Towards a Fully Autonomous Robotic System for Intra-logistics Applications: Applications in the Oil and Gas Industry

by

Juan Manuel Tzintzun Ramos

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Management

Department of Mechanical Engineering

University of Alberta

Abstract

Intra-logistics plays an important role in industry and service activities. However, current intra-logistics systems have several shortcomings for which mobile robots are the most suitable candidate to solve them. Nevertheless, the use of mobile robots should firstly overcome several challenges for becoming a realistic solution. We narrow the scope of this work and focus our attention to improve planning abilities (motion planning, task planning, task allocation) of robots used in intra-logistics operations. Towards that end we approach our problem from a formal methods perspective as it has been shown it may be a solution that can successfully improve important abilities desired by such robotic applications. In this work we present a mixed integer linear programming formulation for the multi-vehicle traveling salesman problem with pick-up and delivery and split load constraints as a new formulation able to capture a bigger set of instances. We then use recent developments in decomposition of formulas given in a subset of linear temporal logic to propose the multi-robot pickup and delivery problem with linear temporal logic. Finally, we explore possible applications of the models presented in intra-logistics operations in the oil and gas industry.

Preface

This thesis is the original work of Juan Manuel Tzintzun Ramos. Two journal papers related with this thesis have been submitted. As such, the thesis follows a hybrid structure.

1. **J. Tzintzun**, T. Nakashima, J. Doucette, R. Ahmad, "Multi-vehicle traveling salesman problem with pick-up and delivery and split load constraints", *Robotics and Autonomous Systems*, (submitted on August 3, 2020).

2. **J.Tzintzun**, T. Nakashima, J. Doucette, R. Ahmad, "Formal methods for the general pick-up and delivery problem", *Journal of Intelligent Robotic Systems*, (submitted on August 3, 2020).

Pongo mi corazón en el futuro. Y espero, nada más. De los dos monsílabos prefiero el más claro, el sencillo, el que despliega un lienzo en el que todo podrá ser. El amor dará firmeza a lo que digo. Estoy con los que creen sin ver, con los que andan sobre las aguas. Cuando el mundo entero o mi mundo se hunden tantas veces, entonces algo relacionado con los pájaros y los lirios me salva. Entonces tengo todas las palabras. Sueño palabras. Fluctuat nec mergitur. Prefiero abril. No sé cómo decirlo. En una calle estrecha de Venecia he encontrado una casa con un lema breve sobre el dintel, inscrito en piedra hace siglos, legible todavía, que franquea la entrada. Ancora spero. Tenemos que elegir. Esa es mi puerta.

– Juan Antonio González Iglesias, CONFIADO, 1964.

Acknowledgements

I would like to thank my supervisors Dr. John Doucette and Dr. Rafiq Ahmad; and Dr. Benjamin T. Nakashima for their support and constant guidance while working on this thesis. Additionally, I would like to thank my colleagues Yufan Zheng, Feiyu Jia, Saraswati Jituri, Mario Soriano, David Baca, Abraham Reyes-Yanes, Abril Alvarado, Samira Doostie, and Tiange Shi for their advice and company during this journey. I would like to thank my family and friends whose support was essential for the completion of this thesis. A special thanks to my friend Nydia Pando for her valuable source of advice and company while we both walked similar paths, a special thanks to my mother Gloria Ramos and father Artemio Tzintzun whose guidance and teachings have been constant through my entire life, a special thanks to my sisters Martha Tzintzun, Karina Tzintzun, Araceli Tzintzun, and Lidia Tzintzun whose love and affection transcends borders. Finally, I would like to thank my partner Gerardo Almada for his unconditional support while completing this journey. Additionally, the author acknowledge the support of this work by Consejo Nacional de Ciencia y Tecnología (CONACyT) and Banco Nacional de Obras y Servicios Públicos (BANOBRAS) through CONACyT-SENER funding 777319-617985-460905.

Contents

List of Tables

List of Figures

List of Symbols

1-PDTSP - One-commodity traveling salesman problem with pickup and delivery
AGV - Automated guided vehicle
BPS - Bosch production system
C - Capacity constraints
cs-LTL - Co-safe linear temporal logic
DARP - Dial-and-ride problem
DCOP - Distributed constraint optimization problem
ERP - Enterprise resource planning
FMHS -Future material handling systems
FSA - Deterministic finite automaton
GPD - General pickup and delivery problem
HSE – Health safety and environment
INC - Increased navigation capability
LTL - Linear temporal logic
MAPD - Multi-agent pickup and delivery
MAPF - Multi-agent path finding problem
MES - Manufacturing Execution System
MILP - Mixed integer linear programming
MP - Motion planning
MPDTSPS - Multi-vehicle traveling salesman problem with pick-up and delivery and split load constraints
MRS - Multi-robot system
MRTA - Multi-robot task allocation
MTL - Metric temporal logic
NFA - Non-deterministic finite automaton
OG - Oil and gas
OP - Online planning
P-VRP - Persistent vehicle routing problem
PA - Product automaton
PD-LTL - Pickup and delivery with linear temporal logic
PDP - Pickup and delivery problem
PDPTW - Pickup and delivery problem with time windows
PERR - Package-exchange robot routing
R - Reconfigurability
S - Scalability
SMPTA - Simultaneous motion planning and task allocation
SMPTP - Simultaneous motion planning and task planning
STL - Signal temporal logic
STPMPTA - Simultaneous task planning motion planning and task allocation
STPTA - Simultaneous task planning and task allocation
TA - Task allocation
TP - Task planning

TS - Transition system
TSP - Traveling salesman problem
TWTL - Time window temporal logic
VRP - Vehicle routing problem
VRPB - Vehicle routing problem with backhauls
VRPPD - Vehicle routing with pickups and deliveries

Chapter 1

Introduction

Intra-logistics is the planning, realization, operation, and optimization of in-house material handling and information flow. It offers opportunities for cost reduction in the manufacturing and the service sector, [1]. Although a lot of systems are available for this purpose, they generally require a complex multilevel control architecture, including integration among systems as enterprise resource planning and other systems, site control, process control software, etc., [2]. These complex system architectures have reached their limits, [2], and flexible and adaptable systems are now required, [2]-[3]. Recently, automated guided vehicles (AGVs) have been used as a key component for intra-logistics systems, [4] , and increasing autonomy has been suggested as a good approach for eliminating problems related to complicated system architectures, [2]-[4]. Clearly, there are still some other factors to overcome before we have an autonomous robot for intra-logistics operations. As an example, while there exist several AGV systems available in the market, for most of them their autonomy is limited by the fixed locations, [5]-[6], where they are restricted to work (predefined paths), which can decrease the production efficiency and have higher energy consumption rate, [7]. Recent studies that keep track of available technologies and current research directions in the robotics field, [8], and provide a deep analysis and guidelines on future directions in intra-logistics and material handling systems, [2], has shown the importance of working towards a new generation of more intelligent, and robust robotic systems that will be able to work alongside human operators in a more collaborative manner. There are a number of notable mobile robots, including Drive Units, a mobile robot originally developed by kiva Systems, [9], and later acquired by Amazon Robotics. These robots are mobile units that retrieve items from storage locations in a warehouse whose path planning through the warehouse relies on a pre-defined graph and the A* algorithm, [9], however, Amazon Robotics has recently announced its will to increase the autonomous mobility of its robots at the same time they announced new mobile robots, Pegasus and Xanthus, [10]. However, the current system cannot be directly applied to manufacturing environments since it assumes a human free environment, [11]. A similar option is the KMP600 used by CarryPick Systems, [12], by Swisslog and KUKA, which follows a comparable structure as Kiva Kystems; mobile robots move on a human free grid, and provide point to point transportation. TRANSCAR, [12], is another mobile unit developed by Swisslog for logistics operations in hospitals. The robot still relies on pre-defined paths. Furthermore, the management system for commanding the robots requires an inter-connection among different management systems available in the hospital. Another interesting project is the PAN-Robots project, [11], that aims to increase the autonomy of AGVs used for pallet transportation working in industrial scenarios alongside human operators. AGV motion is constrained to a predefined roadmap and AGVs still rely on physical infrastructure for correct functioning. The central controller concerned with navigation and mission assignment of the robots is divided into two layers. The top layer deals with task allocation and motion coordination; this top layer system uses the Hungaraian algorithm, [13], and D* algorithm, [14], for the task allocation and for addressing coordination on the robots path, respectively. The bottom layer aims for mobile units to use the A* algorithm, [15], for local path planning. TUG robots, [16], are another set of centrally controlled mobile robots for logistics operations on hospitals. These robots operate with a higher degree of mobility as they do not rely on any physical pre-defined paths (wires, strips ,etc.) however, the systems follow a hierarchical architecture for commanding the robots. FIFI is another robot for intra-logistics operations. Its main particularity is, that it can be commanded by human gestures, [17]. In-house material transportation is still performed manually, in most cases by a human operator with help of mechanical devices such as electric tugs. FIFI robots allow human operators to use the mobile unit for intra-logistic operations without being physically exhausted as the FIFI robot is capable of following, and recognizes other gestures commanded by the human operator. Nevertheless, this drive unit still depend on a human operator to guide the robot. Autobod by Bosch Production Systems (BPS), [6], is another prototype that emerged from the difficulties to adapt existing mobile robots for intra-logistics operations to the BPS. This mobile mobile rely on a pre-defined physical paths. KARIS systems, [3], consists of several mobile robots with docking capabilities. Individual mobile units use ARMO or PRIOR algorithms for motion planning, and DHHT approach for the task allocation problem. Open Shuttle by KNAPP is a mobile unit for diverse material handling tasks. It is centrally commanded by the KNAPP fleet management system that uses swarm intelligence for the mission commanding of the robots. The robots can autonomously navigate through the environment without the need of pre-defined paths. MultiShuttle by Demantic is a system of multiple mobile units that used swarm intelligence to coordinate the robots.

1.1 Intra-logistics

Our objective is to work toward an autonomous robotic system for intra-logistics operations. As stated above, intra-logistics refers to the planning, realization, operation, and optimization of in-house material handling and information flow, [18]-[19].

Intra-logistics encompasses a variety of systems where there is a flow of in-house material or information, [4], which can be modeled as performing a sequence of activities (i.e, moving, picking, sorting, visiting, reading data, sending, etc.), [20]. One example is an AGV used for moving material or collecting data in oil and gas (OG) facilities, [21]-[22]. Our goal in this thesis is to focus on the use of mobile robots and the potential offered by increasing their autonomy as an alternative to alleviate the current crisis of complicated system architectures, [23].

1.1.1 Current intra-logistics systems

Traditionally intra-logistics systems are rigid and difficult to adapt to the different conditions and the changing requirements of various industries. These systems follow a hierarchical multilevel control architecture of subsystems interacting with each other. Typically they consist of between 5 and 8 levels, [2], where normally the ERP is on the top of the hierarchy, following by a manufacturing execution system (MES). The latter is connected to the material handling coordinator, the warehouse management system, and the manufacturing process control system, [2]. At the base of the hierarchy, we find the AGVs, and/or the storage and retrieval systems, both commanded by the former systems. In Figure 1.1 we illustrate a sample architecture, as described. We can see the existing dependency on the interrelated nature of the system for its correct operation. For example, normally, AGVs are centrally commanded to perform material handling operations from upper decision levels like local controllers, material handling coordinators, and all the way to the top by ERP, [2], systems. We mentioned previously that complex system architectures have reached their limits, [2], as we have to deal with more complex organizations. The main shortcoming of the current paradigm is its lack of flexibility, as the complex the architecture makes it more difficult to adapt to the changing requirements encountered in highly variable industrial environments. This lack of flexibility impacts directly on the profitability of automated material handling systems, [2].

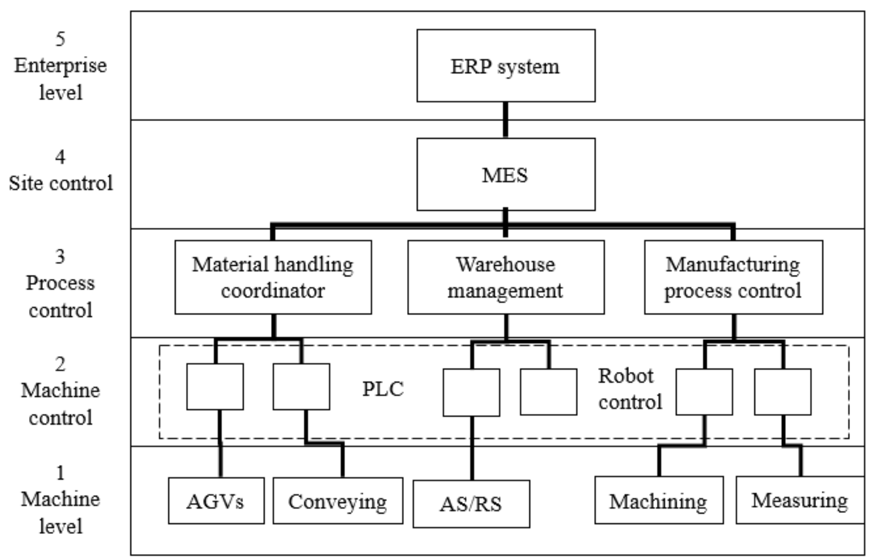

Figure 1.1: An example of a multilevel control architecture (adapted from [2])

Intra-logistics systems in the oil and gas industry

Given the importance of oil and gas in our province, and in the authors's home country, we will look at our work through an OG industry lens. Intra-logistcs in the oil an gas industry has not been entirely described as to the best of the knowledge of the authors. Nevertheless, from definition of intra-logistics we can easily assume it has been practiced. The OG industry can be divided into: exploration, upstream, midstream, refining, and petrochemical sectors, [24]. We excluded from this study the exploration sector, which main objective aims finding natural resources to be exploited. We also excluded midstream operations; midstream operations are the link between extraction or production sites and refineries, which almost uniquely goes through pipelines. Therefore we only describe intra-logistics in the context of the upstream, refining, and petrochemical sectors.

Upstream sector covers facilities for oil and gas production like onshore/offshore wells; refining locations transform natural resources like gas an petroleum into usable materials like gasoline; petrochemical industries use as input refineries products to produce more processed materials like olefins, and aromatics, [24]. We do not intend to give a complete description on the intra-logistics over the different sectors. Instead we summarize the most common intra-logistics operations related with mobile robots in the oil and gas industry, which are: inspection (gauge readings, valve and lever position readings), monitoring (gas level monitoring, check for leakage, acoustic anomalies, surface condition, check for intruders), maintenance (gas and air detector test, gas sampling, pigging, cleaning, refilling), [25]-[26]. Transportation and logistics are important areas for introducing robotic applications, [27], furthermore, it has been shown that operational staff spends a considerable amount of time walking, and transporting things, [28], therefore we can add transportation and material handling into the list since mobile robots performing material handling can decrease the time used in such activities. We review robotics applications about previously listed operations in Chapter 4.

1.1.2 Future material handling systems

Although the term intra-logistics is relatively new there exists a fair amount of research literature about the topic. One of the earliest uses of the term is in a 2012 German publication, [18], and in a 2006 English-language paper, [19]. Available research literature is sparse, therefore, this section is entirely based on, [2]. Material handling systems are categorized in two kinds: connection-based and trip-based, [2]. Connection-based systems (e.g., like conveyors), are systems where goods are moving while the system itself remains static, in contrast to trip-based systems where the system itself moves (e.g., AGVs). The main takeaway of the work of, [2], are three main points: 1) some desired properties for future material handling systems (FMHS) for intra-logistics operations, 2) a set of functions one should improve in order to attain the set of desired properties, and 3) a set of design patters that can help to achieve the desired properties. In this section we will only discuss the desired properties and design patters. The set of functions that drive the development of FMHS will be discussed in Section 1.2 alongside the target abilities detected by, [8]. We list the set of desired properties proposed by, [2] as follows:

1. What You See is What You Get.
2. Plug-and-Play (Plug-and-Work)-capability.
3. Scalability.
4. Reconfigurability.
5. Reliability.
6. Inherent safety.
7. Resource efficiency.
8. Self-adaptability.

Desired property 1 states that deployment of mobile robots for material handling operations should not rely or depend on physical layouts as such any mobile robot should have the capacity to operate in a diverse range of physical locations without the associated cost of adapting the environment to the robot. This can be associate with the deployment phase of a robotic system, which typically means to add physical landmarks or define physical paths for the robots to follow, [29]. The idea is to avoid AGVs systems rely on a hierarchical organized structure which is costly, and static, [4], since they depend on the infrastructure (pre-defined paths) of the physical location they perform their activities.

Desired property 2 refers to robotic design has been widely studied and the state-of-the-art in the area already promise versatility, robustness, and low-cost robotic systems, [30]. The main goal of a re-configurable design is to add, remove or change the systems capabilities simply inserting new components.

Desired property 3 address the idea of a system able to adapt imposed requirements by increasing or decreasing its capabilities. In the context of mobile robots for material handling we see scalability as a multi-robot system able to handle the complexity associated with working with several agents. For example, research projects that main goal is to create multi-robot coordination algorithms with more realistic assumptions, [31].

Desired property 4 mentions the need for re-configurable robotic systems. Although there exist robotic system that shown interesting behaviors these are mostly hand-coded by programmers in a large and tedious process. Recent developments on the use of formal methods to command missions to the robots have already shown really good results, which can definitely be a step forward to address point 4 as such approaches allow non-expert users to specify behaviors or goals, without having to worry about the technicalities of the task, [32], i.e., internal functionality, programming the related script, etc.

Desired property 5 talks about how reliable is a system to complete the task assigned. There always exist the possibility of system components failing.

Individual failures should not make the entire system fail; in contrast the system should be easy reparable, and replaceable if necessary. If there exits failures the system should be aware and repair itself. For example, a multi-robot systems for pick-up and delivery task that considers transfers of material among robots when there exist a failure or delay, [33].

In desired property 6, safety undoubtedly is a main concern about autonomous mobile robots working alongside humans, specially on complex environments such as manufacturing scenarios. This point, inherent safety, aims for a system whose functionality will not endanger people or the goods being transported. For example, the use of motion planning algorithms with safety guarantees to find motion trajectories of a robot that works in an environment with humans, [34].

Desired property 7 is concerned with the correct use on the resources needed by the system to operate. From the perspective of mobile robots for material handling the use of resources is commonly addressed in the problem formulation. For example, in optimal motion planning the objective is to find time optimal trajectories; in multi-robot systems a metric to optimize could be the energy used by the system, [35].

Desired property 8 refers to the ability of the system to adapt to changes in patters that might be found in the flow of material. One should think in different products than can be assembled in the same manufacturing cell, each of them with different patters in the flow of material over the workplace. Till date the authors are not aware of any robotic system for material handling that consider this learning feature and adaptability.

One can easily observe why the use of mobile robots with a high degree of autonomy are being adopted as a key component in intra-logistics systems. They can address every one of the desired properties on future material handling systems for intra-logistics operations. Furthermore, the author suggested design patters to help to achieve such desired properties: 1) modularity and function integration, which incentives modular design. The modules should be independent and easily integrable. 2) decentralized control, individual agents should rely on their own controlling mechanism instead on a central control unit. 3) interaction refers to design system with mechanism for exchanging information among each other. Finally, 4) standardized physical and information interfaces for integration of several modules of the systems.

1.2 Towards an autonomous robotic system for intra-logistics applications: Planning

In Section 1.1.2, we summarized some desired characteristics in future material handling systems and presented a brief discussion of why there exists a paradigm shift from fixed mobile robots to higher autonomous general purpose mobile robots, [23], to justify our focus on the use of mobile robots for intra-logistics operations. As we focus our attention on mobile robots, we should also consider the state of current robotic systems in different application domains as well as their future research directions and targets. Important work in this direction is the *Robotics Roadmap*, [8], a European initiative whose objective was to present an overview of the state and future research directions in robotics. They categorized various target abilities, each activity having different development levels. The set of abilities considered are: configurability, adaptability, interaction ability, dependability, motion ability, manipulation ability, perception ability, decisional autonomy, and cognitive ability. Moreover, most robotic systems are built over several interrelated subsystems which follow a control cycle composed of: perception, localization/mapping, cognition/planning, and acting. See Figure 1.3, for an illustration of how these systems interact.

