12-2022

Decentralized Resource Allocation through Constrained Centroidal Voronoi Tessellations

Bhagyashri Telsang
btelsang@utk.edu

Recommended Citation

Telsang, Bhagyashri, "Decentralized Resource Allocation through Constrained Centroidal Voronoi Tessellations. " PhD diss., University of Tennessee, 2022.
https://trace.tennessee.edu/utk_graddiss/7683

To the Graduate Council:

I am submitting herewith a dissertation written by Bhagyashri Telsang entitled "Decentralized Resource Allocation through Constrained Centroidal Voronoi Tessellations." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Seddik Djouadi, Major Professor

We have read this dissertation and recommend its acceptance:

Seddik Djouadi, Husheng Li, Xiaopeng Zhao, Dan Wilson

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Decentralized Resource Allocation through Constrained Centroidal Voronoi Tessellations

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Bhagyashri Telsang

December 2022

To Rudra

Acknowledgments

I would like to express my deepest appreciation and gratitude to my academic advisor Dr. Seddik Djouadi for the great guidance during my study. His constant and unrelenting support allowed me to maintain my focus on my research and grow not only as a researcher but also as a person. His encouragement to pursue and explore new paths has enabled me to venture deeper into unexplored territories and bring forth some scientific contributions to the community. He understood me as a person and tailored his mentoring accordingly to bring out the best in me. I sincerely thank you for being the best advisor I could ask for.

I would also like to thank my committee – Dr Husheng Li, Dr Xiaopeng Zhao and Dr Dan Wilson – for their time and dedication, helpful comments and suggestions in shaping my dissertation. I would also like to extend my gratitude to Dr Teja Kuruganti, Dr Jin Dong, and Dr Olama, for all the discussions, ideas and the funding without which this work would not have been possible.

In addition, I would like to thank all my colleagues for their assistance and friendships, and the pleasant and positive atmosphere in the workplace, albeit briefly interrupted by Covid lockdowns! This journey has been so exciting and rewarding due to my longest and ever-loving friends Siddharth Mayya and Dheeraj Velicheti, who were always ready for long discussions over crazy ideas, and for growing together. Without all the love and support from my partner Jason Rice who has patiently absorbed my madness and fostered a creative environment for my research, the past few years would have been very grueling. And last but not the least, my sincere love and eternal gratitude to my parents Dr Shrikant Telsang and Dr Jayashri Telsang, and my sister Dr Swarali Telsang, to whom I attribute every positive element in my life; without them I cannot even imagine my life for they give me the strength to live and fight to try to make this world a better place.

Abstract

The advancements in the fields of microelectronics facilitate incorporating team elements like coordination into engineering systems through advanced computing power. Such incorporation is useful since many engineering systems can be characterized as a collection of interacting subsystems each having access to local information, making local decisions, interacting with neighbors, and seeking to optimize local objectives that may well conflict with other subsystems, while also trying to optimize a certain global objective. In this dissertation, we take advantage of such technological advancements to explore the problem of resource allocation through different aspects of the decentralized architecture like information structure in a team.

Introduced in 1968 as a challenging toy example in the field of team decision theory to demonstrate the significance of information structure within a team, the celebrated Witsenhausen counterexample remained unsolved until the analytical person-by-person optimal solution was developed within the past decade. We develop a numerical method to implement the optimal laws and show that our laws coincide with the optimal affine laws. For the region where the optimal laws are non-linear, we show that our laws result in the lowest costs when compared with previously reported costs.

Recognizing that, in the framework of team decision theory, the difficulties arising from the non-classical information structure within a team currently limit its applicability in real-world applications, we move on to investigating Centroidal Voronoi Tessellations (CVTs) to solve the resource allocation problem. In one-dimensional spaces, a line communication network is sufficient to obtain CVTs in a decentralized manner, while being scalable to any number of agents in the team.

We first solve the static resource allocation problem where the amount of resource is fixed. Using such a static allocation solution as an initialization step, we solve the dynamic resource allocation problem in a truly decentralized manner. Furthermore, we allow for flexibility in agents' embedding their local preferences through what we call a *civility model.* We end the dissertation by revisiting the application of Demand-response in smart grids and demonstrate the developed decentralized dynamic resource allocation method to solve the problem of power allocation in a group of building loads.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This is a dissertation document probing the field of multi-agent systems through different approaches. We first begin with our motivation to work in this field in Section 1.1. Then in Section 1.2, we introduce three approaches – centralized, team decision theory, and centroidal Voronoi tessellations – and explore their background and usefulness to the resource allocation problem. We list the contributions from this dissertation in Section 1.2.1 and outline the rest of the document in Section 1.3.

1.1 Motivation

"Global trendsetting, local negotiations"

Team coordination is a ubiquitous phenomenon in nature: species of birds in flocks performing graceful formations, ant colonies regulating their foraging behavior, swarm of bees constructing perennial colonial nests from wax in large size colonies, are just a few among abundant, fascinating examples found in nature. Humans, as a species, are also capable of incredible coordination – building civilizations, exploring outer space, drilling towards the earth's core, building bridges, all of which are team efforts. While humans have a highly complex brain that can support advanced activities, it is truly fascinating to observe the level of coordination that is present in bees or ants, [55].

In fact, by studying ants as models, researchers hypothesize that the human brain shrinkage parallels the expansion of collective intelligence in human societies, [80].

With new advancements in the fields of microelectronics and miniaturization, it is natural to wonder about the potential advantages of incorporating such coordination in engineering systems. In fact, many engineering systems can be characterized as a large scale collection of interacting subsystems each having access to local information, making local decisions, having local interactions with neighbors, and seeking to optimize local objectives that may well be in conflict with other subsystems, while also trying to optimize certain objective of the entire large scale system. The potential benefits of distributed decision architectures include the opportunity for real-time adaptation and robustness to dynamic uncertainties such as individual component failures, non-stationary environments, and adversarial elements. With increasing interest in groups of embedded systems, the field of multi-agent control has grown rapidly in the past few decades, [18].

The field of control systems, over the past seven decades, has developed various control methods each with their specializations. For example, model predictive control (MPC) was developed for control in automation plants and is still widely used in such applications, [15]. While there still remains a lot of room for development in controllers for single plants – like adaptive control – various developed controllers suffice for a satisfactory control of simple subsystems in a multi-agent, or multi-plant, setup. For example, a PID controller performs well in general in controlling a line following robot.

Equipping an agent with a satisfactorily performing controller allows us to introduce and deal with concepts like a team of agents. Under such architecture we can move from an individual agent to a higher level of multiple agents where we can observe and model the interactions between them. The team can then be coordinated and controlled to achieve complex tasks that the agents would be unable to achieve individually, [43]. Like the concept of *Emergence*, we can obtain a sum that is greater than its parts. Humankind has transitioned from being just one of the species to taking over the planet; the ability of cooperation and coordination in humans is largely responsible for it. Similarly, the path forward for the autonomous agents is to team up, and for us as the designers, a vast, largely unexplored field of multi-agents systems awaits.

The envisioned groups of agents can be endowed with communication, sensing and computation capabilities, and promise great efficiency in the realization of even multiple tasks, for example, environmental monitoring, exploratory missions, search and rescue operations. Depending on the application, on the team architecture, on the agent properties, there are different approaches to the problem of multi-agents systems – for example, team decision theory, distributed control, swarm robotics, decentralized control, [26].

However, the question of the scale of decentralization is not only central to all the aforementioned fields, but also a philosophical one. In complete absence of any central element, the actions of individual agents could be meaningless and classified as chaos. In the quoted examples of team coordination in nature, there is an objective that all the bees are working towards, there is a certain global trend that has been set, and the bees work locally to maintain the trend. Subjective as it may be, the amount of centralization necessary to achieve a set objective also depends on the application at hand.

In this dissertation, we consider the problem of resource allocation and provide different solutions to it that vary in their decentralization. As we navigate the world in our day-to-day life, we are constantly presented with a conflicting and paradoxical narrative of excess and scarcity coexisting simultaneously. Despite the tons of food wasted everyday, hunger and starvation, especially among kids, remain a significant global issue. We constantly notice blocks of vacant houses ready to be occupied, but a large number of people being able to live only on the street by those houses and not in them. Exacerbated by climate change, there are regions facing the worst floods and other regions facing their worst drought. In the middle of such conditions, the world is on the brink of food shortage while there are grains rotting in silos. We can continue naming several more examples, but in each of them there is a resource that is not being allocated in the “right” way. While the causes for global crises like food or power shortage are beyond an individual's control, the effects are certainly reverberated through all the levels. Performing resource allocation in the "right" way in that case, would at the very least shield local interests from such global crises up to an extent. In order to explore the problem of resource allocation in a decentralized way, we propose a framework based on Centroidal Voronoi Tessellations (CVTs) due to their ability to inherently capture a global behavior. More importantly, the framework of CVTs allows for simple communication network in a team of agents to reach the global optimal. To delve into the framework of CVTs for resource allocation, we first begin with gaining some background on the aforementioned fields and their relevance to this dissertation.

1.2 Background

We begin by considering the problem of resource allocation through Demand-Response in smart grids. Each agent is a power consuming load that needs to operate at a prescribed setpoint, and as a team of multiple agents, their combined task is to consume a certain amount of power together. The reason we choose this problem is to facilitate the integration of renewable energy in the day-to-day consumption by employing all the energy generated from a renewable source locally in a set of local loads. In this work, we consider building loads because of their flexibility, and their share of participation in the electricity market – hence resulting in a greater impact on the overall integration. Consequently, it is also a step towards making buildings more energy efficient.

There have been promising developments in the area of energy-efficient buildings. For example, [6] and [1], developed an adaptive control approach to maintain the indoor air temperatures of the building at a desirable temperature. [22] formulated an augmented optimal model predictive control (MPC) methodology that is suitable not only for controlling the building indoor air temperatures but also for handling energy constraints. Various innovative methods have been specifically developed for the control of temperatures in buildings. While most studies design the control method independent of building occupants, [33] incorporates active user feedback to maintain the temperatures during the occupancy hours. [58] considers temperature forecasts and historical data in the design of the control technique. This is done by a feedforward scheme based on iterative learning control to extract information from historical data with similar temperature patterns and preemptively account for expected future error. In this part of the dissertation, we employ the Model-free control (MFC) method to maintain indoor air temperatures in building HVACs and water temperatures in water heaters in buildings.

Each of these building loads – HVACs or water heaters – is treated as an agent, and together they are to consume a certain amount of power. To achieve such coordination, we consider an architecture where all the agents act as local controllers, and their combined power consumption is coordinated by a central controller. This central controller communicates back and forth with all the agents, however, the agents do not communicate among themselves. Such framework is currently employed in transactive energy markets – one of the fastest growing areas of research and implementation in the power sector.

As many parts of the world are gradually moving towards competitive transactive energy markets as a means to generate and procure electricity alongside many of the support services required to operate a power system, many countries are pushing the reform of the electricity power sector very positively. For example, Chile pioneered in the 1980s the deregulation of the electric power industry. In today's U.S. retail electricity market, fourteen states have already adequate retail competition with Texas, Illinois, and Ohio respectively having 100%, 60%, and 50% of their residential customers receiving service from electricity suppliers [17]. But, even today, most of the customers have very limited "direct" participation in supporting the grid.

Through the development in the transactive energy market, there have been some interesting and innovative proposals. [81] proposes a data-driven methodology to forecast electricity demand for decentralized energy management. On the electricity consumption end, peer-to-peer (P2P) electricity trading is gaining momentum with time. Analogous to internet servers and clients, P2P electricity trading is the platform where the end consumer becomes a prosumer (functioning as both energy producer and consumer) and exchanges the remaining electricity with other consumers in the power grid [61]. A detailed review of existing P2P trading projects is carried out in [84]. Another major proposal in this direction is load aggregation.

The aforementioned central controller is a load aggregator in transactive energy market. As defined in [52] "An aggregator is a new type of energy service provider which can increase or moderate the electricity consumption of a group of consumers according to the total electricity demand on the grid. An aggregator can also operate on behalf of a group of consumers producing their own electricity by selling the excess electricity they produce." A detailed review of the value of aggregators in the electricity market can be found in [10].

It is worth noting that the two proposals are not mutually exclusive; aggregators can act as participants in the P2P network. Aggregation and optimization of massive portfolios of behind-the-meter assets are likely to grow as a business opportunity because individual prosumers will have limited capabilities and/or financial incentives to deal with capturing the modest value streams of their own microgrids, creating a huge opportunity for the aggregators, [68].

While such a centralized framework is beneficial in certain applications, like transactive energy markets, the cost and the risk of the associated communication overhead can be too high to bear for other applications – for example, a set of mobile robots deployed in a search and rescue mission. In such a mission, one can have the problem such that the only objective of all the robots involved is common and is to search for a certain entity. The advantages of having communication capabilities between the robots is clear in such time-sensitive situations. One then needs to develop a framework to model the flow of information among the robots and design control laws such that the common objective is efficiently achieved. Team decision theory is one such field.

Team decision theory is a mathematical formalism for a stochastic decision problem in which a team, consisting of two or more team members, cooperate to achieve a common control objective. In the framework of team decision theory, there is only a global objective common to all the team members; there is no individual local objective.

The field originated in 1950's with papers by Marschak and Radner, [53] and [65], aimed to solve the objectives of teams within an organization. In the former, Marschak introduced various terminologies and elements relating to an organization team, while Radner, in the latter, mathematically formulates the team decision problem. Then, Ho in [35] in 1970's explores the role of information structures by considering a dynamic team with partially-nested information structure. Later the same decade, Ho brings together in [36] three seemingly unrelated topics – team decision theory, market signaling in economics, and the classical Shannon information theory. In [43], the authors extend Radner's theorem to obtain sufficient conditions to establish global optimality of certain candidate control laws for a certain static team problem with convex cost. In the recent decades, [51] provides a characterization of information structures and their impact on the tractability of team optimization. [77] provides a compilation of all the main results in the field of team decision theory.

The Witsenhausen's counterexample provides an anticipatory explanation for the lack of accelerated developments in the field. While various aforementioned papers explore the complex roles information structures play in the team problem, [37] was the first one to do so. It introduced a simple two player team problem with a non-classical information structure – both the players do not have access to the same information – and showed that the optimal laws are in general not linear for certain parameter values. Even though the problem is similar to a linear quadratic gaussian (LQG) problem, Witsenhausen showed that the results of a centralized LQG problem do not apply to the provided counterexample due to its information structure.

The framework offered in team decision theory is wide and encompasses almost every architecture in a team problem. However, the difficulties arising from the limitations regarding information structure within a team currently raise obstacles in its applicability in real-world situations. For example, solving the power allocation problem considered in the beginning of this section, might prove extremely cumbersome in a team decision framework. Therefore, as a next step in this work, we explore another approach to solving such multi-agent problems – through Voronoi tessellations (VTs).

Voronoi diagram is a partition of a set into subsets containing elements that are close to each other according to a certain metric. Voronoi partition of, say a plane, a two-dimensional region would be a collection of disjoint regions that contain points that are closer to the region they are in than any other region. The collections of such regions is called a Voronoi tessellation. Dirichlet was the first to systematically treat Voronoi tessellations, which was generalized half a century later by Voronoi, [40]. Hence, Voronoi tessellations are also called Dirichlet cells.

Even though they date centuries, Voronoi tessellations have been found to be immensely helpful in various applications ranging from health to computer graphics to natural sciences. The first documented application of Voronoi tessellations appeared in [69] on the 1854 cholera epidemic in London in which it is demonstrated that proximity to a particular well was strongly correlated to deaths due to the disease [40]. In more recent decades, VTs have almost become a common basis tool for path planning algorithms by multi-robot systems in the field of coverage control [19] to such an extent that the VT-based coverage control has been generalized using optimal transport-based control [38]. An adaptive coverage controller is proposed in [5] where the leader in the leader-follower strategy therein distributes the followers within its obstacle-free sensing range, and the optimized distribution is obtained through Centroidal Voronoi Tessellation (CVT). In their study on optimality of multi-robot coverage control, the authors in [20] draw a relationship between CVT configurations and the sufficient condition for optimality through the spatial derivative of the density.

In line with the popularity of CVTs, remarkable amount of contributions have been made to further their development. [25] refines the notion of Constrained CVTs and derives various properties like their characterization as energy minimizers. Focusing on 1-D Voronoi diagrams, [3] develops an optimal algorithm for computing collinear weighted Voronoi diagrams that is conceptually simple and attractive for practical implementations. [29] studies the inverse Voronoi problem in-depth.

Despite the wide applicability and vast development in the literature pertaining to CVTs, there are challenges and open questions, especially in high dimensional spaces. For dimensions greater than one, rigorously verifying that a given CVT is a local minimum can prove difficult, for example [76] uses variational techniques to give a full characterization of the second variation of a CVT and provides sufficient conditions for a CVT to be a local minimum. Moreover, in high dimensional spaces, the number of CVTs under certain conditions and their quality is elusive, and their computation remains difficult.

The regions in the Voronoi tessellations have a “center”. The definition of the center depends on the application at hand – it can be a mass centroid based on certain underlying probability distribution over the region. In any case, the centers of the Voronoi regions are usually the element of interest or a function of it. For example, in a coverage control application, the center is the position of the agent, and accordingly we can use the framework of Voronoi tessellations to model the team dynamics and the information structure in the team. In this work, we employ Voronoi tessellations to solve a resource allocation problem within a team which is formulated as:

$$\begin{aligned} &\min_{z_i \in \mathbb{R}^n} \frac{1}{N} \sum_{i \in I_N} f_i(z_i) \\ \text{such that,} \quad &\sum_{i \in I_N} z_i = B \end{aligned} \tag{1.1}$$

In the resource allocation problem (1.1), B is the amount of resource to be allocated among N agents while minimizing the sum of their individual costs $\{f_i\}_{i \in I_N}$. Seemingly trivial and complex simultaneously, the nature of (1.1) can be broken down into the information structure in the group of agents, the separability of the objective function, and the global constraint. While each agent can minimize the cost function without any dependance on other agents, the global constraint is imposed on the team, hence making the team information structure a significant aspect.

Allowing a dynamic information structure with certain mild conditions, the authors in [44] propose gradient and sub-gradient based algorithms to solve (1.1) when the individual cost functions are differentiable and otherwise, respectively. While [44] proposes the gradient descent algorithm where the agents trade resources in proportion to the gradient difference for their respective utility functions, [83] takes up the allocation problem (5.1) to focus on choosing the proportional weights (to the resource trading) to sufficient conditions for the convergence of the algorithm, and to further improve the rate of convergence. [7] considers the dual of the resource allocation problem and derives two methods using the alternating direction method of multipliers (ADMM) algorithm. Also considering the dual problem, [21] includes uncertainties in the individual cost functions and solves the problem using sub-gradient methods on the distributed Lagrangian. Including economics in the team, [12] considers a stochastic system in which agents allocate shared system resources in response to customer requests that arrive stochastically over time and introduces the notion of a transfer contract to specify compensation among agents when resources are traded. Each agent has a model of how resources are shared by others and makes allocation decisions by maximizing its utility function subject to such model. However, it is worth noting that in most of the work on decentralized resource allocation, the amount of resource to be allocated is fixed over the iterations; the agents move along the feasible solutions to only minimize the cost function.

1.2.1 Contributions

Various aspects of the problem of Demand-Response in smart grids are studied under different conditions in a centralized framework. Recognizing that most of the significant power consuming loads in a building operate under ON-OFF inputs, for example Heating, Ventilation and Air Conditioning (HVAC) or water heater, we analyze the stability of Model-free control (MFC), a control method introduced in [28] that we employ for (local) control of building loads, under various constraints on the local input.

Building on the MFC as a local controller, we develop algorithms for centralized power allocation for homogeneous and heterogeneous loads. For the latter, we develop an allocation method using weighted projections where different loads can be assigned different weights reflecting the preferences of the building occupant, while simultaneously maintaining the power allocation constraint.

Moving away from centralization to team decision theory, we consider the celebrated Witsenhausen counterexample. Although it was introduced as a toy example, its optimal non-linear laws had remained elusive. In this dissertation, we develop a numerical method to implement the optimal laws that were analytically derived in the past decade but had remained unimplemented. We show that our laws coincide with the optimal affine laws in the parameter set where the optimal laws are known to be affine. For the region where the optimal laws are non-linear, we show that our laws result in the lowest costs when compared with the previously reported costs in the literature from heuristic solutions.

In the investigation of employing CVTs to solve the resource allocation problem, we encounter a lack of computational methods that obtain the entire CVT in high-dimensional spaces, and not just the centroids. To that end, we develop a new decomposition method to obtain some of the many non-unique CVTs in a high-dimensional space under the same condition. The underlying idea of the developed method is to construct a high-dimensional CVT from a series of decomposed 1-D CVTs. Such construction results in grid-like tessellations, and are obtained in a simple and efficient manner.

In addition, in the investigation we notice an advantage brought forth to the team information structure by CVTs in one-dimensional spaces. Because the underlying space is 1-D, a simple communication network that is merely a line graph is sufficient to obtain CVTs in a decentralized manner. This further allows the method to be scalable to any number of agents in the team since each member can have at most two neighbors.

Therefore, we exploit such properties of CVTs to solve the resource allocation problem. While most of the solutions in the literature assume a fixed amount of resource to be allocated, we exploit the properties of CVTs to solve a dynamic (time-varying) resource allocation problem in a truly decentralized manner. Along with being a CVT that satisfies the allocation constraint, we allow for flexibility in agents' embedding their local preferences through what we call a *"civility model"*. We demonstrate the applicability of the developed decentralized dynamic resource allocation method on the problem of power allocation in a group of building HVACs.

1.3 Outline of the Dissertation

The demand-response problem in power allocation in transactive energy market is taken up in Chapter 2. The problem is first described in detail graphically and then formulated mathematically. The local control method – MFC – is reviewed in Section 2.1.1, and then its stability under certain input constraints is studied in Section 2.1.2. We then move on to the central controller, and develop two algorithms for power allocations for equally and differently weighted loads in Sections 2.2 and 2.3, respectively.

In Chapter 3, we consider the Witsenhausen counterexample. Specifically, in Section 3.1, we explain the problem in-depth and note the previously-derived analytically optimal solutions. In Section 3.2, we develop a method to numerically implement the optimal control laws. Finally in Section 3.3, we provide simulation results for various parameter sets in the problem, and compare the results we obtain with those previously reported in the literature.

In Chapter 4, we study and analyze CVTs beginning with some definitions and uniqueness properties in Section 4.1, followed by computational methods to obtain one-dimensional CVTs in Section 4.2. Moving to CVTs in high-dimensional spaces, we develop a decomposition method in Section 4.3 and compare it with existing methods, both numerically and qualitatively.

In Chapter 5 we explore the role of CVTs in decentralized resource allocation, starting with the problem statement and complete introduction to the resource allocation problem. We present our motivation for using CVTs to solve the resource allocation problem in Section 5.1 including the sufficiency of a simple information structure in the team to reach a minima in Section 5.1.1. We consider the resource allocation problem with fixed amount of resources – static allocation – and solve it by embedding the resource allocation constraint within the distribution of the CVT in Section 5.2. Using the static allocation solution as the initialization step, we solve the dynamic resource allocation problem in Section 5.3. We apply the developed method in the field of demand-response to solve the problem of power allocation among a group of building HVACs.

Finally, in Chapter 6, we draw some conclusions and provide some explicit lines of future work.

Chapter 2

Centralized Power Allocation in Heterogeneous Building Loads

In a step to move towards a greener planet, renewable sources of energy must be integrated in the day-to-day consumption. However, the uncertainty associated with such sources has been a drag on the progress. One direction to reduce the negative effects of such uncertainty is to use the energy generated from renewable sources in applications that have flexible loads. Buildings consume 40% of the electricity produced in the United States, and have flexible loads like HVAC (heating, ventilation, and air conditioning) units and water heaters.

In this chapter, we consider a solar farm as a source of energy generation, and a group of buildings in the neighborhood of the farm. The problem we are interested in is employing all the energy generated in the farm to meet the energy needs of the buildings in real-time, that is, the generated energy is not stored, and ideally, not wasted either.

Let there be N building loads – we consider HVACs and water heaters – that need to be controlled to maintain certain desired setpoints – indoor air temperature and water temperature in HVAC and water heater, respectively. Denote the generated power from the solar farm at time instant k as $P(k)$. Given the nature of the problem at hand, we also call the generated power as the available power. Let the power consumed by the i^{th} building at time instant k be denoted $P_i(k)$. The power allocation problem, for a tolerance $\epsilon > 0$, is mathematically stated as:

$$P(k) - \epsilon \leq \sum_{i=1}^{N} P_i(k) \leq P(k) + \epsilon \tag{2.1}$$

The power allocation is carried out by a central controller that has access to the states of all the loads. Based on the states of the loads and the available power, the central controller allots each $P_i(k)$. Meanwhile, on the building load side, the local controller responsible for maintaining the load at the desired setpoint does so while accounting for the allotted power. Each local controller tracks the load output against the desired setpoint. Based on the load output and the desired setpoint, it computes an appropriate or "ideal" input to the load. This control input translates into usage of a certain amount of power. The local controller communicates this control input to the central controller, which then based on the information from all the other loads and the available power at that time instant, communicates back to the local controller the input it can supply to the load. This interaction between the central and the local controllers is summarized in the block diagram shown in Fig. 2.1.

In Section 2.1, we will look into the local controller, while in Sections 2.2 and 2.3, we will take up two algorithms by the central controller to allocate the generated power.

2.1 Model-free Control

The local controller monitors the load output, and based on the desired setpoint, computes an appropriate control input. We employ the method of Model-free control (MFC) [28] as our local controller. First, we introduce the control method in Section 2.1.1. As previously discussed, the "ideal" control input need not be the one supplied to the load. The power allocation constraint translates onto the local controller as

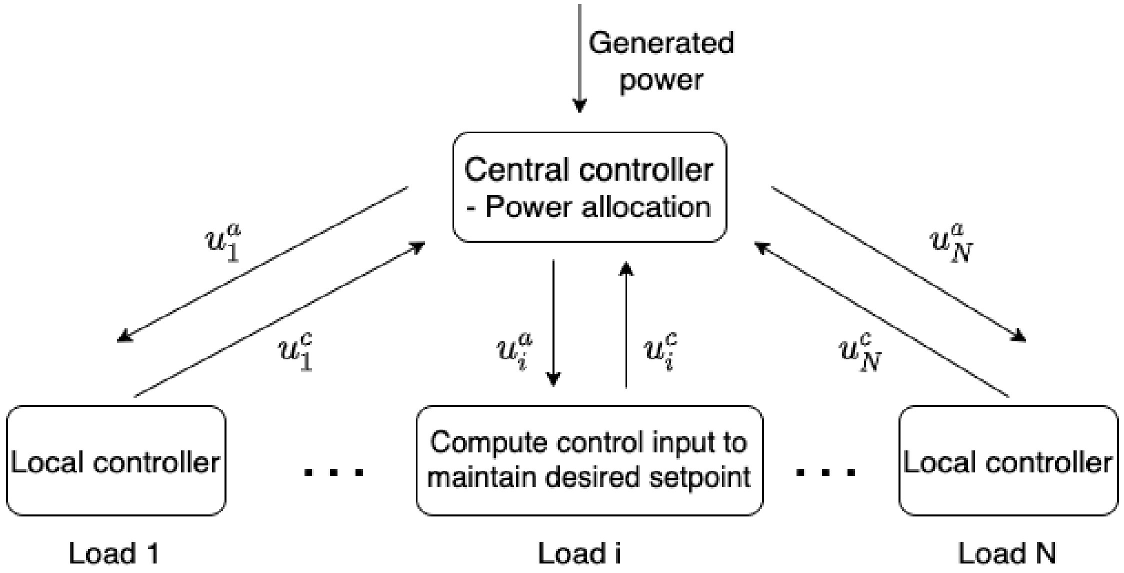


Figure 2.1: Block diagram illustrating the overall problem framework and the flow of information between central and local controllers.

constraints on the input to the load. Accordingly, we study the stability of the local controller – MFC – under different input constraints in Section 2.1.2.

2.1.1 Overview

In this Section, we will review MFC introduced in [28] for a general SISO (single-input single-output) system operating under unconstrained input and look into the associated stability conditions. The SISO system is approximated by an ultra-local model as:

$$\dot{y} = F + \alpha u \tag{2.2}$$

Here, u and y are the input and output of the system, and F describes the poorly known or unknown parts of the system. The parameter α is to correct for the difference in the magnitudes of the input and the output. F is approximated by a piecewise constant function ϕ that is given as [28]:

$$\phi = \frac{-6}{L^3} \int_{t-L}^{t} [(L - 2\sigma)y(\sigma) + \alpha\sigma(L - \sigma)u(\sigma)]d\sigma \tag{2.3}$$

Note that ϕ is estimated using the measurements of the system obtained in the last L seconds and accordingly F is continuously updated. Using the latest F, the intelligent-proportional control law is given by [28]:

$$u = -\frac{F - \dot{y}^* + K_p(y - y^*)}{\alpha} \tag{2.4}$$

Here, y^* is the desired reference trajectory and K_p is the proportional gain. Combining (2.2) and (2.4) provides the error dynamics [28]:

$$\dot{e} + K_p e = 0 \tag{2.5}$$

where $e = y - y^*$ is the tracking error. With t_0 as the initial time, the solution of this differential equation is:

$$e(t) = e(t_0)\exp(-K_p(t - t_0)) \tag{2.6}$$

Without loss of generality, let $t_0 = 0$ for all of the following. Equation (2.6) shows that the error asymptotically decays to 0 for $K_p > 0$, making the tuning of the proportional gain straightforward. The only other tuning parameters that need to be manually set are α and L [28].

Equation (2.6) shows that, for appropriate value of the proportional gain, the system is asymptotically stable for unconstrained control input. However, in reality, for most systems the control input is saturated. Moreover, in many applications external constraints need to be imposed on the control input. For example, in conventional building HVAC systems, the HVAC unit can only be operated in different stages, i.e., the input power to the HVAC, u, can only take a set of distinct values (for example $u \in \{0, 1, 2, \ldots\}$) or it could only be switched on or off i.e., $u \in \{0, 1\}$. In the next section, we will mathematically pose these constraints and evaluate the stability of the MFC design under each of these constraints.

2.1.2 Stability Analysis under Constrained Inputs

In this section, we first consider the constraint where the control input can only take discrete values (as in multi-stage HVAC unit where $u \in \{0, 1, 2, \ldots\}$), and then we consider the constraint where the control input can only be either on or off. The second constraint inherently includes simultaneous saturation and discretization of the control input. The stability analysis and derivations of conditions are published in [74].

Let u be the value of the control input obtained from the model-free control law (2.4) and u_r be the constrained input that is ultimately supplied to the plant. Accordingly, the model of the plant will be transformed as:

$$\dot{y} = F_r + \alpha u_r \tag{2.7}$$

Since it is the constrained input u_r that is ultimately delivered to the plant, F_r is approximated by ϕ_r that will be computed using u_r as:

$$\phi_r = \frac{-6}{L^3} \int_{t-L}^{t} [(L - 2\sigma)y(\sigma) + \alpha\sigma(L - \sigma)u_r(\sigma)]d\sigma \tag{2.8}$$

Using the latest F_r, u will be computed as:

$$u = -\frac{F_r - \dot{y}^* + K_p(y - y^*)}{\alpha} \tag{2.9}$$

Finally, u_r is obtained by constraining u, and the definition of u_r will be specific to the constraint formulation as in the following sections.

Constraint of discrete input

The constraint of discrete control input can be formulated as round-off value of the original control input u, i.e., $u_r = \text{round}(u)$, which is given by:

$$u_r = \lfloor u + \frac{1}{2} \rfloor \tag{2.10}$$

where $\lfloor . \rfloor$ is the floor function defined as:

$$\lfloor u \rfloor = \max\{m \in \mathbb{Z} : m \leq u\} \tag{2.11}$$

and $\mathbb{Z}$ is the set of integers. The following relation holds true between u and u_r:

$$u - \frac{1}{2} \leq u_r \leq u + \frac{1}{2} \tag{2.12}$$

Substituting for u from (2.9) and for u_r from (2.7), we obtain:

$$-\frac{F_r - \dot{y}^* + K_p e}{\alpha} - \frac{1}{2} \leq \frac{1}{\alpha}(\dot{y} - F_r) \leq -\frac{F_r - \dot{y}^* + K_p e}{\alpha} + \frac{1}{2}$$

Consider the left-hand side:

$$\begin{aligned} & -\frac{F_r - \dot{y}^* + K_p e}{\alpha} - \frac{1}{2} \leq \frac{1}{\alpha}(\dot{y} - F_r) \\ \implies & -\frac{\alpha}{2} \leq \dot{e} + K_p e \end{aligned}$$

Similarly, the right-hand side:

$$\begin{aligned} & \frac{1}{\alpha}(\dot{y} - F_r) \leq -\frac{F_r - \dot{y}^* + K_p e}{\alpha} + \frac{1}{2} \\ \implies & \dot{e} + K_p e \leq \frac{\alpha}{2} \end{aligned}$$

Therefore, $\forall u \in \mathbb{R}$, we obtain the following error dynamics when the control input is discretized:

$$-\frac{\alpha}{2} \leq \dot{e} + K_p e \leq \frac{\alpha}{2} \tag{2.13}$$

In order to further investigate the stability conditions when the control input is constrained, we need to solve the associated differential inequality (2.13). $C(.)$ is the space of continuous functions and $C^1(.)$ is the space of continuously differentiable functions. We employ the result of Lemma 1 from [63], which is restated here:

Lemma 2.1. *Let $x_0, y_0 \in \mathbb{R},\ I = [x_0, \infty), a, b \in C(I), y \in C^1(I)$ and*

$$\dot{y} \leq a(x)y(x) + b(x), \forall x \geq x_0, y(x_0) = y_0 \tag{2.14}$$

then $\forall x \geq x_0$ the following holds:

$$y(x) \ \leq y_0 \ \exp\big(\int_{x_0}^{x} a(t)dt\big) + \int_{x_0}^{x} b(s)\exp\big(\int_{s}^{x} a(t)dt\big)ds \tag{2.15}$$

If the converse holds in (2.14), then the converse holds in (2.15) too.

