University of Brasília
Institute of Exact Sciences
Department of Computer Science

Recovery Strategies for Multi-agent Planning in Dynamic Environments

Estratégias de Recuperação para Planejamento Multiagente em Ambientes Dinâmicos

Leonardo Henrique Moreira

Document submitted in partial fullfilment of the requirements to Doctoral Degree in Informatics

Advisor
Prof. Célia Ghedini Ralha

Brazil
2023

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Doutoradoem Informática

Coordinator: Prof. Ricardo Pezzuol Jacobi

Banca examinadora composta por:

Prof. Célia Ghedini Ralha (Orientadora) — CIC/UnB
Prof. Antonín Komenda — Czech Technical University in Prague
Prof. Edison Pignaton de Freitas — PUCRS
Prof. Li Weigang — CIC/UnB

CIP — Catalogação Internacional na Publicação

Moreira, Leonardo Henrique.

Recovery Strategies for Multi-agent Planning in Dynamic Environments

Estratégias de Recuperação para Planejamento Multiagente em Ambientes Dinâmicos / Leonardo Henrique Moreira. Brasília : UnB, 2023.
120 p. : il. ; 29,5 cm.

Tese de Doutorado — Universidade de Brasília, Brasília, 2023.

1. Ambientes dinâmicos, 2. Planejamento Multiagente, 3. Sistemas Multiagentes, 4. Planejamento Automatizado, 5. Estratégias de Recuperação de Planejamento, 6. Replanejamento, 7. Reparo

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

University of Brasília
Institute of Exact Sciences
Department of Computer Science

Recovery Strategies for Multi-agent Planning in Dynamic Environments

Estratégias de Recuperação para Planejamento Multiagente em Ambientes Dinâmicos

Leonardo Henrique Moreira

Document submitted in partial fullfilment of the requirements to Doctoral Degree in Informatics

Prof. Célia Ghedini Ralha (Advisor)
CIC/UnB

Prof. Antonín Komenda
Czech Technical University in Prague

Prof. Edison Pignaton de Freitas
PUCRS

Prof. Li Weigang
CIC/UnB

Prof. Ricardo Pezzuol Jacobi
Coordinator of the Graduate Program in Informatics

Brasília, June 30^{th} 2023

Dedication

To Michele, the best part of my life.

To our greatest blessings: Vitória and Helena.

To my parents, brother, and sister.

And to everyone who one day decided to start the long, tiring, but infinitely motivating journey of a doctorate.

Acknowledgements

First and foremost, I am grateful to God for guiding me toward the right path and providing me with the support of the right people. God has been present throughout every moment of my life, both good and bad.

I want to express my special appreciation to my wife, Michele, and my daughters, Vitória and Helena, who have been my pillars of strength. During my intense study, they had to bear my absence and lack of involvement. However, their unwavering love, care, and understanding made this journey more manageable. Without their support, I would not have been able to achieve my goals.

I am also thankful to my parents and siblings, who have been my role models and have inspired me to become the best version of myself.

I want to thank my committee members, Professors Li Weigang, Edison Pignaton de Freitas and Antonin Komenda, for their valuable contributions, constructive criticism, and, most importantly, their time.

Lastly, I am deeply indebted to my advisor, Professor Célia Ghedini Ralha, for her unwavering patience, dedication, and partnership. She has been a constant source of encouragement and has believed in me, helping me to achieve my dream. Her experience and guidance have been instrumental in completing this project, from my master's to my doctorate.

"In preparing for battle, I have always found that plans are useless but planning is indispensable."

Dwight D. Eisenhower

Abstract

This thesis explores Multi-Agent Planning (MAP) and its application in dynamic environments. MAP combines artificial intelligence planning with multi-agent systems to coordinate intelligent agents achieving individual or group goals. Planning in dynamic environments introduces challenges in coordination and execution due to non-deterministic outcomes. Plan recovery strategies, like replanning and repairing, aim to handle failures and restore desired conditions. A comprehensive literature review highlighted key contributors and institutions in the MAP research offering insights into concepts, techniques, and open challenges. However, combining different recovery strategies for MAP models is a research challenge yet to be accomplished in the present literature. Two problems are discussed in this thesis: incomplete assessments and a need for more research on cohesively integrating replanning and repair techniques. These problems are addressed to improve effectiveness and provide a comprehensive performance evaluation of the strategies. As solutions, we proposed an evaluation method for recovery strategies in dynamic environments to address the first issue. In addition, we designed a plan recovery process that combines replanning and repairing. This approach considers the complexity of planning, coordination during execution, and agents attempting local repairs before seeking assistance from other agents. The results highlight that recovery strategies perform similarly in scenarios with low agent coupling levels. Also, the results provided conditions to conclude that the number of goals and the failures affect execution under different conditions. Finally, the results point out that the plan recovery process design is suitable for various scenarios, highlighting that repairing should be explored in a decentralized approach while replanning in and centralized way.

Keywords: Dynamic Environments; Multi-agent Planning; Multi-agent Systems; Planning Recovery Strategies; Replanning; Repairing

Resumo Expandido

Planejamento é o processo de deliberação tomado por um agente, humano ou robô, ao escolher ações para mudar o ambiente para uma condição desejável [Ghallab et al., 2016]. O processo de planejamento concentra-se na escolha e organização das ações por meio de seus efeitos esperados. Outrossim, o planejamento depende da coordenação das ações selecionadas para fornecer a ordem e o tempo corretos para permitir a cooperação e evitar a competição entre os agentes. Além disso, um agente não é apenas um executor e, portanto, deve integrar planejamento e raciocínio ao longo do processo de execução [Ghallab et al., 2014].

Planejamento multiagente (PMA) é uma área de pesquisa em ciência da computação que integra o planejamento de inteligência artificial com sistemas multiagentes. Embora o planejamento seja uma área de pesquisa madura da inteligência artificial, geralmente com foco em tarefas de agente único, os sistemas multiagentes envolvem agentes inteligentes que trabalham cooperativamente ou competitivamente para atingir objetivos individuais ou de grupo. Além disso, técnicas e modelos de PMA podem ser aplicados em diversas áreas, desde ambientes com alto nível de controle até cenários onde a incerteza é uma condição presente.

Assim, as aplicações PMA podem ser adotadas em cenários comuns a complexos. A pesquisa na área de PMA é aplicável a problemas de mundo real, com interesse crescente em veículos autônomos, logística e automação de armazéns, sistemas multi-robôs e muitos outros domínios. Nesse contexto, dois pontos importantes podem ser destacados em tais aplicações. Primeiro, os agentes dependem ou afetam outros agentes em diferentes níveis, no que diz respeito à interação entre as execuções de suas ações. Em segundo lugar, os agentes precisam reagir dinamicamente às mudanças no estado do ambiente.

Em ambos os casos, cada agente depende da capacidade de monitorar seu próprio estado, possibilitando identificar falhas no sistema. Além disso, os agentes devem se comunicar e negociar para se adaptar efetivamente às mudanças no ambiente. A cooperação desempenha um papel crucial na organização de agentes dentro de um sistema, mas seu significado vai além da mera organização. A cooperação permite que os agentes prestem assistência uns aos outros quando necessário e ajuda a evitar interferências na coordenação de atividades individuais. Portanto, a competição entre os agentes pode ser eliminada, permitindo que eles trabalhem para alcançar objetivos comuns.

Vários fatores influenciam o processo de tomada de decisão e a complexidade de execução dentro de um PMA. A quantidade e capacidade dos recursos envolvidos, as condições operacionais e os procedimentos estratégicos são alguns dos aspectos que impactam significativamente as fases de planejamento, coordenação e execução. Em organizações hierárquicas, como empresas ou unidades militares, uma abordagem centralizada pode ser adotada para otimizar a utilização de recursos. Por outro lado, a execução pode ser descentralizada para permitir ações simultâneas. Sempre que possível, a execução das tarefas deve priorizar o desdobramento pontual e modular das capacidades, evitando a duplicação de ações e a dispersão de recursos, garantindo assim a eficácia e minimizando os custos operacionais e logísticos.

Além disso, em cenários onde os agentes não têm controle total sobre o ambiente, os processos de planejamento, coordenação e execução tornam-se significativamente mais desafiadores. Nesses casos, os agentes operam em um ambiente dinâmico onde os resultados de suas ações não são determinísticas. Consequentemente, ambientes dinâmicos são suscetíveis a eventos inesperados que podem resultar em falhas de planejamento. Uma extensa pesquisa tem sido dedicada a explorar a integração das dimensões de planejamento, coordenação e execução no contexto de múltiplos agentes. No entanto, existe a necessidade de aprofundar a investigação e resolução de questões relativas à dimensão de execução.

Dentro do domínio da pesquisa em planejamento, certas características tipicamente definem as abordagens propostas. Nos estudos classificados como PMA, a responsabilidade pelas capacidades de planejamento ou execução é distribuída entre os agentes. Normalmente, esses estudos introduzem modelos que consideram as ações como a única fonte de mudanças nos estados do ambiente, desconsiderando eventos inesperados decorrentes de interações [Chouhan and Niyogi, 2017, Štolba and Komenda, 2015, Torreño et al., 2014]. Consequentemente, essas abordagens assumem um ambiente totalmente observável e determinístico, sem incerteza quanto aos efeitos das ações.

No entanto, essas características limitam a aplicabilidade dessas abordagens em ambientes dinâmicos onde os agentes devem possuir a capacidade de lidar com falhas. Por exemplo, os agentes podem encontrar falhas de execução, problemas de comunicação e conhecimento limitado sobre os fatos circundantes. Como resultado, essas abordagens, com o planejamento sendo realizado em uma fase única e separada da execução, têm aplicações limitadas. Nesses casos, é impossível lidar com eventos inesperados devido à dificuldade inerente de prever todos os estados possíveis dentro de um sistema multiagente.

Diante disso, este trabalho verificou os modelos de PMA existentes e buscou abordar os problemas referentes à avaliação de desempenho e à combinação de estratégias de recuperação. Dessa forma, uma etapa crucial foi a realização de uma revisão de literatura abordando conceitos, técnicas e desafios do estado-da-arte. A revisão foi um processo conduzido com adoção de protocolos tradicionais e complementares para fornecer uma visão abrangente da área de pesquisa de PMA.

Assim sendo, trabalhos de pesquisa no campo de ambientes dinâmicos frequentemente propõem duas estratégias de recuperação de planos. A primeira estratégia é o replanejamento, que envolve retirar ações de um plano anterior e gerar um novo plano a partir do estado atual para alcançar o estado desejado [Gouidis et al., 2018, Komenda et al., 2012, 2014]. A segunda estratégia é a reparação, que visa reutilizar ações de um plano anterior para restaurar uma condição prevista e desejável [Cashmore et al., 2019, Komenda et al., 2012, 2014, Mohalik et al., 2018]. Embora a estratégia de reparo geralmente exija menos tempo de planejamento em comparação com o replanejamento, a abordagem de replanejamento tende a gerar planos com um número menor de ações do que o reparo [Babli et al., 2023].

Neste ponto, dois problemas podem ser identificados. Em primeiro lugar, os estudos de MAP normalmente analisam valores médios de tempo de planejamento, número de ações e ocorrências de falhas. Portanto, muitas vezes carecem de uma avaliação abrangente que inclua fatores como desvio padrão, nível de confiança dos resultados, e métodos estatísticos de avaliação de estratégias de recuperação. Basear-se apenas em valores médios pode levar a conclusões prematuras sem considerar a distribuição geral dos dados. Além disso, apesar das vantagens e desvantagens associadas a cada estratégia, existe uma lacuna de pesquisa na literatura sobre a integração de técnicas de replanejamento e reparo em modelos PMA. Assim, um segundo problema identificado é que os modelos apresentados não que combinam replanejamento e reparo de maneira coesa e unificada.

Como soluções, um método de avaliação de estratégias de recuperação em ambientes dinâmicos foi proposto para resolver o primeiro problema. Além disso, um processo de recuperação de planos que combina replanejamento e reparo também foi desenhado para ser uma solução para o segundo problema apresentado.

Sendo assim, o objetivo deste trabalho foi apresentar uma análise abrangente das estratégias de recuperação de planos em ambientes dinâmicos para propor um modelo que explore as duas opções de maneira complementar. Logo, esse propósito foi dividido em dois objetivos específicos. O primeiro foi avaliar o desempenho e definir as características de reparação e replanejamento utilizando técnicas estatísticas de teste de hipótese e correlação. Em seguida, combinar as duas estratégias em um único modelo de tal maneira que suas potencialidades sejam exploradas.

Para explorar a lacuna referente ao método de avaliação, essa tese apresenta uma análise mais ampla sobre o desempenho das estratégias de recuperação de planos em ambientes dinâmicos. De maneira diferente dos trabalhos correlatos, foi apresentada uma verificação da correlação entre as métricas que influenciam o rendimento destas estratégias. Assim, como um dos resultados obtidos, destaca-se a análise de que o nível de acomplamento, caracterizado pelo nível de ações públicas, é um fator influenciador das correlações entre as métricas.

Além disso, a lacuna referente à combinação entre as estratégias de recuperação foi trabalhada na prosposta de modelo apresentado. Nesse quesito, o modelo proposto explora o uso de reparação de maneira individual pelos agentes, reservando o uso do replanejamento para uma fase posterior e centralizada, acionada caso a tentativa anterior não tenha tido sucesso. Com isso, as características das estratégias são exploradas de forma complementar, ou seja, a rapidez da reparação com a qualidade e robustez do replanejamento. Cabe também destacar que esse modelo minimiza a troca de mensagens em situações em que uma solução local é viável, o que é particularmente importante em ambientes complexos ou quando os agentes precisam deliberar rapidamente sobre soluções. Outrossim, os resultados forneceram condições para concluir que o número de objetivos e as falhas afetam a execução em diferentes condições. Por fim, os resultados apontam que o desenho do processo de recuperação do plano é adequado para vários cenários, destacando que a reparação deve ser explorada de forma descentralizada enquanto o replanejamento é feito de forma centralizada.

Em resumo, esta tese apresenta e avalia um modelo de PMA aplicável a ambientes dinâmicos, que combina a estratégia de replanejamento e de reparo. Cabe destacar algumas contribuições para a área de pesquisa em PMA. Em primeiro lugar, ressalta-se um método para comparar o desempenho das estratégias de recuperação de planos em diferentes cenários, considerando tanto o planejamento centralizado quanto o distribuído com níveis variados de acoplamento. Ademais, a revisão da literatura traz outra contribuição ao fornecer uma visão abrangente da área de pesquisa de PMA. Ao examinar uma ampla gama de trabalhos acadêmicos, a revisão viabilizou identificar autores e organizações que fizeram contribuições substanciais para o campo de estudo, bem como os documentos mais citados.

Palavras-chave: Ambientes dinâmicos, Planejamento Multiagente, Sistemas Multiagentes, Planejamento Automatizado, Estratégias de Recuperação de Planejamento, Replanejamento, Reparo

Contents

List of Figures

List of Tables

List of Acronyms

AMR Autonomous Mobile Robot.

CoDMAP Competition of Distributed and Multi-agent Planners.

DMAP Distributed and Multi-agent Planning.

ICAPS International Conference on Automated Planning and Scheduling.

IPC International Planning Competition.

JSON JavaScript Object Notation.

MA-PDDL Multi-Agent Planning Domain Definition Language.

MAP Multi-Agent Planning.

MAS Multi-Agent System.

PDDL Planning Domain Definition Language.

SAP Single-Agent Planning.

Chapter 1

Introduction

Planning is the deliberation process of an agent (human or robot) when choosing actions to change the environment to a desirable condition. The planning process focuses on choosing and organizing actions through their expected effects. Likewise, planning depends on coordinating the selected actions to provide the correct order and timing, allowing cooperation and avoiding competition among agents. Moreover, an agent is not only an executor it must integrate planning and reasoning throughout the execution process [Ghallab et al., 2014].

In this context, classical planning is the research branch of Artificial Intelligence that aims to create algorithms and techniques to solve planning problems in deterministic and fully observable environments. It involves generating a sequence of actions that can transform an initial state into a desired goal state while adhering to predetermined rules or constraints. The world is often represented using logical or propositional language, and the planning problem is approached as a search problem that explores possible actions and states using various search algorithms. Multi-Agent Planning (MAP), on the other hand, expands classical planning to include scenarios where multiple autonomous agents are involved. The goal of MAP is to generate plans that coordinate the actions of multiple agents to achieve collective or individual goals, taking into account interactions, dependencies, conflicts, and cooperation.

Also, MAP is a computer science research area integrating planning with Multi-Agent System (MAS). While planning and scheduling is a mature research area of Artificial Intelligence generally focusing on single-agent tasks, MAS involves intelligent agents working cooperatively or competitively to achieve individual or group goals. MAP leverages MAS with practical reasoning agents focusing on means-end decisions with planning capabilities. MAP computational problem has exponential time complexity depending on the number of agents involved in the coordination process, considered NP-hard by some authors [Backstrom, 1998, Kalmár-Nagy et al., 2017].

Overall, MAP builds upon classical planning by extending it to scenarios with multiple agents, considering coordination constraints, incorporating communication and negotiation, and enabling distributed planning.

Furthermore, techniques and models of MAP can be applied in different areas, from an environment with a high level of control to a scenario where uncertainty plays an important role. Thus, research in MAP real-world problems has increasing interest ranging from autonomous driving, logistics and warehouse automation, environmental monitoring, multi-robot systems, and many domains where solutions towards intelligent agents are vital. In such applications, we highlight two key points. First, agents depend on or affect others under different levels regarding the interaction their action executions induce. Second, agents must react dynamically to system state and environment changes.

1.1 Motivation

This thesis associates two complementary research areas. On the one hand, MAS tackles the issues about agents' knowledge, beliefs, and capabilities. As agents coexist in a shared environment, they require communication and coordination processes. In this sense, they employ deliberation functions to interact with the environment. On the other hand, MAP can instantiate one of these functions as planning. Beyond generating a sequence of actions for a group of agents to achieve a specific objective, MAP involves coordinating multiple agents' actions to achieve public or private goals while considering the constraints, dependencies, and interactions among the agents.

Moreover, MAS and MAP typically concern reasoning about the agents' knowledge, goals, capabilities, and environment dynamics. However, we consider MAP can pull MAS when the environment is uncertain and execution, planning, and coordination must be carried out several times to overcome failures. Therefore, MAP can provide strategies that allow agents to achieve their goals in a coordinated manner. Thus, the coordination process involves cooperation among agents with planning capacity that can be distributed or centralized.

An agent's capabilities define its autonomy level. An agent can keep track of its state, allowing it to identify system failures. Additionally, agents can communicate and negotiate with other agents to effectively adapt to changes in the operating environment. This autonomy level is the foundation for a continuous decision-making process that responds dynamically to changes and enables the seamless pursuit of individual and collective goals.

Cooperation is essential for organizing agents within a system, but its importance extends beyond that. It can also lead to synergy among agents, resulting in favorable outcomes. Cooperation allows agents to help each other when needed and avoid interference when coordinating individual activities. In both cases, competition among agents can be reduced, allowing them to work together towards a common goal.

Several factors influence the decision-making process and the complexity of execution within a MAS. The number and capabilities of the resources involved, the operational conditions, and the strategic procedures are some aspects that significantly impact the planning, execution, and coordination phases. In hierarchical organizations such as enterprises or military units, a centralized approach is adopted for operational direction to optimize resource utilization. Conversely, execution may be decentralized to enable simultaneous actions. Whenever possible, task execution should prioritize the timely and modular deployment of capabilities, avoiding action duplication and dispersion of resources, thus ensuring effectiveness and minimizing operational and logistical costs.

Cooperation among agents in a system is vital for organizing them and achieving synergy. By eliminating competition and fostering a collaborative environment, cooperation enables agents to work towards a common goal. To ensure efficient teamwork performance, a systematic approach involving goal evaluation, agent organization, planning, execution, and environmental monitoring is crucial. The agents' hierarchical organization with adaptable autonomy levels adds complexity to interactions. Factors such as resource allocation, capabilities, operational conditions, and strategic procedures significantly influence decision-making and execution complexity. Computational tools, particularly in Artificial Intelligence, support decision-makers in evaluating problems, selecting agents, and decentralizing tasks. By leveraging these tools, the effectiveness and coordination of MAS can improve [Cil and Mala, 2010, Dunin-Keplicz and Verbrugge, 2011, Wooldridge, 2009].

1.2 Problem

The planning process faces the challenge of evaluating available capabilities and resources, which can lead to a combinatorial explosion. Computational aid is essential to choose elements and delegate tasks effectively. Artificial Intelligence research areas, such as Automated Planning and MAP, offer practical approaches to support deliberation throughout efficient teamwork performance. However, in scenarios where agents lack complete control over the environment, planning, coordination, and execution become challenging. Studies classified as MAP assume a fully observable, deterministic environment without uncertainty regarding the effects of actions, limiting their applicability in dynamic environments where agents must handle failures. Further investigation is needed to address these issues and enhance MAP models to evaluate problems, select agents, and decentralize tasks.

At this point, two problems can be identified. First, MAP studies typically analyze mean values of planning time, number of actions, and failure occurrences. Therefore, they often lack a comprehensive assessment that includes factors such as standard deviation, level of confidence in the results, and statistical methods of evaluating recovery strategies. Relying solely on mean values can lead to premature conclusions without considering the overall distribution of the data. A missing analysis in the literature refers to the correlation between the metrics that influence the performance of these strategies. In other words, it remains to discuss how the level of coupling, characterized by the level of public actions, influences the correlations between the metrics.

Furthermore, despite the advantages and disadvantages associated with each strategy, there is a research gap in the literature on integrating replanning and repair techniques in MAP models. Thus, a second problem identified is that the models presented do not combine replanning and repair in a cohesive and unified manner.

