Department of Computer Science

Submitted in part fulfilment for the degree of BEng

# Energy Efficient Task Allocation and Scheduling in Multi-Robot Systems using Genetic Algorithms

Ayman Omiyo Zahir

01 May 2024, v2.0

Supervisor: Alan Millard

## Acknowledgements

This accumulation of work would not be possible without the love and support of my family and friends. To my parents, who've always pushed me to challenge myself and be ambitious. To my friends, who saw something in me when I never did. To my coursemates, with whom I've shared many sleepless nights trying to make something of ourselves.

Special thanks to my supervisor, Alan Millard who listened to my ideas and ramblings and provided much needed guidance, both technically and personally.

# Statement of Ethics

This project has been conducted with no legal, social, ethical or professional implications in accordance with the University of York Computer Science Department's ethical standards. All the relevant sources have been cited, along with all the open-source software used. No personal data was used in this report.

# Table of Contents

# Table of Figures

# Table of Tables

# 1 Executive Summary

Pending

# 2 Introduction

Pending

# 3 Literature Review

This section will outline the background material behind the multi-robot task allocation problem, current approaches to tackling it and the specific application of genetic algorithms in this field.

## 3.1 Multi Robot Task Allocation (MRTA) Overview

The following subsections will give context to the field of multi-robot task allocation, outlining its definition, motivation and applications.

### 3.1.1 What is MRTA?

In recent years, robotic systems research and development have moved from deploying specialised single-robot systems to multi-robot systems (MRS) [1]. These systems use multiple robots to solve larger problems more effectively than a single robot could [2]. However, to effectively build and deploy MRS, we need to answer the question, "Which robot should carry out which task and in what order?" This introduces the Multi-Robot Task Allocation (MRTA) problem, a challenging and popular research area in MRS [3].

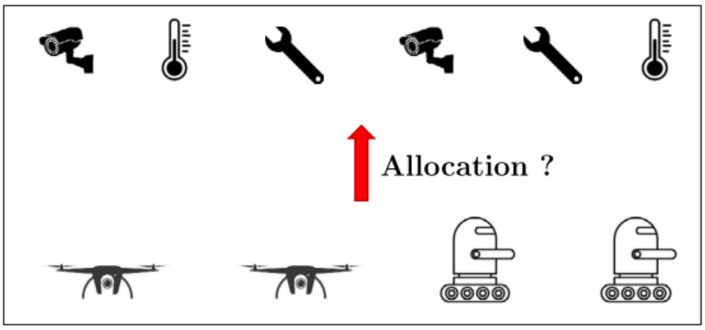



**Fig. 1.** MRTA problem presentation.

The MRTA problem is a combinatorial optimization problem [4]. It involves optimally assigning a set of tasks to a group of robots to optimise an objective function - such as system performance or cost (e.g., total time taken) subject to constraints [5]. Due to its combinatorial nature, MRTA has multiple variants. Academically, MRTA problems and their complexities are classified based on taxonomies that define the relationship between robot capabilities, task requirements, and time [6, 7].

Robot capabilities can be single-task (ST) versus multi-task (MT) robots: Robots can execute only one task at a time or execute multiple tasks simultaneously. Similarly, task requirements can be single-robot (SR) versus multi-robot (MR) tasks: Tasks require exactly one robot to complete or multiple robots. Time is considered either instantaneous (IA) or time-extended (TA) assignment: independent tasks are instantly assigned to robots without future planning versus a sequence of tasks being assigned to robots as time extends as a factor. Other factors, such as whether robots are homogeneous and task dependencies, further complicate the problem space [7].

Therefore, MRTA can be characterised into 8 problems according to these factors with an SR-ST-IA problem being the least complex. However, in more diverse MRS scenarios, MRTA becomes more complex and falls into the realm of NP-hard [8] problems, especially MR and MT problems with heterogeneous robots and multiple constraints [7]. Understanding the problem characteristics is crucial for effectively modelling and solving an MRTA problem.