In our work, we narrow the scope of our research by focusing on the cog-

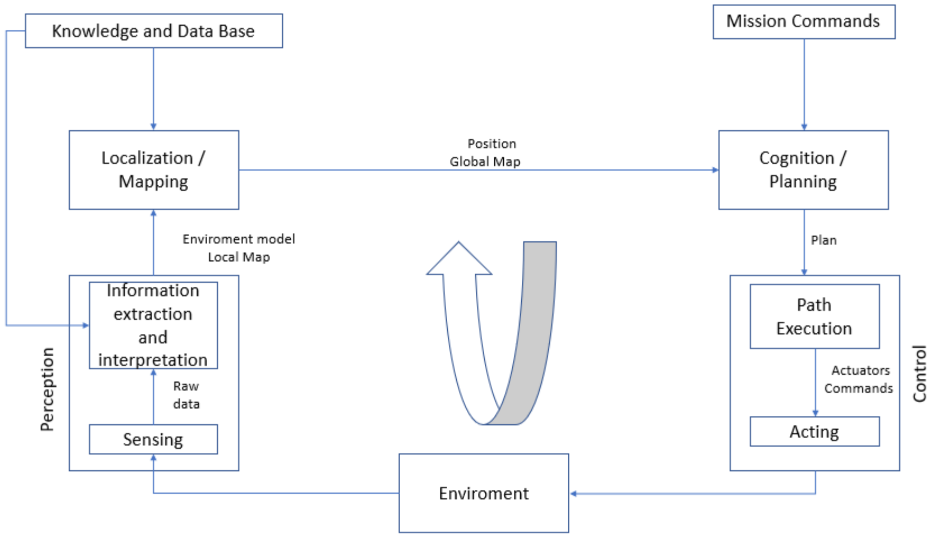

Figure 1.2: Control cycle for mobile robots(adapted from, [36])

nition/planning module. Under the cognition/planning module, we identify several ability targets for robotic systems in intra-logistics operations, and whose development aligns with the desired characteristics in Section 1.1.2. We summarize those ability targets as follows:

1. *Increased navigation ability (motion planning).* It refers to the ability of the robot to move autonomously through diverse environments without the need to adapt pre-defined paths. This target ability is a highly desired characteristic for mobile robots working on industrial settings, commonly used for material handling operations. Normally, they rely on pre-defined paths which are physically implemented as floor lines, [37]-[38]. Therefore, robots are confined to specific locations, and any change in the workflow on the physical location will imply changes on the workplace. This goes in the opposite direction with desired property 1 that main goal is to avoid customization of physical locations, [2]. Future intra-logistics systems should be able to avoid significant changes in the physical layout; therefore, increasing mobile robots navigation ability aligns perfectly with the desired properties for future intra-logistics sys-

tems. Furthermore, mobile robots for intra-logistics applications are normally required to carry on goods for transportation. Goods characteristics like weight should be consider when finding feasible motions for the robots as their dynamics change according to such goods properties, [29].

2. *Task allocation.* As stated before, flexibility is an important feature mobile robots for intra-logistics operations must posses. This flexibility could be given by increasing the autonomy of the robots in which case the *Task allocation* problem is involved, [3]. It refers to the sequencing and allocation of different sub-tasks into robots to achieve a higher level task. The task allocation problem appears in the context of multi-robot settings, when there exist the need to partition a global goal in different sub-tasks that can be allocated to individual robots. Fuerthermore, desired property 3 states the ability of the system to be up and downsized, as needed, to account for different requirements imposed to the system. It is not difficult to see this could be achieved by adding or removing robots.

3. *Integrated task planning, task allocation, and motion planning.* Task planning refers to find a sequence of activities that drives the robot to complete mentioned task; similarly motion planning refers to find a trajectory that drives a robot form an initial to a final configuration. Task allocation refers to the assignment of task to robots. Therefore, the task and motion planning objective is to find a trajectory that drives a robot through a set of locations which sequence describes a set of activities that drives the robot to complete a task. The motion, allocation, and task sequencing are not operations that can be independently performed without affecting the optimality or feasibility of the independent parts.

This target ability aligns with the desired properties in Section 1.2 since integrated task and motion planning can offer reconfigurability. It offers reconfigurability since the control cycle for mobile robots shows mission (task) commanding is an essential input for the planning module. See Figure 1.3. However, complex task commanding normally requires high-level manual programming skills covered by advanced programmers, [39], to program the robots to perform such tasks. The use of formal methods can reduce that dependency.

Furthermore, the three previous target abilities also align with desired property 7, and further development of *Order and energy management* function proposed to develop FMHS, [2]. It refers to resource efficiency, and resource allocation designated to fulfill a task or order. Normally, there is a central controller that assign the order to different robots. Various algorithms can be used for the assignment,e.g. First in, first out assignment, or solvers for vehicle routing problems, [2]. However, a decentralized control should be preferred in which case a coordination strategy should be deployed. In this point the authors add energy management, which generally refers to the optimization of the energy used for the system. As we describe before, the objective of the target abilities is to optimize certain metric (e.g., time, distance, cost) that translate in a better use of resources.

As conclusion, there are several options for mobile robots out there in the market, however, they do not see the problem from an integrated perspective (task allocation, task planning, and motion planning) and furthermore, they do not contemplate the flexibility offered by formal methods to command complicated task to the system.

1.3 Problem statement

Mobile robots with higher degree of autonomy have been shown to be important in overcoming the limitations of to increasingly complex intra-logistics systems. Furthermore, we have presented three target abilities that have been shown to be important in achieving key desired properties for mobile robots in intra-logistics scenarios. With those findings in mind, the objective of this section is to formalize the problem we are addressing.

A mobile robot for intra-logistics applications should focus on performing two basic operations correctly: 1) moving material and 2) collecting and storing information. Moving material involves pickup and drop-off of material among different physical locations. The flow of information does not necessarily involve traveling between locations, though there are situations where the robot may need to move between locations to keep the flow of information moving (e.g. reading data from a sensor in a specific location). Therefore, we can address the problem from the perspective of the class of problems belonging to the *general pickup and delivery* (GPD) problem. The GPD perspective was initially motivated to account for the optimal movement of material, (i.e., the optimal sequence of actions to perform all the pickups and deliveries of material in some set of requested points while considering the capacity of the robots). One important reason for us to consider GPD is that they have been widely studied before, and can add valuable insights. We stated our problem as follows:

Problem 1.3.1. Given a set R of robots represented by a dynamical system $\dot{x}_r = f(x, u, \mu)$ for $r \in R$, where $x \in X \subseteq \mathbb{R}^m$ represents the phase space of the robot, $u : [0, T] \rightarrow U \subseteq \mathbb{R}^l$ represents the control input and $\mu : [0, T] \rightarrow Q_r \subseteq \mathbb{R}$ the changing mass or the robot. Assume the robots are placed in a world $W = \mathbb{R}^2$ with and obstacle region $C_{obs} \subset W$, and a free region $C_{free} \subset W \setminus C_{obs}$

where we also have a set O of objects with related positions y_o and an available quantity m_o at such location for all $o \in O$, a set D of delivery locations with their related positions y_d and request m_d for all $d \in D$. Let M be the mission commanded to the system. The problem is to find a set of $x_r\langle t; x_0, u(\cdot), \mu(\cdot)\rangle$ of optimal time trajectories for all $r \in R$ under the control input $u(\cdot)$ and mass changes $\mu(\cdot)$ over the time interval $[0, T]$ and initial condition x_0, such that the discrete trajectory $\beta_r(x_r\langle t; x_0, u(\cdot), \mu(\cdot)\rangle)$ parameterized by them describe a sequence of actions that accomplish the set of allocated subtasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_{|R|}\}$ in the sense that the completion of $\mathcal{T}_r$ for $r \in R$ implies M is achieved (robots pickup and deliver all the objects) while that maximum capacity C^r_{max} for all $r \in R$ is respected.

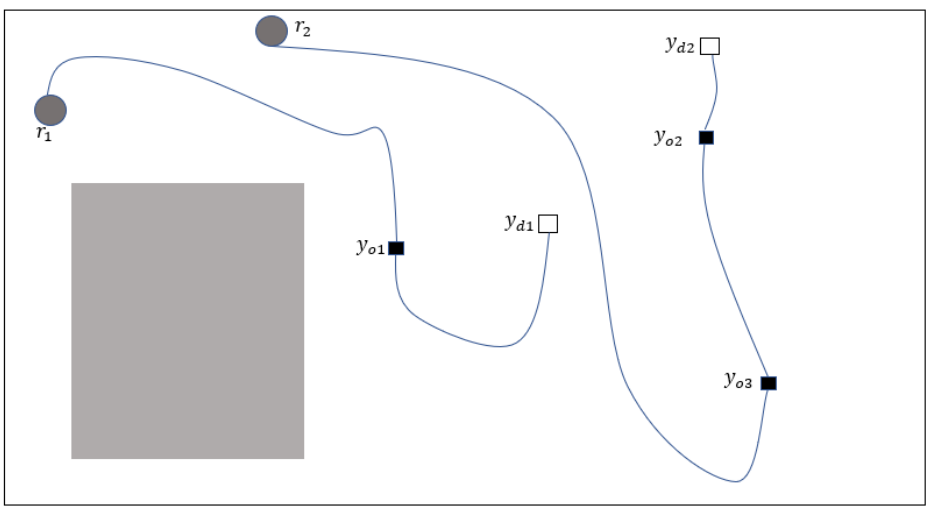

Figure 1.3: Visualization of problem 1.3.1

Figure 1.3 shows a visualization of problem 1.3.1. We have a set of two robots (circles), a set of objects to pick up (black squares), and one delivery location (white square), where $M =$"pickup all the objects and drop them off at the delivery location". The trajectories are the two solid lines.

1.4 Research objectives

Our main research objectives are:

1. Propose a mixed integer linear programming formulation for the *multi-vehicle traveling salesman problem with pickup and delivery and split load constraints.*

2. Propose a formulation of the *pickup and delivery problem with linear temporal logic.*

3. Consider the above two goals in the context of robotic systems for intra-logistics operations in the oil and gas industry.

1.5 Organization of the thesis

The structure of the thesis is as follows. Chapter 2 presents an MILP formulation of the *multiple traveling salesman with pickups and deliveries ans split load constraints.* In chapter 3 we present the *multi-robot pickup and delivery problem with linear temporal logic.* In Chapter 4 we briefly describe the importance of autonomous systems for intra-logistic operations in the oil and gas industry. Finally, in Chapter 5 we present the conclusions of our work.

Chapter 2

Multi-Vehicle Traveling Salesman Problem with Pickup and Delivery and Split Load Constraints.

In this chapter we address problem 1.3.1 under the following remark:

Remark. There already exist a road-map dictating the possible motions of the robots (pre-defined paths); and the problem under considerations neither does it consider the dynamics and kinematics of the nor the presence of obstacles. Under this assumptions we are dealing with just the TP problem (pickup and deliver operations), and the TA problem, i.e., the STPTA.

The concept of intelligent logistic solutions is one of the most important research directions on intelligent manufacturing systems, given the significant amount of waste reduction that is expected by their introduction, [1]. Several combinatorial approaches have been proposed for robotic decision making in manufacturing floors, [7, 40]. In this paper we propose a (0-1) mixed integer linear programming (MILP) formulation for the pick-up and delivery problem (PDP) to address the material handling problem on a set of robots working in a manufacturing site. Several works exist in this area, but there does not appear to be a single formulation that fully captures all the characteristics of our application domain. To the authors' knowledge, our formulation is the first 0-1 MILP formulation for a many-to-many PDP with multiple origins, multiple finals, and split load constraints for a structurally heterogeneous set of robots. This paper is structured as follows. In Section 2.1, we provide a quick overview on PDP and related work. Section 2.2, presents the proposed formulation that captures a more realistic scenario in which it is possible to collect a proportion of the demand/supply and generalize to the case of a node that can be visited more than once. In Section 2.3, we discuss the outcomes of our computational experiments. In Section 2.4, we provide an extension of our methodology to solve an instance of the multi-robot task assignment (MRTA) problem. Finally, in Section 2.5, we present final remarks and a concluding discussion.

2.1 Background and related work

Over the last decades, a considerable amount of research has been done on the *vehicle routing problem* (VRP), [41]-[42]. This is, in part, motivated by the increasing number of real-life applications, such as freight transportation and logistics, [43]-[44]. A special case of the VRP arises when a set of pick-up and delivery requests are defined where a customer demand needs to be collected or delivered to the customer premises (or other location) by a vehicle or fleet of vehicles available at the depot (i.e., VRP with pick-up and delivery), [45]. The problem is to find a minimum cost tour that satisfies all requests while ensuring the capacity of vehicle is not exceeded. As per [46]-[47], the *general pick-up and delivery problem* (GPDP) is divided into two main classes. The first class is *vehicle routing problems with backhauls* (VRPB), where a set of commodities to be delivered must be loaded at one or several depots and all picked up goods must be transported to one or several depots. The second class is *vehicle routing problems with pick-ups and deliv-*

eries (VRPPD), often simplified as *pick-up and delivery problems* (PDP) in which a set of commodities must be transported between nodes. This latter class can be further subdivided into *unpaired requests* and *paired requests*. The first type refers to the situation in which the commodities to be transported in a graph are homogeneous, which means each unit can be used to fulfill the demand, i.e., no pairing relations are needed. The second class refers to problems with paired requests, when there is an assignment for every pick-up node to a respective delivery node. In this paper, we focus on the unpaired PDP class, i.e., problems in which single-commodity objects are transported between origins and destinations. However, it must be noted that the problem can be extended to handle paired relations as shown later in the paper. In the work of [45]-[48], the authors also classify the PDP into (1) *many-to-many* problems, where each commodity may have multiple origins and destinations and every location may be the starting or destination of multiple commodities, (2) *one-to-many* problems, where every location has both, pick-up and delivery requests that the vehicle has to fulfill, and (3) *one-to-one* problems, where we have point-to-point paired relations, and for every pick-up request there is an associated delivery point. Furthermore, [45] provided a framework for the classification of PDP problems. The classification system called for a three-field scheme, delineating the problems along three primary parameters, [*Structure, Visits, Vehicles*]. The first field, *Structure*, specifies the number of origins and destination for the commodities; these could be *many-to-many* ("M-M"), *one-to-many-to-one* ("1-M-1"), and *one-to-one* ("1-1") problems. The second field, *Visits*, provides information about the pick-up and deliver operation sequence on the vertices. "PD" indicates that each customer is visited exactly once for both pick-up and delivery, "P-D" is for the case when the task might be executed together or separated, and "P/D" is for the case where every request has either a pick-up or delivery task to perform but not both.

In the last field, *Vehicles*, the number of vehicles in the system is indicated. PDP may be further classified into *static* problems, where all the information is available at the outset, and *dynamic* problems, where the information required is gradually revealed over time. The problem addressed in the present paper is a *static* problem where the motivation behind our model comes from raw material handling in a manufacturing scenario; we allow the split of the load to improve the performance of the system. The novelty of the proposed model is that we provide a *binary integer linear programming* (BILP) model that considers a multiple origin depot along with multiple final destinations, and split delivery constraints. The authors in [43] provide a comprehensive literature review of various classes of VRP, but none appear to have all of the characteristics of the problem and subsequent model proposed in our work. Figure 2.1 shows the various classes of GPDP. To our knowledge, a problem which considers multiple origins, multiple finals and split load with a heterogeneous set of vehicles has not been addressed in the literature. In addition to material handling in a manufacturing setting, our problem can also be used to represent an instance of the *multi-robot task assignment* (MRTA) problem, where a mobile robot completes pick-up and delivery tasks in an obstacle-free environment. The latter property (an obstacle-free environment) can be relaxed if one computes the trajectories of the robot before optimizing, i.e., if for every edge (i, j), we compute a trajectory from location i to location j. There are several problems in the literature that are related to the problem we are studying. In [48], they formulated the *pick-up and delivery problem with time windows* (PDPTW). This formulation aims to minimize the total routing cost while each node is visited once. It also accounts for precedence constraints, i.e, there are pairing relations to impose that every vehicle visits a pick-up location before visiting delivery nodes. A new compact formulation for the PDPTW is proposed in [49]. This is a two-index formulation, which the authors claim is useful for medium-sized problems. This formulation can be seen as an alternative formulation of the one presented in [48], where the number of constraints and variables seems to be significantly reduced, due to the fact that it is generalized to the case of k vehicles being used to perform the delivery and pick-up tasks. However, the authors only consider a single depot problem, i.e., the vehicles all start at the same location. In [50], the authors propose a MILP formulation for a multi-depot PDP; the difference is that they not consider split load constraints, and more than one vehicle can start at the same initial depot. In our work we assume, there is only one vehicle per initial depot.

Our model can be derived from the open version of the *one-commodity traveling salesman problem with pick-up and deliveries* (1-PDTSP), first proposed by [51] and classified as PDTSP in [47], where an optimal tour is computed such that all pick-up and delivery tasks are completed. Later, in subsequent work, [52] and [53] proposed a branch-and-cut algorithm and developed a heuristic, respectively, for solving the 1-PDTSP. Finally one of the authors in [51] extends the work and generalizes the 1-PDTSP to permit splitting of the load, though they did not consider multiple depots and multiple destinations in [54].

According to [46], the single vehicle PDP with unpaired relations can be formulated from an open version of the *traveling salesman problem*, when the vehicles are not required to go back to the depot. This class of problems is generally also classified as PDTSP, and the 1-PDTSP falls under this classification. In this paper we will use the PDTSP interchangeably with 1-PDTSP, as in [46], to refer to the same problem initially presented in [51]. PDTSP problems rely on various assumptions, e.g., load balance, single commodity scenario, only one possible final location, a single initial depot, and they do not consider split load constraints. The objective of this Chapter is to gen-

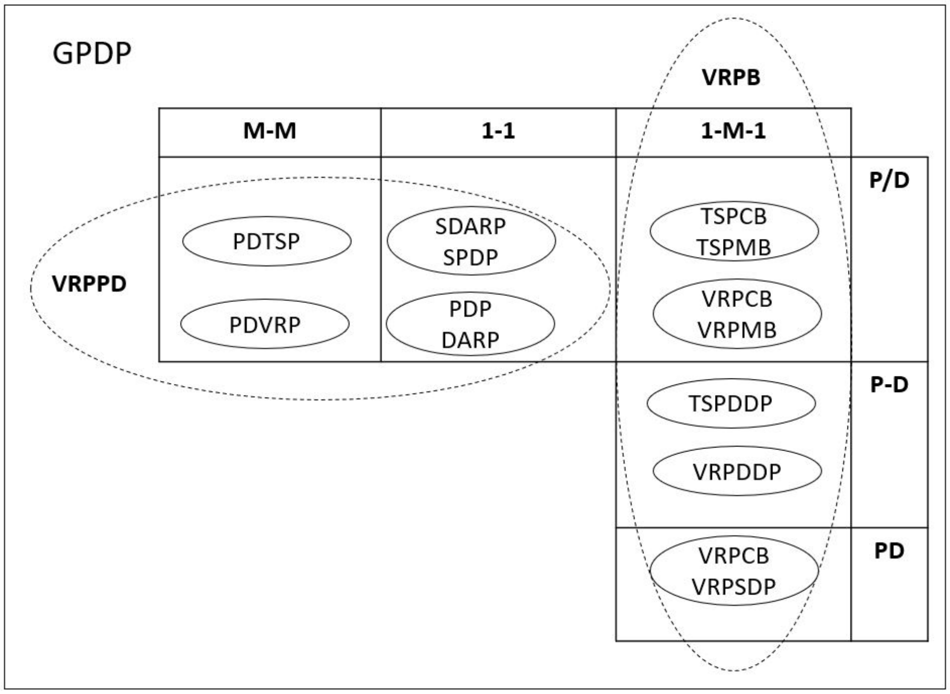

Figure 2.1: Visualization of different classes of pickup and delivery problems

eralize the PDTSP to a more realistic scenario and relax the assumptions of the initial formulation, (consider multiple initial depots, multiple final nodes or final depots, and load split constraints) to what we call the multiple vehicle traveling salesman problem with pick-up and deliveries and split load constraints (MPDTSPS).

2.2 Formulations

In this section, we present common definitions for both problems, the PDTSP and the MPDTSPS; then the formulation of the PDP, as presented in [47]. Finally we will present the MPDTSPS formulation we propose. As our proposed application is in a manufacturing facility with mobile robots, we will use "robot" instead of "vehicle".