Applying the result of Lemma 2.1 to the right-hand side of (2.13), we obtain,

$$e(t) \leq e(0)\exp\big(\int_0^t -K_p d\tau\big) + \int_0^t 0.5\alpha \exp\big(\int_s^t -K_p d\tau\big)ds$$

$$\implies e(t) \leq e(0)\exp(-K_p t) + \frac{0.5\alpha}{K_p}\big(1-\exp(-K_p t)\big) \tag{2.16}$$

Similarly, applying the result of Lemma 2.1 to the left-hand side of (2.13), we obtain,

$$e(t) > e(0)\exp\big(\int_0^t -K_p d\tau\big) - \int_0^t 0.5\alpha \exp\big(\int_s^t -K_p d\tau\big)ds$$

$$\implies e(t) > e(0)\exp(-K_p t) - \frac{0.5\alpha}{K_p}\big(1-\exp(-K_p t)\big) \tag{2.17}$$

Combining (2.16) and (2.17), we obtain the solution of the differential inequality of the error dynamics in (2.13) when the control input is constrained:

$$\begin{aligned} e(0)\exp(-K_p t) - \frac{0.5\alpha}{K_p}\big(1 - \exp(-K_p t)\big) &< e(t) \\ \leq e(0)\exp(-K_p t) + \frac{0.5\alpha}{K_p}&\big(1 - \exp(-K_p t)\big) \end{aligned} \tag{2.18}$$

By comparing (2.18) and (2.6), it can be observed that when the control input is constrained, the error stays within $\pm\frac{0.5\alpha}{K_p}\big(1 - \exp(-K_p t)\big)$ of (2.6). Furthermore, as $t \to \infty$, the error associated with the constrained input will be within $\pm\frac{0.5\alpha}{K_p}$ of the error obtained when the input is not constrained.

Since the error in the case of discretization-constrained control input is within a finite band of the asymptotically decaying error of (2.6), it is concluded that the MFC design leads to bounded output (hence stable controlled system) even when the control input is constrained as in (2.10).

Constraint of on-off input

Now we consider the constraint where the control input can either be on or off, as in the case of a single-stage HVAC unit. Without loss of generality, this constraint can be formulated as:

$$u_r = \begin{cases} 0, & \text{if } u < M \\ 1, & \text{if } u \geq M \end{cases} \tag{2.19}$$

where $0 < M < 1$. Equation (2.19) simultaneously imposes discretization and saturation constraints on the control input. Let us now continue the evaluation on a case-by-case basis.

Case 1: $u < M$

From (2.19) and (2.7), we have $u_r = 0$, $\dot{y} = F_r$ and because $u < M$,

$$\begin{aligned}
&\implies -\frac{1}{\alpha}[F_r - \dot{y}^* + K_p e] < M \\
&\implies \dot{e} + K_p e > -\alpha M \qquad (2.20)
\end{aligned}$$

Case 2: $u \geq M$

Similarly, from (2.19) and (2.7), we have $u_r = 1$, $\dot{y} = F_r + \alpha$ and because $u \geq M$,

$$\begin{aligned}
&\implies -\frac{1}{\alpha}[F_r - \dot{y}^* + K_p e] \geq M \\
&\implies \dot{e} + K_p e \leq \alpha(1 - M) \qquad (2.21)
\end{aligned}$$

Combining (2.20) and (2.21), we obtain the following error dynamics:

$$-\alpha M < \dot{e} + K_p e \leq \alpha(1 - M) \qquad (2.22)$$

It can be seen that the error dynamics in (2.22) are in the same form as the error dynamics associated with the discretization constraint, (2.13). Hence, as in Section 2.1.2, it is concluded that the MFC design leads to a stable system when there is a simultaneous imposition of saturation and discretization constraints.

Simulation on an example HVAC model

The mathematical model for the building HVAC system employed for simulation is given by the state space formulation [13], [54] :

$$\begin{aligned}
\dot{x} &= Ax + Bu + Gw \\
y &= Cx + Du
\end{aligned} \qquad (2.23)$$

where $u \in \mathbb{R}$ is the input to the HVAC system, $w \in \mathbb{R}^2$ is the disturbance, and $y \in \mathbb{R}$ is the measured indoor temperature. u is the only controllable input to the system.

The disturbances considered in w are the external temperature (denoted u_e) and solar radiation (denoted u_s). These are the measurements obtained in the summer of 2017 in Knoxville, TN, USA. The profile of disturbances, a subset of which is employed in the subsequent simulations, is shown in Fig. 2.2.

The system matrices in (2.23) are parameterized as:

$$\begin{aligned}
&A = -1/(RC_1),\ B = -Q_{hvac}/C_1\\
&G = [1/C_1 \ \ 1/(RC_1)]\\
&C = 1,\ D = 0
\end{aligned}$$

where the parameters are:

$$\begin{aligned}
&R = 1/200,\ C_1 = 20a_d v_a c_p\\
&Q_{hvac} = 3 \times 3504 \times 1.5
\end{aligned}$$

Here, $a_d = 1.225kg/m^3$ is the air density, $v_a = 550m^3$ is the volume of the air, and $c_p = 1033J/kgC$ is the specific heat of the air. The continuous time model (2.23) is used in simulations as a discrete-time model with 10 minutes discretization to match the slow nature of the bulky HVAC systems.

MFC is designed for the building HVAC system for three different scenarios: 1) There is no constraint on the control input. 2) The control input is rounded-off (as in Section 2.1.2). 3) The control input is constrained to on or off (as in Section 2.1.2).

In the simulations, we use $\alpha = 1$, $L = 3600$ and $K_p = 10$. The errors in the indoor temperatures obtained in all three scenarios are shown in Fig. 2.3 and the corresponding control inputs are shown in Fig. 2.4.

There are some noteworthy points in the results. In the first scenario, the error is expected to be asymptotically decaying as governed by (2.6). However, we observe an

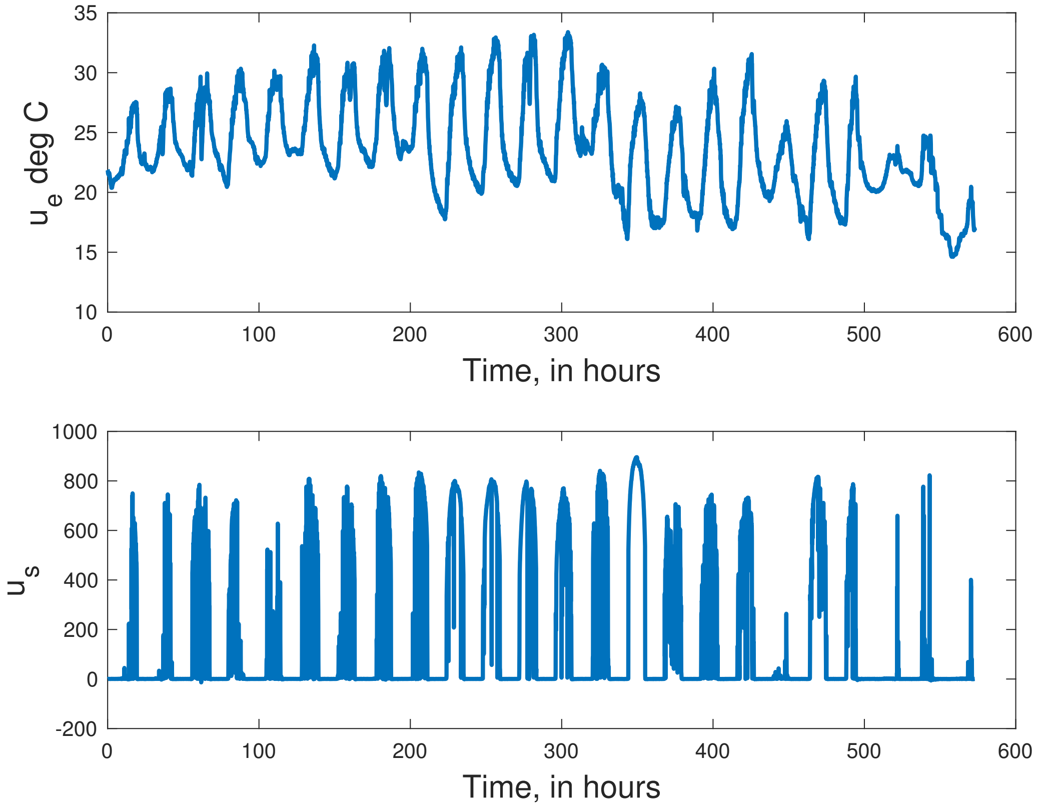


Figure 2.2: Measured disturbances, top: external temperature, bottom: solar radiation., taken in a typical summer in July 2017 in Knoxville, Tennesee.

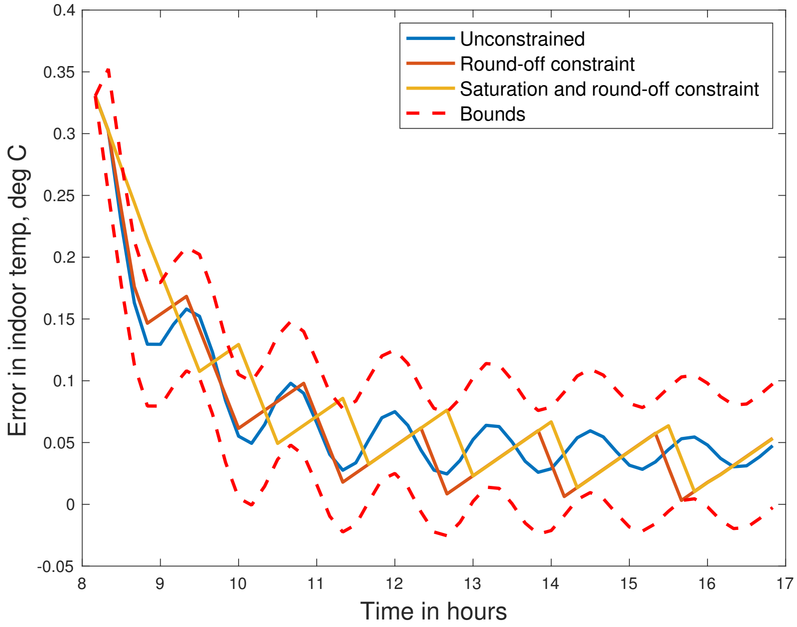


Figure 2.3: Comparison of errors obtained under different constraints.

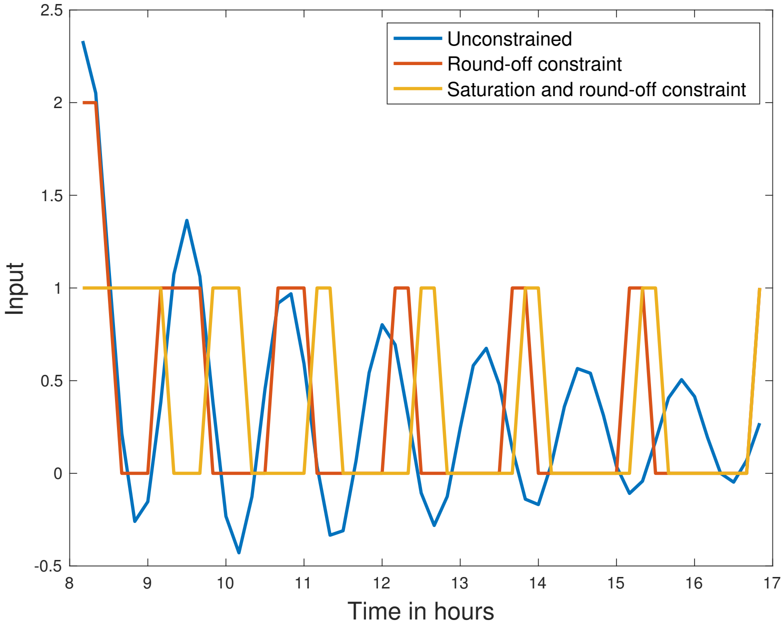


Figure 2.4: Comparison of control inputs obtained under different constraints.

oscillatory response. The amplitude and frequency of the oscillations is dependent on the value of the proportional gain K_p. The reason for the observed oscillations can be attributed to the estimation error, i.e., $\phi_r \neq F_r$.

Let the estimation error be bounded, i.e., $|\phi_r - F_r| \leq \epsilon$, for some $\epsilon > 0$. As in Section 2.1.2, using (2.7) and (2.9), the associated error dynamics can be derived to be:

$$-\epsilon \leq \dot{e} + K_p e \leq \epsilon \tag{2.24}$$

Equation (2.24) conveys that, practically, the error settles within an ϵ band of 0 instead of asymptotically decaying to 0. As the estimation error is not constant over time, the corresponding error margin also varies. Moreover, the estimate (along with the proportional gain) influences the control input (given in (2.4)) and the control input in turn influences the estimate with a lag. Therefore, for the aforementioned reasons, even though we expect asymptotically decaying behavior in the error for any $K_p > 0$, the observed behavior is not compliant with (2.6).

While the error in the first scenario deviates from the expected behavior by oscillations, we observe that the errors in the second and the third scenarios are almost as expected. From the comparison of (2.6) and (2.18), we expect the error in the second and the third scenarios to be within a certain band of the error in the first scenario. The upper and lower bounds of the band are shown in dotted red line in Fig. 2.3. Except for the slight excursion out of the band by the error in the third scenario, we observe that the constrained error stays within a known band of the unconstrained error. The control inputs corresponding to the cases in Fig. 2.3 are shown in Fig. 2.4.

In this Section, we employed MFC as the local controller and studied its stability under various constraints on the control input. In the next two sections, we will take up the central controller that manages the available power among all the loads based on their states.

2.2 Power Allocation under Equal Importance

In reference to Fig. 2.1 from the beginning of this chapter, the N loads considered in this section are equally weighted. That is, no one load has any more priority than any of the other loads. Further dividing the section, first we consider N identical and then nonidentical HVAC units. In the last part of this section, we formulate a relationship between the generated PV power and the total number of buildings required to consume all of the generated power. The method introduced in here, along with the results are published in [72].

Identical and otherwise, in this Section, we consider single-stage HVAC units; they can only be turned ON or OFF. This translates into the following constraint on the HVAC inputs:

$$u = \{0, 1\} \tag{2.25}$$

where, 0 and 1 correspond to the HVAC system turned off and on, respectively. If a unit is turned ON, it consumes the rated power, p_r. The power consumed by the i^{th} load at time k is hence $P_i(k) = p_r u_i(k)$, where $u_i(k)$ is the input to the i^{th} HVAC at time k.

Since our focus includes fully employing the generated PV energy, we limit our analysis to during the day when there is sufficient solar energy. Accordingly, when the HVAC system is turned off, the indoor temperature of the building increases, and when the HVAC system is turned on the building temperature decreases. More specifically, we consider summer time here. That is, the HVAC system considered operates only in cooling mode.

The algorithm to achieve the power allocation objective by the central controller is developed as follows. Employing the MFC method, each local controller computes the “ideal” control input to its load using the previous L seconds of input and output measurements of the load:

$$u_i(k+1) = \mathrm{MFC}(u_i((k-n):k), y_k((k-n):k))$$

Here, n is the previous L seconds of measurements, with $n = t_{samp}L$, where t_{samp} is the sampling time. The function MFC(.) is implemented as reviewed in Section 2.1. All the local controllers communicate their $u_i(k+1)$ to the central controller. Translating each received control input to the power required, the central controller checks if the power allocation constraint (2.1) is satisfied or violated. If it is satisfied, then all the control inputs are rounded, to satisfy the HVAC on-off constraint, and communicated back to the local controllers, which then supply it to their HVAC loads. On the other hand, if the constraint is violated, then the central controller prioritizes the need to cool the buildings by their "ideal" control inputs. For example, if $u_{29} = 0.9$ and $u_{87} = 0.7$, then the 29^{th} building will have a higher priority of cooling than the 87^{th}. Once the priority list is computed, the central controller allocates 1, until the available power is exhausted, to the loads that are at the top of the priority list.

As a further note, consider the case where there are multiple buildings (say N_1 number of buildings) that have the same value of $u_i(k+1)$, and there is not enough generated power to cool them all. In that case, we cool $N_2 < N_1$ number of buildings, and buffer the remaining for the next timestep. Here, N_2 corresponds to the number of buildings that can be cooled using the generated power. When the available power is more than the required power, the aforementioned conflict resolution technique is mirrored and employed.

In our simulations, $\alpha = 1$, $L = 3600$ seconds, $t_{samp} = 600$ seconds, and ϵ is set as 10% of P. The data used for disturbances in (2.23), i.e., external temperature and solar radiation, are the measurements taken on a typical summer day in July 2017, in Knoxville, Tennessee, and are the same as shown in Fig. 2.2.

We consider two cases: all building HVACs are identical, and nonidentical. In both cases, the initial conditions (initial indoor temperatures) of all buildings are randomly assigned.

Analysis for identical buildings

In this section, we consider $N = 100$ identical building HVACs with rated power of 3.5kW, starting at different initial conditions. The MFC method is used in combination with the reviewed methodology to control the indoor temperatures while employing all of the generated power. The results obtained are compared with the standard MPC method developed in [22].

The desired indoor temperature is $23°C$ and the comfort band is defined to be $23°C \pm 1.5°C$. The indoor temperatures obtained in 100 identical buildings with MFC are shown in Fig. 2.5, and the temperatures obtained using Model Predictive Control [14], under the same conditions as in MFC, are shown in Fig. 2.6. The corresponding total power consumed at each timestep for both methods is compared in Fig. 2.7.

It can be observed that while MPC strictly maintains the indoor temperatures within the comfort band, the power consumed is higher than the generated power for a brief period of time. On the other hand, while MFC uses the generated power within permissible limits, the indoor temperatures for a few buildings deviate from the comfort band by less than $0.5°C$ for a brief period of time. This is because the control algorithm implemented with MPC imposes hard constraint on maintaining the indoor temperature, whereas, MFC imposes hard constraint on employing the generated power within permissible limits. If following the power constraint is more important than maintaining the indoor temperature, then the MPC algorithm can be easily reconfigured in a straight-forward manner whereas MFC is not so flexible with changing the constraints. However, MFC is computationally extremely light with its implementation using only a few lines, whereas MPC carries the computational burden.

The results obtained are summarized in Table 2.1 using different tracking error metrics. The root mean square error (RMSE), mean absolute error, and the maximum error in indoor temperature are calculated as:

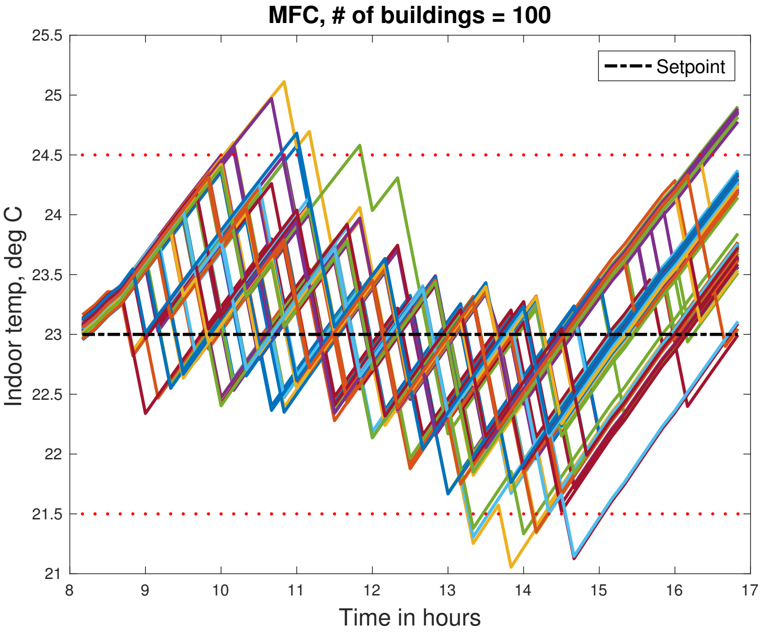


Figure 2.5: Indoor temperatures obtained using MFC for 100 identical buildings

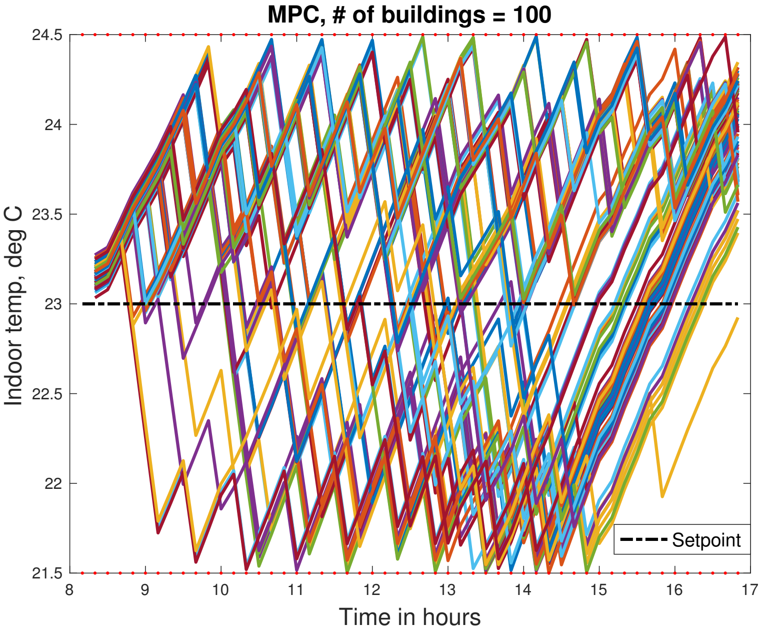


Figure 2.6: Indoor temperatures obtained using MPC for 100 identical buildings

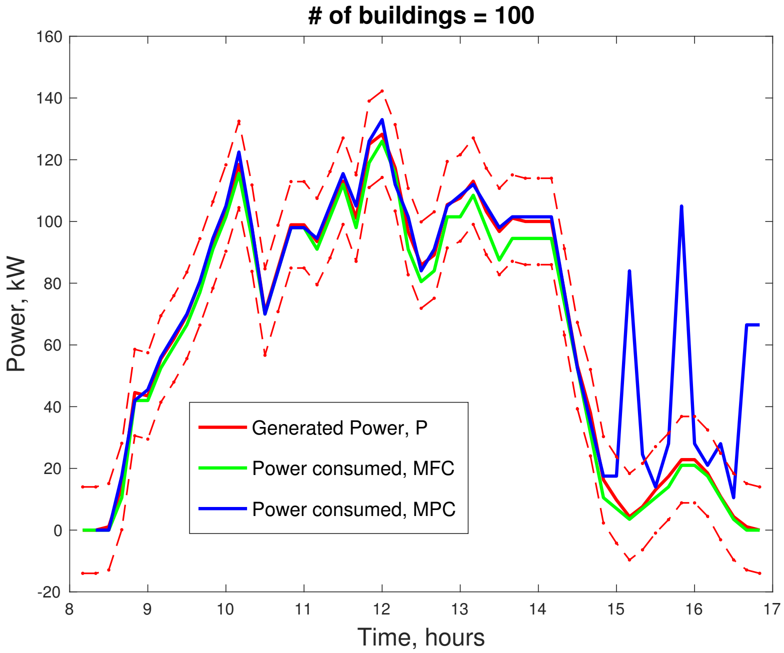


Figure 2.7: Power consumption by 100 identical buildings obtained with MFC and MPC vs generated PV energy. The dotted red lines mark the upper and lower bounds of permissible violation in the power allocation.

Table 2.1: Comparison of MFC and MPC through evaluation of different tracking error metrics

	Indoor temperature		Power consumption	
	MFC	MPC	MFC	MPC
Root mean square error	**0.60**	0.82	**3.64**	20.94
Mean absolute error	**0.49**	0.74	**2.97**	8.22
Maximum error	2.11	**1.50**	**9.24**	82.17

$$RMSE = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_k(i) - y^*)^2}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|y_k(i) - y^*|$$

$$ME = \max\{y_k(i) - y^* : \forall i = 1, 2, \ldots, n\}$$

Here, $y_k(i)$ refers to the measured indoor temperature in k^{th} building at i^{th} timestep, $y^* = 23°C$ is the constant reference signal, T is the total number of timesteps, and $N = 100$ is the total number of buildings. In short, the aggregate error for all buildings is computed as the average of the root mean square error in each building. Similarly, the mean absolute error and maximum error are computed.

Analysis for nonidentical buildings

To generate N nonidentical buildings, we sample from a normal distribution where the state-space parameters from the model (2.23) are considered as the mean, and σ is the standard deviation. Specifically, $\sigma = 0.7 \times 10^{-3}$ for A, $\sigma = 0.8 \times 10^{-3}$ for B, and $\sigma = 5 \times 10^{-3}$ for G. N values, for each state-space parameter, are sampled from the normal distributions and each set is used as a model for the building.

The resulting indoor temperatures and the total power consumption from all the buildings are shown in Fig. 2.8 and Fig. 2.9, respectively.

In both the cases of identical and nonidentical building HVACs, we observe that the power allocation constraint is satisfied, and the indoor air temperatures are not in extreme violation of the comfort band. In this Section, identical or nonidentical HVACs, they all had the same importance. That is, if $u_{29} = 0.9 > u_{87} = 0.7$, then the 29^{th} building will have a higher priority of cooling than the 87^{th}; the latter is not assigned any importance and hence jump the line in the cooling priority.

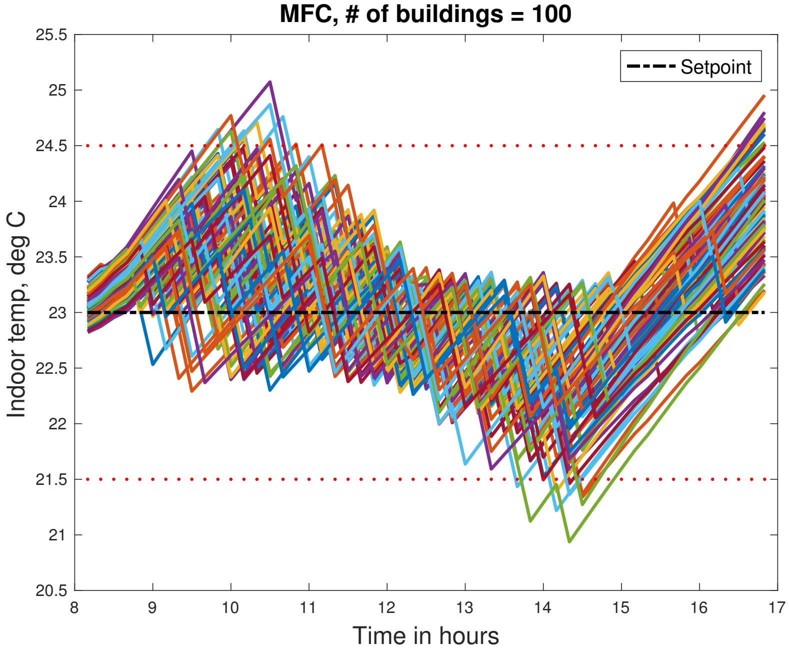


Figure 2.8: Indoor temperatures obtained using MFC for 100 nonidentical buildings

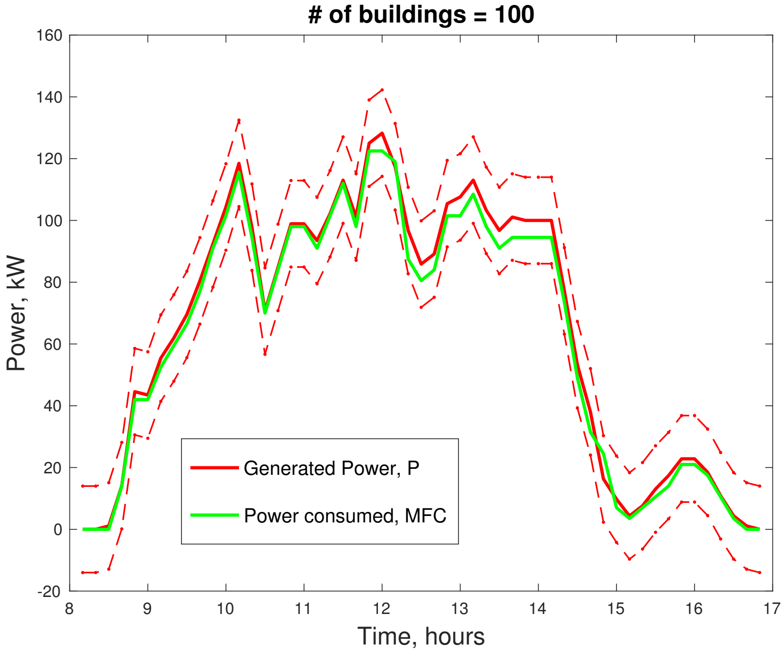


Figure 2.9: Power consumption by 100 nonidentical buildings with MFC vs generated power

In the next Section, we take the case where the loads have the flexibility to be allotted different levels of importance.

2.3 Power allocation using Weighted Projection

In Referring back to the Fig. 2.1 from the beginning of this chapter, N loads are considered therein. Consider the case where one building has multiple loads that are participating in the power allocation. For example, suppose a residential building with participating HVAC and water heater. Inclusion of multiple loads by the same participating building allows one to model the case where the building dwellers can have different preferences. That is, since the power allocation is a hard constraint, the building clients may have strict preferences to maintaining the indoor air temperature and more flexible towards the hot water temperature from the participating water heater. Such preferences can be modeled by assigning different weights (or different importance levels) to different loads. We develop a method to allocate power under such conditions, and the method along with various simulations are published in [70].

Let the number of different types of loads be n, and let N_i be the total number of loads of the i^{th} type, $\forall i = 1, 2, \ldots, n$. Let the total number of participating loads of all types be N. That is, $N = \sum_{i=1}^{n} N_i$. Denote by u_{ij}^{c} the "ideal" control input of the local controller (obtained from the MFC law (2.4)) of the j^{th} load in the i^{th} load type. It is obtained in the same way as the control input for equal importance loads in Section 2.2.

Once the central controller receives the "ideal" control inputs from all the loads, it first does the following sequentially:

1.) For each load type, it computes the priority list of the loads from their control inputs, like in Section 2.2.

2.) Since the loads considered are on-off, it discretizes the control inputs to 0 or 1.

It then computes the "ideal" total power required by all the loads of the i^{th} type as:

$$c_i = p_i \sum_{j=1}^{N_i} u_{ij} \tag{2.26}$$

where p_i is the rated power of the i^{th} load type.

Since there are n different types of loads, $c = (c_1, c_2, \ldots, c_n)$ will be a vector in $\mathbb{R}^n$. The vector c would be the "ideal" amount of power consumed by all the loads, had there been no power-tracking constraint. By "ideal" amount of power it is meant that if c amount of power is allocated, then every unit is maintained at its desired setpoint.

Let x_i be the amount of power consumed at a particular time instant by the i^{th} type load. Since the total power that would be consumed by all types of loads must be equal to the available power at that time instant $P(t)$, the power constraint is an affine hyperplane of dimension $(n-1)$:

$$\sum_{i=1}^{n} x_i = P(t) \tag{2.27}$$

$$\text{such that } \; 0 \leq x_i \leq p_i N_i, \quad \forall i = 1, 2, \ldots, n$$

Since each type of load has a different importance level, this can be modelled by assigning weights to the loads. Let $w_i \in [0, 1]$ be the weight associated with the i^{th} type load such that $\sum_{i=1}^{n} w_i = 1$. Denote the overall weight array by $w = (w_1, w_2, \ldots, w_n)$.

$w_i = 1$ implies $w_j = 0, \forall j \neq i$. This represents the scenario where the i^{th} load is of the highest importance. In such a scenario, it is ideal to allot c_i amount of power to the i^{th} type of load: that is, $x_i = c_i$. Since the power constraint (2.27) needs to be satisfied, one would have:

$$\sum_{j=1 \text{ to } n, j \neq i} x_j + c_i = P(t)$$

Similarly, merging the ideal amount of power for each type of load and the power constraint, a system of n linear equations is obtained, expressed concisely as:

$$\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & 1 & \ddots & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} P(t) - c_1 \\ P(t) - c_2 \\ \vdots \\ P(t) - c_n \end{bmatrix} \tag{2.28}$$

Denote the solution of the system of equations (2.28) as $s = (s_1, s_2, \dots, s_n)$. Then, the point $s^i = (s_1, s_2, \dots, s_{i-1}, c_i, s_{i+1}, \dots, s_n)$ represents the point on the power constraint plane (2.27) that assigns the highest importance on the i^{th} type of load.

Although $0 \leq s^i(i) = c_i \leq p_i N_i$, $s^i(j), \forall j \neq i$ may not satisfy $0 \leq s^i(j) \leq p_j N_j$. That is, the solution of (2.28) might result in consuming negative power. For illustrative purposes, consider $n = 2$. In Figure 2.10, the axes represent the power consumption of the two loads. The red line represents the power constraint plane (line in this case of $n = 2$) given by (2.27). The coordinate (c_1, c_2) represents the ideal power for the two types of loads. The coordinates s^1 and s^2 are obtained from the solution of (2.28), and they correspond to highest importance on the first and second load, respectively. However, it can be seen that $s_2 < 0$ implying negative power consumption by the second load. Therefore, it is needed to ensure that every s^i satisfies the upper and lower bounds on power:

$$0 \leq s^i \leq p.N \tag{2.29}$$

where $p = (p_1, p_2, \dots, p_n)$, $N = (N_1, N_2, \dots, N_n)$ and $p.N$ is their element-wise multiplication: $p.N = (p_1N_1, p_2N_2, \dots, p_nN_n)$. For all $i = \{1, 2, \dots, N\}$, if s^i satisfies (2.29), then it is said to have satisfied the boundary conditions for all the loads.

To ensure that the solution meets the boundary constraint, z^i within the boundaries and closest to s^i is found. This is posed and solved as a constrained optimization problem $\forall i$:

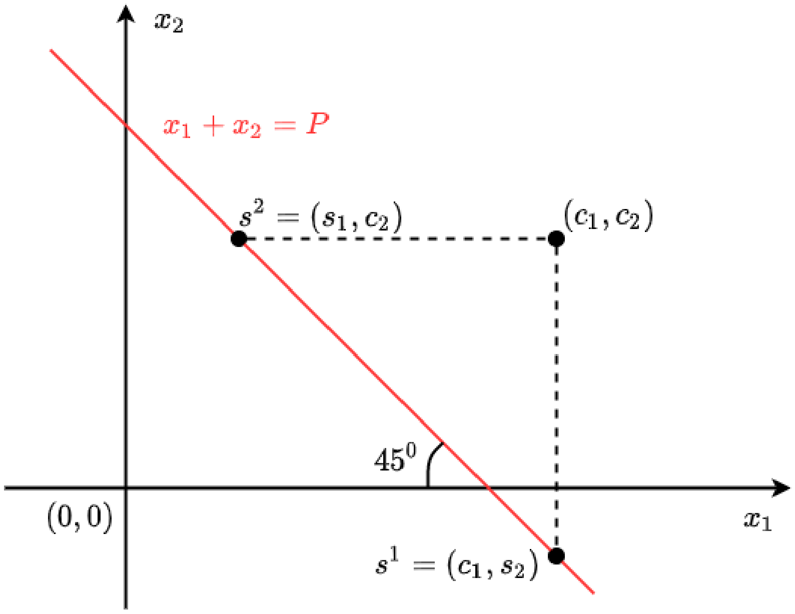


Figure 2.10: Illustration of the two-dimensional power plane.

$$\min_{z^i} ||z^i - s^i||_2^2 \tag{2.30}$$

such that,

$$\sum_{j=1}^{n} z^i(j) = P(t) \text{ and } 0 \leq z^i \leq p.N$$

The optimization problem (2.30) is a constrained linear least square problem that has an explicit solution, see the Appendix section. It can also be solved using the MATLAB command *lsqlin*. The latter is employed for automated implementation.

