeing not to compute the same set twice. There is substantial computational saving with this technique in comparison of the brute force algorithm but this solution doesn't scale very well [45].

Currently, research on the WDP has focused upon three axes: specific cases of the WDP that can be solved in polynomial time, approximation of the WDP and distributed computation of the WDP. In the first case, the WDP has been shown to be solvable in polynomial time in very specific cases [47]. Those cases impose a bid structure to the game. However, this method is hardly generalizable for every game. In the second case, the optimal solution of the WDP is approximated using various methods [26, 35]. Nevertheless, in the general case, the WDP cannot be approximated without breaking the VCG's mechanism properties which guarantee truthfulness [35]. Finally, Parkes et al. developed a method called Distributed Mechanism Design, where computational work is distributed to agents [38].

Approaching this issue from the agent's behavior, Reinforcement Learning is used to model motivation and desirable behavior in a single-agent setting. Specifically, intrinsic Reinforcement Learning is a model in which an agent can distinguish between her extrinsic and intrinsic motivation to learn [9], or develop a sense of curiosity [23]. The premise is that the agent takes decision for her own sake. Additionally, recent research in multiagent learning has focused on the derivation of agent's utility functions that lead to desirable system-wide properties. Based on reward shaping and/or potential functions, such work indirectly aims to internalize an agent's externalities by capturing those externalities in their reward structure [1].

Following this idea, the proposed approach is based on computing off-equilibrium incentives in the form of the Difference Utilities (DU) for learning agents for congestion games. Basically, considering a general-sum repeated one-shot congestion game[4], in one round of the reinforcement learning the agents: (i) declare their preferences in terms of allocation; (ii) compute their reward using the difference reward; (iii) and update their Q-table to move toward system efficiency. Each iteration repeats the same game, and the agents use their rewards as a gradient towards equilibrium, bidding more efficient

[4]A one-state congestion game with only one action per player to be played before the game ends. The game is repeated over and over

allocation at every iteration of the Reinforcement Learning.

We extend the "learnable mechanism design" concept introduced by Parkes [37], by explicitly determining conditions that lead the difference utilities' incentives to guide agents towards achieving efficient allocation. We prove theoretically that the agents are pushed toward an efficient system welfare, and consequently adopt a truthful behavior. Moreover, we show empirically the robustness and the limitations of our approach, first using a variation of the El Farol Bar Problem [4], and then using the CubeSats domain [22]. The results indicate that that this approach: (i) is trivially scalable to significantly larger systems, (ii) produce a reasonable approximation of the VCG mechanism (iii) pushes the agents toward an efficient outcome and partial truthfulness. The contributions of this work are to:

- Under certain assumptions, demonstrate theoretically that at convergence, the DU computes payments, values and utilities equivalent to the VCG payments, values and utilities for each agent respectively, and therefore conserves the truthfulness property;
- Prove that agents adjust towards more efficient allocations when receiving off-equilibrium incentives during repeated games; and
- Show empirically with a real-world application that using DU as off-equilibrium incentives is significantly cheaper to compute than the VCG mechanism and is trivially scalable to larger systems.

This work is organized as follow: Chapter 2 presents the background on VCG, WDP and Difference Utilities. Chapter 3 provides a detailed proof of our approach. Chapter 4 introduces a variation of the El Farol Bar Problem, the CubeSats domain, and the metrics used to measure performance and truthfulness. Finally Chapter 5 presents the results of our experiments, discusses their significance, the computational complexity, and the robustness and limitations of the off-equilibrium incentives.

# Chapter 2: Background

## 2.1 Vickrey-Clarke-Groves Mechanism

### 2.1.1 Theory of Mechanism Design

In the field of Game Theory, Mechanism Design is the study of the formal rules for predicting how a game is going to be played, also called solution concept. Particularly, Mechanism Design focuses on games with private information: the players are unaware of some information, mostly what the other players think. Thus, research in Mechanism Design is interested in studying the optimal design of incentives for a group of players, such that the players disclose their private information. A significant difference from Game Theory in general is the fact that the researcher in Mechanism Design chooses and designs the game, as oppose to inheriting it. But most importantly, the researcher in Mechanism Design is interested by the outcome of that game. A detailed introduction can be found in Chapter 7 of the book by Fusenberg et al. [36].

There exist numerous examples where Mechanism Design is a powerful analytical tool: elections [6], non perfect markets [15], auctions [17], public policies [28].

### 2.1.2 Definition & Notations

In this work, we consider only congestion games. Rosenthal proposed congestion games in 1971 [41]. A congestion game is a game in which the player's payoff is determined by the resource $k$ used by the player, and how many other players used that same resource ($x_k$). Examples are developed in Chapter 4. For the purpose of this background section, consider a congestion game where each agent $i$ bid a preference for an item noted $\hat{\theta}_i$. $\hat{\theta}_i$ is a type, or preference, which encompasses informations about the resources form the point of view of the agent. For instance, in the El Farol Bar Problem (see Chapter 4 ), $\hat{\theta}_i$ is the preference for crowdedness. A preference for crowdedness answers the question "how many people you would like to see to the bar, at the same night than you¿'. Then, $k$ is the night an agent chooses to attend, and $x_k$ is the attendance at that night.

The center allocates to each agent $i$ an item $k$, based on the reported type $\hat{\theta}_i$. Nothing is stopping an agent from reporting a false type to manipulate her outcome. Therefore, an agent's true type $\theta_i$ is deviated by a scalar $h$. In this work, we define $\hat{\theta}_i = h\theta_i$, but it could also have been $\hat{\theta}_i = h + \theta_i$, this is an implementation choice that does not influence the results.

The VCG framework and the DU framework are somewhat distant fields, so the notations have been unified for consistency.

**Definition 1** (Type)**.** *A type defines the preferences of an agent over an item.*

**Definition 2** (Symmetry)**.** *Two agents are symmetric if their type is sampled from the same distribution with the same parameters. (All agents are interchangeable)*

Vickrey assumed that every agent is symmetric, because it facilitates the derivation of an agent's probability to win. It has been widely discussed that in the case where the agents are not symmetric, the Pareto-optimal equilibrium of the auction has a lower value (or expected price to be paid) than a Dutch auction [33, 51]. In addition, symmetric agents are interchangeable, which facilitates our discussion in Section 3.

**Definition 3** (Homogeneity)**.** *A non-empty set of items is homogeneous if the characteristics of each item are equivalent [33].*

It has been shown that heterogenous items might cause some limitations on the VCG mechanism [44], thus we limit this work to homogeneous items.

In the rest of this work, the notations are:

- $\mathcal{N}$ is a non-empty set of symmetric agents.
- $\vec{k}$ is a set of vector $\vec{k}$, the vectors of possible allocation of items for each agent. $\vec{k}$ is of size $|\mathcal{N}|$. Items have unlimited quantities[1], so several agents can have identical items, but each agent is allocated only one item. $k_i$ is the item bid on by agent $i$. $\vec{k}^*$ denotes the set of optimal allocation.
- $\Theta$ is a non-empty set of true type: $\forall i \in \mathcal{N}, \exists \theta_i \in \Theta$.

[1] We are considering congestion games, in which items are resources that can be shared. An unlimited number of agents can (try to) use a resource

- $\hat{\Theta}$ is a non-empty set of *declared* types: $\forall i \in \mathcal{N}, \exists \hat{\theta}_i \in \hat{\Theta}$.
- $\mathcal{H}$ is a non-empty set of all the possible deviations $h$.

### 2.1.3 Theory of VCG

The Vickrey-Clarke-Groves mechanism [10, 21, 51] is a well-known solution to solve resource allocation problems with multiple items and multiple agents. It motivates a truthful participation from agents by penalizing an agent's utilities with the cost of her distortion to the system's value. The VCG mechanism is a special case of the Groves mechanism, which is a protocol guaranteeing *truthfulness* from its agents. A detailed introduction is found in the work by Conitzer [11].

Consider a congestion game: Vickrey's assumption is that the center "could determine with confidence what the equilibrium competitive price would be so that no one could expect to have any influence on it". In other words, if the center guarantees the best allocation possible to the agents, and if the agents pays the VCG payment (see below), then the agents have no reason to lie about their preferences, because they cannot manipulate the center to obtain a better allocation. Basically, the VCG mechanism is a one-shot game: (i) the agents report their type ($\hat{\theta}_i$) to the center (ii) the center computes the WDP (iii) the agents compute their payment based on the WDP. This process is schematized in figure 2.1.

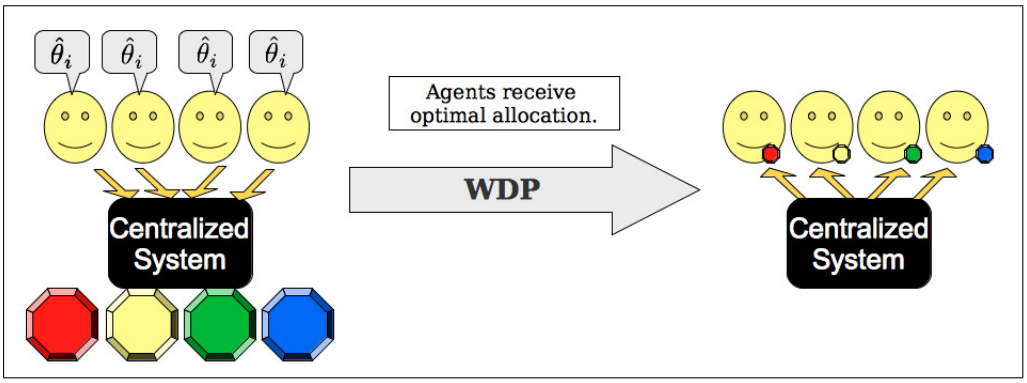



Figure 2.1: VCG mechanism is a one-shot game: (i) the agents report their type ($\hat{\theta}_i$) to the center (ii) the center computes the WDP (iii) the agents compute their payment based on the WDP.