2.2.1 Benchmark formulation: PDTSP

$G = (V, E, w)$ is an undirected weighted graph, where $V = \{I \cup M \cup F\}$ is the set of nodes in the graph, n is the number of pick-up nodes, $\breve{n}$ is the number delivery nodes, $\bar{n}$ is the number of initial depots, and $\hat{n}$ is the number of final locations. $I = \{1, .., \bar{n}\}$ represents the subset of origin nodes, $P = \{\bar{n}+1, .., \bar{n}+n\}$ represents the subset of nodes that serve as pick-up locations, and $D = \{\bar{n}+n+1, .., \bar{n}+\breve{n}+n\}$ represents the subset of nodes that serve as delivery locations. $M = P \cup D$, and $F = \{\breve{n}+\bar{n}+n+1, \dots, \bar{n}+\breve{n}+n+\hat{n}\}$ is the set of final nodes which are the nodes where the vehicles are expected to finish their tour. Finally, E is the set of all arcs, so it consists of the union of:

1. $\{(i, j)\ \forall\ i\ \in I\ \ \forall\ j\ \in M\}$,
2. $\{(i, j)\ \forall\ M\ \in I\ \ \forall\ j\ \in M,\ i \neq j\}$, and
3. $\{(i, j)\ \forall\ \ i\ \in M\ \ \forall\ j\ \in F\}$

Together with the edge and vertex sets above, there is a function $w : E \rightarrow \mathbb{R}_{\geq 0}$ that maps every edge (i, j) to a cost value, normally time or distance, to go from i to j. $K = \{1, .., |I|\}$ is the set of available robots. We define a capacity CP_k associated with every robot. q_i is the amount of commodity supplied or delivered in node i, which will be positive if it represents a pick-up task, and negative if it represents a delivery task. x_{ij} is a binary decision variable that equals 1 if robot travels from vertex i to vertex j, and 0 otherwise. v_j is the real-valued variable representing the load of robot k after visiting node i. t_{ij} is an input parameter for the time the robot will take to go from vertex i to vertex j.

$$\underset{x}{\text{minimize}} \quad \sum_{i,j \quad (i,j)\ \in\ E} t_{ij} x_{ij} \tag{2.1}$$

$$\sum_{i:(i,j)\ \in\ E} x_{ij} = 1,\ \forall\ j \ \ \in\ M\ \cup\ F \tag{2.2}$$

$$\sum_{j:(i,j)\ \in\ E} x_{ij} = 1,\ \forall\ i \in I \cup\ M \tag{2.3}$$

$$v_j \geq\ v_i +\ q_j - \mathcal{M}\,(1 -\ x_{ij})\,,\ \forall\ \ (i,j) \in E \tag{2.4}$$

$$CP_k + q_i \geq v_i \geq q_i,\ \forall\ i \in V, k = 1 \tag{2.5}$$

$$v_i \leq \sum_{j \in V} q_j,\ \forall\ i \in F \tag{2.6}$$

The objective function (2.1) aims to minimize the total travel time of the robots. Constraints (2.2) and (2.3) ensure every vertex is visited by only one robot, and every robot will visit each vertex only once. Constraints (2.4) and (2.5) are the robot loading constraints, which ensure that the quantity of material that is brought into the node is equal to the amount leaving plus the quantity available in that node (thereby assuming all available material is picked up). Constraints (2.6) ensures all requests are fulfilled.

Extra Considerations

As previously discussed, the PDTSP formulation can be generalized to the *single vehicle pick-up and delivery problem* (SPDP). The main difference between both problems is that the former does not account for pairing relations, while the SPDP does have pairing constraints, which means every pick-up node is associated with a corresponding delivery node, i.e., when $n = \breve{n}$. A time variable, h_j, represents the beginning of service of the robot at vertex j, and we add the precedence constraints in constraint (2.7).

$$h_i \leq h_{n+i},\ \forall\ i\ \in P \tag{2.7}$$

The introduction of h_j can also help as a subtour elimination constraints if we add (2.8) when we introduce a negative request,(a negative quantity at the delivery nodes); the load conservation constraints alone will not be enough to eliminate subtours.

$$h_j \geq h_i + (s_i + t_{ij}) - \mathcal{M}\,(1 - x_{ij})\,,\ \forall\ \ (i,j) \in E \tag{2.8}$$

Another common requirement is time window constraints. These constraints can be modeled by equation (2.9), where e_i and l_i represent the earliest and latest time to start service at node i.

$$e_i \leq h_i \leq l_i,\ \forall\ i\ \in V \tag{2.9}$$

2.2.2 Proposed formulation

The PDTSP formulation is only suited for a single robot case, and for only one depot. Another limitation is the fact that the formulation assumes that if a robot visits a node, the vehicle load will be increased by the full amount of the commodity available at the node under consideration, as stated by constraint (2.4). Another assumption is that sets I (origin nodes) and F (final nodes) are disjoint. To extend the formulation to the MPDTSPS, we first modify variables x_{ij}, v_j, h_j, and parameters t_{ij} to manage more than one robot by adding a k superscript to each. x_{ij}^k is a binary decision variable that equals 1 if robot k travels from edge i to edge j, and 0 otherwise. v_{jk} is the real-valued load of robot k after visiting node j. h_{jk} is the beginning of service of robot k at vertex j. We also add two new variables to our model. z_k is a binary decision variable that equals 1 if robot k is active, and 0 otherwise. $y_{ik} \leq 1$ represents the proportion of commodity that robot k takes from node i. Finally, we replace constrains (2.2) and (2.3), with (2.15) and (2.16), while constraints (2.4), (2.5), and (2.6) are reformulated to add the proportion variable as presented in (2.17), (2.18), and (2.19). Additionally, we redefine the set of all arcs E so it consists of the union:

1. $\{(i,j,k):\ \forall\ i\ \in I,\ \ \forall\ j\ \in M, i = k\}$,

2. $\{(i,j,k): \ \forall\ i\ \in I, \ \ \forall\ j\ \in M, \ \forall\ k \in K, \ i \neq j\}$, and

3. $\{(i,j,k): \ \forall\ \ i\ \in M, \ \ \forall\ j\ \in F, \ \forall k \in K\ \}$

$$\underset{x}{\text{minimize}} \sum_{k} \sum_{(i,j):\ (i,j)\ \in\ E} t_{ij}^{k} x_{ij}^{k} \tag{2.10}$$

$$\sum_{j\ \in\ M} x_{ij}^{k} = z_k, \ \forall\ \{(i,k)\ \in I \times K : k = i\} \tag{2.11}$$

$$\sum_{i\ \in\ M} \sum_{k \in K} x_{ij}^{k} \leq 1, \ \forall\ j \in\ \ F \tag{2.12}$$

$$x_{ij}^{k} \leq z_k, \ \forall\ \ (i,j) \in V, \ \forall \in K \tag{2.13}$$

$$\sum_{i:(i,u)\ \in\ E} x_{iu}^{k} - \sum_{j:(u,j)\ \in\ E} x_{uj}^{k} = 0, \ \forall k\ \in k, \ \forall\ u \in M \tag{2.14}$$

$$\sum_{k \in K} y_{ik} = 1, \ \forall\ i \in\ \ M \tag{2.15}$$

$$\sum_{i \in (I \cup M):(i,j,k) \in E} x_{ij}^{k} \geq y_{jk}, \ \forall\ k \in\ \ K, \ \forall\ j \in\ \ M \tag{2.16}$$

$$v_{jk} \geq\ \ v_{ik} +\ \ q_j y_{jk} - \mathcal{M}\left(1 -\ \ x_{ij}^{k}\right), \forall\ \ (i,j) \in\ \ E, \ \forall\ k \in\ \ K \tag{2.17}$$

$$CP_k + q_i y_{ik} \geq v_{ik} \geq\ \ \ q_i y_{ik}, \forall\ i \in\ \ V, \quad \forall\ k \in\ \ K \tag{2.18}$$

$$v_{ik} \leq \sum_{j \in M} q_j y_{jk}, \ \forall\ i \in F, \ \forall\ k \in K \tag{2.19}$$

$$CP_k \geq v_{ik} \geq 0, \ \forall k \in K, \forall\ i \in\ \ M \tag{2.20}$$

$$y_{i,k} \leq 1, \ \forall k \in K, \forall\ i \in\ \ M \tag{2.21}$$

The objective function in (2.10) seeks to minimize the time to fulfill all the requests. Constraint (2.11) restrict the origins to be equal to the number of active robots; that means if robot k is inactive then the associated initial depot cannot have any connection to the intermediate nodes M. Constraint (2.12) ensures there is only one possible final node for every robot. Constraint (2.13) ensures only active robots are considered, and Constraint (2.14) is a flow conservation constraint. Constraints (2.15) and (2.16) restrict the number of times the same vertex is visited according to the proportion of material robots have taken from the vertex. Constraints (2.17) and (2.18) ensure the correct flow of the load,meaning that the flow of the load that enters a node i must change according to the quantity is taken i; they also work as subtour elimination constraints. Constraint (2.19) ensures all the material is collected. Finally, the value $\mathcal{M}$ is set to a big constant value; it suffices to set $\mathcal{M} \geq max\{CP_i, \ldots, CP_{|K|}\}$.

2.3 Computational experiments

MPDTSPS is an NP-hard problem. It can be reduced to 1-PDTSP when we consider only one depot, and as pointed out in [51], 1-PDTSP can be reduced to the TSP. Since the TSP is known to be NP-hard [55], MPDTSPS is also NP-hard. We now validate our formulation and analyze how it scales with the number of requests and initial depots.

Definition 2.3.1. An instance of both optimization problems (PDTSP and MPDTSPS) is defined by the tuple D_i=(G, CP, d) where G is the graph induced by the corresponding I, M and F, CP is the set of the capacities with size $card(K)$, and d is the set of requests.

The evaluation of the formulation aims to empirically test two properties of the formulation, the *correctness* of the formulation, and the *complexity* of the problem, (i.e., how it *scales* as we add more robots and more requests to the system). As discussed earlier, our formulation is a generalization of the PDTSP, which means the PDTSP is a special case of the MPDTSPS when $I = \{1\}$, $F = \{1\}$, $K = \{1\}$ and $y_{i1} = 1 \; \forall i \; \in M$. If the formulation of the MPDTSPS is in fact a *correct* generalization of the PDTSP, the MPDTSPS can be inputted with the same instance as the PDTSP and have the same optimal solution. The formulation was implemented using Python 3.6 and solved using

Gurobi version 8.0. In the implementation, there is an assumption that all robots must finish at the same final node $j \in F$. We ran the experiments on a PC core i5-6400 2.7 GHz with 8 Gb of RAM. All the Gurobi default setups with a MIP tolerance of 0.0001 are kept through all the experiments; this means that the solver will terminate when the difference between the lower and upper objective bound is less that MIP tolerance times the absolute value of the upper bound. To evaluate our formulation, we first define the following instance. $D_1 = (G_1, CP, d)$, where $|I| = 1$, $|F| = 1$, and $|M| = m$ will increase by 5, while CP, and d are set as constants. We generate the $|I| + |F| + |M|$ random points in a [-6199.5,6199.5] $\times$ [-4353,4353] space, each of which correspond to a location of a robot, requests, and final required location, respectively. We generate the random requests in the interval [-50 ,50]. We also assume that $\sum_i q_i = 0$. This last assumption means that we are dealing with balanced requests or demand, i.e., for every pick-up there is a delivery. Finally, $CP_1 = 200$. We solve both optimization problems, PDTSP and MPDTSPS, and show the results in Table 2.1. We can see that when both formulations solve the same instance of the problem, both problems find the same optimal solutions. This shows the MPDTSPS can be reduced to PDTSP as is expected, since every optimal solution for problem PDTSP is also an optimal solution for problem MPDTSPS, if they run over the same instance. We feel that this appropriately validates our model. As stated before, the MPDTSPS is NP-hard. It is natural to expect the complexity of the problem to increase in a exponential-like fashion as the number of robots and the number of request nodes increases. Nevertheless, we are interested to know how much we can increase the complexity of the problem and still be able to use conventional optimization approaches to solve the problem for real test scenarios.

To test the complexity we evaluate our formulation in the following in-

Table 2.1: Optimal solution over D_1

PDTSP			MPDTSPS		
Req	LB	time	Req	LB	time
5	28.4556	0.14	5	28.4556	0.1
10	37.0858	0.6	10	37.0858	1.61
15	34.452	1.09	15	36.452	3.03
20	43.6364	7.01	20	43.6364	4.12
25	43.7166	10	25	43.7166	21.71
30	47.4997	20.33	30	47.4997	9.45
35	49.588	2.99	35	49.588	5.28
40	58.3565	605.07	40	58.3565	1435
45	56.8225	4661.98	45	56.8225	4441.88
50	61.3442	32.63	50	61.3442	71.60

stances with a graph $G_{2,m}$ induced by the following nodes, where $|I| = 2$ and $|F| = 2$, and $|M| = m$ will increase by 5, and CP,d are set as constants. We also test the formulation in the graph $G_{3,m}$, where $|I| = 3$ and $|F| = 3$, and $G_{4,m}$, where $|I| = 4$ and $|F| = 4$. The set of random locations are generated in the same [-6199.5,6199.5]$\times$[-4353,4353] space. The random requests are also generated in the interval [-50 , 50]. We set a runtime limit of 5 hours. We summarize the test parameters in Table 2.2 and shown the results of our experiments in Table 2.3.

In Table 2.2, instances marked with "-" are instances where we could not obtain a feasible solution under the 5 hours limit we set. In every other case, we can see how the number of constraints and variables increases when we add more and more robots. It must be noted that instances up to 15 requests with 4 robots can be solved in a reasonable amount of time for real-time applications. We can also see that the more we constraint the capacity of the robots, the more difficult the problem is to solve. A further observation is that when we have a set of heterogeneous robot's capacities the optimal solution will tend to use the robots which are less constrained in terms of capacities. For example,

Table 2.2: Summary of the parameters in the tested instances

Instance	Graph	Cp	Instance	Graph	Cp
D$_2$	$G_{2,m}$	100 $\forall\ k \in\ K$	D$_5$	$G_{2,m}$	[50,100]
D$_3$	$G_{2,m}$	50 $\forall\ k \in\ K$	D$_6$	$G_{2,m}$	[100,25]
D$_4$	$G_{2,m}$	25 $\forall\ k \in\ K$	D$_7$	$G_{2,m}$	[50,25]
Instance	Graph	Cp	Instance	Graph	Cp
D$_8$	$G_{3,m}$	100, $\forall\ k \in\ K$	D$_1$1	$G_{3,m}$	[50,100,25]
D$_9$	$G_{3,m}$	50, $\forall\ k \in\ K$	D$_1$2	$G_{3,m}$	[100,25,25]
D$_1$0	$G_{3,m}$	25, $\forall\ k \in\ K$	D$_1$3	$G_{3,m}$	[50,25,50]
Instance	Graph	Cp	Instance	Graph	Cp
D$_1$4	$G_{4,m}$	100, $\forall\ k \in\ K$	D$_1$7	$G_{4,m}$	[50,100,25,25]
D$_1$5	$G_{4,m}$	50, $\forall\ k \in\ K$	D$_1$8	$G_{4,m}$	[100,25,100,100]
D$_1$6	$G_{4,m}$	25, $\forall\ k \in\ K$	D$_1$9	$G_{4,m}$	[50,25,50,100]

we can compare the instances D_2 and D_6. The optimal solutions differ just slightly, which arises is by the selection of the robot performing the tour. The robot's initial positions in both instances are the same, and the only difference is the capacity. In instance D_6 the capacity of one of the robots is further constrained and therefore the optimal solution for the instances in D_6 selects a robot which is further form the request but whit a higher capacity.

2.4 An application: Multi-Robot Task Allocation

Human-robot collaboration is important in manufacturing scenarios, [56], since the use of robots has several benefits, e.g., increase in productivity or the reduction of the lead time of the tasks to be realized, [1]. In addition, the introduction of *multi-robot systems* (MRS), is expected to have an even bigger impact on the overall efficiency of the system than single-robot systems. Several approaches have been used to tackle MRS coordination, [57], and there are a lot of new techniques that can be used to find an optimal assignment of the tasks. The most widespread problem in MRS is the task allocation problem, where a set of robots must be assigned to a set of tasks while minimizing a certain cost function; the problem of optimal placement of tasks to robots is known as *multi-robot task allocation* (MRTA), [57]. MRTA has been tackled with several approaches, including MILP, reactive methods, evolutionary computation, and marked based approaches;the latter is one of the more widely used approaches. In this paper, we propose to use the MPDTSPS formulation to tackle the material handling problem in an industrial scenario. We can identify several formulations for versions of this problem in the literature, [45]-[48]. However, a formulation that properly captures all the characteristics of our application domain is not available to our knowledge. The reason behind that is that most of the routing problems are motivated for logistics applications in carrier suppliers, or similar, while the motivation behind our work is an autonomous material handling system in which a special case of PDP arises. The scenario at hand is based on a manufacturing production cell where the robots need to perform several pick-up and delivery tasks as a material handling agent. In Figure 2.2 we show a V-REP simulation of the scenario at hand. In our scenario there are several request points were the robots have either material to pick-up or deliver, while considering capacity constraints. The total handling cost is given by the associated time it takes for the robots to travel along the locations. Our objective function minimizes the handling cost incurred by the robots. The main feature of our formulation is to account for the ability of the robots to deliver or pick up only a proportion of the demand/supply at hand, unlike similar approaches which assume that if a robot visits a location it will deliver or pick-up the whole quantity at the specified location. The assumption that the whole demand/supply must be fulfilled if a robot visits the node will make infeasible every instance of the problem in which $CP < \max_{i \in M} |q_i|$, i.e., the capacity of the robot is lower than the maximum supply/demand required at a given node. This assumption will clearly limit the applicability of several formulations which do not account for the split of the load. As an example, consider the single robot case of the 1-PDTSP formulation which cannot be used for instances in which a robot capacity is exceeded by a supply/demand or a node. In the case of multiple robots, most of the problems in the PDVRP class listed in Figure 2.1 cannot be solved if we do not allow for the split load. Therefore, a formulation able to handle a bigger set of instances of the problem, such as those with $CP < \max_{i \in M}|q_i|$,is needed for the applicability in such a robotic setting. The objective of this section is to test whether the overall system performance, defined as the total time needed for pick-up and delivery of all the objects, improves by allowing the use of multiple robots. The total cost, as defined by the objective function, might remain the same (though not always). However, since the tasks are performed in parallel, the overall time to perform the task is given by the longest path in the set of paths assigned to the robots.

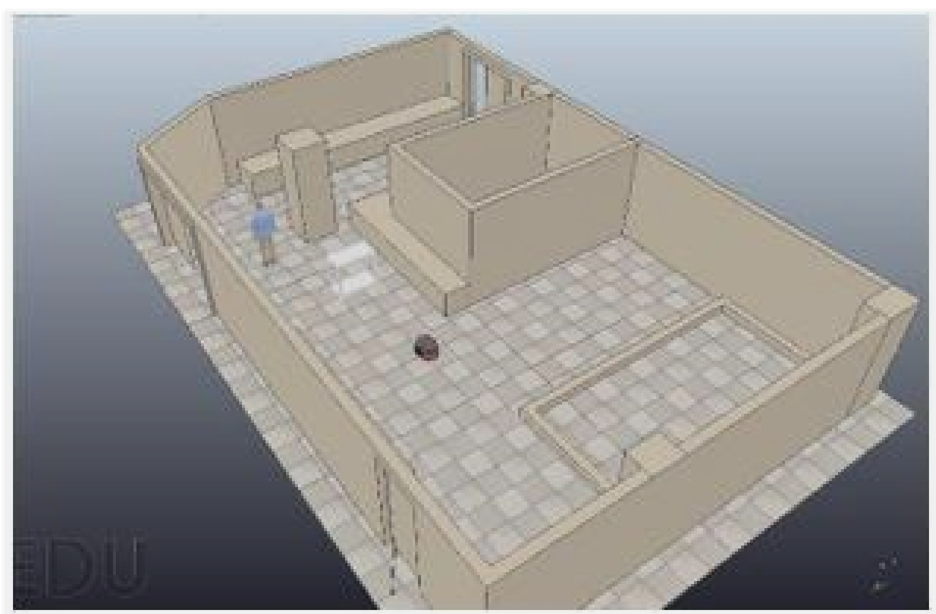

Figure 2.2: V-Rep Simulation of the Laboratory of Intelligent Manufacturing, Design and Automation, were a mobile robot is used to Material Handling Agent in a manufacturing line

The main advantages of a system that accounts for split loads and multiple robots is the system is able to solve a wider range of instances of the problem, which means an increase of the applicability of the previous formulations. Additionally, the system may accomplish missions faster than in the case of a single robot system. In Table tab:mrta we show the time it takes to perform all the pick-ups and deliveries in a single robot setting versus the multiple robot case. One can also see that there are instances that will be infeasible for single robot system without split loads.

Finally, although we can apply the MPDTSPS formulation for small instances of the problem the idea behind the formulation is to have a way to compute optimal solutions for benchmarks against faster algorithms which may have more applicability in real dynamic scenarios.

2.5 Conclusions

The main contribution of this Chapter is to present the *multiple vehicle traveling salesman problem with pick-up and deliveries and split load constraints*, and show that it can be used to improve the performance of a material handling system by considering several robots at the same time, and allowing a split load. Computational experiments show that small instances of the problem for up to 4 robots and 15 pick-up/delivery nodes can be solved to optimallity in a reasonable amount of time, and the introduction of split load capabilities will increase the performance of the system by allowing it to solve a larger set of instances. Our formulation can be categorized as PDVRP under the framework presented in Section 2.1, with the particularity that multiple depots are considered.