For all n types of loads, z^i fulfills the following three conditions:

- It is on the power constraint plane.
- It corresponds to the highest importance on the i^{th} type of load.
- It satisfies the boundary conditions for all loads.

For illustration, the example of $n = 2$ from Figure 2.10 is continued. Imposing the boundary conditions on s^1 and s^2, the two nearest coordinates z^1 and z^2 that are within the boundaries are found. In the considered illustration, while s^1 was out of the boundary, s^2 was not. Accordingly, it is seen in Figure 2.11 that s^1 is “moved” to z^1, where z^1 now corresponds to the point of highest importance on the first load, and s^2 is retained. That is, the infeasible power coordinate s^1 is translated along the power constraint line to z^1 satisfying the boundary conditions.

So far, a methodology is formulated that meets the power constraint and deviates from the customers’ comfort as less as possible for all loads in general. However, the customers’ preferences of one load over another are not yet embedded. To do so, with these n z^i points, define a region $M \in \mathbb{R}^n$ as:

$$M = \{z \in \mathbb{R}^n : z = \sum_{i=1}^{n} w_i z^i \text{ and } \sum_{i=1}^{n} w_i = 1\} \tag{2.31}$$

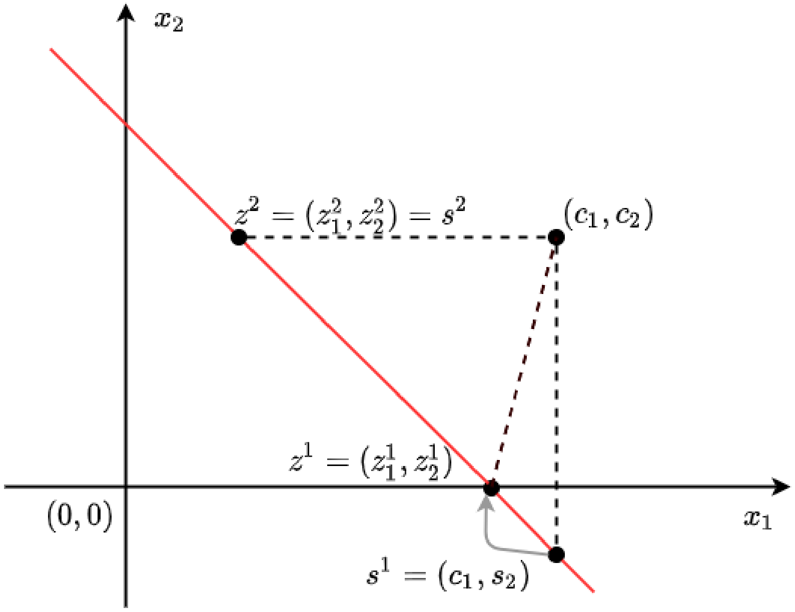


Figure 2.11: Illustration of solution of (2.30) for two types of loads, i.e., $n = 2$.

The polygon M is the set of convex combinations of $\{z^i\}_{i=1}^n$ and M is known as the convex hull of $\{z^i\}_{i=1}^n$, [64]. Therefore, any point in M corresponds to the weighted projection of c onto the power constraint plane, specific to the prescribed weights w. Such a point would hence meet the power constraint and deviate the least possible amount from customers' comfort, based on the customers' preferences. For example, for $n = 2$, consider an HVAC for indoor air cooling and a water heater as the two types of loads. Suppose the customer's comfort level for indoor air temperature is 23°C and the water temperature is 48°C. Suppose the customer's preference is such that it is more important to maintain the indoor air temperature at 23°C than it is to maintain water temperature at 48°C. In such a case, $1 \geq w_1 > w_2 \geq 0$, where w_1 and w_2 are the weights assigned to the HVAC and the water heater respectively.

Continuing with the considered illustration from Figures 2.10 and 2.11, the role of weights w are observed. z^1 and z^2 represent the feasible coordinates, on the power constraint line, of highest importance on the first and second load, respectively. Any intermediate point on the line joining z^1 and z^2 is a convex combination of the end points. Accordingly, the set M is the collection of all such points, where each can be obtained by translating along the line. This is illustrated in Figure 2.12. The boundary is defined with z^1 and z^2 as the end-points. The midpoint of z^1 and z^2 would hence correspond to equal importance to both loads. The set M from (2.31) is the collection of all the points on the power constraint (red line in Figure 2.12) between z^1 and z^2.

Since such a point in M meets all three objectives of the task considered in this paper, denote this final solution by $z \in \mathbb{R}^n$ as:

$$z = \text{proj}(c, w, p, P)$$

where proj is the formulated weighted projection methodology.

Once the point $z = (z_1, z_2, \ldots, z_n) \in \mathbb{R}^n$ is computed, $z_i \in \mathbb{R}_+$ is the amount of power allotted to the i^{th} type load at time instant t. To determine how to allot the z_i amount of power among the N_i number of loads, the value of the priority function π_{ij}

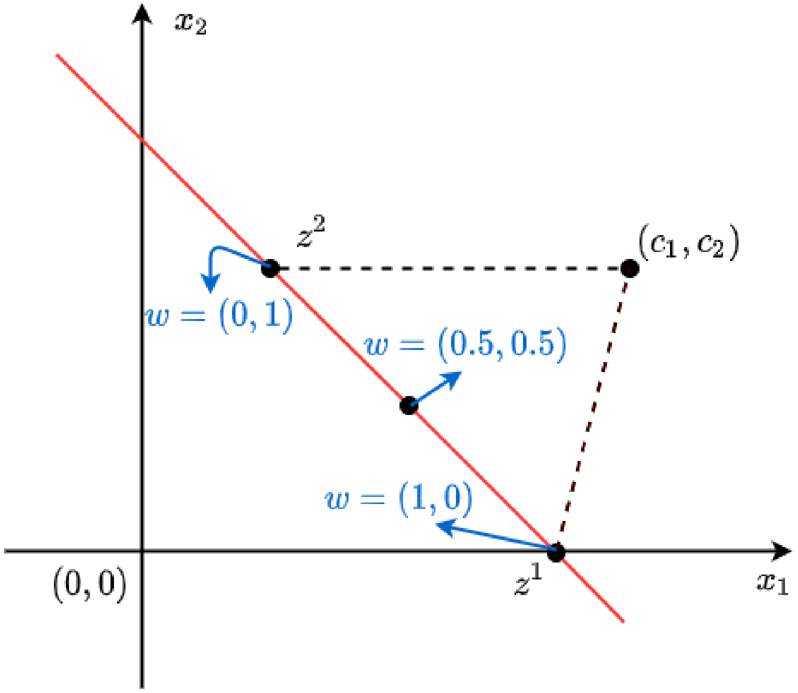


Figure 2.12: Weight-based translation on the power constraint line within the boundary.

is employed. For example, $\pi_{25} > \pi_{29}$ implies in the 2^{nd} type of load, 5^{th} unit needs to be switched on earlier than the 9^{th} unit.

An example: $n = 3$

As an illustration to the discussed theory, consider $n = 3$. The system of equations is:

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} P(t) - c_1 \\ P(t) - c_2 \\ P(t) - c_3 \end{bmatrix}$$

Let the solution of this system of equations be $s = (s_1, s_2, s_3)$ Then, $s^1 = (c_1, s_2, s_3)$ would be a point on the power constraint plane that corresponds to the scenario of highest importance of the first type of load. Similarly, $s^2 = (s_1, c_2, s_3)$ and $s^3 = (s_1, s_2, c_3)$. To ensure boundary conditions, (2.30) is solved to obtain z^1, z^2 and z^3. Then for a specified weight vector w, any $z \in \mathbb{R}^3$ such that $z = z^1 w_1 + z^2 w_2 + z^3 w_3$ would lie in the convex region M and would represent the weighted projection of c onto the power constraint plane.

The formulated methodology is summarized below. At every time instant, for all local controllers and the central controller, do the following:

Local controller: Compute the control input u_{ij}^c (2.4) and communicate to the central controller.

Central controller:

1 For each load type i, compute the priority list π_i from u_{ij}^c, $\forall j = 1$ to N_i

2 Determine the "ideal" power requirement $c \in \mathbb{R}^n$ for all load types from (2.26).

3 Compute the weighted projection $z = \text{proj}\ (c, w, p, P)$ from (2.30)-(2.31).

4 For each load type i, compute u_{ij}^a for all the N_i loads by allocating z_i amount of power based on the priority list π_i.

5 Communicate u_{ij}^a to the local controllers.

Local controller: Supply the assigned input u_{ij}^a to the load.

Chapter 3

The Witsenhausen Counterexample

In the framework of power allocation taken up in Chapter 2, there was one central controller that obtained information from all N local controllers, processed that information in order to meet a certain objective, and communicated back certain control action to the local controllers. All the local controllers aim to maintain the desired setpoint of their loads while together consuming a certain generated power. However, since the local controllers do not communicate with each other, there is a central controller that coordinates their actions. In such a setup, the local controllers form a "team" with the global objective of consuming certain amount of power, and individual objective of maintaining their loads' outputs at desired setpoints.

Team decision theory is a mathematical formalism for a stochastic decision problem in which a team, consisting of two or more team members, cooperates to achieve a common control objective. In the framework of team decision theory, there is only the global objective common to all the team members; there is no individual local objective. The field of team decision theory developed out of the need for a mathematical model of cooperating teams or groups within an organization in which all team members have the same objective yet different information, [77].

The field originated in 1950's with papers by Marschak and Radner, [53] and [65], aimed to solve the objectives of teams within an organization. While there have been some developments, the field has largely stagnated. [77] provides a brief introduction to the field and capitulates the main results in it. While [32] provides some existence results on certain problems with asymmetric information structure, it mentions that significant improvement on the frameworks in the field cannot be made unless new mathematical tools like free probability theory are employed.

The Witsenhausen's counterexample formulated in 1968 is a toy example, with just two team members and a convex objective function, that remained unsolved for decades in the field of team decision theory. It is decentralized stochastic control problem that shows the importance of information structure in a team through a seemingly simple two player team problem. Despite immense attention towards the problem, the non-classical information structure therein had remained the key challenge to obtain the optimal solution until recently [16] (2014), when the Person-by-Person nonlinear optimal laws that satisfy integral equations were derived using Girsanov's transformation.

Here, in this chapter, we compute and implement the nonlinear optimal laws [71]. We begin with a detailed discussion of the counterexample in Section 3.1. Then, in Section 3.2, we employ the Gauss Hermite Quadrature method to approximate the optimal laws that are in integral form and then solve a system of non-linear equations to compute the signaling levels. We then analyze and compare our results with costs previously reported in the literature in Section 3.3.

3.1 Introduction

The Witsenhausen's counterexample [37] is a two-stage stochastic control problem described by the following (state and output) equations, admissible strategies and payoff.

State Equations:

$$x_1 = x_0 + u_1, \tag{3.1}$$

$$x_2 = x_1 - u_2. \tag{3.2}$$

Output Equations:

$$y_0 = x_0, \tag{3.3}$$

$$y_1 = x_1 + v. \tag{3.4}$$

Admissible Strategies:

$$\begin{aligned} u_1 &= \gamma_1(y_0), \\ u_2 &= \gamma_2(y_1). \end{aligned} \tag{3.5}$$

Cost function:

$$J(u_1, u_2) = J(\gamma_1, \gamma_2) \triangleq \mathbb{E}^{u_1,u_2}\Big\{k^2 u_1^2 + x_2^2\Big\}, \;\; k \in \mathbb{R} \tag{3.6}$$

Here, x_0 is a random variable (RV) with known probability density function $p_{x_0}(x)$, $v \in N(0, \sigma^2)$ denotes a zero mean, variance σ^2, Gaussian noise, and x_0 is assumed independent of v. A Gaussian RV x_0 is represented by $x_0 \in N(0, \sigma_x^2)$. Witsenhaussen's fundamental problem is: determine the tuple of strategies (γ_1^*, γ_2^*) that minimize $J(\gamma_1, \gamma_2)$, i.e.,

$$J(\gamma_1^*, \gamma_2^*) \triangleq \inf_{(\gamma_1(y_0), \gamma_2(y_1))} J(\gamma_1, \gamma_2).$$

The complete system and information structure are shown in Fig. 3.1. The information pattern is nonclassical since y_0 is known to the control u_1 applied at the first stage and unknown to the control u_2 applied at the second stage. It is the nonclassical

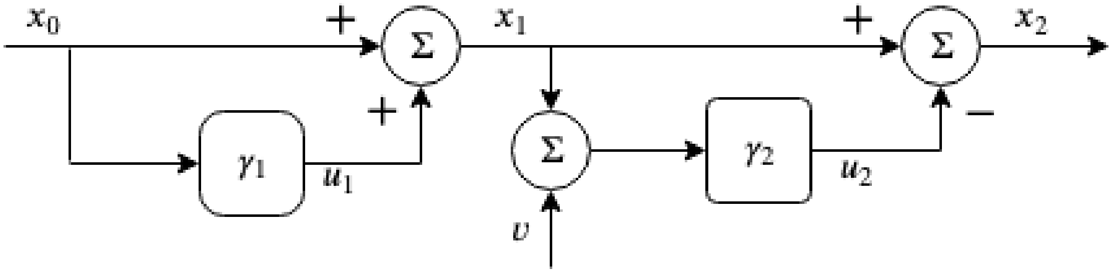


Figure 3.1: Witsenhausen's decentralized stochastic system

information structure that scales up the difficulty of the problem; this was investigated and demonstrated in [37].

Hans Witsenhausen [37] investigated optimal and sub-optimal solutions and demonstrated by construction, that for certain parameters, the optimal affine law is outperformed by a simple nonlinear law. We recapitulate the optimal affine law and the nonlinear law from [37] in Section 3.3. To clarify any existing misconception that the nonlinear law from [37] outperforms all linear laws for all parameters, we highlight the conclusion of Theorem 2 of [37]: it is only for certain parameters that the nonlinear law (3.15), which is not proved to be optimal, outperforms the optimal affine law.

Witsenhausen derived, for a fixed $\gamma_1(x_0)$, the following optimal law for the second controller $\gamma_2^*(y_1)$,

$$\gamma_2^*(y_1) = \mathbb{E}\{x_0 + \gamma_1(x_0)|y_1\}$$

while the optimal first stage law $\gamma_1^*(x_0)$, remained unknown until recently [16], [71]. Since Witsenhausen formulated the counterexample in 1968, several papers appeared in the literature to compute numerically the optimal pay-off. Using finite element analysis, [67] takes up a numerical study of the counterexample by investigating a homogeneous integral equation that arises from the necessary first-order condition for optimiality of the Stage 1 controller. [4] provides a numerical solution to the Witsenhausen counterexample by employing one-hidden-layer neural network as the approximating network. [41] approaches the problem through iterative source-channel coding. [45] develops a hierarchical framework approach to nonconvex optimization problems and to gain additional new insights to the properties of a better solution to the Witsenhausen's counterexample. [56] transforms the problem to an optimization problem over the space of quantile functions and provides a numerical method to generate a sequence of solution approximations. [8] considers a parameterized family of a two-stage control problem with nonclassical information pattern and partition the parameter space into two regions: one where the optimal law is affine and the other where the optimal law is nonlinear. However, an explicit optimal solution had remained elusive until [16] was published in 2014.

Charalambous and Ahmed [16] applied Girsanov's measure transformation to equivalently re-formulated continuous-time and discrete-time decentralized stochastic dynamic decision problems, under a reference probability measure, such that the observations are independent of the controls. When the method is applied to the Witsenhausen's counterexample, the optimal Person-by-Person (PbP) strategies, $(\gamma_1^o(y_0), \gamma_2^o(y_1))$ are obtained, and are given by

$$\begin{aligned}\gamma_1^o(y_0) = &-\frac{1}{2k^2\sigma^2}\mathbb{E}^{\gamma_1^o,\gamma_2^o}\Big\{(y_1 - x_0 - \gamma_1^o)(x_0 + \gamma_1^o - \gamma_2^o)^2\Big\} \\ &- \frac{1}{k^2}\mathbb{E}^{\gamma_1^o,\gamma_2^*}\Big\{x_0 + \gamma_1^o(y_0) - \gamma_2^o|y_0\Big\}\end{aligned}$$

$$\gamma_2^o(y_1) = \mathbb{E}^{\gamma_1^o}\Big\{x_0|y_1\Big\} + \mathbb{E}^{\gamma_1^o}\Big\{\gamma_1^o(y_0)|y_1\Big\}$$

By defining $\overline{\gamma}_1^o(x_0) \triangleq x_0 + \gamma_1^o(x_0)$ then, the optimal tuple is equivalently expressed in integral form,

$$\begin{aligned}\bar{\gamma}_1(x_0) = x_0 - \frac{1}{k^2}\int_{-\infty}^{\infty}\Big\{\frac{1}{2\sigma^2}(\zeta - \bar{\gamma}_1(x_0))(\bar{\gamma}_1(x_0) - \gamma_2(\zeta))^2 + (\bar{\gamma}_1(x_0) - \gamma_2(\zeta))\Big\} \\ \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(\zeta - \bar{\gamma}_1(x_0))^2}{2\sigma^2})d\zeta \quad (3.7)\end{aligned}$$

$$\gamma_2(y_1) = \frac{\int_{-\infty}^{\infty}\bar{\gamma}_1(\xi)\exp\big(-\frac{(y_1-\bar{\gamma}_1(\xi))^2}{2\sigma^2}\big)p_{x_0}(d\xi)}{\int_{-\infty}^{\infty}\exp\big(-\frac{(y_1-\bar{\gamma}_1(\xi))^2}{2\sigma^2}\big)p_{x_0}(d\xi)} \quad (3.8)$$

Here, the tuple of PbP optimal laws are implemented by approximating the integrals (3.7) and (3.8) by employing the Gauss Hermite Quadrature numerical integration method. The resulting coupled approximations are then solved by posing them as a system of nonlinear equations. From the numerical results follows the verification of several of the properties found in Witsenhaussen's paper.

3.2 Numerical Integration of the Optimal Strategies

Consider the optimal strategies in their integral form (3.7) and (3.8). Recognizing that with the exponential function within the integral, the integral form can be reformulated to have a Gaussian exponential function, we employ the Gauss Hermite Quadrature (GHQ) method to implement the optimal strategies.

First, we briefly review the Gauss Hermite Quadrature method. The approximate numerical integration formula for a function $f(x)$ on the infinite range $(-\infty, \infty)$ with the weight function e^{-x^2} is [31]:

$$\int_{-\infty}^{\infty} f(x)e^{-x^2}dx \approx \Sigma_{i=1}^{n} f(x_{i,n})\lambda_{i,n} \tag{3.9}$$

where the abscissas $\{x_{i,n}\}$ are the roots of the n^{th} order Hermite polynomial

$$H_n(x) = -\sqrt{2}^n h_n(\sqrt{2}x) = 0$$

with $h_n(x) = e^{\frac{x^2}{2}} \frac{d^n(e^{\frac{-x^2}{2}})}{dx^n}$ and the weights $\{\lambda_{i,n}\}$ are given by

$$\lambda_{i,n} = \frac{\sqrt{\pi}2^{n+1}n!}{H_n'(x_{i,n})^2}$$

where $H_n^{'}(x) = 2nH_{n-1}(x)$. For $n \leq 10$, the zeros $x_{i,n}$ of the Hermite polynomial $H_n(x)$ and the weights $\lambda_{i,n}$ are calculated in [31]. For higher orders, the zeros and weights are calculated in [62]. It is shown in [30] that the Gauss quadrature rule (3.9) is exact for all continuous functions f that are polynomials of degree $\leq 2n-1$. The implications of quadrature rule to approximate a discontinuous function will be discussed in Section 3.3.

It is in general a difficult problem to compute zeros and weights for any Hermite polynomial and any weight function. Therefore, since the zeros and weights for the aforementioned $H_n(x)$ are calculated in the literature, we transform the optimal strategies (3.7) and (3.8) to have the standard Gaussian function e^{-x^2} as the weight function.

Consider the first law (3.7) and the change of variables as $z = \frac{\zeta - \bar{\gamma}_1(x_0)}{\sqrt{2\sigma^2}}$ and $dz = \frac{d\zeta}{\sqrt{2\sigma^2}}$. Then,

$$\begin{aligned}\bar{\gamma}_1(x_0) = x_0 - \frac{1}{\sqrt{\pi}k^2}\int_{-\infty}^{\infty}\Big\{&\frac{z}{\sqrt{2\sigma^2}}(\bar{\gamma}_1(x_0) - \gamma_2(\sqrt{2\sigma^2}z + \bar{\gamma}_1(x_0)))^2 \\ &+(\bar{\gamma}_1(x_0) - \gamma_2(\sqrt{2\sigma^2}z + \bar{\gamma}_1(x_0)))\Big\}e^{-z^2}dz\end{aligned}$$

Using Gauss Hermite Quadrature approximation (3.9),

$$\begin{aligned}\bar{\gamma}_1(x_0) \approx x_0 - \frac{1}{\sqrt{\pi}k^2}\sum_{i=1}^{n}\Big\{&\frac{z_i}{\sqrt{2\sigma^2}}(\bar{\gamma}_1(x_0) - \gamma_2(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0)))^2 \\ &+(\bar{\gamma}_1(x_0) - \gamma_2(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0)))\Big\}\lambda_i\end{aligned} \tag{3.10}$$

Similarly approximating the second law (3.8) with the change of variable $z = \frac{\xi}{\sqrt{2\sigma_x^2}}$, we get:

$$\gamma_2(y_1) = \frac{\int_{-\infty}^{\infty} \bar{\gamma}_1(\xi) \exp\big(-\frac{(y_1-\bar{\gamma}_1(\xi))^2}{2\sigma^2}\big) \exp\big(-\frac{\xi^2}{2\sigma_x^2}\big) d\xi}{\int_{-\infty}^{\infty} \exp\big(-\frac{(y_1-\bar{\gamma}_1(\xi))^2}{2\sigma^2}\big) \exp\big(-\frac{\xi^2}{2\sigma_x^2}\big) d\xi}$$
$$\approx \frac{\sum_{i=1}^{n} \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_i) \exp\big(-\frac{(y_1-\bar{\gamma}_1(\sqrt{2\sigma_x^2} z_i))^2}{2\sigma^2}\big)\lambda_i}{\sum_{i=1}^{n} \exp\big(-\frac{(y_1-\bar{\gamma}_1(\sqrt{2\sigma_x^2} z_i))^2}{2\sigma^2}\big)\lambda_i} \tag{3.11}$$

Consider (3.10), since z_i and λ_i are the (known) nodes and weights, for certain $x_0 \in \mathbb{R}$, the unknowns are $\bar{\gamma}_1(x_0)$ and $\gamma_2(\sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0)))$ (whose argument is in turn a function of $\bar{\gamma}_1(x_0)$). In order to solve this equation, we employ the expression for $\gamma_2(y_1)$ from (3.11) by having $y_1 = \sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0)$. Substituting $\gamma_2(y_1 = \sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0)))$ from (3.11) in (3.10) to get:

$$\begin{aligned}
\bar{\gamma}_1(x_0) \approx x_0 - \frac{1}{\sqrt{\pi} k^2} \sum_{i=1}^{n} \lambda_i \Bigg\{ & \frac{z_i}{\sqrt{2\sigma^2}} \\
& \Bigg(\bar{\gamma}_1(x_0) - \Bigg(\sum_{j=1}^{n} \big(\bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j) \exp\big(-\frac{(\sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0) - \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j))^2}{2\sigma^2}\big)\lambda_j\big)\Bigg) \Bigg/ \\
& \qquad \Bigg(\sum_{j=1}^{n} \Bigg(\exp\big(-\frac{(\sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0) - \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j))^2}{2\sigma^2}\big)\lambda_j\Bigg)\Bigg)\Bigg)^2 \\
& + \Bigg(\bar{\gamma}_1(x_0) - \Bigg(\sum_{j=1}^{n} \big(\bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j) \exp\big(-\frac{(\sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0) - \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j))^2}{2\sigma^2}\big)\lambda_j\big)\Bigg) \Bigg/ \\
& \qquad \Bigg(\sum_{j=1}^{n} \Bigg(\exp\big(-\frac{(\sqrt{2\sigma^2} z_i + \bar{\gamma}_1(x_0) - \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_j))^2}{2\sigma^2}\big)\lambda_j\Bigg)\Bigg)\Bigg)\Bigg\}
\end{aligned} \tag{3.12}$$

While $x_0 \in \mathbb{R}$ and $\sqrt{2\sigma_x^2} z_i$ are known, $\bar{\gamma}_1(x_0)$ and $\bar{\gamma}_1(\sqrt{2\sigma_x^2} z_i)$ are unknown. Let $s_i = \bar{\gamma}_1(\sqrt{2\sigma_x^2} z_i)$, $\forall i$. For each x_0, (3.12) hence contains $(n + 1)$ number of unknowns, i.e., n s_i's and one $\bar{\gamma}_1(x_0)$:

$$\begin{aligned}\bar{\gamma}_1(x_0) \approx x_0 - \frac{1}{\sqrt{\pi}k^2}\sum_{i=1}^{n}\lambda_i\Bigg\{\frac{z_i}{\sqrt{2\sigma^2}}\Bigg(\bar{\gamma}_1(x_0) - \Bigg(\sum_{j=1}^{n}\big(s_j \exp{(-\frac{(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0) - s_j)^2}{2\sigma^2})}\lambda_j\big)\Bigg)\Bigg/ \\ \Bigg(\sum_{j=1}^{n}\Bigg(\exp{(-\frac{(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0) - s_j)^2}{2\sigma^2})}\lambda_j\Bigg)\Bigg)\Bigg)^2 + \\ \Bigg(\bar{\gamma}_1(x_0) - \Bigg(\sum_{j=1}^{n}\big(s_j \exp{(-\frac{(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0) - s_j)^2}{2\sigma^2})}\lambda_j\big)\Bigg)\Bigg/ \\ \Bigg(\sum_{j=1}^{n}\Bigg(\exp{(-\frac{(\sqrt{2\sigma^2}z_i + \bar{\gamma}_1(x_0) - s_j)^2}{2\sigma^2})}\lambda_j\Bigg)\Bigg)\Bigg)\Bigg\}\end{aligned}$$

Substituting $x_0 = x_{0l} = \sqrt{2\sigma_x^2}z_l$ for each $l \in \{1, 2, \ldots, n\}$, we obtain n nonlinear equations with n s_l's that are unknown, given in (3.13). Each s_l, which is the value of $\bar{\gamma}_1(x_0)$ at nodes selected according to Gauss-Hermite Quadrature, is the signaling level of the control action. Rearranging (3.13) to move all terms on one side, we denote the resulting system of nonlinear equations as $f_{sysnonlin} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$$\begin{aligned}&\forall l = 1, 2, \ldots, n \\ &s_l \approx \sqrt{2\sigma_x^2}z_l - \frac{1}{\sqrt{\pi}k^2}\sum_{i=1}^{n}\lambda_i\Bigg\{\frac{z_i}{\sqrt{2\sigma^2}}\Bigg(s_l - \Bigg(\sum_{j=1}^{n}\big(s_j \exp{(-\frac{(\sqrt{2\sigma^2}z_i + s_l - s_j)^2}{2\sigma^2})}\lambda_j\big)\Bigg)\Bigg/ \\ &\Bigg(\sum_{j=1}^{n}\Bigg(\exp{(-\frac{(\sqrt{2\sigma^2}z_i + s_l - s_j)^2}{2\sigma^2})}\lambda_j\Bigg)\Bigg)\Bigg)^2 + \\ &\Bigg(s_l - \Bigg(\sum_{j=1}^{n}\big(s_j \exp{(-\frac{(\sqrt{2\sigma^2}z_i + s_l - s_j)^2}{2\sigma^2})}\lambda_j\big)\Bigg)\Bigg/ \\ &\Bigg(\sum_{j=1}^{n}\Bigg(\exp{(-\frac{(\sqrt{2\sigma^2}z_i + s_l - s_j)^2}{2\sigma^2})}\lambda_j\Bigg)\Bigg)\Bigg)\Bigg\}\end{aligned} \tag{3.13}$$

The solution of the system of n nonlinear equations (3.13) results in n explicit points, i.e., n signaling levels s_l^*, $\forall l = 1, 2, \ldots, n$, such that $||f_{sysnonlin}(s_1^*, s_2^*, \ldots, s_n^*)||$

is close to zero. Using these n signaling levels, we obtain the value of $\bar{\gamma}_1(x_0)$, $\forall x_0$, by substituting $(s_1^*, s_2^*, \ldots, s_n^*)$ in (3.12) which results in one unknown $\bar{\gamma}_1(x_0)$ and solving the resulting nonlinear equation for each x_0. This is similar to the collocation method used to solve integral equations, [2]. Here, $x_0 = x_{0l} = \sqrt{2\sigma_x^2} z_l$ for each $l \in \{1, 2, \ldots, n\}$ are the collocation points and signaling levels are the value of $\bar{\gamma}_1(x_0)$ at the collocation points.

To obtain the strategy of the second controller, we substitute the signaling levels $(s_1^*, s_2^*, \ldots, s_n^*)$ in (3.11). This directly gives the expression for $\gamma_2(y_1)$ which is evaluated at y_1. It is worth noting here that although $y_1 \in \mathbb{R}$, but because $y_1 = \bar{\gamma}_1(x_0) + v$ from (3.4), the values taken by y_1 are dictated by the strategy of the first controller $\bar{\gamma}_1(x_0)$. Once both the strategies $\bar{\gamma}_1$, γ_2 are obtained, we calculate the total cost J from (3.6).

In this section, we presented the methodology to numerically integrate the derived optimal strategies (3.7) and (3.8). We now briefly summarize it.

Input parameters: k, σ, σ_x, n

Input signals: x_0, v

- Solve $f_{sysnonlin}$ to obtain the signaling levels $(s_1^*, s_2^*, \ldots, s_n^*)$.
- For each x_0, compute $\bar{\gamma}_1(x_0)$
- For all $y_1 = \bar{\gamma}_1(x_0) + v$, compute $\gamma_2(y_1)$

Implementation aspects: We employ the software MATLAB to implement the solution strategies (3.7) and (3.8). The command *fsolve* is used to solve the system of nonlinear equations $f_{sysnonlin}$ and *lsqnonlin* to solve for $\bar{\gamma}_1(x_0)$.

3.3 Results

In this Section, we present the results of our implementation and compare the costs we obtain with the costs previously reported in the literature. The set of parameters in the Witsenhausen counterexample (3.2)-(3.6) are (k, σ, σ_x). For certain set of values of these parameters, the optimal law is affine while for the rest of the region of parameters, the optimal law is non-linear.

In Lemma 1 in [37], Witsenhausen derived the optimal affine laws as:

$$\begin{aligned} \bar{\gamma}_1^{aff}(x_0) &= \nu x_0 \\ \gamma_2(y_1)^{aff} &= \mu y_1 \end{aligned} \tag{3.14}$$

where $\bar{\gamma}_1(x_0) = x_0 + \gamma_1(x_0)$,

$$\mu = \frac{\sigma_x^2 \nu^2}{1 + \sigma_x^2 \nu^2}$$

and $t = \sigma_x \nu$ is a real root of the equation

$$(t - \sigma_x)(1 + t^2)^2 + \frac{1}{k^2} t = 0$$

We denote the cost obtained from the optimal affine laws as $J^{aff} = J(\bar{\gamma}_1^{aff}, \gamma_2^{aff})$. In Theorem 2 of [37] he considers the sample non-linear laws:

$$\begin{aligned} \bar{\gamma}_1^{wit}(x_0) &= \sigma_x \mathrm{sgn}(x_0) \\ \gamma_2^{wit}(y_1) &= \sigma_x \tanh(\sigma_x y_1) \end{aligned} \tag{3.15}$$

and shows that $J^{wit} < J^{aff}$ as $k \to 0$, where J^{wit} is the cost resulting from the nonlinear laws (3.15). That is, $J^{wit} = J(\bar{\gamma}_1^{wit}, \gamma_2^{wit})$.

We denote the cost we obtain from the derived optimal laws (3.7) and (3.8) and implemented using the Gauss Hermite Quadrature numerical integration method detailed in Section 3.2 as J^o. We consider different parameter values of (k, σ, σ_x) and compare the cost we obtain J^o with J^{aff}, J^{wit} and some other costs previously reported in the literature. For additional insight into the results, the total cost is broken into two stages: Stage 1 and Stage 2 costs are the first and the second term, respectively, in the total cost:

$$J(\bar{\gamma}_1, \gamma_2) = \mathbb{E}\big\{k^2(\bar{\gamma}_1(x_0) - x_0)^2 + (\bar{\gamma}_1(x_0) - \gamma_2(y_1))^2\big\} \tag{3.16}$$

We have employed $600,000$ samples for x_0 and v generated according to $\mathcal{N}(0, \sigma_x)$ and $\mathcal{N}(0, \sigma)$ respectively. The order of the Hermite polynomial in GHQ method is $n = 7$. Moreover, as stated in Lemma 1 of [37], the optimal cost is less than $\min(1, k^2\sigma_x^2)$. Accordingly, we verify if the cost J^o is less than $\min(1, k^2\sigma_x^2)$.

Parameters $k = 0.001, \sigma_x = 1000, \sigma = 1$

The total costs obtained are reported in Table 3.1. Note that $J^o < \min(1, k^2\sigma_x^2) = 1$ and so are J^{wit} and J^{aff}. The optimal control laws $\bar{\gamma}_1, \gamma_1$ and γ_2 obtained are shown in Fig. 3.2. As pointed in [37], we observe that $\bar{\gamma}_1$ is indeed symmetric about the origin. Moreover, we obtain four quantization steps, compared to one step resulting from $\bar{\gamma}_1^{wit}$. We also observe that the derived laws result in a strategy such that $\bar{\gamma}_1(x_0) \approx \gamma_2(y_1)$ leading to near zero Stage 2 cost. It is worth pointing out that since γ_2 admits $y_1 = \bar{\gamma}_1(x_0) + v$ as the input, the behavior of γ_2 over the entire real line $\mathbb{R}$ is not apparent.

