2020

Optimal Mission Planning of Autonomous Mobile Agents for Applications in Microgrids, Sensor Networks, and Military Reconnaissance

Casey D. Majhor
Michigan Technological University

OPTIMAL MISSION PLANNING OF AUTONOMOUS MOBILE AGENTS FOR APPLICATIONS IN MICROGRIDS, SENSOR NETWORKS, AND MILITARY RECONNAISSANCE

By

Casey D. Majhor

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mechanical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2020

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mechanical Engineering.

Department of Mechanical Engineering-Engineering Mechanics

Thesis Advisor: *Dr. Wayne W. Weaver*

Committee Member: *Dr. Jeremy P. Bos*

Committee Member: *Dr. Jung Yun Bae*

Department Chair: *Dr. William W. Predebon*

Dedication

This thesis is dedicated to the various members of my immediate and extended family, especially my wife **Shannen C. Majhor.** Thank you for always supporting me.

Contents

List of Figures

List of Tables

Preface

The work presented in this thesis is part of my research conducted in 2017-2020 at Michigan Technological University. Mission optimization of mobile multi-agent systems is critical in energy scarce applications. This work applies the genetic algorithm for mission optimization in applications of autonomous mobile microgrids and agent resource scheduling for intelligence, surveillance, and reconnaissance missions.

Chapter 1 presents an introduction to this thesis by summarizing the optimization applications, problems, and solution method investigated. **Chapter 2** provides background details on previous work in autonomous mobile microgrids and resource scheduling control and optimization, along with the functionality of the genetic algorithm used. **Chapter 3** presents a genetic algorithm approach to optimize energy resource locations and network connections in an autonomous mobile microgrid system. The content in this chapter has been published in reference [2]. **Chapter 4** details a genetic algorithm strategy to optimize the recharging of distributed loads to extend their operational life. The content in this chapter has been published in reference [3]. **Chapter 5** presents a similar strategy to **Chapter 4** in determining the optimal recharging of mobile UAVs to increase their operational time for various reconnaissance missions. Conclusions and future work considerations are presented in **Chapter 6**.

Acknowledgments

I would like to thank my advisor, Dr. Wayne Weaver, for his guidance throughout the course of my master's degree.

I wish to thank my committee members Dr. Jeremy Bos and Dr. Jung Yun Bae for offering me their insight and support throughout the preparation of this thesis.

Thank you to the supporting members of our research lab and others who contributed to publications in this research, including Dr. Shadi Darani, Dr. Rush Robinett, Dr. Ossama Abdelkhalik, John Naglak, Carl Greene, Caleb Kase, and Max McGinty.

Additionally, I would like to thank the U.S. Office of Naval Research for their support on this research project. This work was supported by the U.S. Office of Naval Research under Award N00014-16-1-2422 and does not reflect their views or opinions.

Lastly, I would like to thank my wife Shannen for her constant support throughout my undergraduate and graduate career.

List of Abbreviations

A^*	The Astar Algorithm
AIM	Agile and Interconnected Microgrids
ART	Augmented Reality Tag
CPU	Central Processing Unit
DGR	Distributed Generation Resource
FOB	Forward Operating Base
GA	Genetic Algorithm
ISR	Intelligence, Surveillance, and Reconnaissance
JSP	Job-Shop Scheduling Problem
NP	Non-deterministic Polynomial-time
SMS	Single Machine Scheduling
SOC	State of Charge
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
VSDS	Variable-Size Design Space
WPT	Wireless Power Transfer
WSN	Wireless Sensor Network

Abstract

As technology advances, the use of collaborative autonomous mobile systems for various applications will become evermore prevalent. One interesting application of these multi-agent systems is for autonomous mobile microgrids. These systems will play an increasingly important role in applications such as military special operations for mobile ad hoc power infrastructures and for intelligence, surveillance, and reconnaissance missions. In performing these operations with these autonomous energy assets, there is a crucial need to optimize their functionality according to their specific application and mission. Challenges arise in determining mission characteristics such as how each resource should operate, when, where, and for how long.

This thesis explores solutions in determining optimal mission plans around the applications of autonomous mobile microgrids and resource scheduling with UGVs and UAVs. Optimal network connections, energy asset locations, and cabling trajectories are determined in the mobile microgrid application. The resource scheduling applications investigate the use of a UGV to recharge wireless sensors in a wireless sensor network. Optimal recharging of mobile distributed UAVs performing reconnaissance missions is also explored. With genetic algorithm solution approaches, the results show the proposed methods can provide reasonable *a-priori* mission plans, considering the applied constraints and objective functions in each application. The

contributions of this thesis are: (1) The development and analysis of solution methodologies and mission simulators for *a-priori* mission plan development and testing, for applications in organizing and scheduling power delivery with mobile energy assets. Applying these methods results in (2) the development and analysis of reasonable *a-priori* mission plans for autonomous mobile microgrids/assets, in various scenarios. This work could be extended to include a more diverse set of heterogeneous agents and incorporate dynamic loads to provide power to.

Chapter 1

Introduction

1.1 Overview

Microgrids are systems of interconnected energy sources and loads within a defined boundary. Such systems provide an added benefit to the common utility grid in that they provide localized, distributed, renewable, and more controllable energy resources. These resources may include photo-voltaic (PV) arrays, wind turbines, diesel generators, and energy storage, providing more resilient, clean, and efficient energy. Microgrids have the ability to operate in synchronous with the common utility grid system or independently in "island mode." These energy systems are a relatively new technology being adapted by residential developments, businesses, and

universities.

Autonomous mobile microgrids employ many of the same characteristics as stationary microgrids but have the added benefit of mobility and autonomy. These autonomous microgrids operate by navigating to areas with loads in need of the power infrastructure they provide by directly connecting to or interconnecting to the load or system of loads. Homogeneous or heterogeneous agents with the functionality of energy generation, power conversion, energy storage, and power buses for energy distribution, are asset types used in these mobile energy systems. The assets incorporate power transfer via cabling connection, wireless, or other technologies depending on the mission environment and objectives. Once connected to serve the existing loads, the assets can adapt to the environment as needed to provide more optimal or reliable energy. Environment changes, for example, include changes in the number of available energy assets or loads, load prioritization, or changes in the whether conditions. Adaptability is done by disconnecting from the loads, re-configuring how the mobile microgrid system is connected, and/or actively controlling the power generation and distribution. These mobile systems are beneficial in providing power infrastructure where it is otherwise nonexistent, along with human denied environments, and desolate areas. Useful applications include disaster recovery, planetary exploration, military forward operating bases, and intelligence, surveillance, and reconnaissance (ISR) missions.

Autonomous mobile microgrids, with their ability to adapt to dynamic mission needs,

is an example of an important developing use of collaborative autonomous mobile systems. A critical aspect in deploying these multi-agent systems is to determine which role each asset should play in the mission. Furthermore, what actions and interactions should occur between agents to maximize their use as a complete system? These mission characteristics are crucial to consider in missions where asset energy capacities are scarce, such as in the remote or desolate areas in the applications previous described.

This chapter first briefly introduces details on the two larger related topics of optimization in autonomous mobile microgrids and resource scheduling. Next, the genetic algorithm (GA) method used for optimization in these applications is explained. Motivations and objectives are then presented, followed by an organization description for the remainder of this thesis.

1.2 Autonomous Mobile Microgrid Architecture Optimization

Microgrids are energy systems that can operate dependently or independently of the established grid system. The power system comprises distributed generation resources (DGRs) such as diesel generators, photovoltaic (PV) arrays, and wind turbines. Microgrids enhance the resiliency and efficiency of the traditional power grid system by

operating local DGRs with adaptive control methods [4], [5]. Autonomous mobile microgrids serve a similar function as traditional microgrids, but they are ad hoc in their ability to move to different locations to serve different loads. A concept image of three agents in a mobile microgrid, equipped with a PV array, diesel generator and power conversion are shown in Figure 1.1. These microgrids can reconfigure and adapt to optimize performance as required. One application for specifically using autonomous mobile microgrids is for military electrical infrastructure. For example, when a military forward operating base (FOB) or a specialized unit needs a rendezvous point with power available, these UGVs can traverse to and connect to existing loads to set up the power infrastructure. This saves time in time-critical missions and helps to keep military personnel out of harm's way. Another application for the use of autonomous mobile microgrids is in post natural and man-made disasters such as hurricanes and large explosions, which often leaves thousands of people without power. These mobile agents can maneuver to locations too obstructed for larger vehicles or too dangerous for humans to traverse and connect to critical loads such as emergency medical shelters. For example, Figure 1.2 shows heterogeneous agents interconnected to power a communication tower. When energy and resources are scarce as described in these scenarios, it is crucial to optimize the operation of the microgrid systems as much as possible to maximize their utility and effectiveness. Architecture optimization, as applied to autonomous mobile microgrids, is determining the operational characteristics which satisfy constraints and optimizes an objective function. Optimization of many

operational parameters is possible, including energy source power output settings, functionality and use of energy storage, and the number of assets used. Chapter 3 investigates the operational parameters of optimal network connections, energy asset locations, and cabling trajectories for several case studies.

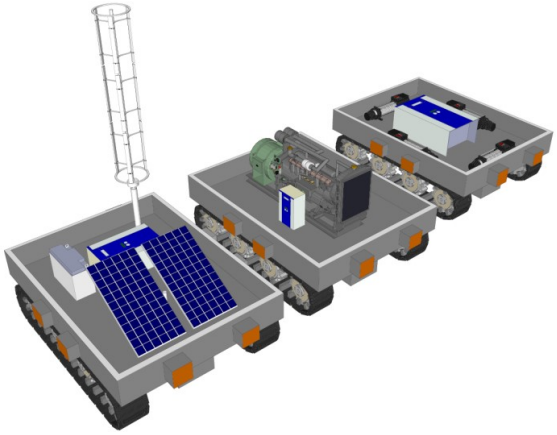

Figure 1.1: Concept image of autonomous robot energy resources with renewable sources, diesel generator, and power conversion © 2019 IEEE.

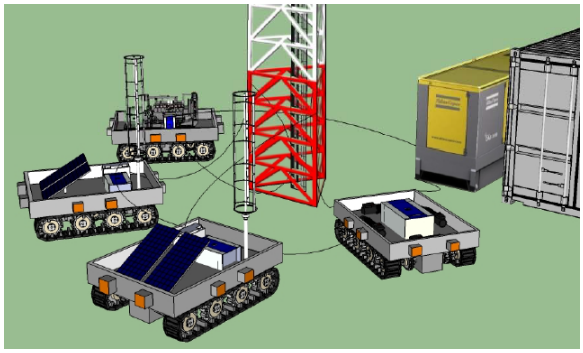

Figure 1.2: Autonomous robot energy resources positioned to power a communication tower © 2019 IEEE.

1.3 Resource Scheduling Optimization

Resource scheduling is a crucial aspect in planning operations for the required collaboration or use of multiple resources. Optimization of the resources' time and energy is vital in achieving a common goal in the best way possible. Resource scheduling applies to many applications, a popular application being scheduling machines for stages in manufacturing processes. For this application, it is important to schedule the machine resources accordingly to maximize their usage. Many factors may play a role in the scheduling of the machines such as how long a specific machine process takes, consumer demand of specific manufactured parts, profit earnings of specific parts, and the overall objective. Machining time and energy is expensive and sub-optimal process scheduling causes extra costs in expenses.

Another application for resource scheduling is in power transfer to other distributed resources to extend their battery life. For example, in using wireless sensor networks (WSN), unmanned aerial vehicles (UAVs), or quadcopters/drones, the on-board battery capacity often limits their operation. Other limited but larger energy resources, such as unmanned ground vehicles (UGVs), can periodically recharge them to extend their battery life and operation. For recharging WSNs, as examined in Chapter 4, periodic recharging to optimize an objective function and satisfy constraints will depend on several factors such as the number of distributed sensors, their locations,

and their respective net power usage. Determining the optimal charging times, with respective durations and locations, is investigated for this case. Similar to WSNs, the optimal use of UAVs in military reconnaissance is examined in this thesis in Chapter 5. As with WSNs, if optimal recharging times, durations, and locations are determined, then the UAVs' missions can be optimally extended, leading to more intelligence gathered, or time saved in this operation. GAs are used to determine optimal solutions in these operational scenarios involving WSNs and UAVs.

1.4 Genetic Algorithms

Because a locally optimal solution for these types of problems and applications described is often insufficient, global heuristic methods such as GAs [6], Particle Swarm algorithm (PSA) [7], and Simulated Annealing [8] are utilized to find a good solution. Although none of these algorithms are guaranteed to converge to a globally optimal solution, they have been shown to provide good global solutions in a variety of applications. Because some problems involve a variable-sized design space (VSDS), zero-one programming can be used to determine optimal design architectures, which is a method applied in this work. This is similar to hiding genes in hidden genes genetic algorithms (HGGA). Since this involves integer constraints, GAs are selected for determining best solutions, for it can easily apply this integer constraint and variety of others. In the context of this thesis, the solutions found using GAs are referred

to as the *optimal solution(s)*. A more in-depth description of GAs is presented in Chapter 2.

1.5 Motivations and Objectives

With the expected increase in use of collaborative autonomous systems to perform tasks normally done by humans or beyond human capability, this thesis presents work supporting the optimal use of these autonomous systems. Specifically, this thesis investigates global optimal solutions to problems presented in mobile microgrids and applications in resource scheduling. The question this work helps answer is "Now that we have the autonomous systems to carry out this task, how to we make the best use their energy resources to complete the mission in an optimal manner?"

1.6 Thesis Organization

In preparing this thesis, papers developed during this research are included. The remainder of this thesis is organized as follows. Chapter 2 examines background details and previous work in optimization and control methods as applied to autonomous mobile microgrids and resource scheduling. Background details on the operation of

GAs is also described. Chapter 3 covers the use of GAs in optimizing mobile microgrid architectures, where several case studies are discussed. Chapter 4 applies the GA optimization method in the scheduling of recharging distributed stationary loads. Chapter 5 extends on Chapter 4 by applying the methods and modeling to optimally recharge UAVs which are performing reconnaissance. Finally, the conclusion of this thesis and future work considerations are presented in Chapter 6.

Chapter 2

Background

2.1 Introduction

This chapter provides background and details for the autonomous microgrid research presented in this thesis. First, supporting information and prior research on autonomous mobile microgrids is presented. Next, details and related research on resource scheduling are explained. Finally, features of the genetic algorithm (GA) optimization method utilized in this work for solving the presented problems are explained.

2.2 Autonomous Mobile Microgrid Architecture Optimization

The vision of the smart grid is to increase the resilience and efficiency of the utility grid in the presence of increasing natural disasters such as hurricanes and floods. This vision includes the integration of resiliency and efficiency into all operations of the present utility grid system, including power generation, transmission, and distribution [9]. A microgrid is an energy system that can operate dependently or independently of the established grid system. The microgrid comprises distributed generation resources (DGRs) such as diesel generators, photovoltaic (PV) arrays, and wind turbines. Microgrids play an important role within the vision of smart grids as they can operate dependently or independently (island mode) of the established grid system by these DGRs. DGRs have solved resiliency issues when problems in the utility grid arise. DGRs, such as diesel generators, are widely used to restore power in buildings or individual homes in times of need. Employing these DGRs nationwide, automatically, systematically, and efficiently to restore power to those in need is a proposed solution to increase grid resiliency. Integrating these energy assets into the utility grid system comes with many challenges, such as the destruction of necessary infrastructure with a natural disaster, along with the communication and control required following the disaster [10]. Despite these challenges, microgrids are

being adapted in many universities, businesses, and residential areas[11].

Autonomous microgrids can help mitigate these issues of unavailable or unreliable electrical grid infrastructure. The center of Agile and Interconnected Microgrids (AIM) at Michigan Technological University is engaging in research to advance the development and understanding of microgrids and energy networks from multiple perspectives. In addition to AIM, other researchers are also working to support this advancements in intelligent microgrids. The work in [12] presents and tests a hierarchical control approach for DC microgrids with simulation case studies based on the Illinois Institute of Technology AC microgrid. Results show the control strategy as a reliable method for a DC microgrid to respond to emergencies and its operation in steady state. Another example of DC control for microgrids is in [13] which demonstrates an autonomous microgrid. The work shows simulated results of multiple DGRs and energy storage systems operating in a variety of modes with smooth transitions between them. These, along with many other control architectures investigated give insight into the utilization of autonomous microgrids [14], [15].

In parallel with ongoing and previous research in power control methods for autonomous microgrids, the additional layer of mobility in energy assets is also being explored and presents additional challenges. [16] describes a multi-agent system of robots for intelligence gathering in post disaster scenarios. Furthermore, [17] presents optimal positioning and real-time allocation of mobile emergency vehicles to reduce

power outage time after natural disasters. [18] also investigates the use of mobile emergency resources for adaptive multi-microgrids for power restoration in extreme conditions such as natural disasters. The optimal microgrid formation method is shown to be effective in serving critical loads and remaining resilient by applying radial or looped topologies with proper positioning of DGRs.

However, previous work in the concept of autonomous mobile microgrids does not consider optimal positioning of energy assets along with optimal network connections in off road, location restricted areas. This detail is important when considering a post natural disaster area with obstacles to avoid or restricted areas of access. These operational area characteristics make a difference in optimal solutions, and the case studies provided reflect this. Much research points to a future with potential for improved and more resilient tactical electrical infrastructure with the use of autonomous mobile microgrids. This is the focus of the work presented in Chapter 3. The next section describes background details and previous work in resource scheduling optimization, which is the focus of Chapters 4 and 5.

2.3 Resource Scheduling Optimization

Resource scheduling has been and continues to be a high interest area of research. Much of this research falls under common terms such as scheduling or task allocation [19]. There are many studies on developed variations of computer science and engineering problems such as Central Processes Unit (CPU) scheduling [20], and the Job-Shop Scheduling Problem (JSP) [21]. These problems are similar in that one needs to decide how to make best use of available resources.

In CPU scheduling, an algorithm or multiple algorithms are used to determine which processes should run in what order. This scheduling is critical to optimize Operating System (OS) functionality. In works such as [22] and [23], an optimized round-robin (RR) CPU scheduling algorithm is proposed which improves the many drawbacks of the traditional RR architecture. CPU scheduling and the Single Machine Scheduling (SMS) problem [24] have a close relationship in that a single machine or resource limits the scheduling of many jobs or processes by only being able to process one job at a time. The JSP extends the SMS problem in the ability to schedule jobs to multiple machines available. [25] presents an improved algorithm for the single machine serial-batch scheduling problem with rejection to minimize the total completion time and the total rejection cost. In [26], GAs are used with modified parameter settings to solve the job shop scheduling problem and is successful in achieving better results

compared to manual scheduling, as applied to real data from a fastener manufacture.

Resource scheduling encompasses another type of optimization problem of task allocation. Task allocation has a close relationship to the work presented in Chapters 4 and 5 because of the described applications of collaborative autonomous systems such as planetary exploration [27] and cooperative surveillance and reconnaissance [28]. Further examples of previous work in collaborative autonomous systems is presented in Chapters 4 and 5.

2.4 Genetic Algorithms

Genetic Algorithms (GAs) are part of a class of evolutionary algorithms in global optimization techniques. GA utilization as an optimization method began in the late 1980s and is still a popular technique used today. This method has been successfully applied to optimize problems in engineering, transportation, scheduling, and many others. Although there is no guarantee that the solutions of GAs are globally optimum, they provide good solutions efficient in many applications [2].

Deoxyribonucleic Acid (DNA) [29] is what makes every person unique. Chromosomes contain many genes which make up the DNA structure. These genes are coded to make specific proteins for cells with different functions in the body. This causes cells

to not read or recognize all genes. For example, a cell in your lungs does not require the same genes as a cell in your eye does to function correctly, so the respective lung and eye cells shut off these genes. The genetic coding can shut off these genes to tell what genes a cell should or should not read [30]. Survival of the fittest and the idea of populations evolving overtime is the basis for the inspiration of GAs in optimization.

GAs use the concept of genes in a chromosome that evolve progressively. A general representation of G_N genes in a chromosome (design variables in a solution) is shown in Figure 2.1, and the general GA process is shown in Figure 2.2. The population-based algorithm creates a population of chromosomes, which represent different solutions in the solution space. Each chromosome comprises genes which represent the design variables which make up a particular solution. The chromosomes go through the processes of selection, crossover, and mutation to explore the solution space and find the optimal solution. The algorithm generates a new population every time the previous population evolves through these processes, which is the next generation. This next generation is the start to the next iteration in the optimization computation. Meeting some stopping criteria such as a maximum number of generations or minimal change in the solutions' cost ends this iterative process. An initial random population is normally used to begin the optimization, but there are options to supplement in solutions, usually some known to the user to be good. Further details of the processes of selection, crossover, and mutation are discussed in the next section.

Figure 2.1: General representation of a single chromosome with G_N genes.

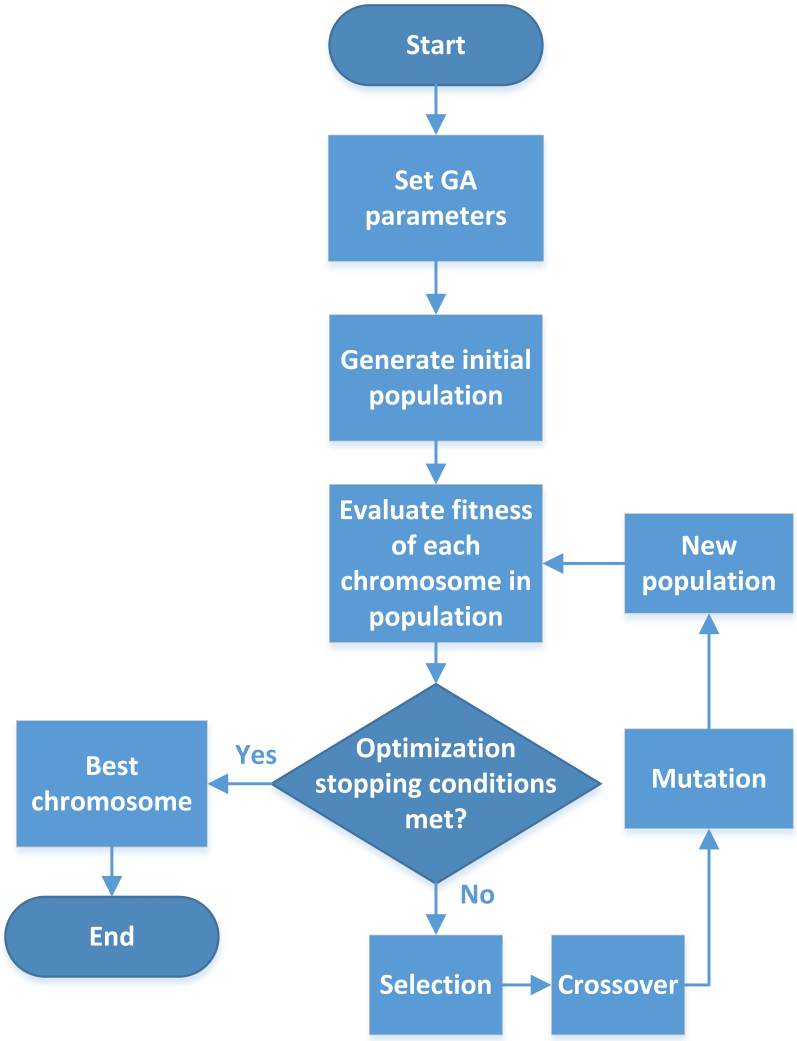

Figure 2.2: Flow chart of genetic algorithms © 2019 IEEE.

Selection is the first main operation in chromosome/gene evolution. It is the process in determining which chromosomes will exist in the next generation and/or will be parents to create children which will exist in the next generation. This selection process has many known methods for use in GAs. One popular method of this is

tournament selection where a percentage of the best performing chromosomes are utilized as parents for creating the next generation or as an elite chromosome to survive to the next generation [31]. Other methods include roulette wheel and stochastic uniform [32].

The crossover operation creates new children from the selected parents to move on to the next generation. This crossover operation also has various methods which can be used. Figure 2.3 shows one of the simplest crossover operations where a segment of each chromosome is exchanged between Parent 1 and Parent 2 to create Child 1 and Child 2. This exchanged chromosome segment is randomly chosen by indexing into the chromosomes at the crossover point. This method is known as a simple crossover [31].

Mutation is the process of replacing a randomly selected design variable (gene) in a design vector (chromosome) with a randomly generated one. This is done to diversify the solutions being searched to expand this search space. This process can be seen in Figure 2.4 [31].

The next chapter presents the problem of determining optimal positioning, connections, and cabling trajectories of energy resources for an autonomous mobile microgrid in the operating field. A variety of test cases are investigated followed by a discussion of the results. Further optimization scenarios are expanded on from the test cases, which are then explained.

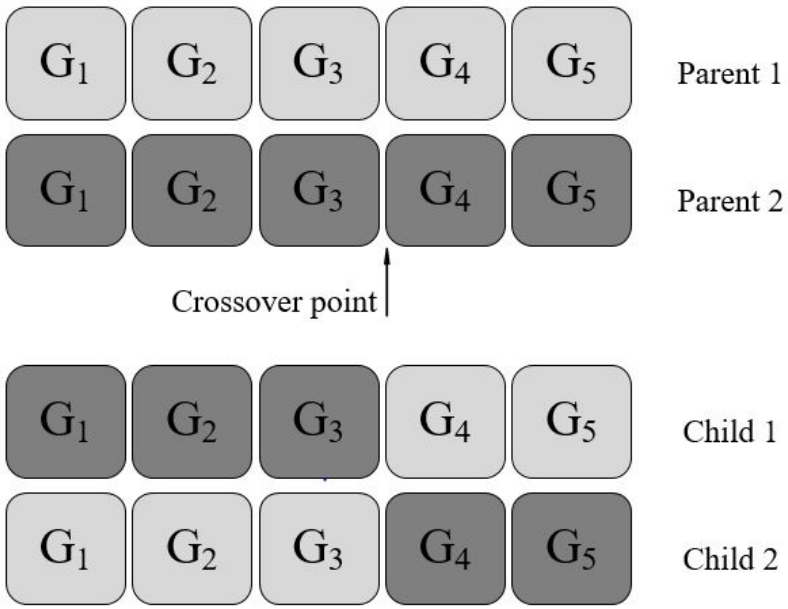

Figure 2.3: Simple crossover operation in GAs.