Table 2.3: Experimental results

(a) Experiment under instance D_2

Req	Obj	LB	GAP	Time	Vars	Constr
5	21.2989	21.2989	0	0.2763	120	275
10	32.9930	32.9930	0	0.8859	320	830
15	46.2799	46.2799	0	9.3830	620	1685
20	49.6040	49.6040	0	21.9862	1020	2840
25	56.7825	56.7825	0	131.4236	1520	4295

(b) Experiment under instance D_3

Req	Obj	LB	GAP	Time	Vars	Constr
5	26.6050	26.6050	0	0.5916	120	275
10	43.5992	43.5992	0	5.6404	320	830
15	58.9569	58.9562	1.2686E-05	3439.3322	620	1685
20	58.9647	58.9647	0	1152.3502	1020	2840
25	69.3481	62.9323	0.0925	18000.6431	1520	4295

(c) Experiment under instance D_4

Req	Obj	LB	GAP	Time	Vars	Constr
5	45.5392	45.5392	0	1.4774	120	275
10	79.5886	79.5886	0	534.5758	320	830
15	81.9277	81.9236	5.09E-05	3721.4718	620	1685
20	105.0117	89.8930	0.143971	18000.75	1020	2840
25	114.8408	86.8111	0.244074	18000.64	1520	4295

(d) Experiment under instance D_5

Req	Obj	LB	GAP	Time	Vars	Constr
5	21.2988	21.2988	0	0.2102	120	275
10	32.9930	32.9930	0	0.7987	320	830
15	47.7114	47.7114	0	10.6682	620	1685
20	51.0355	51.0355	0	32.6063	1020	2840
25	58.2140	58.2140	0	548.8485	1520	4295

(e) Experiment under instance D_6

Req	Obj	LB	GAP	Time	Vars	Constr
5	22.1996	22.1996	0	0.0810	120	275
10	34.4227	34.4227	0	1.1350	320	830
15	46.2798	46.2798	0	1.1551	620	1685
20	49.6040	49.6040	0	8.1258	1020	2840
25	56.7825	56.7825	0	182.03	1520	4295

(f) Experiment under instance D_7

Req	Obj	LB	GAP	Time	Vars	Constr
5	27.6693	27.6694	0	0.2242	120	275
10	44.4999	44.4999	0	3.0609	320	830
15	58.9569	58.9541	4.79E-05	187.3977	620	1685
20	58.9647	58.9647	0	145.1748	1020	2840
25	69.3481	69.3426	7.93E-05	5339.7757	1520	4295

(g) Experiment under instance D_8

Req	Obj	LB	GAP	Time	Vars	Constr
5	22.19962	22.19962	0	0.081078	120	275
10	34.42271	34.42271	0	1.135096	320	830
15	46.27985	46.27985	0	1.155111	620	1685
20	49.60402	49.60402	0	8.125824	1020	2840
25	56.78252	56.78252	0	182.0363	1520	4295

(h) Experiment under instance D_9

Req	Obj	LB	GAP	Time	Vars	Constr
5	27.6693972	27.6694	0	0.2242146	120	275
10	44.4999458	44.49995	0	3.0609474	320	830
15	58.9569284	58.9541	4.79E-05	187.39747	620	1685
20	58.9647159	58.96472	0	145.1748	1020	2840
25	69.3481197	69.34262	7.93E-05	5339.7757	1520	4295

(i) Experiment under instance $D_1$0

Req	Obj	LB	GAP	Time	Vars	Constr
5	62.1438	62.1438	0	1.7563	204	464
10	62.2257	62.2205	8.8E-05	970.4255	519	1339
15	71.1912	71.1892	2.7E-05	5892.477	984	2664
20	104.6525	69.6427	3.3E-01	1800.01	1599	4439
25	-	-	-	-	-	-

(j) Experiment under instance $D_1$1

Req	Obj	LB	GAP	Time	Vars	Constr
5	34.1631	34.1631	0	0.1336	204	464
10	41.7920	41.7920	0	16.9062	519	1339
15	46.1202	46.1202	0	135.1191	984	2664
20	53.9295	53.9295	0	1039.379	1599	4439
25	64.3221	59.5557	0.0741	18000	2364	6664

(k) Experiment under instance $D_1$2

Req	Obj	LB	GAP	Time	Vars	Constr
5	36.2887	36.2887	0	0.2902	204	464
10	43.2256	43.2256	0	8.6002	219	1339
15	45.2268	45.2268	0	22.0172	984	2664
20	55.3604	55.3600	6.95E-06	530.5419	1599	4439
25	67.7706	61.1148	0.0982	18000	2364	6664

(l) Experiment under instance $D_1$3

Req	Obj	LB	GAP	Time	Vars	Constr
5	44.27640	44.2764	0	0.5056	204	464
10	42.8842	42.8842	0	16.1815	519	1339
15	50.0497	50.0497	0	162.8258	984	2664
20	65.0014	54.8088	1.57E-01	18000	1599	4439
25	-	-	-	18000	2364	6664

(m) Experiment under instance $D_1$4

Req	Obj	LB	GAP	Time	Vars	Constr
5	24.3862	24.3862	0	1.0820	304	689
10	32.0625	32.0625	0	6.8976	744	1914
15	36.6206	36.6206	0	92.1157	1384	3739
20	46.2091	42.7083	0.0757	18000	2224	6164
25	53.2018	45.5546	0.1437	18000	3264	9189

(n) Experiment under instance $D_1$5

Req	Obj	LB	GAP	Time	Vars	Constr
5	24.3862	24.3862	0	0.8778	304	689
10	33.9341	33.9341	0	7.7445	744	1914
15	42.7463	42.7455	1.78E-05	1825.0204	1384	3739
20	57.6467	47.7821	0.1711	18000	2224	6164
25	74.9705	48.8648	0.3482	18000	3264	9189

(o) Experiment under instance $D_1$6

Req	Obj	LB	GAP	Time	Vars	Constr
5	28.4800	28.4800	0	0.5605	304	689
10	44.1757	44.1757	0	12.1036	744	1914
15	54.5071	54.5029	7.6E-05	6932.337	1384	3739
20	85.2401	58.7298	0.3110	18000	2224	6164
25	96.6607	62.1029	0.3575	18000	3264	9189

(p) Experiment under instance $D_1$7

Req	Obj	LB	GAP	Time	Vars	Constr
5	25.4207	25.4207	0	0.3400	304	689
10	35.2319	35.2319	0	9.6422	744	1914
15	40.2972	40.2972	0	199.3549	1384	3739
20	48.1639	48.1639	0	1148.3699	2224	6164
25						

(q) Experiment under instance $D_1$8

Req	Obj	LB	GAP	Time	Vars	Constr
5	24.3862	24.3862	0	0.2224	304	689
10	32.0625	32.0625	0	3.7175	744	1914
15	36.6206	36.6206	0	41.2497	1384	3739
20	43.8780	43.8780	0	252.4651	2224	6164
25	-	-	-	-	-	-

(r) Experiment under instance $D_1$9

Req	Obj	LB	GAP	Time	Vars	Constr
5	24.3862	24.3862	0	0.3341	304	689
10	32.7902	32.7902	0	2.9628	744	1914
15	36.6206	36.6206	0	40.1897	1384	3739
20	43.8780	43.8779	3.24E-06	110.3202	2224	6164
25	-	-	-	-	-	-

Table 2.4: Performance of the system with a single robots vs multiple robots

D_8	Obj.	$\|M\|$	D_9	Obj.	$\|M\|$
1	43.938264	10	1	50.261035	10
2	41.512528	10	2	42.953249	10
3	43.387429	10	3	47.448775	10
All	39.9574	10	All	42.8842	10
D_9	Obj.	$\|M\|$	D_13	Obj.	$\|M\|$
1	50.750229	15	1	infeasible	10
2	53.155095	15	2	infeasible	10
3	54.1684	15	3	infeasible	10
All	50.049727	15	All	42.884259	10

Chapter 3

Pick-up and delivery with linear temporal logic

Intralogistics is the integration, management, and optimization of materials and information flows within a facility, and provides opportunities for cost reduction in the manufacturing industry and the service sector. Although an increasing number of intralogistics systems, [20, 58], have become available, those systems typically require a complex multilevel control architecture (integration of ERP systems, site control software, process control software, etc.). Those complex system architectures are now reaching their limits, [2, 4], and flexible and adaptable systems are required instead. Automated guided vehicles (AGV) have been suggested to be suitable candidates for overcoming the drawbacks of such complex architectures. There are now several AGV systems available, Kiva Mobile Fulfillment System, [9, 59], now Amazon Robotics, [10], TUG robots, [16], CoBots, [60], etc., all of which have advantages and disadvantages. In addition, there are other factors to consider before we can achieve a fully autonomous system, what we can call a fully autonomous material handling agent (FAMHA), for example, increasing the motion ability of the robot such that it considers the changes on the mass of the robot when performing loading and unloading operations, [29], and eliminate the need of high-level programming skills for commanding complex task to the robots [39]. Others have proposed some desired properties for future material handling systems, [2], and a larger class of standardized robot capabilities for increasing the autonomy of robotics systems, [8]. Considering most suitable robotic systems are built from several interrelated and integrated subsystems, we narrow our scope to a set of abilities related to planning/cognition capabilities. We summarize those target abilities as follows: 1) increased navigation capability (INC), or the ability of the robot to move autonomously without the need to adopt predefined paths in its environment (INC-P), and the consideration of the effect, in motion, produced by changes in mass due to the loading and unloading of objects (INC-M), 2) task allocation (TA), or the partitioning of global tasks into a number of subtasks, 3) online planning (OP), or the ability to overcome eventualities on partially observable (OP-P), stochastic (OP-S), uncertain (OP-U), or dynamic (OP-D) environments, 4) Interplay among task planning, motion planning and task allocation (I), which refers to coupled approaches to simultaneously address the tree planning problems: simultaneous motion planning and task planning (SMPTP), simultaneous motion planning and task allocation (SMPTA), simultaneous task planning and task allocation (STPTA), Simultaneous task planning, motion planning and task allocation (STPMPTA), 5) reconfigurability (R), it should be easy for human operators to change the robot configuration as needed without significant need for manual intervention (e.g., technicians, programmers, etc.). This ability can be coupled with integrated task and motion ability, as it is highly desired for humans to easily command different robot tasks without significant changes to the configuration of the robot, 6) scalability (S), it refers to the number of elements on the instance the system can handle, i.e., number of robots, and number of requests. Finally, we add an extra consideration,7) capacity constraints (C) that stands for the consideration of the limited capacity of the robots. This is mainly motivated since there are several works that consider pickup and delivery tasks that do not account for the capacity of the robots as a constraint. We organize this Chapter in the following way. In Section 3.1, we preset a quick overview on several works that explicitly mention pickup and delivery tasks. In Section 3.2, we present a brief introduction to several notions and definitions that we use through the rest of the Chapter. In Section resent the multi-vehicle pickup and delivery problem with linear temporal logic (PD-LTL). In Section, resent an algorithm to solve the PD-LTL to optimality. In Section, 3.5, we present the results of some experiments. Finally, in Sections, 3.6 and 3.7 we present some discussion and conclusions.

3.1 Pickup and delivery problems in robotic networks

There exist an extensive amount of research around the set of abilities mentioned before and, equality, a wide amount of approaches to improve current robotic systems. In this work, we classify all approaches that consider the movement of material through a robot network as belonging to the general pickup and delivery problem (GPDP) class. GPDP problems have been widely studied in the operations research community, [45]-[48], [61], and their main objective is to optimize pickup and delivery (PD) operations to increase the efficiency of a system. GPDP problems in robotic networks have been addressed as mixed integer programming (MIP) problems, [62], [60], [63]. Further, heuristics have been proposed to address problems efficiently, [64], [60], or to handle online settings (dynamic environments), [33], [65], [7], [66]. Another class of approaches try to generalize the multi-agent path finding problem (MAPF) to handle more realistic scenarios applicable to intra-logistics operations, [67], as the package-exchange robot-routing (PERR) problem, [68]; the multi-agent pickup and delivery (MAPD) problem, [69]. More recently, the loading and unloading of bays by several robots is addressed in, [70], as a distributed constraint optimization problem, [71], which is solved using a max-sum algorithm. The use of formal methods, [72], offer the ability to handle the set of desired abilities mentioned in Section 1.2. For example, it has been shown to be an effective approach to address the SMPTP using different temporal logics (LTL, MTL, STL, and TWTL) as a specification language, [73, 74, 75, 76, 77], similarly for STPTA [75, 78]. Additionally it offers R since one can easily command complicated robot tasks without having to rely on complicated coding abilities; furthermore, INC is an immediate consequence of SMPTP. Finally, the use of such methods has been use to address uncertain environments (OP-U), [79], and large scale multi-robot systems (S), [80], in recent years. In that sense we put special attention to approaches that have used such methods to approach robotic system dealing with PD operations. The work in [81], used cs-LTL specification formulas to command a team of robots to perform pick-up and delivery tasks. Even though the author considers a multi-agent system, the author assume the robots cannot carry more than one object at the same time, do not account for automatic cs-LTL decomposition, and do not consider capacity as a limitation. Finally, the state space, as defined by the author, adds more complexity to the problem. Automatic decomposition of finite linear temporal logic (LTL) specifications into independent task specifications has been addressed in [82], where the authors give a formal definition of LTL decomposition that is achieved through what they call the "decomposition set" and "essential sequences". In a nutshell, the decomposition set is the set containing the states of an automaton representation of a formula. Each state represents a sub-formula that can be split in the sense that the satisfaction of the split formulas implies the satisfaction on the original formula. Essential sequences are just words on the language, See Section 3.2, of a formula that tell us if a state belong to the decomposition set. This work is improved further through simultaneous task allocation and planning (STAP), [78]. In this approach, the authors have used their previous ideas on decomposability, decomposition set, and essential sequences, to optimally allocate separable sub-tasks of a global mission specification, so the full mission is split among the agents, and therefore avoids computing the combinatorial number of possible states and actions of the full product automaton of the agents. This is solved using a constrained optimal multi-agent planning algorithm, [83], which offers a way to deal with discrete constraints (sequencing on the tasks) and continuous constraints (resources). None of the approaches herein up to this point considers the dynamics of the robot; to our knowledge, the only source in the literature that addresses this aspect for pickup and delivery problems is the work in [84].That method provides an approach for generating time-optimal trajectories for a robot that picks up objects and drops them off in a final location, in a two-dimensional environment, while satisfying a linear temporal specification given in cs-LTL and limited capacity of the robot. The authors also considered how the dynamics of the robot changes when every pick-up and delivery is performed. The approach discretizes the system, which is modeled as a finite weighted transition system. Then, using a product of automaton of the hybrid transition system, [85], and the cs-LTL formula, they used Dijkstra's algorithm to find the shortest path in the product of automaton, which correspond to the time optimal hybrid trajectory of the robot. Additionally, we also consider work that studies the vehicle routing problem with linear temporal logic. Although these studies do not account capacity constraints we believe there are still relevant for the purposes of our work. In [86], the authors presented the persistent vehicle routing problem and algorithmic procedure to solve it. This problem is an extension of the classical vehicle routing problem (VRP), which accounts the use of Time-window temporal logic (TWTL) to command the robots behavior. In particularly, they use TWTL formulas to command persistent surveillance missions to the robots. Another similar extensions are, [87, 88], where the authors use (a special fragment of LTL that does not use the next operator) and metric temporal logic (MTL) to extend the VRP to the formulas given in the formal languages previously mentioned.

3.2 Preliminaries

In this section we briefly preset present preliminary definitions and concepts we use through this work. We denote the set of atomic propositions $\pi \in \Pi$, i.e., a set of Boolean variables representing sentences which hold true at specific states. A word over a set (in theoretical computer science this set is referred as an alphabet) γ is a sequence denoted by $w_\gamma = w_\gamma(1), \ldots, w_\gamma(n)$ for $n \in \mathbb{Z}_+$, where $w_\gamma(i) \in 2^\gamma$. The length of the word is the number of elements in it and it is denoted by $|w_\gamma|$. The set of all finite words over an alphabet γ is denoted by γ^* while the set of infinite words is denoted by γ^ω, [89]. A language is a set of words usually denoted as $\mathcal{L}$; given an alphabet (we define our alphabet as Π) we say that $\mathcal{L}$ is a language over Π, which is denoted as $\mathcal{L}_\Pi$, if $\mathcal{L} \subseteq \Pi^\omega$, [89]. A plan $\beta = s(i), \ldots, s(m)$ is a word over a set of states. Let be $s(i) = (s_1, \ldots, s_l)$ an element of a word, and then the projection operator is defined as $Proj_n(s(i)) = s_n$. A non-deterministic finite automaton (NFA), [89], $\mathcal{A}$ is a tuple ($S_\mathcal{A}$, $S_{\mathcal{A},O}$,$\Pi_\mathcal{A}$,$\delta_\mathcal{A}$,$S_{\mathcal{A},F}$), where $S_\mathcal{A}$ is a finite set of states, $S_{\mathcal{A},O}$ is the set of initial states, $\Pi_\mathcal{A}$ is the input alphabet, $\delta_\mathcal{A} : S_\mathcal{A} \times \Pi_\mathcal{A} \to 2^{S_\mathcal{A}}$ is the transition function, and $S_{\mathcal{A},F}$ is the set of accepting states; $\mathcal{A}$ is deterministic finite automaton (FSA) if $\delta_\mathcal{A} : S_\mathcal{A} \times \Pi_\mathcal{A} \to S_\mathcal{A}$, and $S_{\mathcal{A},O}$ contains a single element. Every NFA can be translated to a FSA. A word, w_Π, describes a run of states, $s_a \in S_\mathcal{A}$, given by $\rho : \mathbb{N} \to S_\mathcal{A}$ if the run of states starts at some initial state $\rho(0) = s_a \in S_{\mathcal{A},O}$, ends in some final state, $s_a \in S_{\mathcal{A},F}$, and $\forall i \geq 0, \rho(t+1) \in \delta(p(t), w_\Pi(t))$ then we say the word, w_Π, is accepted by the automaton. A transition system (TS), [90], is defined as a tuple $\mathcal{T} = (S_{\mathcal{T}}, S_{\mathcal{T},O}, \Sigma_{\mathcal{T}}, \delta_{\mathcal{T}}, \Pi_{\mathcal{T}}, L_{\mathcal{T}})$, where $S_{\mathcal{T}}$ is a set of states, $S_{\mathcal{T},O} \subset S_{\mathcal{T}}$,is a set of initial states, $\Sigma_{\mathcal{T}}$ is a set of actions, $\delta_{\mathcal{T}} \subseteq S_{\mathcal{T}} \times \Sigma_{\mathcal{T}} \times S_{\mathcal{T}}$ is a transition relation, $\Pi_{\mathcal{T}}$ is the set of atomic propositions, and $L_{\mathcal{T}} : S_{\mathcal{T}} \to 2^{\Pi_{\mathcal{T}}}$ is a labeling function. A product automaton (PA), [89], $\mathcal{P} = \mathcal{T} \otimes \mathcal{A}$ is the product of a transition system and a non-deterministic finite automaton. It is defined as the following tuple $(S_{\mathcal{P}}, S_{\mathcal{P},O}, \Sigma_{\mathcal{P}}, \delta_{\mathcal{P}}, L_{\mathcal{P}})$, where $S_{\mathcal{P}} = S_{\mathcal{T}} \times \mathcal{A}$ is a set of states, $S_{\mathcal{P},O} = \{(s_t, s_a) : s_t \in S_{\mathcal{T},O} \wedge \exists s_b \in S_{\mathcal{A}}[(s_a, L_{\mathcal{T}}(s_t), s_b) \in \delta_{\mathcal{A}}]\}$, is a set of initial states, $\Sigma_{\mathcal{P}} = \Sigma_{\mathcal{T}}$ is the set of actions, $\delta_{\mathcal{P}} \subseteq S_{\mathcal{P}} \times \Sigma_P$, $\Pi_{\mathcal{P}} = S_{\mathcal{A}}$ is the set of atomic propositions, and $L_{\mathcal{P}} : S_{\mathcal{T}} \times S_{\mathcal{A}} \to 2^{S_{\mathcal{A}}}$ the label function that is given by $L_{\mathcal{P}}((s_t, s_a)) = s_a$. Linear temporal logic (LTL), [91], is a type of formal logic that extends propositional logic by the addition of temporal modal operators. In robotics, LTL has been shown to be effective in specifying complex desired behaviors for a given robot, [39]. The syntax of LTL shows us the rules for constructing LTL formulas from a set of atomic propositions, while using the operators $\neg$, and $\wedge$ which represent the operators "not" and "and", together with the temporal operators $\mathcal{X}$ and $\mathcal{U}$ that represent the temporal relations "next" and "until", respectively, [90]. Additional operators can be derived from the originals. Given two formulas ϕ_1 and ϕ_2. The "or" operator $\vee$ is defined as $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$, the "eventually" operator $\mathcal{F}$ is defined as $\top\,\mathcal{U}\phi$, and the "always" operator $\mathcal{G}$ is defined as $\mathcal{G} := \neg\mathcal{F} \neg\, \phi$. The semantics of LTL defines the satisfaction of a formula over a sequence of observations, i.e., elements on a word. This is denoted by $w_\Pi \models \phi$. Given a word w_Π, where $w_\Pi(i) \subseteq \Pi$ the semantics of ϕ is given by 1) $w_\Pi(t) \models \top$, 2) $w_\Pi(t) \models \pi \iff \pi \in w_\Pi(t)$, an observation satisfies an atomic proposition if the atomic proposition belongs to the observation, 3) $w_\Pi(t) \models \neg\phi \iff w_\Pi(t) \not\models \phi$, an observation satisfies $\not\phi$ if the observation does not satisfies the formula ϕ, 4) $w_\Pi(t) \models \phi_1 \wedge \phi_2 \iff w_\Pi(t) \models \phi_1$ and $w_\Pi(t) \models \phi_2$, 5) $w_\Pi(t) \models \mathcal{X}\,\phi_1 \iff w_\Pi(t+1) \models \phi_1$, finally we say 6) $w_\Pi\,(t) \models \phi_1\,\mathcal{U}\,\phi_2 \iff \exists t_j \geq t$ such that