1.3 Methodology

This work acknowledges the existing MAP models and aims to address the identified gap in performance evaluation and the combination of recovery strategies. It is necessary to include updates in the related work to ensure a complete analysis and a more robust model for dynamic environments.

To start our research on MAP, we conducted a thorough literature review. This step allowed us to gather information from multiple sources and better understand the current state-of-the-art in the field. The literature review included traditional and complementary methods to ensure a comprehensive and reliable summary of MAP research. The analysis allowed identifying key researchers and institutions making significant contributions to the field.

We looked into strategies for recovering from a plan and found that letting agents make their repairs works better if agents interact less. We improved our research methodology by designing a process, running simulations, and analyzing the results to test this hypothesis.

Our goal with this methodology was to collect enough evidence to determine whether to accept or reject the research hypothesis. We analyzed the effectiveness of combining replanning and repairing strategies to gain insights that could enhance the performance of MAP models.

Therefore, the research questions under study in this thesis are:

1. RQ1: How do replanning and repairing strategies perform when affected by failures in dynamic MAP environments? (see Section 4.3.1)

2. RQ2: How are environment variables related to the performance metrics? (see Section 4.3.2)
3. RQ3: How can replanning and repairing be combined to enhance the performance of MAP models? (see Section 5.3.3)

1.4 Objectives

The main goal of this thesis is to present a comprehensive analysis of plan recovery strategies in dynamic environments proposing a model that explores replan and repair in a complementary way. The main goal is divided into two specific objectives.

Firstly, we assess the performance evaluation of the recovery strategies and highlight their characteristics. We check the correlation between the metrics that influence the performance of these strategies. Thus, we aim to analyze how the coupling level, characterized by the level of public actions, influences the correlations between the metrics.

Secondly, the two strategies are integrated into a single model to maximize their potential. With this, our goal is to explore the strategies' characteristics in a complementary way, the speed of repair with the quality and robustness of the replanning.

This thesis aims to evaluate the combination of replanning and repairing in a MAP model applicable to dynamic environments.

1.5 Contribution

This thesis makes some contributions, including a thorough investigation of existing research in the MAP field, a rigorous analysis of the effectiveness of replanning and repairing strategies, an exploration of the impact of environmental variables on performance metrics, and the development of a dynamic MAP model that combines replanning and repairing techniques.

Publications

Lastly, this work has also produced a series of scientific publications as the references listed in the sequence.

Journal articles

- Leonardo Henrique Moreira, Célia Ghedini Ralha. An Efficient Lightweight Coordination Model to Multi-agent Planning. *Knowledge and Information Systems* 64, 415-439 (2022). https://doi.org/10.1007/s10115-021-01638-5

- Leonardo Henrique Moreira, Célia Ghedini Ralha. Method for Evaluating Plan Recovery Strategies in Dynamic Multi-agent Environments. *Journal of Experimental & Theoretical Artificial Intelligence*, 1-25 (2022). https://doi.org/10.1080/0952813X.2022.2078887.

Conference papers

- Leonardo Henrique Moreira and Célia Ghedini Ralha. Evaluation of Decision-making Strategies for Robots in Intralogistics Problems Using Multi-agent Planning. *In Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, Kraków, Poland, 2021, pages 1272-1279, doi: 10.1109/CEC45853.2021.9504887.
- Leonardo Henrique Moreira and Célia Ghedini Ralha. Plan Recovery Process in Multi-agent Dynamic Environments. *In Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2021, pages 187-194, doi: 10.5220/0010559301870194.
- Leonardo Henrique Moreira, Célia Ghedini Ralha. A Multi-agent Planning Model Applied to Teamwork Management. *In Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro, Brazil, 2018, pages 1-8, doi: 10.1109/CEC.2018.8477856.

1.6 Thesis Outline

The structure of this document follows a format where each chapter addresses a specific objective within the scope of the subject assessed, including works published during the development of the thesis.

Regarding the literature review, it is customary in journal and conference papers to include a section highlighting relevant works related to the research topic. In alignment with this convention, such sections are in the following chapters focusing on the presentation of specific works published during the development of this thesis. However, Chapter 2 provides a comprehensive review of the MAP research area and its relevance to this thesis. This chapter offers an in-depth examination of the related work, providing a thorough understanding of the background and context for the research conducted in this thesis.

To ensure a concise presentation and avoid concept repetition, Chapter 3 provides a selection of the necessary background information. This chapter offers the foundational knowledge required to understand the subsequent chapters, setting the stage for the research analysis and findings.

Chapter 4 presents an empirical method for evaluating recovery strategies in dynamic environments affected by exogenous events. A pipeline of activities, including simulation, data preparation, and evaluation, supports the method. Statistical techniques, such as T-Test and correlation analysis, are employed to assess recovery strategy performance and analyze the relationships among environmental variables. Several case studies demonstrate that repairing offers faster results, while replanning generates superior plans. The coupling level among agents, indicated by the ratio of public actions, significantly impacts the relationship between variables. The findings indicate that repairing excels in time-sensitive scenarios like rescue operations, whereas replanning suits resource-constrained situations like robotics pathfinding. Both strategies are effective in loosely-coupled domains, while in tightly-connected environments, the choice depends on prioritized factors.

Chapter 5 focuses on proposing a plan recovery process for dynamic environments facing failures caused by external events. This process differentiates itself from existing approaches by combining replanning and repairing strategies to offer a staggered solution. It comprises three phases: local repair, asking for help, and replanning. A simulation tool was developed to evaluate the process, considering different levels of interaction among agents. The results indicate that agents' autonomy in local repair yields better outcomes in low-impact environments. Additionally, the study highlighted the impact of the coupling level among agents, as indicated by the ratio of public actions, on the recovery process's complexity and performance metrics, such as planning time, final plan length, and message exchange. This chapter contributes a novel plan recovery process and insights into the influence of interaction levels and agent coupling in dynamic environments.

At last, Chapter 6 presents the main points of this thesis, the contributions, and suggestions for future work. We highlight factors like resource number, capabilities, and coordination challenges influencing MAP and execution complexity. Also, classical MAP models fall short in dynamic environments with exogenous events. Thus, recovery strategies like replanning and repairing join into MAP models. Current proposals overlook statistical evaluation, requiring comprehensive analysis beyond average values for planning time, actions, and failures.

Chapter 2

Literature Review

A Meta-analytic Literature Review on Multi-Agent Planning. Long survey paper submitted on 17 May 2023 to the ACM Computing Surveys (under review).

This chapter presents a MAP literature review with works published from 2014 to 2023, available in Scopus and Web of Science repositories, aiming at a comprehensive overview of concepts, techniques, and challenges. The adopted literature review method involves synthesizing multiple meta-analyses on MAP topic [Mariano and Rocha, 2017]. The review method provides a comprehensive and robust summary of available evidence complementary to other more traditional systematic literature review protocols [Kitchenham, 2004, Kitchenham and Charters, 2007].

2.1 Review Method

The meta-analytic literature review method includes three main stages: research preparation, data presentation with interrelation, and detailing, integrating model and evidence-based validation. The first stage incorporates the survey protocol of the research preparation with the definition of keywords, space-time, and digital repositories. In the second stage, bibliometric laws are applied to information extracted from repositories to analyze the relationships among data. The most historically cited articles are analyzed to depict the evolution of contributions to the theme.

Finally, in the third stage, when the first impressions on the subject are built, deeper analyzes are needed to allow a better understanding of it, as well as selecting the most relevant authors, the main approaches, lines of research, validation via evidence, and delivery of the integrative model through comparison results from different sources. In this sense, integrating and validating models are applied to the evidence obtained from the citation, bibliographic coupling, and co-occurrence mapping study based on bibliometric laws [Mariano et al., 2019]. In the sequence, the three stages are detailed concerning the MAP literature review.

2.1.1 Research Preparation

The first stage of the literature review method aims to answer four questions to define the survey protocol. The answers to the research preparation questions are presented in Table 2.1.

1. What are the strings, keywords, or descriptors to be used in the search?

   The definition of the string is important because it is a factor that can change the results. The library digital repositories usually have the relationship options OR and AND, facilitating the combinations. For compound words, use quotation marks. Furthermore, choosing a string that is neither too permissive, to the point of filtering out unrelated works nor too restrictive, to eliminate interesting papers is important.

2. In which digital peer-reviewed literature databases will the search be applied?

   The Web of Science and Scopus (WoS) databases are used for research since they are well-regarded in various academic communities. WoS is a research platform that covers a wide range of academic disciplines and hosts a massive collection of millions of papers. The papers in the database are connected through citations, allowing researchers to explore the web of scholarly literature across disciplines [Clarivate, 2023]. Scopus is a comprehensive research database and search engine that indexes and provides access to scientific, technical, medical, and social science literature. Scopus offers researchers and professionals access to reliable, up-to-date information and tools to analyze and track research output, citations, and author impact. It covers above 80 million publications, including peer-reviewed journals, conference proceedings, book series, and trade publications. The database also includes a range of metrics and tools to evaluate research performance and monitor trends in various fields [Elsevier, 2023]. A third option is the Google Scholar base, but it lacks standardization of paper titles and data structuring making it different from the previous ones.

3. What is the space-time field of the survey?

The survey space-time is important to observe whether the chosen bases have the same historical coverage. Thus, the same temporal space must be used in all databases. Currently, studies tend to perform searches over 5 to 10 years.

4. What are the knowledge areas involved in the review?

   Regarding the knowledge areas, the researcher must read a sample of the titles of the search results since the reported words may not adhere to their area of interest. It is advisable to use the filter according to the desired research area, allowing the contents to be assertive as possible.

Table (2.1) Research preparation questions and answers.

	Question	Answer	Observation
1.	What are the strings, keywords, or descriptors to be used in the search?	"Multi-Agent Planning"	The chosen descriptor was very generic to avoid data loss. Thus, terms that specialized the search in specific branches, such as classical or probabilistic planning, were not used.
2.	In which digital peer-reviewed literature databases will the search be applied?	WoS and Scopus	Both databases have appropriate temporal coverage and quality scope to address well-cited works. Also, the databases are complementary and regularly used in other surveys.
3.	What is the space-time field of the survey?	2014 - 2023	The lower limit was defined from an article, identified in previous research. Furthermore, the period was defined not to exceed the limit of 10 years, but also not to restrict it only to the most recent works.
4.	What are the knowledge areas involved in the review?	No filter was applied	The results were categorized into areas of interest, such as Computer Science, Artificial Intelligence, and Robotics. Thus, there was no need for filtering.

2.1.2 Data Presentation and Interrelation

In the second stage of the meta-analytic method, important data is extracted from the results. The goal is to provide an initial overview of the research area regarding relevant aspects, such as authors, citations, publication issues, and organizations. Thereby, the following conclusions can be drawn from the works filtered in the databases:

1. Journals and conference with the largest number of documents;

2. Publications rate in journals and conferences across space-time;

3. Most cited authors and those with the largest number of publications;

4. The countries that published the most;

5. Organizations that published the most;

6. Knowledge areas with the highest number of publications;

7. Keyword frequency.

The meta-analytic method is more than a simple summary of metrics. The approach suggests a quantitative investigation employing three laws to highlight bibliometric aspects. Bradford's law is a bibliometric principle that describes the distribution of scientific literature in various disciplines. The author noted that publications in particular fields tend to cluster around a small number of core journals. According to Bradford's Law, scientific literature in a given field can be divided into three zones. The first zone consists of a few core journals that contain a significant portion of the most relevant articles in the field. The second zone has a larger number of journals related to the core journals but with less impact. The third zone presents many peripheral journals that are only marginally relevant to the field [Alabi, 1979].

Price's law states that a small minority of researchers in a given field contribute the majority of publications, while the majority of researchers contribute relatively little. Price's law states that approximately half of the publications in a given field come from the square root of the total number of researchers in that field. Therefore, this law aims to reveal the most important authors considering the citation level.

The 80/20 law, also known as the Pareto Principle or the law of the vital few, is a general principle that states that roughly 80% of the effects come from 20% of the causes. So, it is observed that a small fraction of highly cited papers or authors are responsible for a large proportion of the total citations in a field, while the majority of papers or authors have relatively few citations.

We applied two extra impact factors to enhance the quantitative analysis. The h-index and SCImago Journal Rank (SJR) to evaluate the research impact of researchers and journals. The h-index attempts to capture both the quantity and quality of a researcher's work since it rewards researchers who have published a large number of highly cited papers. The SJR assigns a score to each journal based on the number and quality of citations that the journal's articles receive.

In this context, the cited laws are used to provide conclusions drawn in the second stage of data presentation and interrelation as presented in Table 2.2.

Table (2.2) Aspect for an overview of the research area.

	Conclusion	Laws or principles
1.	Journals and conference with the largest number of documents	Bradford's law, 80/20 law and impact factors
2.	Publications rate in journals and conferences across space-time	Number of documents
3.	Most cited authors and those with largest number of publications	Bradford's law, Prince's law, 80/20 law and citations
4.	The countries that published the most	80/20 law
5.	Organizations that published the most	80/20 law
6.	Knowledge areas with the highest number of publications	80/20 law
7.	Keyword frequency	80/20 law

2.1.3 Detailing, Integrating Model and Evidence-based Validation

Once conclusions are built in the second stage, deeper analysis is needed to allow a better understanding of the review topic. This stage is concerned with selecting the most important authors, the main approaches, lines of research, validation via evidence, and delivery of the integrative model by comparing results from different sources.

To demonstrate the evidence, new bibliometric indices are needed to detect couplings, identifying the relationships between authors, references, and countries in the literature. The co-citation analysis verifies the articles regularly cited together, which may suggest a similarity between these studies. On the other hand, the bibliographic coupling method is based on the premise that articles that cite similar works have an affinity.

Thus, co-citation and bibliographic coupling differ in the level of analysis. While a co-citation is a relationship of similarity between two cited publications, bibliographic coupling is a measure of association between two cited publications. Therefore, bibliographic coupling brings a perspective of research fronts and co-citation of the most used approaches [Vogel and Güttel, 2013].

2.2 Results and Analysis

The Scopus and WoS databases are used to search the works using the definitions of Table 2.1. Exported data included all available fields, highlighting author, title, abstract, keywords, affiliation, and citations. Figure 2.1 presents the distribution of documents returned from each database. The raw data describing the query results are available in `https://tinyurl.com/3czm9naf`.

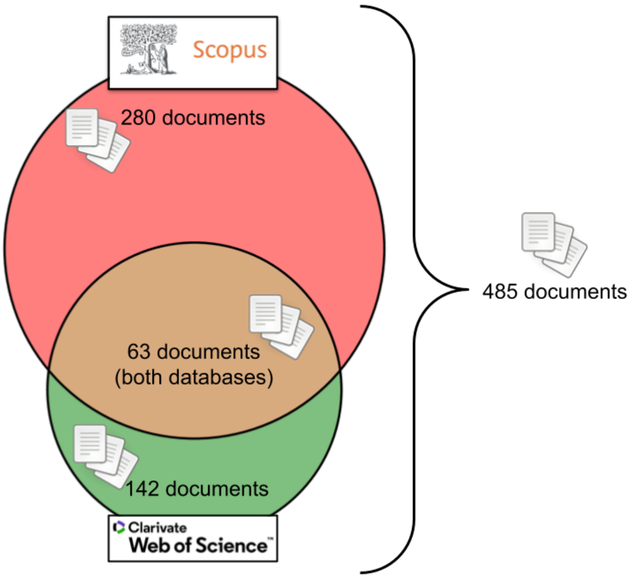


Figure (2.1) Merge of documents obtained from Scopus and WoS databases.

The records from the Scopus and WoS databases were combined to present a more comprehensive literature review. There are 63 documents existing in both databases. Therefore, this review considered 485 documents, with 343 from the Scopus database and 205 from WoS. Table 2.3 presents the list of documents considering the raw document types used in both databases.

Table (2.3) Document types in databases.

Type	Scopus	WoS	Merged
Conference Paper	220	0	220
Proceedings Paper	0	127	127
Article	102	71	110
Conference Review	13	0	13
Book Chapter	5	0	5
Article; Proceedings Paper	0	4	4
Review	1	2	3
ACM Int. Conf. Proc. Ser.	1	0	1
Article; Early Access	0	1	1
Lect. Notes Comput. Sci.	1	0	1
Total	343	205	485

Since data from distinct databases were combined, the difference in the standardization of terms was perceived, proving to be an obstacle to be overcome. Thus, to group the types presented on the Scopus and the WoS databases, some required changes were made to combine the items as presented in Table 2.4.

Table (2.4) Types standardization.

Original Type	Converted Type
Review Article; Early Access Review	Journal Article
Conference Review ACM Int. Conf. Proc. Ser. Lecture Notes In Computer Science Proceedings Paper Article; Proceedings Paper	Conference Paper

After the document types standardization, data listed in Table 2.3 was updated to Table 2.5. In both tables, the values presented in the columns Scopus and WoS represent the number of documents considering repetitions. Thus, the Merged column does not indicate the sum of documents in both databases, but the set union amount.

Table (2.5) Document types in databases.

Type	Scopus	WoS	Merged
Conference Paper	235	131	366
Journal Article	103	74	114
Book Chapter	5	0	5
Total	343	205	485

Thus, an initial overview is designed from the values detailed in Table 2.5. On Scopus, 68,5% of the documents are conference papers, and 30% are journal articles. Similarly, on the WoS, 63,9%, and 36%, respectively. Considering the results obtained from the databases merged, the conference documents have an increased value, namely, 75,5%, while the journal articles achieve 23,5%. From the above, we note that most MAP research works are published at conferences, presenting a ratio of approximately three papers from conferences to journal articles.

Regarding only the journals, a third of the documents are concentrated in 10 journals, considering the merged number of articles. In other words, 10 of the 81 journals retrieved from the survey were responsible for publishing 38 of the 114 articles. Therefore, this behavior is justified by Bradford's law because of the cluster formation around the first ten journals. The list of the journals with the largest number of documents is detailed in Table 2.6, where the rows were ranked considering the merged values. The h-index and SJR values presented were retrieved from the Scimago and h-index sites.[1][2] Also, it is

[1] `https://www.scimagojr.com/`

[2] `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=en`

important to highlight the ACM Computing Surveys in the top 10 list, where two MAP surveys were published [Rizk et al., 2019, Torreño et al., 2017].

Table (2.6) Journals with the largest number of documents.

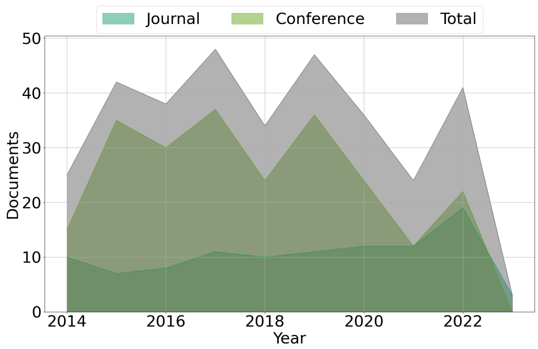
Journal	h-index	SJR	Scopus	WoS	Merged
Autonomous Agents and Multi-Agent Systems	72	0.81	6	6	7
Applied Sciences (Switzerland)	75	0.51	5	0	5
Knowledge and Information Systems	78	0.99	4	3	4
IEEE Robotics and Automation Letters	63	2.21	4	4	4
Artificial Intelligence	155	1.67	4	2	4
Robotics and Autonomous Systems	126	1.2	3	3	3
Journal of Artificial Intelligence Research	123	1.49	2	1	3
International Journal of Robotics Research	170	3.4	3	3	3
Applied Intelligence	72	1.21	3	3	3
ACM Computing Surveys	172	5.09	2	2	2

Furthermore, the analysis of the conferences also showed similar behavior. However, the aggregation rate of publications around a few items is even higher. Thus, 10 conferences were responsible for 160 documents, which represents 44% of the 366 results obtained. Again, Bradford's law can explain the presented behavior. The conferences were ranked considering the merged values and presented in Table 2.7.

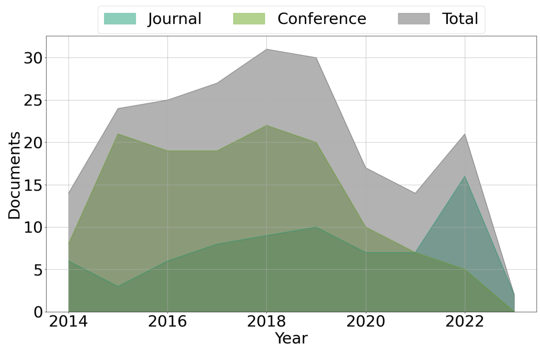
Table (2.7) Conferences with the largest number of documents.

Journal	h-index	Scopus	WoS	Merged
International Conference on Autonomous Agents and Multi-agent Systems	50	40	24	64
International Conference on Automated Planning and Scheduling	32	20	13	28
International Conference on Agents and Artificial Intelligence	19	13	7	15
AAAI Conference On Artificial Intelligence	180	9	9	14
International Joint Conference on Artificial Intelligence	120	9	2	9
International Conference on Intelligent Robots and Systems	128	6	3	6
Advances In Neural Information Processing Systems	278	4	0	4
European Conference on Artificial Intelligence	23	0	4	4
International Conference on Robotics and Automation	116	4	3	4
Intelligent Transportation Systems Conference	76	4	1	4

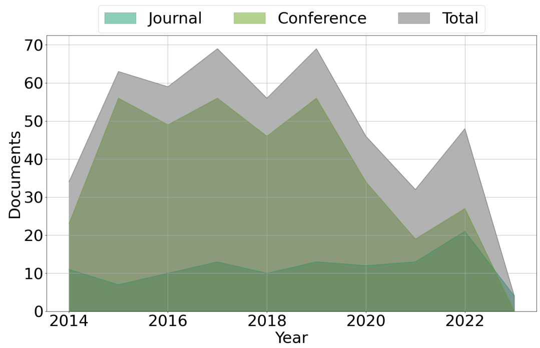
With respect to the publication rate in journals and conferences from 2014 to 2023, the number of documents in each database is illustrated in Figure 2.2. From 2014 to 2020, there was a protagonism of conference publications compared to journals. From 2021, this aspect began to change to the point that the volume of journal documents came near the level of conference papers. Also, considering only the records obtained from the WoS database (Figure 2.2 (b)), the number of journal articles exceeds the number of conference papers. A possible justification for this change in behavior is the effect caused by the COVID-19 pandemic since conferences require physical interaction between researchers. It is plausible to infer that the consequences of social distancing did not affect the publication process in journals.