**Definition 4** (Individual Value)**.** *The individual valuation $v_i(k_i, x_{k_i}, \theta_i)$ is the valuation of agent $i$ for item $k$ given her type $\theta_i$ and attendance $x_{k_i}$. The actual implementation depends on the game.*

**Definition 5** (Allocation Rule)**.** *The allocation rule $g_{eff}$ is the allocation that maximizes the sum of the individual valuations given the types ($\Theta$)*

$$g_{eff}(\Theta) = \arg\max_{\vec{k}\in\vec{k}} \sum_{i\in\mathcal{N}} v_i(k_i, x_{k_i}, \theta_i) \tag{2.1}$$

**Definition 6** (System Objective)**.** *The system objective is the sum of the individual valuations after the game is played.*

$$G(g_{eff}(\hat{\Theta})) = \sum_{i\in\mathcal{N}} v_i(g_{eff}(\hat{\Theta}), \theta_i) \tag{2.2}$$

**Definition 7** (Individual Utility)**.** *An agent's individual utility is the final value of an allocation, after payment, for this agent. This is how the VCG mechanism internalizes the disturbance caused by an agent to the system.*

$$u_i(g_{eff}(\hat{\Theta}), \hat{\Theta}_{-i}) = v_i(g_{eff}(\hat{\Theta}), \theta_i) - p_{VCG,i} \tag{2.3}$$

**Definition 8** (Payment)**.** *The payment internalizes into agent $i$ the disturbances caused by agent $i$ to the system.*

$$p_{VCG,i} = \underbrace{\sum_{n\in\mathcal{N}_{-i}} v_n(g_{eff}(\hat{\Theta}_{-i}), \hat{\theta}_n)}_{\text{World without agent i}} \underbrace{- \sum_{\substack{n\in\mathcal{N} \\ n\neq i}} v_n(g_{eff}(\hat{\Theta}), \hat{\theta}_n)}_{\text{World with agent i, not counting } v_i} \tag{2.4}$$

The payment is computed in two parts. The first part computes the alternative value of the system[2] if agent $i$ were not acting in $\mathcal{N}$. The second part is the value of the system where $i$ acts, but her valuation is not counted. Thus, VCG payment compares the valuations from the same set of agents $N - \{i\}$. If the net payment is positive (i.e. agent $i$ had a negative effect on the system), it decreases agent $i$'s utility. If the net payment is negative (i.e. agent $i$ had a positive effect on the system), it increases agent $i$'s utility.

It is crucial to note that $G$, $p_{VCG,i}$ and $u_i$ are computed using the *declared* types $\hat{\theta}_i$, because $\hat{\Theta}$ is the only information about the system, available to the agent.

#### 2.1.3.1 Truthfulness in VCG mechanism

The VCG mechanism is a strategyproof[3] mechanism guaranteeing complete truthfulness from its participants [11].

**Definition 9** (Truthfulness)**.** *A mechanism is strategyproof if every agent maximizes her utility by revealing her true type $\theta_i$, no matter the strategy of other agents, thus satisfying:*

$$u_i(g(\theta_i, \theta_{-i}), \theta_i) \geq u_i(g(\hat{\theta}_i, \theta_{-i}), \theta_i) \quad \forall i, \forall \theta_i, \forall \hat{\theta}_i, \forall \theta_{-i} \tag{2.5}$$

## 2.2 Winner Determination Problem

The computation of the WDP is a difficult problem because of the need to exhaustively enumerate all possible allocations to find the one which maximizes the system's utility. The previous section showed how the WDP is the key step to guarantee a favorable outcome from the VCG mechanism (Equation 2.1). In this section, we examine the different solutions provided to approximate the WDP.

[2]The notation $X_{-i}$ is a short hand for $X - \{i\}$.

[3]The VCG mechanism is a dominant-strategy mechanism, such that every agent has a best-response strategy no matter what is the other agents' strategy. A mechanism is called strategyproof when it is a dominant-strategy and incentive-compatible.

### 2.2.1 Approximation Solution

Finding an approximation for the WDP seems to be a logical approach for such a problem, but the WDP is not a trivial problem to approximate. It is an equivalent problem to the weighted set-packing problem which is very difficult to approximate [14].

In addition, it is difficult not to violate the VCG mechanism's incentive properties [35]. Many approximation algorithms cannot guarantee an exact solution. Therefore, the mechanism is not guaranteed to be strategyproof anymore.

For instance, in 2002, Lehman et al. [26] investigated the impact of non exact solutions upon the truth revelation properties of VCG. They show that the relaxation of the exact solution does not succeed to keep the truth revelation properties of VCG in the general cases. They do prove that in certain games where the players are restricted, using a greedy optimization method to approximate the WDP will preserve the truth revelation property of VCG. Nevertheless, the restrictions applied to the players force them to be single-minded in the sense that they care only about one bundle of item. In other words, they limit the player to one single bid. Obviously, this is hardly generalizable to real-world auctions, even though it preserves the truthfulness properties of the mechanism.

### 2.2.2 Identify Special cases

The general case of the WDP does not know a polynomial time algorithm able to construct a reasonable worst case bound [45]. However, there exist special cases of the WDP that can. Researchers have identified special cases in which the WDP is solvable in polynomial time. For instance, Sandholm [45] identifies such cases. Those cases are specific bid structures forced onto the game. It can be a limitation on the number of items an agent can bid on at once, or a limitation on the number of times an item can be bid on. The constraints reduce the search space and make the problem easier to solve. Later on, Sandholm et al. published the BOB algorithm [46], which improved previous results on identifying special cases. Basically, the core of the algorithm is a depth-first branch-and-bound tree search. The algorithm goes down the tree of all given (restricted) bids and finds an allocation which reasonably approximates the optimum. This work shows that special cases can be detected, but it does not guarantee that the case will occur. This

issue presents any attempt to generalize that type of algorithms into a general auction setting. In 2005, Sandholm et al. published a better version of BOB, called CABOB [47]. The authors also investigate a more general type of constraint on the bids. This version of the previous algorithm is also a search algorithm along a tree. This version of BOB has been improved: it is faster and makes better use of the structural knowledge of the tree.

That type of approach has two main issues. First, those restrictions are hardly generalizable to all games, and are very restrictive. Second, the polynomial time bound produced is far worse than the optimal which limits the impact on the auction.

### 2.2.3 Distributed Mechanism Design

Another approximation manages the computational limitation of the agents and utilizes it to compute a distributed approximation [35]. However, it requires the development and implementation of an additional algorithm corresponding to the specific combinatorial auction problem.

The Distributed Mechanism Design approach [38] has the agents iteratively compute the game's outcome by sharing the computational burden. This method takes advantage of the properties of a multiagent system. The communication are no longer passing through a single point, avoiding the bottleneck effect. The outcome is no longer computed by a potentially biased auction. This technique transforms the WDP into a distributed optimization problem. In addition, it addresses the risk of an agent manipulating the system, otherwise nothing would be stopping this agent from manipulating the messages sent through her [48].

## 2.3 Multiagent System

Multiagent system is at the intersection of Artificial Intelligence and distributed problem solving. As defined by Russell and Norvig [43], a multiagent system is composed of several intelligent agents interacting with an environment. This construction allows the decomposition of complex problems into smaller pieces, where each agents is responsible for only a small and simple part of the computation. A great advantage of multiagent system is that it can often solve or approximate problems that would be too difficult for

a single monolithic agent.

In section 2.3.1, we present the field of multiagent learning, its advantages and challenges and briefly review Reinforcement Learning and Q-Learning algorithm, as it will be used in our approach. Finally, we introduce the Difference Utilities (DU) framework in section 2.3.2.

### 2.3.1 Multiagent Learning, Reinforcement Learning & Q-Learning

Multiagent Learning is a subfield of multiagent system, in which intelligent agents use Machine Learning and other learning algorithms to perform a task. Often this task is learning sequential actions, classification and approximation. The main advantage of a learning multiagent system is the robustness of the system in changing and/or prone to failures environment.

In this work, the agents use a model free Reinforcement Learning technique called Q-Learning. We do not learn nor assume a probabilistic transition model. A complete introduction of Reinforcement Learning for single agent can be found in the exhaustive work by Sutton and Barto [50].

The paradigm for Q-Learning for a single agent is simple and powerful. We summarized the idea in figure 2.2. The agent uses a table (Q-table) to keep track of its previous experiences, and learn from it. At each time step of the reinforcement learning, the agent: (i) observes the environment; (ii) establishes her belief of the state; (iii) looks up her Q-table, searching for the best action to take in this state; (iv) carry out the chosen action; (v) senses a signal from the environment, which can be interpreted as a reward for this action; (vi) update her Q-table using the reward.

Formally, the update is written as follow:

$$Q_{t+1}(s_t, a_t) = \underbrace{Q_t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[ \overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q_t(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q_t(s_t, a_t)}_{\text{old value}} \right] \tag{2.6}$$

- $s_t$: Observed state at time $t$

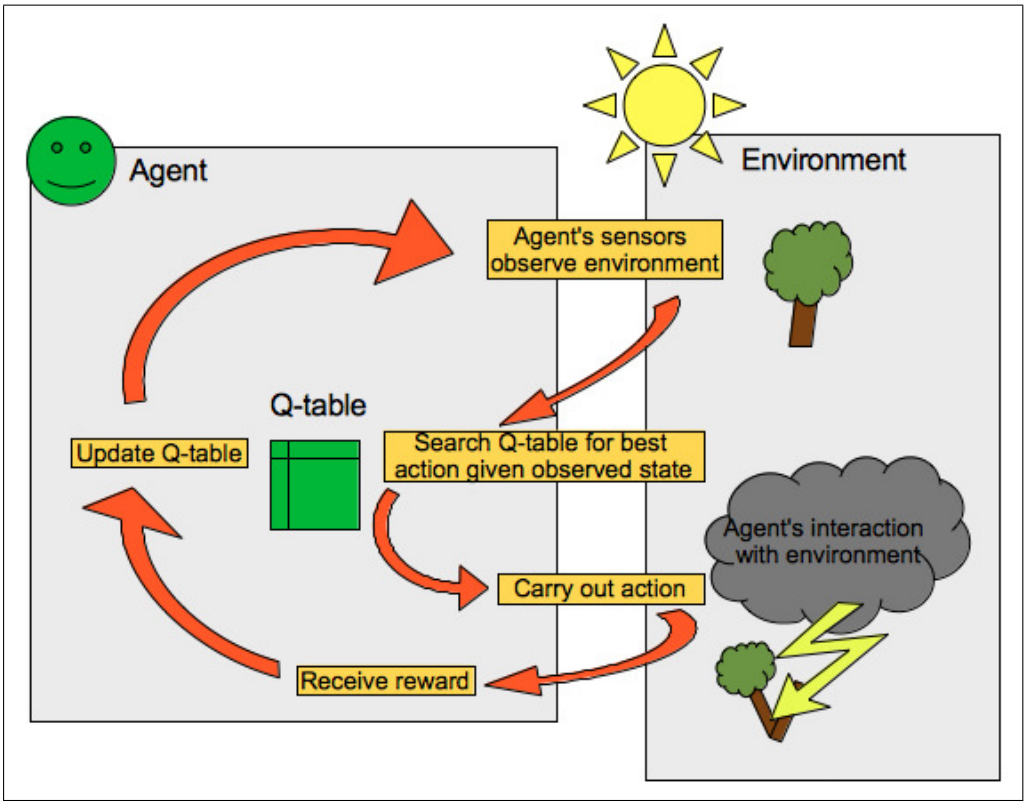



Figure 2.2: Diagram of Learning Iterations of a Single Agent using Q-Learning - At each time step of the reinforcement learning, the agent: (i) observes the environment; (ii) establishes her belief of the state; (iii) looks up her Q-table, searching for the best action to take in this state; (iv) carry out the chosen action; (v) senses a signal from the environment, which can be interpreted as a reward for this action; (vi) update her Q-table using the reward.