$w_\Pi\left(t_j\right) \models \phi_2$ and $\forall\, t_l \in [t, t_j)\,,\; w_\Pi\left(t_l\right) \models \phi_1$. A special fragment of LTL is the cs-LTL. It has the same semantics as LTL with the only difference cs-LTL does not allow the negation operator to appear in front of formulas, which means the syntax of cs-LTL do not allow the expression $\mathcal{G}$ to be defined and since cs-LTL formulas only contain the $\mathcal{X}$, U, and F temporal operators, [92, 72]. A formula ϕ in cs-LTL over an alphabet Π can be always translated into an FSA with input alphabet Π that only accepts the good prefixes of ϕ, [72, 93, 94]. It has been shown that one can find a set of actions that drive a TS from an initial state to a final state while the trajectory described satisfy the cs-LTL formula by finding a path at the corresponding PA [72]. It is the case we will command the mission of the robots using formulas in cs-LTL. A mission M, [82], is a formula given by a cs-LTL formula ϕ, where $\mathcal{T}_1$ are independent tasks that can be given by their respective formulas, $\phi^{(i)}$. The set $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ is called a decomposition of ϕ. We refer the readers to [82] for further details about mission decomposition, and its properties. Notice the mission, M, is an cs-LTL formula, therefore, there is an automaton denoted by $\mathcal{A}_M$, associated with the mission and, [82], the Boolean conjunction of the tasks gives the complete specification of the mission, i.e, $M = \mathcal{T}_1 \wedge, \ldots, \wedge \mathcal{T}_n$. The decomposition set, $\mathcal{D} \subseteq S_{\mathcal{A}_M}$, of $\mathcal{A}_M$ contains all of the states that can be associated with completing of a set of sub-tasks, $\mathcal{T}_1, \ldots, \mathcal{T}_n$. The final states and initial states of $\mathcal{A}_M$ always belong to $\mathcal{D}$, [82]. Given an automaton $\mathcal{A}_M$, a state $s_a \in S_{\mathcal{A}}$ has been proved to belong to the decomposition set if and only if there is a word $w_\Pi = w_\Pi^I w_\Pi^F$, where w_Π^I describes a run from an initial state to s_a and w_Π^F describes a run from s_a to a final state such that $\hat{w}_\Pi = w_\Pi^F w_\Pi^I$ describes an accepting word in $\mathcal{A}_M$. See Theorem 2 (Descomposability), [82].

3.3 Problem formulation

Remark. In this Section, either we assume the existence of a discretization or partition of the continuous phase space of the robot, see, [39], such that we can easily compute the minimum time the robot will take to move from each of the locations in or we do not account the motion of the robots and we assume an obstacle free environment. As such, the problem can be reduced to finding a plan (tour) among the locations and initial robot's position such that the mission is accomplished.

We can think about our problem as a classical pickup and delivery problem, which are mostly formulated as Mixed-integer linear programs. Each desired behavior (DARP, PDTSP, etc) is stated as a set of constraints. In this paper we proposed the use of formal methods to formulate problems under this class; therefore, we can formulate certain problems under the proposed framework such that the desired behavior is given instead as a formula in cs-LTL. The problem can be stated as follows finding a tour among the set of locations such that the sequence of states describes a word that satisfies the mission such that the cost function is minimized. To the best of the knowledge of the authors there is not an approach which considers the use of cs-LTL for commanding a set of multiple robots with restricted capacity for pickup and delivery tasks. Therefore we present the multi-vehicle pickup and delivery problem with linear temporal logic (PDP-LTL) as following:

Problem 3.3.1. Let be R a set of robots, a set O of objects with related positions y_o and an available quantity m_o at such location for all $o \in O$, a set D of delivery locations with their related positions y_d and request m_d for all $d \in D$. Let be Y the union of objects and delivery locations denoted by Y_M as well as the initial position of the robots denoted by Y_R. Let M be the mission commanded to the system. The problem is to find a set of plans (tours) β_r for

$r \in R$ such that it describes a sequence of actions $U(\beta_r) = \sigma_1, \ldots, \sigma_n$ such that the mission M is achieved (robots pickup and deliver the objects following the desired behavior stated by the mission), while the maximum capacity C^r_{max} for $r \in R$ is respected and the cost function $J^*_G = \sum_{i=1}^{|R|} J_r$ is minimized, where, $J_r = \sum_{i=1}^{|U(\beta_r)} W_r(\sigma_r)$ and W_r represents the cost of taking the action in question.

3.4 Approach

Assuming the above elaborations we proceed to solve Problem 3.3.1 by using the method presented in Algorithm 3. This procedure will take as input the mission specification (at this point we assume the mission is given in cs-LTL), $\mathcal{M}$, the set of locations Y with their respective quantities $m_i, \forall i \in Y \setminus Y_R$, and the maximum capacity C^r_{max} for each of the robots. It then will return the individual sequence of actions that minimize the total time robots will take to complete the mission. We start by defining the a robot model as a weighted transition system following similar ideas as in [84].

Definition 3.4.1. Robot model A robot model $\mathcal{R}^r$ is defined as the following tuple, $(S_r, s_{r,o}, \sigma_r, \Pi_r, L_r, W_r)$, where:

1. $S_r \subseteq Y^r \times Q^r_O$, is a set of states the robot can take.
2. $s_{r,o}$, is an initial state.
3. $\Sigma_r \subseteq S_r \times S_r$, is a set of actions.
4. Π_r, is a set of atomic propositions.
5. $L_r : \Sigma_r \to 2^{\Pi_r}$, is the labeling function.
6. $W_r : \Sigma_r \to \mathbb{R}_{\geq 0}$, is weight function.

Let be q_o^r the initial robot's mass, when it is not carrying any object. Then Q_O^r is a discrete set of possible masses given by $Q_O^r = \{q \in \mathbb{R}_{[q_o^r, C_{max}^r]} : q = \Sigma_{i \in \delta} m_i, \delta \in 2^{O \cup D}\}$. Defining Q_O^r on such a way ensures the robot will never exceed its maximum capacity C_{max}^r. It also assumes the robot will collect or deliver the full amount m_i once it visits the respective location as it has been the standard in the GPD literature. $s_{r,o} = (y_o^r, q_o^r)$, where y_o^r represents the initial position of the robot. The set of actions is defined as $\Sigma_r = \{(y_i, q_i, y_j, q_j) \in S_r \times S_r : (y_i, y_j) \in \epsilon_r, q_j - q_i = m_{y_i}\}$, where ϵ_r represents a set of possible motions among the locations in Y and it is defined as $\epsilon_r = \{(y_o^r, y_j) : y_j \in Y\} \cup \{(y_i, y_j) : y_i \in Y, y_j \in Y\}$. The alphabet (set of atomic propositions) is defined as the set of statements like "pickup quantity located in y_o" location for all $o \in O$, and "deliver quantity located in y_d" for all $d \in D$. The labeling function L_r will associate an atomic proposition $\pi \in \Pi$ with a state $(y_j, q_j) \in S_r \iff \exists q_i \in Pred((y_j, q_j))$ such that $q_j - q_i = m_j$ where $Pred$ represents the predecessors of (y_j, q_j). Recently, some authors defined what they called a team model, which main objective is to emulate coordination decisions among robots. To create such connections the authors use what they called switch transitions, [78]. Similar notions are used in the procedure presented by Algorithm 3. We define a similar team model structure and the set of switch transitions in which we modify the defining properties on the team model initial states to eliminate the ordering over the set of robots imposed by the original definitions. Additionally, we eliminate the ordering condition imposed in the original definition on the set of switch transitions. In Figure 3.1, one can see how this changes will affect the structure of the team model.

Definition 3.4.2. Team model A team model G is defined as the following tuple $G = (S_G, S_{0,G}, \Sigma_G, S_{F,G})$

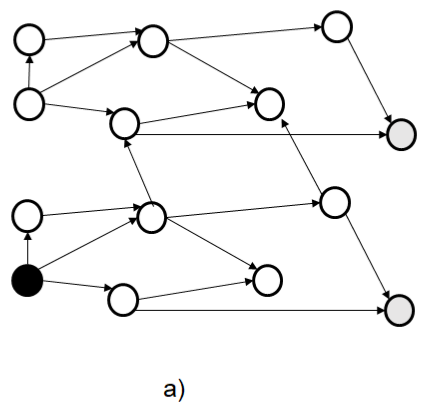

Figure 3.1: Difference between the team model structures defined in [78] figure a) and the team model structure given by Definition 3.4.1, figure b). Black dots represent the initial states of the team model. The switch transitions are now directed toward both sides in b) instead of going only towards one direction as in a).

1. $S_{\mathcal{G}} = \{(r, s_a, s_r) : r \in\ R, (s_a, s_r) \in\ S_p^r\}$ is the set of states.

2. $S_{0,\mathcal{G}} = \{(r, s_q, s_r) \in\ S_G : (s_a, s_r) \in\ S_{0,p}^r\}$ are the initial states.

3. $\Sigma_{\mathcal{G}} = \bigcup_{r=1}^{|R|} \Sigma_p \cup \xi$, is the set of actions.

4. $S_{G,F} = \{(r, s_a, s_r) : s_a \in S_{\mathcal{A},o}\}$, is the set of final states.

Algorithm 3 will initialize a team model with an empty set of states, and an empty set of actions. Initially the cost of the team, J_G^*, is set to infinity. After initialization, the first step is to compute the automaton representation of the mission $\mathcal{A}_M$. This can be done using standard algorithms for such task. See, [94, 95], for further details. The automaton representation of the formula is then used by Algorithm 1, LTLDescomposition procedure, to find the decomposition set. This procedure is based entirely in the work and description presented in [82], where they mention the use of forward search to find essential sequences and later check for states in the automaton $\mathcal{A}_M$ that belong to $\mathcal{D}$.

Definition 3.4.3. Switch transitions An element $((r^i, s_a^i, s_r^i), (r^j, s_a^j, s_r^j)) \in S_G$ belongs to the set of switch transitions $\xi \iff$:

1. $r^i \neq r^j$, it connects different robots.
2. $s_a^i = s_a^j$, it preserves the NFA progress.
3. $s_r^j = s_{r,o}$, it points to an initial robot state.
4. $s_a^i \in \mathcal{D}$, it represents a decomposition choice.

The LTLDecomposition procedure starts by adding the states in the set of final and initial states of the automaton $\mathcal{A}_M$ directly to the decomposition; remember from Section 3.2, such states always belong to $\mathcal{D}$. Afterwards, for every element s_a in $S_{\mathcal{A}_M} \setminus S_{\mathcal{A},O} \cup S_{\mathcal{A},F}$ the ConstructSequence procedure will check if it is possible to construct an essential sequence for such state, if there exist an essential sequence, Section 3.2, then the state is added to$\mathcal{D}$. We do not present a detailed description of ConstructSequence since this follows a simple forward search procedure as mentioned earlier. Next, for every $r \in R$ we compute the product automaton $\mathcal{P}^r$, Section 3.2, between $\mathcal{A}_M$ and the robot model $\mathcal{R}^r$, and we add the respective edges and nodes to the team model G to construct it progressively. We add the switch transitions following Definition 3.4.3, and a target node $s_{G,T}$ and source $s_{G,S}$ with the respective source and target connections defined as follows.

Figure 3.2 shows a graphical representation of the construction of the team model. One can imagine the team model as a disjoint union of graphs (robot models) just linked by a set of switch transitions, which model transitions among independent subtasks in the mission. Finally, we have the problem to find the shortest path connecting the single source or initial node with the target node in the graph-like structure of the team model . This is done by the ShortestPath procedure, which will find the shortest path connecting

Algorithm 1: LTLDecomposition

Input : $\mathcal{A}_M$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow \{\}$;
for $s_a \in S_{\mathcal{A}_M} \setminus S_{\mathcal{A},O} \cup S_{\mathcal{A},F}$ **do**
  if `ConstructSequence(`s_a`)` **then**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_a\}$
  end
  ;
end

the initial node $s_{G,T}$ to a final node $s_{G,S}$. Once more, a detailed description of such procedure is avoided since any path search algorithm may be used; in particular we use Dijkstra's algorithm, [96]. Finally, the ProjectRun procedure will project the final sequence of actions $U(\beta_G^*)$ on the team model into the individual robot action sequences $\{U(\beta_1^*), \ldots, U(\beta_{|R|}^*)\}$.

Algorithm 2: ProjectRun

Input : $U(\beta_G^*)$
Output: $\{U(\beta_1^*), \ldots, U(\beta_{|R|}^*)\}$
$\{U(\beta_1^*), \ldots, U(\beta_{|R|}^*)\}$;
for $t \in |U(\beta_G^*)|$ **do**
  $u_r \leftarrow Proj_R(U(\beta_G^*)(t))$;
  $U(\beta_r^*)||u_r$
end
;

This procedure use the $Proj_R$ operator to project elements over the sequence of action on the team model, $U(\beta_G^*)$, to a robot's action u_r that it is concatenated to the initially empty robots' actions $U(\beta_r^*)$. The complete procedure is finally presented in Algorithm 3

3.4.1 Desired Behaviors

As we mention before the idea of using formal methods for addressing PD operations on robotic networks is to facilitate the incorporation of more complicated behaviors in a more natural language. For a PDTSP-like behavior (see

Algorithm 3: Solution for problem 1.3.1

Input : $R = \{r_1, .., r_{N_R}\}$ robots, a set $O = \{o_1, \ldots, o_{N_O}\}$ of objects with related positions y_o and quantity m_o $\forall o \in O$, and a set $D = \{d_1, \ldots, d_{N_D}\}$ of delivery locations with their related positions y_d and request m_d for all $d \in D$;and a formula $\mathcal{M}$ given in $cs-LTL$ representing the mission

Output: $\{\mathrm{U}(\beta_1)^*, \ldots, U(\beta_{|R|})^*\}$

$S_G = \{\}, \Sigma_G = \{\}, J_G^* = \infty$;

$\mathcal{A}_{\mathcal{M}} \leftarrow \mathcal{M}$;

$D \leftarrow$ `LtlDescomposition(`$\mathcal{A}_{\mathcal{M}}$`)` ;

for $r \in R$ **do**

  $\mathcal{P}^r \leftarrow \mathfrak{R}_r \otimes \mathcal{A}_{\mathcal{M}}$;

  $S_G \leftarrow S_G \cup S^r_{\mathcal{P}}$;

  $\Sigma_G \leftarrow \Sigma_G \cup \Sigma^r_{\mathcal{P}}$

end

$\Sigma_G \leftarrow \Sigma_G \cup \xi \cup \tau$;

$\beta^*_G \leftarrow$ `ShortestPath(`$S_G, s_{G,T}, s_{G,S}$`)`;

$\{U(\beta_1)^*, \ldots, U(\beta_{|R|})^*\} \leftarrow$ `ProjectRun(`β^*_G`)`;

, [45]- [48], [61] for a detailed classification of pickup and delivery problems) the formula dictating such behavior is given by:

$$\phi_{pdtsp} = \bigwedge_{(i \in \ \delta \subseteq \ D \cup O} \mathcal{F}\pi_i \tag{3.1}$$

Which in plain English language can be read as "eventually deliver/pickup quantity at location π_i". Another interesting behavior that is widely studied in robotics is DARP in which we have paired pickup and delivery request. This means that every time there is a pickup the next action to take is to deliver such item before picking up another one, such behavior is given by:

$$\phi_{darp} = \bigwedge_{(i,j) \in \ \delta \subseteq \ D \times O} \mathcal{F}(\pi_i \wedge \mathscr{X}\pi_j) \tag{3.2}$$

This can be read as "eventually pickup quantity at location π_i and immediately next deliver at at location π_j". If one desires to specify the order in which the request are severed; such behavior can be stated as

$$\phi_{ord} = \mathcal{F}(\pi_i \wedge (\mathcal{F}(\pi_{i+1} \wedge (\mathcal{F}(\pi_{i+2} \dots \qquad (3.3)$$

That means "pickup/delive" then "pickup/deliver" and so on.

3.4.2 Complexity

In this section we will review the complexity of the construction of the team model.

Proposition 3.4.1. *An upper bound in the cardinality of the state space of the team model defined by 3.4.1 is given by* $|S_{\mathcal{G}}| \leq |\mathcal{Y}||2^{\phi}| \sum_{n=1}^{|R|} \frac{C^r_{max}}{\Delta_{min}}$

Proof. As D and O are finite we know that $|Q^r_{O \cup D}|$ is finite as well, therefore, $\exists n \in \mathbb{Z}$ such that $|Q^r_{O \cup D}| \sim \mathbb{Z}_n$. Let us denote by $\Delta_{i,i+1}, \dots, \Delta_{i+(n-2),i+(n-1)}$ the distance among the n elements lying in $[q_o, C^r_{max}]$, and $\Delta_{min} =$ $\min\limits_{\Delta} \Delta_{i,i+1}, \dots, \Delta_{i+(n-2),i+(n-1)}$ Then, given the interval $[q_o, C^r_{max}]$ and the smallest of the distances Δ_{min} the set $|Q^r_{O \cup D}|$ cannot contain more than $\frac{C^r_{max}}{\Delta_{min}}$ elements. □

3.4.3 Optimality

The procedure presented in Algorithm 3 returns an optimal plan in the sense that for any action sequence $\{U(\beta_1), \dots, U(\beta_{|R|})\}$ the associated cost J_G is equal or less to the cost of the action sequence $\{U(\beta_1)^*, \dots, U(\beta_{|R|})^*\}$ denoted by j^*_G. This follows directly from the ShortestPath procedure that use Dijkstra's algorithm to find the shortest path in the team model structure G.

3.5 Experiments

The motivation behind this work is the use of mobile robots to fulfill the request of material handling on a manufacturing shop floor. In such a scenario, not only do different behaviors arise (e.g., part feeding, warehouse commissioning, order picking, point-of-use delivery, etc.), but in addition, human-agent integration is critical. Ideally, robot tasks should be expressed in a more natural language, such as co-safe linear temporal logic (cs-LTL) [92, 97].The use of formal languages for programming robot tasks should therefore correspond to specific classes of the general pickup and delivery problem (GPDP). In this section we will show how to use Algorithm 3 to synthesize a plan for different desired behaviors. We implemented the Algorithm 3 in python 3.6.9. For the translation of the formulas into an automaton representation we use Spot, [98]. As the formula belong to the sc-LTL fragment this can be translated into a NFA[1].