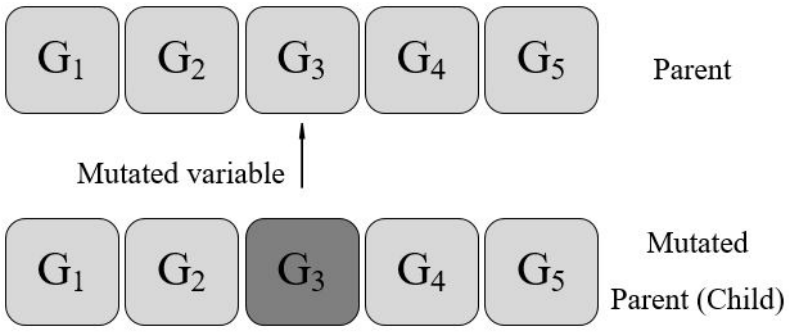

Figure 2.4: Simple mutation operation in GAs.

Chapter 3

Optimal Positioning of Energy Assets in Autonomous Robotic Microgrids for Power Restoration

3.1 Introduction

The application[1] of small connected micro-sources is referred to as distributed energy resources. These micro-sources are energy sources that contribute to the energy utilized in a microgrid system. They can have capacities of 100 kWh, depending on the

[1]The material in this chapter is reprinted in majority from [2]

resource type, and are distributed throughout the power system. They provide power to the end loads and can operate autonomously by disconnecting from the centralized grid. This makes the micro-sources more flexible and resilient while reducing the maintenance and pollution compared to traditional centralized-power distributions. When the distributed energy network is interconnected with storage, control systems, and the loads, the resulting autonomous grid is known as a microgrid. In times of crisis such as natural or man-made disasters, microgrids can disconnect from the main grid and operate autonomously using the local energy generations [33, 34].

Multi autonomous vehicles (multi robots) have been used as mobile sensor networks for surveillance and data collection for different purposes in air, ground, and water [35, 36, 37]. A key contribution of this work is the integration of ground autonomous vehicles and electric power assets to create self-organizing, ad-hoc microgrids. Three autonomous microgrid robots, each with different power network functionality, are shown in Figure 1.1 from Chapter 1. One has renewable energy generation and storage capability, another has a conventional diesel genset and the third contains intelligent power electronics for conversion and connection. After assessing the power requirements and available resources, they would physically organize and electrically interconnect themselves to form a microgrid. For example, Figure 1.2 from Chapter 1 shows the robots interconnected to power a communication tower. The intelligent power electronics and distributed agent based control will regulate power flows at desired voltage and frequency levels to meet load demands.

In a disaster area there can be debris and obstacles in the field of operation [38, 39] where power is to be restored to critical loads [40]. The autonomous microgrid robots that enter the disaster field need to optimally position the energy resources to meet the load requirements for operation as well as to minimize the robotic microgrid assets, such as cabling.

In this work it is assumed that the locations of loads and obstacles are pre-mapped through unmanned aerial vehicles, satellite surveillance [41, 42] or some other technology. The maps and locations are available to the autonomous robots resources to determine optimal positioning, which is the algorithm presented in this chapter. Then, once the optimal positions are determined, the next task is to autonomously navigate the resources to the defined positions, connect the cabling, and energize the network which are techniques presented in [43, 44]. In addition, further resource reallocations and optimizations should be made as disaster recovery efforts occur. As fixed and permanent power and communication links are re-established, the mobile robotic resources can be removed or re-deployed to other areas as conditions change over time. This can be considered as the initial phase of post disaster restoration mission where the primary task is to restore the power in urgent structural zones. The next phases of restoring power in such environments, which usually consist of repairing the structures can take up to weeks or months and is out of the scope of this work. The optimization method in this chapter finds the source location and inter-node connections such that all the loads are fed.

Many algorithms are used in various optimization engineering applications. Genetic Algorithms (GAs) are part of global optimization techniques in evolutionary computation, which have been successfully applied to optimize problems in engineering design, transportation, and scheduling to name a few. GA utilization as an optimization tool began in the late 1980s and continues to be a very popular technique today [31]. Although there is no guarantee that the solution of GAs are globally optimum, they provide local optimum solutions that are sufficient in this application.

In this chapter, the system modeling is expanded in Section 3.2. First, the general rules of numbering the nodes and the lines are explained and then the equations to calculate the voltage at the nodes are discussed. The objective function is defined, and the constraints of the problem are presented. In Section 3.3 the optimization algorithm is described and several test cases are solved in Section 3.4. Section 3.5 discusses the effects of the optimization algorithm setup and the conclusions are then presented in Section 3.6.

3.2 System Modeling

Assume that within a physical field of operation there are N_l number of loads and N_s number of energy sources. The sources represent the mobile robots (Figure 1.1) and the loads represent the critical power usage structure that need to be restored. There

are some obstacles in the field, and the locations of the loads and obstacles are known and fixed. The problem to be solved is to choose the source positions and connections to the loads via lines that optimizes a desired objective function, while satisfying constraints. The objective function can include power loss, bus voltage, power, etc., and some constraints are applied to the problem such as ensuring all loads are served at nominal voltage. The energy sources in this problem will produce DC power, with the problem modeled as a DC circuit. Without loss of generality, the loads can be modeled as resistances and the sources as voltage sources. Figure 3.1 shows an example of the problem with two sources (shown as V_1 and V_2) and three loads (shown as R_1, R_2, and R_3). In the example in Figure 3.1, obstacles are omitted to help simplify the example problem. This problem in the real world would be a natural disaster situation where the loads may be the hospital buildings in a neighborhood and the sources are generators on mobile vehicles. The vehicles can move around the neighborhood to serve the buildings. Since the power storage is limited, they should decide where to have their stations and how to connect to the buildings such that the objective function is minimized.

The general proposed system modeling and optimization procedure is

1. Assume locations and voltage requirements of loads are provided.

2. Assume operators would deploy sufficient generation capacity with robots to support loads.

3. Assume locations of obstacles are provided.

4. Create a candidate electrical nodal network where every load bus is connected to every source bus and every other load bus.

5. The resistance of the lines between buses is proportional to physical distance in the field of operation. Optimize objective function by minimizing line distances with object field or eliminate lines altogether.

3.2.1 Indexing Buses and Lines

In this problem, the numbering of nodes is important to the general mathematical formulation of the system. First, enumerate the nodes in the following order:

† First, source nodes are indexed($i = 1 \dots N_s$).

† Second, load nodes are indexed ($i = N_s + 1 \dots N_s + N_l$).

In the example of Figure 3.1, the source nodes are first indexed (number 1 and 2). Then, the load nodes are indexed (number 3, 4, and 5).

For the next step, enumerate the lines (resistance connections). These are the cables connecting the source nodes to the loads. Start from the first source and connect it to the loads. The indices of the resistance in each line is in accordance of the

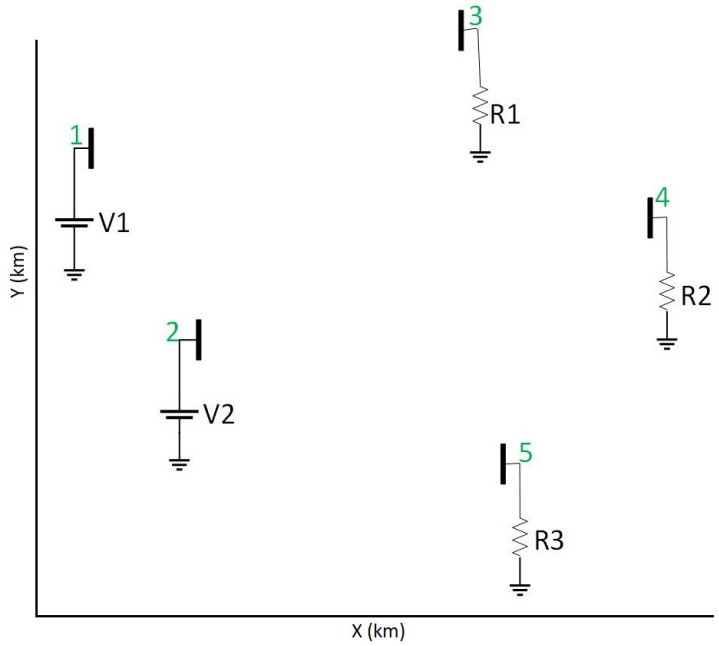

Figure 3.1: node numbering starts from the first source to the last source and then continues from the first load to the last load.

corresponding node. For example, in Figure 3.2 the node number 1 is connected to the node number 3 (R_4), node number 4 (R_5), and node number 5 (R_6). Then, the same is done for the next source and it continues until all sources are connected to the loads.

In the last step, the load nodes are connected to each other, starting from the smallest node number. As shown in Figure 3.2, node number 3 is connected to node number 4 (R_{10}) and 5 (R_{11}) and node number 4 is connected to the only remaining node which is node number 5 with resistance R_{12}. By this sequencing the definition of an admittance matrix [45] is defined and will be presented further in Section 3.2.2.

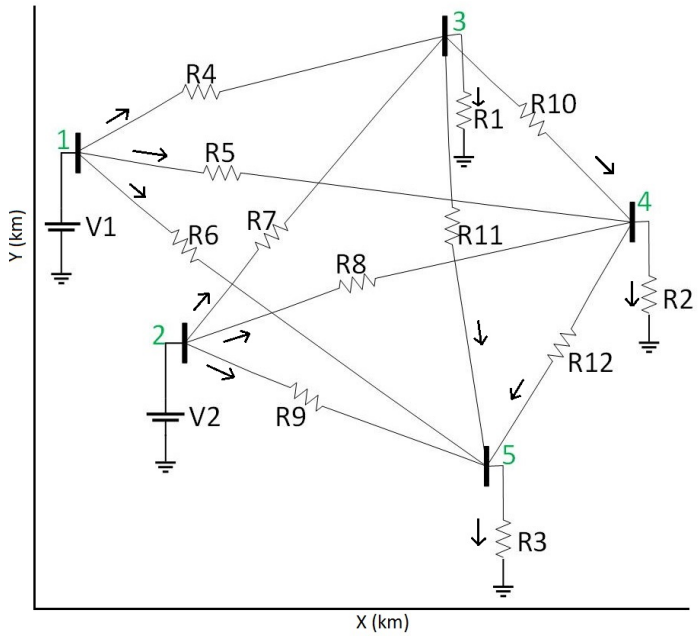

Figure 3.2: line numbering starts from the first source to each load, then second source to each load, then from each load to the remaining loads, from the lowest index to the highest.

3.2.2 Formulate Nodal Bus Admittance Matrix

Consider the system in Figure 3.2 in a grid map. Assume that the total number of nodes is m and the total number of lines is n. The nodal bus equation that relates bus voltage to the injected currents is

$$\mathbf{I} = Y_{bus}\mathbf{V} \tag{3.1}$$

where the admittance matrix [45] can be found from

$$Y_{bus} = A[Y_y]A^T \tag{3.2}$$

where Y_y is

$$Y_y = \begin{bmatrix} \frac{1}{R_1} & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{R_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{R_n} \end{bmatrix} \tag{3.3}$$

where R_i is the ith resistance in the map. The matrix A is the $(m+1) \times n$ incident matrix whose row are the node location and columns are the branch lines.

$$A = \begin{bmatrix} a_{0,1} & \dots & a_{0,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \tag{3.4}$$

$$a_{i,j} \in (-1, 0, 1) \quad \forall i, j. \tag{3.5}$$

This matrix shows the routing of the currents with respect to the node and branch lines. The elements of A are 0 if no connection exists between node and line, 1 if the line is connected to the node and current is leaving the node, or -1 if line is connected to the node and current is entering the node. In addition, note that the first row of A is ground or the node 0. This ground node is at zero voltage, and other voltages are measured with respect to it.

For example, for the system shown in Figure 3.2, the incident matrix from [45] would

be

$$M = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix} \tag{3.6}$$

Note that the rows represent the nodes including ground and nodes 1 to 5. The columns represent the lines (resistance) connected to the nodes. The label notations are shown in Eq. 3.7 for better understanding. As an example, consider the third row, N_2. The first element in this row shows that R_1 is not connected to N_2 and its corresponding value in the matrix is 0. For the element in third row and seventh column (N_2-R_7), the value is 1 which implies that R_7 is connected to N_2 and the current is leaving node N_2. This corresponds to the definition of matrix A and the circuit in Figure 3.2. The rest of the matrix is build similarly.

$$A = \begin{array}{c} \\ gnd \\ N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} R1 & R2 & R3 & R4 & R5 & R6 & R7 & R8 & R9 & R10 & R11 & R12 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix} \end{array} \tag{3.7}$$

The direction of the current is always from the sources to the loads. For consistency, for the connections between the loads, it is assumed that the current flows from the nodes with a smaller index to the nodes with higher index (e.g. from N_3 to N_4 via R_{10} in the example in Figure 3.2). This assumption would not affect the final designed circuit as the connections are optimized by the optimization algorithm. Hence, in the final designed circuit there might be a connection with a current flow from a higher indexed node to a lower indexed node. The details of the optimization process are explained in Section 3.3.

The voltage vector of all the nodes, $\mathbf{V}$, is defined as:

$$\mathbf{V} = \begin{bmatrix} 0 \\ \mathbf{V}_s \\ \mathbf{V}_b \end{bmatrix} \tag{3.8}$$

where $\mathbf{V}_s$ is the voltage vector of the source nodes and $\mathbf{V}_b$ is the voltage vector of the

load nodes. The voltage at each node can be calculated as follows [46]:

$$\mathbf{V}_b = Y_{22}^{-1}\mathbf{I}_b - Y_{22}^{-1}Y_{21}\mathbf{V}_s \tag{3.9}$$

where

$$\left[\begin{array}{c} \mathbf{I}_s \\ \hline \mathbf{I}_b \end{array}\right] = \left[\begin{array}{c|c} Y_{11} & Y_{12} \\ \hline Y_{21} & Y_{22} \end{array}\right] \left[\begin{array}{c} \mathbf{V}_s \\ \hline \mathbf{V}_b \end{array}\right] \tag{3.10}$$

Note that in these equations $\mathbf{I}_b = \mathbf{0}$ because no current is injected in the load nodes, as instead occurs for source nodes. The direction of the current in the source to load connections are always from the source to the load as in Eq. 3.4. The direction of the current in the load to load buses is not known a priori. Note that although the direction changes the signs in Eq. 3.4, it does not affect the solution of the optimization. The voltage at the nodes are calculated according to Eq. 3.9 and Eq. 3.12, where A shows up in Y_{bus} matrix. Since Y_{bus} is a quadratic form of matrix A (Eq. 3.2), the nodes voltages are disassociated to the sign of matrix A. In fact, the correct direction of the currents will be determined after the voltage of each node is calculated.

3.3 Optimization

The resistance of each line is proportional to its length, defined as

$$R_i = \rho L_i / S \tag{3.11}$$

where ρ is the electrical resistivity of the cable material, L is the length of the line, and S is the cross-sectional area of the material. For simplicity, the optimization cost function used in this chapter is defined as minimizing the resistance in the cables. This equates to the objective function

$$J_1 = \sum_{i=1}^{N_t} L_i \tag{3.12}$$

where L_i is the length of the shortest ith line between two nodes if they are connected and N_t is the total number of connections. Hence, the problem is equivalent to minimizing the sum of total cable lengths subject to voltage constraints and the voltage is calculated via Eq. 3.9. Keep in mind that the load lower and upper voltage constraints force the optimizer to find solutions with connection lengths in a certain range. The voltage drop in the cable increases as the cable length increases. Hence, the constraints prevent the optimizer from finding solutions with connections that are "too short" (upper limit constraint) or "too long" (lower limit constraint). From

a practical aspect, the voltage constraint reflects a real word scenario where there would be a voltage range requirement at the load that needs to be met. This choice of objective function for the optimized source positioning is a simple first choice. However, a designer may wish to include terms for load priority, source reliability or source rated capability. These additional terms will be explored in future work. The details of transforming the problem unknowns into variables and the solver algorithms with the given objective function (Eq. 3.12) are presented in Sections 3.3.1 and 3.3.2 and four different example test cases are investigated in Sections 3.4.1-3.4.4.

3.3.1 Shortest Path Algorithm

To calculate the length of the wire between any two nodes on a grid field in the presence of obstacles, a shortest distance algorithm is used. Obstacles are defined as nodes that the path can not pass through. The algorithm chosen here for shortest distance with obstacles is A^* (A-star) [47]. This algorithm is an extension of Dijkstra's algorithm with two sets, fringe (open) and closed. The fringe set records the nodes adjacent to those already evaluated and the closed set records the nodes already evaluated. At the beginning, the fringe set only contains one node, the starting point, and the closed set is empty. At each iteration of the algorithm, A^* selects a point of the fringe set to be the next node. To select that node, A^* selects the path

that minimizes

$$f(n) = g(n) + h(n) \tag{3.13}$$

where $g(n)$ is the cost (distance) of the path from the starting node to node n and h(n) is the estimation of the cost from the node n to the goal node, where in this application, the starting node is the location of the energy source and the goal node is the desired load for it to energize. If two loads are connected, the start and goal nodes would be these two nodes. Node n is the current node being evaluated and the sequence of optimal n nodes forms the shortest path from the start to the goal node. A common method which is selected in this implementation is to calculate $h(n)$ as the Euclidean distance from node n to the goal node. Note that the calculations of the A^* are repeated for each corresponding start and end goal until the shortest paths for all the connections are found. The general A^* algorithm procedure is shown in Figure 3.3 and the details are as follows:

1. Select the node from the fringe (open) set with the lowest f-value. If there are two or more nodes with the same f-value, select the node with the lower h-value.

2. If this newly selected node is the goal node, retrace the path to the start node to find the shortest path from start to goal node.

3. If this newly selected node is not the goal node, evaluate its neighboring nodes

and add them to the open set.

4. If one of the neighboring nodes has already been previously evaluated and the current evaluation results in a lower f-value, then update it. When updating nodes, record the parent node.

5. Add the selected node to the closed set and repeat this process.

3.3.2 Genetic Algorithm

The variables of the problem are the positions of the sources and the connections between nodes. The Genetic Algorithms (GAs) are chosen as the base solver for the problem. In GAs, the variables are simulated as genes and a set of variables (a solution) is called a chromosome. A population consists of several chromosomes that evolve via selection, crossover, and mutation operators through generations to converge to a final solution. In the current problem, for N_s number of sources and N_l number of loads, the total number of possible connections is

$$N_t = N_{sl} + N_{ll} \tag{3.14}$$

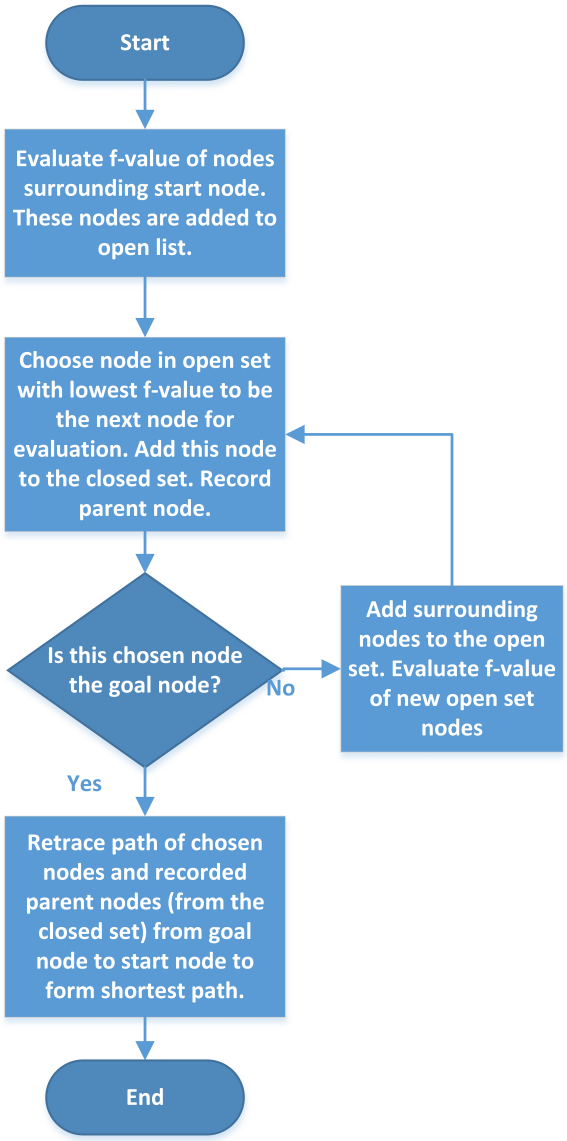

Figure 3.3: Flow chart of A^* algorithm.

where N_{sl} is the number of source to load connections and N_{ll} is the number of load to load connections as

$$N_{sl} = N_s \times N_l$$

$$N_{ll} = \begin{cases} \binom{N_s + N_l}{2} - \binom{N_s}{2} - N_{sl}, & \text{if } N_s > 1 \\ \binom{N_s + N_l}{2} - N_{sl}, & \text{otherwise} \end{cases} \tag{3.15}$$

The total number of variables is $N_s + N_{sl} + N_{ll}$ where the first N_s variables determine the location of the sources and the rest of the variables determine whether a connection exist or not. The last $N_{sl} + N_{ll}$ genes are binary variables that can have values of 0 and 1. If a gene is 1, the corresponding connection is inactive (there is no connection between the corresponding nodes) and if the gene is 0 the connection is active. In some sense this is similar to hiding genes in Hidden Genes Genetic Algorithms (HGGAs) [48, 49, 50]. HGGAs were recently introduced to handle variable sized design space problems via the introduction of tags for the genes [48]. In these algorithms, a binary tag is assigned to the genes that can be hidden or active. If the tag is 1, the corresponding gene is hidden and if it is 0, the gene is active. Two sample solutions with one source and three loads are shown in Figures 3.4 and 3.5. The length of the chromosomes for both solutions is the same, while they have a different number of active variables. In Figure 3.4 solution, all connections are active (all the connection genes have values of 0) and in Figure 3.5 solution, four connections are active (the values of two connection genes are 1). Note that the first variable in both solutions is the position of the source, which is always active.

The general optimization solution procedure is shown in Figure 2.2 in Chapter 2 and the details are as follows

1. Read the first N_s values of the genes to get the position of the sources.

2. Read the rest of the genes to determine which connections are active. Calculate

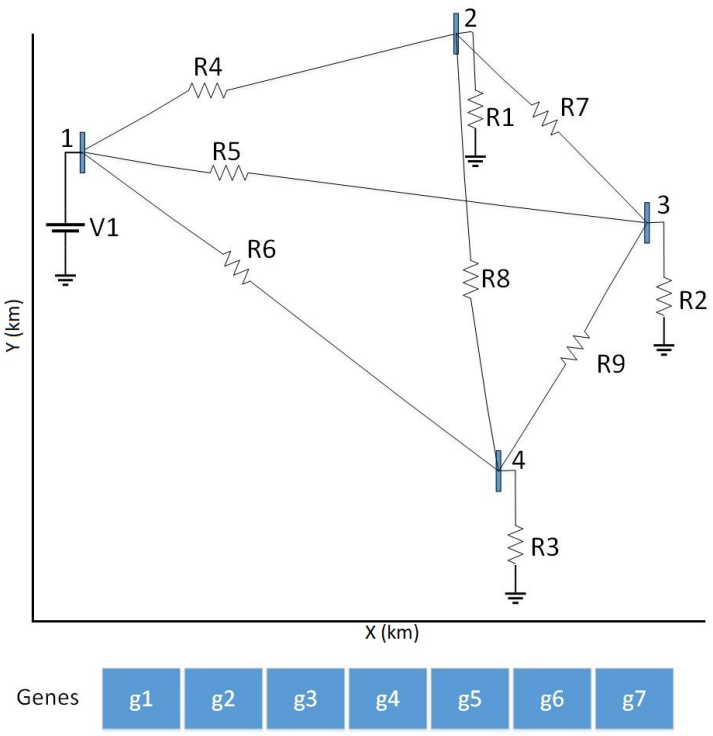

Figure 3.4: A solution with no hidden connections.

the shortest distance for the active connections based on the A^* algorithm.

3. According to Eq. 3.11, the calculated distances are related to the resistances. If a connection is hidden, set the equivalent resistance as infinity ($R_i = \infty$). This corresponds to the physical meaning that if there is no connection between two nodes, there is no current between them.

4. Produce Matrix A according to Eq. 3.4.

5. Calculate Y_y matrix in Eq. 3.3.

6. Calculate Y_{bus} matrix in Eq. 3.2.

7. Solve Eq. 3.9 to calculate the voltage constraint.

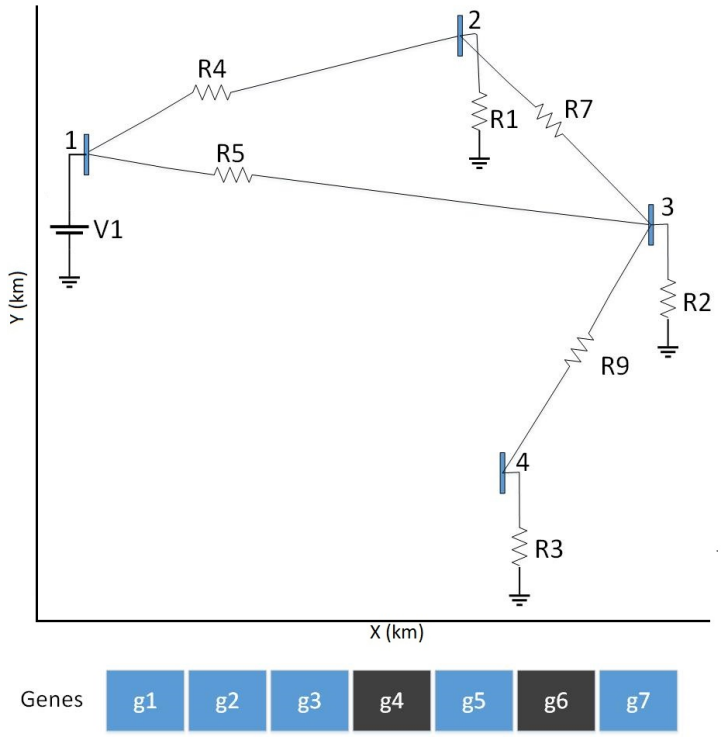

Figure 3.5: A solution with two hidden connections.