- $a_t$: Action taken at time $t$
- $Q_t(s_t, a_t)$: Q-value for state $s$ and action $a$ at time $t$
- $\gamma$: Discount factor, characterizing how much agent values future actions
- $\alpha$: Learning rate, charactering how much agent values new information

However, the games in which we apply VCG and DU are one-shot games. Thus, the Q-table is single state, and $\gamma \max_a Q_t(s_{t+1}, a) = 0$.

**Global Reward** In Chapter 5, we compare systems using off-equilibrium incentives computed by DU, with the system using the global reward. The global reward is the sum of every agent's reward. In this setting, every agent receives the same reward, as opposed to the DU, where the reward is individualized (See below).

### 2.3.2 Difference Utilities

The Difference Utilities (DU) is a shaped reward developed for Reinforcement Learning (RL) and Evolutionary algorithms [1]. It has been used in various real world domains such as Air Traffic control [2], and robot coordination and navigation [1]. The utilization of the DU is motivated by the fact that, similarly to VCG, the DU internalizes an agent's disturbance on the system, into this agent. In 2003, Parkes analyzes the equilibrium play of a mechanism (VCG) being learned by the participants. We extend this work, and shift the focus from the game to the design of the mechanism. We want to show that the game is actually incrementally learned, and lead the difference utilities' incentives to guide agents towards achieving efficient allocation.

It is important to notice that the DU is not specific to congestion problems, and this research is presenting one possible derivation of Difference Utilities: the difference reward [1] (Equation 2.7). Consider a repeated congestion game of multiple items, with simple learning agents. Each agent has an independent Q-table. In one round of the reinforcement learning the agents: (i) declare their strategy $(k_i, \hat{\theta}_i)$, where $k_i$ is agent $i$'s desired item (ii) compute their reward using the difference reward; (iii) and update their Q-table and move toward system efficiency. Figure 2.3 schematizes this process and Algorithm 1 provides details.

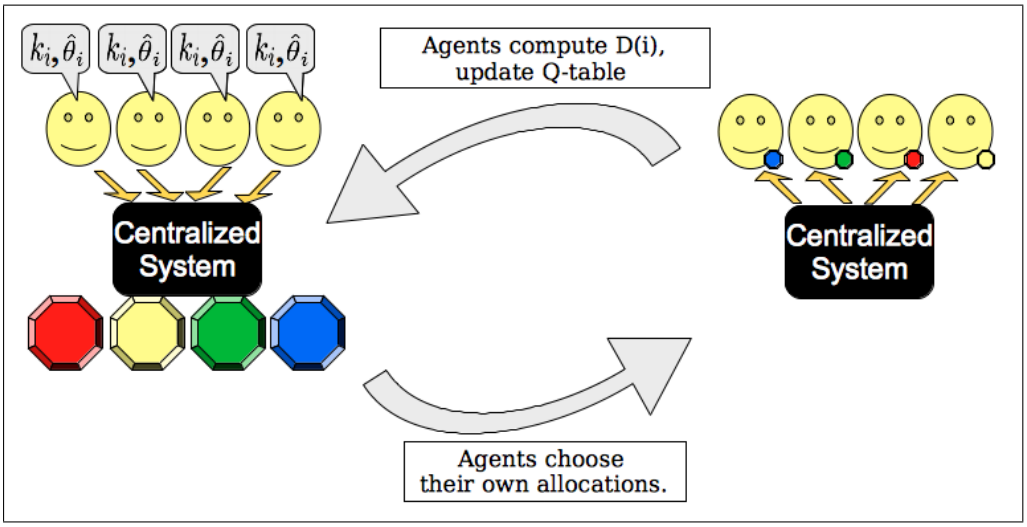



Figure 2.3: Difference Utilities using Reinforcement Learning: in one round of the reinforcement learning the agents: (i) declare their strategy $(k_i, \hat{\theta}_i)$, where $k_i$ is agent $i$'s desired item (ii) compute their reward using the difference reward; (iii) and update their Q-table and move toward system efficiency.

$$D(i) \equiv \underbrace{G(\vec{k}, \hat{\Theta})}_{\text{World with i}} - \underbrace{G(\vec{k}_{-i}, \hat{\Theta}_{-i})}_{\text{World without i}} \tag{2.7}$$

The first argument, $G(\vec{k}, \hat{\Theta})$, is the system's utility computed with the current state of the system. This state is based on $\vec{k}$ and $\hat{\Theta}$ in which $i$ exists, applies strategies and has influence on the global utility of the system. The second argument, $G(\vec{k}_{-i}, \hat{\Theta}_{-i})$ is the system's utility computed with a system state where $i$ does not exist (i.e. the notation $_{-i}$), and therefore, cannot influence $G(\vec{k}_{-i}, \hat{\Theta}_{-i})$ . Thus, the subtraction of those two elements is the disturbance caused by $i$ to the global system's utility.

**Algorithm 1** Reinforcement Learning Algorithm using Difference Utilities

Given $N, M, \Theta, H_i$
**for** Every agent in $N$ **do**
  Draw random_number $\sim U(0,1)$
  **if** random_number $> \epsilon$ **then**
    Agent searches her Q-table for strategy $(k_i, \hat{\theta}_i)$ that maximizes her Q-value
  **else**
    Agent chooses random strategy $(k_i, \hat{\theta}_i)$
  **end if**
**end for**
All agents reveal their own strategy $(k_i, \hat{\theta}_i)$ simultaneously
**for** Every agent in $N$ **do**
  Compute $D(i)$ (Equation 3.4)
  Update Q-table with: $Q(k_i, \hat{\theta}_i)_{t+1} = Q(k_i, \hat{\theta}_i)_t + \alpha\Big[D(i) - Q(k_i, \hat{\theta}_i)_t\Big]$
**end for**
Compute $G(K, \hat{\Theta})$
Update learning rate $\alpha$ and exploration rate $\epsilon$
Repeat until convergence criteria is met.

## 2.4 Summary on VCG and DU

In summary, considering the same congestion game, the VCG mechanism directly computes the optimal allocation whereas the multiagent system computes off-equilibrium incentives and iterate through the same game over and over. Figure 2.1 and 2.3 schematize the two processes.

# Chapter 3: Incremental Learning with Off-Equilibrium Incentives

## 3.1 Assumptions

In this section, we present the assumptions made by this approach. In the reminder of this work, we assume that the communications between the agents have no cost and the channels are safe. We assume that all agents have sufficient computational resources to carry out the computation of their payment, valuation and utility. Finally, we assume that the center is trustworthy.

Consider a general-sum repeated one-shot congestion game which allows the following properties:

**Property 1** (Ex Ante Symmetry)**.** *All agents in $N$ are symmetric, such that their type is sampled from the same distribution with the same parameters.*

**Property 2** (Multi-item Multi-unit homogeneity)**.** *All items are homogeneous, and have an unlimited quantity.*

**Property 3** (System Objective)**.** *The system objective $G$ is the sum of the values $v_i$ of every agent.*

**Property 4.** *The only information available to agent $i$ is her private preference $\theta_i$, and its Q-table.*

**Property 5** (Simultaneity)**.** *All agents declare their own strategy ($k_i$,$\hat{\theta}_i$) simultaneously*

**Property 6.** *All agents are guaranteed to obtain their chosen item $k_i$.*

**Property 7.** *The exploration rate $\epsilon$ and the learning rate $\alpha$ tend to zero as time goes to infinity.*

## 3.2 Equivalence of Difference Utilities and Truthfulness

We defined a multiagent system using Q-learning and the off-equilibrium incentives computed by DU in a general-sum repeated one-shot congestion game where properties 1 to

7 are true. In this section, we assume that DU has converged to an optimal allocation of items among the agents. We show that if the system has reached the optimal allocation, this multiagent system is strictly equivalent to a system computed through the VCG mechanism. This means that at convergence to the optimal allocation, the DU determines the true value of a item to a particular agent, while preserving the truthfulness property.

**Proposition 1.** *Assume a multiagent system using Q-learning and the off-equilibrium incentives computed by DU in a general-sum repeated one-shot congestion game where properties 1 to 7 are true. Therefore, if DU is at the optimal outcome, then DU computes payments, values and utilities equivalent to the VCG payments, values and utilities, for every agent respectively.*

*Proof.* Suppose a multiagent system using Q-learning and the off-equilibrium incentives computed by DU in a general-sum repeated one-shot congestion game where properties 1 through 7 are true. Then, the system objective $G$ is the sum of all individual valuations (Property 3).

$$G(\vec{k}, \hat{\Theta}) = \sum_{n \in \mathcal{N}} v_n(k_n, \hat{\theta}_n) \tag{3.1}$$

Then, we can show that the difference reward $D(i)$ is implemented the same way as $u_i$, the individual utility in the VCG framework (Equation 7).

$$\begin{aligned}
u_i(k_i, \hat{\theta}) &= v_i(k_i, x_{k_i}, \hat{\theta}_i) - p_i && (3.2)\\
&= v_i(k_i, x_{k_i}, \hat{\theta}_i) + \sum_{\substack{n \in N \\ n \neq i}} v_n(k_n, x_{k_n}, \hat{\theta}_n) - \sum_{n \in \mathcal{N}_{-i}} v_n(k_n^{-i}, \hat{\theta}_n^{-i}) && (3.3)\\
&= \sum_{n \in \mathcal{N}} v_n(k_n, \hat{\theta}_n) - \sum_{n \in \mathcal{N}_{-i}} v_n(k_n^{-i}, \hat{\theta}_n^{-i}) && (3.4)\\
&= G(\vec{k}, \hat{\Theta}) - G(\vec{k}_{-i}, \hat{\Theta}_{-i}) && (3.5)\\
&= D(i) && (3.6)
\end{aligned}$$

Thus, the utility, valuation and payment of the agent are implemented the same way as with the VCG mechanism.

However, since the agent chooses her own allocation (Properties 5 and 6), the outcome

of $u_i$, $v_i$ and $p_i$ is off-equilibrium (i.e. $u_i$ is computed with $p_i$ instead of $p_{VCG,i}$ in Equation 3.2). Altogether, they form $D(i)$, the off-equilibrium incentive given to agent $i$ at each iteration, as a reward for her action.