(a) Scopus.



(b) Web of Science.



(c) Merged.

Figure (2.2) Evolution of publications in journal and conferences year.

The analysis considering the authors was carried out according to two aspects. At first, the authors are evaluated on the level of citations and considering the total number of publications. In both cases, the values refer to the merged database. Also, regarding the citations, the merged column stands for the maximum value between Scopus and WoS results.

From the 744 authors retrieved from the merged database, 17 had more than 100 citations, and 525 had at least one citation. However, 220 authors had no citations. As shown in Table 2.8, the most cited author considering Scopus and WoS was Onaindia, with 215 citations. In sequence, Torreño and Komenda were the most cited authors, with 193 and 186, respectively. Two sets of identical lines are in Table 2.8. The first case represents data from the research developed by Awad, Tunstel, and Rizk [Rizk et al., 2019]. This work was a survey cited 134 times in Scopus and 104 in WoS, positioning the authors among the most cited, even considering a single document. For the same reason, the lines referring to authors Banerjee and Kraemer are also repeated in Table 2.8 [Kraemer and Banerjee, 2016].

Table (2.8) Most cited authors.

	Scopus		Web of Science		Merged	
Authors	Docs	Citations	Docs	Citations	Docs	Citations
Onaindia E	12	215	10	154	12	215
Torreño A	8	193	7	138	8	193
Komenda A	29	186	16	135	31	186
Dimarogonas Dv	5	177	5	120	7	177
Awad M	1	134	1	104	1	134
Tunstel Ew	1	134	1	104	1	134
Rizk Y	1	134	1	104	1	134
Banerjee B	1	123	1	92	1	123
Kraemer L	1	123	1	92	1	123
Štolba M	17	122	14	114	19	122

From a second perspective of the number of documents published, only four authors listed in Table 2.8 continue to stand out in Table 2.9 that rank the ten authors with the largest number of documents. Note there are six new authors listed among the most active. Thus, in the merged database (column docs), only five authors published ten or more documents. The most relevant authors are Komenda with 31 documents, Štolba and ToźIăźKa most cited author, with 19 and 17, respectively.

Also, considering the complete set of data and not just those listed in Table 2.9, 555 authors had only one document published. According to the Prince's law, half of the documents come from the square root of the number of authors. In this sense, 27 researchers ($\sqrt{744}$) published 227 documents.

Table (2.9) Authors with the largest number of documents.

	Scopus		Web of Science		Merged	
Authors	Dos	Citations	Docs	Citations	Docs	Citations
Komenda A	29	186	16	135	31	186
Štolba M	17	122	14	114	19	122
ToźIăźKa J	15	82	6	35	17	82
Onaindia E	12	215	10	154	12	215
Shani G	9	44	8	29	10	44
Jakubuv J	9	46	1	11	9	46
Niyogi R	9	30	6	21	9	30
Torreño A	8	193	7	138	8	193
Ralha C	8	11	8	9	8	11
Zilberstein S	7	46	4	9	8	46

Considering the number of citations and publications from 2014 to 2023 in Scopus and Wos databases, we note that contributions in the MAP area are limited to a small group of authors. From the union of the names listed in Tables 2.8 and 2.9, 16 authors are presented, either by citation criterion or by the number of documents. On the other hand, if the relationship provided by the 80/20 law has not been numerically respected, we note by the studied sample that a small number of authors are very effective. Thus, one can see that, within the analysis range, the MAP research community had few very active authors and a large number of researchers with a low level of work or citations.

The number of documents and citations by country is computed from the authors' affiliation in the Scopus and WoS databases. Thus, the document count per country considers the number of countries obtained from the authors' affiliation with authors from the same country counted once. On the other hand, a single document can account for different countries if its authors are from different locations. The visual distribution of countries with the highest number of published documents is shown in Figure 2.3. The United States of America was responsible for a third of the documents. It should be noted that Shlomo Zilberstein (10^{th} author in Table 2.9) accounts for 5% of the American documents. The Czech Republic and China place in second and third place.

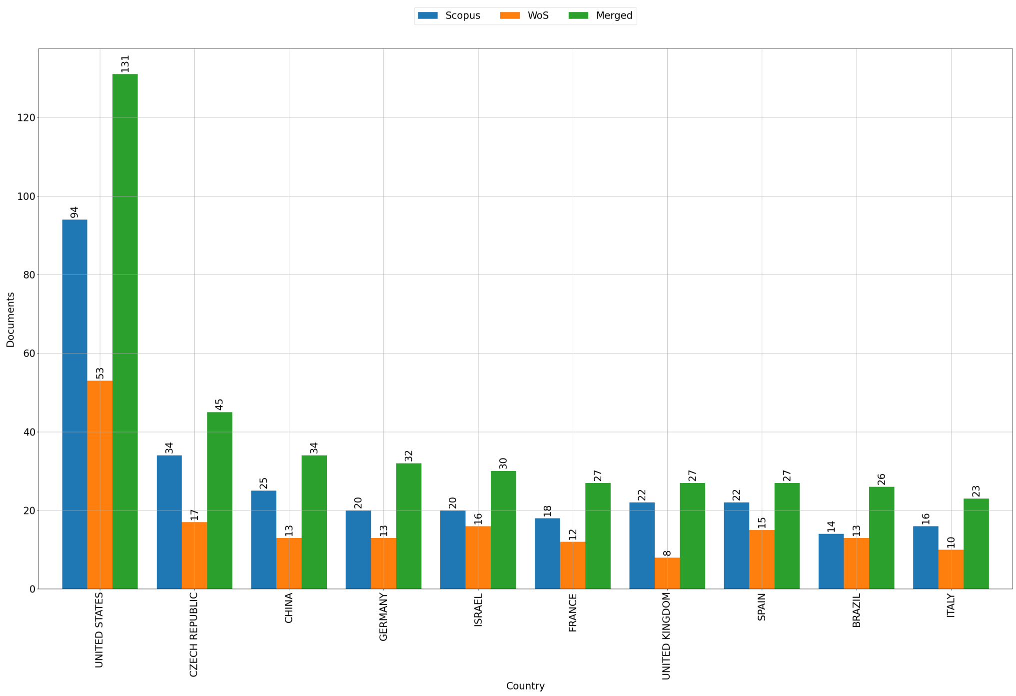


Figure (2.3) Documents by country in WoS, Scopus, and merged databases (Top 10).

According to the number of citations in Figure 2.4, the most visible change is the shift of Spain, from the eighth position considering the number of documents (Table 2.3) to the third place of citations. This change happens due to the coverage of the research conducted by Onaindia and Torreño from Spain. The Czech Republic remained in the second position, sustained mainly by Komenda. Note that the United States of America maintains its first place. One of the reasons for this leadership is the fact that there are authors affiliated with American organizations with works with a high number of citations. For example, Tunstel, Kraemer, and Banerjee (6^{th}, 8^{th} and 9^{th} authors in Table 2.8) stand out with 134, 123, and 123 citations, respectively. These values represent more than 30% of 1165 citations related to the United States of America.

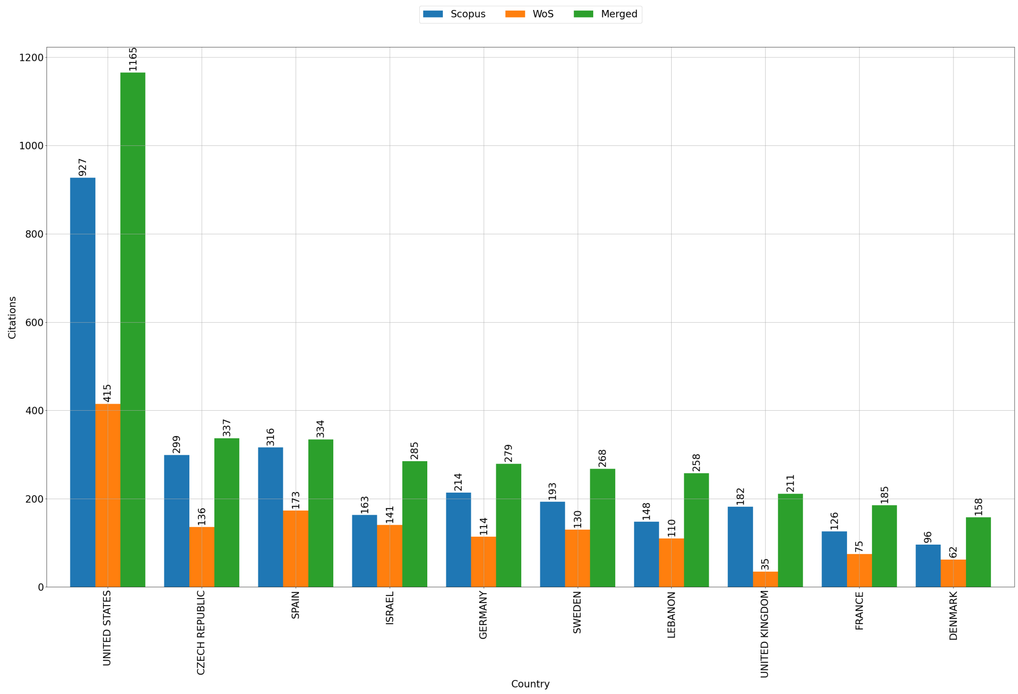


Figure (2.4) Citations by country in WoS, Scopus, and merged databases (Top 10).

We conducted a similar analysis of the number of documents and citations concerning the authors' organizations publishing in the MAP area. The data gathered from the authors' affiliations had no restriction concerning the number of publications or citations. An organization with a single document but well cited can be highlighted. Indeed, this justifies the presence of the United Technologies Research Center in the list due to the research work published by [Rizk et al., 2019]. The lists of the organizations with a mass number of documents and most cited are depicted in Tables 2.10 and 2.11, respectively. It is important to note that Czech Technical University and Universitat Politecnica de Valencia remain in the first positions regardless of the criteria adopted. The other centers listed in Table 2.10, such as Singapore Management University, Carnegie Mellon University, and the Royal Institute of Technology stand for important MAP research centers.

The WoS database was used to identify the top ten knowledge areas with the greatest quantity of publications. In this analysis, the Scopus database was not used because there was no reference to the areas in the exported data. As shown in Figure 2.5, Computer Science covered more than half of the documents surveyed (60.93%). The leadership of this area was due to the relationship of MAP to important Computer Science areas such as Artificial Intelligence (30.21%), Theory and Methods (13.28%), Interdisciplinary Applications (9.11%), Information Systems (5.21%), and Software Engineering (3.12%). In addition, MAP relates to areas such as Engineering, Electrical, and Electronic (11.98%), Robotics (11.72%), Automation and Control Systems (10.16%), Engineering Multidisciplinary (2.86%), and Telecommunications (2.34%).

Table (2.10) Organizations with the largest number of documents.

Organization	Country	Scopus Docs	Scopus Cit.	WoS Docs	WoS Cit.	Merged Docs	Merged Cit.
Czech Technical University	Czech Republic	32	289	16	135	42	326
Ben Gurion University	Israel	21	85	12	113	29	192
Universitat Politecnica de Valencia	Spain	24	358	11	157	27	372
Carnegie Mellon University	USA	19	247	5	9	24	256
University of Michigan	USA	13	64	4	26	17	90
Singapore Management University	Singapore	14	254	2	9	16	263
Delft University of Technology	Netherlands	13	85	6	26	16	93
University of Melbourne	Australia	13	40	2	0	15	40
New Mexico State University	USA	10	63	5	16	14	78
Royal Institute of Technology	Sweden	7	179	7	120	13	250

Table (2.11) Most cited organizations.

Organization	Country	Scopus Docs	Scopus Cit.	WoS Docs	WoS Cit.	Merged Docs	Merged Cit.
Universitat Politecnica de Valencia	Spain	24	358	11	157	27	372
Czech Technical University	Czech Republic	32	289	16	135	42	326
Singapore Management University	Singapore	14	254	2	9	16	263
Carnegie Mellon University	USA	19	247	5	9	24	256
Royal Institute of Technology	Sweden	7	179	7	120	13	250
American University of Beirut	Lebanon	1	134	2	110	3	244
University of Southern California	USA	10	238	1	2	11	240
University of Liverpool	United Kingdom	6	189	2	9	8	198
Ben Gurion University	Israel	21	85	12	113	29	192
United Technologies Research Center	USA	1	134	0	0	1	134

Another aspect analyzed to build the MAP overview area is the definition of the keyword frequency with the adoption of two strategies. First, we evaluated the keyword information exported from columns in the Scopus and WoS reports. Then, we considered the document title and abstract of the records gathered in the merged database. Table 2.12 lists the frequency of most cited words or expressions. Figure 2.6 presents the word clouds with terms such as multi-agent (and variations), planning, decision-making, intelligent agents, and robots. The listed words are strongly related to the string used in the search procedure.

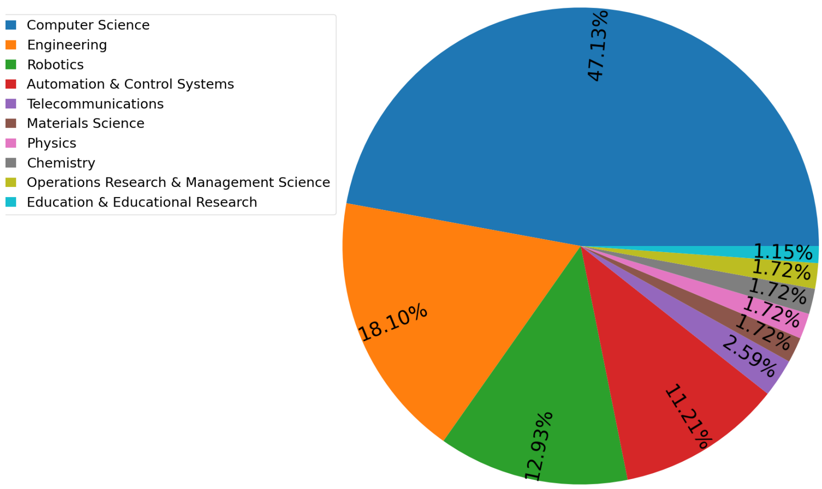


Figure (2.5) Distribution of documents by knowledge area.

Table (2.12) Most cited keywords.

	From keyword field		From title and abstract	
	Keyword	Occurrences	Keyword	Occurrences
1.	multi-agent	735	multi-agent	715
2.	agent system	311	problem	433
3.	agent planning	310	agent planning	388
4.	planning	123	plan	331
5.	planning multi	94	agent	329
6.	system	93	model	284
7.	decision making	75	approach	281
8.	algorithm	73	algorithm	273
9.	intelligent agents	68	action	252
10.	robot	67	planning	216

(a) From keyword field (b) From title and abstract

Figure (2.6) Wordcloud considering document keywords, titles, and abstracts in the WoS and Scopus merged databases.

2.2.1 Co-citation and Bibliographic Coupling Analysis

In this section, we present the results of the third stage of the meta-analytic review method focusing on detailing, integrating model, and evidence-based validation with the co-citation and bibliographic coupling analysis. The co-citation, coupling, co-authorship, and co-occurrence analyses used the free VOSViewer software[3] to create, visualize and explore heat maps based on data networks.

Co-citation Analysis

Figure 2.7 presents the co-citation analysis where authors are evaluated indicating the similarity between their works and research lines. The co-citation chart is built from the times that the works are cited together. In this way, the chosen time window, 2014 to 2023, is respected for the retrieval of articles from the Scopus and WoS database, not being applied to the references that these records can cite. This fact explains the presence of works before 2014 in the figure. Furthermore, it is interesting to mention that works before this time frame represent important research sources in the area. Note the map generated from the co-citation analysis presents three clusters. Among the works that make up each cluster, those within the time window are detailed and the most important works. Each cluster of information is presented in sequence.

Cluster 1 - green color

This cluster involves twelve works, of which only four were published after 2014 as presented in sequence. The authors in Nissim and Brafman [2014] discussed classical planning for multiple cooperative agents to keep their private information and capabilities concealed. The authors proposed the reduction of distributed constraint satisfaction and

[3] `http://www.vosviewer.com/`

partial-order planning techniques. The paper addressed whether a distributed heuristic forward search algorithm can be developed for privacy-preserving classical multi-agent planning. Briefly, the answer is a general approach to distributed state-space search where each agent only expands the part of the state relevant to it. Also, one variant of this approach is a distributed version of the A^* algorithm, which is the first cost-optimal distributed algorithm for privacy-preserving planning.

Figure (2.7) Co-citation clusters under density analysis.

In Maliah et al. [2014], the authors propose a method for identifying landmarks in a privacy-preserving distributed setting within the MA-STRIPS, a multi-agent planning definition language. The agents work together to identify sound landmarks without disclosing their private actions or objectives. Additionally, the MA-STRIPS planner was introduced employing these landmarks.

The work of Brafman [2015] promotes cooperation among diverse agents protecting private and proprietary information. The paper introduces an improved version of the distributed forward-search planning framework developed by Nissim and Brafman [2014] that reveals less information than the original algorithm. Also, the authors formalize the first formal proof and discussion of privacy guarantees for distributed planning and search algorithms. Thus, this work contributes to the privacy-preserving planning algorithms discussion and analysis.

Relevant work in this cluster is the survey developed by Torreño et al. [2017]. The authors provided an overview of cooperative multi-agent planning, including its motivations, challenges, approaches, and applications. The paper reviews the relevant literature, discusses the key issues and techniques in cooperative MAP, and presents a taxonomy regarding agent distribution, computational process, plan synthesis schemes, communication mechanisms, heuristic search, and privacy preservation. However, the overview covers a restricted scope with works exclusively from the 2015 Competition of Distributed and Multi-Agent Planning. Also, the authors identified open research questions and future directions for research, such as integrating planning with execution, temporal planning, and MAP in dynamic environments.

The other seven works that form this cluster were published on dates before the lower limit of analysis of this research (2014) and, therefore, were not detailed [Bernstein et al., 2002, Brafman and Domshlak, 2013, Bylander, 1994, Gerkey and Mataric, 2004, Kocsis and Szepesvári, 2006, Kovács, 2012a, Richter and Westphal, 2010].

Cluster 2 - red color

This cluster is composed of twelve works, but only one was published after 2014. The work of Crosby et al. [2014] presents a transformation of the original problem into several smaller and simpler instances, using the strategy of delegation and distribution of objectives among agents. This transformation forces agents to choose joint actions associated with one subset of goals, with concurrency constraints being satisfied. The approach turns possible the problem solving using a standard single-agent planner.

The other works concentrate on classical planning. In this branch of planning, the purpose is generating a sequence of actions or plans that an agent can follow to achieve a goal in a known environment, assuming no uncertainty or incomplete information about the environment. In classical planning, the agent operates in a deterministic environment, where the effect of each action is known, and there is no need to reason about uncertainty or probability.

The other eleven works that form this cluster were not detailed because of their publication dates before 2014 [Borrajo et al., 2013, Boutilier and Brafman, 2001, Brafman and Domshlak, 2008, Brenner and Nebel, 2009, Crosby et al., 2013, Fikes and Nilsson, 1971, Hoffmann and Nebel, 2001, Nissim et al., 2010, Torreño et al., 2012b, Traverso et al., 2004, Weerdt and Clement, 2009].

Cluster 3 - blue color

This cluster comprises five works, of which only two were published before 2014. In Komenda et al. [2016], the authors present the Cooperative Domain-independent Planners Competition of Distributed and Multi-agent Planners (CoDMAP) competition, focusing on multi-agent planners compatible with multi-agent extensions of classical planning models. The competition goal was to establish a standard problem description language to serve as a model for future MAP competitions. The article provides a cooperative MAP introduction and presents the Multi-Agent Planning Domain Definition Language (MA-PDDL) input language for encoding planning problems. Also, the authors highlighted the importance of privacy in MAP models.

The authors in Štolba et al. [2015] propose a distributed heuristic that provides estimates that are provably equal to estimates obtained by the centralized version of the algorithm. The authors evaluated the heuristic experimentally and showed that the distributed algorithm significantly improves the performance of a multi-agent planner.

In Torreño et al. [2014], the authors present the FMAP solver that combines planning and coordination in a distributed environment while safeguarding information privacy. FMAP is not limited to any particular domain and enables agents to collaborate on refining partial plans. The plan refinement process follows an interaction protocol that involves a significant number of message exchanges. Consequently, FMAP's performance and scalability are impacted by an increase in the number of agents.

Analyzing the three cited works, we note that the similarity resides in the issues of privacy and distributed planning. Both themes configure relevant trends explored in the MAP research area in the last decade.

Bibliographic Coupling Analysis

The coupling analysis highlights research trends and the MAP area's current direction regarding works published from 2014 to 2023. This analysis considers the authors' literature activity and the relationship among documents. Figure 2.8 presents authors clustered in seven groups. The authors and documents are connected by the number of references they share. Information about the clusters is presented in the sequence.