8. Calculate the objective value according to Eq. 3.12.

Note that in the simulations no duplicate location is allowed, i.e. only *one* source is allowed to be placed at any location and also the obstacles positions are forbidden for source allocation. The objective function evaluation will reflect poorly if this constraint or any others such as voltage are not satisfied. For each solution found by the algorithm, the voltage is checked via Eq. 3.9 and if it is not in the voltage constraint range, the algorithm penalizes that solution by giving it high cost.

3.3.3 GA Option Setup

The population size for all the cases is set to $N_{pop} = 10N_{dv}$ where N_{dv} is the number of design variables in each case. The Elite count is set to $floor(0.1\ N_{pop})$ and the number of generations and the stall generation for each case is shown in Table 3.1. The function tolerance for stopping criteria is $f_{tol} = 10^{-5}$ and the crossover fraction is 0.7 % in all the cases. The stall generations means that the algorithm will stop if the average relative change in the best fitness value in S_g consecutive generations is less than or equal to f_{tol}. In all the test cases the load resistances are $R_{load_i} = 10\ \Omega$, the source voltages are 60 V. The voltage at each load is constraint to be in the range of [43.2 60] V. The resistance of the connecting wires are 3.28 Ω/km and the field is 1 $km \times 1\ km$. In the first case, there are no obstacles in the field, but for other cases, it is assumed that there are known obstacles. The variables of the problem are the location of the sources and the connection between sources to loads and loads to loads. If a connection between two nodes exist, the wiring length and path is calculated based on the A^* algorithm. The field is coordinated in grids of 101×101 and the resistance of each connection wire is computed as 3.28 $\Omega/km \times d$ where d is the distance of the wire in kilometers.

Table 3.1
Genetic algorithm options for different cases.

Case	Number of Generations	Stall Generation Limit (S_g)
1	50	20
2	100	50
3	100	50
4	200	50

3.4 Test Cases

Four different test cases are investigated in this section. The complexity of the problems increase in each case by adding obstacles and increasing the number/position of the loads. Each load represents a center in need of voltage/power in an affected area in actual natural disaster situations. The obstacles represent the debris, closed roads, or any other inaccessible areas as a result of a disaster zone. The examples shown here represent the optimal layout plan for a given case. Once the positions are chosen through this method, the autonomous robots would need to maneuver from their starting positions to connect the cabling and park in the designated optimal position before the system can be energized, which is outside of the scope of this work, but can be found in [44].

3.4.1 Case A: One Source, Four Loads, No Obstacles

The first case is a simple example to investigate whether the algorithm produces the feasible solutions. Assume that in a field with no obstacles there are four loads to be supplied and only one source is available. The field is square, and each load is at a corner. The number of design variables in this case is 11. These variables include the location of source (1 variable), the connection between the source and the loads (4 variables), connection between load 1 to loads 2, 3, and 4 (3 variables), connection between load 2 to loads 3 and 4 (2 variables), and connection between load 3 to load 4 (1 variable). The lower bound for the first variable is 1 and the upper bound is 101×101. All other variables can have values of 0 or 1 where 0 indicates the corresponding connection is active and 1 indicates the corresponding connection is inactive.

The problem is solved, and the solution is as Figure 3.6. The source is located at the center and is connected to all the load while all the load to load connections are inactive. This result was expected as we know that this grid produced the least total length of the wires while satisfying the constraints. The voltage and power found for each load is given in Table 3.2.

Table 3.2
Bus voltage and load power solutions for Case A.

Load	Voltage (V_{dc})	Power (W)
1	48.7950	238.0950
2	48.7950	238.0950
3	48.7950	238.0950
4	48.7950	238.0950

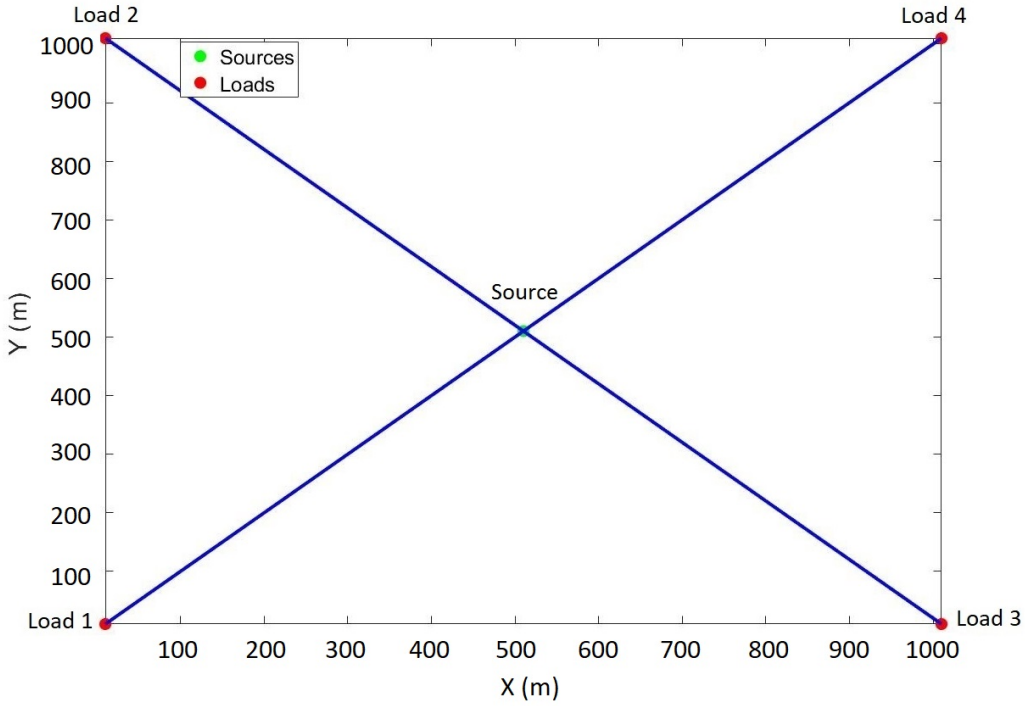

Figure 3.6: Configuration solution of Case A.

3.4.2 Case B, Two Sources, Four Loads, With Obstacles

For a more complicated case, assume that there are obstacles in the field. There are two sources to be placed and connected and four loads which are placed one on each corner of the field. There are a total of 16 design variables for this case. The

solution to this problem shown in Figure 3.7. The first source is located next to the second load and the second source is connected to the first, third, and fourth loads. There is no connection between the loads themselves. In this example, the algorithm managed to find the line-of-sight for the second source to the first, third, and fourth loads. This is a valuable aspect of the algorithm in evaluating its performance in finding the connections with the shortest lengths. The voltage and power found for each load is given in Table 3.3.

Table 3.3
Bus voltage and load power solutions for Case B.

Load	Voltage (V_{dc})	Power (W)
1	48.4806	235.0369
2	59.8058	357.6731
3	52.8181	278.9756
4	47.6522	227.0734

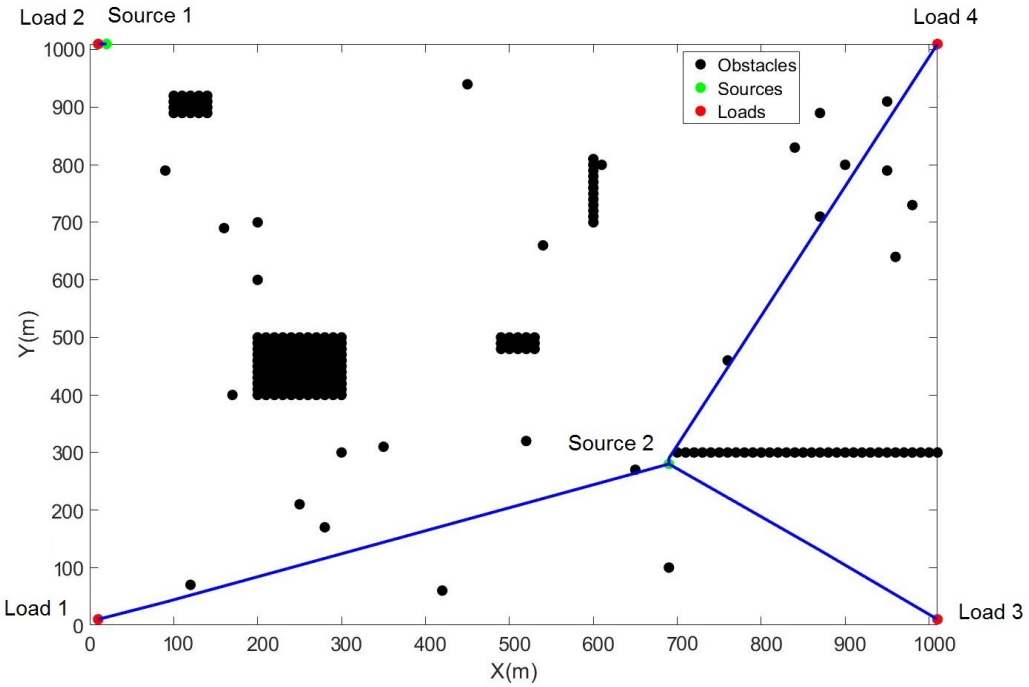

Figure 3.7: Configuration solution of Case B.

3.4.3 Case C: Two Sources, Five Loads, With Obstacles

In this case, there are 5 loads and 2 sources. The source voltage, load resistances, and load voltage constraint is similar to previous cases. The number of design variables is 22 for this case. The results of the simulation are shown in Figure 3.8 and the voltage and power at each load is given in Table 3.4

Table 3.4
Bus voltage and load power solutions for Case C.

Load	Voltage (V_{dc})	Power (W)
1	58.380	340.82
2	50.904	259.13
3	59.806	357.67
4	46.267	214.07
5	51.019	260.29

3.4.4 Case D: Three Sources, Ten Loads, With Obstacles

For a more complicated problem, there are 10 loads and 3 sources which results in 78 design variables. The results are shown in Figure 3.9 and the voltage and power at each load is given in Table 3.5.

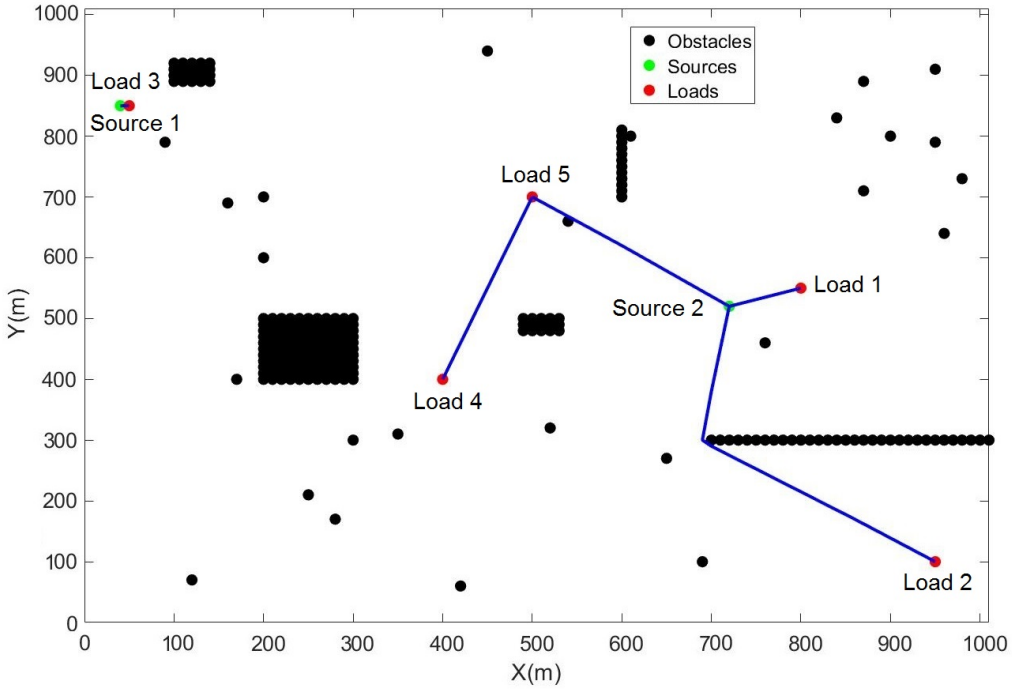

Figure 3.8: Configuration solution of Case C.

Table 3.5
Bus voltage and load power solutions for Case D.

Load	Voltage (V_{dc})	Power (W)
1	54.4631	296.6228
2	56.9498	324.3279
3	53.9759	291.3395
4	54.4499	296.4793
5	59.5674	354.8280
6	49.3561	243.6025
7	54.3362	295.2417
8	54.7357	299.5993
9	59.6299	355.5724
10	52.5618	276.2744

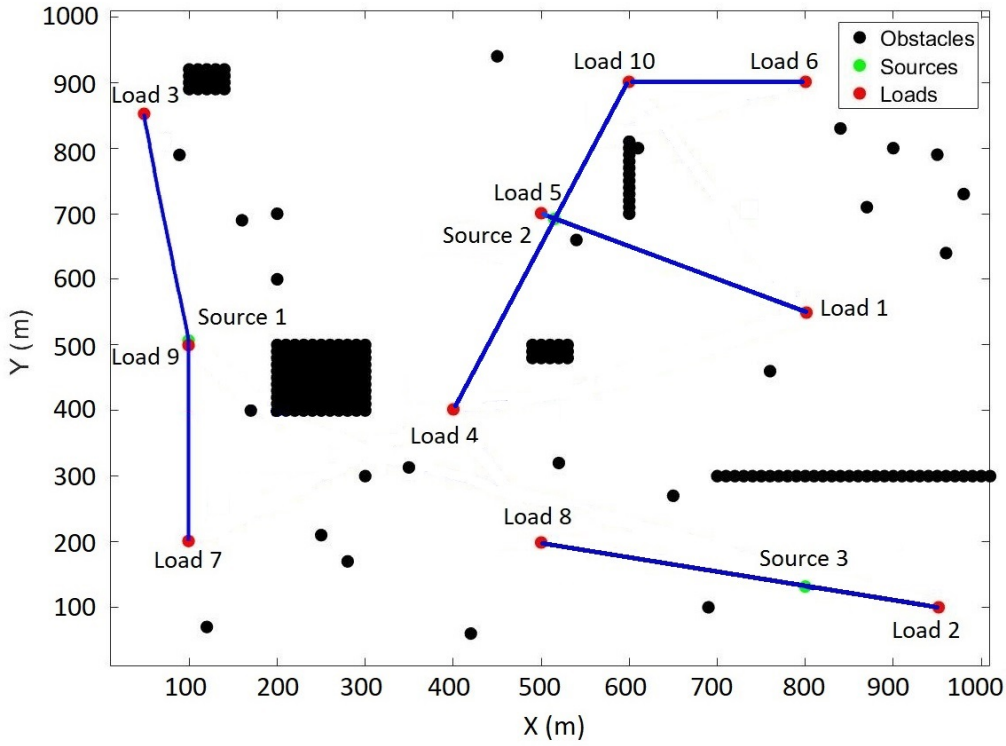

Figure 3.9: Configuration solution of Case D.

3.5 Discussion

In this section, more investigations are done on the optimization algorithm setup, including constraint and objective function. The effects of voltage constraints adjustments on the solutions of Section 3.4 are investigated. Moreover, alternative objective functions are investigated to examine the effectiveness of the proposed algorithm.

3.5.1 Voltage Constraint Adjustment

To investigate the effects of load voltage constraints on the structure solution, an example test case is presented here. The simulations are repeated by decreasing the upper limit on the voltage constraint from 60 V (GA setup 1) to 52.8 V (GA setup 2), while all the other parameters and variables of GA remain the same. The objective function values of all the test cases in both setups are presented in Table 3.6. Note that these objective function values represent the total length of the line connections and are in meters (Eq. 3.12).

Table 3.6
Comparison of the cost function values J_1 for the two GA setups.

	GA setup 1	GA setup 2
Case A	2828.4	2828.4
Case B	1958.3	2000
Case C	1246.1	2133
Case D	2188.8	3955.28

The results of Case A are the same in both setups. However, for the other three cases, there is an increase in the connection lengths in GA setup 2 compared to GA setup 1. As an example, Figure 3.10 shows the circuit configuration of Case D in GA setup 2. The "long" connections of this solution compared to Figure 3.9 is noticeable. In GA setup 2, the source voltages are 60 V while the upper limit on the voltage constraint of the loads are 52.8 V. On the other hand, the voltage drop from the source to the load increases by increasing the cable length. The constraint implies that the

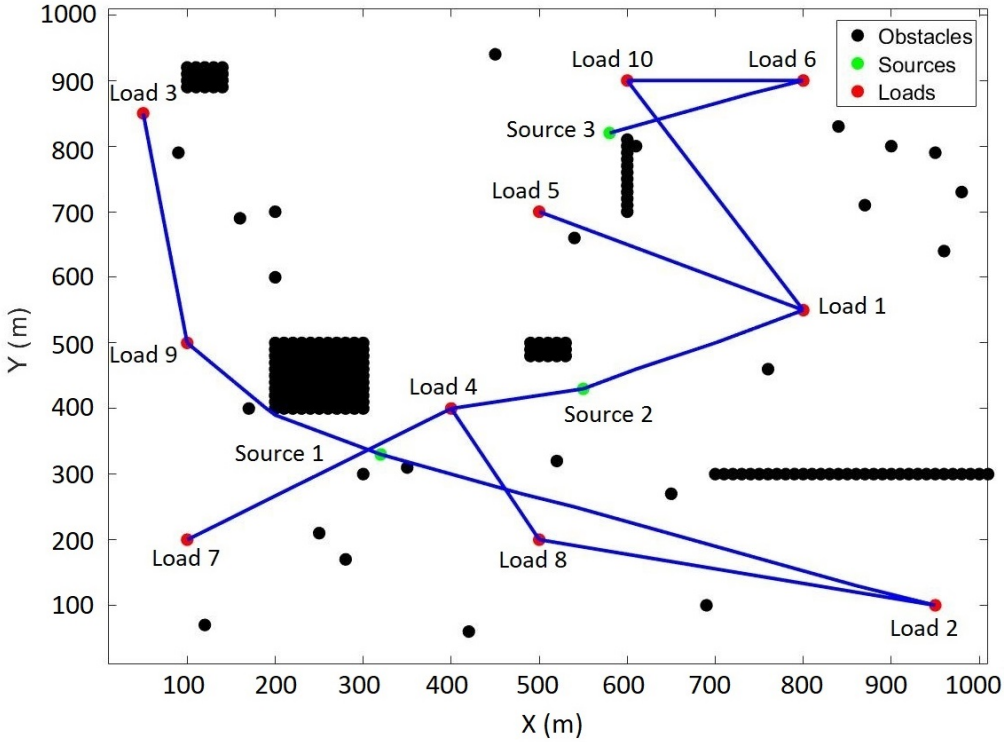

Figure 3.10: Configuration solution of Case D with maximum load voltage constraint of 52.8 V.

source's voltage would be greater than the load's voltage for any possible connection, which causes a minimal distance to be maintained between the source and the load to drop the necessary voltage down to meet the constraint. The results show that by adjusting the optimization parameters and constraints, designers can achieve circuits that satisfy their goals better.

3.5.2 Alternative Objective Functions

Two more objective functions are investigated to study the effectiveness of the algorithm. The second objective function is defined as:

$$J_2 = \sum_{i=1}^{N_t} L_i - 10 \sum_{j=1}^{N_l} P_j \tag{3.16}$$

where P_j is the power at the jth load in watts and is calculated as follows:

$$P_j = V_{b_j}^2 / R_j \tag{3.17}$$

where R_j is the constant resistance of the jth load (connected to ground). With this objective function, the algorithm tries to minimize the total connection length while maximizing the total power at the loads. The weight of 10 implies that the total power has a higher priority compared to cable length. The third objective function is defined as:

$$J_3 = \sum_{i=1}^{N_t} L_i + max(\vec{D}_r) \tag{3.18}$$

where $\vec{D}_r$ is the vector of traveled distance of robots (sources) from their deployment point to their optimal location. It is assumed that all the robots are initially at $(x, y) = (500, 20)$ m in the field and they are deployed at the same time. Assuming the robots have similar and constant velocities, this objective function will minimize

the connection lengths while minimizing the maximum time it takes for the robots to be at their operational (optimum) location. The sooner the robots are at their optimal locations, the sooner the power restoration operations start, which is very important in post-disaster missions. The results of the three objective functions are shown in Table 3.7 for Cases C and D. The lengths are in meter and the power is in watts.

Table 3.7
Comparison of the results for J_1, J_2, and J_3 objective functions for Cases C and D.

Case	Objective Function	$\sum_{i=1}^{N_t} L_i$	$\sum_{j=1}^{N_l} P_j$	$max(\vec{D}_r)$
C	J_1	1246.1	1432.0	701.0
	J_2	1767.7	1590.0	1024.4
	J_3	1512.2	1507.7	430.8
D	J_1	2188.8	3033.9	851.7
	J_2	3029.6	3109.4	683.1
	J_3	2545.0	2631.6	732.3

Comparing J_1 and J_2, it is observed that the total cable length and the total power at the loads is higher for J_2. This is due to the added power term with a weight of 10 in Eq. 3.16 compared to Eq. 3.12. Moreover, analyzing the result of J_3 and J_1 shows that the algorithm was able to find a solution with less travel distance for the sources by compromising the total cable length. Comparing the results of J_2 and J_3 shows that the total power at the loads in both Case C and D is larger for J_2, with an added cost of cable length. J_3 shows solutions with lower total cable lengths plus max source travel distance. Note that for Case C, J_3 results in the lowest max source travel cost but not for Case D. This is due to J_3 viewing the cable length and max

source travel distance as equally weighted costs. J_3 finds a lower overall cost in Case D by allowing the max source travel distance to be larger, as compared to J_2, to find a lower cable distance cost. The results of Cases C and D with J_2 and J_3 objective functions are shown in Figures 3.11-3.14.

J_1, J_2, and J_3 can be formulated in a single objective function as:

$$J_4 = w_1 \sum_{i=1}^{N_t} L_i + w_2 \sum_{j=1}^{N_l} P_j + w_3 max(\vec{D}_r) \tag{3.19}$$

where $\vec{w} = [w_1\ w_2\ w_3]$ are the weight coefficients of total cable length, total power at the loads, and the sources' travel distance, respectively. J_1 is represented by the first of the three terms and can be represented with $\vec{w} = [1\ 0\ 0]$. J_2 is represented in the first two terms, with $\vec{w} = [1\ -10\ 0]$. J_3 is represented by the first and third terms with $\vec{w} = [1\ 0\ 1]$. By adjusting these weight coefficients, the designers can set up the best objective function based on their requirements and priorities. The examples provided here are just a demonstration of the performance and effectiveness of the proposed algorithm.

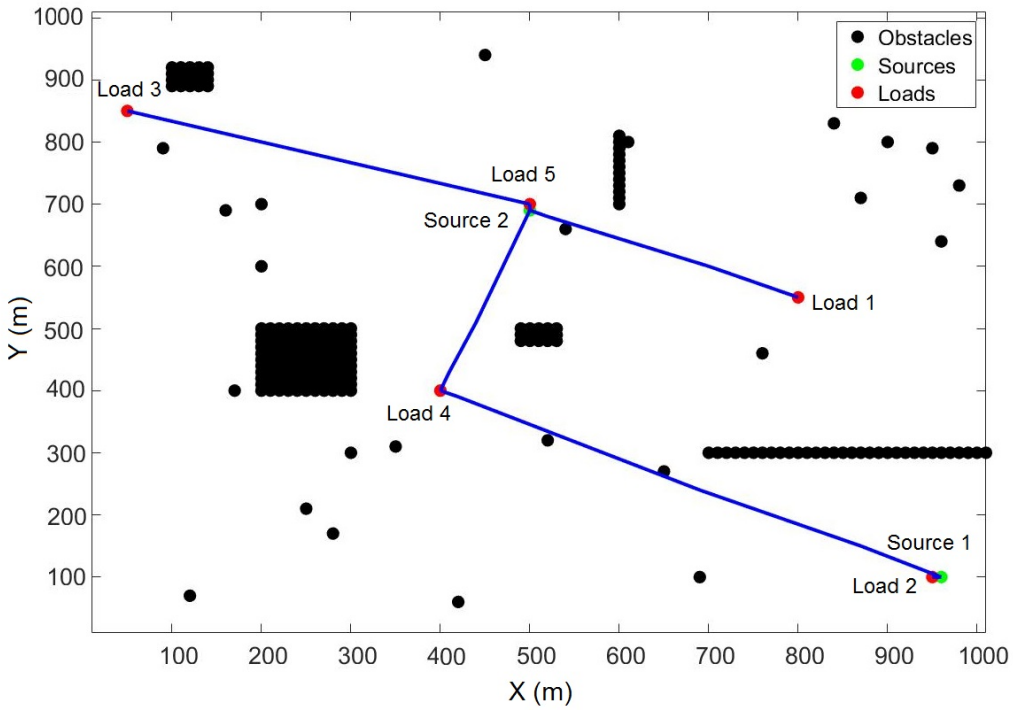

Figure 3.11: Configuration solution of Case C- J_2.