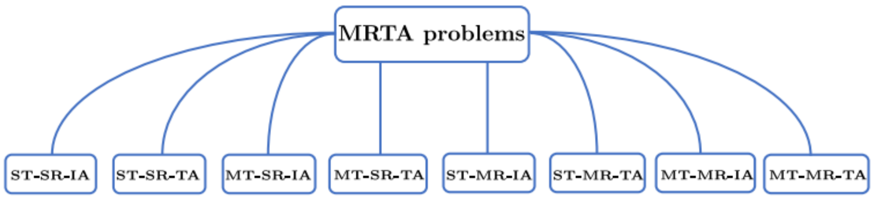



**Fig. 2.** Classification of MRTA problems.

### 3.1.2 Motivation for research in MRTA

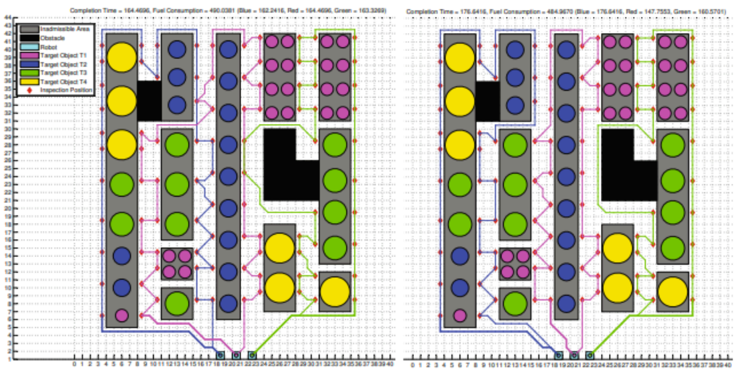
There are numerous advantages to MRS over single robot systems. They can resolve task complexity, such as moving larger or distributed objects, that may be difficult or impossible for a single robot. MRS also improves performance through parallelism, as multiple robots working together can significantly reduce task completion time. It provides robustness and reliability, ensuring tasks are completed even if one robot fails, unlike a single robot system where a failure can halt the entire operation. Furthermore, MRS design tends to be simpler for complex tasks, encouraging the use of simpler, cheaper robots [8]. These advantages have attracted application and research of MRS for a range of industrial and commercial use such as environmental monitoring and logistics.

However, the development of MRTA is vital to fully utilise MRS in real-world applications that are optimal, scalable and consistent.. Through optimization, MRTA allows the creation of systems tailored to specific objectives, making them more commercially viable by maximising task completion or minimising costs, such as travel distance or time [9, 10]. The adaptable and multi-objective nature of the objective function can also be adjusted to minimise energy usage, promoting sustainability and reducing the carbon footprint. This is particularly important given the current challenges of climate change and the environmental impact of technology, such as maintaining data centres [11] or continuous deployment of large scale multi-robot systems [12]. These factors continue to drive research in MRTA, particularly from an optimization standpoint [13].

### 3.1.3 Applications of MRTA

MRTA problems have been studied since the 90s [14], interconnecting disciplines like computer science, mathematics, operations research, and more recently, robotics and artificial intelligence. Effective application of MRTA relies not just on its classification, but also on how it's modelled across these disciplines. For instance, consider the multiple travelling salesman problem (mTSP) which allocates and optimises routes for 'm' salesmen to visit 'n' cities, starting and ending in the same city [16]. By replacing salesmen and cities with robots and tasks, the mTSP problem becomes an ST-SR-TA based MRTA problem [17]. Other related problems include vehicle routing, job shop scheduling, location routing, and linear assignment [18, 19, 20, 21].

With proper modelling, MRTA can be applied to solve real-world problems in areas such as search and rescue [22], surveillance [23], mining [24], environmental monitoring [25], and warehouse logistics [26]. For example, it has been modelled as mTSP and JSP to efficiently allocate robots for loading, unloading, sorting and assembly in automated warehouse scenarios [26, 27], as well as using teams of drones for goods delivery [28]. MRTA is also used to ensure humanitarian safety in extreme scenarios, such as inspecting crops and underground infrastructures [29,30] or forming coalitions for disaster response search and rescue missions [31,32].