Cluster 1 - red color

This cluster includes 14 works focusing on MAP to dynamic environments and real-world applications. The work of Komenda et al. [2014] proposes a method for repairing plans in dynamic environments with multiple agents. This approach handles failures caused by state perturbations or the removal of actions from the plan. The proposed method includes three algorithms: Back on Track (BoT), Lazy Repair (LR), and Repeated Lazy Repair (RLR). The BoT algorithm involves computing a new plan from the failure state by adding new actions to the previous plan end, satisfying the execution conditions. In the LR, the executable part of the plan is computed and executed after a failure. Then, a plan is defined to close the gap between the state brought by the failed plan execution and the desired goal state. The RLR algorithm postpones the repair functionality as long as possible, ignoring failures during multi-agent plan execution. The method evaluation includes Logistics, Rovers, and Satellite International Planning Competition (IPC) domains. Results include evaluation of plan length, planning time, and message exchange output metrics. The authors concluded that repairing strategies present better performance than replanning in terms of communication.

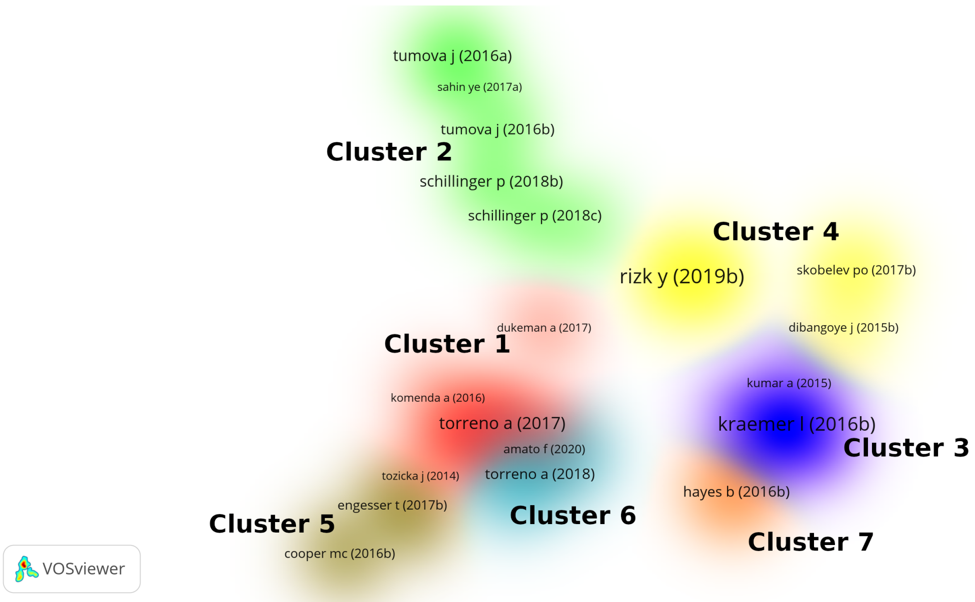


Figure (2.8) Bibliographic coupling Clusters of authors.

The Multi-Agent Distributed and Local Asynchronous (MADLA) planner is introduced by Štolba and Komenda [2017]. Unlike other solvers, MADLA utilizes two types of heuristics. MADLA runs on a distributed state-space forward-chaining multi-heuristic search combining local and distributed heuristics to leverage their individual benefits. The solver processes agents' information in isolation to compute the heuristic value, which is then shared to determine a final value. Additionally, the authors propose an improved privacy-preserving distribution scheme for the Fast-Forward heuristic.

MAPJA is a domain-independent approach proposed in Chouhan and Niyogi [2017]. MAPJA considers the agents' ability to solve MAP problems involving cooperative goals with joint actions. The approach involves checking whether any available multi-agent planner can solve the original problem. Planning is carried out centrally whenever it is impossible to divide the problem.

The works of ToźIăźKa et al. [2014, 2016] presented a MAP approach that combines compilation for a classical planner with a compact representation of local plans using finite-state machines. The approach was proven to be sound and complete. The efficiency is enhanced with distributed delete-relaxation heuristics and approximative local plan analysis. The evaluation includes experiments in fully private settings where only public information can be communicated. The authors analyzed standard multi-agent benchmarks' properties, focusing on classifying private and public information. They claim that the approach outperformed state-of-the-art planners designed for specific privacy classifications.

In Amato et al. [2020], a methodology is proposed to implement a planner of routes within cultural sites by integrating recommendation facilities with agent-based planning techniques. A user-centered recommendation strategy suggests cultural items based on user preferences. Depending on the context, multi-agent planning methods use state space exploration to generate related routes.

The authors in Dukeman and Adams [2017] propose coalition formation with planning to manage the complexity of mission planning with many robots in complex and dynamic domains. This approach allocates the best possible team of robots for each task resulting in the execution of the best possible plans.

The problem of ridesharing on timetabled public transport services using strategic multi-agent planning is investigated in Hrnčíř et al. [2015]. The proposed solution ensures that each individual is better off taking the shared ride than traveling alone, incentivizing participation. Real-world scenarios evaluated the algorithm's scalability and ability to address the trade-off between cost savings and journey duration. The proposed system could serve as the basis for a traveler-oriented ridesharing service and allow stakeholders to determine pricing policies and predict the effects of service changes.

The Greedy Privacy-Preserving Planner (GPPP) is presented in Maliah et al. [2016, 2017]. The GPPP has a multi-agent planning algorithm to generate a coordination plan while preserving the agents' privacy. The GPPP uses domain-independent privacy-preserving heuristics based on landmarks and pattern databases, demonstrating benefits in benchmark domains.

The remaining articles Crosby et al. [2014], Komenda et al. [2016], Torreño et al. [2014, 2017] have already been detailed in Section 2.2.1.

Cluster 2 - green color

This cluster includes six works related to formal methods to MAP with Linear Temporal Logic (LTL) formalism. The works of Tumova and Dimarogonas [2014, 2015, 2016] developed an iterative approach to solving the problem of plan synthesis for MAS with complex, high-level, long-term goals and requests for other agents' collaborations. These works reduce computational complexity by decomposing the problem into finite horizon planning problems that are solved iteratively with event-based synchronization, allowing for efficient adaptation to different agents' step durations.

The authors in Schillinger et al. [2018a,b] propose an automata-based approach to multi-agent plan synthesis for achieving high-level, long-term goals with requests for other agents' collaborations. By decomposing the plan synthesis problem into finite horizon planning problems, the approach reduces computational complexity enabling efficient search for an optimal decomposition and allocation of tasks to robot agents. The work is based on LTL mission specifications leading to the construction of a team model that enables the planning and control of the robot agents.

The work of Sahin et al. [2017] focuses on planning and LTL coordinating large collections of homogeneous agents subject to counting temporal logic constraints. The authors introduce a formal language to capture such tasks and present an optimization-based technique to synthesize plans to guarantee task satisfaction.

Cluster 3 - purple color

This cluster includes two works related to uncertainty and probabilistic techniques to MAP with partially observable Markov Decision Process (MDP) and other probability techniques. In Kumar et al. [2015], the authors developed a class of approximation algorithms for decentralized partially observable Markov decision process (Dec-POMDPs)that improves scalability by reformulating the MAP problem as inference in a mixture of dynamic Bayesian networks, with connections to machine learning. The approach can be extended to MAS with dozens of agents by identifying conditions that facilitate scalability. Experiments on large planning benchmarks confirm the approach's benefits in terms of runtime and scalability.

The work of Kraemer and Banerjee [2016] proposes a multi-agent reinforcement learning for solving Dec-POMDPs that allows agents to rehearse with information that will not be available during policy execution. The approach establishes a weak convergence result and shows faster learning and near-optimal performance on benchmark Dec-POMDP problems compared to existing approximate Dec-POMDP solvers.

Cluster 4 - yellow color

This cluster includes four works presenting different MAP applications. The authors in Rizk et al. [2019] present a survey on recent contributions to heterogeneous multi-robot systems (MRS). They emphasize the challenges of MRS sub-fields, including MAP and control, task decomposition, coalition formation, task allocation, and perception. The authors identify the limitations, remaining challenges, and possible future directions of autonomous MRS and highlight the need for more research to develop end-to-end solutions that minimize human intervention, automate complex task decomposition, and leverage Big Data advancements.

The authors in Cardoso and Ferrando [2021] present a literature review on agent-based programming for MAS. Intelligent and autonomous agents make decisions by reasoning about the world and use various techniques such as negotiation protocols, agent simulation, and MAP. The review focuses on agent programming languages, their extensions, comparisons, and applications.

The work of Skobelev et al. [2017] addresses the scheduling of Earth remote sensing satellites, with criteria including information delivery time, resolution, and cost. The schedule must comply with various constraints, including visibility and operations coordination. A MAP system was developed, with modules for handling dynamically occurring events. The implementation includes dynamic balancing of the interests of the satellites, data receiving points, and observation areas.

The authors in Dibangoye et al. [2015] focus on the impact of decentralized generation on the unit commitment problem in smart grids. The introduction of distributed generator units results in complex unit commitment problems that require distributed computation, privacy, and stochastic planning. The work proposes a novel distributed gradient descent algorithm to address these challenges and evaluates it on a real-time power grid simulator.

Cluster 5 - light brown color

In this cluster, we find two works related to logic formalisms to MAP with epistemic logic and model-checking techniques. In Cooper et al. [2016], the authors propose a logic-based approach for MAP that incorporates communication and knowledge into the planning model. Solving the planning task is reduced to a model-checking problem, kept in a planning domain definition language to provide a correct plan.

The work of Engesser et al. [2017] explores the based epistemic planning to propose an extension to the Dynamic Epistemic Logic. The approach allows for implicit coordination in multi-agent situations enabling the solution of decentralized planning tasks with joint goals without requiring agents to commit joint policies.

Clusters 6 and 7 - blue and orange colors

Privacy and collaboration Cluster 6 groups four works, which were previously detailed in this text Brafman [2015], Nissim and Brafman [2014], Torreño et al. [2014, 2017] in Section 2.2.1.

The work of Hayes and Scassellati [2016], which defines Cluster 7, focuses on collaboration between humans and robots, which requires solving challenging problems such as MAP, state estimation, and goal inference. The authors introduced a novel Hierarchical Task Network called CC-HTN and an algorithm for autonomously constructing them. Their method applies to a wide range of human-robot interaction scenarios. The authors present evaluations on goal inference and transfer learning tasks.

2.2.2 Temporal Overview

With the aim of presenting a map of the works highlighted in the bibliographic coupling analysis, we developed a temporal perspective. The objective is to illustrate the evolution of MAP scientific production from 2014 to 2023. The colors of the items can determine the recency of the works. The color legend ranges from blue for older documents (2014), green from 2016 to 2018, and yellow to red from 2018 to 2020. The legend is placed in the bottom right of Figure 2.9.

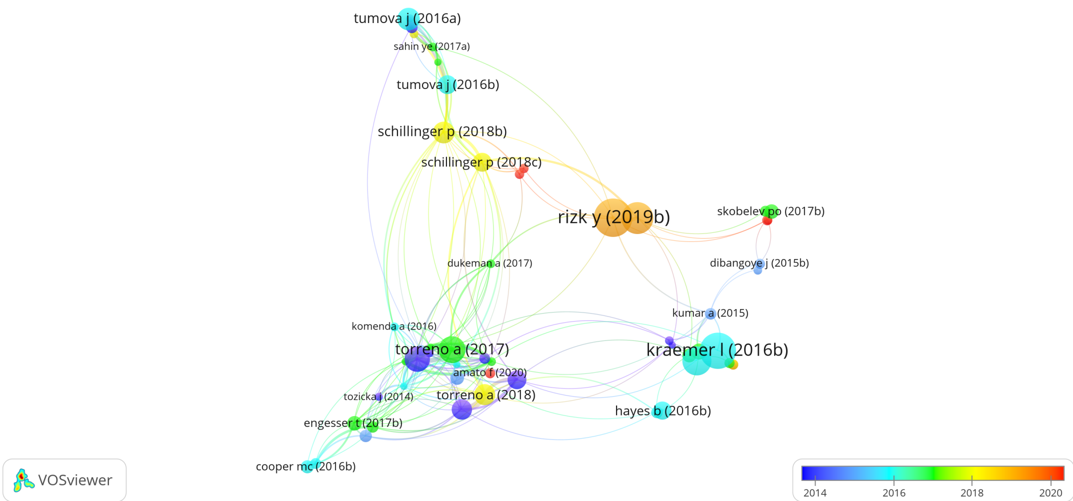


Figure (2.9) Overview of MAP publications from 2014 to 2020.

In the created map, the nodes represent the works whose size depends on the item weight or the number of citations. Furthermore, the arcs between these nodes indicate the relationship of works established by references. Whenever a document is cited by others, there will be an arc between them. Moreover, the greater the proximity, the more similarity between the works. From the information presented, note that most relevant and cited works are dated from 2017 or earlier since the nodes vary between blue and green colors.

Furthermore, note that Rizk et al. [2019] stands out in the central position of the figure because it is a literature review. Due to the nature of the work, it refers to several documents. The same behavior is noticed by the review work of [Torreño et al., 2017]. These two surveys configure centers of gravity that bring together relevant works identifying trends and challenges related to their production time. Therefore, these works are interesting starting points for beginners to study the MAP research area.

2.2.3 Selected Works

In the sequence, we present an overview of works published from 2021 to 2023 that address some challenges related to the combination of planning and executing in multi-agent environments as cited by Ghallab et al. [2014].

The works cited below were not studied in the previous analyses, of co-citation or bibliographic coupling, for being recent. Thus, it is likely that they have not yet been widely studied to the point of presenting a high number of citations, which would justify a prominent position in the analyses. However, it should be noted that the cited documents present contributions that point to the challenges related to planning research.

The authors in Babli et al. [2023] discuss plan repair for autonomous agents operating in dynamic environments with multiple agents. To minimize the negative impact on others, the authors propose plan commitment, a property for responsible plan repair. Their implementation, C-TFLAP, adapts failed plans while committing as much as possible to the original plan. Empirical evidence shows that commitment repair outperforms typical replanning and plan-stability repair in reducing failures and time loss among agents, especially when an agent makes commitments to others. However, determining an appropriate delay threshold for plan repair in a community of agents with reduced communication and privacy is a challenging task that requires further research.

The work of Queffelec et al. [2023] addresses a generalization of the Connected Multi-Agent Path Finding problem, where the graph is discovered during the agents' mission. The study presents a framework for the problem and examines the challenge of finding a strategy to reach a configuration in this setting. The research proves that the problem is PSPACE-complete when all agents must be connected at all times and NEXPTIME-hard in the decentralized case.

The authors in Bezrucav et al. [2022] dealt with failure during the execution of an action. The work defined the concept of an improper state as a condition such as the transitions between predicates did not perform correctly. For instance, considering a moving action in which predicates are used by preconditions and changed by effects describes the position of a robot. Whenever an error occurs between the switching of the states of those predicates, the robot may end in an improper state. In this situation, neither preconditions are held, nor the effects are updated. Hence, the environment model reaches a dead-end condition, and a planner cannot find viable solutions. In this sense, the work provides a framework to check and avoid improper states. At last, the evaluation experiment considers a unique configuration of two robots that need to navigate, grasp, and to objects.

The work developed in Matsuoka and Sawaragi [2022] focuses on a recovery planning system with a general repair strategy. Unlike the previous works, the authors employ a time constraint to control the recovery process. The proposed system depends on a centralized module that is responsible for defining the actions that are required to turn the plan operational again. However, the repair strategy is not implemented as new planning runs. Indeed, there is a finite and well-defined set of failures mapped to possible solutions. The evaluations include two robotic arms that can move and manipulate workpieces. The simulated errors are limited to missing workpieces, abnormal position of a robot, and interference of another robot.

In Moreira and Ralha [2022], the authors present the Lightweight Coordination Multi-Agent Planning (LCMAP) model that balances coordination and privacy through three independent phases. LCMAP has been compared to state-of-the-art models on loosely and tightly coupled domains, demonstrating its efficiency regarding time and plan length during problem-solving. The comparison shows that balancing computational processes and privacy provides efficiency to MAP models. LCMAP is an efficient model for MAP compared to the state-of-the-art.

The work of Saetti and Scala [2022] explored the same stability metric proposed by Fox et al. [2006]. However, the authors extended this former approach and proposed the distance optimization (difference) between plans. The experimental analysis discussed the performance of state-of-art planners compared to the model defined as Repair for Stability (RESA). The recovery problems focused on changes in the initial state and from an optimal and satisfying track of the IPC-2018. The analysis of experiments supported that RESA performed effectively.

The authors in Borrajo and Veloso [2021] proposed a specific replanning strategy approach. The authors developed a technique to identify opportunities to trigger a replanning process to improve the original plan. While other works focused on monitoring the environment to recover from execution failures, this work tried to take advantage of unpredictable events to generate a new and efficient sequence of actions. Shortly, the opportunity is a fact that no action can add or delete. However, if this turns true, the original plan can be improved. The evaluation discussion considered the document's domain in which a robot has a briefcase but no key to open it. So, the robot visits different rooms to grab copies of the documents inside the briefcase. The goal of the robot is to grab all documents. If the robot finds the key, there is an opportunity to improve the plan by ignoring many move actions.

Chrpa et al. [2020b] propose a technique to generate and execute plans that avoid dead-ends states, namely, an environment state such that the agent cannot reach goals by any means. Those plans are defined as robust because during execution the occurrence of non-deterministic events is not enough to lead agents to a dead-end. The proposal is evaluated considering different configurations of single-agent domains: autonomous underwater vehicle (AUV) sampling and the Perestroika game. In the first case, an AUV needs to perform sampling of some objects while avoiding ships passing by that might endanger it. The second domain is based on a game where an agent must navigate through a grid of solid and shrinking platforms to collect all resources that can be placed in solid locations. According to the authors, the technique outperformed the replanning strategies in all problem configurations.

Banfi and Campbell [2019] proposed an algorithm to tackle a grid environment where a robot must cross within a time limit. Obstacles are depicted with probabilistic values and the algorithm goal is to find the path that maximizes the robot's survival chance. In spite of being related to a dynamic environment, it is an optimization problem and a single-agent approach. According to the authors, the algorithm was able to compute the optimal solution in a reasonable time in instances of moderate size (16×22 grid).

Cashmore et al. [2019] tackled the problem of replanning for robots using the Temporal Planning Problem. In this sense, actions have well-defined duration and during their execution, it is likely to replan in order to recover from a failure or to avoid wasting resources, such as time and battery. Basically, the algorithm recovers from a failure between the preconditions and effects of an action under execution. For instance, during the execution of an action, its preconditions may no longer hold while its effects are not achieved yet. This failure within the action duration may create a discrepancy between the model and the real world. Results showed that the algorithm was efficient in situations where there was enough time to develop a new plan and actions were long-lasting.

Dehimi et al. [2018] studied an approach to generate a new plan by each agent whenever there is a change caused by unpredictable changes of the environment, such as new goals or new objects. The approach uses a genetic algorithm where the fitness function is defined based on the constraints to be satisfied. There is no planning phase because the sequence of actions is predefined. Indeed, the dynamic behavior is due to new requests or items which are inserted in the environment in runtime. The platform is JADE (Java Agent DEvelopment Framework) and there is one case study with Dynamic Pick and Delivery Problem (DPDP). The authors concluded that the approach allows the integration of the new actions in the correct order without altering the satisfaction of the constraints of the initial sequence of actions.

Gouidis et al. [2018] proposed DRA^* that is an extension of A^* algorithm that is suited for the repairing of sequential plans and can address two types of changes in the environment: goal-set modifications and actions' costs updates. The main difference between the A^* and DRA^* is that in the latter, when the search for the plan finishes, the closed and open lists are stored, so they can be used in case of replanning. Evaluations consider blocks, depots, grippers, logistics, iconic, and transport IPC domains. The conclusion was that DRA^* outperformed A^* in most of the cases and should be used in planning scenarios.

Mohalik et al. [2018] developed the Hierarchical Iterative Plan Repair approach (*HIPR*) that combines an architecture and algorithm that supports hierarchical agent teams to replan after a hazard occurrence. Agents are organized into a three-level tree. Operating agents are at the tree leaves; managing agents are at the internal levels and there is also a root agent. Agents try to repair their current plan locally, however, when it is impossible, higher agents are sent a signal and try to recover from failures related to preconditions or actions. The approach was validated considering a particular domain that is similar to logistics. Authors indicated the approach efficiency in hierarchical MAS although no practical experiments were presented.

The related work that influenced the development of this thesis has been summarized in Table 2.13. We drew inspiration from these works and focused on addressing the gap we identified during our literature review.

Table (2.13) Overview of related work.

Document	Replanning	Repairing	Evaluation	Contribution	Events
Babli et al. [2023]	✔	✔	average	C-TFLAP	Action failure
Queffelec et al. [2023]	-	-	-	complexity analysis	-
Bezrucav et al. [2022]	✔	-	-	improper states detection	Action failure
Matsuoka and Sawaragi [2022]	-	✔	-	repair strategies based on semantic information	Abnormal positions Interference of objects
Moreira and Ralha [2022]	-	-	-	balances between privacy and coordination	-
Saetti and Scala [2022]				stability metric	New initial state
Borrajo and Veloso [2021]	✔	-	ratio between metrics	opportunities	missing facts
Chrpa et al. [2020b]	-	✔	average	dead-ends states	-
Banfi and Campbell [2019]	-	-	-	mixed-integer nonlinear program	Probabilistic obstacles in path
Cashmore et al. [2019]	✔	-	-	temporal planning problem	Action failure
Dehimi et al. [2018]	✔	-	-	genetic algorithm use	New items
Gouidis et al. [2018]	-	✔	average	DRA^*	Goal-set modification Actions' costs alteration
Mohalik et al. [2018]	✔	✔	-	HIPR	Action impossible
Komenda et al. [2014]	✔	✔	average	repair strategies	Action removal state perturbation
This work	✔	✔	average, correlation T-test	combination of recovery strategies	Action removal State perturbation

2.3 Suplementary Works

Parallel to the literature review focusing on MAP, which was detailed in Section 2.2.3, we explored a literature review highlighting planning and control aspects in the intralogistics research [Fragapane et al., 2021]. This review focused on works categorizing control level, number, type of robots, resource management, and scheduling. One important conclusion of this literature review was the favorable condition attributed to planning techniques as a solution for the decision-making process in real-world problems.