If we assume that this system is at an optimal allocation of items $\vec{k}^*$, then this set $K*$ is equivalent to the output of $g_{eff}(\hat{\Theta})$, in the VCG mechanism.

$$\vec{k}^* \equiv g_{eff}(\hat{\Theta}) \tag{3.7}$$

We also know from the VCG mechanism theory that the agents are guaranteed to be truthful, if given an optimal allocation. Since $\vec{k}*$ is an optimal allocation, we can write:

$$u_i(k_i^*, \theta_i) \geq u_i(k_i^*, \hat{\theta}_i) \tag{3.8}$$
$$\forall i, \theta_i \in \Theta, \hat{\theta}_i \in \hat{\Theta}, k_i^* \in \vec{k}*$$

Therefore, the truthfulness property holds when the algorithm converges to the optimal allocation $\vec{k}*$. □

## 3.3 Gradient to Efficient Outcome

We have shown in a particular class of game that DU can computes an equivalent system to the VCG mechanism when at the optimal point. In this section, we show how the off-equilibrium incentives push the system toward equilibrium, and how they are an off-equilibrium approximation of the VCG payment.

**Proposition 2.** *Assume a multiagent system using Q-learning and the off-equilibrium incentives computed by DU in a general-sum repeated one-shot congestion game where properties 1 to 7 are true. Then, the off-equilibrium incentives push the system toward equilibrium and compute an off-equilibrium approximation of the VCG mechanism.*

*Proof.* Suppose a multiagent system using Q-learning and the DU in a general-sum repeated one-shot congestion game where properties 1 through 7 are true. The theory of Q-learning states that an agent strives to maximize her expected reward [52]. In our approach, we replace this reward by the difference reward $D(i)$. So by using Q-learning, every agent in this system strives to maximize their difference reward $D(i)$. Thus, we need to analyze the effects of agent $i$'s behavior on the maximization of $D(i)$. To do so,

we take the first derivative of $D(i)$, using Equation 3.3:

$$\frac{\partial D(i)}{\partial \vec{a}_i} = \frac{\partial v_i}{\partial \vec{a}_i} + \frac{\partial \sum_{\substack{n \in \mathcal{N} \\ n \neq i}} v_n}{\partial \vec{a}_i} - \frac{\partial \sum_{n \in \mathcal{N}_{-i}} v_n}{\partial \vec{a}_i} \tag{3.9}$$

In Equation 3.9, the last term is null, because agent $i$'s actions[1] have no effect if $i$ does not exist in this system.

$$\frac{\partial D(i)}{\partial \vec{a}_i} = \frac{\partial v_i}{\partial \vec{a}_i} + \frac{\partial \sum_{\substack{n \in \mathcal{N} \\ n \neq i}} v_n}{\partial \vec{a}_i} \tag{3.10}$$

In Equation 3.10, the first term is the marginal effect of $i$'s action on her own valuation. This term is positive, thus $i$'s action must have a positive influence on her own value. The second term is the marginal effect of $\vec{a}_i$ over every other agents' valuation who played in the same game as $i$. This term is positive, thus $i$ must also have a net positive influence on the global system in order to maximize $D(i)$. In other words, if $i$ wants to maximize her own value, then $i$'s action must have a positive influence on her own value and the individual values of the other agents in this system.

But how does maximizing $D(i)$ pushes the system toward a more efficient solution? Going back to Proposition 1, and differentiating Equation 3.5 gives:

$$\frac{\partial D(i)}{\partial \vec{a}_i} = \frac{\partial G(\vec{k}, \hat{\Theta})}{\partial \vec{a}_i} - \cancelto{0}{\frac{\partial G(\vec{k}_{-i}, \hat{\Theta}_{-i})}{\partial \vec{a}_i}} \tag{3.11}$$

Again, the last term is null because agent $i$ does not exist in this system. Combining Equations 3.10 and 3.11:

$$\frac{\partial G(\vec{k}, \hat{\Theta})}{\partial \vec{a}_i} = \frac{\partial v_i}{\partial \vec{a}_i} + \frac{\partial \sum_{\substack{n \in \mathcal{N} \\ n \neq i}} v_n}{\partial \vec{a}_i} \tag{3.12}$$

By maximizing her own utility, agent $i$ maximizes the system objective $G$. This is true for every agent in this system. Thus, the off-equilibrium incentives push the system toward equilibrium and coupled with Proposition 1, compute an off-equilibrium approximation of the VCG mechanism. □

[1] The vector $\vec{a}_i$ is a short hand for the tuple $< k_i, \hat{\theta}_i >$, to illustrate the general action of $i$

## 3.4 Robustness to Manipulation

Robustness to manipulation is the difficulty for an agent to change the outcome of the system to her own benefit.

**Lemma 1.** *Assume a multiagent system using Q-learning and the off-equilibrium incentives computed by DU in a general-sum repeated one-shot congestion game where properties 1 to 7 are true. Then, robustness to manipulation by agents increases proportionally to the number of agents in the system.*

*Proof.* All the agents are symmetric (Property 1), and they compute their individual valuation $v_i$ the same way. Therefore, on average, each agent represents $1/n$ of the total sum of the valuation. Using the terms from Equation 7:

$$\frac{v_i(k_i, x_{k_i}, \hat{\theta}_i)}{G(\vec{k}, \hat{\Theta})} \propto \frac{1}{|\mathcal{N}|} \tag{3.13}$$

This means that for a non-trivial number of agents, no agent can maximize her off-equilibrium incentive (Equation 3.10) by maximizing only her own value. Since there is no communication possible between the agents (Properties 4 and 5), no group of agents can collude to pull toward such sub-optimal equilibrium either. Hence, as the number of agents increases, so does robustness to manipulation. □

# Chapter 4: Domains & Metrics

This section introduces the domains and metrics used in the experiments. We present two domains: a variation of the El-Farol Bar Problem, and the CubeSats domain. This variation of the El-Farol Bar Problem is a toy model that allows us to implement all the properties and constraints associated with our approach. The second domain demonstrates that the relaxation of some constraint still allow a good approximation of the social optimum in a coordination problem. The CubeSats domain is also a real-world application.

In a second part of this chapter, we present the metrics used to measure the truthfulness and efficiency of VCG and DU. The goal of this work is to show that the efficiency of a coordination problem can be increased by nudging the agents to tell the truth. Thus, we want to be able to measure the truthfulness of the agents, and also the efficiency of a system compared to the optimal solution provided by the center.

## 4.1 Domain 1: Variation of El Farol Bar Problem

The El Farol Bar problem originated in the work by W. Brian Arthur in 1994 [4]. It describes a finite population which wants to go to the El Farol Bar[1] "every Thursday". The El Farol bar is quite small. If more than a certain percentage $x$ of the population decides to go on Thursday night, then the patrons will all have a worse time than if they stayed at home, because it is too crowded. Reciprocally, if less than $x\%$ of the population chooses to go on Thursday night, they would all have had a better time staying home. To make this game interesting, everyone has to decide *simultaneously* if they will go Thursday night or not. No one can patiently wait to see how many people are going tonight, and try to maximize its individual utility.

In the case of a single-stage El Farol Bar problem[2], it has been shown by Arthur that there exists a unique symmetric Nash equilibrium of mixed strategies. Each player

[1] An actual bar in Santa Fe, New Mexico, USA

[2] In Game Theory, "single stage" means that the game is played only once, there is no repetition

chooses to go on Thursday with a known probability, which is function of the number of players. Variants of the player's profiles have been studied in [20].

Later on, Wolpert and Tumer [53] considered a multiple night version of the El Farol Bar problem, where each individual chooses amongst multiple night, which one to go to. They developed a reinforcement learning algorithm to solve this coordination problem. In this early version of the Difference Utilities, they choose to maximize the bartender utility. For each night, the bartender's utility is maximized when the bar is close to the optimal attendance for the bartender to work comfortably: not too busy, not too empty. In other words, the system objective ($G$) is the sum of every *night*'s utility given the attendance each night.

$$G = \sum_{m=1}^{|M|} x_m . e^{-x_m/c} \tag{4.1}$$

where $x_m$ is the attendance at night $m$ and $c$ is the optimal attendance in the bartender's opinion. This function is maximized if every $x$ is equal to $c$.

In this work, we use a multiple night variation of this model. Firstly introduced in David Parkes' study on learnable mechanism [37], this variation mainly changes the system objective. $G$ becomes the sum of the individual values of every agents in $N$. In words, this means that the system interest switched from the bartender to the patrons. In order to create a model of what would happen in a VCG auction, Parkes added a private preference for crowdedness. Each agent has a private preference $\theta_i$, which encode how much people they optimally would like to see when they go to the bar. Moreover, the individuals can lie about their preferences, in order to manipulate the system. This is captured by $\hat{\theta}_i$. The individual valuation $v_i$ is implemented with $x_{k_i}$ the attendance at the chosen night $k_i$, and the declared type $\hat{\theta}_i$ of agent $i$:

$$v_i(k_i, x_{k_i}, \hat{\theta}_i) = x_{k_i} . e^{-x_{k_i}/\hat{\theta}_i} \tag{4.2}$$

Therefore, the system objective is constructed as follow:

$$G(K, \hat{\Theta}) = \sum_{i=1}^{N} v_i(k_i, x_{k_i}, \hat{\theta}_i) \tag{4.3}$$

$$= \sum_{i=1}^{N} x_{k_i} e^{-x_{k_i}/\hat{\theta}_i} \tag{4.4}$$

### 4.1.1 Resimulation

The difference reward requires to compute the system objective of the game without each agent respectively. This might seem very computationally expensive, but in the case of the El-Farol Bar Problem, there is no need to re-simulate the entire game. To compute $G(z_{-i})$, it is sufficient to compute the sum of all individual valuations except $i$, and subtract 1 from the attendance of the night $i$ had previously chosen. Thus, every other agent who went the same night than $i$, will virtually see $i$ disappear of their valuation.

**Example:** Consider a very simple setup with 2 agents, who chose to go to the same night. Then, the difference reward of the agent will be

$$D(1) = \underbrace{x_{k_1}.e^{-x_{k_1}/\hat{\theta}_1}}_{x_{k_1}=2} + \underbrace{x_{k_2}.e^{-x_{k_2}/\hat{\theta}_2}}_{x_{k_1}=2} - \underbrace{x_{k_2}.e^{-x_{k_2}/\hat{\theta}_2}}_{x_{k_2}=1}$$

But all in all, this Variation of the El Farol Bar Problem is just a proof of concept and a toy model. In the next section, we present a real-world application of our incremental computation of VCG: the CubeSats domain. We construct two different models based on the CubeSats dynamics. Those models are more realistic, but also demonstrates the limits of our approach, based on the assumptions listed in Section 3.1

## 4.2 Domain 2: CubeSats

We use the CubeSats domain to demonstrate the approximation of VCG by the off-equilibrium incentives in a real-world context. The CubeSats are very small satellites, called picosatellites (about 10 cm by 10 cm by 10 cm), designed and deployed by the Stanford University OPAL microsatellite program [22]. A CubeSat is equipped with basic scientific instrumentation and launched into Low Earth Orbit (LEO) generally

between 350 and 750 km. The small size of a CubeSat limits its capabilities, and this is why it is crucial to optimize the utilization of its resource, especially when there is a multitude of CubeSats in orbit. An efficient coordination of multiple CubeSats can lead to an increase in the value of the CubeSats' resources and observations. Few solutions to this coordination problem have used learning algorithms [24].