Let us consider the following study case where we have three robots located in $y_1 = [6.4, 7.3]$, $y_2 = [7.0, 6.9]$, and $y_3 = [6.2, 0.49]$. Each robot has a maximum capacity of $C^1_{max} = C^2_{max} = C^3_{max} = 10$. Let us assume we have a set of three objects located at, $y_{o_4} = [5.7, 5.7]$, $y_{o_5} = [7.1, 5.1]$, and $y_{o_8} = [2.7, 5.0]$ with quantities to pick-up, $m_{o_4} = 1$, $m_{o_5} = 1$, and $m_{o_8} = 8$.There is a set of dropping locations located at $y_{d_6} = [3.8, 8.4]$, $y_{d_7} = [5.0, 3.9]$. The quantities to drop at these points are $m_{d_6} = -5$, and $m_{d_7} = -5$. See Figure 3.4. We could use formula 3.1 to command a PDTSP-like behavior,in which case we have the following mission:

$$\mathcal{M}_{pdtsp} = \mathcal{F}\pi_{o_3} \wedge \mathcal{F}\pi_{o_4} \wedge \mathcal{F}\pi_{o_8} \wedge \mathcal{F}\pi_{d_6} \wedge \mathcal{F}\pi_{d_7} \tag{3.4}$$

The decomposition set includes all the states in the NFA representation of formula 3.4, which means every sub-task of formula 3.4 can be executed independently, for example, we can split the formula into independent pickup and delivery task so each robot can execute them in parallel. The plan, β^*_G returned by Algorithm 3 is $s_{G,S}, s^2_2, s^2_5, s^2_4, s^2_8, s^2_7, s^2_6, s_{G,T}$, which can be interpreted

[1] The implementation of the algorithm is available at https://github.com/juantztz/pd-ltl

using the Algorithm 2, as robot 2 will perform all the pickups and deliveries following the tour presented before. See Figure 3.4. Assume we have the same drop and pickup locations but with a different quantities. If one desires to command a DARP-like behavior(i.e., add precedence specifications) we could use the following formula, which specifies robot should only deliver at location d_7 after delivering at location d_6 is given by the following formula:

$$\mathcal{M}_{darp} = \mathcal{F}\pi_{o_3} \wedge \mathcal{F}\pi_{o_4} \wedge \mathcal{F}\pi_{o_8} \wedge \mathcal{F}(\pi_{d_6} \wedge \mathcal{X}\pi_{d_7}) \tag{3.5}$$

The plan, β^*_G returned by Algorithm 3 is $s_{G,S}, s^2_2, s^2_5, s^2_4, s^2_8, s^2_6, s^2_7, s_{G,T}$, which can be interpreted using the Algorithm 2, as robot 2 will perform all the pickups and deliveries following the tour presented before. This tour fulfill the condition imposed by formula 3.5. See Figure 3.5.

3.6 Discussion

Resource constraints were studied in the context of formal methods (cs-LTL was used as a specification language), [78], we could use such constraints to model the capacity of the robots, and study PD operations. Instead we consider the mass of the robots as part of its state, and model the capacity constraints as a bound in the state of masses for three reasons: 1) it is important to consider the effect that changing masses (when loading and unloading of objects) may have in the motion of the robots (how it will affect its dynamics) as stated in Section 3, 2) lower upper bounds in the cardinality of the state space of the team model structure by considering the mass of the robots as part of the state 3) easier incorporation of the motion equations of the robot.In Section 3.4.1 we present a set of formulas for tasks as expect for some problems in GPDP; it should be noticed that one can use different specification languages, for example time window temporal logic (TWTL), to command tasks that include time windows.

3.7 Conclusions

We generalize previous work on pick-and-delivery problems using cs-LTL to handle more than one depot and multiple robots using recent developments in decomposition of cs-LTL formulas to present the pick multi-vehicle pickup and delivery problem with linear temporal logic, and proposed an algorithmic procedure to solve it to optimality. Future work includes studying a different set of real time temporal logic as a possibility to handle split load constraints. Although we do not consider robot's dynamics in this work, in principle, such consideration can be included in Algorithm 3 if we use include a discretization procedure. Some discretization algorithms for the continuous state space of the robots are discussed in [39].

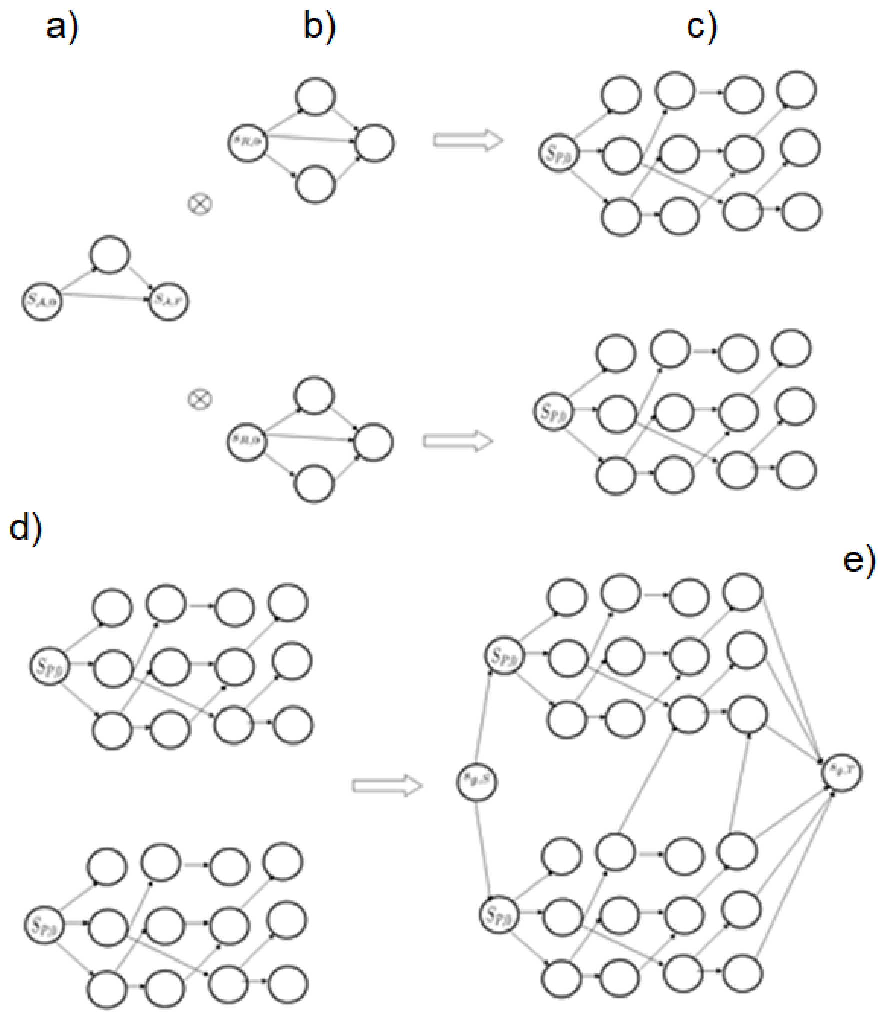

Figure 3.2: The progressive construction of the team model a) It will start with the construction of the automata, b) we construct the robot models Definition 3.4.1, and c) equal number of PA are constructed, e) Finally we add the corresponding switch transitions and source and target nodes.

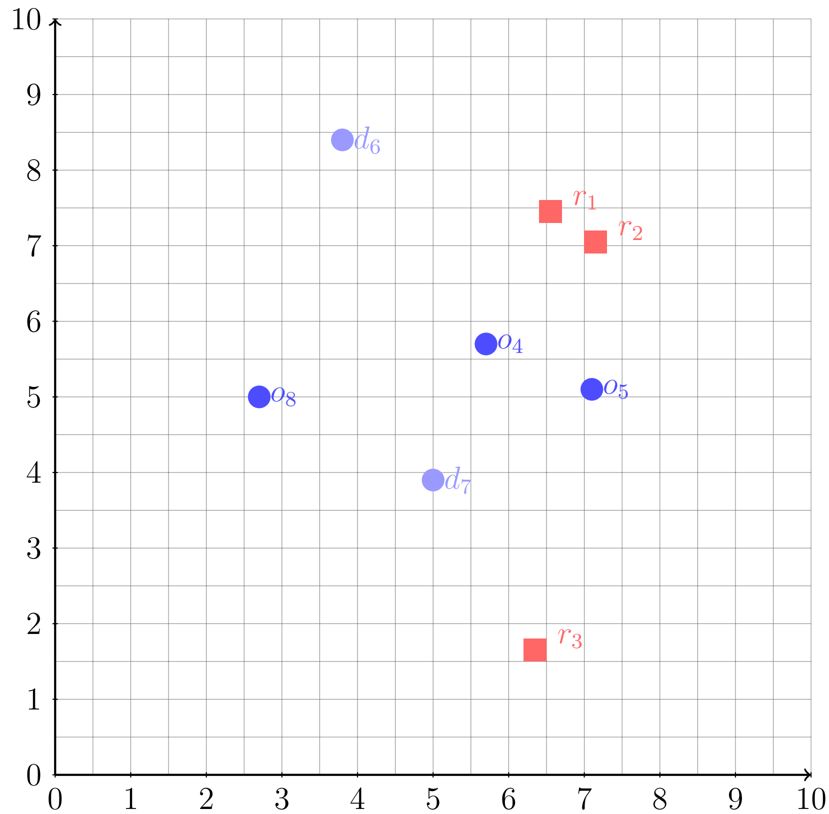

Figure 3.3: Visual representation of the experimental setup. Dark blue dots represent pickup locations, light blue dots represent dropping locations, and red squares represent the initial locations of the robot

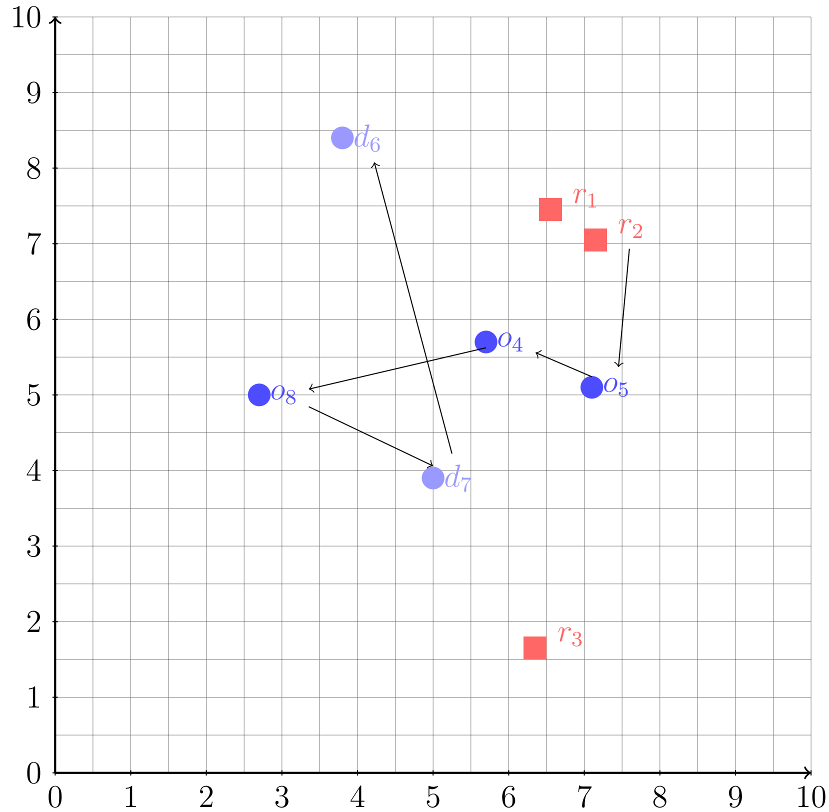

Figure 3.4: Solution found for a PDTSP-like behaviour 3.4

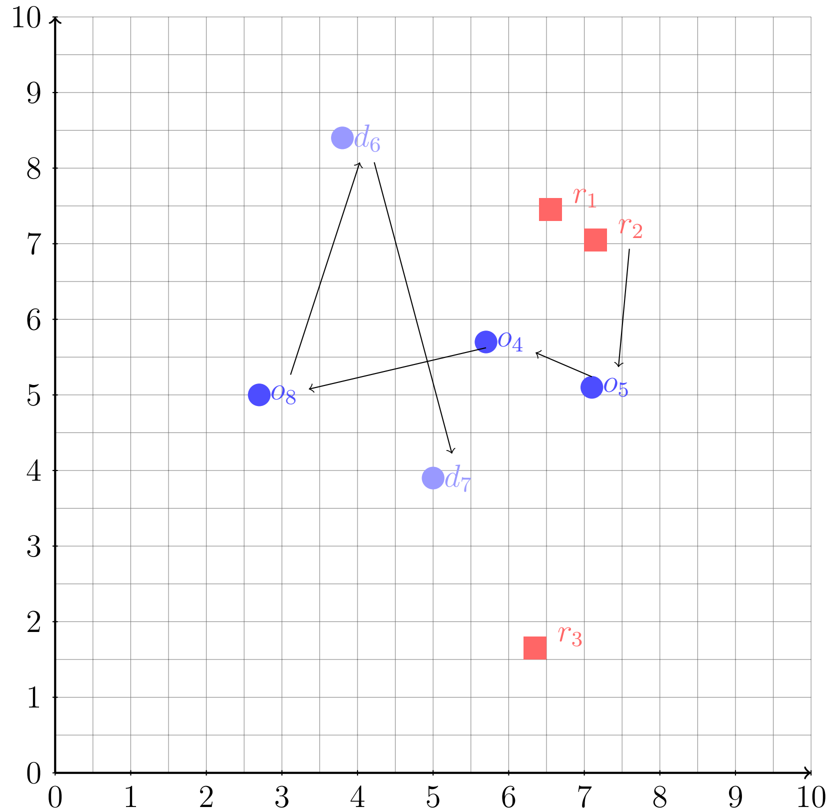

Figure 3.5: Solution found for a DARP-like behaviour 3.5

Chapter 4

Importance of autonomous robots for intra-logistics operations in the oil and gas industry.

The objective of this section is to shown the importance and the impact of autonomous agents could have in the Oil and Gas industry. Robotics and automation of several operations or activities that used to rely on humans is now present in many areas of our lives. From cleaning vacuums to agriculture applications, industrial robotics is an emerging field full of successful applications. While robotic applications are now more a norm than an option in many industries, it is not the case in the oil and gas industry, which still relies in manual or semi-autonomous operations (e.g.,remotely controlled oil and gas facilities). This is likely the case mostly because the industry is considered to be a high risk industry and therefore relying in complete autonomous agents needs a high degree of reliability. Although completely autonomous agents are risky given the sensitive materials used in the oil and gas industry, the lack of autonomy, robustness, and human dependent agents are also an area of opportunity, [99], for the following reasons: 1) robots are less likely to make errors and more reliable than humans in extreme conditions that are found in the oil and gas industry, and 2) they can work 24 hours per day and perform repetitive task for longer periods of time. However, this potential is barely exploited given economical, organizational, and social barriers, [100]. In the face of an increasing demand of fossils fuels, which is targeting the use of non conventional sources of oil and gas, the use of robots in the oil and gas industry is expected to have significant impact over the efficiency and production of the oil and gas facilities. Nevertheless, reducing the number of staff members in facilities, improving health, safety and environment (HSE) standards is the main force moving the incremental use of robots in the oil and gas industry. For example, the running cost of an offshore platform per day is around $\$35,000$ USD to $\$150,000$ USD, [99]-[101], and the most commonly scheduled operations in offshore platforms are inspection, monitoring, or maintenance, [28]; these are activities that can be easily allocated to robotic agents. Furthermore, inspection, maintenance, site survey, drilling, production, and repair, transportation, and logistics are important areas for introducing robotic applications, [25, 26, 27]. In Section 4.1 we detail various intra-logistics activities autonomous robots can perform in the oil and gas industry. In Section 4.2 we review the application of mobile robots, mobile manipulators, and manipulators that address the activities listed in Section 4.1 to conclude that the same areas of opportunity, as stated in Section 1.4, are relevant for robotic system in the oil and gas sector. Finally, in Section 4.3 we show how the developments of Chapter 3 can be applied to address operations as presented in Section 4.1.

4.1 Preliminaries

Before mentioning specific robot applications in the oil and gas sector we briefly review the sector itself and summarize important considerations we should keep in mind when developing such robotic systems. For a more extensive analysis of the current uses, improvements, and future development or applications of robots in the oil and gas industry, we need to explore and detail the current activities, staff, and operations that the oil and gas industry utilizes. The oil and gas industry can be broadly categorized into offshore and onshore facilities, though this categorization seems too general for the purpose of this review; we are most interested in a possible categorization that provides more information about the type of activities and processes inside the facilities. The work in [99] proposes a categorization better suited for this purpose, where mentions the oil and gas industry can be divided into *upstream*, *midstream*, and *downstream* sectors. In general terms, *upstream* activities are the activities related to the direct extraction and procurement of the gas and oil. *Midstream* activities are activities such as transportation, processing, and storage of the resource, and *downstream* activities are related to the use and transformation of the oil and gas to obtain more usable products. The following list shows examples of various activities and their classifications.

1. Upstream: exploration, recovery, and production of crude oil from reservoir, seismic analysis, exploratory drilling.

2. Midstream: pipelines, oil tankers.

3. Downstream: refining of crude oil, distillation, and reforming to generate plastics, fertilizer, and other chemicals.

As we mentioned before, most of the applications of robots are in the upstream and downstream activities, and most of the commonly scheduled operations a mobile robot can do are:

1. Inspection: gauge readings, and valve and lever positions.

2. Monitoring: monitor gas levels, check for leaks, and acoustic anomalies, monitor surface conditions, and check for intruders.

3. Maintenance: gas and air detector test, gas sampling, pigging, cleaning, refilling.

It has been shown that operational staff in the oil and gas industry spends a considerable amount of time walking and transporting materials, [28], therefore we can add transportation and material handling to the list, since mobile robots performing material handling can decrease the time used in such activities. In that sense, we can see that activities like inspection, site surveying, monitoring, and transportation enter into the set of operations under the definition of intra-logistics from Section 5.3.

4.2 Robotics in hydrocarbons industry

In this section we summarize, commercially available options and relevant research developments of mobile robots, manipulator robots, and mobile manipulators that can be applied to several activities in the upstream and downstream oil and gas sector in the category of intra-logistics operations. New challenges in oil and gas require a high level of automation, [102], which means the oil and gas industry can start using solutions than have been already implemented in other industries and adapt them to the conditions of the oil and gas sector. In addition, they can develop new technologies to fulfill the specific needs of the oil and gas industry, including intelligent drilling systems, smart inspection, and manipulation, [99, 101]. Most of the applications focus in the upstream and downstream processes, [101], however, the use of robots in the oil and gas industry is still immature, and the commercial options are limited, [100]. In this paper we restrict our review to the use of mobile robots, [26], manipulation robots, [22], and mobile manipulators, [99]. We also focus our attention on the application of robots for intra-logistics operations as mentioned in the previous section.

4.2.1 Current research developments and commercial options

The first application of mobile robots in offshore facilities for tasks such as maintenance and inspection was done by Fraunhofer IPA labs, [25]. This robot, MIMROex, can be controlled by humans, or act autonomously. It consist of a mobile platform equipped with a robot arm. The robot is able to plan collision free trajectories inside facilities, build maps, and execute preprogrammed tasks. Sensabot, [103], is a mobile robot developed by Carnegie Mellon University in 2011. It safely monitors hazardous and remote facilities. They claim the robot can perform the same tasks as a human operator does, without exposing humans to extreme conditions, such as monitoring the condition of production equipment. The robot is not autonomous, rather it is human-operated. The prototype was a success, and proved the maturity of the robotics field to provide secure and reliable applications in the field. Future developments include adding manipulation capabilities. DORIS, [104], is an other research project for designing a prototype of a mobile robot for remote supervision and data acquisition on offshore facilities. The main limitation of the robot is that is rail guided, and therefore its motion is limited. The ARGOS Challenge was a competition carried out from 2013 to 2017 whose objective was to apply mobile robots in the oil and gas industry. One of the most relevant proposed solution in the challenge was ANYmal, [21, 5], a four leg robot able to localize itself, detect obstacles, and plan trajectories in a predefined graph. Its main innovation is the ability to walk stairs, so it can access several floors inside within a facility. ROBOGAS, [105], is another mobile robot for gas leak-detection in a refinery and gas transportation company. The robot enabled autonomous path planning and obstacle avoidance. In [106], the authors used manipulator robots for inspection and maintenance in an offshore oil platform. A mobile manipulator was proposed, [107], for refinery inspections. The robot used the A^* algorithm for navigating facilities according to commands entered by the user. In [108], the authors use ABB robots to perform remote inspection in oil and gas facilities. Although they all represent significant advances in the field, none of the previous developments work towards the integration of task and motion planning approaches described in Section 1.1.2:.