3.6 Conclusion

In this chapter, a general framework and solution methodology was presented to optimally position energy sources within a field with loads and obstacles. In addition, the interconnection between the sources and loads were optimized to achieve the most efficient operation within a given set of constraints. In the proposed algorithm, no prior knowledge about the location of the sources or the connections structure is needed. The results of the example test cases show valid and reasonable results, however, further refinement can be added to include load priority, source reliability, etc. In addition, the general framework for the problem including the system modeling,

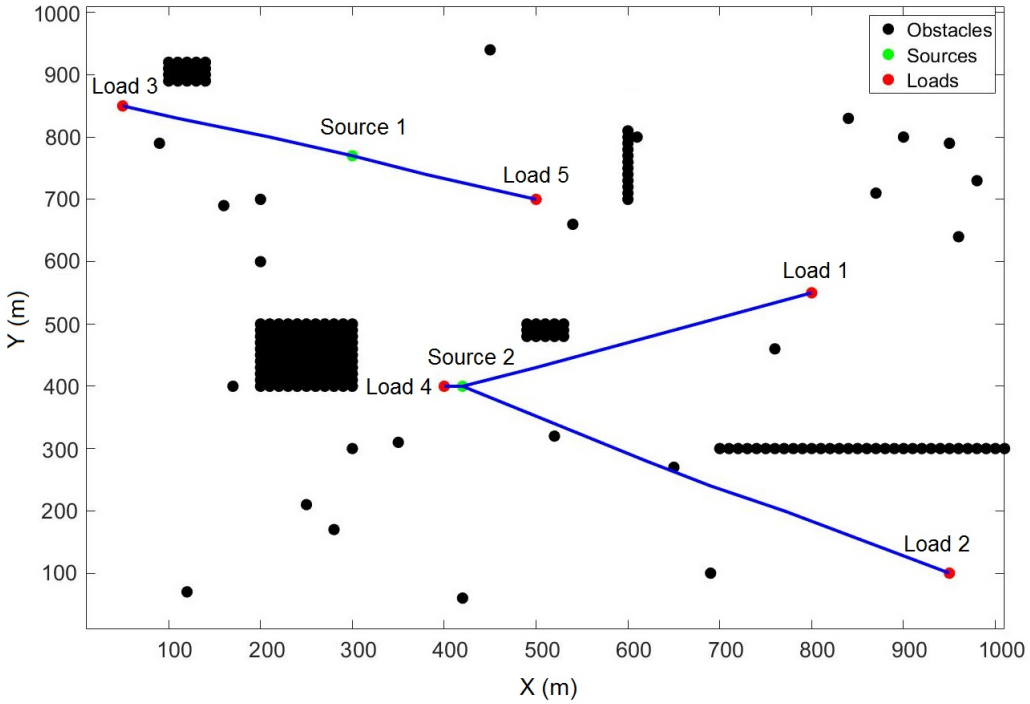

Figure 3.12: Configuration solution of Case C- J_3.

objective functions, constraints, and genetic algorithm solution can also be extended from the DC distribution used in this work to include AC and hybrid AC/DC system as well.

In the next chapter, the concept of a sub-microgrid system consisting of a UGV traversing to and transferring power wirelessly to distributed loads is presented. A system simulator is developed in which several algorithms to control the charging of the loads with a UGV are applied. The algorithms are evaluated based on costs associated with load operation and UGV traversal time. The results demonstrate the robustness of the system simulator, as well as the advantages and disadvantages of the algorithms presented.

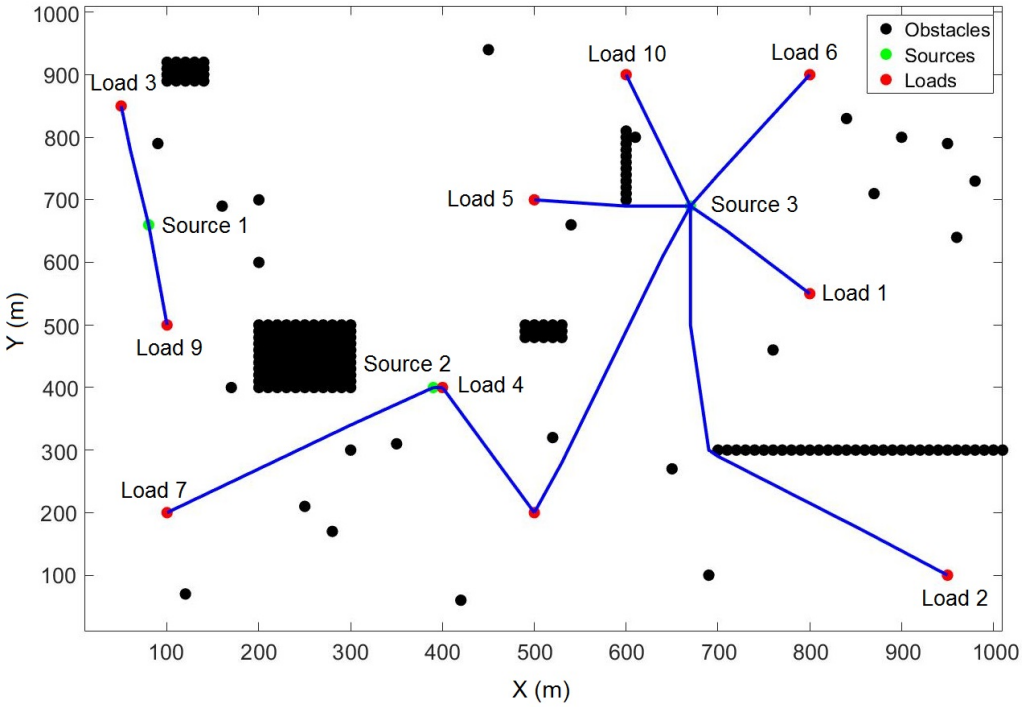

Figure 3.13: Configuration solution of Case D- J_2.

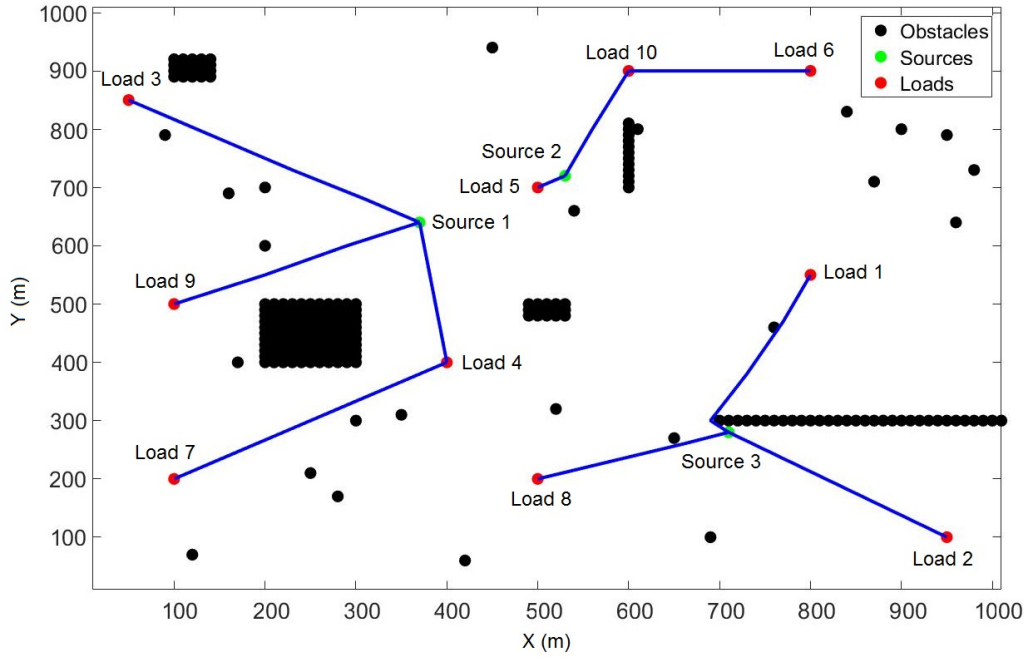

Figure 3.14: Configuration solution of Case D- J_3.

Chapter 4

Recharging of Distributed Loads via Schedule Optimization with Autonomous Mobile Energy Assets

4.1 Introduction

The development [1] and utilization of autonomous mobile agents has substantially increased over the past decade. This has driven a better understanding of performance in these systems to increase their utility. One relatively new application for the use

[1]The material in this chapter is reprinted in majority from [3]

of these autonomous mobile robots is in microgrid systems [51], [52]. These mobile microgrids use distributed energy resource robots such as photo-voltaic (PV) arrays with energy storage, diesel generators, and controllable power electronics. Three such mobile resources are shown in Figure 1.1 from Chapter 1. A mobile microgrid system can adapt to changes in the environment to optimize its applicability, depending on load demand. The agents can connect and work together to power other systems such as a communication tower for example, shown in Figure 1.2 from Chapter 1. This application of collaborative mobile energy resources exploited for microgrids can be extended to a myriad of unique missions. This includes the use of these unmanned ground vehicles (UGVs) in recharging other distributed resources such as wireless sensor networks (WSN) [53] or unmanned aerial vehicles (UAVs) to extend their mission life in operations such as planetary data collection, surveillance, or reconnaissance. Along with properly developing these UGVs for missions, a better understanding of how the agents should collaborate to fulfill a mission with limited resources needs to be considered. This chapter proposes a method to optimally allocate resources for mission fulfillment. Specifically, this chapter investigates how resources should be allocated to sustain the operational life of distributed loads, which could represent WSN, UAVs, etc. To explore this concept, a system simulator in MATLAB® is developed to model the distributed loads, agent routing, and recharging systems.

As discussed in Chapter 3, autonomous mobile energy assets are used to form an adaptive and temporary microgrid system to power critical loads for power restoration.

A genetic algorithm (GA) optimization method is used in several cases to determine optimal positioning of mobile sources to minimize objective functions, which enhance the utility of the resources. In [54], UAVs are utilized to recharge WSN, with a focus on the optimization of dedicated sinks in the network to maximize network life. In [55], the concept of utilizing many rechargeable UAVs as nodes in a WSN is explored, along with a proposed wireless recharging technology architecture. In [56], distributed and adaptive methods to traverse and wirelessly replenish the energy of WSNs are discussed.

Contrary to the referenced previous related work, this chapter focuses on the utilization of specific hardware used in practice for modeling, simulating, and testing the algorithms used for recharging distributed loads. The contribution to this chapter is a method to model and optimize scheduling algorithms for recharging distributed battery systems using MATLAB/Simulink. Real data from testing a Clearpath Husky UGV in the same mission scenario as presented is used for Monte Carlo (MC) simulation [57] to statistically analyze risk involved in the proposed algorithms.

The remainder of this chapter is organized as follows. The system modeling is explained in Section 4.2. In Section 4.3, several scheduling algorithms are proposed which are implemented in simulation case studies in Section 4.4. The hardware architecture that the case studies are based on is explained in Section 4.5. This hardware

is utilized for data collection in MC simulation analysis shown in Section 4.6. Section 4.7 compares the outcomes of the case studies and discusses variability in the MC simulations. Conclusions and future work are then presented in Section 4.8.

4.2 System Modeling

Consider N_s number of energy sources and N_l number of loads. The sources represent the mobile robots in Figures 1.1 and 1.2 from Chapter 1 and the loads represent distributed critical loads that need to be recharged periodically to maintain their utility. The loads could be systems such as wireless sensor networks, heating/cooling systems, or UAVs for example. The problem to be solved is to determine the sequence of individual wireless recharging of the loads that optimizes a desired objective function and satisfies constraints. This combinatorial optimization problem is generally computationally difficult. [56] formulates a similar charger optimization problem and notes that it is NP-complete [58], as is the traveling salesman problem. The objective function can include mobile robot traversal time, energy usage, down-time of loads, etc., and constraints applied can include maintaining a minimum state of charge in loads. Without the loss of generality, and to mimic the hardware utilized in practice, the recharging energy source is modeled as a constant DC voltage source, and the loads are modeled as generic rechargeable batteries. Each battery is connected to their own resistive load, modeled as resistors. The number of sources and loads are

assumed to be known and constant, with the location of the loads assumed to be known and fixed.

The system dynamics and operations in conducting this load recharging mission with a UGV is modeled in MATLAB/Simulink. This system includes the loads, which are modeled as independent rechargeable batteries, dispersed in an operating field. Time characteristics associated with the UGV traversing from load to load and docking to wirelessly recharge these individual loads are modeled with system time delays. Figure 4.1 shows an example of the problem with a single source and three loads. The example shows the trajectories of a UGV to the loads (L) to charge them in the sequence $\{L_2, L_3, L_1, L_3\}$,where the charge time (t) spent at each of these respective loads is $\{t, 2t, 3t, t\}$. In this example, obstacles are omitted to help simplify the problem. In a real world mission, this scenario could represent distributed sensors in data collection/monitoring systems as part of a lunar or planetary exploration mission. The loads represent the distributed sensors and the routes represent the paths traversed by the mobile ground rovers, equipped with photo-voltaic (PV) arrays and/or energy storage systems. The rover can move around the operational area to recharge the loads and since the loads are constantly using power to serve their purpose in continuously gathering and transmitting data/intelligence, they will eventually run out of energy. The rover needs to determine which loads to recharge, at what time, in what sequence, and for how long over some known time span, T. Proper mission operations are paramount to optimize the use of resources such as sensor battery

lifespan or rover traversal energy usage by investigating different mission outcomes in simulation. The approach to modeling the sub-systems involved in the scheduling algorithms are discussed next.

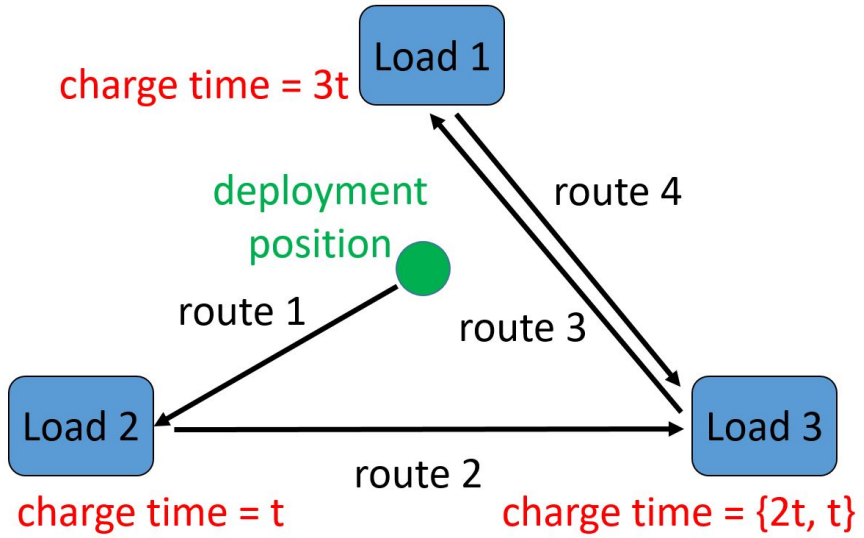

Figure 4.1: Example solution to distributed optimal load recharging schedule.

4.2.1 Rechargeable Battery Systems

The dispersed rechargeable battery systems are modeled using the Simulink-Simscape^{TM} generic dynamic rechargeable battery model found in the Simulink library under Simscape / Electrical / Specialized Power Systems / Electric Drives / Extra Sources, which is derived in [59]. The parameters in the model reflecting the circuitry in the batteries are adjustable by the user to attain specific battery dynamics [60]. A desired battery discharge profile for this work, consisting of the exponential,

nominal, and maximum discharge zones, can be seen in Figure 4.2. This is done by inputting the battery parameters given from a manufacturer of a similar battery used in this work, and is described in Section 4.5.1, into the battery model.

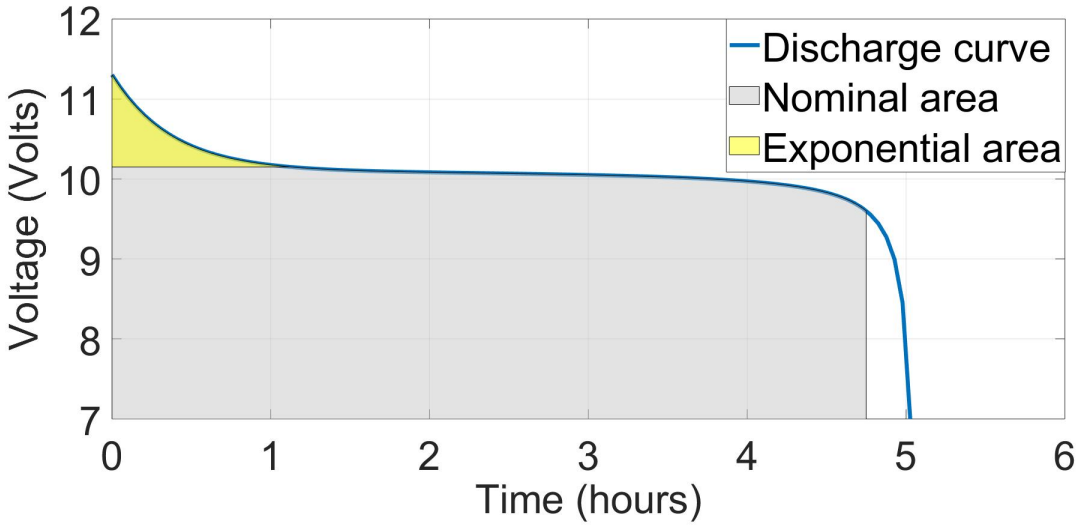

Figure 4.2: Discharge profile for simulated 9.6 V, 2000 mAh rechargeable NiMH batteries at 0.2 C.

4.2.2 Time Dependencies

Through the use of Stateflow®, a control logic tool available in MATLAB/Simulink, the time dependencies occurring in the battery dynamics from discharging and charging, and time attributed to the robot traversal and docking can be accounted for and considered when determining which load to recharge at what time. This is done with time delays in the simulation to represent an average time to fulfill these processes. The battery systems respond to these time instances as they would in a real system,

with self discharge, voltage dynamics, and updated state of charge (SOC) values due to this discharging and recharging.

The general procedure for the model progression is as follows. When the simulation is initiated, the battery load systems are switched on and feedback of the SOC from all batteries are input to the Stateflow logic controller. This SOC feedback along with the current location of the mobile energy source are used in the controller for determining which load to recharge at the current time and for how long. Once a decision is made, the UGV is committed to that operation for some time period, depending on the algorithm used and the states of the system. Once some criteria such as battery SOC or charge time is met, the charging decision process repeats, resulting in the next decision of which load to recharge in the sequence. To prevent battery fatigue in the loads, maintain battery efficiency overtime, and ensure some factor of safety if the UGV fails, if a battery SOC falls below some threshold defined by the user, it is disconnected from its respective load and enters load down-time. The general procedure for the model progression is shown in Figure 4.3.

4.3 Scheduling Algorithms

The problem to be solved, as stated in Section 4.2, can be viewed as a variation of the Job-Shop Scheduling Problem (JSP) [21] where the jobs are the entire process

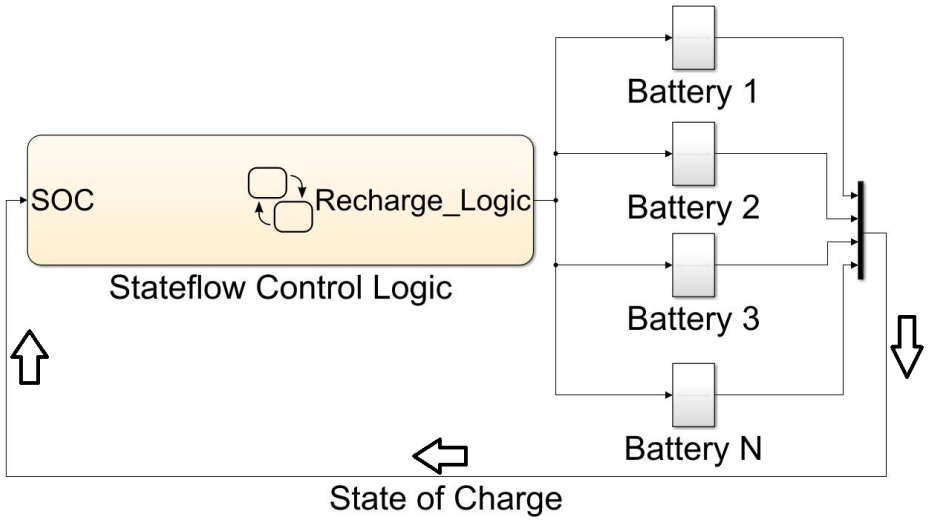

Figure 4.3: Overview of Simulink/Stateflow control model flow diagram for algorithm development and testing of a-priori mission plans.

of traversing to, docking, and recharging individual loads and the machines are the UGVs. This is a variation of the JSP due to a loose classification of when a job should begin, how long it takes to complete, and the cyclic nature of the recharging jobs themselves. Depending on the algorithm used, the load may be allowed to be partially recharged to give the UGVs more flexibility in which load to recharge and for how long. This gives the UGVs the opportunity to leave the load it is recharging to make it to another load before that load runs out of energy or has to shut down to preserve power. This gives rise to the criteria used in the proposed scheduling algorithms.

In the case studies and hardware architecture discussed in Sections 4.4 and 4.5, respectively, a single UGV resource is utilized, therefore, the problem can be reduced to a Single-Machine Scheduling (SMS) [24] problem. This SMS problem is prominent

in CPU scheduling in computer operating systems. First Come, First Serve (FCFS) and Round-Robin (RR) [20] are two popular algorithms used and discussed in CPU scheduling which our case studies are based on. These algorithms are chosen because of the reactive nature of them and the low computational complexity in applying them to real hardware. These algorithms are described next, along with a flexible genetic algorithm (GA) approach, capable of optimizing an objective function.

Scheduling algorithms can be described as either preemptive or non-preemptive. Preemptive schedules allow for interrupting jobs to be continued or finished at a later time. Non-preemptive schedules require processing of the job until it is complete. Within the context of the mission described in this work, completing a job is analogous to traversing to the load and completely recharging it to 100%. However, as mentioned in Section 4.2, if a load reaches some minimal SOC defined by the operator, the battery enters down-time. Load down-time is when a load SOC has reached some minimum value, defined by the operator, and that load should shut down to prevent extensive discharge to preserve the life cycle of the battery. The down-time of the ith load is calculated by

$$t_{down_i} = \sum_{j=1}^{N_d} \int_{a_j}^{b_j} dt \tag{4.1}$$

where

$$a_j = SOC_i < SOC_{min} \tag{4.2}$$

$$b_j = SOC_i > SOC_{min},\ a_j\ \epsilon. \tag{4.3}$$

N_d is the number of down-time occurrences, a_j and b_j are the jth down-time occurrence start and end times, respectively. SOC_i is the SOC of the ith load and SOC_{min} is a minimum SOC desired to be maintained in a mission.

Depending on the method used, a job could be queued when it reaches some minimal SOC defined by the operator or the jobs could always be queued but prioritized to prevent down-time of specific loads. As applied to our jobs, preemption refers to a UGV leaving a load before it is fully charged. Taking all the above into consideration, the objective function applied in all case studies to evaluate the algorithms is defined as

$$J = \sum_{i=1}^{N_l} t_{down_i} + t_{travel} \tag{4.4}$$

where t_{travel} is the total time spend traveling between loads and is calculated as the simulation time window, T, minus the total time spent charging all loads. The scheduling algorithms investigated in this chapter are discussed next.

4.3.1 First Come, First Serve

In the FCFS algorithm, jobs are sorted by the order in which they are queued. In this scenario, this is considered analogous to sorting by load SOC. When determining

which load to go charge, all load SOC are compared and the load with the lowest SOC is chosen. This process is repeated after the previously selected load has been fully recharged. The logic used in this algorithm is

$$L_{i_{charge}} = min(SOC_1, SOC_2, SOC_3, ...SOC_{N_l}) \tag{4.5}$$

for

$$SOC_{i_{charge}} < 100\% \tag{4.6}$$

where $L_{i_{charge}}$ and $SOC_{i_{charge}}$ are the ith load to go recharge and that ith load's SOC, respectively. The function in equation 4.5 identifies the load with the lowest SOC and equation 4.6 is the condition to maintain that load as the load to charge. Within the context of this work, this method is referred to as the FCFS-1 method. This method is considered non-preemptive because the mobile source is not allowed to leave a load until it has reached full charge or an appropriate SOC desired and defined by $SOC_{i_{charge}}$. A benefit of this FCFS inspired method is that it inherently prioritizes the load with the lowest SOC and acts to prevent load down-time. Another benefit is that it does not need an a-priori mission plan to be executed in a real-time system. It is a reactive algorithm which makes decisions in real-time, responding to unforeseen mission dynamics likely to occur in a real system. A disadvantage is that the mobile source cannot leave the current load it is charging before it reaches full charge, which could cause another load to reach down-time. A potential solution to this problem

is to prioritize a specific load and allow the mobile source to leave a load before it is fully charged to prevent the prioritized load from incurring down-time. The algorithm logic utilized with this alternative condition is the same as in equation 4.5 but has a different condition than equation 4.6, which is

$$SOC_{i_{charge}} < 100\% \;||\; t_{priority} <= t_{traversal} \tag{4.7}$$

where $t_{priority}$ is the time before the prioritized load will reach down-time and $t_{traversal}$ is the time it takes for the load to reach the prioritized load from its current location. This $t_{priority}$ value can be calculated by assuming the loads are discharged at a constant rate. Within the context of this chapter, this modified FCFS based method is referred to as the FCFS-2 method.

4.3.2 Genetic Algorithm Optimized Round-Robin

In the Round Robin (RR) algorithm, each job is given the same amount of time to be processed. After the time dedicated to that specific job has passed, the next job in the queue is processed, and the previous job moves to the tail end of the queue and the algorithm continues in this cycle. For our mission scenario, this is considered similar to dedicating the same amount of recharge time to each load and repeating this charging on the loads in a cycle. A benefit of this RR based method is that the

loads are treated equally and preemption is possible if the assigned charge time is short enough. A disadvantage is that if the assigned charge time is too short, the mobile source may spend too much time traversing from load to load throughout the mission, which may be inefficient. The logic used in determining the ith load to charge in this algorithm is

$$L_{i_{charge}} = \{L_1, L_2, L_3, ...L_{N_l}, L_1, L_2, L_3, ...L_{N_l}, ...\} \tag{4.8}$$

which is the sequence order of loads to go charge. Switching between loads occurs when the current load in the sequence receives

$$t_{i_{charge}} = t_{RR} \tag{4.9}$$

where $t_{i_{charge}}$ is the time spend charging each load and t_{RR} is the RR charge time. A genetic algorithm is used to determine this single optimal RR charge time. Equation 4.8 shows a sequence of increasing load identification number when the sequence could be in any order as long as each load is charged once and exactly once per cycle and this cycle is repeated.

4.3.3 Genetic Algorithm Optimized Flexible Round-Robin

As described in Chapter 2, GAs utilize the concept of genes in a chromosome which evolve progressively. As a global population based optimization algorithm, it is made up of a population of chromosomes, which represent different solutions in the solution space. Each chromosome is made up of genes which represent the design variables. The chromosomes go through the processes of selection, crossover, and mutation to explore the solution space and find the optimal solution. Every time the population evolves through these processes a new population is generated and the next generation is created, which is the start to the next iteration in the optimization computation. This iterative process continues until some stopping criteria is met. Generally, an initial random population is used to begin the optimization, but there are options to supplement in solutions which may be known to the user to be good. More details on GAs can be found in Section 2.4.