### 4.2.1 System & Dynamics

**Points Of Interest** : Each POI $k \in M$ has a value $p_k$, which is the payment returned when observing $k$. At each iteration, $p_k$ is drawn from a uniform distribution $U \sim (0, 100)$, and the POIs' are randomly redistributed among the 10 areas.

**CubeSats** : We assume similar system dynamics as in [24]. Any given cube is in circular orbits around Earth. Its altitude is between 350 and 750 km randomly initialized at the beginning of the experiment. Each satellite travel 1/100th of its orbit at each time step (each time step correspond to a new occurrence of the game). The observation range of a satellite at each time step is a discretized area formed by a grid of 100x100, whom the cube is at the center.

The Q-table of an agent is constructed with 2 dimensions: all possible values of the POIs, and all possible deviations from $\theta_i$. The bid of a cube is a pair $(\hat{\theta}_i, k_i)$, a declared preference with a POI $k$. So when a cube receives a reward, the cube updates her Q-table at the indexes corresponding to $p_k$ and the deviation used to compute $\hat{\theta}_i$. Note that the individual preference $\theta_i$'s stay the same across iterations and statistical run.

**Center** : The center is an entity which facilitates communication between agents in different areas. The agents declare their bid by sending it to the center, which signal the other agents to modify their reward accordingly. It does not provide actions to the cubes.

### 4.2.2 CubeSats Model 1

In one round of the RL the cubes: (i) declare a bid $(\hat{\theta}_i, k)$ (ii) are signaled to compute $G$ and $G_{-i}$ (iii) receive their reward and update their Q-table. This model respects all the assumptions presented in Section 3.1. Specifically, we guarantee that the agents are

assigned to their chosen $k$. So it is possible for two agents to be assigned the same POI. In this case, both will carry out the observation, but the value will be divided by two. We write the individual valuation as:

$$v_{i,k}(k, \hat{\theta}_{i,k}) = \frac{p_k}{x_k} \times \exp - \frac{\frac{p_k}{x_k}}{\hat{\theta}_{i,k}} \tag{4.5}$$

- $p_k$: Value of POI $k$
- $x_k$: Number of Cubes assigned to POI $k$
- $\hat{\theta}_{i,k}$ : The *declared* preference of cube $i$ for POI $k$.

### 4.2.3 CubeSats Model 2

In one round of the RL: (i) the cubes declare a list of pairs $(\hat{\theta}_i, k)$ sorted by most wanted to least wanted (ii) the center uses a variation of the Gale-Shapley algorithm to solve this Stable Matching Problem (SMP) (iii) assigned cubes receive their reward and update their Q-table. This bidding structure is violating Property 6, because there is no guarantee anymore that a cube will receive her desired POI.

The individual valuation is written below. In this case, $\hat{\theta}_i$ represent the optimal value that a POI can have for a Cube. Therefore, this preference can encompass costs to retrieve the observation, distance to the POI, state belief... etc.

$$v_{i,k}(k, \hat{\theta}_{i,k}) = p_k \times \exp - \frac{p_k}{\hat{\theta}_{i,k}} \tag{4.6}$$

- $p_k$: Value of POI $k$
- $\hat{\theta}_{i,k}$ : The *declared* preference of cube $i$ for POI $k$.

We use a variation of the Gale-Shapley algorithm [18], where the number of POI and the number of agents are not necessarily equal. Because of the geographical distribution of the POIs and Cubes around the world, it is possible that some Cube and POI will remain unassigned at the end of this algorithm. When a cube is not assigned, the cube

does not receive any reward, and do not update her Q-table. This is because if we were to update the Q-table of a non-selected cube, the question is then which cell of the Q-table to update? Updating every Q-value corresponding to the dimensions chosen to build the list of bids does not make a lot of sense in term of learning. Since the Q-table starts with very small scrambled values, when the table is updated, it will only be for a choice that provided comprehensible feedback to the Cube.

**Algorithm 2** Variation of the Gale-Shapley algorithm to solve our Stable Matching Problem

Initialize all POIs $k \in M$ and cubes $i \in N$ to free
**while** free POI $k$ who still has a cube $i$ to propose to **do**
  Cube $i$ is POI $k$'s highest ranked among the cube to whom $k$ has not yet proposed
  **if** Cube $i$ is free **then**
    POI $k$ is assigned to Cube $i$
  **else**
    Some Cube $i$ has already been assigned to POI $k'$
    **if** $\hat{\theta}_{i,k'} < \hat{\theta}_{i,k}$ **then**
      POI $k$ is assigned to Cube $i$
      POI $k'$ is unassigned.
    **else**
      Cube $i$ has already been assigned to POI $k'$
      If Cube $i$ was the last cube in POI $k$'s list, then $k$ is marked as not free anymore, but remained unassigned. POI $k$ will not propose anymore.
    **end if**
  **end if**
**end while**

## 4.3 Metrics

We measure how the off-equilibrium incentives approach is approximating the VCG mechanism by accounting for two things: the efficiency of the system in term of system objective $G$, and the truthfulness of the system in terms of mean of the deviation from the truth.

We measure how close our approximation gets from the optimal system objective produced by the VCG mechanism, we simply compute the ratio of the system objective of DU over optimal system objective.

$$\frac{G(\vec{k}, \hat{\Theta})}{G(g_{eff}(\hat{\Theta}), \hat{\Theta})} \tag{4.7}$$

In addition, to measure the truthfulness of the cubes, we follow Parkes' method and compute the mean of the deviation from the truth [37]. Recall that $\hat{\theta}_i = h\theta_i$ where $h$ is the deviation from the truth. When $h = 1$, $\hat{\theta}_i = \theta_i$, which means that the agent is completely truthful. Thus, by measuring $h$, we know how close from truthfulness the system was at each iteration. The agent is completely truthful when $h^* = 1$. So the distance to the truth is $|h - h^*|$, and the mean of the deviation is written:

$$\frac{1}{|N|} \sum_{n \in \mathcal{N}} |h_n - h^*| \tag{4.8}$$

The lower this value is, the closer the agents are to be completely truthful during this game.

# Chapter 5: Empirical Results

## 5.1 Domain 1: Variation of El Farol Bar Problem

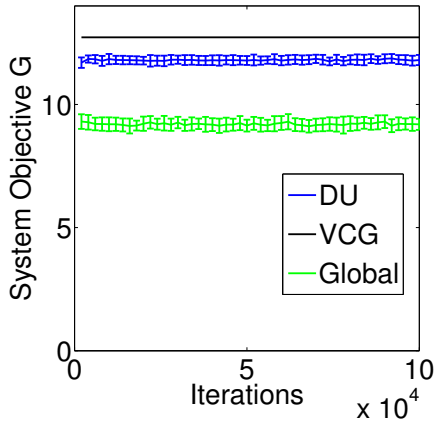
### 5.1.1 Incremental approximation of VCG

Here, we directly compare the off-equilibrium incentives with VCG. We generate a set of 15 agents' individual preferences $\Theta = \{1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 5\}$ and compute the WDP for 3 items. This is a very small system, yet this is the largest system for which we are able to compute the WDP using a standard computer. The optimal allocation produces a system objective $G(g_{eff}(\hat{\Theta}), \hat{\Theta}) = 12.7268$. Knowing $G(g_{eff}(\hat{\Theta}), \hat{\Theta})$, we can compare this value with the value of the system objective $G$ obtain with reinforcement learning using (i) off-equilibrium incentives and (ii) the standard global reward. This comparison is presented in Figure (5.1a), where we present the results over 100,000 iterations. The off-equilibrium incentives produce a system objective which approximates the value of the VCG system objective $G(g_{eff}(\hat{\Theta}), \hat{\Theta})$ within 10 to 5% in the last 10,000 iterations (see the performance ratio presented in Section 4.3). Moreover, the off-equilibrium incentives compute a more efficient system objective than a system using the global reward.

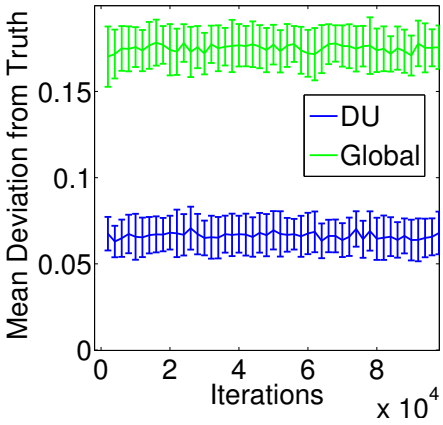
Additionally, Figure (5.1b) displays the mean of the deviation from the truth (See Equation 4.8 in Section 4.3). This curve is decreasing and converges to values between 0.06 and 0.08. Since complete truthfulness is at 0 (No deviation from the truth), those values indicate that the agents converge toward a significantly "honest" system, in which agents are almost completely truthful. Furthermore, the off-equilibrium incentives achieves a mean of the deviation closer to complete truthfulness than a system using global reward.

### 5.1.2 Scalability

In this section, we study the scalability of the off-equilibrium incentives. Our goal is to show that the off-equilibrium incentives approach can compute systems that are signif-




(a) Comparison of the System Objective $G$ of Variation of El Farol Bar Problem using DU and VCG



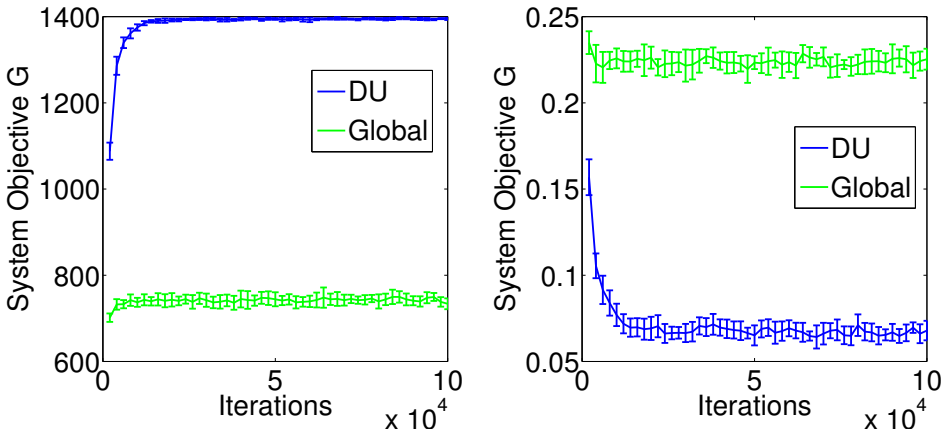

(b) Mean of the Deviation from the Truth

Figure 5.1: Variation of El Farol Bar Problem using DU with 15 agents and 3 items - Parameters: $\alpha = 0.9$, $\epsilon = 0.01$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5\}$, 100,000 iterations

icantly too large for a tractable WDP. We also that the agents are pushed toward an efficient outcome, while leaning toward partial truthfulness.