4.2.2 Opportunities and challenges in oil and gas industry

After reviewing the relevant research projects in the area, we can see that most robots either cannot autonomously navigate in the environment or they only consider motion/path planning algorithms to navigate from an initial point to a final point. Moreover, task allocation, and task planning problems are not considered. Furthermore, the sequence of activities (task planning) is instructed to the robot directly by a human operator, (i.e. the robot is given a set of actions to perform sequentially), so more efficient solutions cannot be computed by the robot itself. To the best of our knowledge, there is no robotic solution which combines task and motion planning for robots in the oil and gas industry. Specifically, there is no evident use of a formal language for the specification of more general missions that a robot can allocate by itself with the objective to do it in the most effective possible, as presented in Chapter 3. One of the main desired properties of intra-logistics systems, as explained in Section 1.2, is that new tasks should be easily given to the robots without the need of a specialist to program the robot every time there is a new request; this is also a desired property of robotic systems applications in the oil and gas sector, [72]. Another important observation is that we could not find robotic applications specifically for material handing in the oil and gas industry, although it has been mentioned that it represents an important area of application, [28].

4.3 Applications

It is important to address challenging sub-problems like the ones presented in Section 1 on reliable and intelligent robotic systems, [108, 22]. Furthermore, an important landmark for robots working in the oil and gas industry is the incremental autonomy and reliability of the robots, which should now be improving after the first real implementations, [27]. In this section, we explore the application of the models presented in Chapter 3, which try to approach the general pickup and delivery problem in robotic networks, since we believe the applications of this methods will have a significant impact in mobile and manipulator robots performing several intra-logistics operations.

The introduction of robots into challenging oil and gas environments is marked by a tendency to implement robotic systems in real-life simulated scenarios, i.e., simulated oil and gas facilities in research labs. We limit our application domain to that of a conceptual experiment and leave simulation and real-life simulated scenarios for future work. In addition, we assume there exists a roadmap of the environment and the robots rely on pre-computed paths. In that sense, we will only shown examples of the models presented in Chapter 3.

4.3.1 Case scenario: Inspection

Inspection is one of the key activities in oil and gas operations that we believe can be delegated to robots. Most inspection activities are related to monitoring of pipelines, or reading data from valves or other sensors. Those valves or sensors are located in various areas which the robots need to visit to get readings from them.

Example 4.3.1. (Inspection) In this example, $O = \{\text{sensor}_1, \text{sensor}_2, \text{sensor}_3, \text{sensor}_4\}$ represents the sensors with related positions y_o and an available quantity $m_o = 0$ for all $o \in O$, while $D = \{\text{upload}\}$ is the upload data location with position y_d and request $m_d = 0$ for all $d \in D$. Let the mission be "read all sensor data at location $\text{sensor}_1, \text{sensor}_2, \text{sensor}_3, \text{sensor}_4$; before every reading upload current data at upload location", which can be represented by formula $M = \mathcal{F}(''read''_1 \wedge \mathscr{X}''upload'') \wedge \mathcal{F}(''read''_2 \wedge \mathscr{X}''upload'') \wedge \mathcal{F}(''read''_3 \wedge \mathscr{X}''upload'')$. Then we can use the method presented in 3 to obtain a set of action trajectories such that they accomplish M.

4.3.2 Case scenario: Material handling

In the previous example, there was no quantity to be collected or delivered, as we were dealing only with information flow. The example we present next deals with associated quantities to be collected at specific points.

Example 4.3.2. (Material handling) We have three objects $O = \{o_1, o_2, o_3\}$ to be collected located at $y_{o_1}, y_{o_2}, y_{o_3}$ and quantities $m_{o_1} = 15, m_{o_2} = 8, m_{o_3} = 9$, while $D = \{d_1, d_2, d_3\}$ are the delivery locations with position $y_{d_1}, y_{d_2}, y_{d_3}$ and required quantities $m_{d_1} = -8, m_{d_2} = -23, m_{d_3} = -1$. Let the mission be "collect objects o_3, o_1, o_2, in that order; then deliver the objects at locations d_3, d_1, d_2, in that order". This can be represented by formula $M = \mathcal{F}(''pickupo''_3 \wedge (\mathcal{F}(''pickupo''_1 \wedge (\mathcal{F}(''pickupo''_2 \wedge (\mathcal{F}(''deliverd''_3 \wedge (\mathcal{F}(''deliverd''_1 \wedge \mathcal{F}(''deliverd''_2)))))$. Similarly we can use method 3 for solving such problem.

Chapter 5

Conclusion

The main idea of this work was to work forward mobile robots with a higher degree of autonomy. We focus our attention on the planning cognition algorithms used by mobile robots. We find that there are three main areas of research in which improvement are significant for our purposes. These area were described in Section 1.2. Given the recent success of formal languages in robotics, we follow the trend to develop a method to obtain action trajectories of a group or robots under pickup and delivery tasks. Finally, we explore the applicability of our methods in the oil and gas industry.

5.1 Research contributions

The contributions of this work are:

1. A mixed integer linear programming formulation for the multi-vehicle pick up and delivery problem with split load constraints.

2. The multi-vehicle pickup and delivery problem with linear temporal logic, and a method to solve the problem.

3. A discussion on how works can be integrated into our approach to account for the motion equations of the robot.

4. A demonstration that autonomous agents for intra-logistics applications

in the oil and gas industry is a strong area of opportunity for future research.

5.2 Research limitations

Our main research limitations are that our approach still relies in the creation of construction of the automata from any formula given in linear temporal logic.

5.3 Future research

There are several future research directions to explore that we summarize in this section. Areas of future interest for the work presented in Chapter 2 are the following:

1. As presented in Chapter 2, our formulation 2.2.2 aims to minimize the total traveling time of the entire system (set of robots). We would like to reformulate it such that it minimize the individual traveling time, which means we are dealing with a multi-objective optimization problem.

2. Create custom heuristics or explore other approaches for solving larger instances of the problem in Chapter 2. The use of reinforcement learning has been shown to be a suitable candidate to solve similar problems.

3. Investigate the applicability of sample based motion planing algorithms to solve problems like those in section 2.2.2. Such algorithms have been successfully applied in automated planning, so we believe they can be adapted to solve our problem.

For the work presented in Chapter 3, we extended our formulation to account for the use of formal languages as an alternative to state constraints or behaviour to our model. Normally, the formulation of linear constraints is a tedious trial and error process. The formulation becomes easier by the incorporation of formal languages. However, the construction of the automata from linear temporal logic is computationally expensive, therefore, our approach can be used only with small formulas.

1. Another important area of research would be the use of other types of formal languages, which do not rely on the construction of automata such as signal temporal logic.

2. Finally, we would like to explore the possibility of exploring deeper the applications of our methods in robotics in the oil and gas industry.

Bibliography

[1] P. Tamás, B. Illés, and P. Dobos. "Waste reduction possibilities for manufacturing systems in the industry 4.0." In: *IOP Conference Series: Materials Science and Engineering* 161.1 (Nov. 2016), p. 012074. ISSN: 1757-8981. DOI: 10.1088/1757-899X/161/1/012074. URL: https://iopscience.iop.org/article/10.1088/1757-899X/161/1/012074.

[2] Kai Furmans, Zäzilia Seibold, and Andreas Trenkle. "Future Technologies in Intralogistics and Material Handling." In: *Operations, Logistics and Supply Chain Management*. Cham: Springer, Cham, 2019, pp. 545–574. DOI: 10.1007/978-3-319-92447-2_24. URL: http://link.springer.com/10.1007/978-3-319-92447-2%7B%5C_%7D24.

[3] Dali Sun. "Adaptive Task Allocation, Localization and Motion Planning for the Multi-Robot System." PhD thesis. Universität Freiburg, 2017, p. 138.

[4] Mustafa Güller, Yılmaz Uygun, and E. Karakaya. "Multi-Agent Simulation for Concept of Cellular Transport System in Intralogistics." In: *Lecture Notes in Mechanical Engineering*. Vol. 7. Springer Heidelberg, 2013, pp. 233–244. DOI: 10.1007/978-3-319-00557-7_19. URL: http://link.springer.com/10.1007/978-3-319-00557-7%7B%5C_%7D19.

[5] Matthias Heutger and Markus Kuckelhaus. *Self-driving vehicles in logistics: A DHL perspective on implications and use cases for the logistics industry*. Tech. rep. 2014, pp. 1–39. URL: https://discover.dhl.com/content/dam/dhl/downloads/interim/full/dhl-self-driving-vehicles.pdf.

[6] Jon Martin et al. "An Autonomous Transport Vehicle in an existing manufacturing facility with focus on the docking maneuver task." In: *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. Nagoya: IEEE, 2017, pp. 365–370. ISBN: 978-1-5090-6088-7. DOI: 10.1109/ICCAR.2017.7942719. URL: http://ieeexplore.ieee.org/document/7942719/.

[7] Quang-Vinh Dang et al. "Scheduling a single mobile robot for part-feeding tasks of production lines." In: *Journal of Intelligent Manufacturing* 25.6 (2014), pp. 1271–1287. ISSN: 0956-5515. DOI: 10.1007/s10845-013-0729-y. URL: http://link.springer.com/10.1007/s10845-013-0729-y.

[8] EUrobotics. *Robotics 2020 Multi-Annual Roadmap*. Tech. rep. 2017, pp. 178–228. DOI: 10.3917/deba.064.0181. URL: https://www.eu-robotics.net/cms/upload/topic%7B%5C_%7Dgroups/H2020%7B%5C_%7DRobotics%7B%5C_%7DMulti-Annual%7B%5C_%7DRoadmap%7B%5C_%7DICT-2017B.pdf%7B%5C%%7D0Ahttps://www.eu-robotics.net/cms/upload/topic%7B%5C_%7Dgroups/H2020%7B%5C_%7DRobotics%7B%5C_%7DMulti-Annual%7B%5C_%7DRoadmap%7B%5C_%7DICT-2017B.pdf%7B%5C%%7D0Ahttp://www.eu-robotics.net/cms/index.php?idcat=.

[9] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. "Coordinating Hundreads of Cooperative, Autonomous Vehicles in Warehouses." In: *AI Magazine* 29.1 (2008), p. 9. ISSN: 0738-4602. DOI: 10.1609/aimag.v29i1.2082. arXiv: 1605.03373. URL: http://portal.acm.org/citation.cfm?doid=234313.234327%7B%5C%%7D0Ahttps://doi.org/10.1007/s10514-018-9726-5%7B%5C%%7D0Ahttp://www.academia.edu/download/30491528/seminarpaper.pdf%7B%5C%%7D5Cnhttp://www.aaai.org/ojs/index.php/aimagazine/article/view/2082%7B%5C%%7D0Ahttp://ieeexplore.ieee.org/d.

[10] Evan Ackerman. *Amazon Uses 800 Robots To Run This Warehouse IEEE Spectrum - IEEE Spectrum*. 2019. URL: https://spectrum.ieee.org/automaton/robotics/industrial-robots/amazon-introduces-two-new-warehouse-robots (visited on 12/01/2019).

[11] Lorenzo Sabattini et al. "The PAN-Robots Project: Advanced Automated Guided Vehicle Systems for Industrial Logistics." In: *IEEE Robotics & Automation Magazine* 25.1 (Mar. 2018), pp. 55–64. ISSN: 1070-9932. DOI: 10.1109/MRA.2017.2700325. URL: https://ieeexplore.ieee.org/document/8103931/.

[12] Swisslog. *Swisslog*. URL: https://www.swisslog.com/es-mx/search-results-page?q=carrypick (visited on 08/08/2020).

[13] Harold W. Kuhn. "The Hungarian method for the assignment problem." In: *Naval Research Logistics Quarterly* 2.1-2 (Mar. 1955), pp. 83–97. ISSN: 00281441. DOI: 10.1002/nav.3800020109. URL: http://doi.wiley.com/10.1002/nav.3800020109.

[14] Anthony Stentz. "Optimal and efficient path planning for unknown and dynamic environments." In: *International Journal of Robotics and Automation* 10.3 (1993), pp. 89–100. ISSN: 08268185.

[15] Peter Hart, Nils Nilsson, and Bertram Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107. ISSN: 0536-1567. DOI: 10.1109/TSSC.1968.300136. URL: http://ieeexplore.ieee.org/document/4082128/.

[16] Richard Bloss. “Mobile hospital robots cure numerous logistic needs.” In: *Industrial Robot* 38.6 (2011), pp. 567–571. ISSN: 0143991X. DOI: 10.1108/01439911111179075.

[17] Andreas Trenkle, Michel Gohl, and Kai Furmans. “Interpretation of pointing gestures for the gesture controlled transportation robot FIFI;” in: *2015 Annual IEEE Systems Conference (SysCon) Proceedings*. IEEE, 2015, pp. 721–726. ISBN: 978-1-4799-5927-3. DOI: 10.1109/SYSCON.2015.7116836. URL: http://ieeexplore.ieee.org/document/7116836/.

[18] G. Kartnig, B. Grösel, and N. Zrnic. “Past, state-of-the-art and future of intralogistics in relation to megatrends.” In: *FME Transactions* 40.4 (2012), pp. 193–200. ISSN: 14512092.

[19] Dieter Arnold. *Intralogistik*. Ed. by Dieter Arnold. Vol. 16. 5. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 52–56. ISBN: 978-3-540-29657-7. DOI: 10.1007/978-3-540-29658-4. URL: http://link.springer.com/10.1007/978-3-540-29658-4.

[20] Theresa Beyer et al. “Agent-based dimensioning to support the planning of Intra-Logistics systems.” In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Berlin: IEEE, 2016, pp. 1–4. ISBN: 978-1-5090-1314-2. DOI: 10.1109/ETFA.2016.7733647. URL: http://ieeexplore.ieee.org/document/7733647/.

[21] Marco Hutter et al. “Towards a Generic Solution for Inspection of Industrial Sites.” In: *Field and Service Robotics*. 2018, pp. 575–589. DOI: 10.1007/978-3-319-67361-5_37. URL: http://link.springer.com/10.1007/978-3-319-67361-5%7B%5C_%7D37.

[22] D A Anisi et al. “Robot automation in oil and gas facilities: Indoor and onsite demonstrations.” In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 4729–4734. ISBN: 978-1-4244-6674-0. DOI: 10.1109/IROS.2010.5649281. URL: http://ieeexplore.ieee.org/document/5649281/.

[23] Daniel Claes et al. “Decentralised online planning for multi-robot warehouse commissioning.” In: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*. Vol. 1. Sao Paulo, May 2017, pp. 492–500. ISBN: 9781510855076. URL: www.ifaamas.org.

[24] Håvard Devold. *Oil and gas production handbook An introduction to oil and gas production, transport, refining and petrochemical industry*. Tech. rep. ABB, 2013. URL: www.wikimedia.org..

[25] Matthias Bengel et al. "Mobile robots for offshore inspection and manipulation." In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. st. Louis: IEEE, Oct. 2009, pp. 3317–3322. ISBN: 978-1-4244-3803-7. DOI: 10.1109/IROS.2009.5353885. URL: http://ieeexplore.ieee.org/document/5353885/.

[26] Samuel Soldan, Gero Bonow, and Andreas Kroll. "RoboGas Inspector - A mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results." In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Vol. 1. PART 1. Trondheim, 2012, pp. 33–38. ISBN: 9783902661999. DOI: 10.3182/20120531-2-NO-4020.00005.

[27] Aksel A. Transeth et al. "Robotics for the Petroleum Industry - Challenges and Opportunities." In: *SPE Middle East Intelligent Energy Conference and Exhibition*. Vol. 2. October. Manama: Society of Petroleum Engineers, Oct. 2013, pp. 282–289. ISBN: 9781629934358. DOI: 10.2118/167417-MS. URL: http://www.onepetro.org/doi/10.2118/167417-MS.

[28] Birgit Graf and Kai Pfeiffer. *Mobile robotics for offshore automation*. Tech. rep. 2008. URL: www.offshore-robotics.com%20http://www.robot.uji.es/documents/rise08/reports/Graf.pdf.

[29] Henrik Andreasson et al. "Autonomous Transport Vehicles: Where We Are and What Is Missing." In: *IEEE Robotics & Automation Magazine* 22.1 (2015), pp. 64–75. ISSN: 1070-9932. DOI: 10.1109/MRA.2014.2381357. URL: http://ieeexplore.ieee.org/document/7059356/.

[30] Jungwon Seo, Jamie Paik, and Mark Yim. "Modular Reconfigurable Robotics." In: *Annual Review of Control, Robotics, and Autonomous Systems* 2.1 (May 2019), pp. 63–88. ISSN: 2573-5144. DOI: 10.1146/annurev-control-053018-023834. URL: https://www.annualreviews.org/doi/10.1146/annurev-control-053018-023834.

[31] Hang Ma et al. "Overview: A Hierarchical Framework for Plan Generation and Execution in Multirobot Systems." In: *IEEE Intelligent Systems* 32.6 (2017), pp. 6–12. ISSN: 1541-1672. DOI: 10.1109/MIS.2017.4531217. URL: www.computer.org/intelligent%20http://ieeexplore.ieee.org/document/8268002/.

[32] Spyros Maniatopoulos et al. "Reactive high-level behavior synthesis for an Atlas humanoid robot." In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm: IEEE, May 2016, pp. 4192–4199. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487613. URL: http://ieeexplore.ieee.org/document/7487613/.

[33] Brian Coltin and Manuela Veloso. "Online pickup and delivery planning with transfers for mobile robots." In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, 2014, pp. 5786–5791. ISBN: 978-1-4799-3685-4. DOI: 10.1109/ICRA.2014.6907709. URL: http://ieeexplore.ieee.org/document/6907709/.

[34] Shih Yun Lo, Shani Alkoby, and Peter Stone. "Robust motion planning and safety benchmarking in human workspaces." In: *CEUR Workshop Proceedings*. Vol. 2301. 2019.

[35] Derya Aksaray, Cristian-Ioan Vasile, and Calin Belta. "Dynamic routing of energy-aware vehicles with Temporal Logic Constraints." In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockolm: IEEE, May 2016, pp. 3141–3146. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487481. URL: http://ieeexplore.ieee.org/document/7487481/.

[36] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Second. 2011, p. 472. ISBN: 9780262015356.

[37] Niki Kousi et al. "Scheduling of smart intra – factory material supply operations using mobile robots." In: *International Journal of Production Research* 57.3 (Feb. 2019), pp. 801–814. ISSN: 0020-7543. DOI: 10.1080/00207543.2018.1483587. URL: https://www.tandfonline.com/doi/full/10.1080/00207543.2018.1483587.

[38] Federico Pecora and Marcello Cirillo. "A Constraint-Based Approach for Multiple Non-Holonomic Vehicle Coordination in Industrial Scenarios." In: *ICAPS 2012 Workshop on Combining Task and Motion Planning for Real-World Applications*. Sao Paulo, 2012, pp. 45–52. URL: http://www.kivasystems.com/.

[39] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. "Synthesis for Robots: Guarantees and Feedback for Robot Behavior." In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (May 2018), pp. 211–236. ISSN: 2573-5144. DOI: 10.1146/annurev-control-060117-104838. URL: https://doi.org/10.1146/annurev-control-060117-%20https://www.annualreviews.org/doi/10.1146/annurev-control-060117-104838.

[40] Izabela Nielsen, Ngoc Anh Dung Do, and Peter Nielsen. "Scheduling Part-Feeding Tasks for a Single Robot with Feeding Quantity Consideration." In: *Advances in Intelligent Systems and Computing*. Vol. 373. Springer Verlag, 2015, pp. 349–356. ISBN: 9783319196374. DOI: 10.1007/978-3-319-19638-1_40. URL: http://link.springer.com/10.1007/978-3-319-19638-1%7B%5C_%7D40.

[41] Gilbert Laporte. "What you should know about the vehicle routing problem." In: *Naval Research Logistics* 54.8 (Dec. 2007), pp. 811–819. ISSN: 0894069X. DOI: 10.1002/nav.20261. URL: http://doi.wiley.com/10.1002/nav.20261.

[42] Tonci Caric and Hrvoje Gold. *The Vehicle Routing Problem.* Ed. by Paolo Toth and Daniele Vigo. SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics, Jan. 2002, pp. 1–367. ISBN: 978-0-89871-498-2. DOI: 10.1137/1.9780898718515. URL: http://epubs.siam.org/doi/book/10.1137/1.9780898718515.

[43] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. "The vehicle routing problem: State of the art classification and review." In: *Computers & Industrial Engineering* 99.1 (Sept. 2016), pp. 300–313. ISSN: 03608352. DOI: 10.1016/j.cie.2015.12.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360835215004775.

[44] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. "The vehicle routing problem: A taxonomic review." In: *Computers & Industrial Engineering* 57.4 (Nov. 2009), pp. 1472–1483. ISSN: 03608352. DOI: 10.1016/j.cie.2009.05.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360835209001405.

[45] Gerardo Berbeglia et al. "Static pickup and delivery problems: a classification scheme and survey." In: *TOP* 15.1 (May 2007), pp. 1–31. ISSN: 1134-5764. DOI: 10.1007/s11750-007-0009-0.

[46] Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. "A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations." In: *Journal fur Betriebswirtschaft* 58.2 (June 2008), pp. 81–117. ISSN: 03449327. DOI: 10.1007/s11301-008-0036-4.