In applying the GA approach, the design variables are a sequence of integers representative of the time spent charging each load. Similar to the RR method, a cyclic load charging scheme is applied but flexibility is given in the charge time for each load charging occurrence. As an example, for Figure 4.1, if the cycle sequence is $\{L_2, L_3, L_1, ...\}$, then the GA solution for the first two cycles is represented as $\{t, 2t, 3t, 0t, t, 0t\}$. This algorithm has the same sequence of loads to go charge and is

represented in Equation 4.8 but the charging condition is represented as

$$t_{i_{charge}} = t_{RR_i} \tag{4.10}$$

where $t_{i_{charge}}$ is the time spend charging each load and t_{RR_i} is a set of N RR charge times. The GA is used to determine this optimal set of N RR charge times which is dependent on the number of load-to-load traversals in the 24 hour time window, which ultimately depends on the charging time lengths in N. This causes the problem have a variable-sized design space.

4.4 Simulation Case Studies

There are several simulation case studies presented in this section, reflecting the methods described in Section 4.3. All cases employ a single mobile energy asset and three distributed loads. The simulation time window, T, is over a twenty-four-hour period and infinite energy capacity is considered for the mobile source, considering that multiple UGVs in theory could work one at a time, in shifts to accomplish this mission. The distributed loads are assumed to be equidistant from each other with initial deployment of the mobile source in the center of the loads, as shown in Figure 4.1. The time considered for the mobile source to travel from the initial deployment location to any of the loads, and from any load to a different load is 35 and

60 seconds, respectively. All loads begin the mission relatively close to 100% SOC. Slight deviations between the initial load SOCs are implemented to reflect real world hardware. Slightly different resistive loads are applied to each load and a constant 12 V source is used in charging the loads. These load parameters can be see in Table 4.1. For all case studies, the minimum SOC to be maintained to prevent down-time is 50%. The loads and charging system modeled in these case studies are based on the hardware described next in Section 4.5. The rechargeable batteries are 9.6 V, 2000 mAh rechargeable batteries, and the wireless power transfer (WPT) modules produce 12 V. The parameters chosen in the battery models are selected to replicate the discharge profile given in the specification data sheet of a similar battery type battery. The Simulink and actual battery discharge profiles are shown in Figures 4.2 and 4.4.

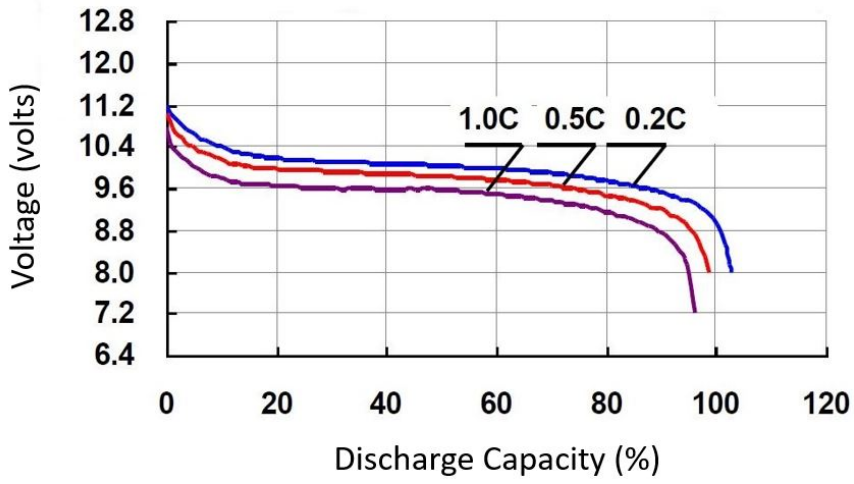

Figure 4.4: Discharge profile for 9.6 V, 2000 mAh rechargeable NiMH batteries from Tenergy manufacturer [1].

Table 4.1
Load parameters for all simulation case studies.

Load	Initial SOC (%)	Load Resistance ($Ohms$)	Charge Voltage (V_{dc})
1	99.7	28	12
2	99.5	28.5	12
3	99.3	29	12

4.4.1 Case A: First Come, First Serve Based Methods

In this first simulation the FCFS-1 based method is applied and the SOC of the batteries over time in the 24 hour mission time can be seen in Figure 4.5. The down-time for each load and the total traversal time is given in Table 4.2.

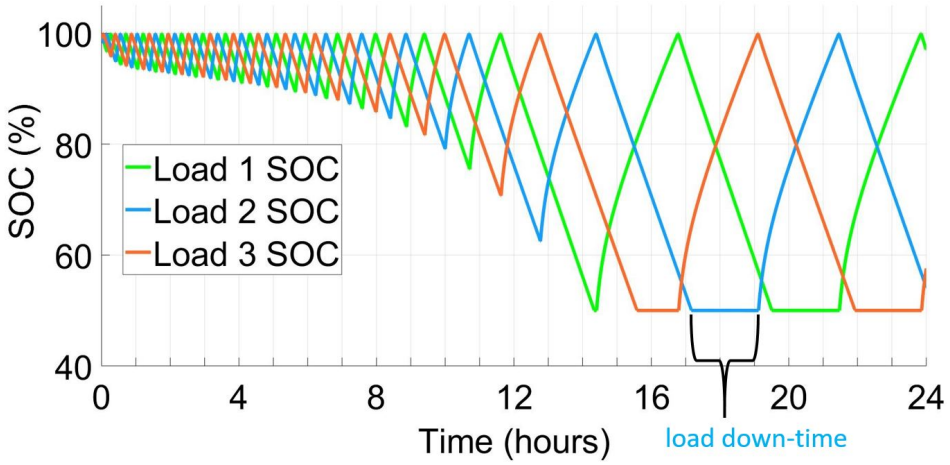

Figure 4.5: SOC for all loads in Case A-1 with the load down-time shown and annotated for Load 2.

From this simulation result, it is noted that this simple method cannot provide enough power over the 24 hour mission time to prevent down-time of any load. This is not a

Table 4.2
Individual load down-time and UGV travel time results for Case A-1.

L_1 Down-time (min)	L_2 Down-time (min)	L_3 Down-time (min)	Travel Time (min)
122.2	117.2	189.1	49.4

desired mission result so the previous FCFS-1 method can be modified to prioritize a specific load to prevent this down-time, as described in equations 4.5 and 4.7 as the FCFS-2 method. In Case A-2, $Load_1$ is prioritized and the SOC of the batteries throughout the mission time can be seen in Figure 4.6. The down-time for each load as well as the total traversal time is given in Table 4.3.

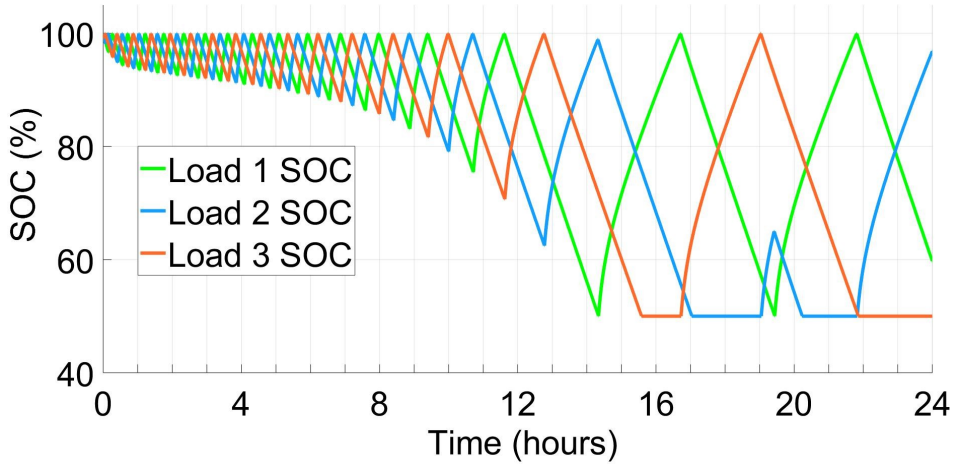

Figure 4.6: SOC for all loads in Case A-2.

Table 4.3
Individual load down-time and UGV travel time results for Case A-2.

L_1 Down-time (min)	L_2 Down-time (min)	L_3 Down-time (min)	Travel Time (min)
0	216.2	197.1	49.4

4.4.2 Case B: Round Robin Based Method with GA Optimized Charging Time

The GA approach, as explained in Section 4.3, can be applied in this RR inspired algorithm for determining an optimal charge time, applied to each load in a cyclic operation. An optimal cyclical charge time is found to minimize the objective function defined in equation 4.4. This objective function attempts to minimize down-time of all the loads and the time spent traveling between loads over the course of the 24 hour simulation time window. The cyclic charging procedure follows in the order of $\{L_1, L_2, L_3, ...\}$. This cycle is repeated for the entire mission. The upper and lower bounds for the t_{RR} design variable is 2 hours and 20 minutes, respectively. The GA optimizer is able to find an optimal t_{RR} of 1760 seconds or 29.3 minutes, resulting in an objective function cost of 48.4 minutes. The SOC of the batteries over time in applying this mission plan are shown in Figure 4.7. The down-time for each load and the total traversal time is given in Table 4.4.

Table 4.4
Individual load down-time and UGV travel time results for Case B.

L_1 Down-time (min)	L_2 Down-time (min)	L_3 Down-time (min)	Travel Time (min)
0	0	0	48.4

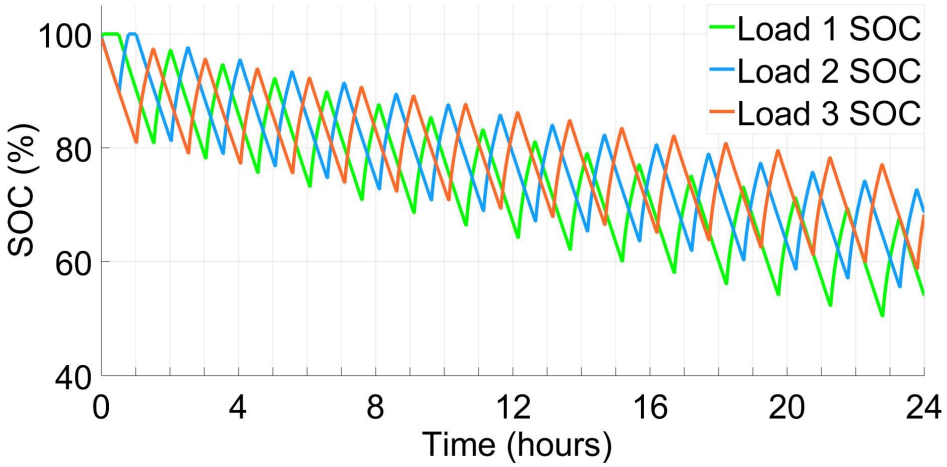

Figure 4.7: SOC for all loads in Case B.

4.4.3 Case C: Flexible Round Robin Based Method with GA Optimized Charging Times

This flexible RR based scheduling algorithm is similar to the RR based algorithm in that there is a designated cycle in load charging, however the time in charging each load can vary throughout the entire mission. When a load is charged, the gene charge time, t, as explained in Section 4.2, is set to a possible lower bound of 10 minutes, and upper bound of 2 hours. With a mission time window of 24 hours and neglecting travel time, this equates to a maximum of 144 design variables that may need to be optimized. The number of design variables utilized in the simulation is dependent on the number of times the mobile source transitions loads to charge. As more transitions occur, more time is dedicated to traversing

and not charging, so not all design variables may be utilized. Any variables not utilized within the simulation time window are neglected. The optimal solution from Case B is seeded into the GA initial population in this case. The same objective function as shown in equation 4.4 is applied to this optimization method. The GA optimizer is able find an optimal set of 45 recharge times, resulting in an objective function cost of 45.35 minutes. The distributed loads' battery SOCs over time can be seen in Figure 4.8. The results of the cost components of this objective function are shown in Table 4.5. The respective 45 optimized charge times in minutes are $\{30.1, 28.52, 34.25, 29.57, 28.8, 29.45, 30.7, 44.15, 29.17, 30.08, 31.57, 24.03, 36.9, 28.68, 28.02, 30.83, 30.27, 33.57, 29.58, 27.33, 31.18, 28.75, 40.78, 29.65, 40.73, 30.57, 29.27, 31.93, 31.05, 28.97, 28.6, 31.1, 28.32, 28.85, 27.67, 43.08, 35.57, 29.13, 29.43, 30.2, 29.82, 30.38, 31.08, 29.25, 27.53\}$.

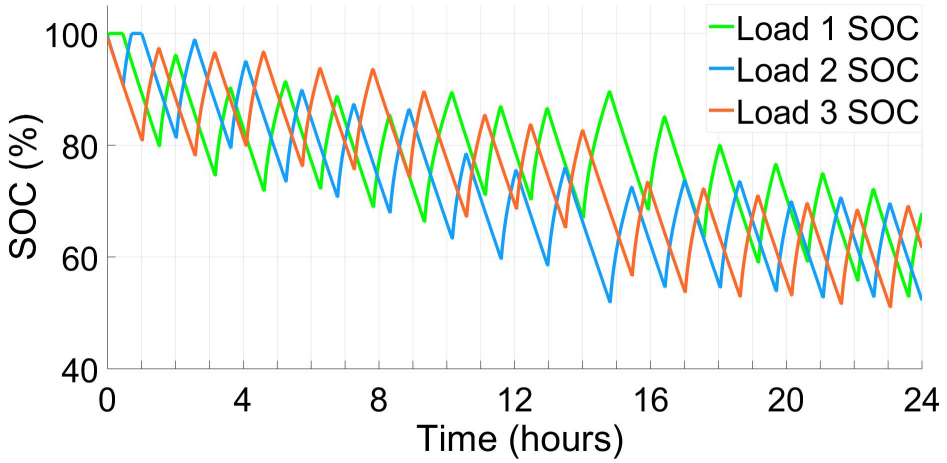

Figure 4.8: SOC for all loads in Case C.

Table 4.5
Individual load down-time and UGV travel time results for Case C.

L_1 Down-time (min)	L_2 Down-time (min)	L_3 Down-time (min)	Travel Time (min)
0	0	0	45.35

4.5 Hardware Architecture

The simulation studies presented in Section 4.4 are based on hardware being developed to implement this distributed load recharge mission. This hardware consists of a Clearpath Husky A200 UGV, 9.6 V, 2000 mAh rechargeable batteries, and a wireless inductive charging system capable of recharging these batteries. These three central systems are briefly discussed next.

4.5.1 Rechargeable Battery Loads

The rechargeable batteries used are 9.6 V, 2000 mAh, nickle metal hydride (NiMH). These battery packs contain 8 AA type 1.2 V cells. Discharge specifications for these batteries used could not be obtained but specifications for a similar Tenergy brand pack with the same nominal voltage and energy capacity were found and is shown in Figure 4.4. This battery was chosen because of its simplicity in that there is no

charging control in the battery system and it is rated for the max current and voltage input that the wireless inductive charging system produces.

4.5.2 Wireless Power Transfer

The near-field wireless power transmission works on the principle of inductive coupling. In the charge coupling setup, there are two separate modules, one transmitting module and one receiving module. The transmitting module is located on the front of the Husky UGV and the receiving module is located on the stationary load as shown in Figure 4.9. These modules have no physical connection between them. Power is transferred due to the magnetic flux from the transmitting module inducing an alternating current on the receiving module coil. The transmitting modules rated input capacity is 24 V and 1 A DC. The receiving modules rated output capacity is 12 V, 2 A DC and is connected to the 9.6 V battery for charging. Data was collected in recharging this battery with the WPT system and the current supplied to it can be seen in Figure 4.10. From this data, the net current from the battery can be integrated over time to estimate the battery SOC, known as coulomb counting. This is shown in Figure 4.11.

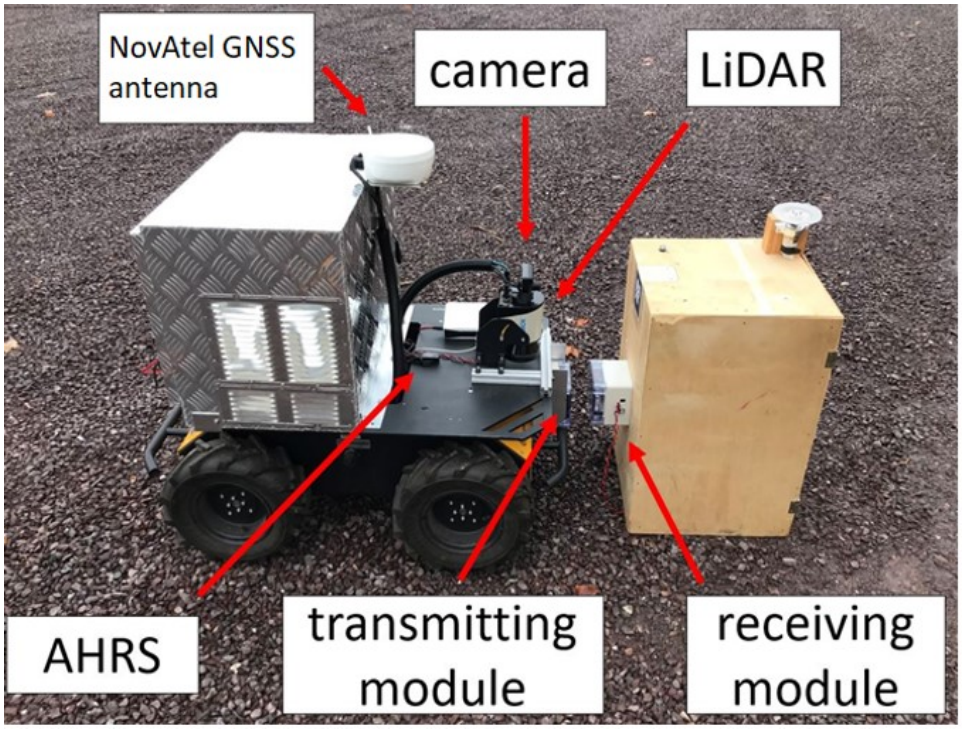

Figure 4.9: Clearpath Husky UGV with transmitting module in close proximity to load receiving module for wireless docking to recharge a distributed load.

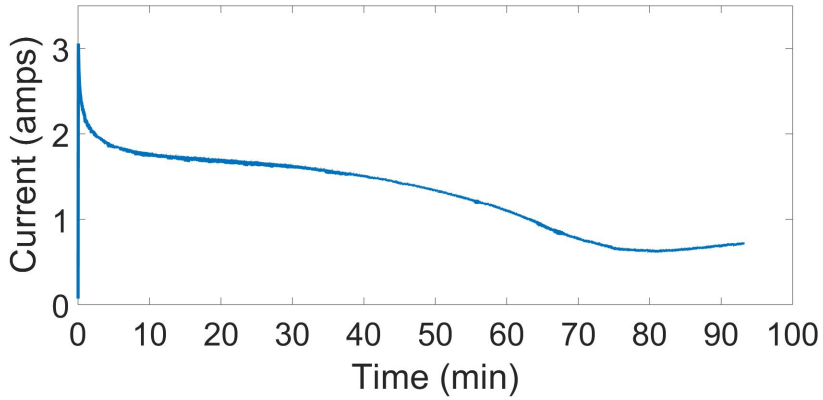

Figure 4.10: Current supplied to 9.6 V, 2000 mAh, NiMH rechargeable battery during wireless charging testing.

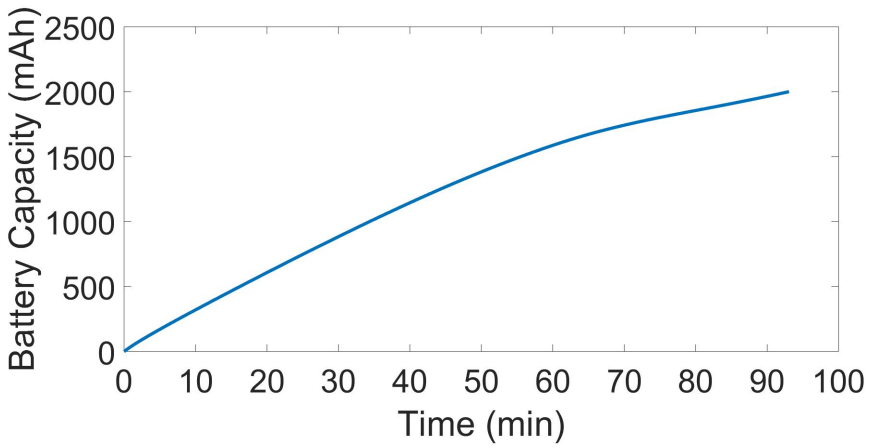

Figure 4.11: Energy capacity of 9.6 V, 2000 mAh, NiMH rechargeable battery during wireless charging testing, using coulomb counting.

4.5.3 Husky UGV

The mobile UGV incorporated in this mission application is a Clearpath Husky A200. For localization and navigational purposes, the UGV is equipped with a SICK two-dimensional LiDAR, a NovAtel GNSS Antenna, a LORD Micro Strain Attitude Heading Reference System (AHRS), and a Sony PlayStation Eye camera. The transmitting module in the WPT system is mounted at the front of the UGV. Figure 4.9 shows the UGV docked at one of the distributed loads for wireless recharging. This UGV in particular is fitted with a Honda EU1000i gas generator and a switching power supply to offer on-board battery recharging to extend its mission time and for power transfer to the distributed loads.

4.6 Monte Carlo Simulation Analysis

The Round-Robin algorithms presented in Section 4.4 provide *a-priori* schedules to fulfill the recharge mission. In simulation, all methods assumed exact parameters such as mobile source travel time from deployment and between loads. These *a-priori* schedules applied on hardware would more accurately follow the outputs of the simulations if the model exactly represented the hardware and physical system parameters. However, because of variability apparent in hardware and not yet modeled in simulation, errors in time delays may accumulate, resulting in the hardware and the schedules becoming inaccurate. To better analyze and understand these probable outcomes in applying these *a-priori* and reactive scheduling algorithms on hardware, MC simulations can be carried out with the use of statistical data measured in the real system to better understand the potential outcomes.

Using the hardware as described in Section 4.5, the mission scenario simulated in the case studies and shown in Figure 4.1 is replicated for hardware testing and can be seen in Figure 4.13. It is assumed that the loads are equidistant based on this hardware setup and traversal between any two loads is relatively equivalent. The testing procedure and process is carried out as follows. Starting from the location and pose as shown in Figure 4.13, the UGV randomly selects between the three loads to traverse to, and dock for wireless recharging. Once a load is chosen, the UGV achieves

a load corresponding way-point and angular pose. The UGV then pans by rotating in place to identify the load chosen via augmented reality tag (ART) identification. Once acquired, the UGV traverses to and docks on the load for wireless charging. A sample of 10 measurements of the Husky UGV traversing to and wirelessly docking on loads are logged to understand and model the time variability associated with the different processes. The individual processes involved to move from one load to another are categorized as T_{AR} and T_{mobile}. T_{AR} is the time attributed to acquiring the next AR tag and T_{mobile} is the sum of time attributed to traversing from a load to the next way-point and from that way-point to docking on the load.

The T_{AR} process begins when the UGV finishes traversing to its corresponding load way-point and achieves the commanded pose with respect to the load. From here, the UGV then pans to acquire the ART. It always initially pans counter-clockwise to approximately 90 degrees then back to its initial pose then clockwise approximately 90 degrees before returning back to its original pose again. This process is repeated until the ART is acquired. For modeling purposes, the T_{AR} process is considered a function of two random variables, the initial pose achieved by the UGV with respect to the load it is acquiring, and the amount of times the UGV misses acquiring the load ART. The initial pose is modeled as a Gaussian distribution with a mean of 0 radians and a standard deviation of $\pi/12$ radians or about 15 degrees. A miss is defined as the UGV's extended vertical reference plane sweeping past the load ART without acquiring it, which represents the UGV's monocular camera vision. This

number of misses is modeled as a Poisson random variable with a mean of 0.55, per testing observations. A load is considered acquired upon the UGVs extended vertical reference plane coming in contacting with the load ART after the last randomly prescribed number of ART misses. With a measured approximate angular turn rate of 0.2 radians per second and the inputs of the randomly selected number of misses and initial UGV pose, the T_{AR} time is calculated with a mathematical function. The UGV in this T_{AR} state can be seen in Figure 4.12. This is a simplification and estimate of the actual ART acquisition process since in reality the monocular camera has an angular field of view.

The T_{mobile} process time was measured from experimentation, resulting in a mean of 43.1 seconds. For modeling purposes, this data is used to create a noncentral t-distribution, which is randomly drawn from to produce T_{mobile} for each load-to-load traversal. From this time measurement and T_{AR},

$$T_{total} = T_{mobile} + T_{AR} \tag{4.11}$$

where T_{total} is the total time to complete the process of load-to-load docking. The same process is done to calculate a randomly drawn initial deployment-to-load docking time but data measured from the initial way-point to dock is used. This is expressed as DL_{total} where

$$DL_{total} = T_{way-pt} + T_{AR} \tag{4.12}$$

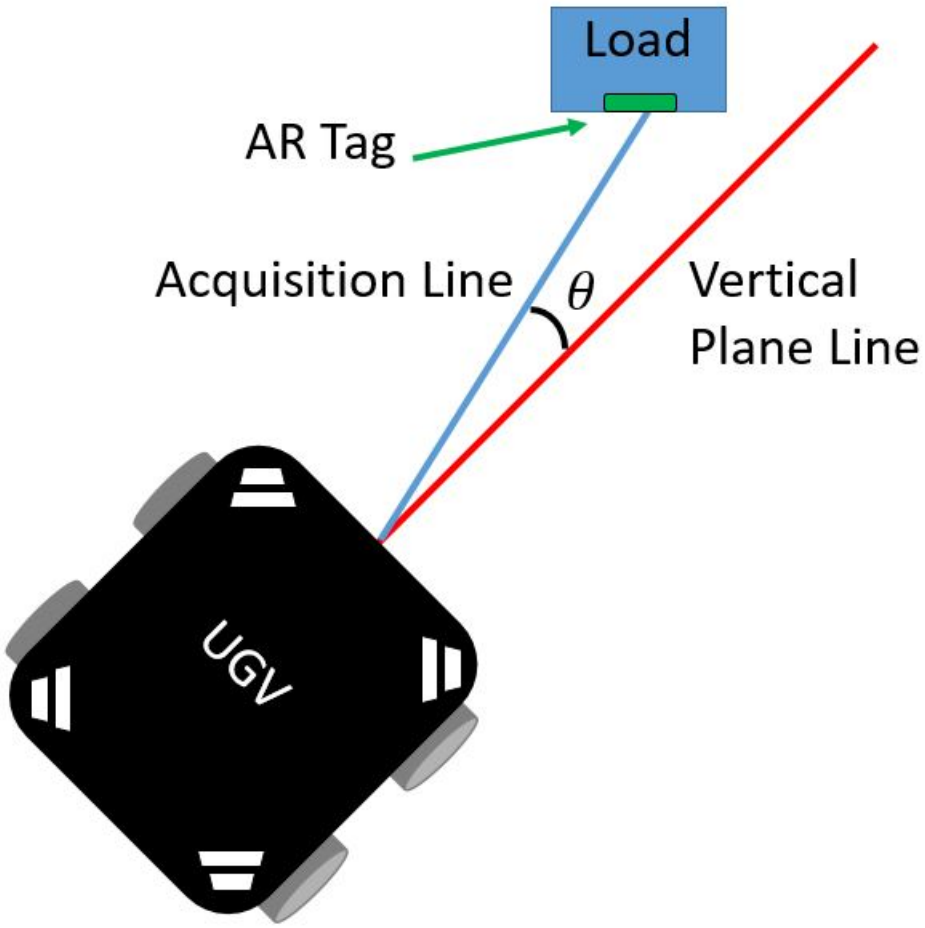

Figure 4.12: Bird's eye view of UGV in the ART acquisition state with θ offset angle from the load.

and T_{way-pt} is randomly chosen from a t-distribution from data gathered from the deployment-to-load times and T_{AR} is the same value as noted for Equation 4.11.