Parameters $k = 1, \sigma_x = 1, \sigma = 1$

As pointed in [82], this set of parameter values ($k \nless 0.56$ and σ_x is not large) is in the region where affine laws are optimal. The optimal control laws (3.7) and (3.8) are compared with optimal affine laws in Fig. 3.3. It is seen that the resulting laws are the same as the optimal affine laws. We further compare the cost with J^{aff} and

Table 3.1: Total cost, $k = 0.001$, $\sigma_x = 1000$

	Stage 1	Stage 2	Total Cost
J^{aff}	0.9984	9.9843×10^{-7}	0.9984
J^{wit}	0.4041	0	0.4041
J^{o}	0.1137	1.1368×10^{-7}	0.1137

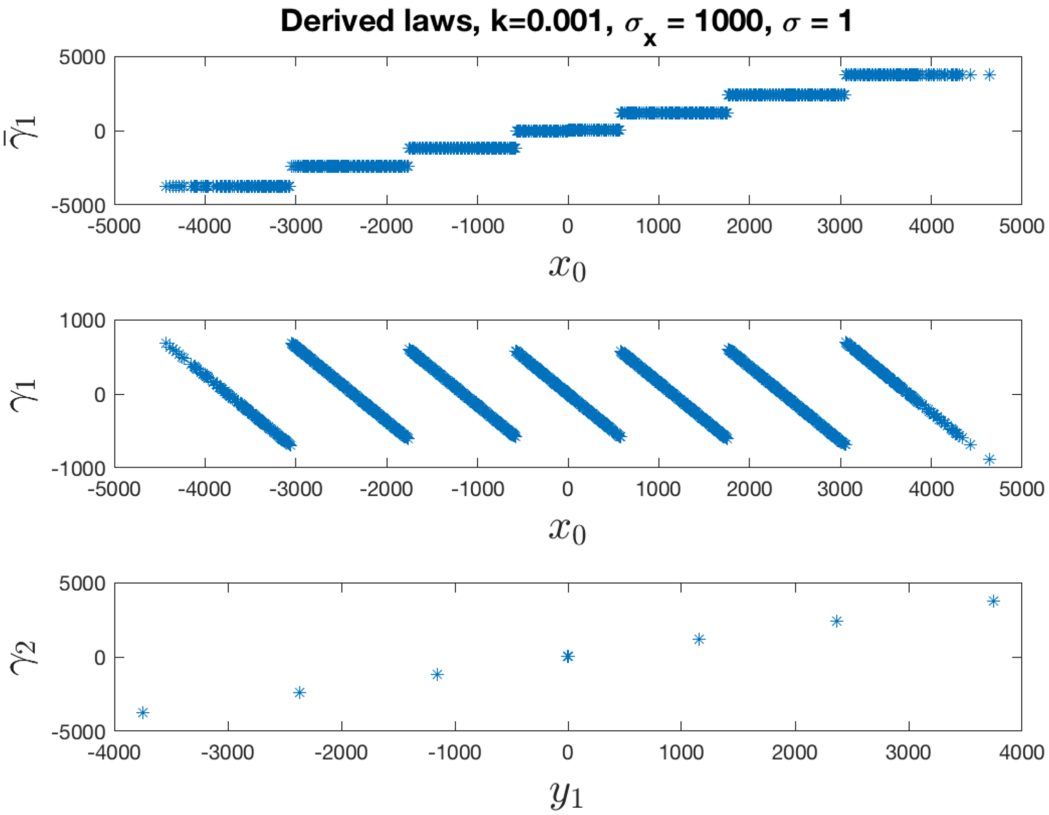


Figure 3.2: Optimal control laws (3.7) and (3.8)

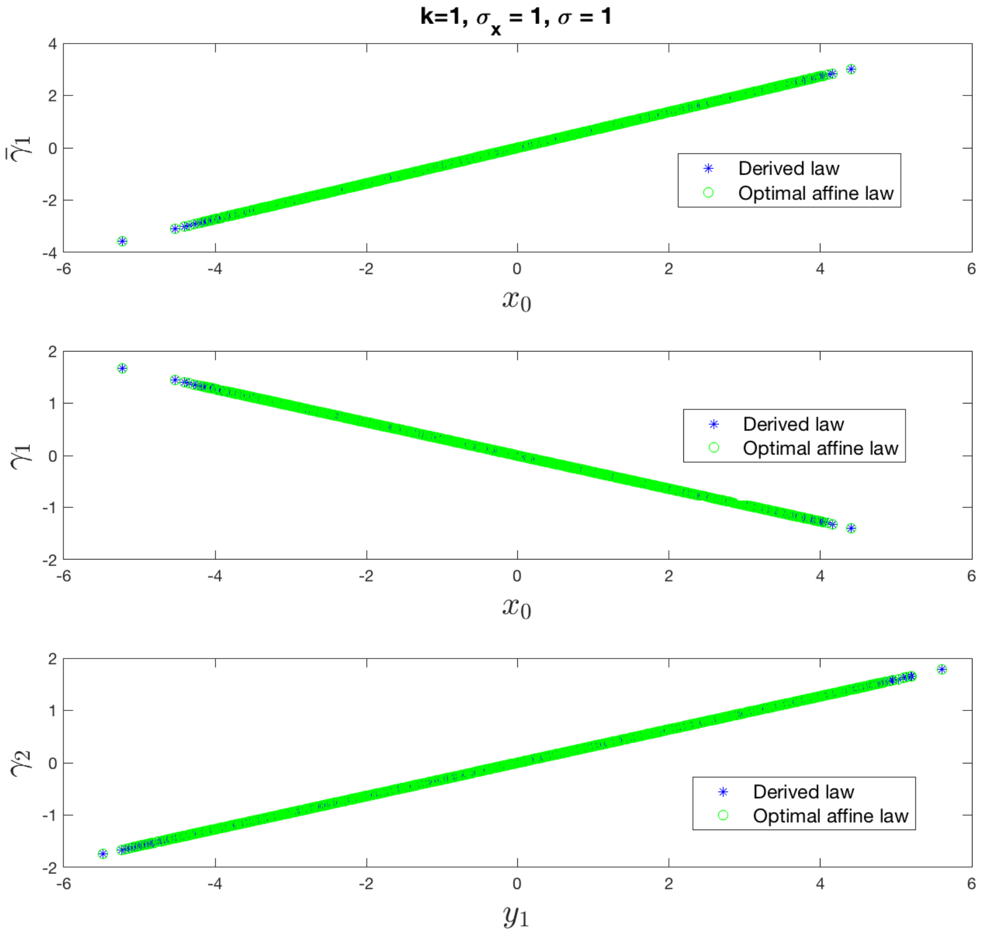


Figure 3.3: Comparison of the optimal control laws and the special class of optimal affine laws

Table 3.2: Total cost, $k = 1, \sigma_x = 1$

	Stage 1	Stage 2	Total Cost
J^{aff}	0.1011	0.3174	0.4185004143524\u200b74
J^{wit}	0.4043	0.4480	0.852287449358227
J^{o}	0.1011	0.3174	0.418500469701766

J^{wit} in Table 3.2. The negligible difference in J^{aff} and J^{o} is attributed to numerical inaccuracy in the implementation of (3.7) and (3.8) through approximate numerical integration method.

Comparison with [9]

A class of nonlinear policies initially introduced in [37] and further analyzed and improved upon in [8] is given by:

$$\begin{aligned} \gamma_1^{bb}(x_0) &= \epsilon^{bb}\text{sgn}(x_0) + \lambda^{bb}x_0 \\ \gamma_2^{bb}(y_1) &= \mathbb{E}[\epsilon^{bb}\text{sgn}(x_0) + \lambda^{bb}x_0|y_1] \end{aligned} \tag{3.17}$$

where ϵ^{bb} and λ^{bb} are parameters to be optimized over. For $k = 0.01, \sigma_x = \sqrt{80}$ and $\sigma = 1$, [9] picks $\epsilon^{bb} = 5$ and $\lambda^{bb} = 0.01006$ in the law (3.17) and reports the cost to be $J^{bb} = 0.3309$. Furthermore, the authors in [57] mention that they obtain the same cost of 0.3309 with the algorithm developed therein. The optimal law $\bar{\gamma}_1(x_0)$ obtained from (3.7) is shown in Fig. 3.4. The corresponding total cost is compared with J^{bb}, J^{wit} and the optimal affine cost J^{aff} in Table 3.3. The stage 2 cost from both J^{o} and J^{aff} is of the order 10^{-7} and from J^{wit} it is 0.

Parameters $k = 0.2, \sigma_x = 5, \sigma = 1$

The last set of parameters we consider has been the most studied case and has enabled more insights into the solution of the Witsenhausen counterexample. [4] provides a numerical solution by employing one-hidden-layer neural network as an approximating network. The cost obtained therein is denoted J^{nn}. Lee, Lau and Ho present a hierarchical search approach in [45]. Therein, they impose $\bar{\gamma}_1$ to be a non-decreasing, step function that is symmetric about the origin. For a number of steps, they find the signaling levels (value of $\bar{\gamma}_1$ at the step) and the breakpoints (x_0 where the step change occurs). They also find that the cost objective is lower for slightly sloped steps

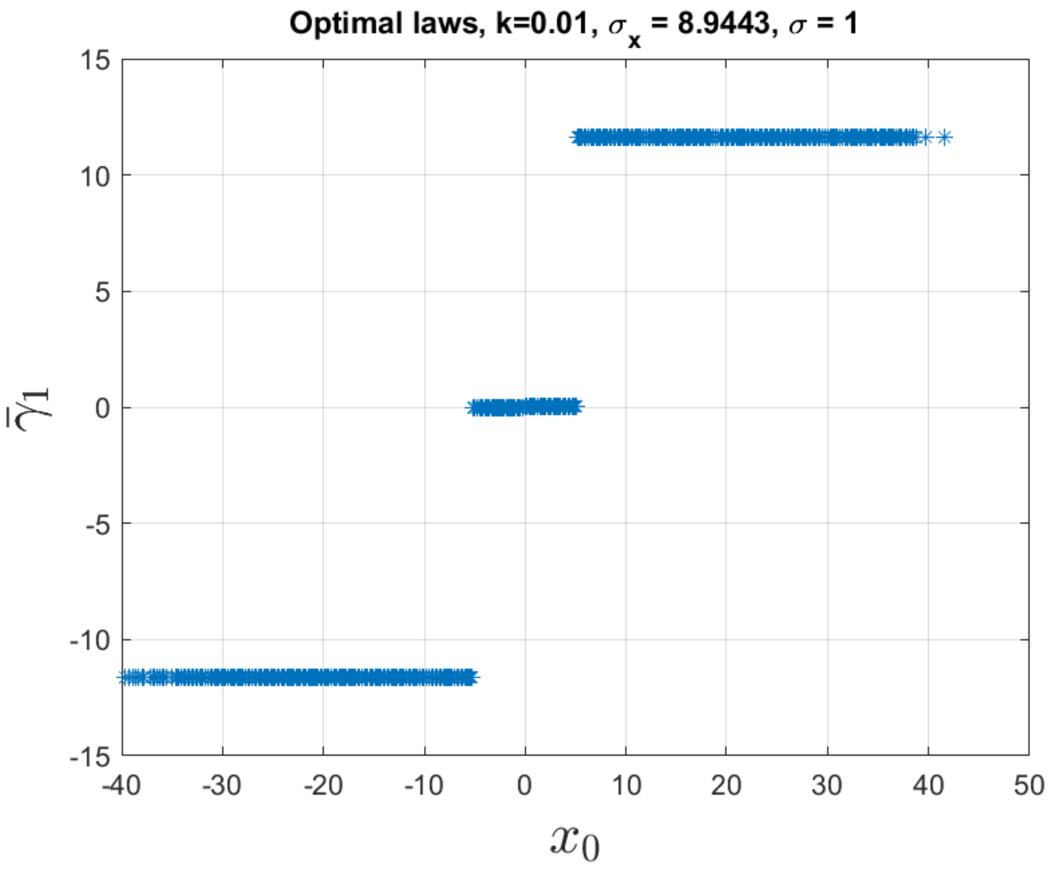


Figure 3.4: Optimal control laws for the parameters in [9]

Table 3.3: Total cost obtained from different solutions

	$k = 0.01, \sigma = 1,\ \sigma_x = \sqrt{80}$
J^{aff}	0.007986277332674
J^{wit}	0.003232551870223
J^{bb}	0.3309
J^{o}	0.001566775786064

than perfectly leveled steps. Through comparison of their costs for different number of steps, they find that $7-$step solution yields the lowest cost. The cost obtained in [45] is denoted J^{llh} here and the signaling levels therein are $s^* = \{0, \pm 6.5, \pm 13.2, \pm 19.9\}$.

In our work, the solution of (3.13) yields the signaling levels

$$s^{**} = \{0, \pm 6.15, \pm 12.8, \pm 19.8\}$$

and $||f_{sysnonlin}(s^{**})|| = 10^{-15}$ while $||f_{sysnonlin}(s^*)|| = 0.7$. Following up on the notes from Section 3.2, the Gauss quadrature rule is not exact for the set of parameters $k = 0.2, \sigma_x = 5, \sigma = 1$ because this parameter set lies in the region where the optimal laws are non-linear. Moreover, the optimal non-linear laws are not continuous; they are only piecewise continuous. As a result, the inaccuracy in the approximation using Gauss quadrature rule reveals itself through the system of nonlinear equations $f_{sysnonlin}$. The cost we obtain for signaling levels s^* and s^{**} are $J^o_* = 0.16$ and $J^o_{**} = 0.1712$, respectively.

The strategy of the first controller, $\bar{\gamma}_1(x_0)$, we obtained for the signaling levels s^* and s^{**} are shown in Fig 3.5. Although we do not externally impose symmetry, it can be observed that $\bar{\gamma}_1$ is symmetric about origin and is non-decreasing. We zoom in on one of the 7 steps and observe in the left column of Fig 3.6 that the steps are slightly sloped. Further zooming in, we see in the right column of Fig 3.6 that each signaling level is further comprised of a number of closely spaced steps. Similar to this result, the authors in [45] added segments in each of the 7 steps to obtain the cost $J^{llh} = 0.167313205338$. We compare both the costs with previously reported costs in the literature in Table 3.4. Further in agreement with the findings in [45], we obtain the lowest cost for 7 steps, $J^o_* * = 0.1712$.

With the parameter set $k = 0.2, \sigma_x = 5, \sigma = 1$, the number of steps we obtain is the same as the value of the Gauss quadrature rule parameter n. However, this is not necessarily the case for all parameter sets; for example see Section 3.3. The parameter set $k = 1, \sigma_x = 5, \sigma = 1$ is known to lie in a region where the optimal law is affine,

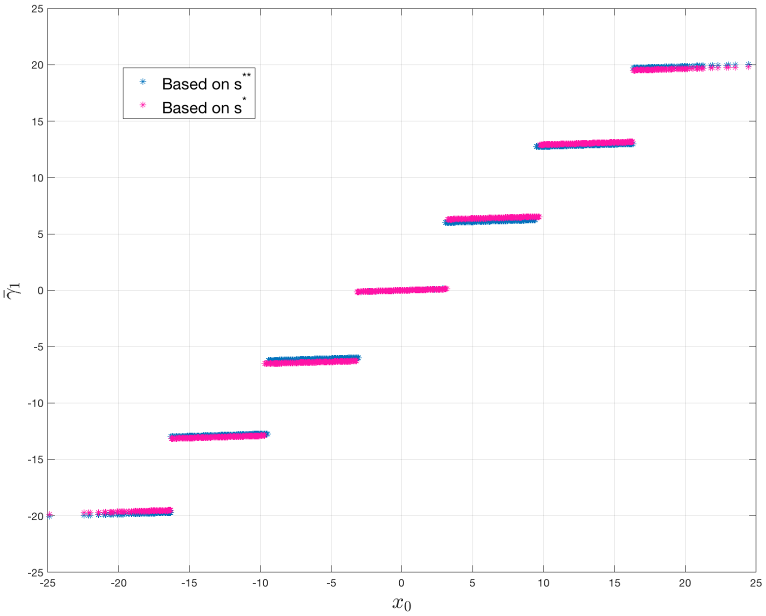


Figure 3.5: Comparison of obtained signaling levels s^{**} with those from [45]

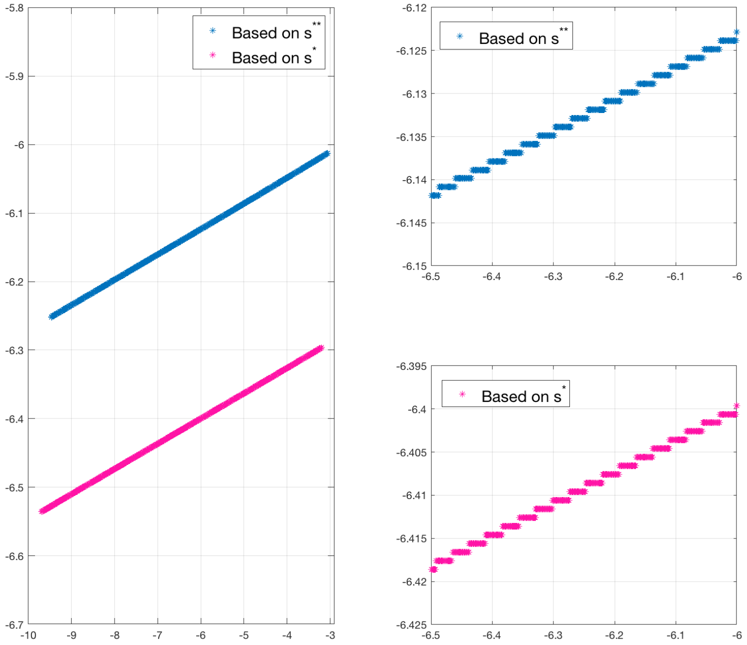


Figure 3.6: Magnifying one signaling level from Fig. 3.5 highlights that the levels are slightly sloped

Table 3.4: Reported and obtained costs, $k = 0.2, \sigma_x = 5, \sigma = 1$

	Stage 1	Stage 2	Total Cost
J^{aff}	0.0017428616051158	0.956950417234115	0.958693278839234
J^{wit}	0.403507741927546	$2.134488364684996 \times 10^{-6}$	0.403509876415911
J^{nn}[4]	-	-	0.1735
J^{llh} [45]	0.131884081844	0.035429123524	0.167313205368
J^{o}_{*}	0.128541364988695	0.038385613344897	0.166926978333592
J^{o}_{**}	0.120110042087359	0.051158481289032	0.171268523376388

and even though we employ $n = 7$ steps for GHQ, the resulting control laws are affine. Likewise, as seen in Fig 3.4, the parameter set lies in the region where the optimal law is non-linear and we obtain a three-step control law for $\bar{\gamma}_1$ for the GHQ parameter $n = 7$.

Chapter 4

Centroidal Voronoi Tessellations and their Computation

The field of team decision theory is broad, encompassing almost every architecture in a team problem. However, as seen in Chapter 3, the field still remains largely in its theoretical development stage. One of the challenges preventing its applicability in real-world problems is the difficulty in obtaining optimal decentralized solutions when the information structure is non-classical. Accordingly, solving the resource allocation problem might prove extremely cumbersome in a team decision framework. Therefore, as a next step in this work, we explore another approach – through Centroidal Voronoi tessellations (CVTs).

In this Chapter, we formalize the definition of CVTs and study its properties in Section 4.1. We look into the computation of CVTs using Lloyd's algorithm in Section 4.2.1, formalize a method to analytically obtain CVTs in one-dimensional spaces in Section 4.2.2, and compare the two in Section 4.2.3. We then develop a CVT computation method for higher dimensional spaces in Section 4.3.1 and present the corresponding results in Section 4.3.2.

4.1 Centroidal Voronoi Tessellation

Consider $\Omega \subset \mathbb{R}^n$. Let $V_i \subset \Omega$, and $z_i \in \Omega$. Denote by I the set of indices: $I_N = \{1, 2, \ldots, N\}$, $N \in \mathbb{N}$. Let $\rho(x)$ be the density function associated with Ω. The following are some helpful definitions to build the understanding of a Centroidal Voronoi Tessellation, [24].

1 Tessellation: A collection of N number of subsets of Ω is called a tessellation of Ω if the subsets have disjoint interiors and their union makes up Ω. That is, $\{V_i\}_{i \in I_N}$ is a tessellation of Ω if $V_i \cap V_j = \emptyset$ for $i \neq j$, and $\cup_{i \in I} \bar{V}_i = \bar{\Omega}$, where $\bar{V}_i$ is the closure of V_i.

2 Voronoi region and generators: The Voronoi region $\hat{V}_i$ corresponding to a point z_i is a collection of points in Ω that are closer to z_i than z_j, $\forall i \neq j$. That is, $\hat{V}_i = \{x \in \Omega : ||x - z_i|| < ||x - z_j||,\ i \neq j \text{ and } i, j \in I_N\}$. Correspondingly, the points $\{z_i\}_{i \in I_N}$ are called the Voronoi generators.

3 Voronoi tessellation: The set of all the Voronoi regions, $\{V_i\}_{i \in I_N}$, corresponding to the generators $\{z_i\}_{i \in I_N}$ is called a Voronoi tessellation (VT).

Let $\mathbf{z} = \{z_i\}_{i \in I_N}$ denote the set of Voronoi generators in Ω. Similarly, denote by $\mathbf{V_z} = \{V_i\}_{i \in I_N}$ the set of Voronoi regions corresponding to the set of generators $\mathbf{z}$. A Voronoi tessellation formed from generator $\mathbf{z}$ would hence be denoted $VT = \{\mathbf{z}, \mathbf{V_z}\}$.

For a visual demonstration, consider $n = 1$ and let $\Omega \subset \mathbb{R}$ be $[0, 15]$. Consider two sets of generators $\mathbf{z}^1 = \{3, 4, 9\}$ and $\mathbf{z}^2 = \{6, 9, 14\}$ in Ω. Fig. 4.1 shows their corresponding Voronoi regions $\mathbf{V}^1_\mathbf{z}$ and $\mathbf{V}^2_\mathbf{z}$. The top line indicates the region Ω, the middle graph shows the Voronoi tessellation corresponding to $\mathbf{z}^1$, and the bottom graph corresponding to $\mathbf{z}^2$. The generators are marked as dots in colors red, green and blue, and their Voronoi regions are marked as solid lines in respective colors for both the Voronoi tessellations. Notice that the generators do not appear to the "centers" in their Voronoi regions. To be precise, we need to include the probability distribution of the region Ω in order to define its "center", or the mass centroid to be exact.

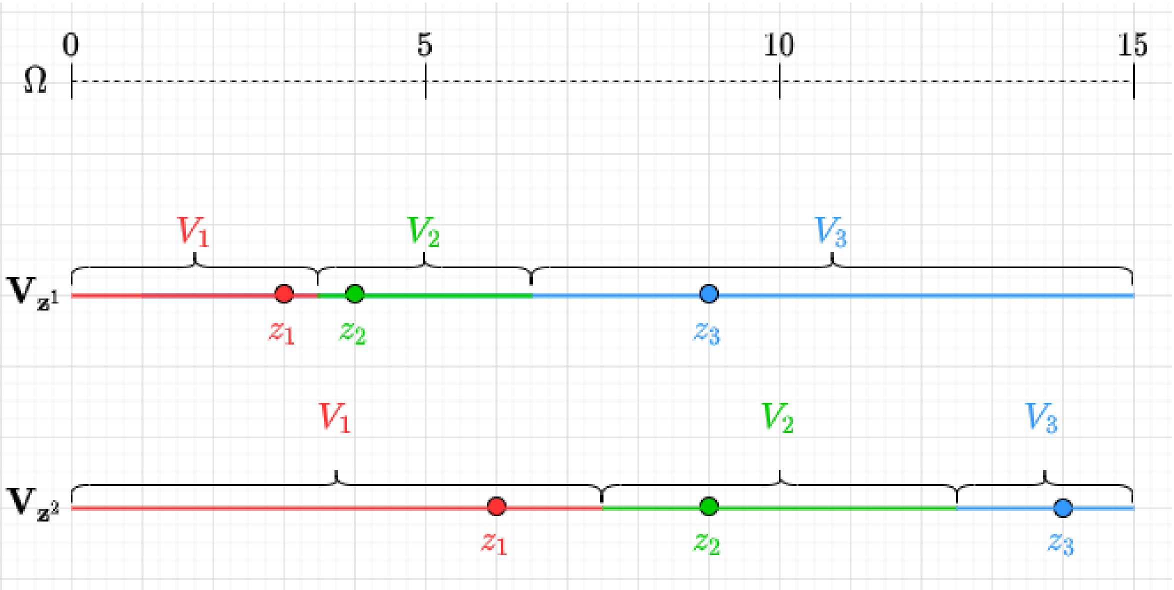


Figure 4.1: Voronoi Tessellations of $\mathbf{z}^1$ and $\mathbf{z}^2$ in the region $[0, 15]$.

The mass centroid of a region $V_i \subset \Omega$ under the probability density function $\rho(x)$ is defined as:

$$z^c_{V_i,\rho} = \frac{\int_{V_i} x\rho(x)dx}{\int_{V_i} \rho(x)dx} \tag{4.1}$$

Consider the same one-dimensional region from Fig. 4.1 with generators $\mathbf{z}^1$ and $\mathbf{z}^2$ and their corresponding Voronoi regions $\mathbf{V}^1_{\mathbf{z}}$ and $\mathbf{V}^2_{\mathbf{z}}$. Consider Uniform and Normal distributions on Ω: $\mathcal{U}(0, 15)$ and $\mathcal{N}(7.5, 1)$. Fig. 4.2 shows the mass centroids of the two Voronoi tessellations under the considered distributions. The centroids of the Voronoi regions $\mathbf{V}^1_{\mathbf{z}}$ for Uniform and Normal distributions are shown in the center graph, marked by star and square symbols, respectively. Notice that the centroids corresponding to the uniform distribution are at the center of the Voronoi regions, whereas the centroids corresponding to the normal distribution are closer to each other and seeming to approach to cluster around the specified mean 7.5. Similarly, the bottom graph shows the centroids of the Voronoi regions of $\mathbf{z}^1$. The key point to note in Fig. 4.2 is that $\mathbf{z}^1 \neq z^c_{\mathbf{V}_{\mathbf{z}^1},U} \neq z^c_{\mathbf{V}_{\mathbf{z}^1},N}$. That is, the generators $\mathbf{z}^1$ are not the mass centroids for either of the considered distributions. Same is true for $\mathbf{z}^2$.

Accordingly, a Voronoi tessellation where the generators of all the Voronoi regions are the mass centroids of their respective Voronoi regions is called a *Centroidal Voronoi Tessellation* (CVT), [24]. Since it is not the case in Fig. 4.2, the two Voronoi tessellations showed therein are not CVTs. In order to obtain the CVT, one can follow its definition and move the generators to the mass centroids. The Voronoi regions corresponding to the new generator points is obtained, and their mass centroids computed. If the generators are the mass centroids, then the tessellation is a centroidal Voronoi tessellation. If not, the procedure can be repeated until a CVT is obtained. Such an iterative method is called Lloyd's algorithm, and will be further discussed in Section 4.2.1.The CVT obtained for the region $\Omega = [0, 15]$ under the distributions $\mathcal{U}(0, 15)$ and $\mathcal{N}(7.5, 1)$ is shown in Fig. 4.3.

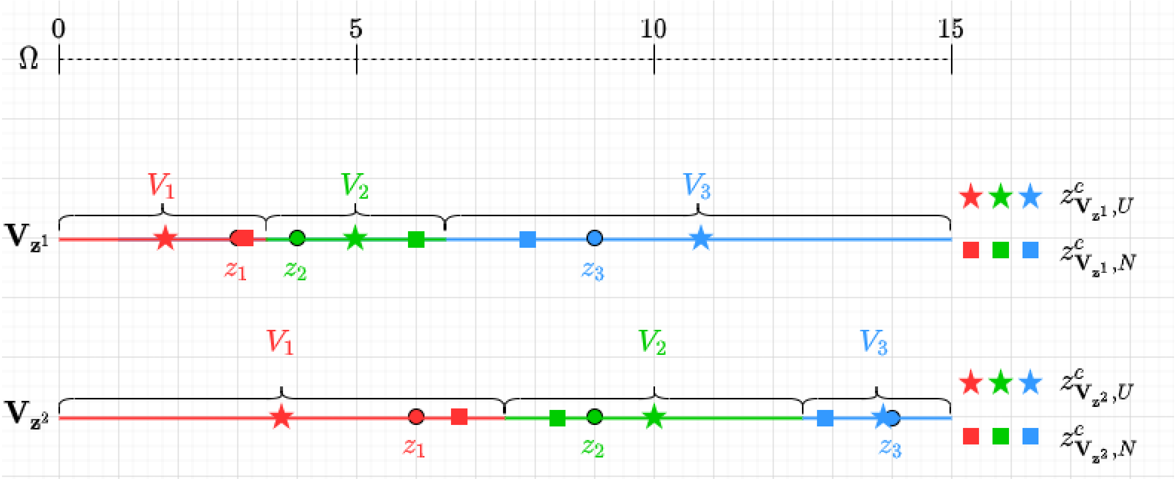

0
5
10
15
Ω
V_1
V_2
V_3
$\mathbf{V}_{\mathbf{z}^1}$
z_1
z_2
z_3
$z^c_{\mathbf{V}_{\mathbf{z}^1},U}$
$z^c_{\mathbf{V}_{\mathbf{z}^1},N}$
V_1
V_2
V_3
$\mathbf{V}_{\mathbf{z}^2}$
z_1
z_2
z_3
$z^c_{\mathbf{V}_{\mathbf{z}^2},U}$
$z^c_{\mathbf{V}_{\mathbf{z}^2},N}$

Figure 4.2: Mass centroids of Voronoi Tessellations of $\mathbf{z}^1$ and $\mathbf{z}^2$ in the region $[0, 15]$ under Uniform and Normal distributions.

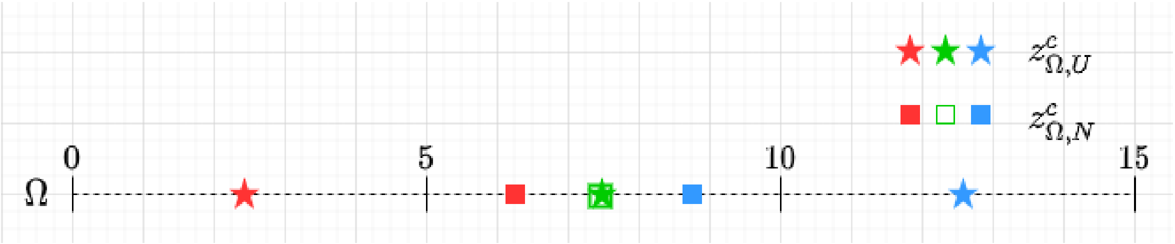


Figure 4.3: Centroidal Voronoi Tessellations of $[0, 15]$ under Uniform and Normal distributions.

Moreover, the centroids are independent of the generators $\mathbf{z}^1$ and $\mathbf{z}^2$, as seen in Fig. 4.3. One may be tempted to conclude that CVT is unique in this example, and it is indeed true for one-dimension under certain assumptions. However, in general, the solution to finding the CVT is not unique. We will now briefly look into the uniqueness properties of CVTs.

4.1.1 Uniqueness of Centroidal Voronoi Tessellations

In order to discuss the uniqueness of CVTs, it is helpful to first mathematically formulate the problem of finding them. Given a region $\Omega \subset \mathbb{R}^n$, a positive integer N, and a probability density function $\rho(x)$ on Ω, consider the functional $\mathcal{F}$ with any N points $z_i \in \Omega$ and any tessellation $\{V_i\}_{i\in I_N}$ of Ω as input to its arguments:

$$\mathcal{F}((z_i, V_i), i \in I_N) = \sum_{i\in I_N} \int_{x\in V_i} \rho(x)||x - z_i||^2 dx \tag{4.2}$$

Proposition 3.1 in [24] states that a necessary condition for the function $\mathcal{F}$ to be minimized is that $\{V_i\}_{i\in I_N}$ are the Voronoi regions corresponding to $\{z_i\}_{i\in I_N}$, and simultaneously, $\{z_i\}_{i\in I_N}$ are the centroids of their respective Voronoi regions. In other words, the minimizer of the functional $\mathcal{F}$ is a Centroidal Voronoi Tessellation.

Additionally, if the tessellation in (4.2) is fixed to be the Voronoi tessellation of $\{z_i\}_{i\in I_N}$, then the following functional $\mathcal{K}$ has the same minimizer as $\mathcal{F}$, [24].

$$\mathcal{K}((z_i), i \in I_N) = \sum_{i\in I_N} \int_{x\in V_i} \rho(x)||x - z_i||^2 dx \tag{4.3}$$

This functional $\mathcal{K}$ is also referred to as the energy of the tessellation or the quantization energy. It is stated and proved in Lemma 3.4 in [24] that $\mathcal{K}$ is continuous and that it possesses a global minimum. Moving from the existence of the CVT to its uniqueness, [24] also mentions that $\mathcal{K}$ may have local minimizers. Any more results on the uniqueness of CVTs require additional assumptions.

It is showed in [27] that the solution of (4.3) is unique in one-dimensional regions with a logarithmically concave continuous probability density function of finite second moment. As reiterated in [76], for $n = 1$, the logarithmic concavity condition implies that any CVT is a local minimum, and further, that there is a unique CVT that is both a local and a global minimum of $\mathcal{K}(\mathbf{z})$, where $\mathbf{z} = \{z_i\}_{i \in I_N}$. Accordingly, since the two distributions – Uniform and Normal – considered in Fig. 4.2 on the one-dimensional region $\Omega = [0, 15]$ are log-concave with finite second moment, we have that the CVTs showed therein are the global minima for the two distributions.

The solution of (4.3) are also called scalar quantizers for $n = 1$ and vector quantizers for higher dimensions. The conditions on the uniqueness of vector quantizers for the general case, that is, no assumptions on the region, density or the number of quantizers N, remain an open area of research. However, it is proved in [76] that for $N = 2$, there does not exist a unique CVT for any density for dimensions greater than one.

For a graphic illustration, one may consider a square in $\mathbb{R}^2$ with two generator points under Uniform distribution. As shown in Fig. 4.4, there are multiple CVTs: all the three Voronoi tessellations shown are centroidal. Additional CVTs can be obtained through rotations.

To summarize, the solution of (4.3) is unique in one-dimensional regions for log-concave density functions but remains an open and actively sought out question in higher dimensions for general cases. In the next Section, we look into ways to obtain the centroidal Voronoi tessellation.

4.2 Computation of CVT in 1-D Spaces

Given a region $\Omega \subset \mathbb{R}^n$, a number of generators N, and a density function $\rho(x)$ over Ω, there are various iterative algorithms to compute a CVT in Ω. As noted in the previous Section, in general, the CVT need not be unique for any dimensional region unless certain conditions are imposed on the density function. Accordingly, the solutions rendered by all the algorithms, to compute the CVT, need not be the unique global

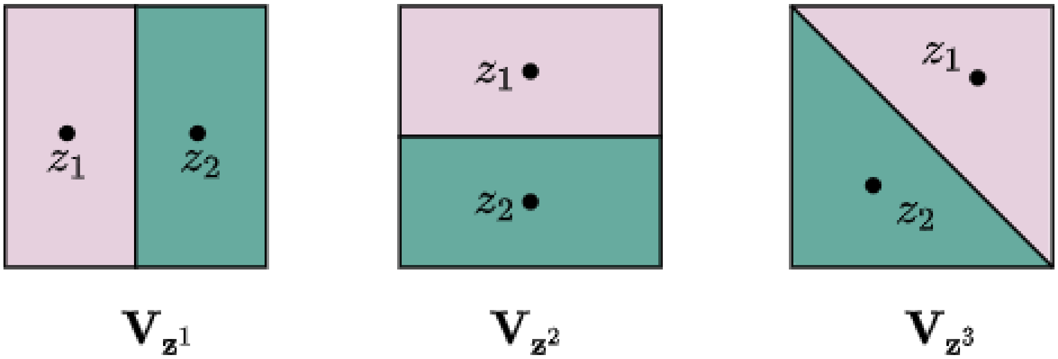

z_1
z_2
z_1
z_2
z_1
z_2
$\mathbf{V}_{\mathbf{z}^1}$
$\mathbf{V}_{\mathbf{z}^2}$
$\mathbf{V}_{\mathbf{z}^3}$

Figure 4.4: Two generator centroidal Voronoi tessellations of a square with uniform density.

minimizers. In this Section, we first describe perhaps the most widely employed of all the algorithms: Lloyd's algorithm, and then develop a computation method to obtain the exact solution of the problem of finding the CVT in one-dimensional regions.