However, these instances pose challenges as the systems often operate under imperfect conditions with inaccurate environmental information. This includes robot failures, unexpected events, moving obstacles, dynamic tasks, and more, making dynamic task allocation difficult [33, 34]. Hence, recently, there's been a significant increase in efforts for dynamic MRTA research [35-38]. For instance, researchers have successfully achieved a level of dynamic MRTA under task uncertainty in pick-and-place automation and drone delivery [35, 36]. Further state-of-the-art includes a game-theoretic approach for task allocation in dynamically changing environments for trash collection [37]. Additionally, recent research has focused on allocating tasks to tackle real world complexities where it's crucial to optimise multiple objectives simultaneously - ensuring safety, reducing energy and cost while still returning profits [39, 40].

## 3.2 General topology and approaches to MRTA

The following subsections outline general MRTA topologies and the three main approaches to applying and solving various MRTA related problems discussed: market-based, behaviour-based and optimization-based, with a focus on the later.

### 3.2.1 MRTA Topologies: Organisation and Assignment

MRS and MRTA can be further categorised into various topologies. In terms of organisation we have centralised and decentralised methods. In centralised methods the coordination of individual robots is intentional [x] and controlled beforehand through a central agent based on global information [41]. This facilitates efficiency by reducing duplication of effort and resources, thus saving time and costs [42]. It is the more popular method [43] due to easier implementation. However, it has disadvantages due to a lack of robustness. If the central system fails then the entire system fails. Also, the central system becomes less efficient as more robots and tasks are introduced thus restricting scalability [44]. Crucially, the computational overhead makes it difficult to implement for dynamic scenarios. Hence, it is ideal in smaller static environments where global information is easily available.

Conversely, decentral systems have no central agent. Instead allocation is emergent as robots communicate and delegate actions amongst themselves based on local information during missions [44]. It is robust and ideal for dynamic scenarios - robots can make faster real time decisions and if one robot fails the others can still complete the task without a central point of failure [45]. Due to this distributed nature it is less computational intensive and thus scalability is no longer an issue [46]. However, without global knowledge,  it can produce suboptimal solutions at best. These solutions are also emergent and thus we are not aware of them ahead of time [47]. Hybrid hierarchical approaches are also used in practice to leverage the best of both central and decentral approaches [x2].

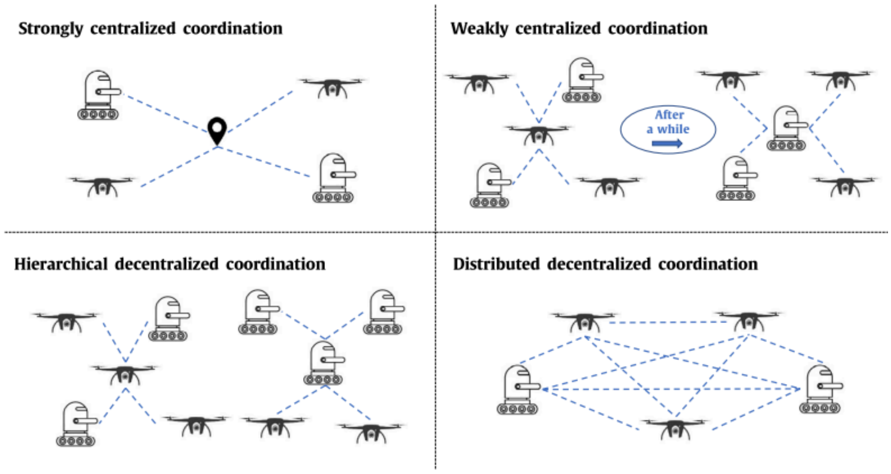



**Fig. 4.** Centralized vs. Decentralized coordination.

Task assignment can be approached in three ways: Offline, Iterated, and Online [48]. Offline assignment handles the allocation of known tasks within a known environment, with allocation done before the mission starts. This approach is ideal for centralised methods when the environment is static and tasks are scheduled, such as in warehouse automation. However, it's not suitable for dynamic scenarios where new information becomes available during the mission [49].

Iterated assignment, on the other hand, deals with the allocation of incoming tasks in dynamic scenarios. Whenever a new task is added, the algorithm releases the robots from their previous tasks, regardless of whether they've been completed, and assigns them to the remaining tasks [48].

Online assignment also manages the allocation of incoming tasks, but unlike Iterated assignment, robots don't cancel their previously assigned tasks. Instead, they receive new tasks after the completion of their previous ones. Both Iterated and Online assignments are ideal for decentralised  or real time approaches [48].