A set consisting of 100 randomly drawn T_{total} times and one DL_{total} is used for each simulation run. A thousand random sets are generated and instantiated into a mission simulator in Simulink for the MC analysis. In each simulation run, any of the 100 load-to-load times not utilized are neglected. Each algorithm is simulated a thousand

times with this same load-to-load and deployment-to-load time variability to better analyze how the physical system will respond to the algorithm and mission plans to be implemented. The data associated with this time variability data is shown in Table 4.6. The deployment-to-load time variability data can be seen in Table 4.7

Table 4.6
Monte Carlo load-to-load time variability data.

T_{total} avg (sec)	T_{total} max (sec)	T_{total} min (sec)	T_{total} std (sec)
62.5	225	21	17

Table 4.7
Monte Carlo deployment-to-load time variability data.

DL_{total} avg (sec)	DL_{total} max (sec)	DL_{total} min (sec)	DL_{total} std (sec)
47.8	100.9	18.8	13.9

For the 1000 Monte Carlo simulation runs for each algorithm, the objective function average, maximum, minimum, and standard deviation are shown in Table 4.8.

Table 4.8
Monte Carlo comparison of the case study objective function results.

Case Study	J_{avg} (min)	J_{max} (min)	J_{min} (min)	J_{std} (min)
Case A-1	516.9	622.7	392.9	35.6
Case A-2	477.4	594.8	353.8	34.5
Case B	52	62.5	45.3	3
Case C	49.6	63.6	42.6	3.4

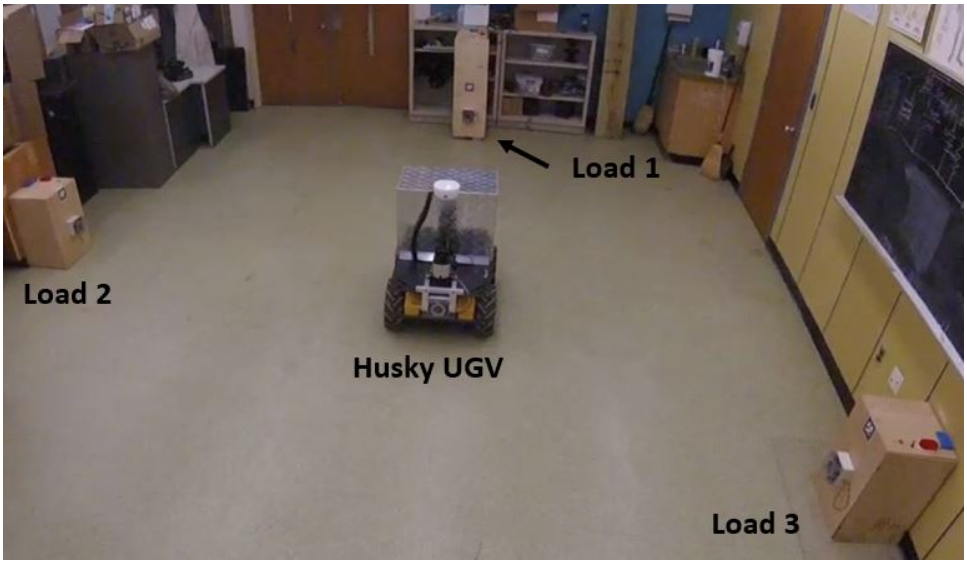

Figure 4.13: Hardware replication of the simulated distributed three load scenario for hardware testing.

4.7 Discussion

The simulation case studies are theoretical results representative of potential outcomes in applying these algorithms on the Husky UGV and distributed load systems described. A constant traversal and docking time of 60 seconds is assumed for the initial experiments. The MC analysis gives insight to the potential best and worst statistical case outcomes in applying these mission plans to hardware platforms. Data from testing the scenario with a Husky UGV is documented and utilized to add variability for MC analysis. These initial simulation results and the MC simulation analysis results will be discussed next.

4.7.1 Simulation Case Studies

The FCFS based methods are appealing because they are reactive to the model states and can easily be applied in real-time if load SOC feedback can be estimated and utilized. These methods would not require an *a-priori* mission plan and would require less computational complexity in using simple logic to determine which load to go charge at the current time step. Latching to a decision of fully recharging a load in these algorithms may be more appropriate in minimizing traveling between loads. However, this depends on the constraints that need to be satisfied by those commanding the mission. In evaluating the FCFS-1 method presented in Case A-1, it resulted in the worst performance with an objective function evaluation of 477.9 minutes, with 49.4 minutes attributed to travel time. The FCFS-2 priority based method presented in Case A-2 was successful in prioritizing Load 1 to keep it running for the entire mission without load down-time. It resulted in a second worst objective function evaluation of 462.7 minutes, with 49.4 minutes attributed to travel time, the same amount as in Case A-1.

The RR inspired method is a good approach in it's ability to provide each load equal charging. It can apply an objective function to minimize in the algorithm, giving options for constraints and for the type of goal desired to be achieved. It avoids down-time of all three loads and achieves the second best objective function value of

48.4 minutes, all attributed to travel time.

The flexible RR GA based method adds more complexity to a potential solution, allowing a large variety of charging times to apply to different loads. From the results of Case C, the algorithm avoids down-time of all three loads and it is successful in determining a solution with the best objective function of 45.35 minutes, all attributed to travel time. With this applied objective function in evaluating all the algorithms, the flexible RR GA method in Case C is the best option to choose from and performed the best. The RR method with a single optimized recharge time in Case B is a good next option. For all case studies, their objective function cost results are broken down by their attributes for comparison and are shown in Table 4.9.

Table 4.9
Comparison of the case study objective function results.

Case Study	L_1 Down-time (min)	L_2 Down-time (min)	L_3 Down-time (min)	Total Down-time (min)	Travel Time (min)	J (min)
Case A-1	122.2	117.2	189.1	428.5	49.4	477.9
Case A-2	0	216.2	197.1	413.3	49.4	462.7
Case B	0	0	0	0	48.4	48.4
Case C	0	0	0	0	45.35	45.35

4.7.2 Monte Carlo Simulations

The MC simulations show the results of applying 1000 simulation runs for each algorithm scheduling strategy, each utilizing the same randomly drawn variable load-to-load traversal and docking times. The results from Table 4.8 show a large standard deviation in the objective cost in Cases A-1 and A-2. Cases B and C show relatively low objective costs where the larger standard deviation in Case C with a higher maximum and lower minimum. This leads to believe that Case C may have a higher risk but higher reward associated with implementing it. These results are further expanded in Table 4.10 which shows the one thousand simulation runs average cost, average load down-time, average UGV travel and charge time, and average ratio of travel to charge time for each algorithm. Note that all ratio values are all very close which shows that the effectiveness of each algorithm does not seem to reflect the overall UGV average travel and charge time. This shows that the effectiveness of the algorithm is not solely determined by how much total time the UGV spends traveling between and charging loads. This points to the importance of when the UGV load charging and traversing events occur in time, and for which loads these actions are directed towards overtime. Also note that the average load down-time for cases B and C is 1 and 1.8 minutes, respectively, which is not zero as one may expect. This is due to implementing algorithms which previously assumed a 60 second traversal time prior to adding the traversal variability times. Also, it is not noted in the tables

but the FCFS-2 algorithm has an average load down-time of 14 seconds, which was previously 0 without traversal variability. Again, this is due to assuming a 60 second traversal and docking time. This could be improved by assuming a more conservative traversal and docking time but there is a trade off in performance in doing this, which is expected to decrease if implemented. These In determining to implement these algorithms, one should consider these statistical output results of the average, best, worst, and standard deviation as shown in Tables 4.8 and 4.10.

Table 4.10
Monte Carlo comparison of the case study objective function results.

Case Study	J_{avg} (min)	Average load down-time (min)	Average travel time (min)	Average charge time (min)	Average travel/charge time ratio
Case A-1	516.9	469.3	47.6	1392.4	0.034
Case A-2	477.4	428.1	49.3	1390.9	0.035
Case B	52	1	51	1389	0.037
Case C	49.6	1.8	47.8	1392.2	0.034

4.8 Conclusion

In this chapter, a general framework and several solution methods are presented with a goal to optimally recharge distributed loads in an operating field, utilizing a mobile energy asset with wireless inductive charging. Both real-time reactive algorithms and algorithms resulting in *a-priori* mission plans are explored. Results of the scheduling algorithms, including a flexible genetic algorithm approach, are compared

and discussed. A Monte Carlo analysis performed provides additional data about the variability in the algorithms' outputs when real-world based variability measurements are included in the simulations. The flexible round-robin genetic algorithm approach is found to perform the best. This method and modeling used can be extended to various types of distributed loads such as recharging WSNs or UAVs used for data collection, surveillance, or reconnaissance. It can also be extended to include results from applying these load scheduling algorithms on the Husky UGV hardware with one or multiple mobile UGVs. Simulation studies can be expanded to include multiple mobile energy sources and additional heterogeneous distributed loads in a multitude of different time frames.

In the next chapter, an extension of this optimization strategy is presented which optimizes the recharging of UAVs with a UGV to complete various reconnaissance mission objectives. Different combinations of resources are considered in achieving different mission goals. Results of various case studies show the robustness of the method presented.

Chapter 5

Optimal Mission Routing of UAVs and Collaborative Recharging UGVs for Intelligence, Surveillance, and Reconnaissance

5.1 Introduction

As modern warfare becomes more dangerous to military personnel operating and leading missions on foot in potentially hostile environments, the use of real-time data

for smart and autonomous systems will become more involved in military operations. To reduce the risk of harm to military soldiers and civilian bystanders, the utility of these autonomous systems needs to be optimized to ensure an efficient, effective, and fast operation. This chapter proposes algorithm optimization strategies of task allocation for unmanned aerial and ground vehicles (UAVs and UGVs) for military reconnaissance (recon). The objective is to create an optimal mission plan to perform reconnaissance in target areas, along with recharge the UAVs with UGVs wirelessly, to extend and optimize their operational life based on the mission at hand. This work investigates two different reconnaissance missions, one with the goal of minimizing the mission time and another to maximize the area reconned. The results show that the optimization approach and methods can be regarded as a reliable schedule optimization tool for this application of wireless recharging of mobile UAVs with mobile UGVs. The proposed approach can be extended to other operations such as mobile robotic explosive ordnance disposal (EOD) teams scanning for improvised explosive devices (IEDs).

Military reconnaissance is a concept that has been used for hundreds of years. The purpose is to gain as much intelligence as possible to minimize risk before another operation is carried out. Reconnaissance in itself is a dangerous and life-threatening mission, even when performed by the most trained team of military specialists. The reconnaissance team is exploring areas in which they have little existing knowledge of, such as the area landscape, terrain, and potential hostile enemies. Intelligence

gained from reconnaissance may be utilized before advancing or relocating a forward operating base (FOB) or used for other future mission plans. With the increasing use of intelligence gathering devices in military practices such as UGVs and UAVs, military personal are at a lower risk of harm by utilizing these agents in place of a human life to perform a mission. The use of these devices will become more frequent as variables such as battery life, computation expenses, and data communication are improved.

One of the major challenges in utilizing UAVs for reconnaissance missions is their limited battery capacity. Without recharging or exchanging batteries, a single UAV is extremely constrained in its ability to investigate large areas in need of reconnaissance. One step in improving this is optimizing the mission in which these autonomous agents are used. Operating these agents in coordination with each other, with the ability for them to work as a team, is crucial. Like most military operations, a plan needs to be established before the mission is conducted to understand how the mission will be coordinated and carried out. This plan needs to be as efficient and optimal as possible. There are essentially infinitely many ways to define or establish what is optimal, and this is often mission dependent. The objective functions involved may seek to minimize mission time, energy utilized, intelligence gathered, or a combination of these.

The question which this work begins to help answer is "Now that we have the autonomous systems to carry out this task, how to we make the best use their energy resources to complete the mission in an optimal manner?" This chapter proposes algorithm optimization strategies of task allocation for UAVs and UGVs for military reconnaissance. The objective is to create a mission plan for each resource utilized in the reconnaissance operation, with the UAVs conducting the reconnaissance and a UGV traversing to targets along the UAVs' paths as rendezvous positions for periodic recharging. The UAV portion of the mission plan defines the targets and target reconnaissance sequence for each UAV utilized. The UGV portion of the mission plan defines which UAV it will recharge, as well as when, where, and how long. The mission plan outcome is dependent on the optimization goals for the reconnaissance mission. A concept example of a UAV following a mapped mission plan trajectory and a UGV recharging the UAV at various rendezvous mission plan locations is shown in Figure 5.1. This work investigates two different types of recon missions, one with the goal of minimizing the mission time and another to maximize the area reconned. The results show that the optimization approach and methods can be regarded as a reliable schedule optimization tool for this application of wireless recharging of energy constrained mobile UAVs with energy constrained mobile UGVs. The proposed approach can be extended to other operations such as mobile robotic explosive ordnance disposal (EOD) teams scanning for improvised explosive devices (IEDs).

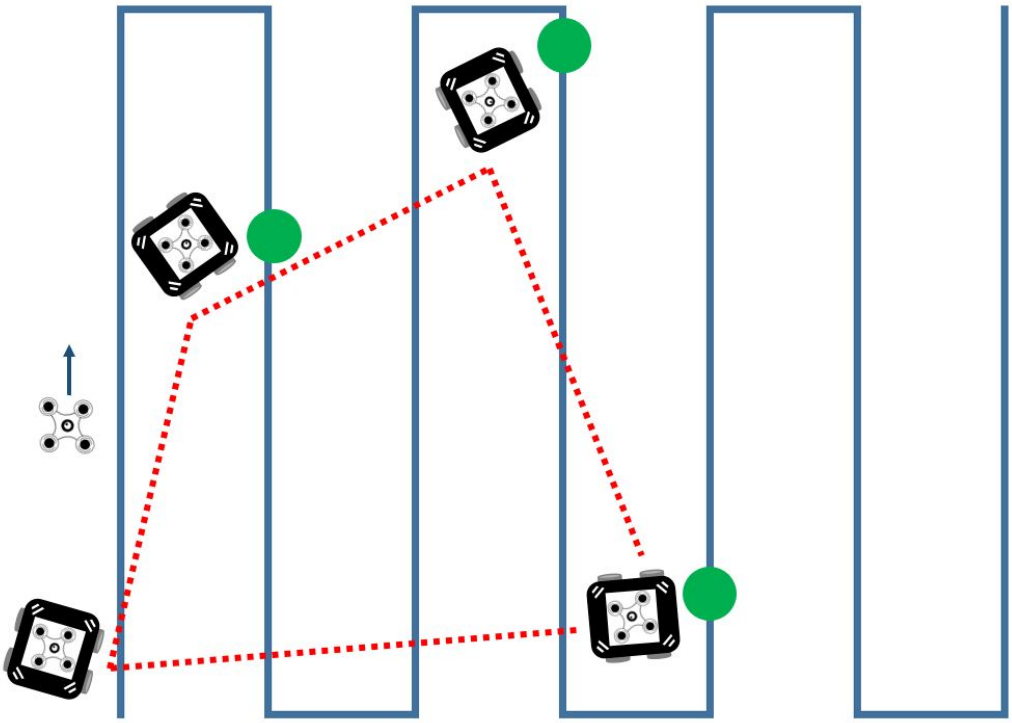

Figure 5.1: Bird's-eye view of collaborative recon mission concept with a UAV following the blue line trajectory and a UGV following red dotted-line trajectory with recharging rendezvous locations at the green dots.

5.2 Related Work

Many optimization problems involving limited resources with time or energy constraints can be constructed as task allocation [19] or scheduling problems, with [61] contributing one of the earliest works on production schedule optimization. Within this larger research area, as technology has developed, new facets around combinatorial optimization problems [62] have come into focus. These include variations of the Traveling Salesman Problem (TSP) [63], such as the Vehicle Routing Problem (VRP), and further variations of VRP [64]. Technology development surrounding autonomous

mobile robots and drones has inspired the use of these resources in problems like the VRP.

Research in determining optimal flight routes, requiring optimal recharging locations, is investigated in works such as [65] and [66]. The problem is formulated as a variation to the classic TSP. Their work only considers the use of fixed charging stations to recharge the UAV. Also, only a single UAV is utilized so there is no need to schedule the use of a given charging station.

Research in collaborative UAVs and UGVs has been investigated in several studies, each with different considerations, depending on the application. Because of this, many different problem formulations are presented. In [67], the problem is approached and formulated into a two-echelon ground vehicle and its mounted unmanned aerial vehicle cooperative routing problem (2E-GUCRP), for intelligence, surveillance, and reconnaissance (ISR) missions. The work presents a mixed integer programming (MIP) solution method along with two improved heuristic approaches. The studies do not include the use of multiple UAVs as this work does. Also, the work only considers a minimal mission time objective, leading to certain aspects of the supporting UGV to be neglected, such as its limited operational life.

Research specifically in maximal area coverage is investigated in [68] where a UAV is launched from an aircraft carrier which moves continuously in a straight line. A GA method is proposed which outperforms the nearest neighbor and hill-climbing

algorithm approach. Although this work has a similar objective in mind, the problem formulation and constraints are much different as compared to how it is presented in this work. Also, only a single UAV is deployed for the mission considered, where multiple UAVs are considered in this chapter.

[69] presents a similar problem related to the work in this chapter as a Fuel Constrained UAV Routing Problem using Mobile Refueling Stations (FCURP-MRS). The work incorporates a greedy strategy to solve the problem with a simulator in MATLAB. As with previous work mentioned, this work only incorporates a single UAV and does not consider the energy capacity of the mobile refueling station.

With the previous mentioned literature in mind, very little existing research in this area of collaborative UAV/UGV systems considers the battery capacity or operational life of the supporting UGV, which is critical in operations to maximize area coverage, and should be at least taken into consideration in minimal time objective missions. Furthermore, most of the existing research in this field assumes the use of a single UAV and UGV and does not analyze benefits of additional resources whether it be a UAV or UGV. There is also little analysis of the optimal mission outcomes such as the UAV/UGV SOC over the course of the mission, as well as standby, charging, and travel time of resources to support what really happens during the mission.

In this work, UAVs are utilized to perform reconnaissance and gain intelligence of hostile enemies or any unknown presence in the desired area. Due to the extent of

the mission at hand and utilizing a limited number of UAVs with limited battery life, a UGV is provided for mission assistance to charge the UAVs when needed via near field wireless power transfer. Equipped with much larger battery storage, the UGV is able to intercept the UAVs periodically at rendezvous locations to recharge them. It is assumed that obstacles, and terrain have been pre-mapped through the use of another UAV, satellite imagery data, or some other technology. The mission goal is to conduct reconnaissance through surveillance, sensors, radar, etc. in the mapped area. This reconnaissance is carried out by coordinating the UAVs to stop at target points where reconnaissance around that area is conducted. These targets areas make up the entirety of the area in need of reconnaissance and are assumed to meet along certain latitude/longitudes so as to not overlap.

Individual agent properties such as state of charge, as well as terrain data, rendezvous target points, and recon areas are available to the autonomous entities to determine optimal targets to visit, by which UAV, and in what sequence, to determine optimal path plans. A genetic algorithm to determine these optimal plans, along with optimal UAV-charging times, locations, and duration is the focus of this chapter.

The remainder of this chapter is organized as follows. The system modeling is explained in Section 5.3. In Section 5.4 the GA optimization method is discussed with several case studies presented in Section 5.5. Section 5.6 compares the outcomes of the case studies and discusses their relevant importance. Conclusions and future work

are then presented in Section 5.7.

5.3 Problem Formulation and System Modeling

Within an area of desired reconnaissance, consider that there are N_T distributed target areas that require reconnaissance with N_A UAVs and N_G UGVs utilized for this purpose. While airborne, the UAVs perform the reconnaissance and the UGVs support this mission with the ability to recharge the UAVs periodically at different target locations when necessary. Part of the problem to be solved in this mission is to determine which targets should be reconned by which UAV, and in what sequence. Another part of the problem to be solved is to determine the recharging sequence, locations, and duration of individual wireless recharging of the UAVs with the UGVs, that optimizes a desired objective function and satisfies constraints. These two parts are solved separately as what is referred to in this work as the Phase-One and Phase-Two mission optimizations, respectively as presented. These types of traveling combinatorial optimization problems are in general computationally difficult. [70] formulates a similar type of mobile charger optimization problem and proves that it is NP-hard[58].

The system modeling within the context of this chapter is similar to the concepts explained in [3] and Chapter 4. Stateflow, a control logic design tool available in MATLAB/Simulink are utilized to take the mission system dynamics and operations

into account, including time attributed to the traversing of UAVs and UGVs, reconnaissance by the UAVs, and battery SOC of the resources. Further details of the system/problem modeling are discussed next in separate subsections.

5.3.1 UAVs

The UAV and UGV batteries are modeled using the Simulink-Simscape^{TM} generic dynamic rechargeable battery model found in the Simulink Specialized Power Systems library. With the available user input parameters, the UAV rechargeable batteries are modeled as 9.6 V, 2000 mAh, NiMH, the same modeling as the loads presented in Chapter 4. The UAVs are assumed to always be operating in one of three different operating modes, which are reconnaissance, charging, and standby. Reconnaissance mode represents a UAV either performing recon or traveling between target areas. When in this mode, the UAVs are assumed to be traveling at a constant speed of 5 m/s and this is simulated by apply a 5 Ω resistive load to its battery.

Charging mode operates when the UAV has been intercepted by, and lands on top of a UGV for wireless power transfer. The receiving module in the wireless charging systems, as described in Chapter 4, is assumed to be incorporated in and on the bottom of the UAV. This receiving module operates at 12 V and 2 A, converted from the transmitting module's 24 V, 1 A. This 1 A draw from the UGV battery and

2 A input to the UAV load battery are utilized in the model to reflect the hardware architecture that has been developed in previous research. This mode is simulated by applying a constant 12 V DC voltage across its battery, as was done in Chapter 4 to simulate the charging of the distributed loads.

Standby mode is in operation when the UAV is at a charging rendezvous target location before a UGV has arrived there. This mode has no resistive loads associated with it and the SOC of the corresponding UAV remains constant during this mode. The UGV system modeling is discussed next.

5.3.2 UGVs

The UGV battery is modeled as a lead acid 24 V, 20 Ah, which is the same battery type and specifications used in the Clearpath Husky A200 UGVs discussed in Chapter 4. The UGV is assumed to always be operating in one of three different modes, which are travel, charging, and standby. Travel mode represents a UGV traveling between rendezvous recharging target areas. When in this mode, the UGV is assumed to be traveling at a constant speed of 1 m/s, which is consistent with the max rated speed of a Clearpath Husky. This is simulated by applying a 3.4 Ω resistive load to its battery.

Charging mode operates when the UAV has been intercepted by, and lands on top

of a UGV for wireless charging. The transmitting module in the wireless charging systems, as described in Chapter 4, is assumed to be nested on top of the UGV for the UAV to land on for charging. This 1 A draw from the UGV battery and 2 A input to the UAV load battery are utilized in the model to reflect the hardware architecture that has been developed in previous research. This mode is simulated by applying a 9.3 Ω load on its battery.

As with the UAVs, standby mode is in operation when the UGV is at a charging target location before a UAV has arrived there and it is waiting for it to land on it for charging. This mode is simulated by applying a 15 Ω load on its battery. These resistive loads applied to the UGV in these simulated modes are chosen based on average results from monitoring the current draw from a Clearpath Husky UGV 24 V battery in these three operating modes. This is done to get a more realistic estimate of the energy consumption in these operating modes. The sampling of the three UGV operating modes can be seen in Figure 5.2. See [59] for more details on the Simulink generic dynamic rechargeable battery model used and see Chapter 4 for more details on the use of Stateflow for system modeling.