We guided our work from the subset of related work to the recovery process, merging both literature reviews. The most common aspects in those works are optimizing time and number of actions, using simulation modeling, and focusing on different scenarios. We detail a list of crucial intralogistics works related to this thesis development.

Authors in Fragapane et al. [2020] and De Ryck et al. [2020] discuss the task allocation in which robots use a decentralized process for task assignment following simulation and mathematical modeling, respectively. While in Maniya and Bhatt [2011], the authors study the material handling operation based on a methodology for multi-attribute selection of automated guided vehicles. The proposal shows a decision-making process that details the main attributes of the automated vehicle.

A discrete event simulation was used to propose a control system for a warehouse scenario where autonomous robots transport materials. Actions are planned and coordinated under a scenario free of failures Kousi et al. [2019].

Path planning and scheduling are combined in a decentralized strategy to enable robots to operate in a manufacturing environment. The proposal focuses on planning and control of the movements and activities of the robots to provide a collision-free system Demesure et al. [2017].

Authors describe a decentralized system where every module uses operations logic based on local conditions and message exchanging in Gue et al. [2014]. The performance of the approach is evaluated under different scenarios, and they conclude it is deadlock-free.

This second literature review highlighted a gap regarding evaluating the impacts of a centralized or decentralized decision-making process over system performance. Moreover, the analysis considering an environment where robots may face failures during the execution of their tasks is also missing. Therefore, our work differs from the related work because we evaluate the impact of different decision-making strategies in a dynamic environment prone to failure.

Chapter 3 provides concepts to understand the rest of the chapters of this thesis.

Chapter 3

Preliminaries

Automated planning addresses the problem of finding a sequence of actions, transforming the environment from its initial state to some goal state. Works in this area tackle the planning process that is executed only once before acting in an offline approach. However, online planning presents challenges derived from integration with acting and the need to respond in complex real-world environments [Ghallab et al., 2016, Torreño et al., 2017].

3.1 MAS and MAP Concepts

A MAS is formed by a set of entities able to sense and change the environment state using their sensors and actuators, respectively [Weiss, 2013, Wooldridge, 2009]. In a MAS, some autonomous agents interact through a defined communication protocol allowing competitive or cooperative behavior with a coordination model. Moreover, agents can be goal-oriented, meaning they seek to achieve public or private goals. These goals are satisfied by the execution of a sequence of actions based on deterministic effects. Agents and the environment are formalized in Definitions 1 and 2, respectively.

Definition 1 Agents are assumed to have a finite set of possible actions that transform the state of the environment.

$$Ag = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$$

Thus, in a shared environment, competitive agents require a negotiation protocol, while cooperative agents need a planning protocol to define individual and group goals. Planning approaches implement centralized or distributed strategies for solving planning problems [Weiss, 2013]. Furthermore, the number of agents working in the same environment can also categorize the planning process [Jonsson and Rovatsos, 2011].

Definition 2 The environment is defined by a finite set E of discrete and instantaneous states, where a set of propositions forms the s_i state.

$$E = \{s_0, s_1, \ldots\}$$

The environment assumes a simple configuration when only a single agent remains. In this case, actions and goals belong to a single source. Consequently, it is impossible to delegate responsibilities. Also, there is no need for a coordination protocol since there are no agent interactions. Accordingly, a Single-Agent Planning (SAP) task can be formalized by Definition 3.

Definition 3 A SAP task can be formalized by a tuple $\Pi =< F, A, I, G >$, where:

- F is formed by a set of propositions;
- A is formed by a set of actions derived from operators;
- I is an initial state, $I \subseteq F$;
- G is formed by a set of goals, $G \subseteq F$.

On the other hand, in the presence of more agents, the planning model must coordinate interdependence among agents and their tasks. For each agent Ag_i, there is a SAP task $\Pi_{Ag_i} =< F_{Ag_i}, A_{Ag_i}, I_{Ag_i}, G_{Ag_i} >$. Therefore, a MAP task describes the combination of all tasks $\{\Pi_{Ag_1}, \Pi_{Ag_2}, \ldots, \Pi_{Ag_m}\}$ as Definition 4.

A mutex is an inconsistency among propositions. We assume that in the MAP setting, there are no mutexes in the initial state and the set of goals among agents. Therefore, $I = \bigcup\limits_{i=1}^{m} I_{Ag_i}$ and $G = \bigcup\limits_{i=1}^{m} G_{Ag_i}$.

Agents continuously interact with the environment, changing it from some initial state s_0 by performing some action α_i. This interaction between agent and environment is called run as presented in Definition 5.

Definition 4 A MAP task can be formalized by a tuple $\tau =< Ag, \bar{F}, \bar{A}, I, G >$, where:

- Ag $= \{Ag_1, Ag_2, \ldots, Ag_m\}$ is formed by a finite set of agents;
- $\bar{F} = \bigcup_{i=1}^{m} F_{Ag_i}$ is formed by a set of propositions;
- $\bar{A}$ is formed by a set of grounded actions, $\bar{A} = \bigcup_{i=1}^{m} A_{Ag_i}$;
- I is an initial state, $I \subseteq \bar{F}$;
- G is formed by a set of goals, $G \subseteq \bar{F}$.

Definition 5 A run of an agent in an environment is a sequence of interleaved environment states and actions.

$$run : s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots \xrightarrow{\alpha_n} s_n$$

Actions connect agents and the environment state. However, to define actions, the operators must be previously defined. At this point, it is possible to draw the first relationship between MAS and planning definitions. The Definitions 6 and 7 complement Definition 1 by the formalization of the items in the Ag set, namely, actions (α).

Definition 6 An operator is a schema that defines actions using variables (parameters). An operator is a tuple $\theta =< name(\theta), pre(\theta), eff(\theta) >$, where:

- $name(\theta)$ is an identification to the operator;
- $pre(\theta)$ is formed by a set of preconditions that stands for literals required to apply the operator; and
- $eff(\theta)$ is formed by a set of effects that stands for literals which are added or deleted from the state of the world after executing the operator, $eff^+(\theta)$ and $eff^-(\theta)$, respectively.

Definition 7 An action is an instantiated operator or an operator where objects replace parameters.

The operators, types, and variables define the planning domain. Literals updated on action's effects are called relevant facts. Only relevant facts must remain after an encoding phase to reduce the search space to explore in the planning phase. A tuple formed by all available actions, logical propositions, the environment initial state I, and the goal G is defined as a planning problem.

The transition caused by an action α applied in a state s is defined by Equation 3.1. For now, the transitions are considered to be affected only by the execution of an action.

$$\gamma(s, \alpha) = (s \setminus eff^{-}(\alpha)) \cup eff^{+}(\alpha) \tag{3.1}$$

The planning model purpose receives actions, initial state, and goals as inputs and produces a finite sequence of actions as output: a plan $\pi = [\alpha_1, \alpha_2, \ldots, \alpha_n]$. When a plan π is applied sequentially from the initial state I, it will generate a state s_n, where goals $G \subseteq s_n$. Equation 3.2 presents the evolution from the initial state to s_n.

$$\begin{aligned} \Gamma(I, \pi) &= \Gamma(I, [\alpha_1, \alpha_2, \ldots, \alpha_n]) \\ &= \Gamma(\Gamma(I, \alpha_1), [\alpha_2, \ldots, \alpha_n]) \\ &= \Gamma(\Gamma(\Gamma(I, \alpha_1), \alpha_2), [\alpha_3, \ldots, \alpha_n]) \\ &\vdots \\ &= \gamma(s_{n-1}, \alpha_n) = s_n \end{aligned} \tag{3.2}$$

In this context, run-time refers to the period when the agents carry out the planned actions. Moreover, it involves executing the strategies, with each agent performing their assigned tasks or activities based on the coordinated plan. The run-time concept is formalized according to Definition 8 [Jayaputera et al., 2007].

Definition 8 The run-time concept is centered on the action implementation, communication between agents, synchronization, and monitoring of the plan's progress.

In a multi-agent environment, plans of different agents coexist. In this sense, the single-agent plans need to be coordinated π_{Ag_i}. Let the sequence of actions be $\pi = [\alpha_1^1, \alpha_1^2, \alpha_2^1, \alpha_1^3, \alpha_2^2]$. The different π_{Ag_i} single-agent plans are described in Equation 3.3.

$$\pi_{Ag_i} = [\alpha_t^{Ag_i}, \ldots, \alpha_{m_{Ag_i}}^{Ag_i}], \text{ such that } {\alpha_y}^{Ag_i} \in A_{Ag_i}, t \leq y \leq m \tag{3.3}$$

The $\pi = [\alpha_1^1, \alpha_1^2, \alpha_2^1, \alpha_1^3, \alpha_2^2]$ describes actions from single-agent plans of three different agents, as follows.

$$\begin{aligned}
\pi_{Ag_1} &= [\alpha_1^1, \alpha_2^1, \varepsilon] \\
\pi_{Ag_2} &= [\alpha_1^2, \varepsilon, \alpha_3^2] \\
\pi_{Ag_3} &= [\varepsilon, \alpha_2^3, \varepsilon]
\end{aligned}$$

Each action in π_{Ag_i} is scheduled in a specific step t of execution. Therefore, the coordination defines which actions can be carried out at the same step t and which must stay idle. At the same step t, actions from different agents can be carried out simultaneously. For instance, $\alpha_{t=1}^1$ and $\alpha_{t=1}^2$. However, at other points, one action depends on the effects of an earlier executed action. In the same way, α_3^2 depends on α_2^1 or α_2^3. Thus, to maintain coordination, plans may need representations of these idle states between actions, which is done using empty actions, according to Definition 9.

Definition 9 An empty action ε stands for an idle state of the agent, where $pre(\varepsilon) = \emptyset$ and $eff(\varepsilon) = \emptyset$. An empty action causes no transition, namely, $\gamma(s, \varepsilon) = s$.

The coordination process guarantees that actions scheduled for the same step don't compete for resources or even undo the effects of each other. In other words, coordination protects all the causal links among actions. Furthermore, these actions can start in any order, namely, $\alpha_{t=1}^1$ before $\alpha_{t=1}^2$ or $\alpha_{t=1}^2$ before $\alpha_{t=1}^1$. So, actions can only be scheduled to the same step iff $\varphi(\alpha_t^i, \alpha_t^j) = \emptyset$ according to Equation 3.4.

$$\begin{gathered}
\varphi(\alpha_1, \alpha_2) = ({}^*\alpha_1 \cup {\alpha_1}^*) \cap ({}^*\alpha_2 \cup {\alpha_2}^*), \\
\text{where } {}^*\alpha_i = pre(\alpha_i) \text{ and } {\alpha_i}^* = eff(\alpha_i).
\end{gathered} \tag{3.4}$$

When two or more agents execute their plans simultaneously, they can compete for some resources or even undo the effects of each other's actions. However, these plans can provide cooperation with a minimum level of coordination when plans are independent according to Definition 10.

Definition 10 Two plans π_1 and π_2 are independent iff:

$$\begin{aligned}
({}^*\delta_{\pi_1} \cup \delta^*_{\pi_1}) \cap ({}^*\delta_{\pi_2} \cup \delta^*_{\pi_2}) &= \emptyset, \text{ where:} \\
{}^*\delta_\pi &= \bigcup_{i=1}^{n} pre(\alpha_i) | \alpha_i \in \pi = [a_1, \ldots, a_n], \\
\delta^*_\pi &= \bigcup_{i=1}^{n} eff(\alpha_i) | \alpha_i \in \pi = [a_1, \ldots, a_n].
\end{aligned}$$

This interference between agents and their plans derives from the fact that actions can be public or private, according to Definitions 11, 12, 13 and 14 [Brafman and Domshlak, 2008].

Definition 11 An action α^i is public whenever some propositions of its preconditions or effect appear in an action that belongs to another agent.

$$\exists j : i \neq j, i, j \in Ag, \alpha^i \in A^i, \alpha^j \in A^j | \varphi(\alpha^i, \alpha^j) \neq \emptyset$$

Definition 12 An action α^i is private whenever it does not affect nor depend on an action that belongs to another agent.

$$\nexists j : i \neq j, i, j \in Ag, \alpha^i \in A^i, \alpha^j \in A^j | \varphi(\alpha^i, \alpha^j) = \emptyset$$

Definition 13 The set of all public actions is defined by

$$A^{Pub} = \{\alpha | \exists i, j : i \neq j, i, j \in Ag, \alpha \in A^i, \alpha' \in A^j, \text{ and } \varphi(\alpha^i, \alpha^j) \neq \emptyset\}.$$

Definition 14 The set of all private actions is defined by

$$A^{Priv} = \bar{A} \setminus A^{Pub}.$$

Thus, A^{Priv} defines a set of actions that can be organized strictly by the agent that owns them. When planning is only about private actions, the process can be performed locally because the actions do not depend or are not dependent on other agents' actions Komenda et al. [2014]. This characteristic affects the coordination process since it is unnecessary to handle competition or cooperation issues, for instance, related to resource use. The coordination complexity can be formalized according to Brafman and Domshlak [2008], as presented in Definition 15.

Definition 15 The coordination complexity is a function of the number of actions executed by an agent that affect or depend on other agents. This function considers the level of coupling, that is, the interactions necessary to allow agents to control the dependency between them.

Intuitively, the less coupled the agents are, the easier to find a plan that solves the problem. Therefore, the greater the number of private actions, the easier the coordination. Likewise, the smaller the number of public actions, the less complex the coordination process. Similar to a single-agent plan, a multi-agent one is a sequence of actions but with special conditions presented in Definition 16.

Definition 16 A multi-plan ρ, with $\pi[i,t] = \alpha_t^i, i \in Ag, t \geq 1$ is a sequence of actions that can be executed in parallel by different agents at the same step t. Thus, the set of actions of a plan ρ to be executed at t is $A_{\rho t} = \{\alpha_t^i | \forall i, j \in Ag : i \neq j, \alpha_t^i \in \rho, \varphi(\alpha_t^i, \alpha_t^j) = \emptyset\}$.

The presence of multiple agents makes the execution of different actions at the step possible. So, in multi-agent plans, actions scheduled to the same step must be independent of each other (Equation 3.4) to guarantee cooperation and avoid competition. Thus, a multi-agent plan execution must be coordinated. An example of a data representation of a multi-agent plan is described by:

$$\rho = \begin{bmatrix} \alpha_1^1 & \alpha_2^1 & \varepsilon \\ \alpha_1^2 & \varepsilon & \alpha_2^2 \\ \varepsilon & \alpha_1^3 & \varepsilon \end{bmatrix}$$

The transitions from the initial state I caused by the actions in multi-agent plan ρ are described by $\Gamma(I, \rho)$, according to Equation 3.5.

$$\begin{aligned} \Gamma(I, \rho) &= \Gamma(I, [A_{\rho_1}, A_{\rho_2}, A_{\rho_3}, \ldots, A_{\rho_n}]) \\ &= \Gamma(\Gamma(I, A_{\rho_1}), [A_{\rho_2}, A_{\rho_3}, \ldots, A_{\rho_n}]) \\ &= \Gamma(\Gamma(\Gamma(I, A_{\rho_1}), A_{\rho_2}), [A_{\rho_3}, \ldots, A_{\rho_n}]) \\ &\vdots \\ &= \Gamma(s_{n-1}, A_{\rho_n}) = \bigcup^{i \in Ag} \gamma(s_{n-1}, \alpha_n^i) = s_n \end{aligned} \tag{3.5}$$

At last, it is important to highlight that there will be no propositions inconsistency in states derived from $\Gamma(I, \rho)$, since all actions in $A_{\rho t}$ are independent of each other (Equation 3.4).

3.2 Dynamic Environments

The planning process can face problems that can not be fully observed in a single and early stage. Some unpredicted events likely happen between two sequential states rather than the execution of an action. For instance, a new obstacle may be perceived in a robot's trajectory. Those events are called exogenous events. In this sense, agents face real-world challenges, according to Definition 17 [Jayaputera et al., 2007].

> **Definition 17** The real-world aspect refers to the external environment and agents' limitations during plan execution. This topic encompasses uncertainties, dynamic changes, and external events that can impact the plan's execution.

Real-world considerations involve factors like resource availability, the presence of other entities or agents in the environment, the need for adaptation or recovery in response to changing circumstances, and potential conflicts or coordination challenges due to interactions with various elements.

In this work, an environment where exogenous events can happen is considered dynamic, where an agent must continuously perceive the environment to compare current and predicted states.

Hence, under a dynamic configuration, an environment needs more elements and details to be defined because of these exogenous events caused by agents or existing objects. So, the Definition 2 must be extended to formalize a dynamic environment according to Definition 18.

> **Definition 18** The environment is a set formed by agents, objects, and events. The agents operate, sense, and act upon the environment. A dynamic environment is when exogenous events update their state.

Therefore, the execution cycle of an agent must follow the conceptual view illustrated in Figure 3.1. An agent percepts the execution platform to interpret the environmental signals. Whenever there is a difference between the current and predicted states, an agent analyses the planning situation to determine a new sequence of actions. Each action is converted to a command that updates the environment through the execution platform. Moreover, agents in a shared environment can exchange information such as goals and messages using a communication protocol. However, the concept view of Figure 3.1 is not restricted to a simulated or closed world. Indeed, agents perform actions in their surrounding environments using actuators. In the meanwhile, agents can face unpredicted events detected by reading the signals from the environment, such as a failure of one of the actuators.

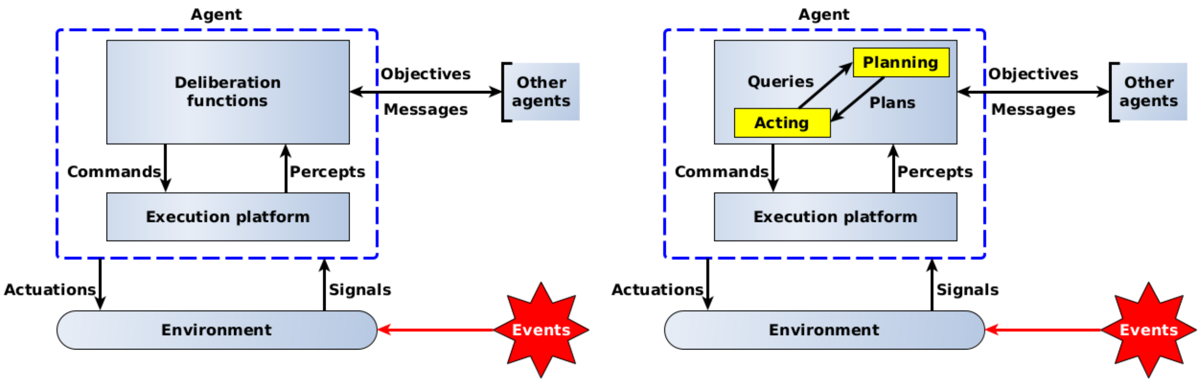


Figure (3.1) Conceptual view of agents in a dynamic environment (adapted from [Ghallab et al., 2016]).

The continuous comparison of current and expected states is vital to detect exogenous events. Thus, agents realize failures before executing some action through this monitoring process. This work assumes that agents' assessment of failures is perfect. Therefore, actions can not be carried out when failures impair the expected conditions that support execution. Definition 19 formalizes failures.

Definition 19 Plan failure is a discrepancy between the expected and current states caused by non-deterministic changes in the effects produced by the execution of actions.

Furthermore, MAP can be understood as the planning and executing process distributed over multiple agents [Torreño et al., 2017]. The agent distribution characteristic focuses on the number of agents and their roles while finding a solution for the problem. The planning entities are the agents involved in the reasoning stage of synthesizing the sequence of actions (plan). Executors are agents committed to executing actions, such as a robot or a software entity in a simulator. So, combining the number of planning and executor agents summarizes four schemes as illustrated in Table 3.1.

Table (3.1) Agent distribution [Torreño et al., 2017].

Agents		Plannning	
		1	n
Execution	1	Single-agent planning	Factored planning
	n	Planning **for** multiple agents	Planning **by** multiple agents

Moreover, in this work, we assume some premises to deal with MAP similarly to related work [Borrajo and Fernández, 2019, Cashmore et al., 2019, Komenda et al., 2014, Mohalik et al., 2018, Štolba and Komenda, 2015, Torreño et al., 2012a, 2014]:

- the world state is formed by a set of logical propositions;
- the environment is fully observable, and agents access information immediately;
- agents are collaborative, not competitive or self-interested, without a private goal;
- actions are unit cost and instantaneous;
- action is performed by only one agent, each action belongs to an agent;
- exogenous events are likely to take place in the environment; and
- communication process is free of errors.

Recovery Strategies

In this work, dynamicity means adapting and responding to environmental changes and other agents' behavior. It involves planning, coordination, monitoring, and executing actions while considering the non-deterministic nature of the environment and agent interactions. Dynamicity allows agents to adjust their plans and actions based on new information, goals, or constraints and to work together with other agents to achieve individual or collective goals. In MAS, it is crucial to ensure flexibility, robustness, and efficiency in complex and dynamic environments.