### 3.2.2 Market-based approaches

Market based approaches are inspired by concepts in economic theory such as auctions and voting. An auction involves assigning a set of goods or services to a set of bidders according to their bids and auction requirements [50]. This directly parallels MRTA and thus auction based approaches involve explicit communication and negotiation where robots bid for tasks based on their capabilities, applying market theory to optimise desired objectives [51]. They exhibit desirable features such as efficient solutions despite local information and limited resources, robustness against uncertainty [55] and scalability [52] making them popular for decentral auctioneer systems such as CNP [53] for negotiating cooperative tasks and Trader-Bots [54] for dynamic environments. However, they lack formalisation in designing appropriate functions and developing negotiation protocols [55], making them complicated to develop. Despite robustness, they generally produce poorer solutions and require excessive computation and communication compared to other approaches [56].

### 3.2.3 Behaviour-based approaches

In behaviour based approaches tasks are divided into behaviours [57,58]. These behaviours motivate robots to perform tasks according to predefined rules for prioritisation and behaviour combinations without inter-robot negotiation. ALLIANCE [59, 60] and BLE [61] are both pioneering decentral algorithms in this approach. More recent work attempts MRTA by drawing behaviours from game theory [62], social choice theory [63] and animal behaviours [64].

### 3.2.4 Optimization-based approaches

Optimization based approaches have emerged as the most popular and diverse method to solving MRTA related problems, outperforming other strategies in terms of performance and numeric complexity [65-67]. They exist in a wide variety and thus help solve a range of MRTA problems of different natures and complexities. However, their biggest strength lies in the ability to globally explore new solutions in the search space due to the randomness of stochastic algorithms used [68-69]. Optimization approaches fall into two categories: deterministic, which guarantees the repeatability of optimal solutions, and stochastic, which employs randomness to find diverse solutions for larger optimization problems.

The deterministic Hungarian Algorithm (HA), introduced in 1955 [70], is a widely used solution to MRTA problems. It has been refined over the years [70-80] and applied to warehouse control and search and rescue operations [71, 72]. However, it struggles to efficiently manage scenarios where the number of tasks surpasses the number of robots, a frequent situation in current multi-robot systems [x11].

Another notable deterministic approach is ILP and mixed linear programming (MILP) models. Recently, ILP [73, 74] and MILP [75] were applied successfully to implement MRS for disaster responses, mapping survival kits to potential victims. MILP was further used for maximising coverage in unknown environments [76]. While providing optimal solutions for the real world that outperform HA [77], they lack robustness and are computationally expensive, especially for the large scale problems in dynamic MRTA seen today [78].

Other deterministic techniques include graph-based methods such as uninformed and informed search. Recently, reinforcement learning techniques were applied to graph neural networks to handle larger MRTA problems [79, 80] at limited capacity.

As deterministic methods become computationally intensive for larger problems, stochastic methods provide a faster alternative to finding optimal solutions. This makes them useful for large scale real time solutions. Trajectory or heuristics-based methods like simulated annealing and tabu search [81, 82] rely on a single solution that traverses the search space to find the global solution with non-zero probability. However, single solution methods tend to get stuck in sub-optimal solutions at local optima [x10].

On the other hand, population or metaheuristics based methods solve problems from a population of candidate solutions through an iterative process that continually improves the population [83]. It constitutes genetic algorithms (GA) from evolutionary computation [83] and particle swarm (PSO), ant colony (ACO) and bee colony optimization (BCO) techniques from swarm intelligence [84]. These bio-inspired methods, which simulate the efficiency and adaptability found in nature, have emerged as a promising approach for MRTA.

Particularly, genetic algorithms emerge as the most popular optimization approach for solving MRTA problems [85]. Pivotal research was conducted in studies [86-89] that used GA to solve allocation and routing problems for environmental inspection of fixed locations in known environments. Since then, more advanced GA have been applied in various MRTA scenarios both centrally and decentrally [90, 91], and in hybrid with other methods [92-94]. While

GA is slower to converge to global solutions, it ensures diversity of solutions and is thus ideal when optimising for multiple objectives closer to real world scenarios. Many of them optimise for energy consumption making GA a more sustainable approach [list].