4.2.1 Lloyd's Algorithm

Introduced in [47] to find the optimal quantization in pulse-code modulation, Lloyd's algorithm, also known as Voronoi iteration or relaxation, has been modified or adapted in various fields. At the core of it, Lloyd's algorithm is an iteration between constructing Voronoi tessellations and their centroids:

```
Given: $\Omega \subset \mathbb{R}^n$, $N$, $\rho(x)$
Initialize: Generators $\mathbf{z} = \{z_i\}_{i \in I_N}$, where each $z_i \in \Omega$

    1 Construct the Voronoi tessellation $\mathbf{V_z}$.

    2 Compute the mass centroids $z^c_{\mathbf{V}_{\mathbf{z},\rho}}$ of $\mathbf{V_z}$.

    3 If the computed centroids meet certain stopping criteria then terminate. If
      not, then set $\mathbf{z} = z^c_{\mathbf{V}_{\mathbf{z},\rho}}$, and return to Step 1.
```

Since we are looking for a CVT, the search/iterations stop when the centroids of the Voronoi regions are the generators. Accordingly, the stopping criteria can be defined to have the generators same as the centroids with some tolerance. For example, $|\mathbf{z} - z^c_{\mathbf{V}_{\mathbf{z},\rho}}| \leq 10^{-6}$ can be a stopping criteria. As a result, the accuracy of the resulting solution depends on the specified tolerance.

Even though Lloyd's algorithm is iterative and approximate, it has certain desirable convergence properties. Various global convergence properties of the Lloyd's algorithm are rigorously proved in [23]. Moreover, [23] also surveys and presents the complete compilation of all the results concerning the convergence of the Lloyd's algorithm. Additionally, the authors of [23] also rigorously prove various global convergence theorems: the global convergence of subsequences for any density functions, the global convergence of the whole sequence in one-dimensional space, and the global convergence under some non-degeneracy conditions. Specifically for one-dimensional spaces with log-concave density functions, the local convergence using the Lloyd's algorithm has been proved in [42]. Since the CVT in one-dimensional spaces with log-concave density functions is both a local and global minimum of (4.3), one can conclude that the Lloyd's algorithm converges to the unique globally minimizing CVT in such cases. In higher dimensions, as discussed in Section 4.1.1, the CVT need not be unique, and the Lloyd's algorithm converges to one of them. Moreover, Lloyd's algorithm has linear convergence rate that is slow for large-scale problems. Therefore, depending on the application at hand, various algorithms that have faster convergence than Lloyd's have been proposed, [46], [79], [34].

Finding the exact solution of the problem of finding the CVT is highly non-trivial, especially as the dimension of the space grows. For one-dimensional spaces, we formalize a method to analytically compute the exact solution.

4.2.2 System of Nonlinear Equations

The core idea behind the analytical computation of the CVT in one-dimensional region is to parameterize the Voronoi regions in terms of their centroids. In a region $\Omega = [a, b] \subset \mathbb{R}$, without loss of generality, let the N generator points be $z_1 < z_2 < \ldots < z_N \in \Omega$. By definition of Voronoi regions as described in Section 4.1, the Voronoi regions of these generator points are given as:

$$\begin{aligned} V_1 &= [a, \frac{z_1+z_2}{2}] \\ V_2 &= [\frac{z_1+z_2}{2}, \frac{z_2+z_3}{2}] \\ &\vdots \\ V_i &= [\frac{z_{i-1}+z_i}{2}, \frac{z_i+z_{i+1}}{2}] \\ &\vdots \\ V_N &= [\frac{z_{N-1}+z_N}{2}, b] \end{aligned} \tag{4.4}$$

Since, by definition of CVT, the N generators of the centroidal Voronoi tessellation in Ω with density $\rho(x)$ are the centroids of the Voronoi regions, the centroids from (4.1) can be rewritten in terms of the parameterized Voronoi regions from (4.4) with centroids as generators. The resulting centroids are given as:

$$\begin{aligned} z_1^c &= \frac{\int_a^{\frac{z_1^c+z_2^c}{2}} x\rho(x)dx}{\int_a^{\frac{z_1^c+z_2^c}{2}} \rho(x)dx} \\ z_2^c &= \frac{\int_{\frac{z_1^c+z_2^c}{2}}^{\frac{z_2^c+z_3^c}{2}} x\rho(x)dx}{\int_{\frac{z_1^c+z_2^c}{2}}^{\frac{z_2^c+z_3^c}{2}} \rho(x)dx} \\ &\vdots \\ z_i^c &= \frac{\int_{\frac{z_{i-1}^c+z_i^c}{2}}^{\frac{z_i^c+z_{i+1}^c}{2}} x\rho(x)dx}{\int_{\frac{z_{i-1}^c+z_i^c}{2}}^{\frac{z_i^c+z_{i+1}^c}{2}} \rho(x)dx} \\ &\vdots \\ z_N^c &= \frac{\int_{\frac{z_{N-1}^c+z_N^c}{2}}^{b} x\rho(x)dx}{\int_{\frac{z_{N-1}^c+z_N^c}{2}}^{b} \rho(x)dx} \end{aligned} \tag{4.5}$$

where $z^c_{V_i,\rho}$ from (4.1) is denoted as z^c_i for ease of notation. In (4.5), there are N number of unknowns: z^c_i's, and N equations. Therefore, solving this system of nonlinear equations (SNLE) will result in the generators of the centroidal Voronoi tessellation, which is the exact solution of the functional $\mathcal{K}$ from (4.3).

There are various methods to solve the SNLE from (4.5), and depending on the density function, one can choose the most suitable method. The two main approaches to analytically solving an SNLE are by substitution and elimination. If the exact solutions cannot be determined by algebraic methods, then one can approximate the solutions using numerical methods, [66]. An additional advantage of employing numerical methods is that they usually allow the problem framework to remain generic. For example one need not choose a numerical method specifically for a density function. Here, the SNLE from (4.5) is solved in MATLAB using the command fsolve that uses the Powell's dog leg algorithm, an iterative optimization algorithm similar to the Levenberg-Marquardt algorithm. In the next Section, along with a brief comparison of the two methods discussed in this section, their solutions are presented.

4.2.3 Numerical Results

One of the main differences between Lloyd's algorithm and the SNLE method is that the former is an approximate method to obtain a CVT, while the latter is not. However, since the CVT is unique in one-dimensional spaces (see Section 4.1.1), and in conjunction with the convergence properties of the Lloyd's algorithm, it is inferred that the solution from the Lloyd's algorithm is indeed the centroidal Voronoi tessellation. On the other hand, while the SNLE method can be exact, in order to preserve the generalizability in the framework, we take a numerical approach to solve it. Therefore, while the solution can be sensitive to the initial conditions, with an appropriate choice of the initial conditions and given the uniqueness of the CVT in one-dimensional spaces, the solution from both the methods are expected to be very close to each other.

In the subsequent figures, we will compare the solutions from both the methods under various conditions. The region $\Omega = [0, 20]$ is the x-axis that is shown while the number of generators N and the density function are explicitly mentioned in each graph. Fig. 4.5 shows the locations of the N generators in the CVT in Ω under a Gaussian distribution with mean 10 and standard deviation 2. The top graph considers 5 generators, the middle one 15, and the bottom one considers 45 generators. The solution $\mathbf{z}^l$ from the Lloyd's algorithm is marked by the diamond symbol, and the solution of the SNLE by the asterisk symbol. The generator points are marked by N different colors while keeping the corresponding solutions from the two methods in the same color but different symbol. That is, z_i^l and z_i^c are marked in the same color but different symbols. To preserve the readability of the graphs, the corresponding Voronoi regions of $\mathbf{z}^l$ and $\mathbf{z}^c$ are not shown. However, one can visualize the Voronoi regions with the aid of the generators and their Voronoi regions from Fig. 4.2.

Similarly, Fig. 4.6 shows the centroidal Voronoi tessellations with 25 generators in $\Omega = [0, 20]$ under an Exponential distribution with the rate parameter λ increasing from 0.1 to 0.5 to 1 from top to bottom. While the number of generators were varied in Fig. 4.5 under the fixed, symmetric Gaussian distribution, the number of generators is fixed in Fig. 4.6 under the varying, asymmetric Exponential distribution.

The two methods discussed in Section 4.2.1 and 4.2.2 are scalable to any number of generators, theoretically. The Lloyd's algorithm in particular is more robust to a very high N in comparison with the SNLE method, since the latter involves solving a system of N equations. A small demonstration of the scalability of the two methods is shown in Fig. 4.7 where the CVT is obtained for 100 generators under three different density functions: [log-concave] Gaussian with mean 10 and deviation 2, [not log-concave] Cauchy with parameters $x_0 = 10,\ \gamma = 2$, and [log-concave] Gamma distribution with shape parameter 2 and rate parameter 5. An additional noteworthy point is the non-specificity of the two methods regarding the underlying probability densities; while uniqueness may not be guaranteed in non-log-concave distributions, the two methods are applicable to any probability distribution in one-dimension.

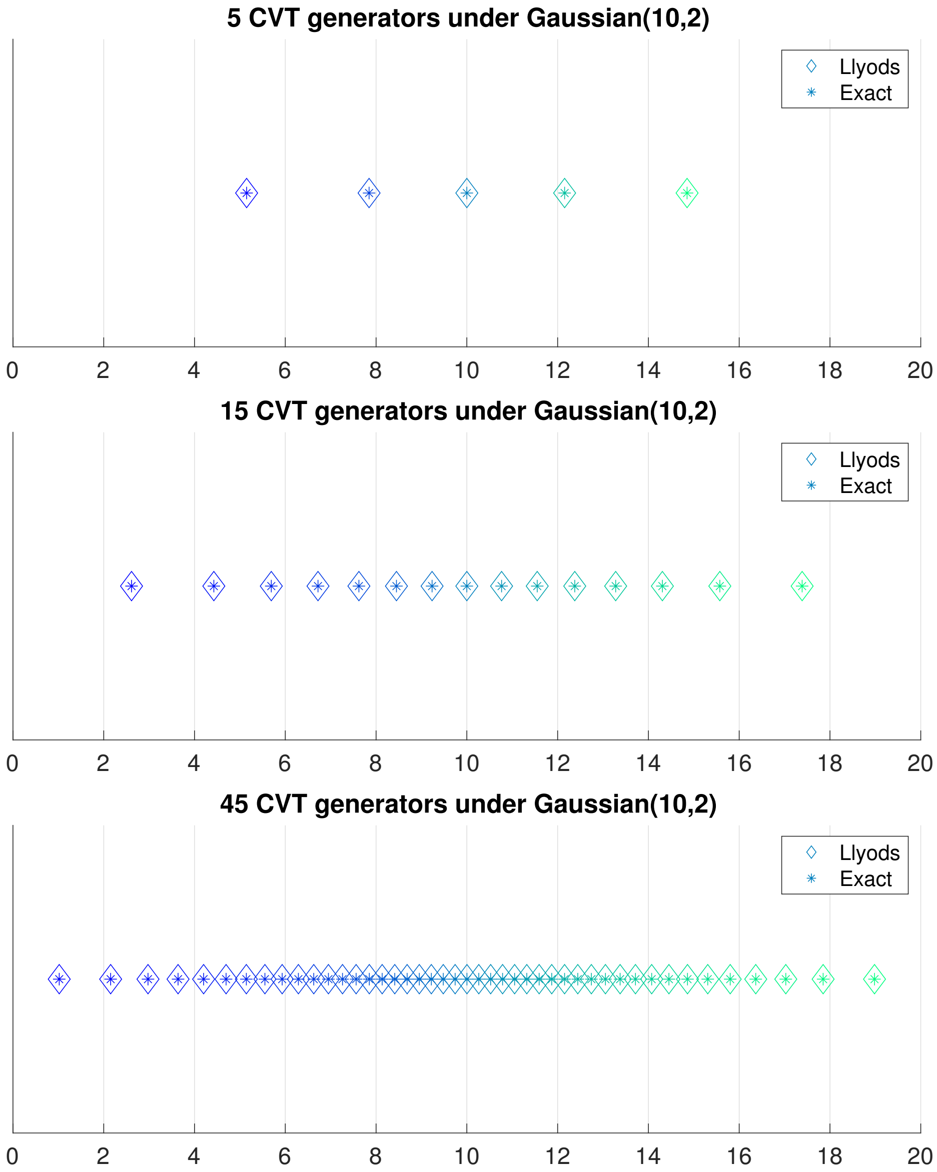


Figure 4.5: Centroidal Voronoi tessellations from 5 (top), 15 (middle) and 45 (bottom) generators in $\Omega = [0, 20]$ under Gaussian distribution with mean 10 and standard deviation 2.

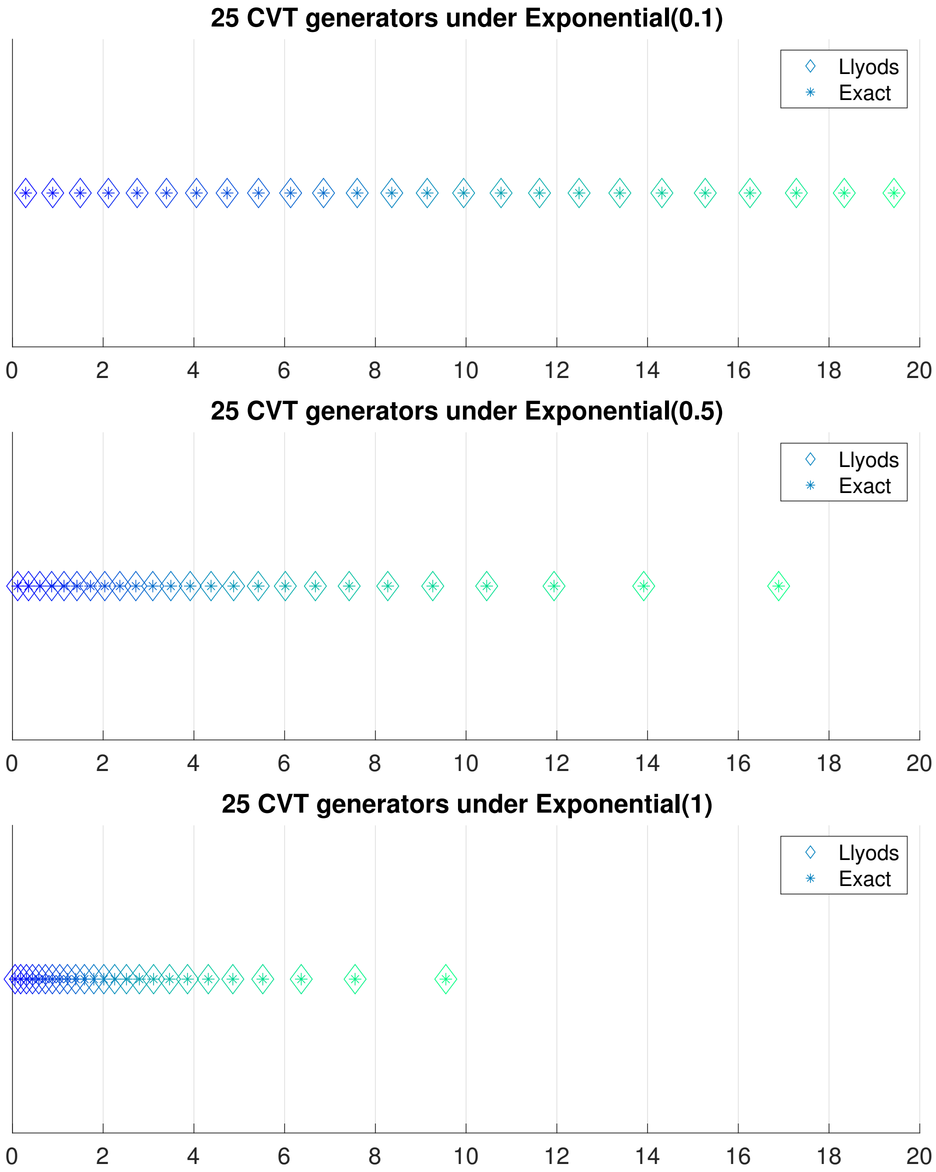


Figure 4.6: Centroidal Voronoi tessellations from 25 generators in $\Omega = [0, 20]$ under Exponential distribution with the rate parameters 0.1 (top), 0.5 (middle) and 1 (bottom).

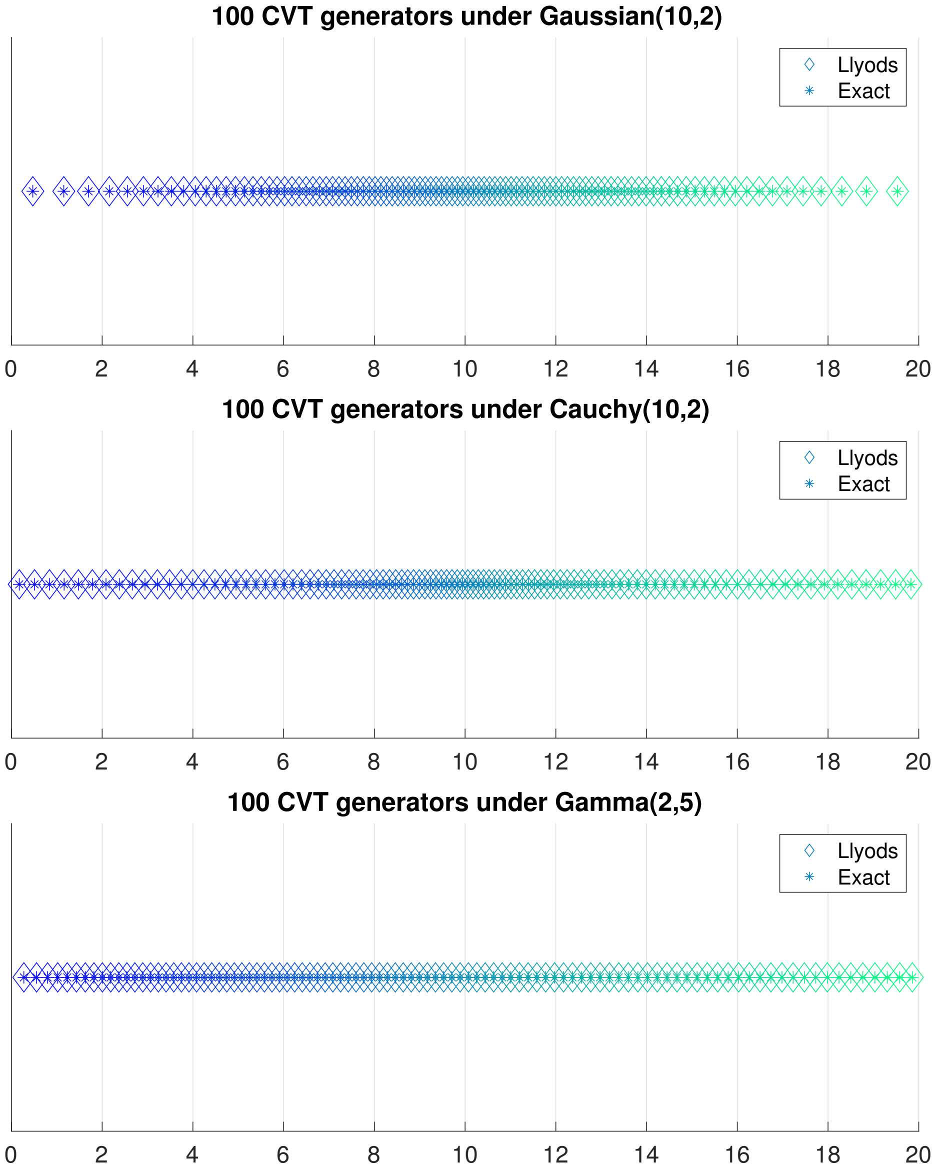


Figure 4.7: Centroidal Voronoi tessellations from 100 generators in $\Omega = [0, 20]$ under (top) Gaussian distribution with mean 10 and standard deviation 2, (middle) Cauchy with parameters $x_0 = 10,\ \gamma = 2$ and Gamma distribution with shape parameter 2 and rate parameter 5.

To summarize, in this section, we looked into ways to compute the Centroidal Voronoi Tessellations in one-dimensional spaces. The first approach is the widely employed Lloyd's algorithm, and the second approach is the method that is based on solving a system of non-linear equations from Section 4.2.2. The two methods are then compared through their solutions under different scenarios.

4.3 Computation of CVT in n-D Spaces

4.3.1 Decomposition Method

From looking into computational methods for CVTs in one-dimensional spaces in the previous section, we move to higher dimensions here. In this section, we first propose a simple method to obtain a tessellation in a higher dimensional space from CVTs in 1-D spaces. Then, we present the proof that a tessellation constructed in such a way is also a CVT.

Consider a region $\Omega \subset \mathbb{R}^n,\ n > 1$, and $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$. Let $\rho(.)$ be the probability density function over Ω, and $\rho_i(.)$ be the density function over $\Omega_i, \forall i \in I_n = \{1, 2, \ldots, n\}$. That is, $\rho(.)$ is the joint density, and $\{\rho_i(.)\}_{i \in I_n}$ are the marginal densities.

Let the number of generators in a CVT of Ω_i under $\rho_i(.)$ be N_i, and let the number of generators in a CVT of Ω under $\rho(.)$ be N, where $N = N_1 \times \ldots \times N_n$. Denote a CVT in Ω_i as $\{\mathbf{z}_i^*, \mathbf{V}_{\mathbf{z}_i^*}\}$. Here, $\mathbf{z}_i^* = \{z_{i,j}^*\}_{j \in I_{N_i}}$ is the set of all the centroids of the CVT in Ω_i, and $\mathbf{V}_{\mathbf{z}_i^*} = \{V_{z_{i,j}^*}\}_{j \in I_{N_i}}$ is the set of their respective Voronoi regions. Similarly, denote a CVT in Ω as $\{\mathbf{z}^*, \mathbf{V}_{\mathbf{z}^*}\}$, where $\mathbf{z}^* = \{z_k^*\}_{k \in I_N}$ denotes the centroids, $\mathbf{V}_{\mathbf{z}^*} = \{V_{z_k^*}\}_{k \in I_N}$ denotes their corresponding Voronoi regions.

The set of centroids in Ω can be given as a matrix $\mathbf{z}^* = [z_1^*\ z_2^* \ldots z_N^*] \in \mathbb{R}^{n \times N}$, where each matrix column $\{z_k^*\}_{k \in I_N} \in \mathbb{R}^n$ denotes a centroid in $\mathbb{R}^n$. Similarly, the set of centroids in Ω_i is given as a vector $\mathbf{z}_i^* = [z_{i,1}^*\ z_{i,2}^* \ldots z_{i,N_i}^*] \in \mathbb{R}^{N_i}$, where each element is a centroid in $\mathbb{R}$. Additionally, note that $V_{z_k^*} \subset \mathbb{R}^n$ while $V_{z_{i,j}^*} \subset \mathbb{R}$.

Let $I_{n\times N}$ denote the matrix containing all combinations of vectors $I_{N_i}, \forall i \in I_n$. That is, define the k^{th} of the N columns of $I_{n\times N}$ as $\{[j_1\ j_2 \ldots\ j_n]^T$such that $j_i \in I_{N_i},\ i \in I_n\}$ For example, if $n = 2, N_1 = 2, N_2 = 3$, then $N = 2 \times 3 = 6$ and

$$I_{n\times N} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \tag{4.6}$$

With all the notations defined, we now present a straightforward method of constructing a tessellation in Ω from CVTs in $\Omega_i' s$:

```
**Tessellation construction in $\Omega$**
For each dimension $i \in I_n$, construct a CVT in $\Omega_i$: $\{\mathbf{z}_i^*, \mathbf{V}_{\mathbf{z}_i^*}\}$
Obtain the $n$ coordinates of each centroid in $\Omega$ and its Voronoi region as:
    $\forall k \in I_N$ :
        $\forall i \in I_n$ :
            $z_k^*(i) = z^*_{I_{n\times N}(i,k)}$
            $V_{z_k^*} = V_{z^*_{I_{n\times N}(i,k)}}$
The set of all the centroids $\{z_k^*\}_{k\in I_N}$ and their Voronoi regions $\{V_{z_k^*}\}_{k\in I_N}$ make the tessellation in $\Omega$: $\{\mathbf{z}^*, \mathbf{V}_{\mathbf{z}^*}\}$
```

Having obtained the tessellation, we show in the following theorem that $\{\mathbf{z}^*, \mathbf{V}_{\mathbf{z}^*}\}$ constructed from $\{\mathbf{z}_i^*, \mathbf{V}_{\mathbf{z}_i^*}\}_{i\in I_n}$ is a CVT in Ω, which was published in [73].

Theorem: Let $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n \subset \mathbb{R}^n,\ n > 1$. Let $\rho(.)$ be the joint density function over Ω, and $\rho_i(.)$ be the density function over Ω_i, $\forall i \in I_n$. If $\rho(x_1, \ldots, x_n) = \prod_{i\in I_n} \rho_i(x_i)$, then $\forall i \in I_n$, $\forall k \in I_N$ and $k_i = I_{n\times N}(i,k)$, we have:

$$z_k^* = (z^*_{1,k_1},\ z^*_{2,k_2}, \ldots, z^*_{n,k_n}) \tag{4.7}$$

$$V_{z_k^*} = V_{z^*_{1,k_1}} \times V_{z^*_{2,k_2}} \times \ldots \times V_{z^*_{n,k_n}} \tag{4.8}$$

Proof:

Consider $x \in V_{z^*_{1,k_1}} \times \ldots \times V_{z^*_{n,k_n}}$, $\forall i \in I_n$, and denote $x = (x_1, \ldots, x_n)$. Because $V_{z^*_{i,k_i}}$ is the Voronoi region of z^*_{i,k_i}, $\forall i \in I_n$, we have for any $j_i \in I_{N_i}$:

$$||x_i - z^*_{i,k_i}||_2 \leq ||x_i - z^*_{i,j_i}||_2 \implies (x_i - z^*_{i,k_i})^2 \leq (x_i - z^*_{i,j_i})^2 \tag{4.9}$$

Summing (4.9) $\forall i \in I_n$,

$$\begin{aligned} (x_1 - z^*_{1,k_1})^2 + \ldots + (x_n - z^*_{n,k_n})^2 \leq & \ (x_1 - z^*_{1,j_1})^2 + \ldots + (x_n - z^*_{n,j_n})^2 \\ \implies \sqrt{((x_1 - z^*_{1,k_1})^2 + \ldots + (x_n - z^*_{n,k_n})^2)} \leq & \sqrt{((x_1 - z^*_{1,j_1})^2 + \ldots + (x_n - z^*_{n,j_n})^2)} \end{aligned}$$

Let $\hat{z}^*_k = (z^*_{1,k_1}, z^*_{2,k_2}, \ldots, z^*_{n,k_n})$, then:

$$||x - \hat{z}^*_k||_2 \quad \leq \quad ||x - z^*_j|| \implies V_{\hat{z}^*_k} = V_{z^*_{1,k_1}} \times \ldots \times V_{z^*_{n,k_n}} \tag{4.10}$$

That is, $V_{z^*_{1,k_1}} \times \ldots \times V_{z^*_{n,k_1}}$ is the Voronoi region of $\hat{z}^*_k$.

Consider the i^{th} coordinate of $\hat{z}^*_k$. Since z^*_{i,k_i} is the centroid of $V_{z^*_{i,k_i}}$, by definition of centroid, we have:

$$\begin{aligned} z^*_{i,k_i} &= \frac{\int_{V_{z^*_{i,k_i}}} x_i \rho_1(x_i) dx_i}{\int_{V_{z^*_{i,k_i}}} \rho_n(x_i) dx_i} \\ &= \frac{\int_{V_{z^*_{1,k_1}}} \rho_1(x_1) dx_1}{\int_{V_{z^*_{1,k_1}}} \rho_1(x_1) dx_1} \times \ldots \times \frac{\int_{V_{z^*_{i,k_i}}} x_i \rho_i(x_i) dx_i}{\int_{V_{z^*_{i,k_i}}} \rho_i(x_i) dx_i} \times \ldots \frac{\int_{V_{z^*_{n,k_n}}} \rho_n(x_n) dx_n}{\int_{V_{z^*_{n,k_n}}} \rho_n(x_n) dx_n} \end{aligned} \tag{4.11}$$

Because the events in Ω_i are independent of those in Ω_j, $\forall i \neq j, \ i, j \in I_n$, we have $\rho(x_1, \ldots, x_n) = \rho_1(x_1) \times \ldots \times \rho_n(x_n)$. Substituting this relation in (4.11) implies:

$$z^*_{i,k_i} = \frac{\int_{V_{z^*_{1,k_1}}} \dots \int_{V_{z^*_{n,k_n}}} x_i \rho(x_1, \dots, x_n) dx_1 \dots dx_n}{\int_{V_{z^*_{1,k_1}}} \dots \int_{V_{z^*_{n,k}}} \rho(x_1, \dots, x_n) dx_1 \dots dx_n}$$

which, by (4.1), is the i^{th} coordinate of the k^{th} of the N centroids – z^*_k – in Ω with density $\rho(.)$. That is, $z^*_k(i) = z^*_{i,k_i}$. Since this holds for all $i \in I_n$ coordinates, we have $z^*_k = \hat{z}^*_k = (z^*_{1,k_1}, z^*_{2,k_2}, \dots, z^*_{n,k_n})$, and hence proving (4.7). On the other hand, since $V_{z^*_{1,k_1}} \times \dots \times V_{z^*_{n,k_1}}$ is the Voronoi region of $\hat{z}^*_k$ from (4.10), and $z^*_k = \hat{z}^*_k$, we have $V_{z^*_{1,k_1}} \times \dots \times V_{z^*_{n,k_1}}$ is the Voronoi region of z^*_k, hence proving (4.7). Since this holds for all N centroids in Ω, we have that $\mathbf{V}_{\mathbf{z}^*} = \{V_{z^*_k}\}_{k \in I_N}$ is the Voronoi partition of $\mathbf{z}^* = \{z^*_k\}_{k \in I_N}$.

Having shown that $V_{z^*_{1,k_1}} \times \dots \times V_{z^*_{n,k_1}}$ is the Voronoi region of $\hat{z}^*_k$ and that $\hat{z}^*_k$ is its centroid, we now show that $\{V_{z^*_k}\}_{k \in I_N}$ is a tessellation in Ω. Because $\mathbf{V}_{\mathbf{z}^*_i}$ is a tessellation in Ω_i, $\forall i \in I_n$, we have:

$$\bigcup_{k \in I_N} \prod_{i \in I_n} V_{z^*_{i,k_i}} = \prod_{i \in I_n} \bigcup_{j_i \in I_{N_i}} V_{z^*_{i,j_i}} = \prod_{i \in I_n} \Omega_i = \Omega$$

$$\begin{aligned} \forall\ k \neq l,\ V_{z^*_k} \bigcap V_{z^*_l} &= \prod_{i \in I_n} V_{z^*_{i,k_i}} \bigcap \prod_{i \in I_n} V_{z^*_{i,l_i}} \\ &= \prod_{i \in I_n} (V_{z^*_{i,k_i}} \cap V_{z^*_{i,l_i}}) = \emptyset \end{aligned}$$

Hence, $\{V_{z^*_k}\}_{k \in I_N}$ is a tessellation in Ω.

Since $\mathbf{z}^*$ are the centroids of partitions $\mathbf{V}_{\mathbf{z}^*}$, $\mathbf{V}_{\mathbf{z}^*} = \{V_{z^*_k}\}_{k \in I_N}$ are the Voronoi regions of $\mathbf{z}^*$, and $\{V_{z^*_k}\}_{k \in I_N}$ is a tessellation in Ω, we have that $\{\mathbf{z}^*, \mathbf{V}_{\mathbf{z}^*}\}$ is a CVT in Ω with density $\rho(.)$.

□

Consider $\Omega = [0,20] \times [0,10]$ in $\mathbb{R}^2$ with density $\rho(.) \sim \mathcal{N}(\mu, \Sigma)$, where $\mu = (12,7)$ and $\Sigma = [4 \quad 0; 0 \quad 1]$. Denote the CVT as $\{\mathbf{z}^*, \mathbf{V}_{\mathbf{z}^*}\}$, where $\mathbf{z}^* = (z_1^*, \ldots, z_6^*)$, and $z_k^* \in \mathbb{R}^2, \forall k \in I_6 = \{1, \ldots, 6\}$.

On the other hand, let $N_1 = 3$ and $N_2 = 2$. Consider the unique CVT in $\Omega_1 = [0,20]$ for $\rho_1(.) \sim \mathcal{N}(12,4)$, which is denoted $\{\mathbf{z}_1^*, \mathbf{V}_{\mathbf{z}_1^*}\}$. Note $\mathbf{z}_1^* = (z_{1,1}^*, z_{1,2}^*, z_{1,3}^*)$, and $z_{1,j}^* \in \mathbb{R}, \forall j \in I_3$. Similarly, consider the unique CVT in $\Omega_2 = [0,10]$ for $\rho_1(.) \sim \mathcal{N}(7,1)$, which is denoted $\{\mathbf{z}_2^*, \mathbf{V}_{\mathbf{z}_2^*}\}$. Note $\mathbf{z}_2^* = (z_{2,1}^*, z_{1,2}^*)$, and $z_{1,j}^* \in \mathbb{R}, \forall j \in I_2$. These generators are shown in Fig 4.8: the region Ω_1 and the CVT generators in it are shown in pink, and the region Ω_2 and the CVT generators in it are shown in blue.

Suppose $N_i = N$ and $N_j = 1, \ \forall j \neq i$ for some $i, j \in I_n, \ n > 1$. This corresponds to the case where all the centroids in $\mathbb{R}^n$ are "aligned" along the i^{th} dimension, that is, the centroids only differ in their i^{th} coordinate. In such a case, the decomposition of obtaining the CVT in $\mathbb{R}^n$ into n CVTs in $\mathbb{R}$ is equivalent to obtaining a CVT in $\mathbb{R}$ with all other dimensions held constant. While the proposed method of obtaining a CVT in higher dimensions by employing a combination of CVTs in $\mathbb{R}$ does not result in every possible CVT of the higher dimension under the given conditions, we are guaranteed to obtain at least one of them in a straightforward manner with minimal computation.