[47] Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. "A survey on pickup and delivery problems." In: *Journal fur Betriebswirtschaft* 58.1 (May 2008), pp. 21–51. ISSN: 03449327. DOI: 10.1007/s11301-008-0033-7.

[48] Maria Battarra, Jean-François Cordeau, and Manuel Iori. "Pickup-and-Delivery Problems for Goods Transportation." In: *Vehicle Routing: Problems, Methods, and Applications.* Society for Industrial and Applied Mathematics, 2014, pp. 161–191. URL: http://www.siam.org/journals/ojsa.php.

[49] Maria Gabriela S. Furtado, Pedro Munari, and Reinaldo Morabito. "Pickup and delivery problem with time windows: A new compact two-index formulation." In: *Operations Research Letters* 45.4 (July 2017), pp. 334–341. ISSN: 01676377. DOI: 10.1016/j.orl.2017.04.013.

[50] Gábor Nagy and Saıd Salhi. "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries." In: *European Journal of Operational Research* 162.1 (2005), pp. 126–141. ISSN: 03772217. DOI: 10.1016/j.ejor.2002.11.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221703008361.

[51] Hipólito Hernández-Pérez and Juan-José Salazar-González. "The One-Commodity Pickup-and-Delivery Travelling Salesman Problem." In: *Combinatorial Optimization—Eureka, You Shrink!* Berlin: Springer, Berlin, Heidelberg, 2003, pp. 89–104. DOI: 10.1007/3-540-36478-1_10. URL: http://link.springer.com/10.1007/3-540-36478-1%7B%5C_%7D10.

[52] Hipólito Hernández-Pérez and Juan José Salazar-González. "A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery." In: *Discrete Applied Mathematics* 145.1 (2004), pp. 126–139. ISSN: 0166218X. DOI: 10.1016/j.dam.2003.09.013.

[53] Hipólito Hernández-Pérez and Juan-José Salazar-González. "Heuristics for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem." In: *Transportation Science* 38.2 (May 2004), pp. 245–255. ISSN: 0041-1655. DOI: 10.1287/trsc.1030.0086. URL: http://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0086.

[54] Juan José Salazar-González and Beatriz Santos-Hernández. "The split-demand one-commodity pickup-and-delivery travelling salesman problem." In: *Transportation Research Part B: Methodological* 75.May (May 2015), pp. 58–73. ISSN: 01912615. DOI: 10.1016/j.trb.2015.02.014.

[55] Michael R. Garey and David S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* June 1990, p. 338.

[56] Rafiq Ahmad and Peter Plapper. "Human-Robot Collaboration: Twofold Strategy Algorithm to Avoid Collisions Using ToF Sensor." In: *International Journal of Materials, Mechanics and Manufacturing* 4.2 (May 2015), pp. 144–147. ISSN: 17938198. DOI: 10.7763/IJMMM.2016.V4.243. URL: http://www.ijmmm.org/index.php?m=content%7B%5C&%7Dc=index%7B%5C&%7Da=show%7B%5C&%7Dcatid=43%7B%5C&%7Did=291.

[57] Ernesto Nunes et al. "A taxonomy for task allocation problems with temporal and ordering constraints." In: *Robotics and Autonomous Systems* 90.C (2017), pp. 55–70. ISSN: 09218890. DOI: 10.1016/j.robot.2016.10.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889016306157.

[58] Jan Schuhmacher and Vera Hummel. "Self-organization of changeable intralogistics systems at the ESB Logistics Learning Factory." In: *Procedia Manufacturing* 31.1 (Apr. 2019), pp. 194–199. ISSN: 23519789.

DOI: 10.1016/j.promfg.2019.03.031. URL: https://linkinghub.elsevier.com/retrieve/pii/S2351978919303956.

[59] Raffaello D'Andrea and Peter Wurman. "Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities." In: *2008 IEEE International Conference on Technologies for Practical Robot Applications*. MA: IEEE, 2008, pp. 80–83. ISBN: 978-1-4244-2791-8. DOI: 10.1109/TEPRA.2008.4686677. URL: http://ieeexplore.ieee.org/document/4686677/.

[60] Brian Coltin and Manuela Veloso. "Optimizing for Transfers in a Multi-vehicle Collection and Delivery Problem." In: *Springer Tracts in Advanced Robotics*. Vol. 104. Springer Verlag, 2014, pp. 91–103. ISBN: 9783642551451. DOI: 10.1007/978-3-642-55146-8_7. URL: http://link.springer.com/10.1007/978-3-642-55146-8%7B%5C_%7D7.

[61] Karl F Doerner and Juan-José Salazar-González. "Pickup-and-Delivery Problems for People Transportation." In: *Vehicle Routing: Problems, Methods, and Applications*. 2014, pp. 193–212. URL: http://www.siam.org/journals/ojsa.php.

[62] Manuela Veloso et al. "Symbiotic-Autonomous Service Robots for User-Requested Tasks in a Multi-Floor Building." In: *Cognitive Assistive Systems (IROS)*. Algarve, Oct. 2012, pp. 19–25. URL: http://repository.cmu.edu/compscihttp://repository.cmu.edu/compsci/2801.

[63] Neil Mathew, Stephen L. Smith, and Steven L. Waslander. "Planning Paths for Package Delivery in Heterogeneous Multirobot Teams." In: *IEEE Transactions on Automation Science and Engineering* 12.4 (2015), pp. 1298–1308. ISSN: 15455955. DOI: 10.1109/TASE.2015.2461213.

[64] Brian Coltin and Manuela Veloso. *Scheduling for Transfers in Pickup and Delivery Problems with Very Large Neighborhood Search*. 2014. URL: www.aaai.org.

[65] Zhe Liu et al. "Distributed pair-wised transportation planning with incidental deliveries for multiple mobile robots." In: *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. Okinawa: IEEE, 2017, pp. 194–199. ISBN: 978-1-5386-2035-9. DOI: 10.1109/RCAR.2017.8311859. URL: http://ieeexplore.ieee.org/document/8311859/.

[66] Zhe Liu et al. "An Incidental Delivery Based Method for Resolving Multirobot Pairwised Transportation Problems." In: *IEEE Transactions on Intelligent Transportation Systems* 17.7 (July 2016), pp. 1852–1866. ISSN: 15249050. DOI: 10.1109/TITS.2015.2508783.

[67] Hang Ma et al. "Overview: Generalizations of Multi-Agent Path Finding to Real-World Scenarios." In: *Arxiv* (Feb. 2017). arXiv: 1702.05515. URL: http://arxiv.org/abs/1702.05515.

[68] Hang Ma et al. "Multi-agent path finding with payload transfers and the package-exchange robot-routing problem." In: *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016).* Phoenix, Feb. 2016, pp. 3166–3173. ISBN: 9781577357605. URL: https://www.researchgate.net/profile/Hang%7B%5C_%7DMa7/publication/296706300%7B%5C_%7DMulti-Agent%7B%5C_%7DPath%7B%5C_%7DFinding%7B%5C_%7Dwith%7B%5C_%7DPayload%7B%5C_%7DTransfers%7B%5C_%7Dand%7B%5C_%7Dthe%7B%5C_%7DPackage-Exchange%7B%5C_%7DRobot-Routing%7B%5C_%7DProblem/links/56d9f4d708aebe4638bb9d49.pdf.

[69] Hang Ma et al. "Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks." In: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017),* Sao Paulo, May 2017, pp. 837–845. arXiv: 1705.10868. URL: http://arxiv.org/abs/1705.10868.

[70] Alessandro Farinelli et al. "Advanced approaches for multi-robot coordination in logistic scenarios." In: *Robotics and Autonomous Systems* 90.1 (2017), pp. 34–44. ISSN: 09218890. DOI: 10.1016/j.robot.2016.08.010.

[71] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. "Distributed Constraint Optimization Problems and Applications: A Survey." In: *Journal of Artificial Intelligence Research* 61.1 (2018), pp. 623–698. ISSN: 1076-9757. DOI: 10.1613/jair.5565. URL: https://jair.org/index.php/jair/article/view/11185.

[72] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Vol. 89. Studies in Systems, Decision and Control. Cham: Springer International Publishing, 2017, p. 284. ISBN: 978-3-319-50762-0. DOI: 10.1007/978-3-319-50763-7. URL: http://www.springer.com/series/13304%20http://link.springer.com/10.1007/978-3-319-50763-7.

[73] Erion Plaku. "Planning in Discrete and Continuous Spaces: From LTL Tasks to Robot Motions." In: *Conference Towards Autonomous Robotic Systems*. 2012, pp. 331–342. DOI: 10.1007/978-3-642-32527-4_30. URL: http://link.springer.com/10.1007/978-3-642-32527-4%7B%5C_%7D30.

[74] Alphan Ulusoy, Stephen L. Smith, and Calin Belta. "Optimal Multi-Robot Path Planning with LTL Constraints: Guaranteeing Correctness through Synchronization." In: *Springer Tracts in Advanced Robotics*. Vol. 104. Springer Verlag, 2014, pp. 337–351. ISBN: 9783642551451. DOI: 10.1007/978-3-642-55146-8_24. arXiv: 1207.2415. URL: http://link.springer.com/10.1007/978-3-642-55146-8%7B%5C_%7D24.

[75] Felipe J. Montana, Jun Liu, and Tony J. Dodd. "Sampling-Based Path Planning for Multi-robot Systems with Co-Safe Linear Temporal Logic Specifications." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10471 LNCS. Lecture Notes in Computer Science. 2017, pp. 150–164. ISBN: 9783319671123. DOI: 10.1007/978-3-319-67113-0_10. URL: http://link.springer.com/10.1007/978-3-319-67113-0%7B%5C_%7D10.

[76] Ryan Luna et al. "Asymptotically Optimal Stochastic Motion Planning with Temporal Goals." In: *Springer Tracts in Advanced Robotics*. Vol. 107. Springer, Cham, 2015, pp. 335–352. ISBN: 9783319165943. DOI: 10.1007/978-3-319-16595-0_20. URL: http://link.springer.com/10.1007/978-3-319-16595-0%7B%5C_%7D20.

[77] Erion Plaku and Sertac Karaman. "Motion planning with temporal-logic specifications: Progress and challenges." In: *AI Communications*. Vol. 29. 1. 2016, pp. 151–162. DOI: 10.3233/AIC-150682.

[78] Philipp Schillinger, Mathias Bürger, and Dimos V. Dimarogonas. "Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems." In: *International Journal of Robotics Research* 37.7 (June 2018), pp. 818–838. ISSN: 17413176. DOI: 10.1177/0278364918774135.

[79] Fatma Faruq et al. "Simultaneous Task Allocation and Planning Under Uncertainty." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, 2018, pp. 3559–3564. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594404. arXiv: 1803.02906. URL: https://ieeexplore.ieee.org/document/8594404/.

[80] Yiannis Kantaros and Michael M. Zavlanos. "STyLuS * : A Temporal Logic Optimal Control Synthesis Algorithm for Large-Scale Multi-Robot Systems." In: *The International Journal of Robotics Research* 39.7 (June 2020), pp. 812–836. ISSN: 0278-3649. DOI: 10.1177/0278364920913922. arXiv: 1809.08345. URL: http://arxiv.org/abs/1809.08345%20http://journals.sagepub.com/doi/10.1177/0278364920913922.

[81] Etienne Dargaud. "Pick-Up and Delivery Planning in Multi-Agent Systems under Temporal Logic Specifications." PhD thesis. KTH Royal Institute of Technology, 2013, p. 34.

[82] Philipp Schillinger, Mathias Bürger, and Dimos V. Dimarogonas. "Decomposition of Finite LTL Specifications for Efficient Multi-agent Planning." In: *Distributed Autonomous Robotic Systems*. Springer, Cham, 2018, pp. 253–267. DOI: 10.1007/978-3-319-73008-0_18. URL: http://link.springer.com/10.1007/978-3-319-73008-0%7B%5C_%7D18.

[83] Philipp Schillinger, Mathias Burger, and Dimos V. Dimarogonas. "Multi-objective search for optimal multi-robot planning with finite LTL specifications and resource constraints." In: *Proceedings - IEEE International Conference on Robotics and Automation*. 2017, pp. 768–774. ISBN: 9781509046331. DOI: 10.1109/ICRA.2017.7989094.

[84] Vladislav Nenchev, Calin Belta, and Jorg Raisch. "Optimal motion planning with temporal logic and switching constraints." In: *2015 European Control Conference (ECC)*. IEEE, July 2015, pp. 1141–1146. ISBN: 978-3-9524-2693-7. DOI: 10.1109/ECC.2015.7330693. URL: http://ieeexplore.ieee.org/document/7330693/.

[85] Jean-François Raskin. "An Introduction to Hybrid Automata." In: *Handbook of Networked and Embedded Control Systems*. Boston, MA: Birkhäuser Boston, 2005, pp. 491–517. DOI: 10.1007/0-8176-4404-0_21. URL: http://link.springer.com/10.1007/0-8176-4404-0%7B%5C_%7D21.

[86] Cristian-ioan Vasile and Calin Belta. "An Automata-Theoretic Approach to the Vehicle Routing Problem." In: *Robotics: Science and Systems X*. Berkeley: Robotics: Science and Systems Foundation, 2014, pp. 1–9. ISBN: 9780992374709. DOI: 10.15607/RSS.2014.X.045. URL: http://www.roboticsproceedings.org/rss10/p45.pdf.

[87] S. Karaman and E. Frazzoli. "Linear temporal logic vehicle routing with applications to multi-UAV mission planning." In: *International Journal of Robust and Nonlinear Control* 21.12 (2011), pp. 1372–1395. ISSN: 10498923. DOI: 10.1002/rnc.1715.

[88] Sertac Karaman and Emilio Frazzoli. "Vehicle Routing Problem with Metric Temporal Logic Specifications." In: *2008 47th IEEE Conference on Decision and Control*. Cancun: IEEE, Dec. 2008, pp. 3953–3958. ISBN: 978-1-4244-3123-6. DOI: 10.1109/CDC.2008.4739366. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4739366.

[89] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Third. Vol. 32. 1. Pearson/Addison Wesley, 2001, p. 60. ISBN: 978-1292039053. DOI: 10.1145/568438.568455.

[90] Christel. Baier and Joost-Pieter. Katoen. *Principles Of Model Checking*. Vol. 950. MIT Press, 2008, pp. I–XVII, 1–975. ISBN: 9780262026499. DOI: 10.1093/comjnl/bxp025. URL: http://mitpress.mit.edu/books/principles-model-checking.

[91] Amir Pnueli. "The temporal logic of programs." In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. Providence: IEEE, Sept. 1977, pp. 46–57. DOI: 10.1109/SFCS.1977.32. URL: http://ieeexplore.ieee.org/document/4567924/.

[92] Orna Kupferman and MosheY. Vardi. “Model Checking of Safety Properties.” In: *Formal Methods in System Design*. Vol. 19. 3. 1999, pp. 172–183. ISBN: 3540662022. DOI: 10.1007/3-540-48683-6_17. URL: http://link.springer.com/10.1007/3-540-48683-6%7B%5C_%7D17.

[93] A Prasad Sistla. “Safety, liveness and fairness in temporal logic.” In: *Formal Aspects of Computing* 6.5 (1994), pp. 495–511. ISSN: 0934-5043. DOI: 10.1007/BF01211865. URL: http://link.springer.com/10.1007/BF01211865.

[94] Timo Latvala. “Efficient Model Checking of Safety Properties.” In: *Proceedings - International Conference on Application of Concurrency to System Design, ACSD*. Berlin: Springer, Berlin, Heidelberg, 2003, pp. 74–88. ISBN: 9780769540665. DOI: 10.1007/3-540-44829-2_5. URL: http://link.springer.com/10.1007/3-540-44829-2%7B%5C_%7D5.

[95] Pierre Wolper. “Constructing Automata from Temporal Logic Formulas : A Tutorial.” In: *Lectures on Formal Methods and Performance Analysis*. Berlin: Springer, Berlin, Heidelberg, 2001, pp. 261–277. ISBN: 3-540-42479-2. DOI: 10.1007/3-540-44667-2_7. URL: http://www.montefiore.ulg.ac.be/%E2%88%BCpw/%20http://link.springer.com/10.1007/3-540-44667-2%7B%5C_%7D7.

[96] E. W. Dijkstra. “A note on two problems in connexion with graphs.” In: *Numerische Mathematik* 1.1 (1959), pp. 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390. URL: http://link.springer.com/10.1007/BF01386390.

[97] Rob Gerth et al. “Simple On-the-fly Automatic Verification of Linear Temporal Logic.” In: *Protocol Specification, Testing and Verification XV. PSTV 1995. IFIP Advances in Information and Communication Technology*. Boston, 1996, pp. 3–18. DOI: 10.1007/978-0-387-34892-6_1. URL: http://link.springer.com/10.1007/978-0-387-34892-6%7B%5C_%7D1.

[98] Alexandre Duret-Lutz et al. “Spot 2.0 — A Framework for LTL and \$\$\omega \$\$ -Automata Manipulation.” In: 2016, pp. 122–129. DOI: 10.1007/978-3-319-46520-3_8. URL: https://spot.lrde.epita.%20http://link.springer.com/10.1007/978-3-319-46520-3%7B%5C_%7D8.

[99] Amit Shukla and Hamad Karki. “Application of robotics in onshore oil and gas industry—A review Part I.” In: *Robotics and Autonomous Systems* 75.2 (Jan. 2016), pp. 490–507. ISSN: 09218890. DOI: 10.1016/j.robot.2015.09.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889015002006.

[100] Bahadur Ibrahimov and Manafaddin Namazov. “Robotics in petroleum and safety requirements forcing Open Innovation to be embraced.” In: *IFAC-PapersOnLine* 51.30 (Jan. 2018), pp. 688–692. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.11.215.

[101] Amit Shukla and Hamad Karki. “Application of robotics in offshore oil and gas industry— A review Part II.” In: *Robotics and Autonomous Systems* 75.2 (Jan. 2016), pp. 508–524. ISSN: 09218890. DOI: 10.1016/j.robot.2015.09.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889015002018.

[102] Charlotte Skourup and John Pretlove. *Remote inspection and intervention*. Tech. rep. 2012, pp. 0–5.

[103] _ JPT staff. “Sensabot: A Safe and Cost-Effective Inspection Solution.” In: *Journal of Petroleum Technology* 64.10 (Oct. 2012), pp. 32–34. ISSN: 0149-2136. DOI: 10.2118/1012-0032-JPT. URL: http://www.onepetro.org/doi/10.2118/1012-0032-JPT.

[104] Eduardo Nunes et al. “DORIS - Monitoring Robot for Offshore Facilities.” In: *OTC Brasil*. Vol. 2. Rio de Janeiro: Offshore Technology Conference, Oct. 2013, pp. 927–939. ISBN: 9781629933887. DOI: 10.4043/24386-MS. URL: http://www.onepetro.org/doi/10.4043/24386-MS.

[105] Samuel Soldan et al. “Towards Autonomous Robotic Systems for Remote Gas Leak Detection and Localization in Industrial Environments.” In: *Springer Tracts in Advanced Robotics*. Vol. 92. 2014, pp. 233–247. ISBN: 9783642406850. DOI: 10.1007/978-3-642-40686-7_16. URL: http://link.springer.com/10.1007/978-3-642-40686-7%7B%5C_%7D16.

[106] Erik Kyrkjebo, Pal Liljeback, and Aksel A. Transeth. “A Robotic Concept for Remote Inspection and Maintenance on Oil Platforms.” In: *Volume 1: Offshore Technology*. Vol. 1. Hawaii: ASMEDC, Jan. 2009, pp. 667–674. ISBN: 978-0-7918-4341-3. DOI: 10.1115/OMAE2009-79702. URL: https://asmedigitalcollection.asme.org/OMAE/proceedings/OMAE2009/43413/667/338308.

[107] John P. H. Steele et al. “Development of an Oil and Gas Refinery Inspection Robot.” In: *Volume 4A: Dynamics, Vibration, and Control*. Vol. 4A. montreal: American Society of Mechanical Engineers, Nov. 2014, p. 1. ISBN: 978-0-7918-4647-6. DOI: 10.1115/IMECE2014-36358. URL: https://proceedings.asmedigitalcollection.asme.org%20https://asmedigitalcollection.asme.org/IMECE/proceedings/IMECE2014/46476/Montreal,%20Quebec,%20Canada/262930.

[108] David A. Anisi and Charlotte Skourup. “A step-wise approach to oil and gas robotics.” In: *IFAC Proceedings Volumes* 45.8 (May 2012), pp. 47–52. ISSN: 14746670. DOI: 10.3182/20120531-2-NO-4020.

00022. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474667015372463.