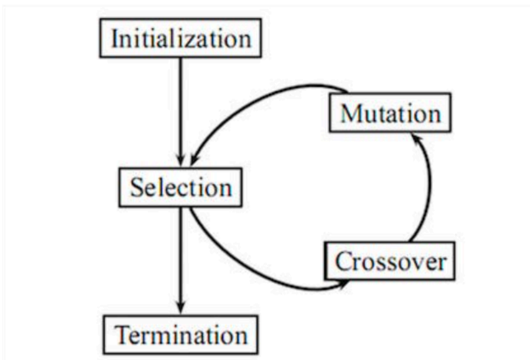
## 3.3 Evolutionary Algorithms for MRTA

The following subsections outline evolutionary algorithms, focusing on genetic algorithms and how they are utilised for MRTA.

### 3.3.1 Evolutionary algorithms

Evolutionary algorithms are inspired by Darwinian evolution and natural selection [95]. The power of evolution in nature is evident in the diverse species in our world and thus provides an alternative approach to finding optimum solutions by mimicking evolution.

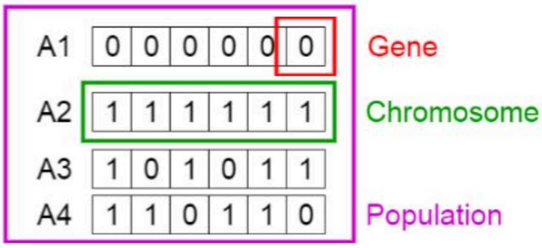
Initially, a population of candidate solutions represented as genomes [96] is randomly generated and evaluated against some fitness function and the best solutions are selected for the next generation. Mutations and crossovers are then applied to the genes of the selected population to add random variations, hoping to find new and improved solutions for the next generations. This process is repeated, mimicking evolution, until some termination condition is fulfilled, as depicted below [97].




Given the increasing complexity of MRTA problems, these algorithms provide a suitable alternative. Among these, genetic algorithms have gained popularity [98]. They tackle the MRTA problem by representing it as a genome. A population of genome solutions is incrementally evolved by optimising a fitness function through specific mutators and crossovers tailored for allocation and scheduling issues [99]. Designing and evaluating the genome and fitness function is challenging, and much effort has been put into creating efficient solutions. However, most of these solutions have not been tested in real-world conditions, leading to a "reality gap" [100] when transitioning from simulation to reality. This remains a significant challenge in MRTA development.

### 3.3.2 Genetic Algorithms in MRTA

Genetic algorithms (GA) are amongst the most popular evolutionary algorithms and they show superior performance for solving scheduling problems [105] like MRTA. To improve their performance, a variety of genetic operators have been developed for various problems. These operators include encoding schemes, crossover, mutation, selection and defining the fitness function.




The encoding scheme defines how to represent the problem as a population of genomes, consisting of individual genes. Genomes are typically depicted as arrays [102], and the genes within them are encoded based on the given problem. The schemes include binary, octal, hexadecimal, permutation, value and tree based encoding. The goal is to select a scheme that produces a genome that most closely resembles the problem. For instance, binary encoding represents genomes as bit arrays - making them quick to mutate but difficult to represent problems [103]. Permutation encoding, typically used in MRTA [104], presents genome arrays as a sequence of numbers - making it suitable for problems involving task ordering. However, designing MRTA problems as genomes is challenging and requires more advanced genome representations for specific problems such as dividing the array or using multiple separate arrays to represent tasks, robots or constraints separately [106, 107].

Selection is crucial in the convergence of the evolution process towards good solutions as it determines which individuals are selected for reproductions, ensuring that the better individuals make up the latter generations while also ensuring the population remains diverse to avoid local optima. Two of the most common techniques include roulette wheel [108] and tournament [109] with the former performing well in mTSP problems [110].