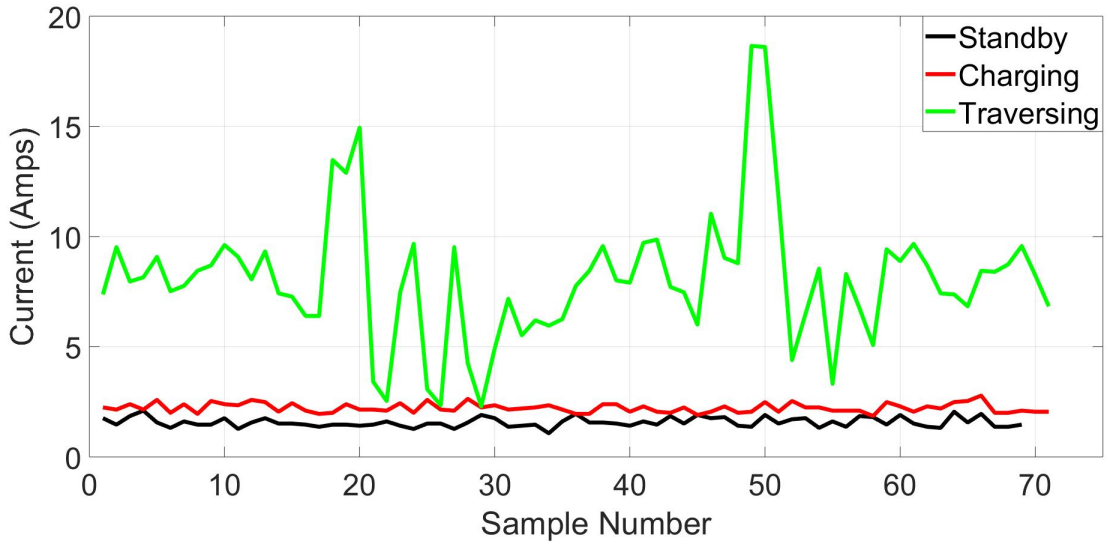

Figure 5.2: Sampling of current draw from a Clearpath Husky UGV 24 V battery in standby, charging, and traversing modes.

5.3.3 Target Areas

The operational area in which missions are conducted is considered relatively flat, and the travel trajectories are considered in 2D. The central locations of the target areas are assumed to be known and constant. These target areas are considered homogeneous in that they all take the same amount of time to be reconned, which is 10 minutes each. The euclidean distance is assumed for travel between target areas by the resources.

5.3.4 Mission Details

In all mission types investigated, all resources begin at the same deployment location of (0,0) (lower left corner of map) and all batteries at 100% SOC. A mission is considered complete when all resources return to this initial deployment location, after their mission commands have been completed. This mission completeness is a constraint and is a requirement for a potential optimal solution.

Some lower level control logic is built into the simulator when carrying out the mission. After a UAV travels to its commanded target area, there is a check to make sure it has enough energy to complete that target reconnaissance. This check is based on a constant energy decrease due to the drone flying. There is also a check if the current target area is part of the UGVs charging mission plan. If the UAV has enough energy at this point, and is not part of the UGVs charging mission plan, it performs reconnaissance on its current commanded target. If this check fails, the UAV enters standby mode and waits for a UGV to intercept it at that location for charging. After a UAV finishes reconnaissance in a target area, there is a check to make sure it has enough energy to travel to the next target area. There is also a check if the current target is part of the UGVs charging mission plan. If this check fails, the result is the same as the previous check. Because of these checks, a UAV cannot perform partial reconnaissance or partial travel to another target before it is charged. When

recharging a UAV, once it reaches 100%, charging is automatically stopped regardless of the assigned charge duration for that instance and the resources continue with their mission assignments.

With all the above considered, proper mission operations and coordination of the UAVs and UGVs are needed to optimize their utility, depending on the mission objective. Next, the GA optimization strategy is introduced along with the mission objectives pursued.

5.4 Optimization

In optimizing this mission described in its entirety, the problem for some of the case studies are split into two phases. The first phase identifies the optimal sequence of targets to be reconned by the drones. Phase-two optimizes the sequence of locations and durations in which the UGV charges the drones to support them in extending their operation. Phase-One outputs the mission plan for the drones, which is an input to Phase-Two. These optimization phases are discussed in more detail next.

5.4.1 Phase-One: UAV Recon Routing Optimization

Cases A and C utilize this Phase-One optimization because of their overarching mission objective is to minimize the mission time. The mission is considered complete when all resources have returned to their initial deployment location. This phase contributes to the mission goal by determining an optimal drone mission plan consisting of the targets and sequence each UAV should recon. For this mission objective, an initial partial desired target sequence with some tolerance is enforced to allow military members or other autonomous mobile systems to move to another forward operating base (FOB), target 11 in this case, in real-time as the target areas are deemed safe by reconnaissance. If a single drone is utilized then the objective function in this phase is

$$J_{1.1} = travel + penalty \tag{5.1}$$

where *travel* is the total distance traveled by the drone between targets, such that every target is visited once and the drone starts and ends at position 1 (initial deployment). *penalty* is a penalty cost assessed which is the result of comparing a desired sequence, with some order tolerance, to the solution found.

If multiple drones are utilized then this Phase-One objective function becomes

$$J_{1.2} = max(travel_1, travel_2, ...travel_{N_A}) + penalty \tag{5.2}$$

where $travel_{N_A}$ is the total distance traveled by the Nth UAV. This turns a component of the objective function to a mini-max objective to minimize the maximum total travel of a UAV. This is done to try to distribute travel by all the drones evenly and to share the task of reconnaissance. *penalty* is the same as described in equation 5.1.

The problem solved in this phase is similar to the traveling salesman problem (TSP) but an additional penalty condition is enforced to find a solution that prioritizes the order in which the targets are reconned. The GA is used in this phase, where the design variables are the sequence in which the targets are visited for reconnaissance in the case where a single drone is used. If N drones are deployed in the mission, the design variables are split into N sections, each in the sequential order in which the targets should be reconned by the respective drones.

5.4.2 Phase-Two: UAV-UGV Charging Optimization

This phase optimizes the sequence, locations, and durations in which the UAVs are intercepted by the UGVs for charging using the GA. For the case studies examined in this chapter, two different Phase-Two objective functions are applied for multiple mission goals. Cases A and C seek to minimize the mission time where this objective

function is represented as

$$J_{2.1} = max(time_1, ...time_{N_A}, time_1, ...time_{N_G}) + penalty \tag{5.3}$$

where $time_{N_A}$ and $time_{N_G}$ are the mission completion times of the Nth UAV and UGV, respectively. *penalty* is an additional cost if all resources do not complete their mission plans. A resources' mission is considered complete when it has fulfilled its mission plan and has returned to the initial deployment location.

Cases B and D attempt to maximize the total number of target areas reconned. This objective function is represented by

$$J_{2.2} = -1 * \sum_{N=1}^{N_A} target_N + penalty \tag{5.4}$$

where $target_N$ is the number of targets reconned by the Nth UAV. *penalty* is an additional cost if all resources do not complete their mission plans. In all case missions the UGVs' operational energy is limited to a single full charge.

This Phase-Two optimization has design variables which are represented partially by a zero/one vector of length N_T, which corresponds to all UAV's mission plans. A zero or one reflects either not intercepting or indeed intercepting a UAV at that corresponding target for charging, respectively. Another section of the design vector reflects the charging times and corresponds to the ones in the zero/one vector portion.

Table 5.1
Mission parameters for all case studies.

Parameter	UAV	UGV
Standby load (Ω)	0	15
Charge/Discharge	$12V$	9Ω
Travel load (Ω)	5	3.4
Initial SOC (%)	100	100

In this approach, the number of times a UGV recharges a UAV is a variable (variable-sized design space) and allows the optimizer to find the optimal decisions to be made.

5.5 Simulation Case Studies

Four simulation case studies are presented in this section. Of these case studies, two different combinations of UAV and UGV resources are explored. Cases A and B utilize a single UAV and UGV and Cases C and D utilize two UAVs and a single UGV. A maximum simulation time window is set to 24 hours, and the simulation will end earlier once the mission is complete.

Table 5.1 shows the UAV and UGV mission parameters used in all case studies.

5.5.1 Case A: One UAV, One UGV, 15 Targets, Minimal Mission Time

In this first case study, a single UAV is required to perform reconnaissance on 15 different target areas. A single UGV is used to provide UAV battery recharging support periodically. The objective is to minimize the mission time, as described as the objective function $J_{1.1}$ in Equation 5.1.

In Phase-One, the GA optimization results in multiple UAV mission plans with costs of 2140.5 meters. The mission plan selected for the continued Phase-Two optimization is $\{5, 2, 6, 10, 7, 11, 12, 15, 16, 8, 4, 3, 14, 13, 9, 1\}$ and is shown as the dotted blue line in Figure 5.3. Part of this result is due to the enforced penalty in the objective function which desired an initial sequence of $\{2|5, 6, 7|10, 11, 12|15, 16\}$ to prevent a penalty. The vertical line | between two target referenced numbers in this desired sequence means they both need recon, but the order of them does not matter.

The optimal solution for Phase-Two results in a mission time of 5.35 hours. The operational components of this mission time are broken down and shown in Table 5.2. The optimal UGV mission plan consists of charging the UAV at target locations $\{11, 12, 15, 4, 3, 14, 9\}$ before returning to the FOB at $(0, 0)$. The respective optimal charging times at these target locations in minutes are found to be

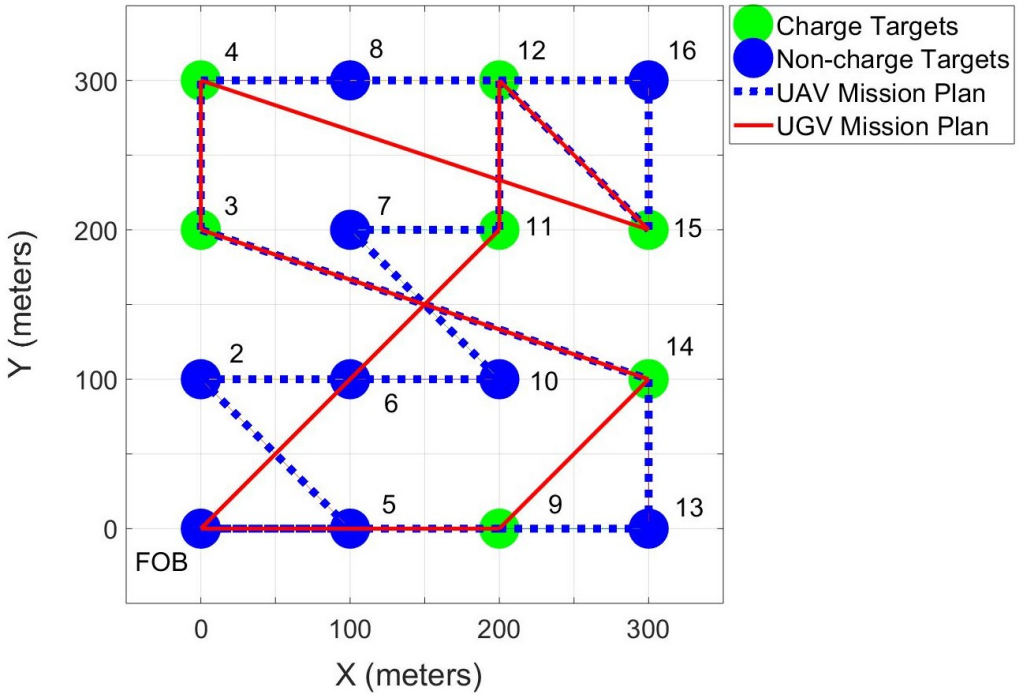

Figure 5.3: Case A UAV (dotted blue) and UGV (solid red) optimal mission routes with charging locations highlighted (green).

$\{21, 29, 28, 39, 15, 15, 15\}$. These optimal UAV and UGV mission trajectories over the course of the mission can be seen in dotted blue and solid red lines, respectively, in Figure 5.3. The SOC of the UAV and UGV over the course of the mission can be seen in blue and red, respectively, in Figure 5.4.

Table 5.2
Operational time components for Case A.

Time Measurement (Hours)	UAV	UGV
Recon/Travel	2.63	0.44
Charging	2.72	2.72
Standby	0	2.07
Mission Time	5.35	5.23

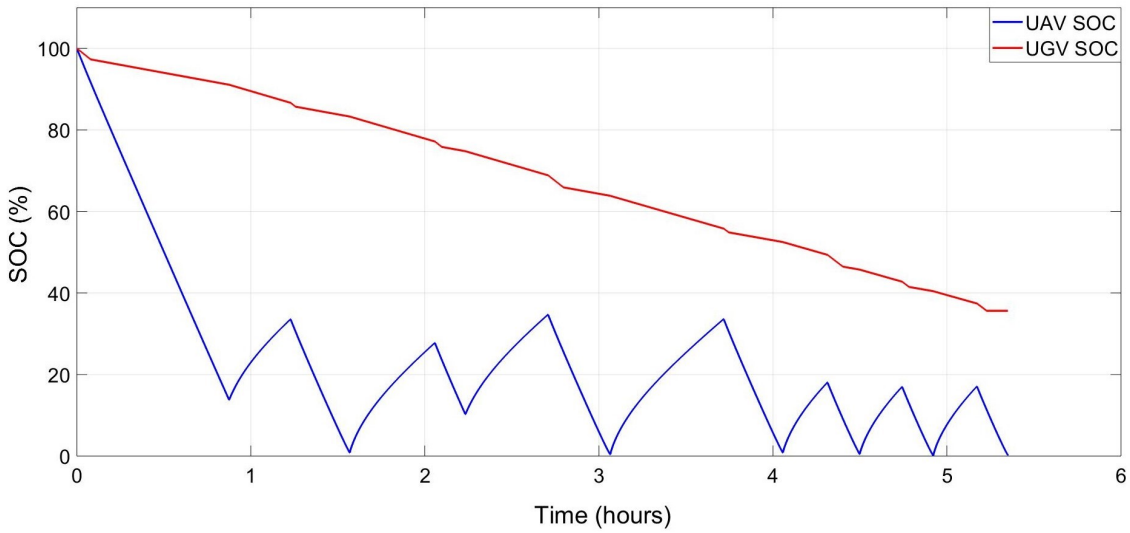

Figure 5.4: Case A mission SOC for the UAV (blue) and UGV (red).

5.5.2 Case B: One UAV, One UGV, Max Area Coverage

In this case study, a single UAV is required to perform reconnaissance on as many target areas as possible, up to 49 targets, which are organized in a 10 by 5 grid pattern. As in Case A, a single UGV is used to provide UAV battery recharging support periodically. The objective is to maximize the total area reconned by the UAV, as described in the objective function $J_2.1$ in Equation 5.3. The UGV battery is limited to a single charge which constrains both energy attributed to travel, charging the UAV, and standby. In this case, Phase-One optimization is neglected and reconnaissance is performed in a serpentine-shaped manner, resulting in a UAV mission plan that initially progresses north of the FOB towards Target 10, then east to Target 20, then south towards Target 11. This UAV mission plan is shown as the blue dotted line in

Figure 5.5

The Phase-Two optimal solution results in 19 target areas reconned with a mission time of 9.01 hours. The operational components of this mission time are broken down and shown in Table 5.3. The optimal UGV mission plan consists of charging the UAV at target locations $\{2, 6, 8, 10, 20, 19, 18, 17, 16\}$ before returning to the FOB at $(0, 0)$. The respective optimal charging times at these target locations in minutes are found to be $\{22, 110, 51, 51, 12, 32, 22, 22, 22\}$. The optimal UAV and UGV mission plans over the course of the mission can be seen in blue and red, respectively, in Figure 5.5. The SOCs of the UAV and UGV over the course of the mission can be seen in blue and red, respectively, in Figure 5.6.

Table 5.3
Operational time components for Case B.

Time Measurement (Hours)	UAV	UGV
Travel/Recon Time	3.28	0.53
Charge Time	5.73	5.73
Standby Time	0	2.03
Mission Time	9.01	8.29

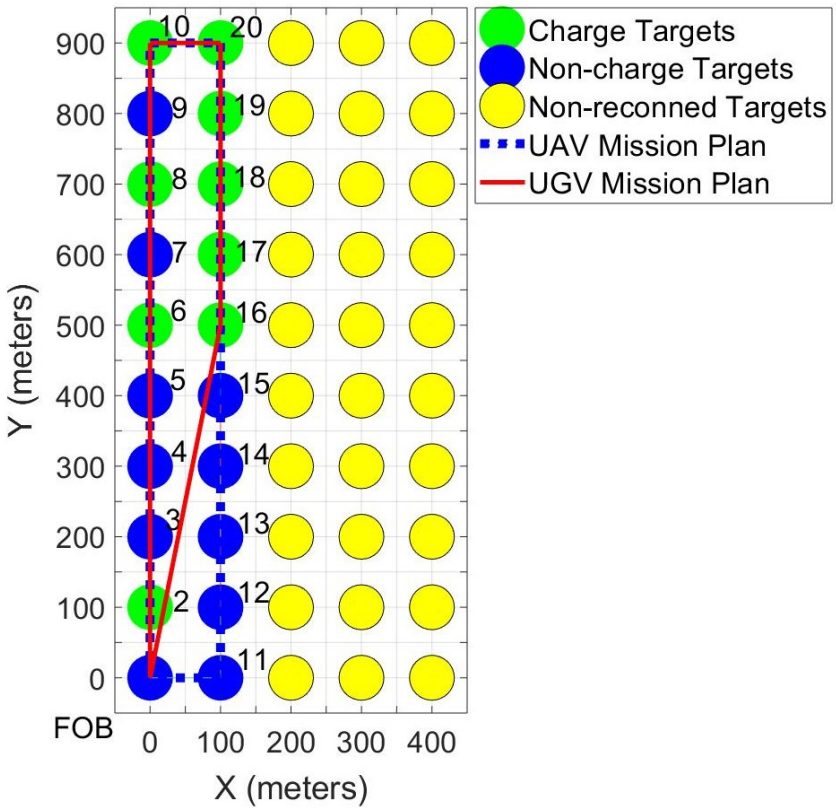

Figure 5.5: Case B UAV and UGV optimal mission routes with charging locations highlighted green.

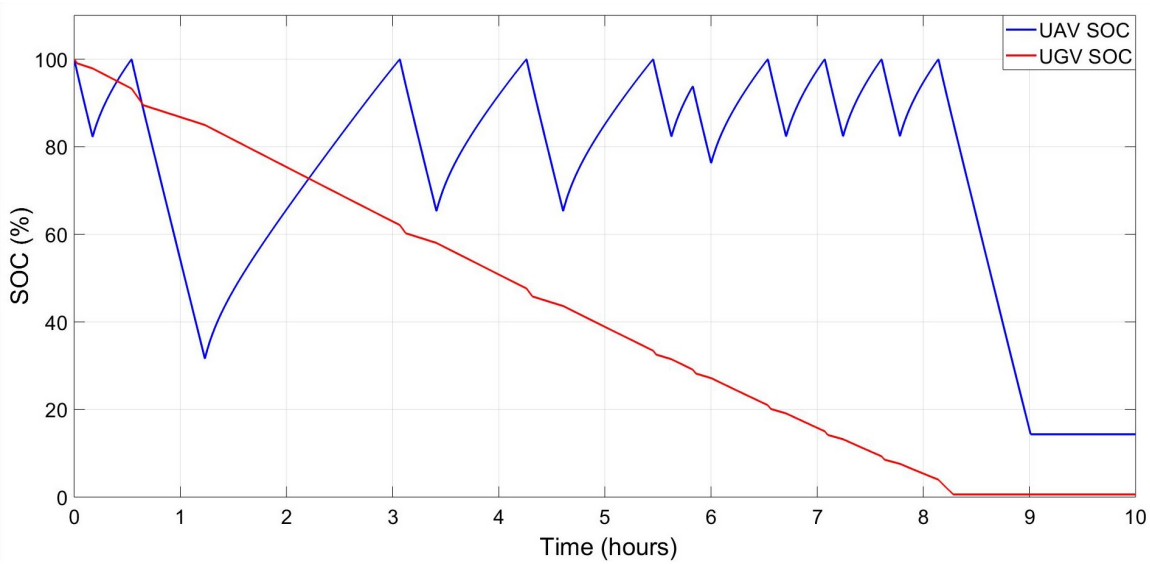

Figure 5.6: Case B mission SOC for the UAV and UGV.

5.5.3 Case C: Two UAVs, One UGV, 15 Targets, Minimal Mission Time

For a more complicated case, two UAVs are utilized to perform reconnaissance on the same 15 target areas as in Case A. A single UGV is used to provide UAV battery recharging support to the two UAVs. As in Case A, the objective is to minimize the mission time.

In Phase-One, the GA optimization results in UAV mission plans with a cost of 1141.42 meters, where UAV_1 and UAV_2 target mission plans are $\{5, 6, 10, 15, 16, 8, 4, 3\}$ and $\{2, 7, 11, 12, 14, 13, 9\}$, respectively, before returning to the original FOB at (0,0). These mission plans for UAV_1 and UAV_2 are shown as the dotted blue and dotted black lines, respectively, in Figure 5.7.

The optimal solution for Phase-Two results in a mission time of 1.96 hours. The operational components of this mission time are broken down and shown in Table 5.4. The optimal UGV mission plan comprises charging UAV_1 and UAV_2 at target locations $\{5, 16, 3\}$ and $\{11\}$, respectively, before returning to the FOB at $(0, 0)$. The respective optimal charging times at these target locations in minutes are found to be $\{8, 11, 14\}$ and $\{25\}$. These optimal UGV mission plans over the course of the mission can be seen as the solid red lines in Figure 5.7. The SOC of all the resources

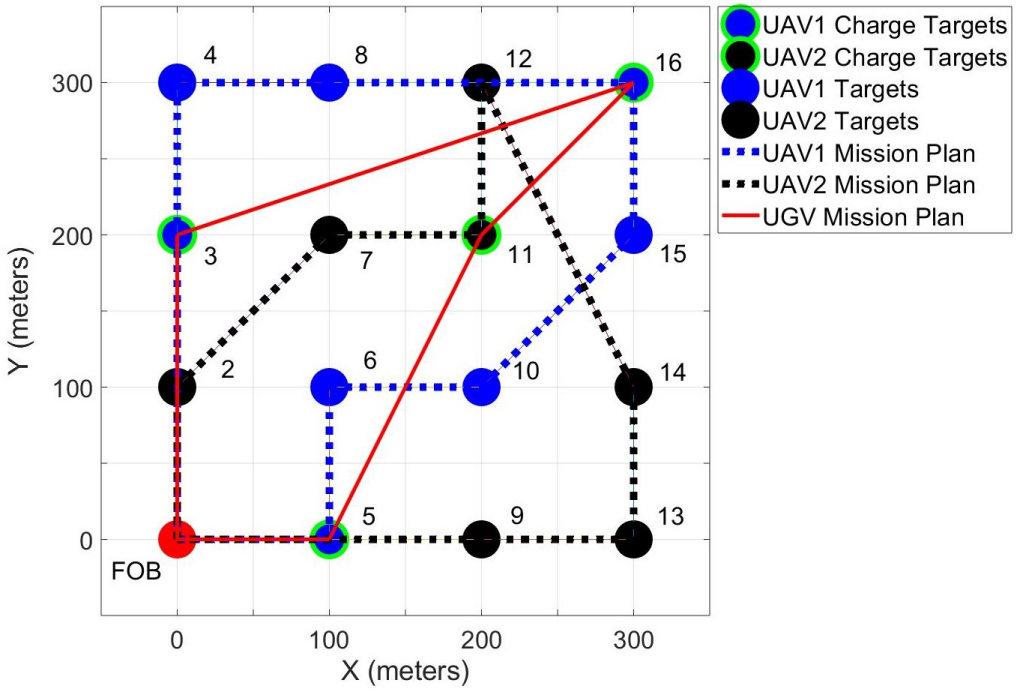

Figure 5.7: Case C UAV1 (dotted blue), UAV2 (dotted black), and UGV (red) optimal mission routes with charging locations shown with a green outline.

over the course of the mission can be seen in Figure 5.8.

Table 5.4
Operational time components for Case C.

Time Measurement (Hours)	UAV1	UAV2	UGV
Recon/Travel	1.4	1.23	0.27
Charging	0.56	0.42	0.98
Standby	0	0	0.59
Mission Time	1.96	1.65	1.84

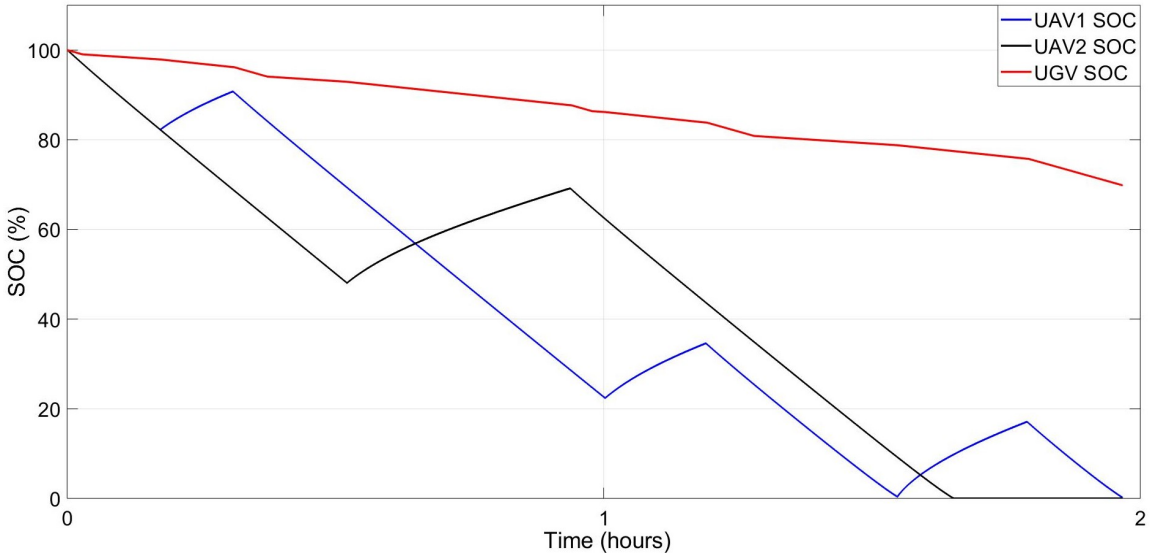

Figure 5.8: Case C mission SOC for UAV1 (blue), UAV2 (black), and UGV (red).

5.5.4 Case D: Two UAVs, One UGV, Max Area Coverage

In this final case study, two UAVs are utilized to perform reconnaissance on the same 49 target areas as in Case B. A single UGV is used to provide UAV battery recharging support to the two UAVs. As in Case B, the objective is to maximize the area reconned by the UAVs, thus maximizing the number of target areas reconned. The UAV mission plan used in Case B is also used here, but it is split up between the two UAVs. UAV_1 is assigned 25 target areas and UAV_2 is assigned 24 target areas. These are assigned on an alternating basis in the same serpentine pattern as in Case B.