We presents results from a system with 100 agents and 10 nights, computed with the off-equilibrium incentives and the global reward. This system is too large to compute an exact solution using the WDP.

Figures (5.2a) and (5.2b) display the results from this experiment. We observe that the off-equilibrium incentives perform significantly better than a system using global reward. Additionally, the off-equilibrium incentives pushes the agents toward a preferable almost complete truthfulness, in comparison with the global reward. This shows that this approach is : (i) preferable to the global reward (ii) scalable to large system where VCG is intractable (iii) push the agents toward more efficient outcome in terms of system objective and truthfulness.




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

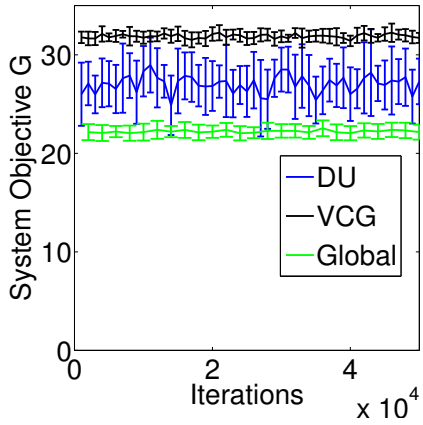
Figure 5.2: Variation of El Farol Bar Problem using DU with 100 agents and 10 items - Parameters: $\alpha = 0.9$, $\epsilon = 0.01$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5\}$, 100,000 iterations

But all in all, this Variation of the El Farol Bar Problem is just a proof of concept and a toy model. In the next section, we will present results from a real-world application: the CubeSats domain.

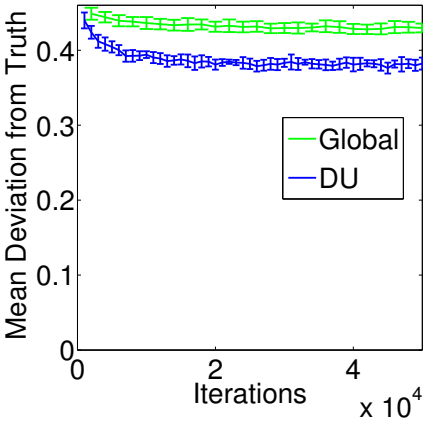
## 5.2 Domain 2: CubeSats Domain

In this section, we study a considerably more complex model in the CubeSats domain (see Section 4.1). We analyze various settings to observe the effect of large scale systems, highly congested and abundant systems, their computational complexity and finally, the limitations of the off-equilibrium incentives in this model.

### 5.2.1 Incremental approximation of VCG



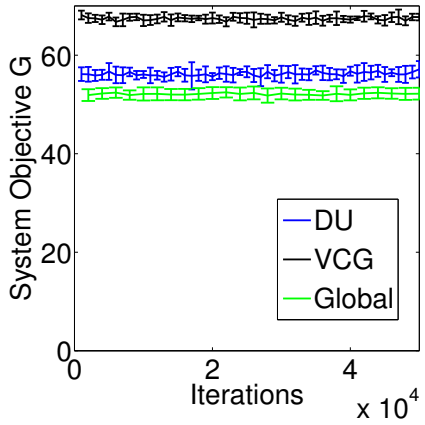

(a) Comparison of System Objective $G$ obtained with DU, and System Objective obtained with VCG



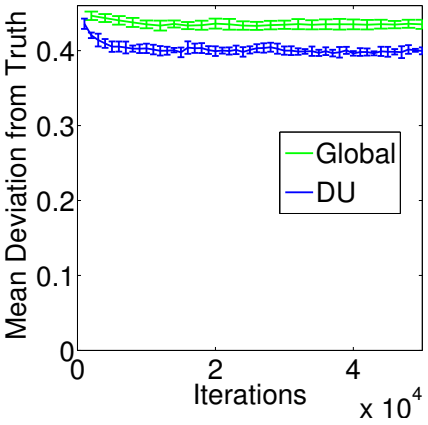

(b) Mean of the Deviation from the Truth

Figure 5.3: CubeSat system with 15 cubes and 2 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 50,000 iterations, 10 areas

First, we want to directly compare the off-equilibrium incentives based approximation with the VCG mechanism for the same system. We start with a small system of 15 cubes randomly distributed among 10 areas around the globe. At each iteration, the values and positions of the POIs are sampled again, and the cubes move according to their




(a) Comparison of System Objective $G$ obtained with DU, and System Objective obtained with VCG




(b) Mean of the Deviation from the Truth

Figure 5.4: CubeSats system with 15 cubes and 7 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 50,000 iterations, 10 areas.

own trajectory. This means that the allocation problem is different at each iteration. So we compute the WDP at each iteration, for each set of item's values to determine the optimal allocation and system objective $G(g_{eff}(\hat{\Theta}), \hat{\Theta})$.

In Figures (5.3) and (5.4) compare the VCG's system objective with off-equilibrium incentives' system objective and the global reward system objective over $50,000$ itera-

tions. Those two experiments have different degrees of congestions. Figures (5.3a) and (5.3b) are results from a system with more than 90% of congestion (2-to-15 quantity of POI for agents). Figures (5.4a) and (5.4b) have 50% of congestion, with 7 POIs for 15 cubes.

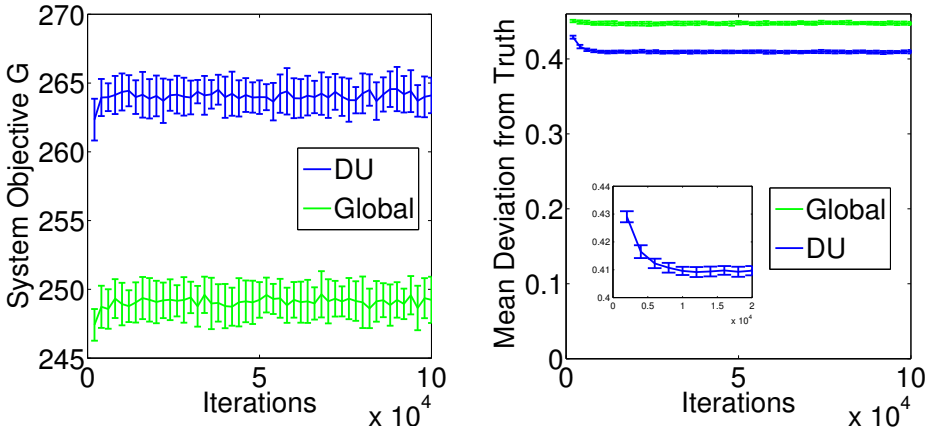
Figure (5.3a) indicates that the off-equilibrium incentives approximate the VCG mechanism between 85 and 90% after a small number of iteration. In addition, Figure (5.3b) displays a decrease in the mean of the deviation from the truth. However, the mean of the deviation from the truth does not converge as well as in the variation of the El-Farol Bar Problem. We believe this is due to the fact that this domain is significantly more complex and dynamic than the Bar Problem. In the CubeSats domain, not every range of value is available at every iteration, making it unlikely for a cube to maximize her individual valuation. Similarly, Figures (5.4a) and (5.4b) show that the off-equilibrium incentives approximate the system objective of the VCG mechanism around 85%, while the mean from the deviation from the truth converges to a sub-optimal value. Additionally, the off-equilibrium incentives always perform more efficiently than the global reward.

### 5.2.2 Scalability

The main advantage of a multiagent system using reinforcement learning is that it can trivially compute significantly larger systems than the VCG mechanism. In this section, we present the results obtained with a CubeSats system of 100 cubes and $1,000$ cubes. In each setting, we look at 90% and 50% of congestion, as previously. It is important to note that those systems are intractable for the VCG mechanism. We demonstrate that using the off-equilibrium incentives also leads to better results than the global reward.

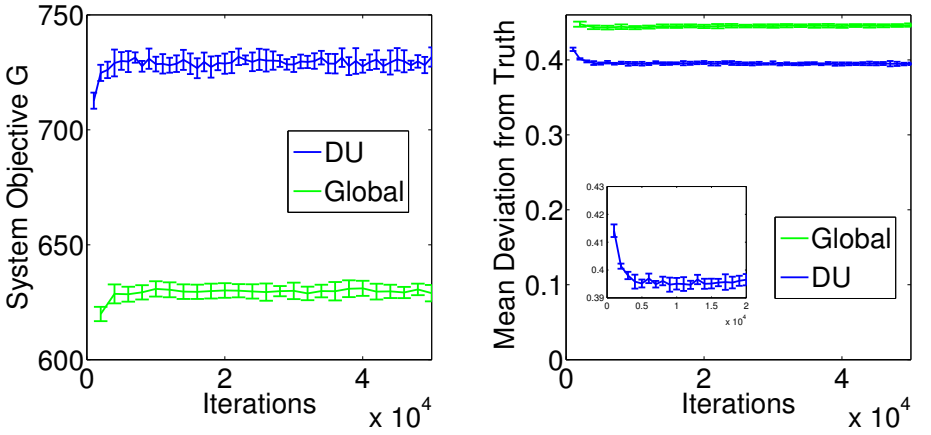
Figures (5.5a) and (5.5b) shows the system objective $G$ and the mean of the deviation from the truth of an experiment with 100 cubes and 10 POIs. We observe the expected behavior of the mean of the deviation from the truth converging toward a sub-optimal value. Again, we believe this is an inherent outcome of a dynamic system like the CubeSats. We observe identical results when we decrease the congestion to 50% (Figures 5.6a and 5.6b). Additionally, the off-equilibrium incentives always perform more efficiently than the global reward.

We can push the model even further. Figures (5.7a), (5.7b), (5.8a), and (5.8b) display




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

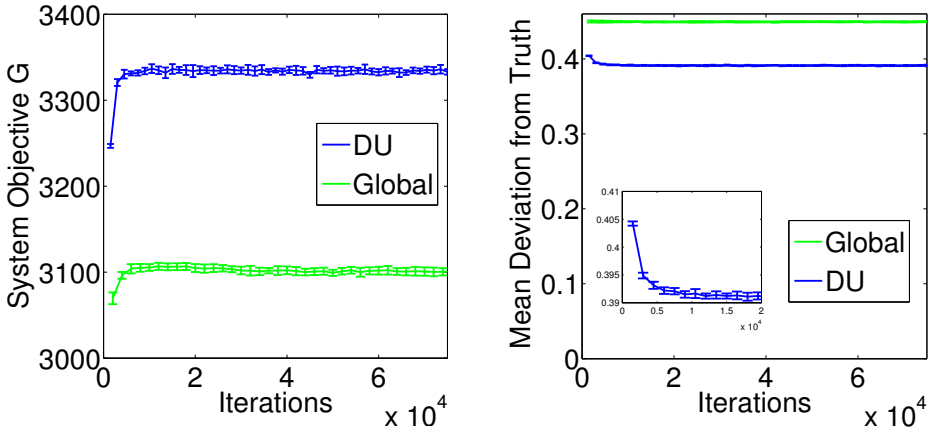
Figure 5.5: CubeSat system using DU with 100 cubes and 10 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 100,000 iterations, 10 areas.