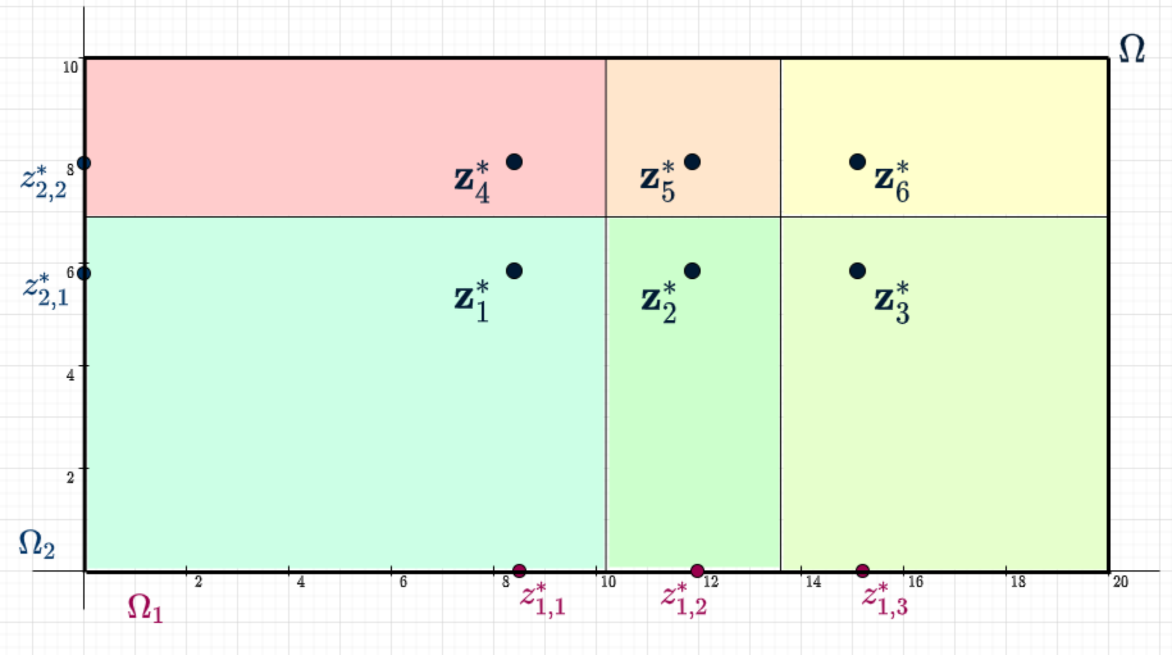


Figure 4.8: CVT in Ω with density $\mathcal{N}(\mu, \Sigma)$ where $\mu = [12; 7]$ and $\Sigma = [4 \;\; 0; 0 \;\; 1]$.

4.3.2 Numerical Results

In this Section, we present a set of numerical results to demonstrate the ease of extension in higher dimensions through the time required to compute the CVT and its energy. Additionally, for 2 and 3 dimensional spaces we also present the tessellations graphically. To obtain the CVTs in 1-D spaces, one can employ Lloyd's algorithm or solve the system of nonlinear equations; the latter being more desirable when $N_i's$ are low, in the order of 10s.

We compare the proposed decomposition method to obtain a CVT in higher dimensional space with a popular probabilistic method – MacQueen's which was introduced in [50]. The elegant MacQueen's algorithm requires Monte Carlo sampling for initialization and randomization in every iteration to compute the centroids in a given space under certain density. Its performance vastly depends on the Monte Carlo samples, and accordingly on the method employed to generate such samples. The authors in [39] compare their proposed method to obtain a CVT with MacQueen's method for 1-D spaces and employ rejection sampling to obtain Monte Carlo samples. However, the rejection method does not readily scale to higher dimensions; its high rejection rate makes it extremely inefficient to generate Monte Carlo samples according to a desired distribution in higher dimensional spaces [60]. Since our focus is CVT in high-dimensional spaces, we employ Metropolis-Hasting algorithm [60] to obtain the Monte Carlo samples for MacQueen's method. The termination criteria we employ for implementation of MacQueen's is the change in the norm of the centroids over each update; if the norm changes less than 10^{-6} we terminate the MacQueen's iterations.

Additionally, [39] compares their results with those from MacQueen's through the CVT energy for 1-D spaces and graphically for two-dimensional spaces. While obtaining the Voronoi partitions is very straightforward in 1-D spaces, its computation in higher dimensions is difficult. The requirement of the knowledge of Voronoi cell boundary of each centroid along with the computation of the cell energy, which involves the computation of "area" of arbitrary high-dimensional polygons, makes the computation of the tessellation energy in high dimensional spaces very difficult. Therefore, following [39] we compare our results with those from MacQueen's visually in two and three dimensions, and through computation time in higher dimensional spaces.

Consider $\Omega = [-1, 1] \times [-1, 1]$. Following the cases taken up in [39], we let the density function over Ω be e^{-10x^2}. The CVT with 16 centroids in each dimension, obtained using the proposed decomposition method and the MacQueen's method are shown in Fig. 4.9. As designed and expected, the tessellation from the decomposition has a well drawn out grid-like structure with the intensity of centroids being higher in the center of Ω. The resulting tessellation is a CVT with (low) energy of 2.9×10^{-4} and was obtained in a computational time as less as 13.18 minutes in an ordinary laptop – MacBook Air 2015 with 2.2 GHz Dual-Core Intel Core i7.

The next case of demonstration is in the region $\Omega = [0, 20] \times [0, 20]$ with density $\mathcal{N}(\mu, \Sigma)$, where $\mu = (5, 6.5)$ and $\Sigma = [2 \quad 0; 0 \quad 1]$. The resulting CVTs with 3000 centroids, obtained using the proposed method and the MacQueen's method are shown in Fig. 4.10. While we can observe a similar CVT pattern from the two methods, we can see the laid-out grid-like tessellation from the proposed method. Additionally, notice the lack of the CVT energy in the MacQueen's method.

Moving to 3 dimensional spaces, we consider $\Omega = [0, 10] \times [0, 10] \times [0, 10]$ with density $\mathcal{N}(\mu, \Sigma)$, where $\mu = (6, 5, 3.5)$ and $\Sigma = [2 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 1]$. The resulting well-aligned grid-like CVT with 16 centroids in each dimension, with energy 0.1616, is shown in Fig. 4.11. In contrast to the aligned CVT, the solution tessellation from MacQueen's under the same conditions is also shown in Fig. 4.11.

The convergence of the MacQueen's iterates to a CVT is not guaranteed for all conditions, and due to its probabilistic nature, its performance vastly depends on the Monte Carlo samples. Since we employ the deterministic Lloyd's method to compute the CVTs in one dimensional spaces for which convergence is proven, the proposed algorithm converges to a CVT without the performance being vastly tied to the initial samples. Moreover, our proposed decomposition method also provides deep insight

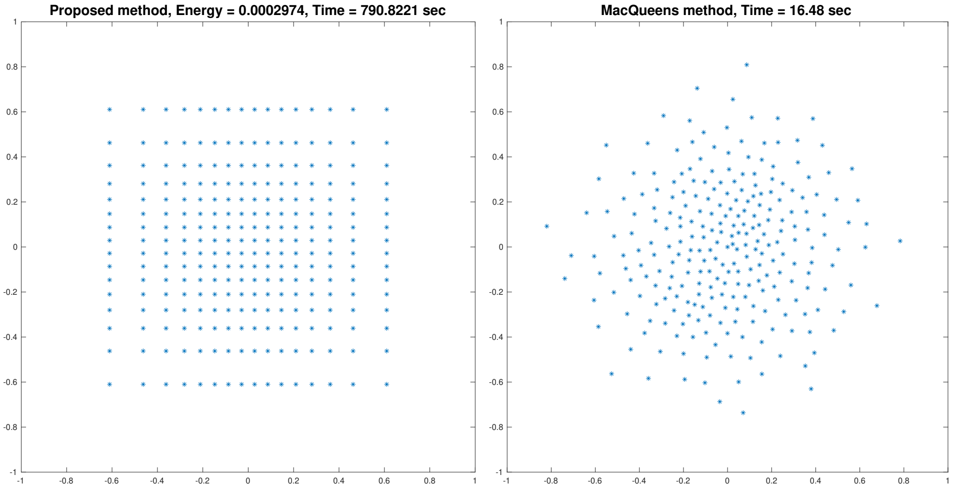


Figure 4.9: CVT of 256 centroids under $\rho_1(x_1) = \rho_2(x_2) = e^{-10x^2}$.

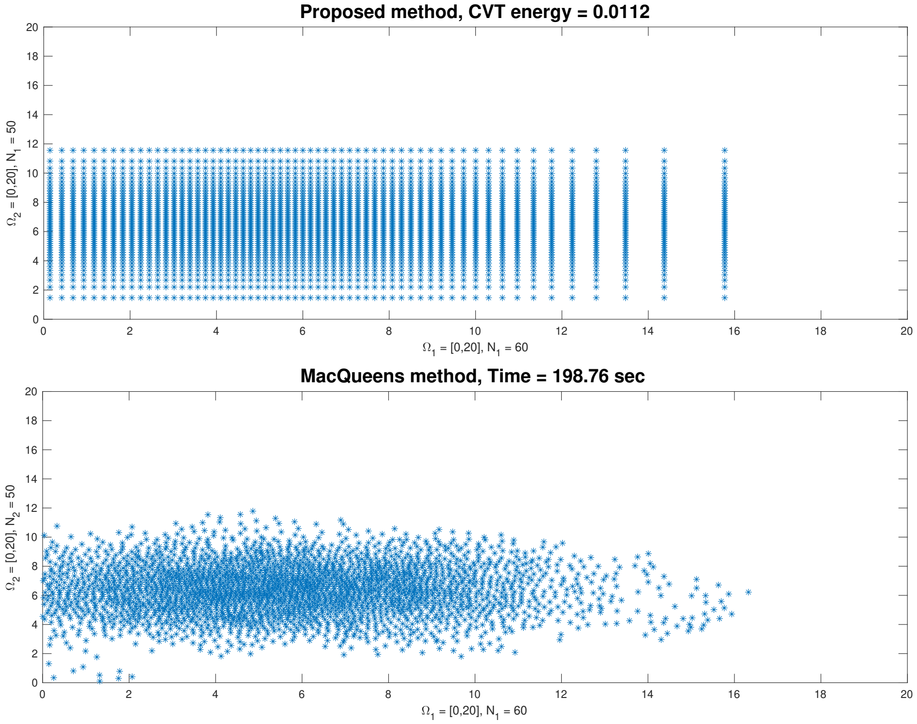
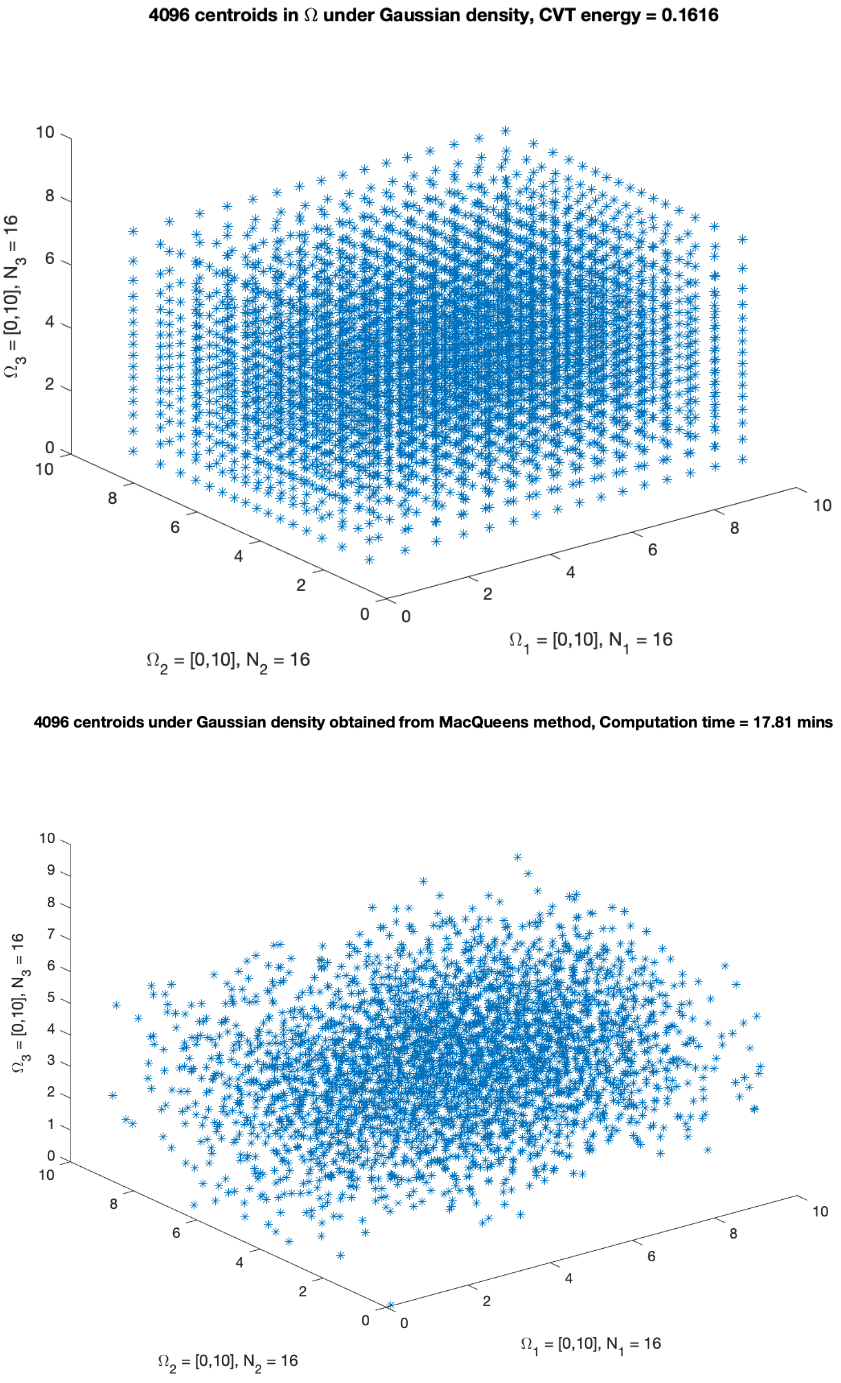


Figure 4.10: CVT of 3000 centroids with Gaussian density.


4096 centroids in Ω under Gaussian density, CVT energy = 0.1616

4096 centroids under Gaussian density obtained from MacQueens method, Computation time = 17.81 mins

Figure 4.11: Scalability and generalizability to any density: CVT of 4096 centroids under a Gaussian density.

into the tessellation (and hence the application at hand that requires the CVT) by allowing the evaluation of the quality of all the solutions (CVTs) through their energies. This is because we decompose our high dimensional spaces into a series of 1-D spaces for which the Voronoi partitions, and hence the tessellation energy, are readily obtained. However, since Lloyd's computes the mass centroids at each iteration while in MacQueen's the only characterization of the density function is through the Monte Carlo samples, our results have higher computation time than MacQueen's. We must note here that while we employ Lloyd's to obtain CVTs in 1-D spaces, one could employ MacQueen's or other CVT computation methods for the decomposed CVTs in 1-D.

One of the areas where higher dimensional CVTs have found an application is in the field of evolutionary optimization. Recently introduced, MAP-elites [59] is an algorithm that illuminates search spaces in evolutionary optimization, allowing researchers to understand how interesting attributes of solutions combine to affect performance. To scale up the MAP-elites algorithm, the authors in [78] employ CVTs, and therein, following [39], MacQueen's method is used to obtain the CVTs and show the sufficiency of using 5000 centroids. In line with their result, we keep the total number of centroids in our results of high-dimensional CVTs, around the same. Similar to illuminating search spaces using MAP-elites, CVTs have been useful in the field of fluid dynamics and control through their role in finite-element analysis for discretization in space dimensions [11]. While the authors in [11] use CVT-based clustering for reduced-order modeling under uniform density, one could employ CVTs to model the underlying space according to candidate density functions. Specifically in such space discretization applications, the proposed method allows us a clear insight into the underlying solution (or search) space by evaluating all possible grid-like CVTs under any density function. For example, by varying the number of centroids per dimension we can obtain a number of different tessellations, and although this would not be exhaustive and there will still be more (non grid-like) CVTs under the same conditions, we can get a better idea about the solutions at different points in the underlying space through the energies of all the grid-like CVTs obtained from the proposed method.

In our last block of presentation of numerical results, we consider dimensions higher than 3 and vary $N_i, \forall i \in I_n$ such that $N = \prod_i N_i$ is around 5000. The corresponding results are given in Table 4.1 where we see that the computation time decreases with the increase in dimension n. This is because with increasing n, we decrease N_i to keep N around 5000. Hence, the computation of CVT in 1-D spaces with fewer centroids is faster. The low energy of all the tessellations is also worth noting. On the other hand, the computation time required to compute the centroids using MacQueen's method are lower but the solutions are opaque since it is difficult to evaluate their quality through their tessellation energy.

While the proposed method allows us to obtain a number of CVTs and their energies in a straightforward fashion, it suffers from the curse of dimensionality. Considering 32 dimensions and 2 centroids per dimension, the problem requirement scales to a total of 2^{32} centroids. While the maximum array size allocated varies by the program and the software, such exponential growth in the number of centroids practically limits the proposed method to under 30 dimensions. However, the applications where the proposed decomposition method would be most beneficial do not require dimensions in hundreds. For example, the number of features in MAP-elites are typically less than 10; in [75] the authors consider a four-dimensional problem. The exploration of the solution space using finite element analysis is in space dimensions. In such applications the proposed method provides insightful tessellations at various markers in the solution space even for finer discretizations.

Table 4.1: Proposed method for $\Omega_i = [-1, 1], \forall i \in I_n$ under e^{-10x^2}. Note the absence of CVT energy from MacQueen's.

n	$N = \prod_i N_i$	CVT energy $\times 10^{-3}$	Time (min)	MacQueen's time (min)
4	$4096 = 8^4$	0.68	6.174	2.644
5	$4096 = 4^3 \times 8^2$	1.2	4.248	1.433
8	$6561 = 3^8$	0.74	0.641	1.556
12	$4096 = 2^{12}$	0.21	0.480	1.425

Chapter 5

Resource Allocation using Centroidal Voronoi Tessellations

In the study and exploration CVTs in the previous Chapter 4, we developed a computational method to obtain CVTs in high-dimensional spaces. In the process, we observed an advantage that one-dimensional CVTs would bring to the resource allocation problem: A simple information structure that is sufficient to obtain the CVT (a minima).

To elaborate on the advantage, let us first formalize the resource allocation problem following [44]:

$$\begin{aligned} \min_{z_i \in \mathbb{R}^n} \frac{1}{N} \sum_{i \in I_N} f_i(z_i) \\ \text{such that,} \quad \sum_{i \in I_N} z_i = r \end{aligned} \tag{5.1}$$

In the resource allocation problem (5.1), r amount of resources is to be allocated among N agents while minimizing the sum of their individual costs $\{f_i\}_{i \in I_N}$. Seemingly trivial and complex simultaneously, the nature of (5.1) can be broken down into the information structure in the group of agents, the separability of the objective function, and the global constraint. While each agent can minimize the cost function without any dependance on other agents, the global constraint is imposed on the team, hence making the team information structure a significant aspect.

Like most of the work on the resource allocation problem, the authors in [44] assume the individual cost functions $\{f_i\}_{i \in I_N}$, to be convex. In the case where the cost functions are differentiable, they propose a gradient descent consensus algorithm. And when the cost functions are not necessarily differentiable, they present a sub-gradient based algorithm. While they let the team information structure be dynamic, they impose reasonable mild conditions on the team information structure like connectedness, and start at an initial feasible condition. However, it is worth noting that the amount of resource to be allocated, r, is fixed over the iterations; the agents move along the feasible solutions to only minimize the cost function in (5.1).

While the resource allocation problem (5.1) has received attention and contributions from the community, most of the works, as noted in Section 1.2, assume the allocation amount to be fixed, and they start at a feasible constraint, [83]. In this work, we work towards the resource allocation problem for dynamically changing the allocation amount under the assumption that the amount to be allocated (or shared) is known by the all the agents beforehand.

Our approach to the problem involves Centroidal Voronoi tessellations (CVTs), and accordingly we begin with our motivation for the same in Section 5.1 and solve the static resource allocation problem in Section 5.2. We move to the dynamic (time-varying) resource allocation problem and propose a decentralized solution in Section 5.3. Finally, in Section 5.4 we demonstrate the developed decentralized dynamic resource allocation method on a demand-response problem of power allocation in a group of building loads.

5.1 Motivation

Recall the CVT minimization problem from Section 4.1.1:

$$\min_{z_i} \sum_{i \in I_N} \int_{x \in V_i} \rho(x) ||x - z_i||^2 dx$$

Including the resource allocation constraint from (5.1), we have:

$$\begin{gathered}\min_{z_i} \sum_{i \in I_N} \int_{x \in V_i} \rho(x) ||x - z_i||^2 dx \\ \text{subject to} \\ \sum_{i \in I_N} z_i = r \end{gathered} \tag{5.2}$$

Comparing (5.2) with the resource allocation problem (5.1) we see that the individual objective functions from (5.2) are $\{\int_{x \in V_i} \rho(x) ||x - z_i||^2 dx\}_{i \in I_N}$. While the objective function in (5.2) is separable, a common (global) distribution $\rho(.)$ governs all the individual objective functions. Therefore, this enables the embedding of a desired aggregate behavior in the team through such distributions; the desired aggregate behavior can arise from modeling individual preferences or from an external global trendsetting factor depending on the application at hand.

Additionally, as seen in Section 4.2.1, CVTs obtained through Lloyd's algorithm only require communication with neighbors. In 1-D spaces, it translates to a line communication graph. To grasp the advantage of CVTs in simple communication networks, we review and formalize certain basic concepts from network theory.

5.1.1 Network theory Basics

Let $\mathcal{Z}$ denote the undirected resource network graph with vertex set $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ (where z_i is the resource allocated to the i^{th} agent in I_N) and edge set $\mathcal{E}_Z$. The set of neighbors of i according to the resource network graph, or resource neighbors in short, is given by [48]:

$$\mathcal{N}_{\mathcal{Z}_i} = \{j \in I_N : z_k < z_j < z_i,\ \forall j, k \in I_N\} \cup \{j \in I_N : z_i < z_j < z_k,\ \forall j, k \in I_N\} \quad (5.3)$$

If $j \in \mathcal{N}_{\mathcal{Z}_i}$, then the agents i and j are resource neighbors and there exists an edge between them. Hence, the pair $\{z_i, z_j\} \in \mathcal{E}_Z$. Moreover, if a pair $\{z_i, z_j\}$ is in the edge set, then the agents i and j are neighbors. And, since $\mathcal{Z}$ is an undirected graph, if $j \in \mathcal{N}_{\mathcal{Z}_i}$ then $i \in \mathcal{N}_{\mathcal{Z}_j}$, and vice versa. Summarizing mathematically, by definition, we have the following:

$$j \in \mathcal{N}_{\mathcal{Z}_i} \iff \{z_i, z_j\} \in \mathcal{E}_Z \iff i \in \mathcal{N}_{\mathcal{Z}_j} \quad (5.4)$$

Let $\mathcal{C}$ denote the communication network graph among all the agents in I_N. The vertex set is I_N and denote the edge set by $\mathcal{E}_C$. Without loss of generality, let the element of communication be their respective resource quantities $\{z_i\}_{i \in I_N}$. We note here that the communication network graph need not be undirected. If an agent i communicates its resource position z_i to an agent j, then j need not communicate back z_j. For simplicity, we assume that $\mathcal{C}$ is undirected. If the agents i and j are communication neighbors, then agent i is aware of z_j and agent j is aware of z_i. Accordingly, by definition, $\{i, j\} \in \mathcal{E}_C \iff i \in \mathcal{N}_{\mathcal{C}_j}$ and $j \in \mathcal{N}_{\mathcal{C}_i}$, where $\mathcal{N}_{\mathcal{C}_i}$ is the set of communication neighbors of the agent i.

For a quick demonstration, consider an example of 5 agents with resource positions $z_1 = 4.8, z_2 = 2, z_3 = 6, z_4 = 0.5$, and $z_5 = 3$. Fig. 5.1 shows the resource network graph $\mathcal{Z}$ and three different communication graphs $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$.

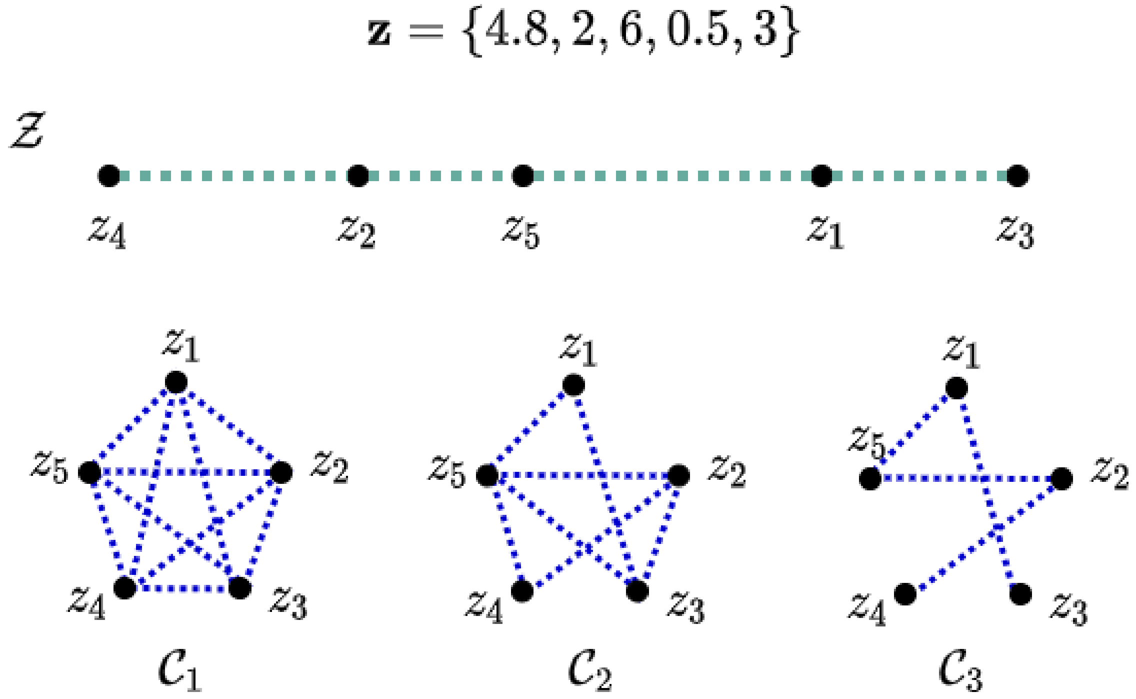


Figure 5.1: Resource network $\mathcal{Z}$ (top graph) and three example communication graphs, $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ (bottom graphs) for 5 agents with resources at $\mathbf{z}$.

The vertices (or nodes) in all the graphs therein are marked by black dots. The edges in the resource graph are shown in dotted green lines and the edges in the communication graphs are shown in dotted blue lines. Following the definition of resource neighbors from (5.3), the resource neighbors of all the agents are:

$$\begin{aligned}
\mathcal{N}_{\mathcal{Z}_1} &= \{3, 5\} \\
\mathcal{N}_{\mathcal{Z}_2} &= \{4, 5\} \\
\mathcal{N}_{\mathcal{Z}_3} &= \{1\} \\
\mathcal{N}_{\mathcal{Z}_4} &= \{2\} \\
\mathcal{N}_{\mathcal{Z}_5} &= \{1, 2\}
\end{aligned}$$

A compact way to indicate the neighbors in a graph is to use the Adjacency matrix, [48]. The adjacency matrix of a graph, $A_{\mathcal{G}}$, is a square matrix whose elements are 1 or 0 corresponding to whether the pair of vertices in question are neighbors in the graph or not. The matrix is of the size $N \times N$, and its rows and columns are labeled by the N vertices. If an agent $i \in \mathcal{N}_{\mathcal{Z}_j}$, then the i^{th} row and j^{th} column of the adjacency matrix is 1; otherwise it is 0. The adjacency matrix is also called the connection matrix. The adjacency matrix of the resource graph $\mathcal{Z}$ from Fig. 5.1 is given by:

$$A_{\mathcal{Z}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Similarly, the adjacency matrices of the communication graphs $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ are:

$$A_{\mathcal{C}_1} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad A_{\mathcal{C}_2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad A_{\mathcal{C}_3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The symmetry in the adjacency matrices across their diagonals is a presentation of the fact that we assume the graphs to be undirected. Notice that the adjacency matrix of the first communication graph, $\mathcal{C}_1$, has all off-diagonal elements as 1, highlighting the fully-connected nature of the graph. As seen in Fig. 5.1, the number of connections decrease as we proceed from $\mathcal{C}_1$ to $\mathcal{C}_3$. This decrease is proportionally seen in their adjacency matrices with the decrease in the number of 1's in the matrices.

Suppose the resource and the communication graphs are the same: $\mathcal{Z} = \mathcal{C}$. Under this assumption, if two agents are resource neighbors, then they are also communication neighbors, and are hence aware of each other's resource positions. Since our region of interest is one-dimensional, the resource network graph is always a line graph, as also seen in Fig. 5.1. That is, each agent i can have at most 2 resource neighbors. However, depending on the communication network graph $\mathcal{C}$, each agent i can be aware of any number of agents' resource position that are also its communication neighbors. The simplest communication graph $\mathcal{C}$ such that $\mathcal{Z} \subset \mathcal{C}$, is $\mathcal{C} = \mathcal{Z}$, and accordingly $\mathcal{C}$ is also a line graph. In the aforementioned example from Fig. 5.1, it is worth noting that the adjacency matrices of $\mathcal{Z}$ and $\mathcal{C}_3$ are the same; $\mathcal{C}_3$ is essentially a line graph.

CVTs in one-dimensional spaces have an inherent line structure in their resource graphs. Consider the top plot in Fig. 4.5. Without loss of generality, let the 5 agents therein be ordered 1 to 5 from left to right, that is, $z_1 < z_2 < \ldots < z_5$. By definition of the resource neighbors from (5.3), we have that the resource neighbor of agent 1 is agent 2, the resource neighbors of agent 2 are agents 1 and 3, and so on.

In the implementation of Lloyd's algorithm, it is evident that each agent requires the resource positions of its resource neighbors for them all to converge to the CVT. Any agent having resource positions of agents that are not its neighbors is redundant, since it is not used in the iterative steps in the algorithm. Therefore, if each agent were to communicate only with its resource neighbors, then all the agents would converge to the CVT through Lloyd's algorithm with minimal communication. That is, one would have obtained the CVT in a decentralized manner. Therefore, one may obtain CVTs in a decentralized manner using one of the simplest communication graphs: a line graph that is also the same as the resource graph.

To summarize, CVTs provide a natural way of embedding a desired distribution in the solution, along with obtaining the solution in a straightforward decentralized approach with minimal requirements on the team information structure. In the next Section, we take up the resource allocation problem (5.1) which is a constrained CVT minimization problem, and solve it using the analytical CVT computation method from Section 4.2.2.

5.2 Static Allocation

In this Section we consider the question of how to employ CVTs to solve the resource allocation problem for a fixed amount of resource, and analytically solve it as a constrained CVT minimization problem. The underlying idea of the solution is to pose the generators of the CVT as the resource amounts allocated to the agents, and accordingly, they must sum up to the available amount of resource, r.

Recall the resource allocation constrained CVT minimization problem (5.2) introduced in Section 5.1. For a fixed amount of resource, we solve (5.2) by transforming it to (4.3) through the elimination of the constraint.

One way to eliminate the constraint is to choose the design parameters defining $\rho(x)$ such that the resource allocation constraint is embedded in them. The "free" design parameter to be chosen can then be treated as an unknown along with the N generators

z_i's. Resorting to the SNLE method developed in Section 4.2.2, while we now have $N+1$ number of unknowns, we also have $N+1$ equations. That is, N equations from the definition of mass centroid for the generators, and one resource allocation constraint equation. Accordingly, the solution of this system of $N+1$ nonlinear equations results in the N CVT generators and, more importantly, the design parameter such that the resource allocation constraint is met. We can now formalize this overview of the idea.

Suppose the density function $\rho(.)$ is defined by n_ρ number of parameters: $v = (v_1, v_2, \dots, v_{n_\rho})$. Let $v_k \in v$ be unknown, and all the other parameters defining the density function be known and fixed. To highlight the dependence of the density function on the free parameter v_k, denote the density function as $\rho(x, v_k)$, where x is the support of the considered probability distribution. The N number of nonlinear equations from (4.5) and the resource allocation constraint (5.1) will together constitute the following SNLE with $N+1$ unknowns – $(z_1^c, z_2^c, \dots, z_N^c, v_k)$.

$$
\begin{aligned}
z_1^c =& \frac{\int_a^{\frac{z_1^c+z_2^c}{2}} x\rho(x,v_k)dx}{\int_a^{\frac{z_1^c+z_2^c}{2}} \rho(x,v_k)dx} \\
z_2^c =& \frac{\int_{\frac{z_1^c+z_2^c}{2}}^{\frac{z_2^c+z_3^c}{2}} x\rho(x,v_k)dx}{\int_{\frac{z_1^c+z_2^c}{2}}^{\frac{z_2^c+z_3^c}{2}} \rho(x,v_k)dx} \\
&\vdots \\
z_i^c =& \frac{\int_{\frac{z_{i-1}^c+z_i^c}{2}}^{\frac{z_i^c+z_{i+1}^c}{2}} x\rho(x,v_k)dx}{\int_{\frac{z_{i-1}^c+z_i^c}{2}}^{\frac{z_i^c+z_{i+1}^c}{2}} \rho(x,v_k)dx} \\
&\vdots \\
z_N^c =& \frac{\int_{\frac{z_{N-1}^c+z_N^c}{2}}^{b} x\rho(x,v_k)dx}{\int_{\frac{z_{N-1}^c+z_N^c}{2}}^{b} \rho(x,v_k)dx} \\
&\sum_{i=1}^{N} z_i^c = r \qquad (5.5)
\end{aligned}
$$

The solution of this system of nonlinear equations, $(z_1^c, z_2^c, \ldots, z_N^c, v_k)$, will be such that $(z_1^c, z_2^c, \ldots, z_N^c)$ are the N number of CVT generators in the region $\Omega = [a, b]$ under the density function $\rho(x, \bar{v}_k)$, and such that the sum of the generators is r. While in this method we already obtain the solution of (5.2), this is a centralized approach. The main solution of interest here is the solved design parameter $\bar{v}_k$ which is fed to the Lloyd's algorithm in its initialization step. In such an approach, all the agents can still maintain communication only with their resource neighbors, and since they are all initialized with the same design parameters, all the agents are minimizing the cost function (4.3) under the same specifications to obtain the CVT. Therefore, we achieve the CVT in a decentralized approach through Lloyd's algorithm while also meeting the resource allocation constraint.