The Phase-Two optimal solution results in 26 target areas reconned with a mission

time of 7.98 hours. The operational components of this mission time are broken down and shown in Table 5.5. The optimal UGV mission plan consists of charging UAV1 and UAV2 at target locations $\{4, 6, 8, 15, 22\}$ and $\{3, 7, 9, 18, 12\}$ respectively before returning to the FOB at $(0, 0)$. The respective optimal charging times at these target locations in minutes are found to be $\{162.1, 91.5, 65.9, 58.63, 34.3\}$ and $\{70.75, 155, 126, 75.75, 54\}$. It is worth mentioning again that if a UAV reaches 100% SOC during the scheduled recharging period, the remainder of the scheduled charging time is negated and the assets continue on with their respective mission plans. The optimal UAV and UGV trajectories/mission plan over the course of the mission can be seen in dotted blue, dotted black, and red lines in Figure 5.9. The SOCs of the UAV and UGV over the course of the mission can be seen in blue, black, and red in Figure 5.10.

Table 5.5
Operational time components for Case D.

Time Measurement (Hours)	UAV1	UAV2	UGV
Recon/Travel	2.33	2.34	0.52
Charging	3.17	3.41	6.58
Standby	2.48	1.99	0.56
Mission Time	7.98	7.74	7.66

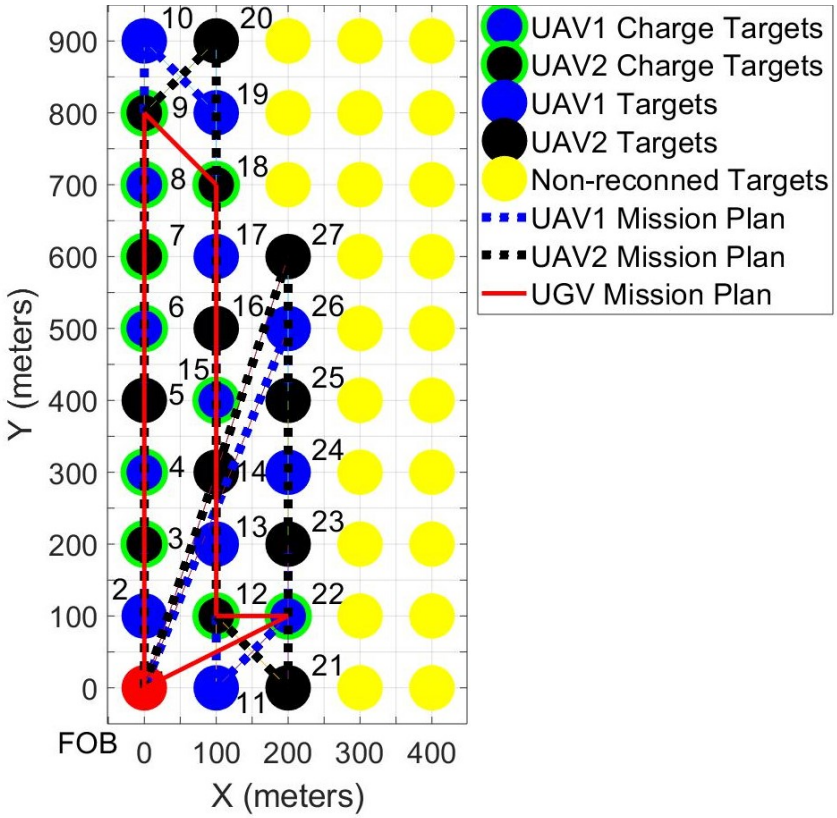

Figure 5.9: Case D UAV1, UAV2, and UGV optimal mission routes with charging locations highlighted green.

5.6 Discussion

The simulation case studies result in theoretical, optimal *a-priori* mission plans for several operations, utilizing UAVs and a UGV. In the case studies presented, two different combinations of UAVs and UGVs, each evaluated for two different objective functions, are examined.

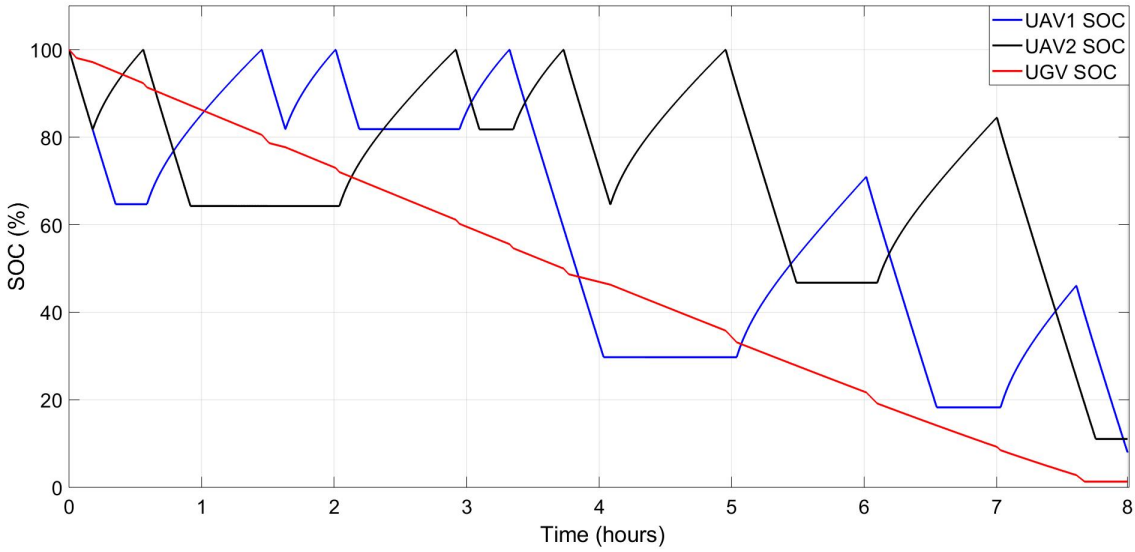

Figure 5.10: Case D mission SOC for UAV1, UAV2, and the UGV. Note the flat UAV SOC regions represent the UAVs in standby mode, waiting to be charged by the UGV.

5.6.0.1 Cases A and C: Minimal Mission Time

With Phase-One optimization applied in Cases A and C, optimal UAV mission routes are determined. A comparison of the total distances traveled in each case are shown in Table 5.6. It is noted that the use of two UAVs instead of one decreases the total travel between target areas by about 1000 meters, while still meeting the recon sequence constraint presented in these cases. This data can be interpreted to better understand the amount of stress put on one or more UAVs to achieve the mission in minimal time, and help determine an appropriate amount of UAVs to employ accordingly.

Phase-Two optimization applied to these two cases determines a UGV mission plan for

Table 5.6
Phase-One: Comparison between Cases A and C for $J_{1.1}$ and $J_{1.2}$, respectively.

Case Study	Distance Traveled Between Targets (meters)
Case A	2140.5
Case C	1141.4

optimal charging locations and respective durations. A comparison of these mission objective functions for each set of teams is shown in Table 5.7. Note that for Cases A and C, the mission time is over two times as long for Case A which uses a single UAV versus Case B using two UAVs. This comparative outcome is important because it confirms for this time critical scenario that if only two hours are available, the mission should be able to be completed in that time frame and to make the decision to utilize two UAVs instead of one. One limiting factor in this mission is the availability of the UGV to the UAVs for charging. If the UAVs have standby time and are waiting for the UGV for recharging, that is time wasted. Although this does not occur in this scenario, adding additional UAVs for this mission may not add additional benefits without additional UGVs added. An analysis of using different resources is important for these types of missions because it gives the mission coordinator a better understanding of the work involved from each resource if used and the mission outcome as a whole.

Table 5.7
Phase-Two: Comparison of Cases A and C for $J_{2.1}$.

Case Study	Minimal Mission Time (hours)
Case A	5.35
Case C	1.96

5.6.0.2 Cases B and D: Max Area Reconnaissance

Phase-One optimization is not applied in Cases B and D, and a serpentine mission routes for the UAVs are assumed. For Phase-Two optimization, a comparison of the total number of targets reconned in each case are shown in Table 5.8. It is noted that the use of two UAVs instead of one increases the total number of targets reconned by seven. One limiting factor in this scenario is the battery life of the UGV. If more UAVs were added to this scenario, one may expect there to be some point in which additional UAVs no longer significantly increases the number of targets reconned. This information can aid a better understand the limitations of using a specific combination of UAVs and UGVs.

Table 5.8
Phase-Two: Comparison of Cases B and D for $J_{2.2}$.

Case Study	Number of Targets Reconned
Case B	19
Case D	26

5.7 Conclusion

In this chapter, the general framework for a solution method is presented with a goal to optimally distribute UAVs for military reconnaissance missions, as well as optimally utilize a UGV for support in charging the UAVs. Two different reconnaissance missions are explored, including minimal mission time and maximal area coverage. Each of these missions investigated the use of different assets. The results of optimal UAV mission plans for specific targets to recon, as well as optimal UGV mission plans to intercept and charge one or multiple UAVs is shown. Using a GA solution method gives promising successful results for *a-priori* reconnaissance mission plans. This method and modeling can be extended to other types of missions and objectives such as UAV and UGV detection of IEDs. This work can be expanded to include more simulations with different team resource combinations to better analyze the optimal number of UAV and/or UGV resources for a specific mission operation.

In the next chapter, a deeper connection between the optimization applications discussed in Chapters 3, 4, and 5 is presented. The importance of each of these chapters and applications is discussed. Future work considerations are also presented.

Chapter 6

Conclusion

The work in this thesis investigates optimal resource planning and scheduling as applied to the applications of autonomous mobile microgrids, WSNs, and military ISR missions. The case studies explore optimal positions and connections in a mobile microgrid system and the optimal recharging of distributed loads involved in WSNs and military reconnaissance. The results are important to the understanding of how to make the best use of resources to accomplish an objective optimally.

In Chapter 3, the optimal mobile energy source locations and connections in a microgrid system with known load resistances, constraints, and locations are determined using GAs. Several case studies are examined involving different locations and quantities of loads to service and the number of mobile energy resources available. Multiple

objective functions are examined, presenting reasonable results, which shows the robustness of the GA utilized. Specifically, in this application, the investigation is important to acknowledge the importance of the objective function and constraints applied in the problem, regarding the mission goal.

Chapter 4 shifts the optimization perspective to include the energy availability over-time of distributed loads such as in a wireless sensor/surveillance network. Several greedy algorithmic approaches are examined along with a GA approach to determine the optimal procedure of recharging the loads. The results are evaluated and compared based on the costs of recharging UGV travel and load down-time. The work in this chapter is significant because it incorporates and optimizes the use of wireless power transfer in a sub-microgrid system, in addition to the wired power infrastructure presented in chapter 3.

Finally, Chapter 5 extends on the optimization concept in Chapter 4 by considering mobile loads as UAVs in a reconnaissance mission. The optimal UAV and UGV mission plans are determined for two different mission goals, each evaluated using two different combinations of UAVs and a UGV. The results in this chapter is influential in this area of research because they examine and compare the use of multiple UAVs in two different mission types. This aspect of optimal autonomous energy agent use builds onto the use of a sub-microgrid system which could be part of a larger group of heterogeneous energy resources in a mobile microgrid system. Each of these energy

resources could have a different functionality, with agents in Chapter 5 supporting intelligence, surveillance, and reconnaissance missions.

This thesis explains and demonstrates the contributions of the development and analysis of solution methodologies and mission simulators for *a-priori* mission plan development and testing, for applications in organizing and scheduling power delivery with mobile energy assets. Using the methods described throughout this work results in the development and analysis of reasonable *a-priori* mission plans for autonomous mobile microgrids/assets, in the various scenarios presented.

There are some additional future considerations to be examined in the continuation of this work. In Chapter 3, it would be desired to consider the energy life of the resources or priority of a specific load, and perform an architecture re-optimization at some point. This could be initiated by a source running out of energy, a new source being introduced, or load shedding. Case studies in Chapter 4 can be extended by incorporating a more complicated or random load distribution, including the number of loads, their placement, and their respective load profiles. It would also be interesting to determine some optimal number of resources needed to fulfill a mission, meeting specific energy or SOC constraints. The content in Chapter 5 could be expanded by developing the Phase-One and Phase-Two optimization procedures as a joint optimization to see if the objective function cost improves. Also, one could perform a similar Monte Carlo analysis as was done in Chapter 4 to determine performance

when faced with input uncertainties. Because of mission uncertainties, as a next step, real-time mission planning in all Chapters' 3, 4, and 5 scenarios should be considered as well.

References

[1] Tenergy, "Tenergy specification approval sheet," 2010, (Accessed 16 October 2019). [Online]. Available: https://sep.yimg.com/ty/cdn/theshorelinemarket/Tenergy-11401-01-datasheet.pdf?t=1569340118&#jmui=f%3A1426%3A305419896%3A0TxxzXRL5IC3m23ZFKcbEB-m

[2] © 2019 IEEE. Reprinted, with permission, from S. Darani and C. D. Majhor and W. Weaver and R. Robinett and O. Abdelkhalik, Optimal positioning of energy assets in autonomous robotic microgrids for power restoration, IEEE Transactions on Industrial Informatics, February/2019.

[3] © 2020 IEEE. Reprinted, with permission, from C. D. Majhor, J. E. Naglak, C. S. Greene, W. W. Weaver and J. P. Bos, Recharging of distributed loads via schedule optimization with autonomous mobile energy assets, IEEE Aerospace Conference, March/2020.

[4] R. H. Lasseter, "Microgrids," in *IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, vol. 1, Jan 2002, pp. 305–308 vol.1.

[5] R. H. Lasseter and P. Paigi, "Microgrid: a conceptual solution," in *IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, vol. 6, June 2004, pp. 4285–4290 Vol.6.

[6] O. Kramer, *Genetic Algorithm Essentials*. Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007%2F978-3-319-52156-5

[7] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010, vol. 93.

[8] P. J. M. van Laarhoven and E. H. L. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Springer Netherlands, 1987, pp. 7–15. [Online]. Available: https://doi.org/10.1007%2F978-94-015-7744-1_2

[9] B. B. Huang, G. H. Xie, W. Z. Kong, and Q. H. Li, "Study on smart grid and key technology system to promote the development of distributed generation," in *IEEE PES Innovative Smart Grid Technologies*, May 2012, pp. 1–4.

[10] C. Chen, J. Wang, F. Qiu, and D. Zhao, "Resilient distribution system by microgrids formation after natural disasters," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 958–966, March 2016.

[11] Elisa Wood, "Who uses microgrids and why?" 2017, (Accessed 27 January 2020). [Online]. Available: https://microgridknowledge.com/microgrids-businesses-institutions/

[12] L. Che and M. Shahidehpour, "Dc microgrids: Economic operation and enhancement of resilience by hierarchical control," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2517–2526, Sep. 2014.

[13] T. Dragičević, J. M. Guerrero, J. C. Vasquez, and D. Škrlec, "Supervisory control of an adaptive-droop regulated dc microgrid with battery management capability," *IEEE Transactions on Power Electronics*, vol. 29, no. 2, pp. 695–706, Feb 2014.

[14] M. D. Cook, E. H. Trinklein, G. G. Parker, M. J. Heath, W. W. Weaver, R. D. Robinett, and D. G. Wilson, "Reduced order model verification of a dc microgrid for controller design and determination of storage requirements," *International Journal of Electrical Power Energy Systems*, vol. 114, p. 105404, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014206151832684X

[15] M. D. Cook, E. H. Trinklein, G. G. Parker, R. D. Robinett, and W. W. Weaver, "Optimal and decentralized control strategies for inverter-based ac microgrids," *Energies*, vol. 12, no. 18, p. 3529, 2019.

[16] J. Gregory, J. Fink, E. Stump, J. Twigg, J. Rogers, D. Baran, N. Fung, and S. Young, *Application of Multi-Robot Systems to Disaster-Relief Scenarios with Limited Communication.* Cham: Springer International Publishing, 2016, pp. 639–653. [Online]. Available: https://doi.org/10.1007/978-3-319-27702-8_42

[17] S. Lei, J. Wang, C. Chen, and Y. Hou, "Mobile emergency generator pre-positioning and real-time allocation for resilient response to natural disasters," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2030–2041, May 2018.

[18] L. Che and M. Shahidehpour, "Adaptive formation of microgrids with mobile emergency resources for critical service restoration in extreme conditions," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 742–753, Jan 2019.

[19] A. Koubaa, H. Bennaceur, I. Chaari, S. Trigui, A. Ammar, M.-F. Sriti, M. Alajlan, O. Cheikhrouhou, and Y. Javed, *General Background on Multi-robot Task Allocation.* Cham: Springer International Publishing, 2018, pp. 129–144. [Online]. Available: https://doi.org/10.1007/978-3-319-77042-0_6

[20] I. Qureshi, "Cpu scheduling algorithms: A survey," *International Journal of Advanced Networking and Applications*, vol. 5, no. 4, p. 1968, 2014.

[21] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms—i. representation," *Computers & industrial engineering*, vol. 30, no. 4, pp. 983–997, 1996.

[22] A. Singh, P. Goyal, and S. Batra, "An optimized round robin scheduling algorithm for cpu scheduling," *International Journal on Computer Science and Engineering*, vol. 2, no. 07, pp. 2383–2385, 2010.

[23] M. K. Mishra and F. Rashid, "An improved round robin cpu scheduling algorithm with varying time quantum," *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 4, p. 1, 2014.

[24] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli, and A. Soukhal, *Single Machine Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 57–145. [Online]. Available: https://doi.org/10.1007/978-3-642-41880-8_3

[25] Y. Yin, T. C. E. Cheng, D. Wang, and C. Wu, "Improved algorithms for single-machine serial-batch scheduling with rejection to minimize total completion time and total rejection cost," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 11, pp. 1578–1588, Nov 2016.

[26] T. Liu, Y. Chen, and J. Chou, "Solving distributed and flexible job-shop scheduling problems for a real-world fastener manufacturer," *IEEE Access*, vol. 2, pp. 1598–1606, 2014.

[27] F. Cordes, S. Planthaber, I. Ahrns, T. Birnschein, S. Bartsch, and F. Kirchner, "Cooperating reconfigurable robots for autonomous planetary sample return missions," in *ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, June 2009, pp. 665–673.

[28] M. Petrlík, V. Vonásek, and M. Saska, "Coverage optimization in the cooperative surveillance task using multiple micro aerial vehicles," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct 2019, pp. 4373–4380.

[29] C. R. Calladine, *Understanding DNA : The Molecule and How It Works.* Academic Press, 2004, vol. 3rd ed. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=189462&site=ehost-live

[30] O. Abdelkhalik and S. Darani, "Evolving hidden genes in genetic algorithms for systems architecture optimization," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 1–11, 2018.

[31] P. Venkataraman, *Applied Optimization with MATLAB Programming.* New Jersey, USA: Wiley & Sons, Inc, 2009.

[32] The Mathworks Inc, "Genetic algorithm options," 2020, (Accessed 26 February 2020). [Online]. Available: https://www.mathworks.com/help/gads/genetic-algorithm-options.html

[33] A. Kwasinski, V. Krishnamurthy, J. Song, and R. Sharma, "Availability evaluation of micro-grids for resistant power supply during natural disasters," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 2007–2018, Dec 2012.

[34] C. Chen, J. Wang, F. Qiu, and D. Zhao, "Resilient distribution system by microgrids formation after natural disasters," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 958–966, March 2016.

[35] W. Weaver, M. Mahmoudian, and P. G.G., "Autonomous mobile power blocks for prepositioned power conversion and distribution," in *NDIA Ground Vehicle Systems Engineering and Technology Symposium*, 2012, pp. 1–5.

[36] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54–68, 2014.

[37] C. W. de Silva, "Some issues and applications of multi-robot cooperation," in *IEEE International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2016, pp. 2–2.

[38] A. Kwasinski, W. W. Weaver, P. L. Chapman, and P. T. Krein, "Telecommunications power plant damage assessment for hurricane katrina; site survey and follow-up results," *IEEE Systems Journal*, vol. 3, no. 3, pp. 277–287, Sept 2009.

[39] H. Sugiyama, T. Tsujioka, and M. Murata, "Collaborative movement of rescue robots for reliable and effective networking in disaster area," in *IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2005, pp. 1–7.

[40] Y. Wang, C. Chen, J. Wang, and R. Baldick, "Research on resilience of power systems under natural disasters;a review," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1604–1613, March 2016.

[41] I. Nourbakhsh, R. Sargent, A. Wright, K. Cramer, B. McClendon, and M. Jones, "Mapping disaster zones," *Nature*, vol. 439, no. 7078, pp. 787–788, 2006.

[42] S. M. Adams and C. J. Friedland, "A survey of unmanned aerial vehicle (uav) usage for imagery collection in disaster research and management," in *International Workshop on Remote Sensing for Disaster Response*, 2011, p. 8.

[43] B. Moridian, N. Mahmoudian, W. W. Weaver, and R. D. Robinett, "Robotic power distribution system for post-disaster operations," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2015, pp. 1–6.

[44] ——, "Postdisaster electric power recovery using autonomous vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 62–72, Jan 2017.

[45] K. Thulasiraman, "Circuit theory," *Encyclopedia of Physical Science and Technology*, pp. 831–841, 2002.

[46] N. C. Ekneligoda and W. W. Weaver, "Game theoretic bus selection method for loads in multibus dc power systems," *IEEE Transaction on Industrial Electronics*, vol. 61, no. 4, pp. 1669–1678, 2014.

[47] R. S. Einar S. Ueland and A. R. Dahl, "Marine autonomous exploration using a lidar and slam," in *ASME 36th International Conference on Ocean, Offshore and Arctic Engineering*, Trondheim, Norway, June 2017.

[48] S. Ahmadi Darani, "System architecture optimization using hidden genes genetic algorithms with applications in space trajectory optimization," 2018.

[49] O. Abdelkhalik and S. Darani, "Hidden genes genetic algorithms for systems architecture optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, Denver, CO, July 2016.

[50] S. Darani and O. Abdelkhalik, "Space trajectory optimization using hidden-genes genetic algorithms," *Journal of Spacecraft and Rockets*, 2017.

[51] B. Moridian, N. Mahmoudian, W. W. Weaver, and R. D. Robinett, "Postdisaster electric power recovery using autonomous vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 62–72, Jan 2017.

[52] W. W. Weaver, N. Mahmoudian, and G. Parker, "Autonomous mobile power blocks for prepositioned power conversion and distribution," in *TARDEC Ground Vehicle Systems Engineering and Technology Symposium*, 2012.

[53] A. Talukder, A. Panangadan, T. Herrington, A. Blumberg, and N. Georgias, "Autonomous adaptive resource management in sensor network systems for environmental monitoring," in *IEEE Aerospace Conference*, March 2008, pp. 1–9.

[54] J. Johnson, E. Basha, and C. Detweiler, "Charge selection algorithms for maximizing sensor network life with uav-based limited wireless recharging," in *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, April 2013, pp. 159–164.

[55] L. Li, X. Yixiang, H. Xiaoguang, D. Haibin, Z. Hanyu, C. Jun, and L. Jin, "A new rechargeable wsns based multi-uavs network and topology control algorithm," in *IEEE 10th Conference on Industrial Electronics and Applications*, June 2015, pp. 507–512.

[56] C. M. Angelopoulos, S. Nikoletseas, and T. P. Raptis, "Efficient wireless recharging in sensor networks," in *IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 298–300.

[57] N. T. Thomopoulos, *Essentials of Monte Carlo Simulation.* Springer New York, 2013. [Online]. Available: https://doi.org/10.1007%2F978-1-4614-6022-0

[58] M. R. Garey and D. S. Johnson, *Computers and intractability.* freeman San Francisco, 1979, vol. 174.

[59] O. Tremblay, L. Dessaint, and A. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *IEEE Vehicle Power and Propulsion Conference*, Sep. 2007, pp. 284–289.

[60] The Mathworks Inc, "Battery generic battery model," 2019, (Accessed 16 October 2019). [Online]. Available: https://www.mathworks.com/help/physmod/sps/powersys/ref/battery.html

[61] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," *Naval research logistics quarterly*, vol. 1, no. 1, pp. 61–68, 1954.

[62] B. Korte and J. Vygen, *Combinatorial Optimization*. Springer Berlin Heidelberg, 2018. [Online]. Available: https://doi.org/10.1007%2F978-3-662-56039-6

[63] A. P. Punnen, *The Traveling Salesman Problem: Applications, Formulations and Variations*. Boston, MA: Springer US, 2007, pp. 1–28. [Online]. Available: https://doi.org/10.1007/0-306-48213-4_1

[64] N. Pillay and R. Qu, *Vehicle Routing Problems*. Cham: Springer International Publishing, 2018, pp. 51–60. [Online]. Available: https://doi.org/10.1007/978-3-319-96514-7_7

[65] C.-M. Tseng, C.-K. Chau, K. Elbassioni, and M. Khonji, "Autonomous recharging and flight mission planning for battery-operated autonomous drones," *arXiv preprint arXiv:1703.10049*, 2017.

[66] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 287–294, Jan 2014.

[67] Y. Liu, Z. Luo, Z. Liu, J. Shi, and G. Cheng, "Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions," *IEEE Access*, vol. 7, pp. 63 504–63 518, 2019.

[68] H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air

vehicle deployed on a moving carrier," *Soft Computing*, vol. 20, no. 7, pp. 2905–2920, 2016.

[69] P. Maini and P. Sujit, "On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 1370–1377.

[70] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 128–142, Feb 2015.

[71] D. Lee, S. A. Zaheer, and J. Kim, "Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6871–6881, Dec 2014.

[72] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[73] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269,271, 1959-12.

[74] A. Koubaa, *Robot Path Planning and Cooperation Foundations, Algorithms and Experimentations*, ser. Studies in Computational Intelligence, 772. Cham: Springer International Publishing, 2018.

Appendix A

Copyright Permissions

The copyright permissions and correspondences for the papers used in this thesis are provided in this section.

Casey Majhor <cmajhor@mtu.edu> 11:43 AM (18 minutes ago)
to Shadi

Dear Ms. Darani,

I am writing my master's thesis and I would like to reuse the content in our recent publication. Since you are the senior author of this paper and a substantial portion of the original paper may be used, I was wondering if you would agree to reusing the content in my thesis.

Optimal Positioning of Energy Assets in Autonomous Robotic Microgrids for Power Restoration, IEEE Transactions on Industrial Informatics
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9424

Thank you,
Casey Majhor

Shadi Darani 11:45 AM (15 minutes ago)
to me

Hi Casey,

You have my permission to reuse it for your thesis.

Good Luck!
Shadi