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

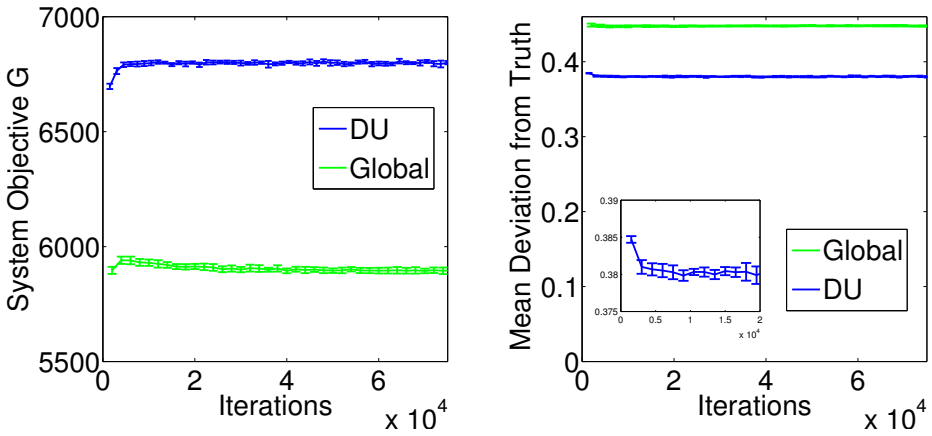
Figure 5.6: CubeSat system using DU with 100 cubes and 50 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 50,000 iterations, 10 areas.

the results obtained with 1,000 cubes, and respectively 90% and 50% of congestion. We obtained similar behavior from the system as previously presented. Additionally, the off-equilibrium incentives always perform more efficiently than the global reward.




(a) System Objective $G$. (b) Mean of the Deviation from the Truth

Figure 5.7: CubeSat system using DU with 1,000 cubes and 100 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 75,000 iterations, 10 areas.

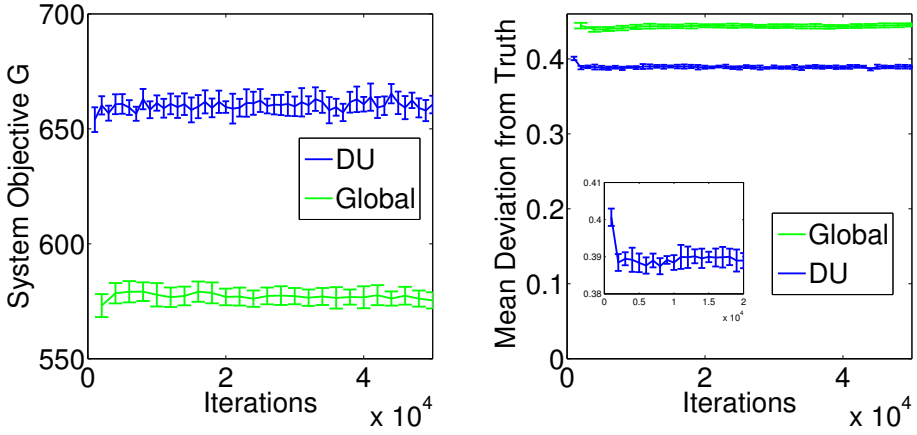



(a) System Objective $G$ (b) Mean of the Deviation from the Truth

Figure 5.8: CubeSat system using DU with 1,000 cubes and 500 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 75,000 iterations, 10 areas.

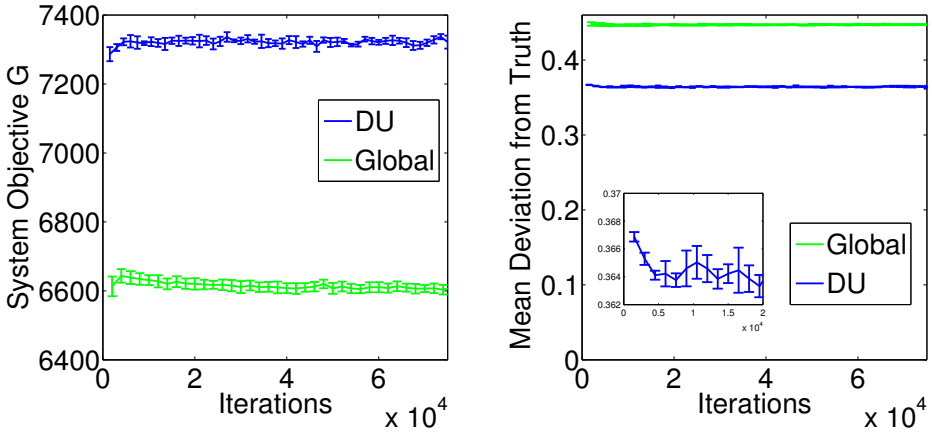
### 5.2.3 Congestion

In this section, we present the results obtained with we decrease the congestion. At this point, every Cube have at least one POI available. The experiments include 100




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

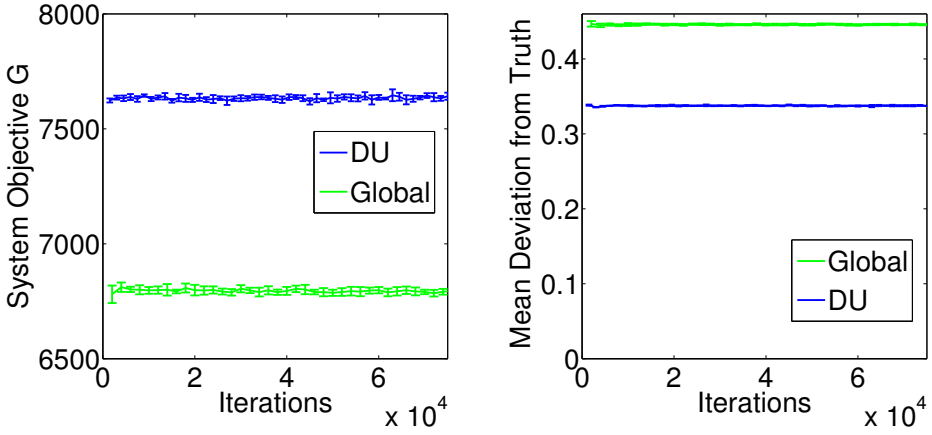
Figure 5.9: CubeSat system using DU with 100 cubes and 100 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 50,000 iterations, 10 areas.




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

Figure 5.10: CubeSat system using DU with 1,000 cubes and 1,000 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 75,000 iterations, 10 areas.

and 1,000 cubes, with a congestion of 1-to-1, and 5-to-1 (5 POIs for one cube). It is important to note that this is a congestion game, and according to Property 6, the




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

Figure 5.11: CubeSat system using DU with 1,000 cubes and 5,000 POI's - Parameters: $\alpha = 0.8$, $\epsilon = 0.2$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 75 000 iterations, 10 areas.

agents are guaranteed to obtain the POI they asked. We know that the cubes' individual valuation (Equation 4.5) are influenced by the attendance at one cube. Thus, when the competition for observing one POI decreases, we expect the attendance for each POI to decrease.

Results are showed in Figures 5.9, 5.10 and 5.11. Again, in every setting, the off-equilibrium incentives perform more efficiently than the global reward, both in terms of system objective and truthfulness. And again, we observe a convergence toward partial truthfulness from the agents. It's crucial to note that VCG mechanism is intractable in this setting.

### 5.2.4 Computational Complexity

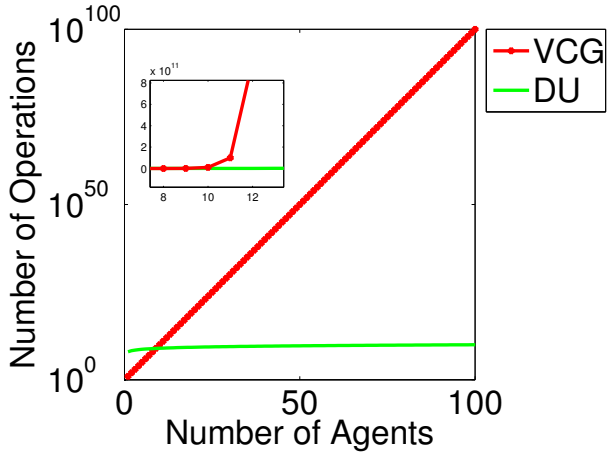


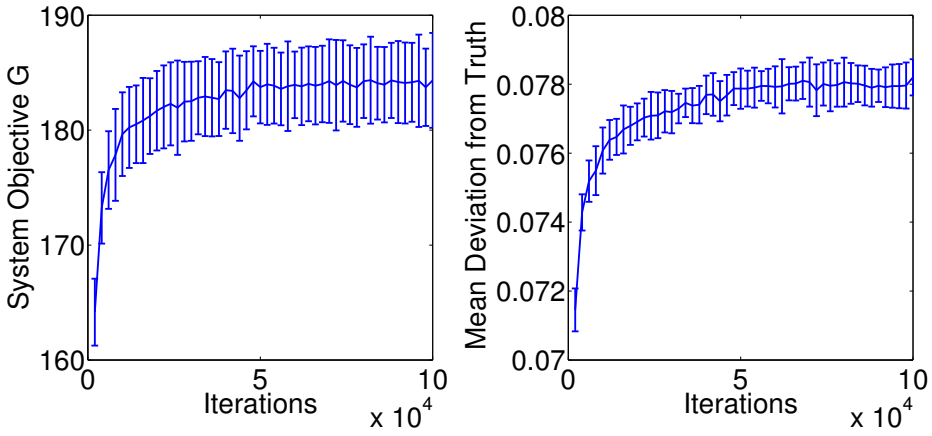

Figure 5.12: Comparison of Number of Operation to Complete Game - Parameters: 10 POIs, 10 possible deviation from truth, 100 possible values of POIs, 100,000 iterations during RL

Previous sections show that off-equilibrium incentives reasonably approximate the outcomes of VCG. The main advantage of this approach is its computational complexity in comparison with VCG. In order to solve the WDP, the center computes every possible combination of items and agents, so the computational complexity is $O(m^{|\mathcal{N}|})$ (where $m$ is the number of items). In the case of the CubeSats using RL, the complexity is based on the search for the best strategy in the Q-table. The Q-table's dimensions are: (i) all possible values of POIs and (ii) all possible deviations in $\mathcal{H}$. Each agent has a separate Q-table, and the search occurs at every iteration. The trick is that the computation of $G_{-i}$ does not require an actual re-simulation. Removing agent $i$ from the system is equivalent to removing agent $i$ from the attendance ($x_k$) of the POI chosen by $i$.