While we consider the resource allocation constraint in the described approach, it is worth noting that this approach can be extended to a different constraint in (5.1), where the constraint can be multiplicative or any function with n input arguments to $\mathbb{R}$, in place of the summation constraint therein. Or, one can embed as many constraints as the number of "free" parameters in the density function, and employ the same developed elimination-of-constraint approach to solve the resulting multi-constrained problem.

Implementation aspects: In the process of fixing the other design parameters, it is important to choose them in a way that renders a feasible solution from (5.5). For example, we cannot have $r = 100$ with 5 agents in $\Omega = [0, 10]$ for any density function. In this case, at the very least, Ω should include the mean $r/N = 20$. Another aspect is the scalability of this method. Since obtaining the free design parameter involves solving the system of $N + 1$ equations, the computational complexity increases with N. However, since the communication graph is still a line graph, Lloyd's algorithm is scalable to any N.

We now demonstrate the method with different simulation cases. In Fig. 5.2, the region Ω and the number of agents are fixed as $[0, 100]$ and 50, respectively, and the density is Gaussian. Out of the three examples therein, the top two have the same variance but are required to allocate different amounts of resources – 2500 in the first and 1500 in the second – among the same number of agents. Accordingly, we can observe the allocated resources among all the agents on average are lower in the second case than the first. Moving from the second case to the third, that is, the bottom graph in Fig. 5.2, the variance is increased while keeping all other parameters the same. In all these three cases, the free design parameter is μ – the mean of the Gaussian distribution. The solution of the free parameter obtained from solving the $N+1$ equations from (5.5), is shown in the figures, and is used to initialize the Lloyd's algorithm. The generators obtained from the Lloyd's algorithm and the generators from solving (5.5) are plotted together. We can observe that the two solutions are very

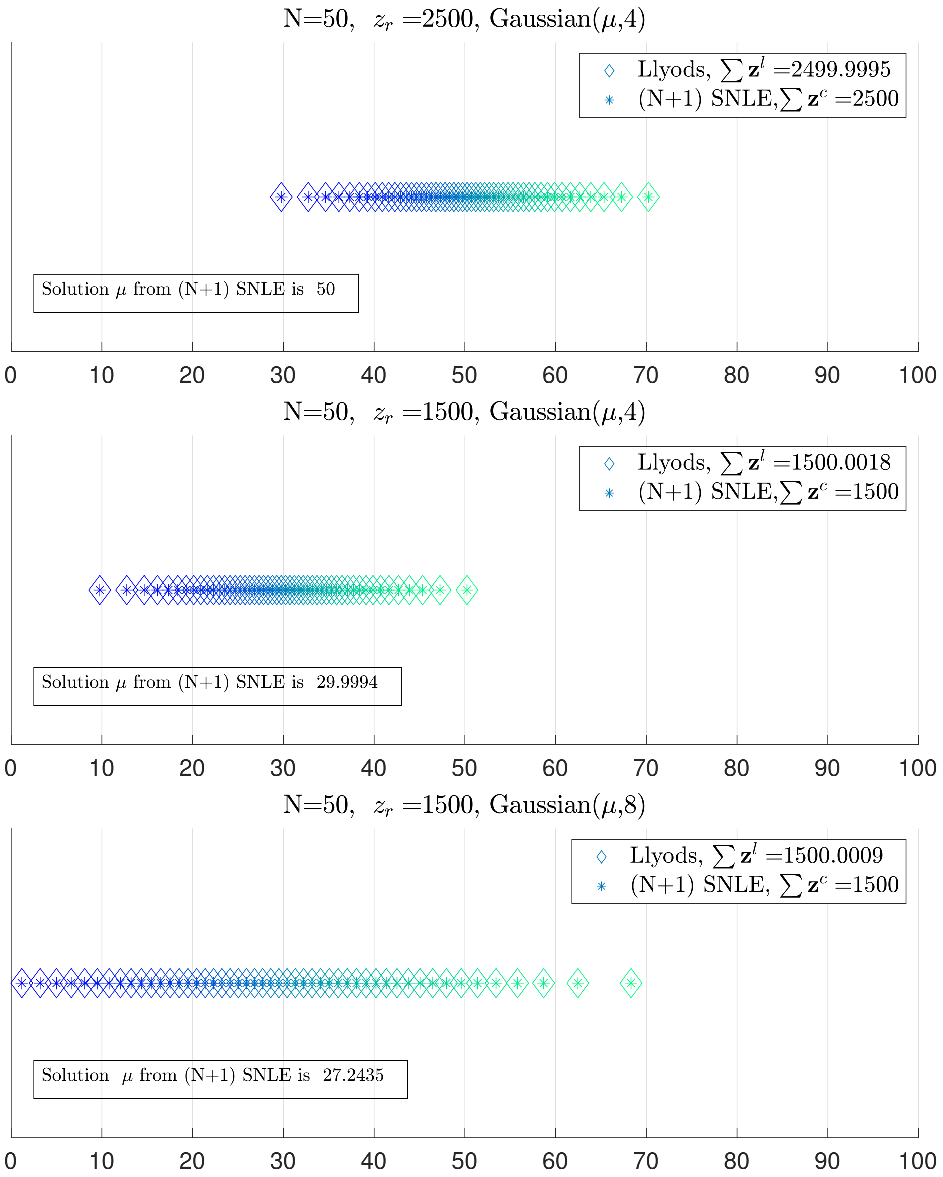


Figure 5.2: Allocation of z_r amount of resource among 50 agents in $\Omega = [0, 100]$ under Gaussian distribution for specified variances – 4 (top and middle) and 8 (bottom). The mean of the distribution is the solution v_k from (5.5).

close to each other. Additionally, both the solutions sum up to z_r with an acceptable error.

Similarly, we present another set of simulations in Fig. 5.3. In the three cases therein, the region, the number of agents and the total amount of resource to be allocated are the same. The difference in the three cases is the underlying distributions – Gamma distribution in the top figure, Exponential in the middle and Gaussian distribution in the bottom figure. Like in Fig. 5.2, the solutions from the two approaches are close to each other and also sum up to z_r.

In the aforementioned two figures, the free parameter in the cases of Gaussian distribution has been the mean μ and the free parameter for Gamma distribution has been the shape parameter k. In Fig. 5.4, we let the standard deviation σ be the free parameter for Gaussian and the scale parameter θ be the free parameter for Gamma distribution.

Like most of the work addressing the resource allocation problem, we presented a solution to the static resource allocation problem, that is, the amount of resource to be allocated is fixed. In the next Section, we consider the resource allocation problem for a time-varying amount of resource in a team of agents.

5.3 Dynamic Resource Allocation

In the previous section, we solved the static resource allocation problem by using the centralized system of nonlinear equations. However, extending the same approach to varying amount of resource-to-be-allocated results in a centralized approach. Therefore, in this Section we focus on developing a decentralized approach to the dynamic resource allocation problem.

Employing the static resource allocation problem as the initialization step, our solution approach to the dynamic resource allocation problem under a Normal distribution involves a one-step update that maintains the dynamic resource allocation constraint

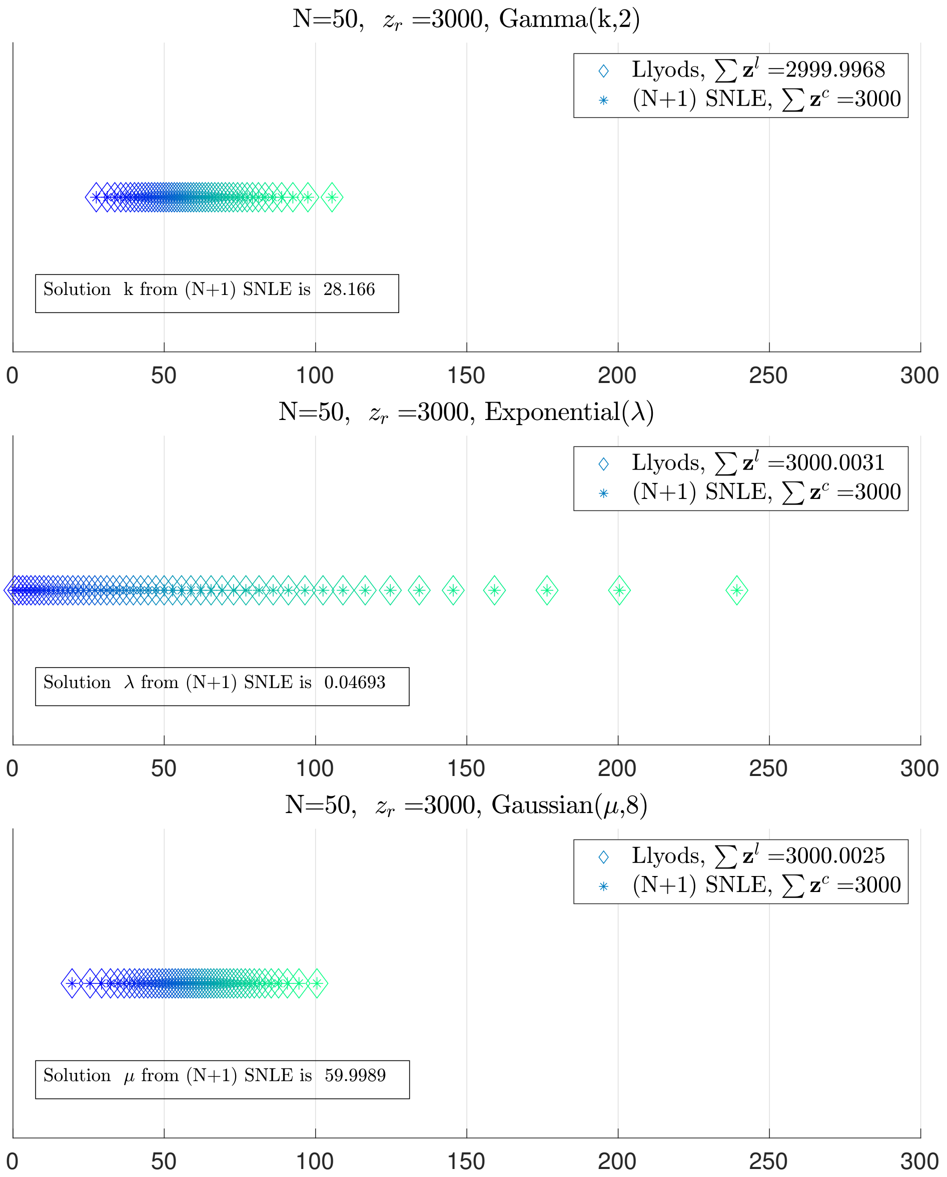


Figure 5.3: Allocation of z_r amount of resource among 50 agents in $\Omega = [0, 300]$ under three different distributions. One of the parameters of the distributions is the solution v_k from (5.5).

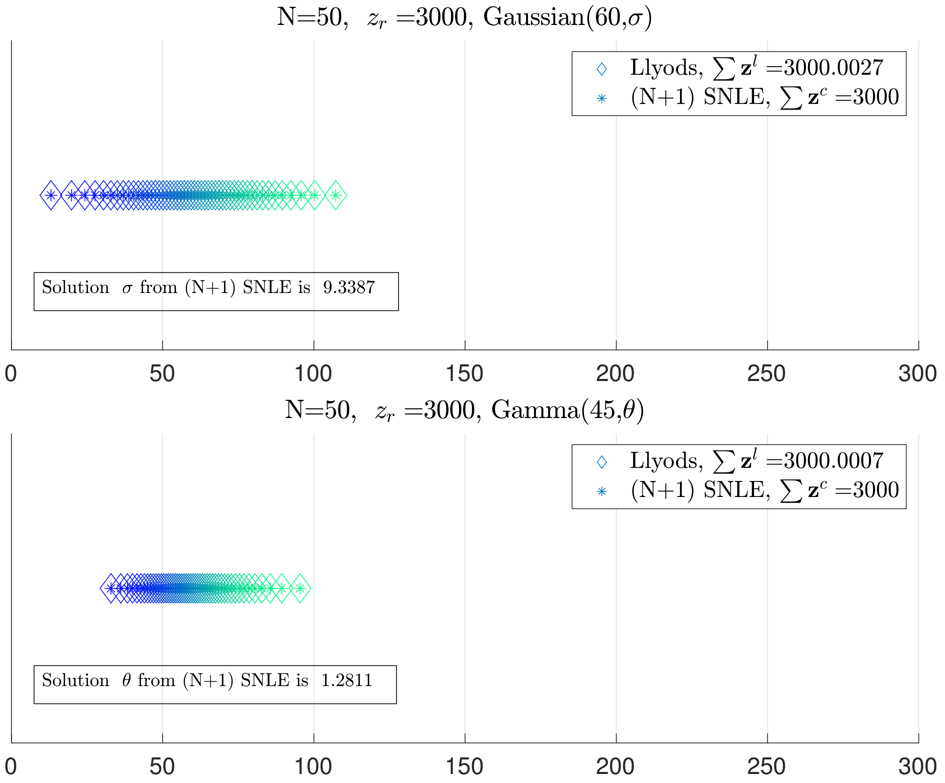


Figure 5.4: Allocation of z_r amount of resource among 50 agents in $\Omega = [0, 300]$ for Gaussian and Gamma distributions with standard deviation σ and scale parameter θ being the free parameters that are obtained as a solution of (5.5).

while preserving the CVT. We employ the following Lemma 5.1 to obtain such a one-step update in Theorem 5.2. Through the design process, we assume that the amount of resource to be allocated among all the agents over the considered time duration is known to all the agents.

Suppose $\rho(.) = \mathcal{N}(\mu, \sigma^2)$ and the "free" parameter is μ. Then we have:

Lemma 5.1. *Suppose at time k, $\{z_i(k)\}_{i \in I_N}$ are the centroids of the CVT in $\Omega \subset \mathbb{R}$ with density $\rho(.) = \mathcal{N}(\mu(k), \sigma^2)$. Then the following relationship holds between the time-updated centroids:*

$$z_i(k+1) - z_i(k) = z_j(k+1) - z_j(k) = \mu(k+1) - \mu(k) = -\delta \tag{5.6}$$

Proof: Let $\mu(k+1) = \mu(k) - \delta$. Because $\{z_i(k)\}_{i \in I_N}$ are the centroids with normal distribution, we have by definition:

$$z_i(k) = \frac{\int_{V_i(k)} x e^{\frac{(x-\mu(k))^2}{2\sigma^2}} dx}{\int_{V_i(k)} e^{\frac{(x-\mu(k))^2}{2\sigma^2}} dx}$$

Similarly, writing out the mass centroid for the next time instant $k+1$ using $\mu(k+1) = \mu(k) - \delta$, we have:

$$z_i(k+1) = \frac{\int_{V_i(k+1)} x e^{\frac{(x-(\mu(k)-\delta))^2}{2\sigma^2}} dx}{\int_{V_i(k)} e^{\frac{(x-(\mu(k)-\delta))^2}{2\sigma^2}} dx} \tag{5.7}$$

Suppose $V_i(k) = [a, b] \subset \Omega$. Consider the change of variables $y = x - \delta$. Then the mass centroids transform as:

$$\begin{aligned}
z_i(k) &= \frac{\int_a^b x e^{\frac{(x-\mu(k))^2}{2\sigma^2}} dx}{\int_a^b e^{\frac{(x-\mu(k))^2}{2\sigma^2}} dx} \\
&= \frac{\int_{a-\delta}^{b-\delta} (y+\delta) e^{\frac{(y+\delta-\mu(k))^2}{2\sigma^2}} dy}{\int_{a-\delta}^{b-\delta} e^{\frac{(y+\delta-\mu(k))^2}{2\sigma^2}} dy} \\
&= \frac{\int_{a-\delta}^{b-\delta} (y+\delta) e^{\frac{(y-(\mu(k)-\delta))^2}{2\sigma^2}} dy}{\int_{a-\delta}^{b-\delta} e^{\frac{(y-(\mu(k)-\delta))^2}{2\sigma^2}} dy} \\
&= \frac{\int_{a-\delta}^{b-\delta} y e^{\frac{(y-(\mu(k)-\delta))^2}{2\sigma^2}} dy + \int_{a-\delta}^{b-\delta} \delta e^{\frac{(y-(\mu(k)-\delta))^2}{2\sigma^2}} dy}{\int_{a-\delta}^{b-\delta} e^{\frac{(y-(\mu(k)-\delta))^2}{2\sigma^2}} dy} \\
&= \frac{\int_{V_i(k+1)} y e^{\frac{(y-\mu(k+1))^2}{2\sigma^2}} dy}{\int_{V_i(k+1)} e^{\frac{(y-\mu(k+1))^2}{2\sigma^2}} dy} + \delta \frac{\int_{V_i(k+1)} e^{\frac{(y-\mu(k+1))^2}{2\sigma^2}} dy}{\int_{V_i(k+1)} e^{\frac{(y-\mu(k+1))^2}{2\sigma^2}} dy} \\
&= z_i(k+1) + \delta \\
\implies z_i(k+1) - z_i(k) &= -\delta \qquad (5.8)
\end{aligned}$$

Since (5.8) holds for all $i \in I_N$ and $\mu(k+1) = \mu(k) - \delta$, we have (5.6) proved.

□

Theorem 5.2. *Suppose the initial condition is a static resource allocation solution at discrete-time k and are at the following conditions: $\{z_i(k)\}_{i \in I_N}$ s.t $\sum_{i \in I_N} z_i(k) = r(k)$, $\{z_i(k)\}_{i \in I_N} \sim \mathcal{N}(\mu(k), \sigma^2)$. Suppose the resource to be allocated at the next time instant is $r(k+1)$. If agents update their resources as*

$$z_i(k+1) = z_i(k) + \frac{1}{N}(r(k+1) - r(k)) \qquad (5.9)$$

then the resulting solution satisfies the following:

1. $\sum_{i \in I_N} z_i(k+1) = r(k+1)$

2. $\{z_i(k+1)\}_{i \in I_N} \sim \mathcal{N}(\mu(k+1), \sigma^2)$

where $\mu(k+1)$ is a solution of the $(N+1)$ SNLE (5.5).

Proof: Obtain the time-difference of the summation of the resources:

$$\begin{aligned}\sum_{i\in I_N} z_i(k+1) - \sum_{i\in I_N} z_i(k) &= \sum_{i\in I_N} z_i(k+1) - z_i(k)\\ &= -N\delta\\ &= N(\mu(k+1)-\mu(k))\\ &= r(k+1)-r(k) \qquad (5.10)\end{aligned}$$

□

Following the Theorem 5.2 we obtain the CVT that satisfies the dynamic resource allocation constraint for the desired Normal distribution in a decentralized manner. While this fulfills our objective, it can be observed that the approach is quite rigid. In practical applications where the agents have their own set of dynamics and are trying to navigate around certain local objectives as well, this approach can be restrictive. Therefore, to extend its applicability we introduce some flexibility in the design by allowing for (local) negotiations between communication, and hence resource, neighbors through what we call a "*civility model*".

Before detailing the civility model, let us introduce and recall some new and old notations, respectively. For each agent $i \in I_N$, denote its desired resource amount at time k that meets its local objective as $u_i(k)$. For example, if the agent i is responsible for the control of a certain system modeled as a state-space, such $u_i(k)$ could be the control input from a state-feedback controller or from an LQR or any such local controller like MFC from Section 2.1.1. Since we are operating in 1-D spaces, recall from Section 5.1.1 that the resource and communication graphs are the same. Following the same notation therein, denote the communication graph at time k as $\mathcal{C}^k$, and the neighbors of agent i at time k as $\mathcal{N}_{\mathcal{C}^k_i}$.

Initialization: All agents are aware of the total resources $r(k), \forall k \in T$ and the initial communication network $\mathcal{C}^{k-1}$. Solve the static allocation problem for the resource $r(k-1)$.

Following the initialization for decentralized dynamic resource allocation, the civility model for local negotiations is developed as follows.

Civility model for local negotiations

For every agent $i \in I_N$, at every time $k \in T$, do:

1 Compute the resource update $z_i(k)$ from (5.9). Compute $u_i(k)$ based on the local requirements, possibly from the local controller.

2 Compute the neighbor of interest as $\hat{j} = \{j \in \mathcal{N}_{\mathcal{C}_i^k} \cup i$ such that $||u_i(k) - z_{\hat{j}}(k)|| < ||u_i(k) - z_j(k)||\}$.

3 Swap resources with the neighbor of interest $\hat{j}$ from the previous step, if $\hat{j}$ indicates it has not already been taken. This results in $z_i(k) = z_{\hat{j}}(k)$. If $\hat{j}$ has already negotiated with its other neighbor and is hence taken, or if $\hat{j} = i$, then implement the resource update $z_i(k)$ from Step 1.

It is worth noting that the communication network is dynamically updated in a decentralized manner, and that such an update naturally follows from the resource swap during the local negotiations. We call this approach the civility model because if a neighbor asks to swap, the agent complies with it regardless of its own local requirement. And hence, since all the agents follow the same model, no agent is at a disadvantage in following such approach.

To demonstrate the clarity and effectiveness of the proposed method to dynamically allocate resources in a decentralized manner, we revisit the application of demand-response in smart grids from Chapter 2. Specifically, we consider a group of Heating, Ventilation, and Air Conditioning (HVAC) units that have their local objectives to maintain their indoor air temperatures according to certain desired setpoints, but are also required to respond to certain demand (power) curve by consuming the available power as a team of agents.

5.4 Application to Demand Response

To demonstrate the developed method, we consider power allocation in a group of building HVACs. In this application of demand-response, the agents are the building HVACs. The resources to be allocated to all the agents is power consumed by the HVACs to maintain the indoor air temperatures. We adapt the state-space model from [49] to simulate the indoor air temperatures for each agent i as:

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i u_i(t) + G_i w_i(t) \\ y_i(t) &= C_i x_i(t) + D_i u_i(t) \end{aligned} \tag{5.11}$$

The input u_i is the power consumption of the HVAC (agent i), the output y_i is the indoor air temperature, and w_i is the vector of disturbances – outdoor air temperature and solar radiation. The system matrices for each agent are given by:

$$A_i = \begin{bmatrix} \frac{-(K_1^i+K_2^i+K_3^i+K_5^i)}{C_1^i} & \frac{(K_1^i+K_2^i)}{C_1^i} & \frac{K_5^i}{C_1^i} \\ \frac{K_1^i+K_2^i}{C_2^i} & \frac{-(K_1^i+K_2^i)}{C_2^i} & 0 \\ \frac{K_1^i}{C_3^i} & 0 & \frac{-(K_4^i+K_5^i)}{C_3^i} \end{bmatrix}$$

$$B_i = \begin{bmatrix} \frac{1}{C_1^i} + \frac{1}{C_2^i} \\ 0 \\ 0 \end{bmatrix} G_i = \begin{bmatrix} \frac{K_3^i}{C_1^i} & \frac{1}{C_1^i} \\ 0 & \frac{1}{C_2^i} \\ \frac{K_4^i}{C_3^i} & 0 \end{bmatrix} C_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

with D_i being a zero matrix. The system parameters, which are resistances and capacitances in the thermal dynamics of the building model, for each agent i are obtained as realizations of the following normal distributions:

$$\begin{aligned} K_1 &\sim \mathcal{N}(16.48, 0.1) & K_5 &\sim \mathcal{N}(23.04, 0.1) \\ K_2 &\sim \mathcal{N}(108.5, 0.1) & C_1 &\sim \mathcal{N}(9.36 \times 10^5, 1) \\ K_3 &\sim \mathcal{N}(5, 0.1) & C_2 &\sim \mathcal{N}(2.97 \times 10^6, 1) \\ K_4 &\sim \mathcal{N}(30.5, 0.1) & C_3 &\sim \mathcal{N}(6.695 \times 10^5, 1) \end{aligned}$$

We implement the agent's model by discretizing the state-space model (5.11) with a sampling time of 10 minutes. In the HVAC model, the input u_i corresponds to cooling when negative and corresponds to heating when positive. Regardless, its absolute value is the power consumed, and therefore we use that for local negotiations and let the individual agent decide whether to use the allocated power for heating or cooling based on its local control. To maintain the indoor air temperatures from a local control, we employ a state-feedback controller for pole placement for every agent to determine its $u_i(k)$. We consider the same disturbances for all the agents; the outdoor air temperature and the solar radiation we use for our simulations are shown in Fig. 5.5. The disturbances are derived from variation in a typical summer day in Knoxville, Tennessee.

To begin the dynamic resource allocation we initialize $\rho(.)$ as $\mathcal{N}(\mu, \sigma^2)$, and following Section 5.2, solve the first-time allocation (initialization) as a static allocation problem. Communicating to all the agents the resulting mean μ, we begin decentralized dynamic allocation as laid out in Section 5.3.

Even though the performance of the developed approach depends on the total available resource and the local requirements, the civility model allows for flexibility,

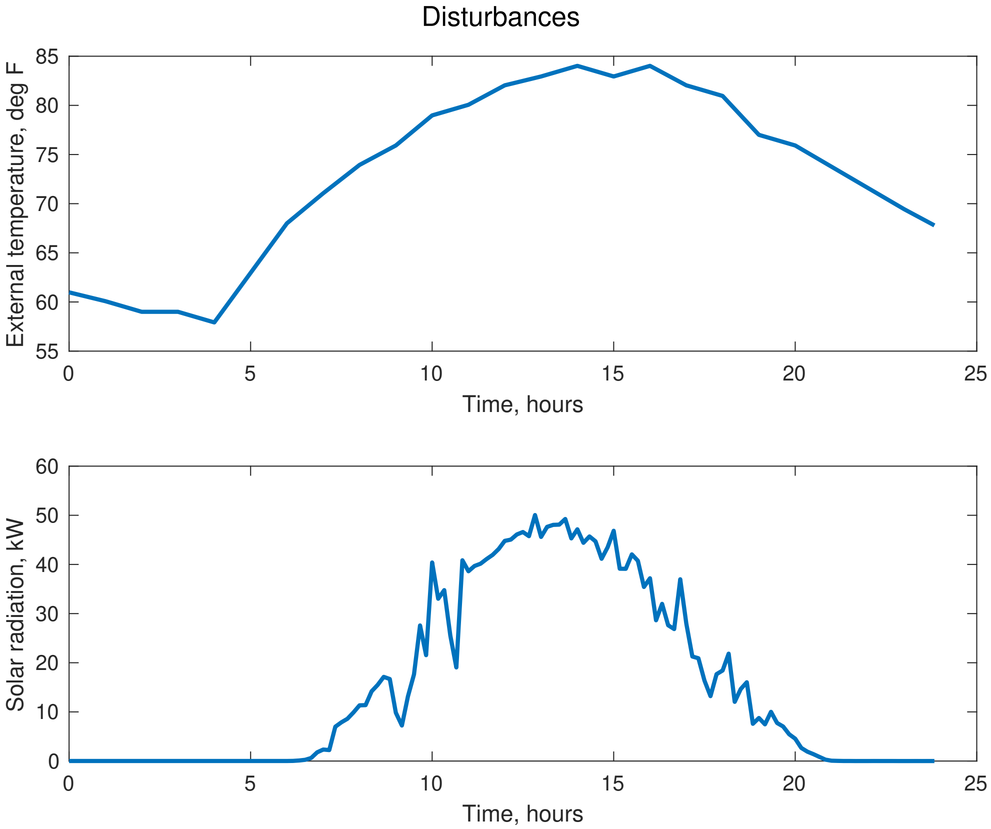


Figure 5.5: Disturbances in the HVAC model (5.11)

which could be necessary for local disturbances or improper selection for the (desired) distribution in the tessellation. To explain the graphical setup, we begin with $N = 5$; Fig 5.6, on the left, shows the power consumption of all the agents at every time instant, and on the right shows their total power consumption versus the available power. Augmenting, Fig 5.7 shows the individual indoor air temperatures when the agents implement the allocated power from Fig 5.6.

Next, we demonstrate the civility model from Section 5.3 by allowing for swapping through local negotiations. We first show only 5 agents in Fig. 5.8 continuing the previous case and then demonstrating for 15 agents in Fig. 5.9. For every agent, the power consumption and the indoor air temperature are shown in the same color through the simulation duration. For example, agent 2 is shown in red. Thus one can follow the agents' negotiations and the resulting swaps and communication network by following the individual power consumption of the agents through their colors. In the subsequent cases, we do not show the satisfaction of the resource allocation constraint through a dedicated figure since we can concisely express it numerically as the error between total power consumption of all the agents and the available power; we use the l_2 norm to compute the power consumption error.

The strengths of the developed method lie in its robustness in maintaining the resource allocation constraint while accounting for local preferences in a truly decentralized manner. To demonstrate the same, we perturb the setpoints of certain agents and observe the corresponding resource negotiations and the air temperatures in Fig 5.10. We can observe the increased amount of negotiations through the increased number of swaps spreading throughout the team to correct for the disturbances for some of the agents. Quantifying the swaps, we have that out of 144 time-steps in the simulation, each agent swapped 126.9 times on average and that every agent has been a neighbor of almost every other agent. This suggests a high degree of variation in the communication network, further suggesting that the amount of information is so fragmented among all the agents that it is sufficient to meet the resource allocation

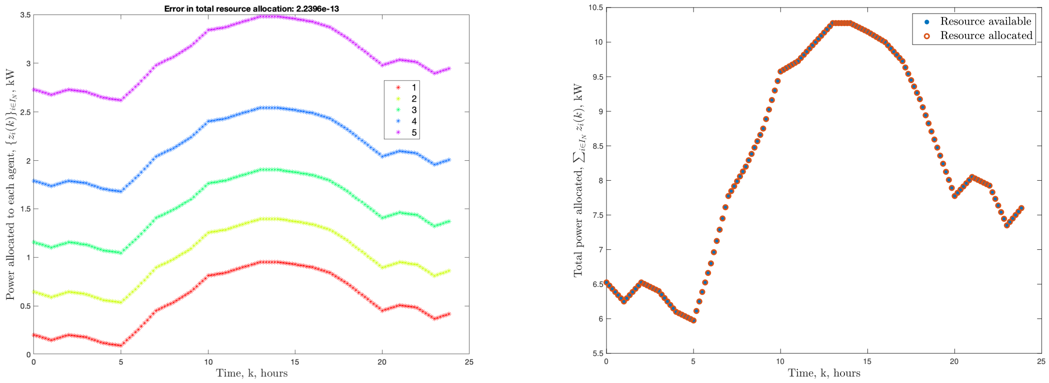


Figure 5.6: Baseline power consumptions: Agents acting based on the resource allocation constraint without the civility model. Left: Individual power consumption. Right: Total power consumption.

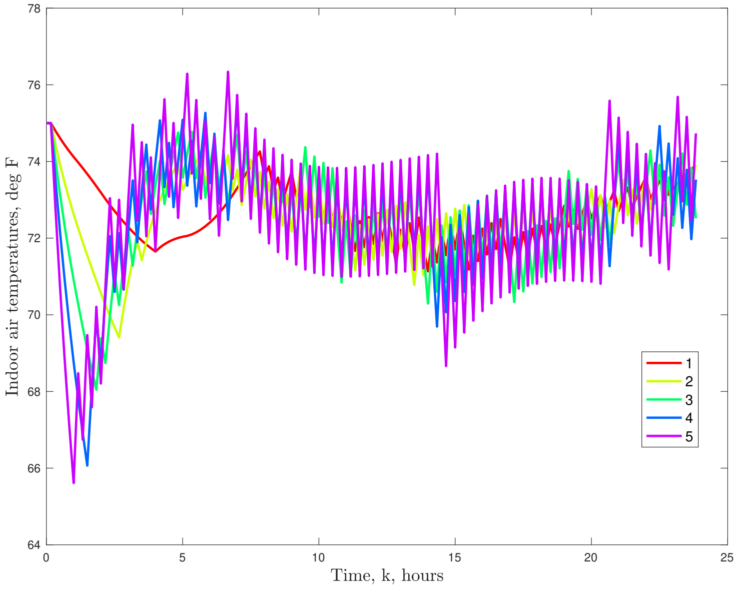


Figure 5.7: Baseline indoor air temperatures: Agents acting based on the resource allocation constraint without the civility model.

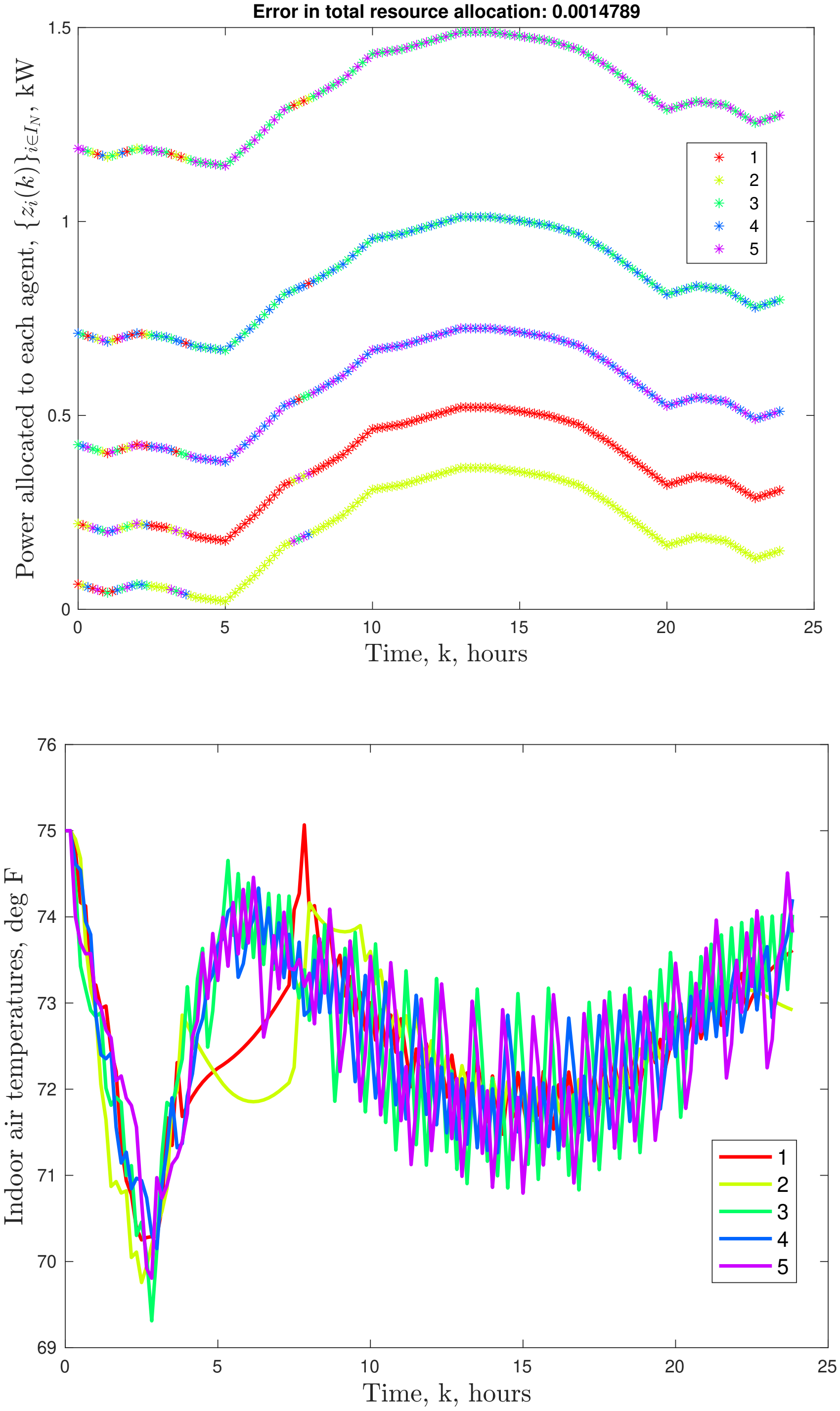


Figure 5.8: Civility model with local state feedback controller for 5 agents.