Crossover operators create new and diverse offspring genomes by combining elements from two or more parent genomes. These operators identify and carry over superior parts of the genomes, preserving them in subsequent generations. Notable crossover operators include single-point, two-point, k-point, uniform, partially matched (PMX), order (OX), precedence preserving (PPX), shuffle, reduced surrogate (RCX), and cycle. Designing crossover operators for permutation-based genome representations in MRTA can be challenging. The objective is to exchange genome bits without disrupting beneficial orderings. Therefore, operators such as PMX, OX, PPX, RCX, cycle, and edge crossover have been developed.

While OX crossover provides better exploration than other techniques, they are less efficient in solving TSP [111]. Similarly, cycle and RCX tend to converge prematurely towards local optima instead of global. Overall, PMX has become the most popular operator in MRTA due to its superior performance and exploration capabilities. Proposed in 1985 to solve the TSP, it's now one of the most used operators for ordering problems [112].

Mutation operators help maintain the genetic diversity of populations across generations. They facilitate the search for global optima by applying random mutations to the genes, mirroring biological processes. Typically, certain or random genes from the genome array are slightly modified. The level of mutation and its probability is regulated by parameters that require fine-tuning. Moreover, the type of mutation used depends on the encoding and the problem itself. For permutation in MRTA, it is not viable to consider each gene independently for mutation, as this could disrupt the optimal task order. To tackle this, well-known operators like swap, shuffle, insert, scramble and inversion mutation are used. However, other mutators such as uniform mutation are often combined in MRTA [113].

-fitness function important
- Problems and challenges:
- end with justification of research

Look into how it works in general followed by specific application to MRTA (add examples of candidate uses)
- genome representation
- **fitness function**
- crossover operators
- mutators
- selection (less focus)

Additionally delve into the common problems with GA in general and in MRTA

- Link with main mrta paper for mrta specific gaps
- Link with main ga paper for ga specific gaps
    - All the above + extras (trim to focus on research questions)
    - Robot task ratio

FINISH THIS SECTION

### 3.3.3 NSGA II: Multi-Objective Optimization & Pareto Fronts

### 3.3.4 Online vs Offline Simulation

These are not needed for now

# 4 Methods and Implementation

## 4.1 Problem Definition

## 4.2 DEAP Framework: Justification

### 4.3 Minimal Simulation using PIL: Justification and Coding

## 4.4 Implementation of GA

### 4.4.1 Genome Representation

### 4.4.2 Fitness Functions

### 4.4.3 Crossover Operators

### 4.4.4 Mutation Operators

### 4.4.5 Hyperparameter Setup

### 4.5 Project Aims

Base aims through implementation of **Q1, Q2** (see below):

- **fine tune best parameters for the 2 fitness functions**
- **evaluate base fitness function vs enhanced**
- **scalability of tasks and robots over time against fitness across 2 solutions - find best tradeoff**
- qualitative evaluation of solutions outside of fitness metrics

But could potentially extend to (these depend on implementation of **Q3, Q4, Q5**):

- **do all of the above for more complex MRTA scenario**
- **evaluating & comparing NSGA 2 against enhanced fitness functions solutions**

# 5 Results and evaluations

This section will outline the background material behind the multi-robot task allocation problem, the current approaches to tackling it and the application of genetic algorithms in this field. Process of experiment justified

## 5.1 Using base fitness function F1

The following subsections will give context to the field of multi-robot task allocation, outlining its definition, motivation and applications.

## 5.2 Using enhanced fitness function F2

The following subsections will give context to the field of multi-robot task allocation, outlining its definition, motivation and applications.

### 5.3 Single vs Multi-objective / Effect of robots and tasks

The following subsections will give context to the field of multi-robot task allocation, outlining its definition, motivation and applications

# 6 Conclusions

Pending conclusions

# Appendix A

Delete as necessary.

Fusce congue mauris leo, nec sagittis justo convallis quis. Maecenas varius tellus vel nulla facilisis accumsan. Nunc in facilisis nulla. Suspendisse ac molestie ante. Duis vulputate sit amet tellus eget efficitur. Cras ut dolor at dolor molestie dignissim. Proin eu massa dapibus, feugiat quam auctor, porttitor urna. Integer porttitor, libero a egestas lacinia, neque leo luctus elit, id iaculis leo quam porta est. Integer ut porttitor tortor.

# Bibliography


| | |
|---|---|
| [1] | J. Zobel, Writing for computer science, Springer, 2015. |