Figure 5.12 presents a comparison in the number of operations needed to compute one CubeSats simulation depending on the number of agents. Note that the scale of the number of operation in **logarithmic**. The computation is significantly cheaper for a non-trivial number of agent when using our approach, even with conservative parameters.

### 5.2.5 Limitations

In this section, we show what happen when we leave the domain of congestion games. In Section 3.1, we presented several assumptions that are required in order to define a congestion game in which the off-equilibrium incentives can compute a gradient leading the agents to more efficient outcome. Those assumptions are strong, especially Property 6. Thus, we remove this property from the game, therefore being in an auction setting, instead of a pure congestion game. If we modified the CubeSat model presented in Section 4.1 from a congestion game to an auction, then we obtain the CubeSat model developed in Section 4.2.3. This CubeSat model does not satisfy Property 6.




(a) System Objective $G$ (b) Mean of the Deviation from the Truth

Figure 5.13: CubeSat system using DU with 100 cubes and 10 POI's - Parameters: $\alpha = 0.85$, $\epsilon = 0.15$, $\mathcal{H} = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, 100,000 iterations, 10 areas.

Now, the cubes declare a list of pairs $(\hat{\theta}_i, k)$ sorted from most preferred to least preferred of every POI present in range. So, in one round of RL (i) the cubes declare their list of pairs (ii) the center allocates one cube per POI by solving this Stable Matching Problem (SMP) using the Gale-Shapley algorithm [18] (the POIs propose to their higher $\hat{\theta}_i$ first) (iii) the allocated cubes receive their reward and update. The center re-solves the SMP, skipping agent $i$ during allocations, to compute $G_{-i}$. This is an expensive computation (In this case, SMP is $O(|\mathcal{N}|m)$ where $m$ is the number of items), but this is still lesser than VCG for larger systems.

Figure 5.13 exhibits the results obtained with a CubeSat system of 100 cubes and 10 POIs using the off-equilibrium incentives. We observe that when removing Property 6, the agents are not pushed toward truthfulness anymore. On the contrary, they are now almost completely not truthful. Thus, the off-equilibrium incentives - as they are implemented right now - should not be used in non-congestion games.

Future work should study how to shape the individual valuation such that the off-equilibrium incentives can be used without Property 6, and therefore also be applied in an auction.

# Chapter 6: Conclusion

During congestion games, selfish agents will try to manipulate the system for their own interest, often time at the expense of the social welfare. This manipulation can be prevented using the VCG mechanism, which internalizes the externalities caused by an agent by computing the proper payment. The VCG must guarantee the optimal allocation given all the agent's preference. This optimal allocation is based a key step called the WDP, which requires the computation of all possible allocation of item to agent. This step is extremely computationally expensive, and make VCG an unpractical mechanism for non-trivial number of agents.

This work proposes a multiagent learning system using Reinforcement Learning and off-equilibrium incentives to incrementally approximate the VCG mechanism outcome. The off-equilibrium incentives compute a personalized reward which internalizes the marginal effect of an agent on the system. These individualized rewards are used as gradients to iteratively move toward equilibrium at each round of the RL.

We prove theoretically that if a system using DU is at equilibrium, then the off-equilibrium incentives are now at equilibrium and compute an equivalent payment to the VCG payment. Then, we demonstrate that while we cannot prove convergence of a multiagent system to an optimal outcome, we can show that the off-equilibrium incentives force every agent to maximize the value of the system objective $G$. Finally, we show that the system's robustness to manipulation increases with the number of agents.

Additionally, we test the off-equilibrium incentives using two domains: (i) a variation of the El-Farol Bar Problem used as proof of concept (ii) and the CubeSats domain, a dynamic real-world application. The results obtained from the variation of the El Farol Bar Problem indicate that the off-equilibrium incentives can reasonably approximate the VCG mechanism, while preserving significantly the truth. Additionally, the CubeSats results show that congestion games with dynamic settings can be approximated using the off-equilibrium incentives. This domain is significantly more challenging than the Bar Problem due to the values assigned on each POI and the randomness of the availability of each value (because we sample the value of each POI at each iteration). Moreover,

we analyze the system's reaction to different level of congestion and size. It proves that (i) this approach is trivially scalable for larger systems for which VCG is intractable; (ii) and induces the trade-off computational complexity vs exact answer; (iii) the off-equilibrium incentives always perform more efficiently than the global reward. Finally, we tested the limits of the off-equilibrium incentives when we leave congestion games for more general auctions. This means that relaxing the assumption made by Property 6 cancel the incentive toward truthfulness to the agents.

This limitation brings more questions. It is very likely that the Property 6 is heavily dependent on the implementation of the individual valuations. In the case of congestion games, it is the need for congestion itself that allows for this assumption to work. Future works could include the study of individual valuations such that the off-equilibrium incentives performs in auction domains.

Future work will include the analysis of more various types of games and individual valuations in which this concept can be applied.

# Bibliography

[1] A.K. Agogino and K. Tumer. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains. *Autonomous Agents and Multi-Agent Systems*, 17(2):320–338, 2008.

[2] A.K. Agogino and K. Tumer. A multiagent approach to managing air traffic flow. *Autonomous Agents and Multi-Agent Systems*, pages 1–25, 2012.

[3] M. Andres Figliozzi, H.S. Mahmassani, and P. Jaillet. Framework for study of carrier strategies in auction-based transportation marketplace. *Transportation Research Record: Journal of the Transportation Research Board*, 1854(1):162–170, 2003.

[4] W.B. Arthur. Inductive reasoning and bounded rationality. *The American economic review*, 84(2):406–411, 1994.

[5] D. P. Bertsekas and D. A. Castanon. The auction algorithm for the transportation problem. *Annals of Operations Research*, 20(1):67–96, 1989.

[6] S. J. Brams. Mathematics and democracy: Designing better voting and fair-division procedures. *Mathematical and Computer Modelling*, 48(9):1666–1670, 2008.

[7] J. Bredin, R. T Maheswaran, C. Imer, T. Başar, D. Kotz, and D. Rus. A game-theoretic formulation of multi-agent resource allocation. In *Proceedings of the fourth international conference on Autonomous agents*, pages 349–356. ACM, 2000.

[8] C. Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2003.

[9] N. Chentanez, A. G Barto, and S. P. Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2004.

[10] E. H Clarke. Multipart pricing of public goods. *Public choice*, pages 17–33, 1971.

[11] V. Conitzer. Auction protocols. In *Algorithms and theory of computation handbook*. Chapman & Hall/CRC, 2010.

[12] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the Eighteenth conference on UAI*, pages 103–110. Morgan Kaufmann Publishers Inc., 2002.

[13] P. Cramton. How best to auction oil rights. 2007.

[14] S. De Vries and R.V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.

[15] C. Dellarocas. Reputation mechanism design in online trading environments with pure moral hazard. *Information Systems Research*, 16(2):209–230, 2005.

[16] R. Engelbrecht-Wiggans. State of the artauctions and bidding models: a survey. *Management Science*, 26(2):119–142, 1980.

[17] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism design for policy routing. *Distributed Computing*, 18(4):293–305, 2006.

[18] D. Gale and L. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[19] A. Galstyan, K. Czajkowski, and K. Lerman. Resource allocation in the grid using reinforcement learning. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1314–1315. IEEE Computer Society, 2004.

[20] H. Gintis. *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton University Press, 2000.

[21] T. Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[22] H. Heidt, J. Puig-Suari, A. S. Moore, S. Nakasuka, and R. J. Twiggs. Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation. In *Proceedings of the 14th Annual AIAA/USU*, pages 1–19, 2000.

[23] T. Hester and P. Stone. Intrinsically motivated model learning for a developing curious agent. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[24] C. HolmesParker and A. Agogino. Agent-based resource allocation in dynamically formed cubesat constellations. AAMAS '11, pages 1157–1158, 2011.

[25] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. Biddingbot: a multiagent support system for cooperative bidding in multiple auctions. In *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, pages 399–400. IEEE, 2000.

[26] D. Lehmann, L.I. O'callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM (JACM)*, 49(5):577–602, 2002.

[27] T Luiz, L. A Barroso, et al. *Electricity auctions: An overview of efficient practices.* World Bank Publications, 2011.

[28] E. S Maskin. Mechanism design: How to implement social goals. *The American Economic Review*, 98(3):567–576, 2008.

[29] R. P. McAfee and J. McMillan. Auctions and bidding. *Journal of economic literature*, 25(2):699–738, 1987.

[30] R. P. McAfee and J. McMillan. Analyzing the airwaves auction. *The Journal of Economic Perspectives*, 10(1):159–175, 1996.

[31] P. Milgrom. Auctioning the radio spectrum. *Auction Theory for Privatization*, 1995.

[32] P. Milgrom. *Putting auction theory to work*. Cambridge University Press, 2004.

[33] P. R. Milgrom. The economics of competitive bidding: a selective survey. *Social goals and social organization: Essays in memory of Elisha Pazner*, pages 261–292, 1985.

[34] W. S. Misiolek and H. W. Elder. Exclusionary manipulation of markets for pollution rights. *Journal of Environmental Economics and Management*, 16(2):156–166, 1989.

[35] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research*, 29(1):19–47, 2007.

[36] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[37] D.C. Parkes. On learnable mechanism design. *Collectives and the design of complex systems*, pages 107–131, 2004.

[38] D.C. Parkes and J. Shneidman. Distributed implementations of vickrey-clarke-groves mechanisms. AAMAS'04, pages 261–268, 2004.

[39] S. D Ramchurn, D. Huynh, N. R. Jennings, et al. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[40] S. J Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, pages 402–417, 1982.

[41] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[42] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinational auctions. *Management science*, 44(8):1131–1147, 1998.

[43] S. Russell, P. Norvig, and E. Davis. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Englewood Cliffs, 2010.

[44] T. Sandholm. Limitations of the vickrey auction in computational multiagent systems. In *Proceedings of the Second International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306, 1996.

[45] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1–54, 2002.

[46] T. Sandholm and S. Suri. Bob: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*, 145(1):33–58, 2003.

[47] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Cabob: A fast optimal algorithm for winner determination in combinatorial auctions. *Management Science*, 51(3):374–390, 2005.

[48] J. Shneidman and D.C. Parkes. Using redundancy to improve robustness of distributed mechanism implementations. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 276–277. ACM, 2003.

[49] Y. S. Son, R. Baldick, K. Lee, and S. Siddiqi. Short-term electricity market auction game analysis: uniform and pay-as-bid pricing. *Power Systems, IEEE Transactions on*, 19(4):1990–1998, 2004.

[50] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[51] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, pages 8–37, 1961.

[52] C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[53] D.H. Wolpert, K.R. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *EPL (Europhysics Letters)*, 49:708, 2000.