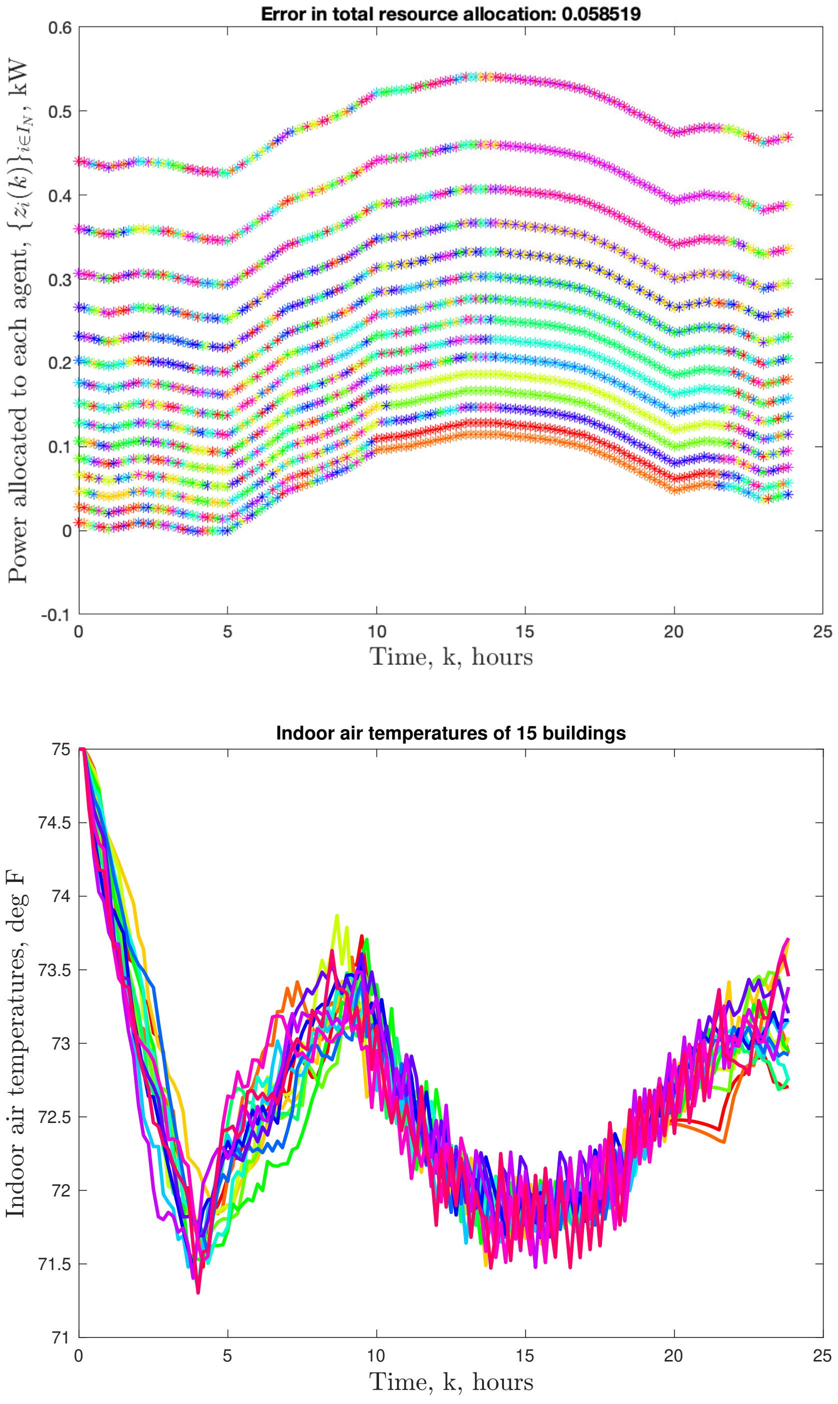


Figure 5.9: Civility model with local state feedback controller for 15 agents. The temperature setpoints for all the HVACs are at $72°F$.

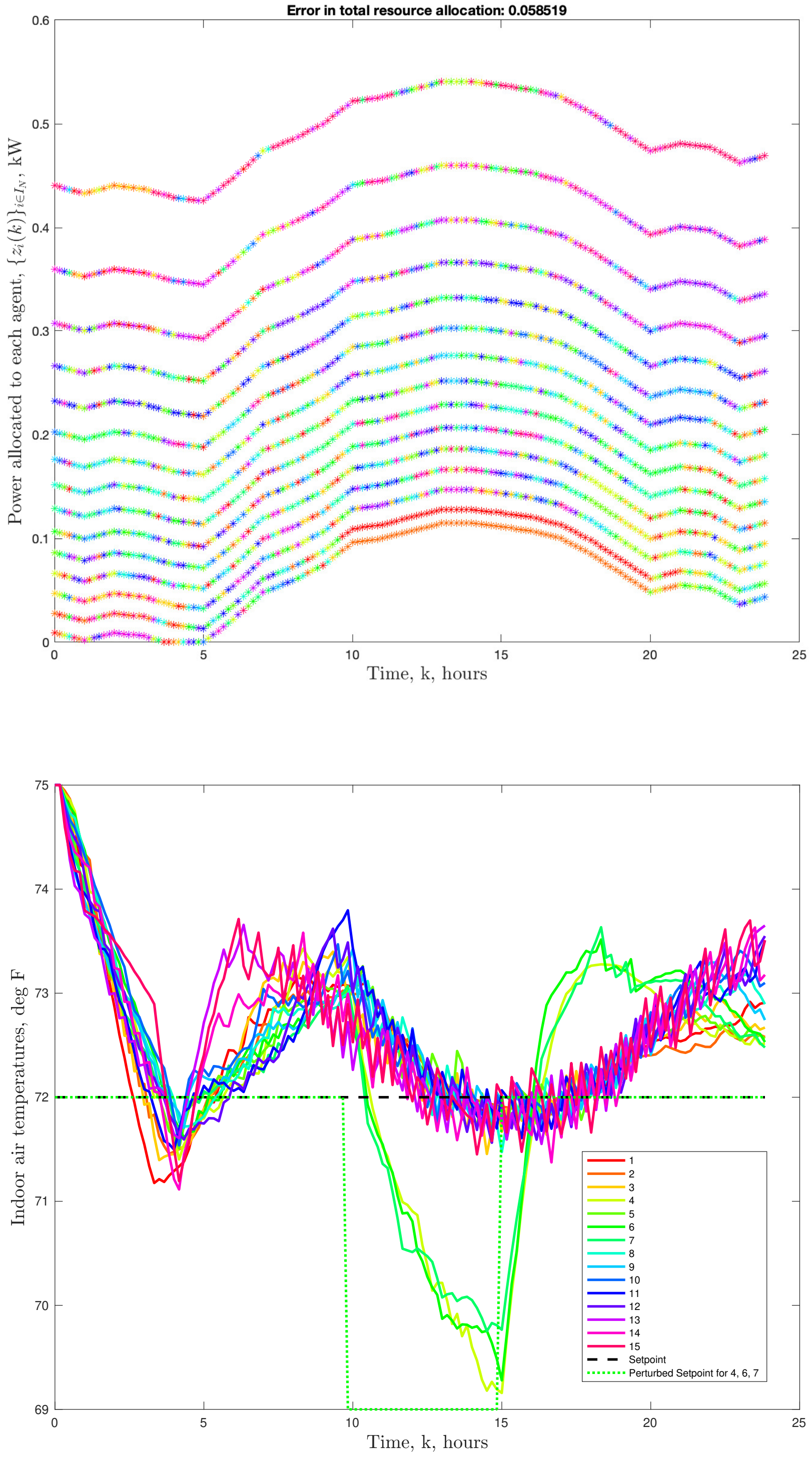


Figure 5.10: With swaps and local state feedback controller of 15 agents under disturbed setpoints

constraint while following the desired distribution in the tessellation, but not enough for any agent to recreate the behavior of any other agent.

The dynamic resource allocation solution proposed in this chapter follows the idea of "Global trendsetting, local negotiations" introduced in Chapter 1. Here, the the global trend is for the agents' resources to be Normally distributed while summing up to the available power, and the local negotiations happen to maintain the balance between following such global trend and accounting for the local requirements simultaneously.

Chapter 6

Conclusions and Future Work

In this Chapter, we first derive certain conclusions from each of the chapters in this dissertation, and then end it with some promising lines of future work.

6.1 Conclusions

In this dissertation, we considered the topic of resource allocation in multi-agent systems, and probed it through different frameworks. We started in Chapter 2 with a centralized architecture to allocate a certain amount of generated power from a renewable source to a group of building loads. In such a framework, one central controller coordinated the power consumption of all the building loads, and the local controllers were responsible for maintaining their loads with communication to the central controller only (and none with other building loads). While such architecture is currently employed in real-world applications like the transactive energy market, the costs and risks associated with the communication overhead render the framework rather costly.

Team decision theory provides a relatively lighter and a generic architecture to model a team. Taken up in Chapter 3, team decision theory models the information structure within a team, allows for a global objective to be optimized by the team whose members do not have any local or individual objective to optimize. The celebrated Witsenhausen counterexample effectively shows the complexity a non-classical information structure can introduce even in a simple two-player team problem with linear dynamics and quadratic cost. Even though we developed a method to implement the analytically optimal control laws, the method is rather cumbersome and its applicability in real-world applications remains an open question.

Therefore, we move on to consider Centroidal Voronoi Tessellations to solve the resource allocation problem. We study them in Chapter 4 and compare different computational methods through their advantages and disadvantages, usability in the decentralized framework, and extension to computation in higher dimensions. In the process, we observe the lack of a computational method to obtain a complete CVT in a high dimensional space, not just the centroids. Therefore, we develop a computational method to obtain some of the many non-unique CVTs in a high-dimensional space. Although such CVTs are grid-like tessellations, the developed method renders them in a simple and efficient manner, and hence allowing a user to obtain a number of the non-unique CVTs under the same conditions to improve the understanding of a considered solution space.

Having studied CVTs to a certain depth, we move on to solving the resource allocation problem in Chapter 5 by first precisely stating the resource allocation problem and explaining the motivation for employing CVTs as the solution. Using the developed system of non-linear equations method we solve the static resource allocation problem by embedding the resource allocation constraint within the distribution of the tessellation. Recognizing such approach to be centralized and employing it only as an initialization step, we consider a dynamic resource allocation problem where the amount of resource to be allocated varies with time and develop a decentralized solution through a combination of a single-update for tessellations with Normal distributions and a *civility model* for local negotiations between the agents. The developed approach involves a dynamic (time-varying) communication network that is merely a line graph at all times, hence resulting in minimal communication requirements in the team.

We now conclude this dissertation by comparing the centralized and the decentralized power allocation methods developed in Chapters 2 and 5, respectively. The centralized power allocation approach has fewer and fixed communication links, but suffers from the following disadvantages:

- Lack of privacy and security.
- Rigid architecture, hence causing difficulty in, for example, inclusion of local preferences resulting from any number of causes like local disturbances.
- Does not easily scale to a large number of agents.
- Difficulty in accounting for continuous-input systems or high degree of heterogeneity in agents.

The developed decentralized dynamic resource allocation method, as demonstrated with application to power allocation among a group of HVACs, rectifies the drawbacks of the centralized approach to a large extent. The advantages can be summarized through the following points.

- The dynamic communication network, with any agent having at most two neighbors at a given time, results in incomplete information about an agent to another, thus preserving privacy to an extent.
- The civility model allows for dynamic flexibility to address changes in local preferences.
- Easily scalable to any number of agents because the communication network is a line graph regardless of the number of participating agents.
- There is no restriction on the dynamics or the type of the agents. As long as all the agents consume the same resource quantity, they are flexible to further use the allocated resource as locally required.

6.2 Future Work

The use of CVTs to solve a dynamic resource allocation problem in a team has proved promising. We are interested in further exploring this work along following lines.

1. **Analysis of the privacy level:** The amount of information is so fragmented among all the agents that it is sufficient to meet the resource allocation constraint while following the desired distribution in the tessellation but not enough to assemble information about an agent to easily recreate its behavior. Accordingly, we find it interesting to look into the level of security inherently offered in this framework, make recommendations for the applications that prioritize privacy and security, and further improve the security using the developed framework.

2. **Extension to other constraints:** The resource allocation constraint is to maintain the sum of the individual resources. The developed static allocation framework, extends as is, to any constraint that takes in all the agents' resources and has a range in $\mathbb{R}$. Accordingly, its extension to relevant applications can be explored.

3. **Generalization to more distributions:** The single-update for dynamic allocation assumes the desired distribution in the tessellation is a Normal distribution. While this assumption models a large number of applications, it will be interesting to generalize such update rule to other distributions.

4. **Explicitly model the distributions:** Following up on the previous point, we can employ historical information to model a desired distribution of the allocated resources to set the global trend.

5. **Extension to higher dimensions:** Ultimately, while we consider one-dimensional resources, it might be worth looking into the extension of the developed decentralized dynamic allocation method to higher dimensional resources,

possibly using the developed computational method to obtain CVTs in high-dimensional spaces.

Bibliography

[1] Abouaïssa, H., Hasan, O., Join, C., Fliess, M., and Defer, D. (2017). Energy saving for building heating via a simple and efficient model-free control design: First steps with computer simulations. *arXiv:1708.03800*. 4

[2] Atkinson, K., Graham, I., and Sloan, I. (1983). Piecewise continuous collocation for integral equations. *SIAM Journal on Numerical Analysis*, 20(1):172–186. 55

[3] Aurenhammer, F. (1986). The one-dimensional weighted Voronoi diagram. *Information Processing Letters*, 22(3):119–123. 8

[4] Baglietto, M., Parisini, T., and Zoppoli, R. (2001). Numerical solutions to the Witsenhausen counterexample by approximating networks. *IEEE Transactions on Automatic Control*, 46(9):1471–1477. 49, 60, 64

[5] Bai, Y., Wang, Y., Svinin, M., Magid, E., and Sun, R. (2022). Adaptive multi-agent coverage control with obstacle avoidance. *IEEE Control Systems Letters*, 6:944–949. 8

[6] Baldi, S., Michailidis, I., Kosmatopoulos, E., and Ioannou, P. (2014). A plug and play computationally efficient approach for control design of large-scale nonlinear systems using cosimulation: A combination of two ingredients. *IEEE Control Systems*, 34(5):56–71. 4

[7] Banjac, G., Rey, F., Goulart, P., and Lygeros, J. (2019). Decentralized resource allocation via dual consensus ADMM. In *2019 American Control Conference (ACC)*, pages 2789–2794. 10

[8] Bansal, R. and Basar, T. (1987). Stochastic teams with nonclassical information revisited: When is an affine law optimal? *IEEE Transactions on Automatic Control*, 32(6):554–559. 49, 60

[9] Basar, T. (2008). Variations on the theme of the Witsenhausen counterexample. In *2008 47th IEEE Conference on Decision and Control*, pages 1614–1619. xii, 60, 61

[10] Burger, S., Chaves-Ávila, J., Batlle, C., and Pérez-Arriaga, I. (2017). A review of the value of aggregators in electricity systems. *Renewable and Sustainable Energy Reviews*, 77:395–405. 6

[11] Burkardt, J., Gunzburger, M., and Lee, H.-C. (2006). POD and CVT-based reduced-order modeling of navier–stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1):337–355. 92

[12] Cai, H. (2012). Decentralized control of stochastic dynamic systems with applications to resource allocation and portfolio management. 10

[13] Callaway, D. S. (2009). Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy. *Energy Conversion and Management*, 50(5):1389–1400. 23

[14] Camacho, E. F., B. C. (2007). Model predictive control. 30

[15] Camacho, E. F. and Bordons, C. A. (1997). *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Heidelberg. 2

[16] Charalambous, C. D. and Ahmed, N. U. (2014). Equivalence of decentralized stochastic dynamic decision systems via girsanov's measure transformation. In *53rd IEEE Conference on Decision and Control*, pages 439–444. 46, 49, 50

[17] Chen, T., Pourbabak, H., and Su, W. (2019). Electricity market reform. *The Energy Internet*, pages 97–121. 5

[18] Cortés, J. and Bullo, F. (2005). Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574. 2

[19] Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255. 8

[20] Davydov, A. and Diaz-Mercado, Y. (2020). Sparsity structure and optimality of multi-robot coverage control. *IEEE Control Systems Letters*, 4(1):13–18. 8

[21] Doan, T. T. and Beck, C. L. (2021). Distributed resource allocation over dynamic networks with uncertainty. *IEEE Transactions on Automatic Control*, 66(9):4378–4384. 10

[22] Dong, J., Djouadi, S., Kuruganti, T., and Olama, M. (2017). Augmented optimal control for buildings under high penetration of solar photovoltaic generation. *Proceedings of the IEEE Conference on Control Technology and Applications*, pages 2158–2163. 4, 30

[23] Du, Q., Emelianenko, M., and Ju, L. (2006). Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis*, 44(1):102–119. 74

[24] Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676. 67, 69, 71

[25] Du, Q., Gunzburger, M. D., and Ju, L. (2003). Constrained centroidal Voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing*, 24(5):1488–1506. 8

[26] Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition. 3

[27] Fleischer, P. (1964). Sufficient conditions for achieving minimum distortion in a quantizer. *IEEE International Convention Record, Pt I*, pages 104–111. 72

[28] Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12):2228–2252. 10, 15, 17, 18

[29] Goberna, M., Martínez-Legaz, J., and Vera de Serio, V. (2016). The Voronoi inverse mapping. *Linear Algebra and its Applications*, 504:248–271. 8

[30] Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–s10. 52

[31] Greenwood, R. and Miller, J. (1947). Zeros of the hermite polynomials and weights for Gauss mechanical quadrature formula. *Bulletin of the American Mathematical Society*, 54(8):765–769. 51, 52

[32] Gupta, A. (2014). *Dynamic sequential decision problems with asymmetric information: Some existence results*. PhD thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-08-08. 46

[33] Gupta, S. K., Kar, K., Mishra, S., and J.Wen (2014). Building temperature control with active occupant feedback. *IFAC Proceedings Volumes*, 47(3):851–856. 4

[34] Hatless, J., Wei, H., and L, C. (2015). Fast methods for computing centroidal Voronoi tessellations. *Journal of Scientific Computing*, 63:185–212. 75

[35] Ho, Y.-c. and Chu, K.-c. (1971). Team decision theory and information structures in optimal control problems: Part i. In *1971 IEEE Conference on Decision and Control*, pages 383–387. 7

[36] Ho, Y.-C., Kastner, M., and Wong, E. (1978). Teams, signaling, and information theory. *IEEE Transactions on Automatic Control*, 23(2):305–312. 7

[37] H.S.Witsenhausen (1968). A counterexample in stochastic optimal control. *SIAM Journal on Control*, 6(1). 7, 46, 49, 56, 57, 60

[38] Inoue, D., Ito, Y., and Yoshida, H. (2021). Optimal transport-based coverage control for swarm robot systems: Generalization of the Voronoi tessellation-based method. *IEEE Control Systems Letters*, 5(4):1483–1488. 8

[39] Ju, L., Du, Q., and Gunzburger, M. (2002). Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. *Parallel Computing*, 28(10):1477–1500. 88, 89, 92

[40] Ju, L., Ringler, T., and Gunzburger, M. (2011). *Voronoi Tessellations and Their Application to Climate and Global Modeling*, pages 313–342. Springer Berlin Heidelberg, Berlin, Heidelberg. 8

[41] Karlsson, J., Gattami, A., Oechtering, T. J., and Skoglund, M. (2011). Iterative source-channel coding approach to Witsenhausen's counterexample. pages 5348–5353. 49

[42] Kieffer, J. (1983). Uniqueness of locally optimal quantizer for log-concave density and convex error weighting function. *IEEE Transactions on Information Theory*, 29(1):42–47. 75

[43] Krainak, J., Speyer, J., and Marcus, S. (1982). Static team problems–Part I: Sufficient conditions and the exponential cost criterion. *IEEE Transactions on Automatic Control*, 27(4):839–848. 2, 7

[44] Lakshmanan, H. and de Farias, D. P. (2008). Decentralized resource allocation in dynamic networks of agents. *SIAM Journal on Optimization*, 19(2):911–940. 10, 95, 96

[45] Lee, J. T., Lau, E., and Ho, Y.-C. (2001). The Witsenhausen counterexample: a hierarchical search approach for nonconvex optimization problems. *IEEE Trans. Automat. Contr.*, 46:382–397. xii, 49, 60, 62, 63, 64

[46] Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.-M., Lu, L., and Yang, C. (2009). On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph.*, 28(4). 75

[47] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137. 74

[48] Loebl, M. (2010). *Introduction to Graph Theory*, pages 13–49. Vieweg+Teubner, Wiesbaden. 98, 100

[49] Ma, X., Dong, J., Djouadi, S. M., Nutaro, J. J., and Kuruganti, T. (2015). Stochastic control of energy efficient buildings: A semidefinite programming approach. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 780–785. 114

[50] Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. 88

[51] Mahajan, A., Martins, N. C., Rotkowitz, M. C., and Yüksel, S. (2012). Information structures in optimal decentralized control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 1291–1306. 7

[52] Malizou, A. (2018). Electricity aggregators: Starting off on the right foot with consumers. *BEUC, The European Consumer Organization*. 6

[53] Marschak, J. (1955). Elements for a theory of teams. *Management Science*, 1(2):127–137. 7, 45

[54] Mathieu, J. L., Koch, S., and Callaway, D. S. (2013). State estimation and control of electric loads to manage real-time energy imbalance. *IEEE Transactions on Power Systems*, 28(1):430–440. 23

[55] Mayya, S., Pierpaoli, P., Nair, G., and Egerstedt, M. (2019). Localization in densely packed swarms using interrobot collisions as a sensing modality. *IEEE Transactions on Robotics*, 35(1):21–34. 1

[56] McEneaney, W. M. and Han, S. H. (2015). Optimization formulation and monotonic solution method for the Witsenhausen problem. *Automatica*, 55:55 – 65. 49

[57] McEneaney, W. M., Han, S. H., and Liu, A. (2011). An optimization approach to the Witsenhausen counterexample. pages 5023–5028. 60

[58] Minakais, M., Mishra, S., and Wen, J. T. (2014). Groundhog day: Iterative learning for building temperature control. *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 948–953. 5

[59] Mouret, J. and Clune, J. (2015). Illuminating search spaces by mapping elites. *CoRR*, abs/1504.04909. 92

[60] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. 88

[61] Park, C. and Yong, T. (2017). Comparative review and discussion on P2P electricity trading. *Energy Procedia*, 128:3–9. 6

[62] Pimbley, J. (2017). Hermite polynomials and Gauss quadrature. 52

[63] Popo, D. and Lungu, N. (2002). On some differential inequalities. *Seminar on Fixed Point Theory Cluj-Napoca*, 3:323–326. 20

[64] Preparata, F. P. and Shamos, M. I. (1985). *Computational Geometry: An Introduction*. Springer-Verlag, Berlin, Heidelberg. 41

[65] Radner, R. (1962). Team decision problems. *The Annals of Mathematical Statistics*, 33(3):857–881. 7, 45

[66] Remani, C. (2013). Numerical methods for solving systems of nonlinear equations. *Lakehead University Thunder Bay, Ontario, Canada*. 77

[67] Romvary, J. J. (2015). A numerical study of Witsenhausen's counterexample. *DSpace MIT*. 49

[68] Sioshansi, F. (2018). How aggregators will alter fundamentals of electricity business. *Energypost.eu*. 6

[69] Snow, J. (1855). *On the Mode of Communication of Cholera*. 8

[70] Telsang, B., Amasyali, K., Chen, Y., Olama, M., and Djouadi, S. (2021a). Power allocation by load aggregator with heterogeneous loads using weighted projection. *Energy and Buildings*, 244:110955. 35

[71] Telsang, B., Djouadi, S., and Charalambous, C. (2021b). Numerical evaluation of exact person-by-person optimal nonlinear control strategies of the Witsenhausen counterexample. In *2021 American Control Conference (ACC)*, pages 1250–1255. 46, 49

[72] Telsang, B., Djouadi, S., Olama, M., Kuruganti, T., Dong, J., and Xue, Y. (2018). Model-free control of building HVAC systems to accommodate solar photovoltaicenergy. In *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pages 1–7. 28

[73] Telsang, B. and Djouadi, S. M. (2022). Computation of centroidal Voronoi tessellations in high dimensional spaces. *IEEE Control Systems Letters*, 6:3313–3318. 83

[74] Telsang, B., Olama, M., Djouadi, S., Dong, J., and Kuruganti, T. (2019). Stability analysis of model-free control under constrained inputs for control of building HVAC systems. In *2019 American Control Conference (ACC)*, pages 5878–5883. 18

[75] Urquhart, N., Guckert, M., and Powers, S. (2019). Increasing trust in meta-heuristics by using MAP-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pages 1345–1348, New York, NY, USA. Association for Computing Machinery. 93

[76] Urschel, J. C. (2017). On the characterization and uniqueness of centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis*, 55(3):1525–1547. 9, 72

[77] van Schuppen, J. and Villa, T., editors (2015). *Coordination Control of Distributed Systems*. Lecture Notes in Control and Information Sciences. Springer. 7, 45, 46

[78] Vassiliades, V., Chatzilygeroudis, K., and Mouret, J.-B. (2018). Using centroidal Voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22(4):623–630. 92

[79] Wang, X., Ying, X., Liu, Y.-J., Xin, S.-Q., Wang, W., Gu, X., Mueller-Wittig, W., and He, Y. (2015). Intrinsic computation of centroidal Voronoi tessellation on meshes. *Computer-Aided Design*, 58:51–61. Solid and Physical Modeling 2014. 75

[80] Wegner, D. M., Giuliano, T., and Hertel, P. T. (1985). *Cognitive Interdependence in Close Relationships*, pages 253–276. Springer New York, New York, NY. 2

[81] Williams, S. and Short, M. (2020). Electricity demand forecasting for decentralised energy management. *Energy and Built Environment*, 1(2):178 – 186. 5

[82] Wu, Y. and Verdú, S. (2011). Witsenhausen's counterexample: A view from optimal transport theory. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5732–5737. 57

[83] Xiao, L. and Boyd, S. (2006). Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3):469–488. 10, 96

[84] Zhang, C., Wu, J., Long, C., and Cheng, M. (2017). Review of existing peer-to-peer energy trading projects. *Energy Procedia*, 105:2563–2568. 6

Appendix

A Explicit solution to the constrained linear least square problem

In this Section, an explicit solution to the optimization problem (2.30) is presented. Restating the problem:

$$\min_{x \in \mathbb{R}^n} ||x - s||_2^2 \tag{1}$$

such that,

$$\sum_{j=1}^{n} x(j) = P \text{ and } 0 \leq x \leq p.N$$

The first constraint is the equality constraint to obtain a solution that is on the power constraint plane (2.27). The second constraint is the inequality constraint to ensure that the solution lies within the specified boundaries. Let us denote these power and boundary constraint planes explicity as:

$$
\begin{aligned}
Q_c : x_1{+}x_2 + \ldots + x_n &= P \\
\underline{Q}_1 : x_1 &= 0 \\
\underline{Q}_2 : x_2 &= 0 \\
&\vdots \\
\underline{Q}_n : x_n &= 0 \\
\bar{Q}_1 : x_1 &= p_1 N_1 \\
\bar{Q}_2 : x_2 &= p_2 N_2 \\
&\vdots \\
\bar{Q}_n : x_n &= p_n N_n
\end{aligned}
$$

Intersecting every boundary constraint plane with the constraint plane Q_c results in n lines: $\underline{l}_1, \underline{l}_2, \ldots, \underline{l}_n$ and $\bar{l}_1, \bar{l}_2, \ldots, \bar{l}_n$, where $\forall k = 1, 2, \ldots, n$, $\underline{l}_k$ is the intersection of $\underline{Q}_k$ and Q_C, and $\bar{l}_k$ is the intersection of $\bar{Q}_k$ and Q_c. Mutual intersection of all the lines $\underline{l}_k$ and $\bar{l}_k$ results in at most $O = n(n-1)$ number of points on the power constraint plane, denoted $a_1, a_2, \ldots a_{n(n-1)}$.

If $p_k N_k > P$, $\forall k$, then there will be n vertices, else $n(n-1)$ vertices. As a rule of thumb, for certain k, if $p_k N_k > P$, then one intersection point is obtained; else $(n-1)$ number of intersection points are obtained.

The surface on the power constraint plane that is enclosed by these vertices $a_1, a_2, \ldots a_{n(n-1)}$ is the feasible region, denoted R, for the solution of the optimization problem (1). That is, if $x \in R$ then such x is on the power constraint plane and within the boundary conditions.

Since R is convex, denote a point in R as:

$$x = \mu_1 a_1 + \mu_2 a_2 + \ldots + \mu_n a_n \tag{2}$$

where each $\mu_k\ in[0,1]$ and all the coefficients μ_k sum up to 1. Accordingly, reformulate the original optimization problem as:

$$\min_{\mu \in \mathbb{R}^n} ||\mu_1 a_1 + \mu_2 a_2 + \ldots + \mu_n a_n - s||_2^2 \tag{3}$$

such that,

$$\sum_{k=1}^{n} \mu_k = P \text{ and } -\mu \leq 0$$

Optimization problem (3) can be solved using Karush-Kuhn-Tucker conditions. To demonstrate, consider the following example.

$$s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -2.13 \\ 1.3628 \end{bmatrix} \times 10^5, \qquad P = 0.7 \times 10^5, \quad n = 3$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 4.5 \\ 3.5 \end{bmatrix} \times 10^3 \qquad N = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}$$

While $s_1 + s_2 + s_3 = P$ (meeting the equality constraint), s violates the boundary condition since $s_2 < 0$ (inequality constraint). The upper boundary is the dot product of p and N:

$$p.N = \begin{bmatrix} p_1 N_1 \\ p_2 N_2 \\ p_3 N_3 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 4.5 \\ 3.5 \end{bmatrix} \times 10^5$$

Since $p_k N_k > P \quad \forall k = 1, 2, 3$, intersection of all the lines $\underline{l}_k$ and $\bar{l}_k$ results in the feasible region given by the following 3 vertices:

$$a_1 = \begin{bmatrix} P \\ 0 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 0 \\ P \\ 0 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 0 \\ 0 \\ P \end{bmatrix},$$

For this demonstration, the reformulated optimization problem (3) becomes:

$$\min_{\mu \in \mathbb{R}^3} \left|\left| \begin{bmatrix} \mu_1 P - s_1 \\ \mu_2 P - s_2 \\ \mu_3 P - s_3 \end{bmatrix} \right|\right|_2^2$$

such that,

$$\mu_1 + \mu_2 + \mu_3 = 1 \text{ and } -\mu \leq 0$$

The Lagrangian function and its partial derivative with respect to μ is given by:

$$L(\mu, u, v) = (\mu_1 P - s_1)^2 + (\mu_2 P - s_2)^2 + (\mu_3 P - s_3)^2 - u_1\mu_1 - u_2\mu_2 - u_3\mu_3 + v(\mu_1 + \mu_2 + \mu_3 - 1)$$
$$\frac{\partial L}{\partial \mu_k} = 2(\mu_k P - s_k)P - u_k + v, \quad k = 1, 2, 3$$

The Karush-Kuhn-Tucker conditions (KKT) are:

KKT1: $\frac{\partial L}{\partial \mu_k} = 0, \quad k = 1, 2, 3$

KKT2: $u_k \mu_k = 0, \quad k = 1, 2, 3$

KKT3: (a) $-\mu_k \leq 0, \quad k = 1, 2, 3$

(b) $\mu_1 + \mu_2 + \mu_3 - 1 = 0$

KKT4: $u_k \geq 0, \quad k = 1, 2, 3$

Since $s_2 < 0$ violates the inequality constraint, set $u_2 > 0$ in order to keep this constraint active. Therefore, from KKT2, $\mu_2 = 0$. From KKT1, the first condition, $u_k = 2P(\mu_k P - s_k) + v$. Since s_1 and s_3 satisfy the inequality constraint, these two constraints are kept inactive by having $u_1, u_3 = 0$. Therefore,

$$
\begin{aligned}
& u_1 = 0 \\
\implies & 2P(\mu_1 P - s_1) + v = 0 \qquad (4) \\
& u_3 = 0 \\
\implies & 2P(\mu_3 P - s_3) + v = 0 \qquad (5)
\end{aligned}
$$

From KKT3b condition with $\mu_2 = 0$,

$$\mu_1 + \mu_3 = 1 \qquad (6)$$

Combining equations (4)-(6) results three equations and three variables: μ_1, μ_3, v. Solving these three equations results in:

$$\mu^* = \begin{bmatrix} 0.689 \\ 0 \\ 0.311 \end{bmatrix}$$

Substituting the solution μ^* in the original optimization problem (1):

$$
\begin{aligned}
z^* = \mu_1 a_1 + \mu_2 a_2 + \mu_3 a_3 = & \begin{bmatrix} 4.997 \\ 0 \\ 2.2568 \end{bmatrix} \times 10^4 \\
\implies ||z^* - s||_2 = 6.133 \times 10^4 &
\end{aligned}
$$

Vita

Bhagyashri Telsang received the Bachelor of Engineering degree in Instrumentation and Control Engineering from Manipal Institute of Technology, Manipal, India in 2014. She then received her Master of Science degree in Systems and Control from Delft University of Technology, Delft, The Netherlands in 2016. She is currently pursuing the PhD degree in the department of Electrical Engineering and Computer Science at the University of Tennessee in Knoxville, USA, with the advisor Dr Seddik M Djouadi. Her research interests include multi-agent systems, decentralized control, filtering and identification, and smart grid and smart buildings.