Definition 20 Dynamicity allows agents in multi-agent systems to adapt and respond to environmental changes and other agents' behavior.

As presented, exogenous events can cause failures during plan executions. In most cases, these events affect the environment state and action features. The first type of failure creates a perturbation in the state where the representation of the environment, a set of logical propositions, is updated by an unpredicted reason.

For instance, let the environment state in instant t_0 be $s_0 = \{p_1, p_2, p_3\}$ and in the next instant $s_1 = \{p_1, p_2, p_4\}$. Consider no action executing in the interval. Thus the state perturbation, namely, the deletion of p_3 and the insertion of p_4, could only be performed by an exogenous event. Regarding action issues, failures can impact preconditions, impairing the execution or the effects, leading to a non-deterministic state.

In this sense, agents need a process to update their plans regarding a new and unpredicted condition of the environment that they eventually face during the execution of their actions.

Essentially, MAP models concerning dynamic environments describe two recovery strategies. Both possibilities are described in Definitions 21 and 22. They follow different approaches and provide distinct results from the same failure. Hence, the newly generated plans highlight differences.

Definition 21 Replanning is a recovery strategy that deletes actions of a previous plan and builds a new sequence from the current state in which a failure was detected towards the goal state.

Definition 22 Repairing is the attempt to return the environment to the expected state by adding new actions just after the point where the current plan failed to continue the execution of a previous plan.

To illustrate the difference between the final plan related to replanning and repairing strategies, consider the conditions presented in Table 3.2 after the occurrence of a single failure.

Table (3.2) The difference in results of recovery strategies.

Strategy	Initial plan	Actions performed	Failure after	Final plan
Replanning	$[\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m]$	$[\alpha_1, \alpha_2]$	α_2	$[\alpha_1, \alpha_2, \alpha_{3'}, \ldots, \alpha_{m'}]$
Repairing	$[\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m]$	$[\alpha_1, \alpha_2]$	α_2	$[\alpha_1, \alpha_2, \beta, \alpha_3, \ldots, \alpha_m]$

Note that the replanning strategy provided a new plan replacing the suffix of the initial plan, starting from action α_3 by a new sequence of actions $\alpha_{3'}, \ldots, \alpha_{m'}$. Here, nothing can be assumed about the final plan length, which can either be bigger ($m' > m$) or smaller ($m' < m$) than the initial one. In contrast, the repairing strategy adds a new item β, a single action or a sequence of actions, after α_2 and then preserves the suffix $\alpha_3, \ldots, \alpha_m$. In this case, the final plan length depends on the repairing patch β; the number of actions will be $|\beta|+m$[1]. Likewise, in replanning, there is no previous conclusion about plan length. Therefore, the factor that defines the final amount of actions is the failure because this perturbation affects the new plan patches $\alpha_{3'}, \ldots, \alpha_{m'}$ (replanning) and β (repairing). Thus, repairing would build smaller plans than replanning if $|\beta| + m < m'$.

3.3 Statistical Techniques

Various statistical techniques are available to determine if there is adequate evidence to support or refute a particular statement or hypothesis regarding a population. These techniques can also be used to evaluate the magnitude and direction of the correlation between variables.

[1] $|\beta|$ denotes the number of actions described by β.

In this work, two statistical techniques are used to evaluate the recovery strategies' performance and behavior: T-Test inferential hypothesis test and correlation analysis. The T-Test refers to formulating statistically significant conclusions about an experiment by choosing between two alternatives [Igual et al., 2017]. The first alternative, labeled as the null hypothesis (H0), is the assumption that the means of two sets of data are not significantly different from each other. The alternative hypothesis (H1) describes that there is a difference between the means. The output of a hypothesis test is the probability (p-value) of observing data at least as favorable to H1 with the current data set if H0 is true. The smaller the p-value, the stronger evidence H0 is false.

Therefore, in the T-test, the p-value is a metric that guides to rejecting the null hypothesis (H0). In other words, claiming that the two samples, algorithms, or strategies have different means and, hence, distinguished performance levels. Furthermore, the p-value needs to be compared with a critical value[2] λ that stands for the test's acceptable limits. Usually, λ is set to 0.05 and represents that in 5% of the experiments, the null hypothesis will be rejected. In this sense, p-values $> \lambda = 0.05$ give no evidence to reject H0; $\lambda = 0.01 <$ p-values $< \lambda = 0.05$ show evidence; and p-values $< \lambda = 0.01$ lead to a piece of strong evidence that the samples have different means and they are different.

To investigate the relationship between a pair of variables, we used the correlation technique to show whether and how strongly the variables are related Igual et al. [2017]. The correlation value ranges from -1 to 1. The closer the value is to 1 or -1, the more related the two variables are. We checked two different types of correlation to avoid errors. Pearson's correlation describes the linear relationship between two variables and Spearman's correlation for non-linear monotonic relationships.

At last, the importance of using T-test and correlation techniques to evaluate the recovery strategies is two-fold. First, average values are affected by outliers caused by errors during simulation or sampling, adding abnormally high or low scores to the data set. Instead, the T-test provides robustness and simplicity of interpretation through the quantification of how different to samples are and if this difference is statistically significant. Furthermore, the correlation techniques quantify, even in the presence of outliers, the impacts of an input variable over a performance indicator. This analysis is not often explored in MAP works, especially in those related to dynamic environments. Therefore, combining both techniques leads to conclusions about the recovery strategies with a wider perspective than the related work.

[2]Critical value is usually described by α, but in this work, we use λ to avoid a misunderstanding with actions.

Chapter 4

Method for Evaluating Plan Recovery Strategies

Method for Evaluating Plan Recovery Strategies in Dynamic Multi-agent Environments. Full article published in Journal of Experimental & Theoretical Artificial Intelligence, p. 1-25, 2022.

4.1 Introduction

Autonomous agents, planning, and acting deliberation are research subjects in Artificial Intelligence because they propose solutions to real-world activities. Logistic and robotic in real scenarios are examples where multiple agents interact dynamically. Agents usually face challenges during execution in real-world scenarios since unpredicted events may interfere with their planned tasks. Hence, agents must deliberate to overcome failures [Moreira and Ralha, 2021a]. Dynamic environments require agents to perform a properly timed monitoring process, linking planning and acting [Ghallab et al., 2014].

In MAS, an interaction space is a set formed by agents, objects, and events. Each agent has its own beliefs about the interaction space, which is a common and shared environment where agents interact with other agents and objects. Unlike agents, objects do not have deliberation functions. Agents' interaction happens through the execution of actions toward the satisfaction of individual or collective objectives. Also, unexpected events can emerge from the interactions among agents in complex adaptive MAS, interfering with agents' initial planned tasks.

Although not usual in MAS, autonomous agents can apply a planning process as part of the deliberation function to evaluate actions considering their effects. Agents with the planning process choose specific actions and act in a sequence of tasks that achieves the desired goal. This sequence is called a plan that remains workable as long as unexpected events do not update the environment [Chrpa et al., 2020a, Ingrand and Ghallab, 2017]. The planning process supports the execution and coordination of selected actions, providing the correct order and timing. In scenarios with cooperative agents, planning allows agents' cooperation and avoids competition by the use of shared resources [Ghallab et al., 2014].

In the planning research area, some characteristics can categorize the proposed approaches. In works classified as MAP, agents distribute the planning responsibility or execution capabilities. Usually, such studies propose models that assume actions as the only source of the changes to the environment states, ignoring unexpected events from interactions [Chouhan and Niyogi, 2017, Štolba and Komenda, 2015, Torreño et al., 2014]. In this sense, these works consider a full-observable environment, deterministic, and no uncertainty about the actions' effects.

However, these characteristics limit their application in real-world environments where agents require abilities to face failures. For example, agents might face execution failures, communication problems, and limited knowledge about the facts that surround them. So, such approaches performed in an earlier and single-phase disconnected from execution have narrow applications. They cannot handle unexpected events because it is hard to predict all the states of a MAS.

Works published in the MAP area that consider dynamic environments model two plan recovery strategies. The replanning strategy deletes actions of a previous plan, building a new one from the current to the desired state [Gouidis et al., 2018, Komenda et al., 2012, 2014]. The repairing strategy tries to reuse actions of an earlier plan to reestablish a desirable and predicted condition [Cashmore et al., 2019, Komenda et al., 2012, 2014, Mohalik et al., 2018]. Usually, MAP works explore the average values of planning time, the number of actions, and failure occurrences. They rarely describe assessment aspects such as standard deviation and results' confidence level. Thus, there is a lack of statistical evaluation methods for recovery strategies in MAP. Such works support their analysis of average values, leading to hasty observations.

Thus, we investigated the performance of the strategies with a centralized MAP approach and different experiments varying the environment variables: number of agents, goals, actions, failure probability, and coupling level among agents. We are concerned about plan length and planning time performance metrics collected from the output of the simulations. With that specifications, we define two research questions (RQ):

- RQ1. How do replanning and repairing strategies perform when affected by failures in dynamic MAP environments?
- RQ2. How are environment variables related to the performance metrics?

Towards these questions investigation, we propose a domain-independent method for evaluating plan recovery strategies in dynamic multi-agent environments. We designed an empirical evaluation method that applies statistical techniques, such as the T-Test inferential hypothesis and correlation analysis [Igual et al., 2017] to evaluate the plan recovery strategies' performance (RQ1) and the relationships among input variables and output metrics (RQ2), respectively.

The contribution of this work is the development of a method to compare the performance of the plan recovery strategies under different planning conditions and interactions among agents. We extended the works highlighted in the literature review with additional analysis through the proposed method, leading to performance conclusions based on techniques that are more statistically significant than the analysis of average values. The proposed method encompasses centralized planning carried out by a planner agent with different coupling levels among executor agents. Also, the method includes a three-phase approach that bridges planning, acting, and monitoring through an evaluation of the relationship between input planning and output execution variables.

The rest of the document is structured as follows. In Section 4.2, we describe the statistical evaluation method. In Section 4.3, we present the experiments and discuss the results. Finally, we present the conclusion and future work in Section 4.4.

4.2 Proposed Method

The proposed method allows the plan recovery strategies and their evaluations to combine the three dimensions related to MAP in dynamic environments: planning, coordination, and execution. So, data is generated and collected to evaluate the performance of strategies in various scenarios, considering different dimensions. These three dimensions work together in a loop to address the gap identified in the related research [Ghallab et al., 2014, Micalizio and Torasso, 2007, Torreño et al., 2017].

Thus, the analysis preparation cycle can be summarized as simulation, recovery, and analysis as presented in Figure 4.1. The simulation starts with the planning process, and whenever a failure happens, agents coordinate their activities to recover.

From the conceptual view of agents in a dynamic environment (Figure 3.1), the proposed method to simulate and evaluate different configurations of MAP problems is illustrated in Figure 4.2. The simulations and evaluations follow a planning, acting, and monitoring loop. The red stars in Figure 4.2 represent the exogenous events that cause failures during the execution of agents' actions. The agents exchange messages and goals to coordinate the activities. Also, they cooperate to overcome failures after detection. The agents interact with the environment and other agents through their actions. Moreover, they sense the signals to detect differences between their beliefs and the environmental state. During the sequence of action execution, the agents monitor the environment to decide if they can act according to the plan or if they need to trigger a recovery process.

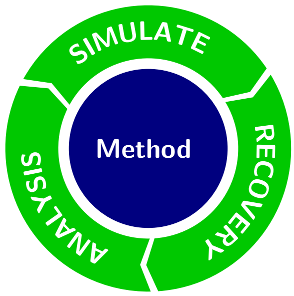


Figure (4.1) The analysis preparation cycle.

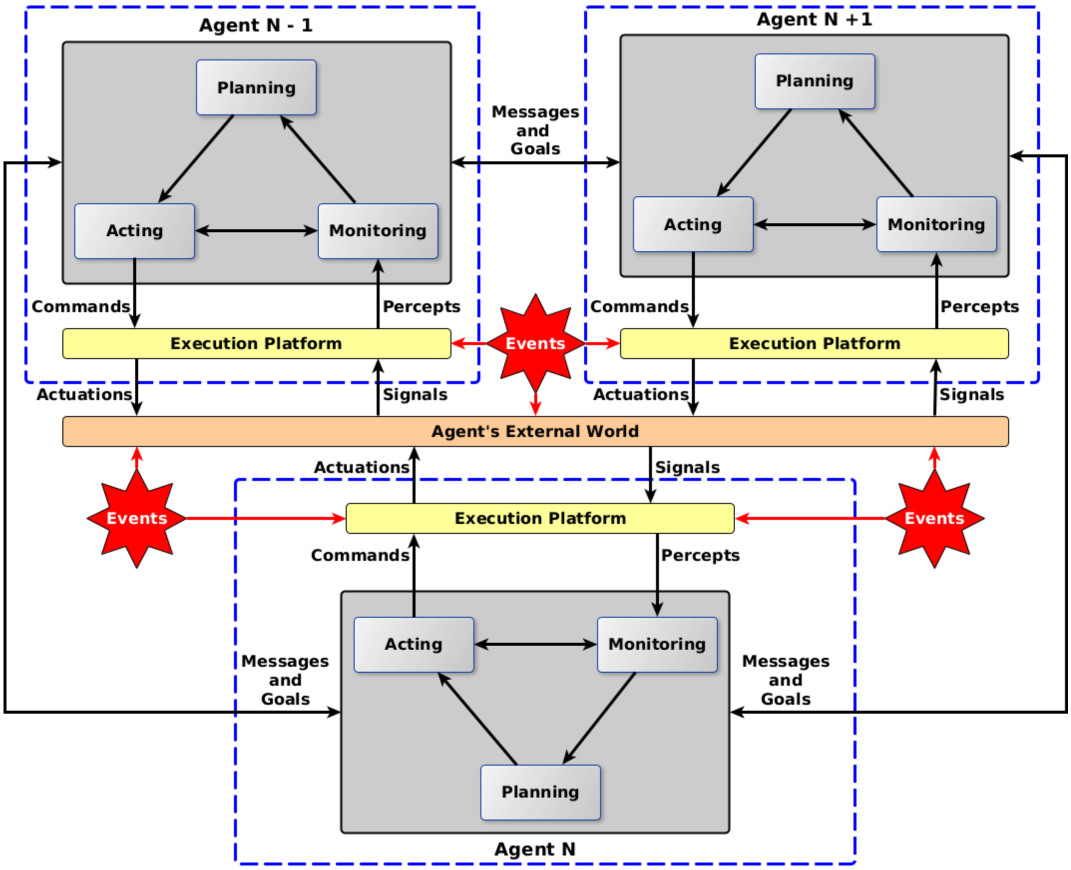


Figure (4.2) MAP in a dynamic environment.

In Section 4.2.1, we present the first phase related to simulation issues. The preparation of the simulation data is presented in Section 4.2.2. Finally, in Section 4.2.3, the statistical method to evaluate the performance of the recovery strategies (RQ1) and the relationships between variables that define the environment (RQ2) are presented.

4.2.1 Simulation

The simulation phase is composed of three steps: definition of the case studies and variables, simulation of the experiments, and the compilation of the simulation results.

Simulation Architecture

The simulation architecture in this work allows evaluation of the plan recovery strategies combining the three dimensions related to MAP in dynamic environments: planning, coordination, and execution. These dimensions are handled within a loop where data is generated and collected to allow the performance evaluation of the strategies in different conditions. Thus, the simulation architecture design can be summarised as an interaction loop between simulation, recovery, and analysis.

The user specifies the simulation inputs with a file in JSON[1] format that provides information about the domain, problem, and planner. Moreover, the user defines other properties, such as failure probability (in an open interval $]0.0; 1.0[$), timeout (minutes), and recovery strategy (repairing or replanning). These input parameters are handled and the planning problem is parsed and encoded to identify the initial state, goals, ground actions, and relevant facts. The simulation inputs are used to complete the process of instantiating the environment. Then, the simulation enters into a loop where agents act by executing their plans and monitoring the environment, and whenever necessary, they trigger a planning process to recover from failures. This loop ends in a state where the goals are held or if the simulation reaches the timeout. Finally, data is analyzed from the raw results and some output metrics are generated and sent to the user completing the process.

The simulation architecture is presented in Figure 4.3, where labels and arrows in blue stand for the features inherited from the conceptual view of agents in a dynamic environment presented in Figure 3.1. It was implemented in a platform that embedded the PDDL4J[2] JAVA library to support the planning research area [Pellier and Fiorino, 2018]. The PDDL4J is based on the Planning Domain Definition Language (PDDL) language and offers useful functions:

[1]https://www.json.org/json-en.html
[2]https://github.com/pellierd/pddl4j

- parsing: the PDDL files are parsed and instantiated as objects of specific classes;
- pre-processing: operators are converted into ground actions based on the problems properties;
- pre-solving: checks whether the planning is solvable;
- immutable propositions are erased, only the relevant facts are kept; and
- classical heuristics and planners.

Regarding the simulation engine, this work was based on the Repast Symphony[3] platform for supporting agent-based modeling and simulation [North et al., 2013]. The reason for using this tool was motivated by its (i) flexibility to code agents' behavior as JAVA methods; (ii) possibility to run batch simulations; (iii) ability to summarise results of the simulation in text files; and (iv) possibility to distribute simulation on different machines.

It's important to note that the method pipeline describes the order of activities rather than the specific tools used. Meaning there are many different tools that can be combined to create an evaluation tool. For example, JADE[4], JaCaMo[5] and NetLogo[6] can be used to support simulation systems, while the R Project[7] and Python tools like Pandas[8] and Matplotlib[9] can be used for statistical computing and generating graphics.

The file displayed in Figure 4.3 is called the experiments file. This file contains valuable information about the MAP task, plus supporting batch simulation. It is written in JSON format and includes various experiment configurations. Each configuration handles a unique *configurationId* attribute and provides essential details such as the location of the domain (Line 5) and problem (Line 6) files (*domainPath* and *problemPath*), the planner utilized by the PDDL4J library, and the maximum time limit for finding a solution. The *configurations* attribute is an array of experiment configurations (*configurationId* - Line 16). Listing 4.1 presents all the attributes in this file.

[3]https://repast.github.io/
[4]https://jade.tilab.com/
[5]http://jacamo.sourceforge.net/
[6]https://ccl.northwestern.edu/netlogo/
[7]https://www.r-project.org/
[8]https://pandas.pydata.org/
[9]https://matplotlib.org/

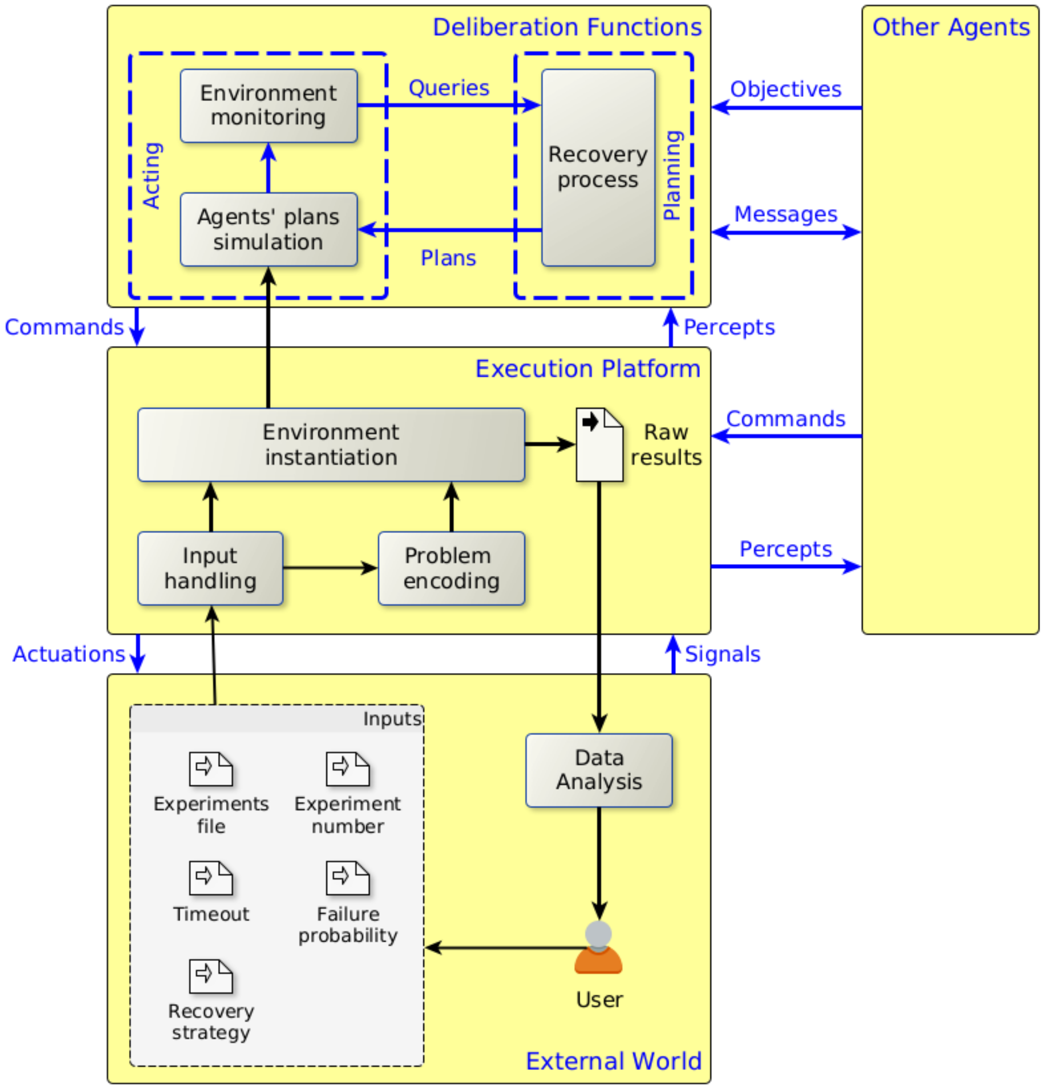


Figure (4.3) The simulation architecture.

```
"configurations": [{
    "configurationId": "logistics1",
    "domainPath": "logistics/domain.pddl",
    "problemPath": "logistics/problem.pddl",
    "planner": {
        "name": "FF",
        "timeout": 300000,
        "heuristic": "FAST_FORWARD",
        "staticState": true,
        "traceLevel": 0
    }
},{
    "configurationId": "logistics2",
    ...
```

Listing (4.1) Experiments file example.

Case Studies and Variables

The case studies applied in the experiments were based on the domains and problems used in CoDMAP that were carried out with the workshop on Distributed and Multi-agent Planning (DMAP) at the International Conference on Automated Planning and Scheduling (ICAPS) 2015. CoDMAP was organized to consolidate the planners in terms of input format; to promote the development of multi-agent planners, both inside and outside of the research community; and to provide a proof-of-concept of a potential future multi-agent planning track of the IPC (CoDMAP, 2015). The case studies were defined using the Multi-Agent Planning Domain Definition Language (MA-PDDL) [Kovács, 2012b], designed to model MAP problems.

Summarily, a planning domain is a description of the operators, types, and variables that are used to describe the environment. A planning problem is a concrete instance of an environment, in which every object is defined by a type, inclusive of the agents, and relevant literals are set to true. When a literal is not set, it is considered false, according to the closed-world assumption [Ghallab et al., 2016].

Regarding the most used case studies described in the selected works from the literature review, the domains chosen from CoDMAP were:

- Satellite - each agent symbolizes a satellite that is defined by its attributes regarding the position, orientation (direction), and available instruments. Although they have different qualities, this condition favors the transformation of the original problem because agents do not need cooperation to carry out their actions. The problem is to scale satellite observations that include collecting and storing data using different instruments to observe a collection of targets;

- Logistics - the agents in this domain are planes and trucks. The delivery of some packages involves the cooperation of elements of both types since the trip between cities is realized by planes and the displacement within the same city is the responsibility of trucks; and

- Driverlog - similar to logistics. Agents are drivers that drive trucks between locations. Drivers' walking requires a traversal of different paths from those used for trucks' driving, and there is always one intermediate location on a footpath between two road junctions. Trucks can be loaded or unloaded with packages, the goal is to transport them.

These case studies were selected because they provide different levels of public action. Therefore, from this subset of planning domains and problems, it was possible to run simulations with distinct coordination complexity because the interaction among agents is defined by the dependencies and effects of the agent's actions.

Simulation Experiments and Results

The purpose of running experiments is to generate data to evaluate the performance of planning recovery strategies. Thus, the hypothesis to be checked is whether the replanning and repairing strategies perform differently under the same conditions, such as failure probability, number of agents, number of goals, and coupling level among agents.

The failures are simulated in the following way. The effects of an action α can be undone following a continuous uniform distribution after its execution. So, whether an action fails, the number of propositions $eff(\alpha)$ that may be affected is also randomly defined by the same distribution. Finally, to simulate the failure, the propositions randomly selected from $eff(\alpha)$ are returned to the same condition before the execution of the action, namely $pre(\alpha)$. In this way, we avoid inconsistencies in the state of the environment.

We used a uniform distribution ranging from 0.1 to 0.9 in increments of 0.1. We coded replanning and repairing recovery strategies based on BoT and LR approaches outlined in Komenda et al. [2014]. For each domain, we simulated three different problems 30 times. For example, in the Satellite domain, we used the p05-pfile5, p07-pfile7, and p09-pfile9 problem files to simulate different numbers of agents and goals. These simulations took over 40 hours to complete, with 4,860 simulations conducted across three domains, nine failure probabilities, three problems, two strategies, and 30 simulations. Table 4.1 summarizes the setup details for each domain, including the number of agents, goals, and actions in each domain. The files used in the simulations are available at http://agents.fel.cvut.cz/codmap/domains/. The column labels in the table indicate the meaning of each parameter as follows:

- Domain: planning problem defined in CoDMAP;
- Problem: file that defines the planning problem;
- Id: identification of the problem file that was used to select the simulation inputs and designate the experiments with unique values for each domain;
- Agents: number of executor or planner agents;
- Goals: number of literals to be set true;
- Actions: total number of actions instantiated from planning problem definitions;
- Public: total number of public actions, according to Definition 13; and
- Ratio (%): percentage of public actions.

The experiments were carried out in a computer with an Intel Core i7-10510U CPU@ 1.80GHz, eight executable threads, and 16 GB RAM. The operational system was Linux

Ubuntu 20.04 LTS 64-bit. The FF planner algorithm provided within the PDDL4J library was used in all experiments [Hoffmann and Nebel, 2001].

4.2.2 Data Preparation

The first step of data preparation is the definition of metrics to be evaluated. The plan length stands for the sum of actions carried out by the agents. The planning time metric summarises the total time spent during planning activities. Failures describe how many times agents detected differences between expected and current states.

Similar to selected works, such as Cashmore et al. [2019], Gouidis et al. [2018], Komenda et al. [2014], we use several agents and goals, failure probability as input variables, while final plan length, planning time, and failure are output variables. We aimed to explore a twofold analysis. The agents and goals are counted quantitatively (number of) while the actions are classified qualitatively considering their public or private aspects. The purpose of this categorization was to allow the investigation of the research question regarding the similarities that the domains and problems present. In other words, the case studies were meant to be evaluated considering the similar complexity faced to find a solution rather than individual domains.

The next steps of data preparation are about building charts and confidence intervals using the metrics. Charts are useful to allow a visual evaluation of the difference between the performance of recovery strategies. The metrics were evaluated regarding their mean values. The Pandas tool facilitates the computation of the average value by providing different grouping options [Pandas development team, 2021]. For instance, simulation outputs are grouped by regarding the domain, the experiment configuration, recovery strategy, and failure probability. The confidence interval (CI) is a range of values that best represents some parameter of interest. A CI has a θ probability of containing the true underlying parameter, in our case the mean [Igual et al., 2017].

Table (4.1) Setup description including simulation details.

Domain	Problem	Id	Agents	Goals	Actions	Public	Ratio (%)
Satellite	p05-pfile5	0	3	6	497	200	40.24
Satellite	p07-pfile7	2	4	7	756	204	26.98
Satellite	p09-pfile9	4	5	10	1,473	390	26.48
Logistics	problogistics-6-0	2	3	6	78	48	61.54
Logistics	problogistics-7-0	3	4	7	174	108	62.07
Logistics	problogistics-10-0	8	5	10	308	192	62.34
Driverlog	pfile5	4	3	7	288	252	87.50
Driverlog	pfile14	9	3	8	1,878	1,710	91.05
Driverlog	pfile15	14	4	9	4,896	4,544	92.81

4.2.3 Data Evaluation

This phase applies three steps to study the performance of the recovery strategies: hypothesis test; the correlation between two variables; and the variation of correlation. The performance of the recovery strategies was evaluated using the output metrics collected from the simulations.

The first research question (RQ1), regarding the performance of the recovering plan strategies in a dynamic multi-agent environment affected by state perturbation, is studied using the T-test hypothesis test.

To investigate the research question (RQ2) about the relationship between the variables, we use Pearson's and Spearman's correlation statistical techniques to describe how closely related the two variables are. Furthermore, the correlation between variables is also checked regarding the coupling level among agents.

4.3 Experiments and Discussion

The results are discussed regarding the two research questions. In Section 4.3.1, we detail the performance of the recovery strategies while we study the relationships among the inputs and the outputs in Section 4.3.2.

4.3.1 RQ1 Investigation

Regarding the hypothesis test of experiments, the values presented in Table 4.2 show the p-values of the final plan length, planning time, and failure metrics. By analyzing these values, one can reject or accept the null hypothesis that the recovery strategies, replanning and repairing, have different means for each metric.

Table (4.2) Hypothesis test of the mean difference between replanning and repairing.

Metric	Satellite	Logistics	Driverlog
Final Plan Length	0.0767	0.0	0.001
Planning Time	0.1489	0.0	0.0
Failures	0.8106	0.0	0.103

In Figures 4.4, 4.5 and 4.6, the values in the horizontal axis stand for the experiments, while the bars describe the average value computed from all simulations grouped by experiment. The charts related to planning time (Figures 4.4(b), 4.5(b) and 4.6(b)) are displayed using a logarithmic scale in the vertical axis because of scale adjusts. The red and blue bars stand for the mean values of repairing and replanning strategies, respectively. The 95% confidence interval is presented at the top of the bar. Each pair of columns stands for one of the three different problems of the domain, where the number of agents and goals are varied.

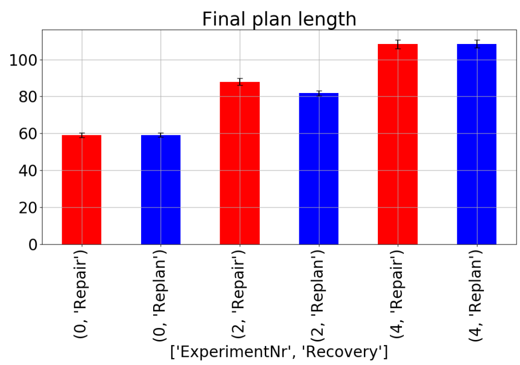
The first analysis was conducted from the simulation results of the Satellite domain where the interaction among the agents varies from 26% to 40%. This condition leads to a planning environment where actions are mostly private and, therefore, their executions do not affect or depend on other actions. Since the interaction among agents shows low levels, the new actions built from the recovery process were similar, regardless of the strategy.

This similarity presented in the Satellite domain can be visualized in Figures 4.4(a)-(c) where bars from the same experiment number show similar heights (means). Due to this similarity, the T-Test resulted in p-values higher than 0.05 which confirms that there are no differences at all regarding the performance of the strategies. Furthermore, the presence of overlapping confidence intervals depicts the fact that the true mean value may be equal in both recovery strategies.

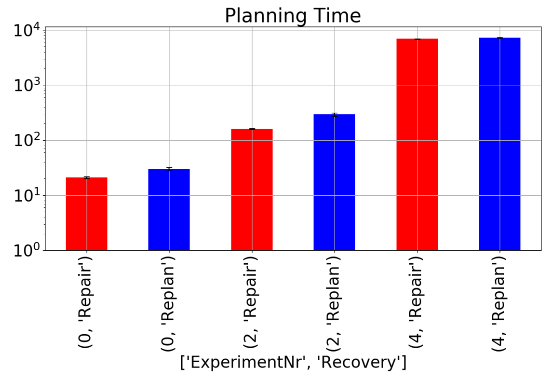
A closer analysis of the middle bars in Figure 4.4(a) details the importance of a T-test. The average values of the repairing and replanning strategies, 87.98 and 81.46 respectively, may guide to an unwise conclusion that the latter showed a better performance than the first strategy, however, this difference has no statistical significance and was probably caused by outliers. The robustness of the T-test technique avoids this misunderstanding because its p-value of $0.0767 > 0.05$ gives no evidence to reject the null hypothesis.

In the logistics domain experiments, the strategies' behavior changed. Now, the actions are mostly public with a level greater than 60%; hence agents affect or depend on other agents' actions more often. In this context, the repairing strategy proved to be limited because adding extra actions to preserve the original plan could not recover the plan efficiently. Thus, a side effect of the temporary solution to the failure is the appending of many actions that do not remove the agent of the failed state because of focusing on the failure instead of the goal. Hence, the final plan length and failure metrics are greater than those related to the replanning strategy. Due to the strong interaction among agents, efficient recovery requires a global analysis and a reset of the current plan, which is possible from the replanning strategy.

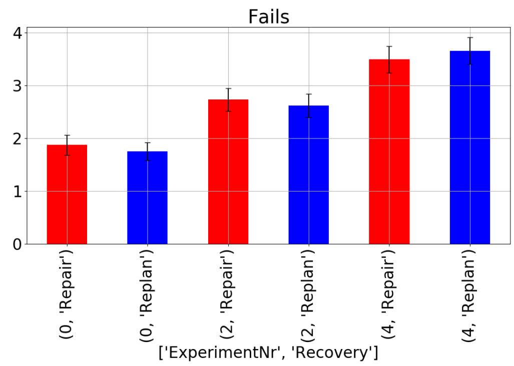
Therefore, the p-values (Table 4.2) are all equal 0.0 because the strategies had explicit distinct performances; in other words, one can reject the null hypothesis that the mean of the results of each strategy is equal. It is possible to identify the difference between the performance of the recovery strategies by analyzing Figure 4.5 (a)-(c). Regarding the final plan length metric, the repairing had higher values than the replanning strategy. However, the increase in the number of agents, goals, and coupling level among agents show that the replanning becomes slower than repairing while comparing the planning time metric.



(a) Final plan length (number of actions).



(b) Planning time (milliseconds).



(c) Failure (units).

Figure (4.4) Output metrics of the Satellite domain.

The analysis of the failure number follows the same pattern as the final plan length. A partial conclusion that can be drawn from these experiments is that, under more coupled domains, replanning builds better plans (smaller) than the repairing strategy at the cost of being slower.

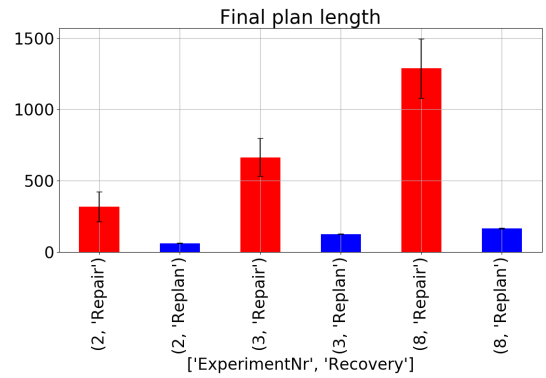
The recovery strategies in the experiments of Driverlog domains had special results regarding the performance of the earlier domains. The agents tried out a strong interaction that turned the planning process more complex than the previous domains because of the ratio of public actions around 90%. The time spent on the replanning strategy was more significant than the repairing. However, the final plans had fewer actions. In this sense, the T-test must confirm the difference between the performance of the recovery strategies.

The hypothesis test results on the final length and planning time metrics showed that the strategies had different performances based on the p-values of 0.001 and 0.0 in Table 1. However, the evaluation of failures had a different outcome. While rejecting the null hypothesis with a significance level of 0.05 (95%) was impossible for the fails case, it was rejected for the other two metrics. The comparison is illustrated in Figure 2(c), where the overlapping confidence intervals indicate no significant difference between the true mean values. From the Driverlog experiments, we can conclude that in tightly coupled domains, replanning builds smaller plans than repairing. However, these plans are not effective enough to reduce the number of failures as they do in the logistics domain.

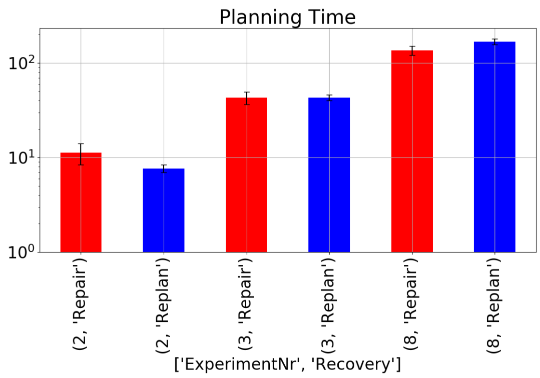
At last, by the current results and evaluation, it is possible to draw some partial conclusions about RQ1. First, about planning time, the more tightly coupled the problem is, the slower the replanning strategy is. However, the cost of planning time is balanced by better plans than replanning builds. Furthermore, there is no significant difference between the strategies in loosely coupled scenarios, such as Satellite domains.

4.3.2 RQ2 Investigation

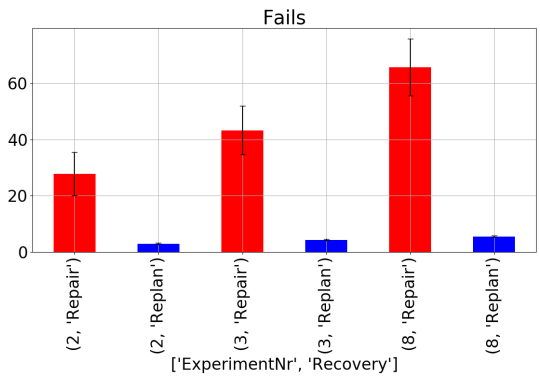
After analyzing the recovery strategies' performance, the next step in the proposed evaluation method is studying the relationship between the input variables and the output metrics. After checking the null hypothesis, computing the correlation between the variables can be seen as a verification of the earlier tests. In other words, whether the T-test indicates no significant difference between strategies, the correlation values are expected to be similar. Otherwise, different values must be output by performing Pearson's and Spearman's correlation techniques.



(a) Final plan length (number of actions).



(b) Planning time (milliseconds).

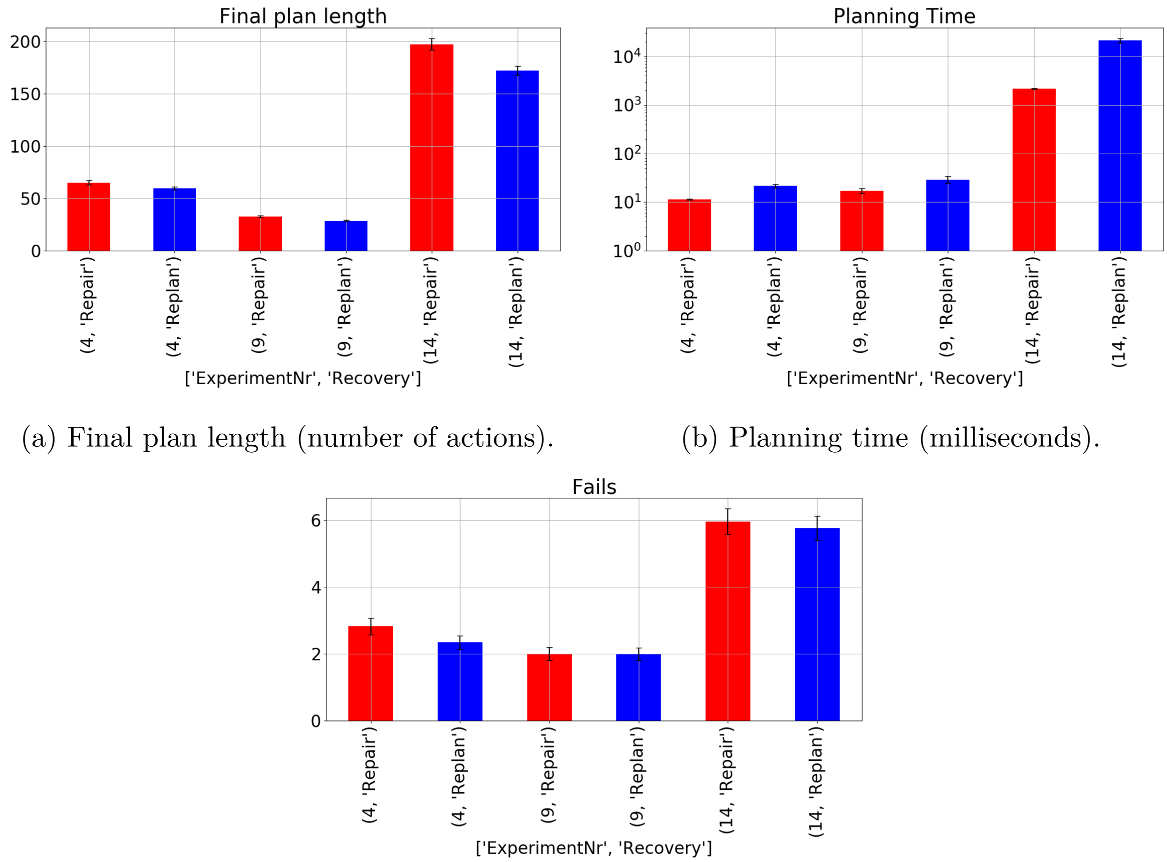


(c) Failure (units).

Figure (4.5) Output metrics of the logistics domain.

The relationship between variables is illustrated in Figures 4.7, 4.8, and 4.9 using a matrix presentation. Moreover, the values follow a heat map scale, where the darker the cell, the closer to 1.0 the value is. Since a correlation matrix is symmetric, $corr(a, b) = corr(b, a)$, Figures 4.7, 4.8 and 4.9 were modified to save space. Thus the cells below and above the diagonal represent Spearman's and Pearson's values, respectively.

Regarding the Satellite domain (Figure 4.7), the number of agents and goals are strongly related to the final plan and planning times output metrics in both correlation methods. It is possible to identify that the relationship between those parameters and metrics are over ranges from 0.734 and 0.969 (Figure 4.7(a)) to 0.819 and 0.954 (Figure 4.7(b)), repairing and replanning, respectively. Moreover, the similarity among the values in the correlation matrices is justified by the fact that there was no significant difference between the repairing and replanning strategies in the Satellite domain, according to the results of the T-test.

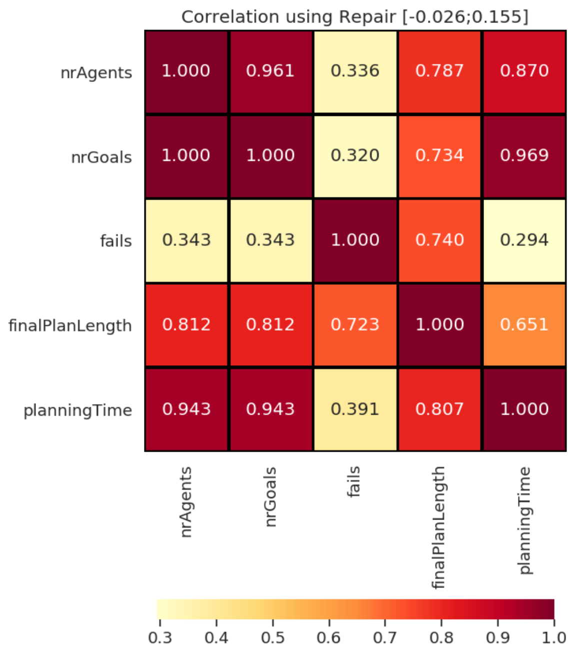


(a) Final plan length (number of actions).

(b) Planning time (milliseconds).

(c) Failure (units).

Figure (4.6) Output metrics of the Driverlog domain.



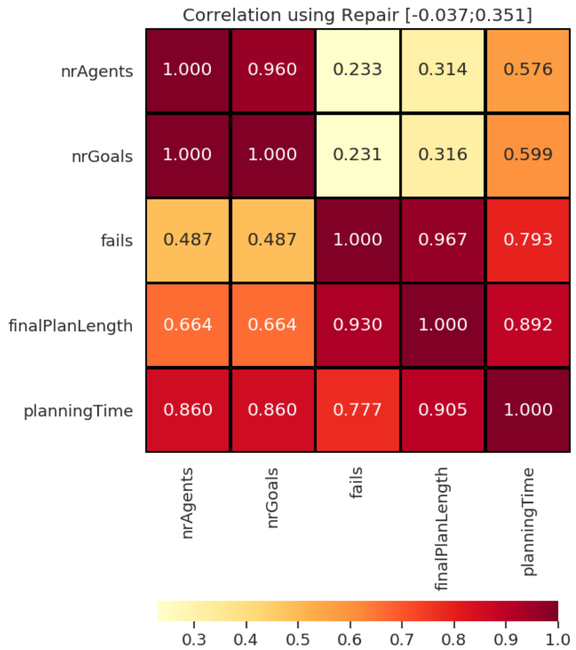
(a) Correlation using repairing.

(b) Correlation using replanning.

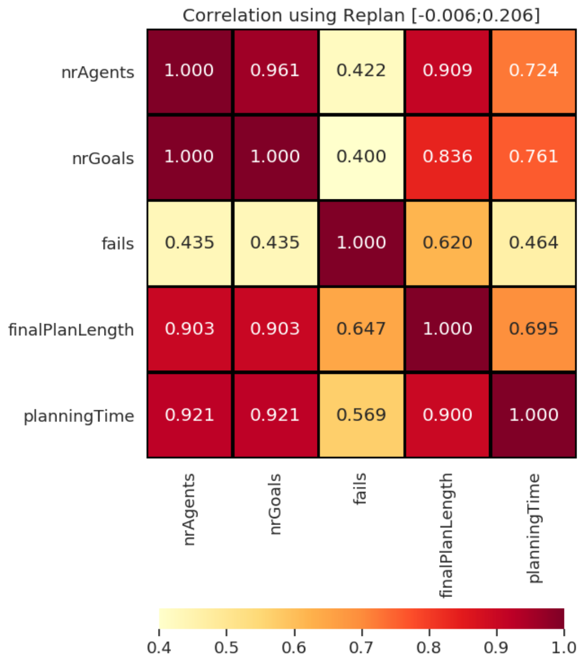
Figure (4.7) Correlation in the Satellite domain.

Considering the logistics domain (Figure 4.8), the correlation values were not so close. The behavior was expected because the hypothesis tests indicated a significant difference between the performance of the strategies. Furthermore, Spearman's method presented higher correlation values between the input and output variables than Pearson's. Theses values vary ranges from 0.314 and 0.86 (Figure 4.8(a)) to 0.724 and 0.921 (Figure 4.8(b)), repairing and replanning, respectively. In the logistics domain, agents try out a stronger interaction because of the ratio of public actions that is greater than 60%, hence the complexity of running the planning process is higher too. The replanning strategy is impaired by the increase in the number of agents and goals. The correlation values related to planning time and these input metrics, $0.724, 0.761, 0.921$, highlight that replanning becomes slower than the repairing strategy, where the values are $0.576, 0.599, 0.86$. These correlation values reinforce the partial conclusion about both strategies in logistics domains, namely, replanning builds better plans than the repairing strategy at the cost of being slower.

Regarding the Driverlog domain (Figure 4.9), the matrices were more closely similar to the Satellite. The final plan and planning time metrics are strongly related to the number of agents and goals. However, when using Pearson's method, the planning time showed the highest correlation value with the final plan metric. It is justified by the fact that problems with a higher number of actions in the solution plan tend to require more time to be solved.



(a) Correlation using repairing.



(b) Correlation using replanning.

Figure (4.8) Correlation in the logistics domain.

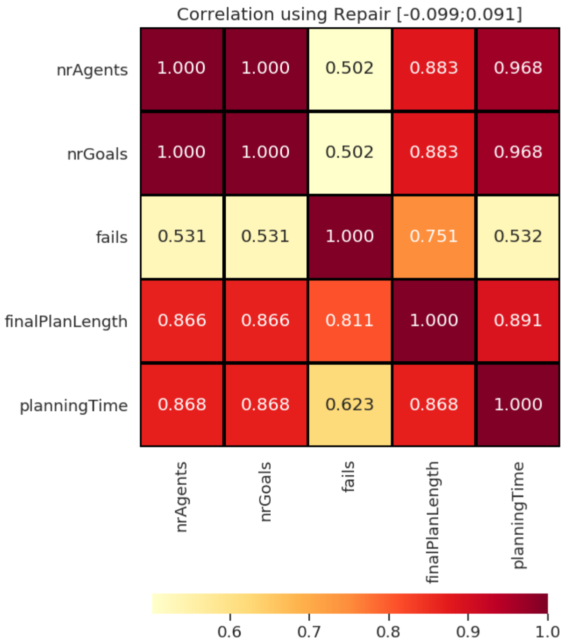
The correlation matrix comparison of different methods is highlighted in Figure 4.10, where red, orange, and yellow colors stand for the difference between Spearman's and Pearson's, greater, equal, or less than zero, respectively. The cells below and above the diagonal stand for the repairing and replanning strategies, respectively.

Considering the Satellite domain (Figure 4.10(a)), the similarity correlation matrices are justified by the impossibility of rejecting the null hypothesis. Thus, the greatest difference, 0.155, occurred considering the relation between planning time and final plan length using the repairing strategy. The Spearman's values were 40% higher than Pearson's (8 of the 20[10]).

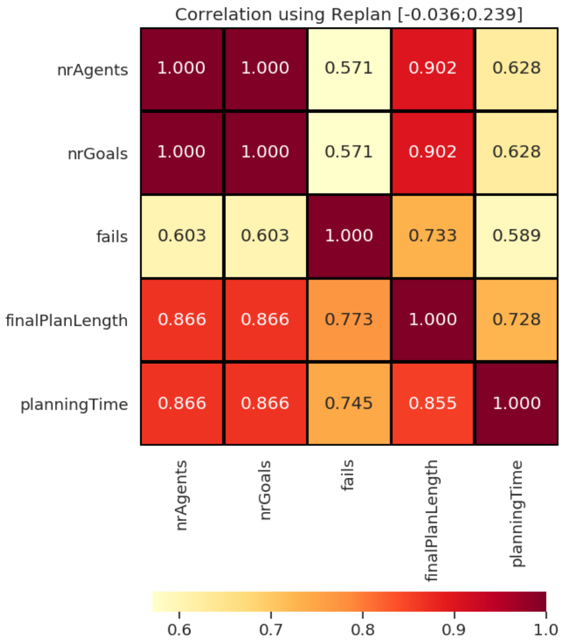
Regarding the logistics domain (Figure 4.10(b)), the greatest difference, 0.351, was related to the number of agents and final plan length. Moreover, at least 40% of the values (8 and 9 in 20), Spearman's method returned a correlation value higher than Pearson's one (red cells in Figure 4.10 (b)), indicating that the relationship between variables is better described by a non-linear method because of the presence of outliers in the metrics.

Considering the Driverlog domain (Figure 4.10(c)), the only relationship that can be considered is the one between planning time and other variables, when the replanning strategy is used. In other cases, the difference values are less than 0.1 and indicate that the strategies only affect the planning time metric.

[10]The cells in diagonal are not considered because their correlation values are always 1.0.



(a) Correlation using repairing.



(b) Correlation using replanning.

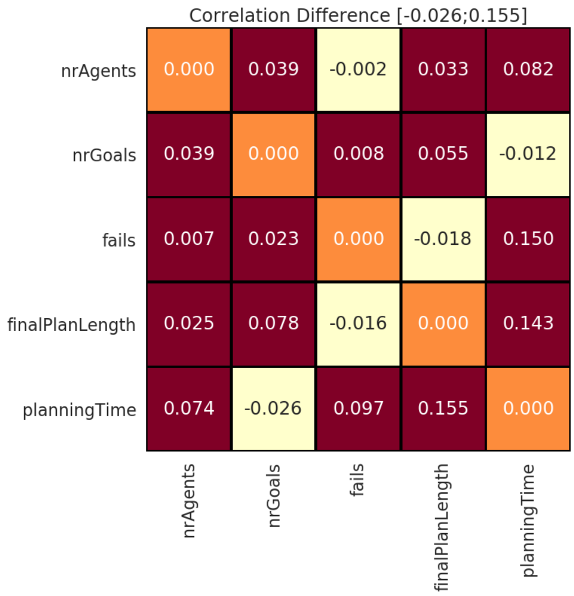
Figure (4.9) Correlation in the Driverlog domain.

At last, it is possible to highlight a pattern, namely, Spearman's correlation values are higher than Pearson's ones. Therefore, a partial conclusion is that the relationship between variables is better described by a non-linear method because outliers are likely to be present in output metrics.

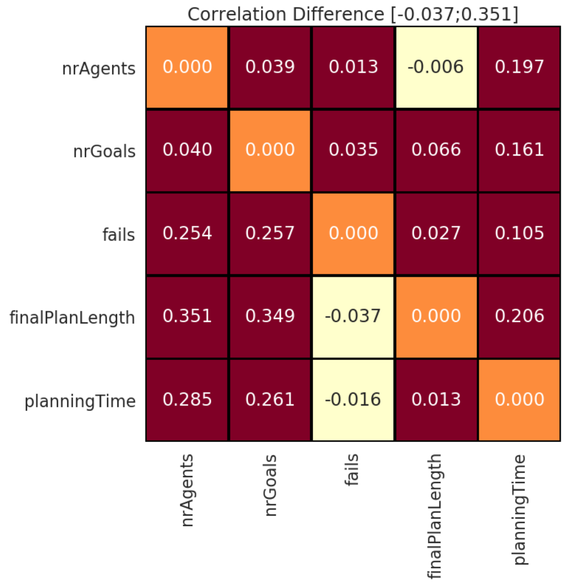
Moreover, the observation that the number of fails follows the same pattern of final plan length, that was made in the RQ1 investigation of logistics domains, is now justified by the correlation values. Regardless of the domain and recovery strategy, it is possible to highlight that the relation between fails and the final plan length is always strong, at least equal to 0.62 (replanning and Pearson's correlation in Figure 4.8(b)).

During the RQ2 investigation, we also checked how the correlation values are affected by the coupling level among agents. This level was defined from the public action (Definition 13) ratio in each experiment configuration. In this sense, we compared the relationship between final plan length $\times$ fails and planning time $\times$ fails. The correlated values of different pairs of variables were not checked because the scope of this work is the performance of recovery strategies after agents detect failures.

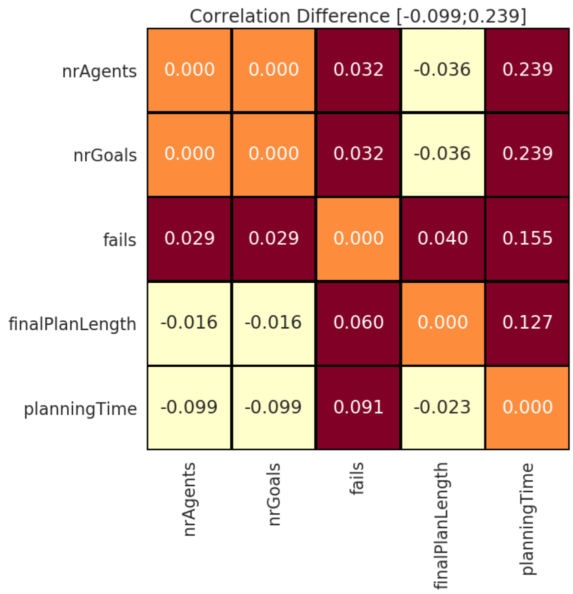
The evolution of the correlation values derived from both techniques is shown in Figure 4.11 where the horizontal axis describes the level of public actions in each domain (Table 4.1), red circles and blue crosses stand for the relationship values presented by using repairing and replanning, respectively. The final plan length metric is more strongly related to failure when the repairing strategy is applied (Figure 4.11 (a) and (b)). Regarding the planning time metric (Figure 4.11(c) and (d)), the replanning strategy presented a higher value in the extreme edges (Satellite and Driverlog). Specifically considering the logistics domain (public actions = 0.6), the relationship between variables had a precise pattern, namely, values from the repairing strategy were always higher than the ones from replanning.



(a) Satellite.



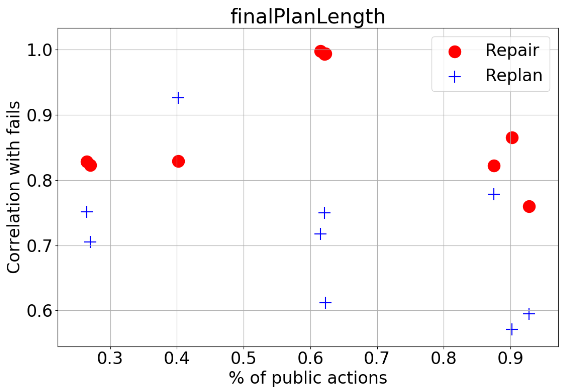
(b) Logistics



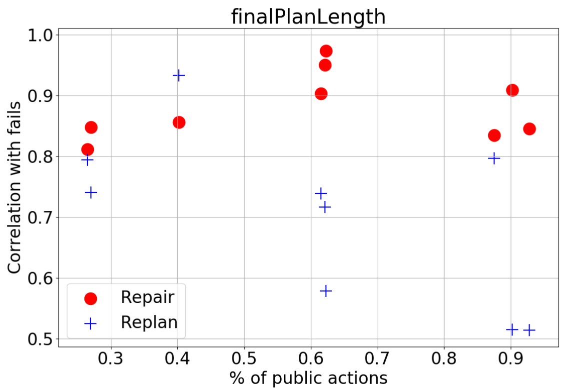
(c) Driverlog.

Figure (4.10) Correlation differences for Satellite, Logistics, and Driverlog domains.

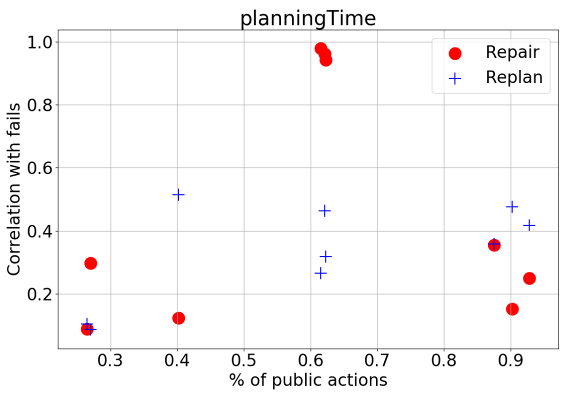
At last, the variation of the ratio of the public action causes an impact on the relationships between variables, mainly in an intermediate level of coupling, such as presented in the logistics domain experiments.



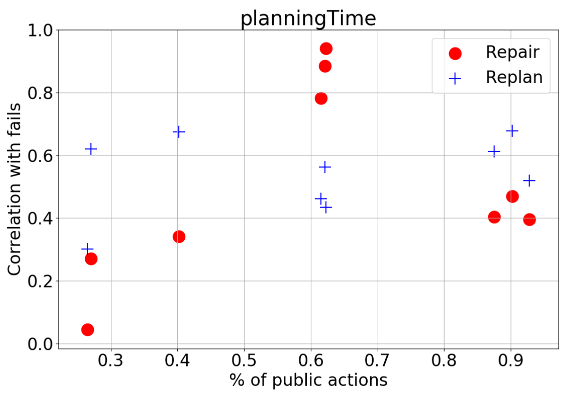
(a) Final plan length - Pearson's.



(b) Final plan length - Spearman's.



(c) Planning time - Pearson's.



(d) Planning time - Spearman's.

Figure (4.11) Correlation evolution by coupling level.

4.4 Final Discussion

We conducted several case studies and analyzed related research, and our findings suggest that repairing is faster than replanning when it comes to recovery strategies. However, replanning generates superior plans. We also discovered that the level of coupling among agents, as determined by the ratio of public action, impacts the relationship between variables. These results significantly impact time-sensitive dynamic environments, such as rescue operations. In scenarios where resource conservation is critical, replanning is preferred since it generates plans that avoid resource misuse, resulting in a better end plan. Regarding agents' interaction, both strategies work well in loosely-coupled domains where most actions are not public. However, in tightly connected domains, the choice between the two strategies depends on what needs to be preserved - time or resources.

We have proposed a MAP model that aims to recover plans in dynamic environments by providing replanning and repairing strategies together but in different stages. Our analysis led us to develop a configuration focusing on individual repairing and global replanning solutions. Our model, detailed in Chapter 5, includes decentralized and centralized components, and we evaluated its performance under various experiments. Additionally, we discussed this model in two articles published and evaluated by the academia [Moreira and Ralha, 2021a,b].

Chapter 5

Plan Recovery Process

Plan Recovery Process in Multi-agent Dynamic Environments. Full article published in Proc. of 18th Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO), p. 187-194, 2021.

Evaluation of Decision-making Strategies for Robots in Intralogistics Problems Using Multi-agent Planning. Full article published in Proc. of IEEE Congress on Evolutionary Computation (CEC), p. 1272-1279, 2021.

5.1 Introduction

MAP applications are applicable in ordinary to complex scenarios related to intralogistics problems such as warehousing and manufacturing. Intralogistics operations involve the planning, execution, and connection of all the stages of a company logistics process, such as automated guided vehicle movements, material flow, and interaction with humans as coworkers. There are two points to highlight in those scenarios. First, agents depend on or affect other agents under different levels regarding the interaction their action executions induce. Second, agents must react dynamically to system state and environment changes.

Consider the logistic task of delivering a package as an example. In an environment where no exogenous happen, a box placed in a position will maintain that state while another action is not executed. However, in a dynamic environment, this package state may be updated by an unexpected drop caused by a careless employee or wrong positioning. Moreover, an action may not be complete, although its conditions are held, because of a momentary error. Thus, planned actions will not stand because of that failure. Under such conditions, the agent committed to this failed action must trigger a plan recovery process to reestablish the required conditions to continue the plan execution.

In this context, an agent can be classified as an Autonomous Mobile Robot (AMR) because of the autonomy level derived from its abilities. Each robot can monitor its state to spot system failures. Also, a robot communicates and negotiates with other robots to adapt to changes in the operating environment. Therefore, the autonomy of an AMR provides the conditions of continuous decision-making whose goal is to react dynamically to changes and allow robots to work towards an uninterrupted commitment to fulfilling individual and global goals.

The recovery process is presented in related work following different strategies. First, under a centralized strategy, agents are strongly connected to a root point that accesses global information to achieve optimal performance. On a decentralized strategy, robots access only local information to search for local optimal solutions for the system. The centralized and decentralized strategies differ in performance according to the size and complexity of the system.

Regardless of the strategy, the research can be categorized according to the metric to be optimized, the method of analysis, and the application area. Most of the works are concerned with optimizing time and the number of actions by using simulation to evaluate performance in different scenarios De Ryck et al. [2020], Demesure et al. [2017], Fragapane et al. [2020], Hellmann et al. [2019], Kousi et al. [2019], Wan et al. [2017], Zhang et al. [2018].

The hypothesis defined in this work is that agents' autonomy in performing local decisions is better explored in environments with low levels of interaction. Here, we examined the third research question (RQ3): How can replanning and repairing be combined to enhance the performance of MAP models?

The rest of the document is structured as follows. In Section 5.2, we detail the proposed plan recovery process along with the simulation tool, while in Section 5.3, we describe the experiments and then discuss the results following the statistical evaluation method. At this point, we detail in Subsection 5.3.1 and 5.3.2 two different evaluation analyses published in two artciles [Moreira and Ralha, 2021a,b]. Finally, we present the conclusion in Section 5.4.

5.2 Plan Recovery Process

There are proposals in research that aim to address failures during the execution of a plan by using two different strategies for recovery: replanning and repairing. Although both approaches have pros and cons, it is still important to integrate both strategies into MAP models that consider the agents' abilities.

The recovery model underwent various changes during the project's development before its final design. The goal was to move away from a rigid setup that relied on a single strategy and instead test a combination of repairing and replanning.

In the first configuration of the simulation, there were agents with two roles: executor and coordinator. The coordinator was in charge of planning, while the executor agents reported any failures to the coordinator. The coordinator then adjusted the plans accordingly to undo the impact of the failure and coordinate new actions among the agents. The transitions of the simulation states are presented in Figure 5.1.

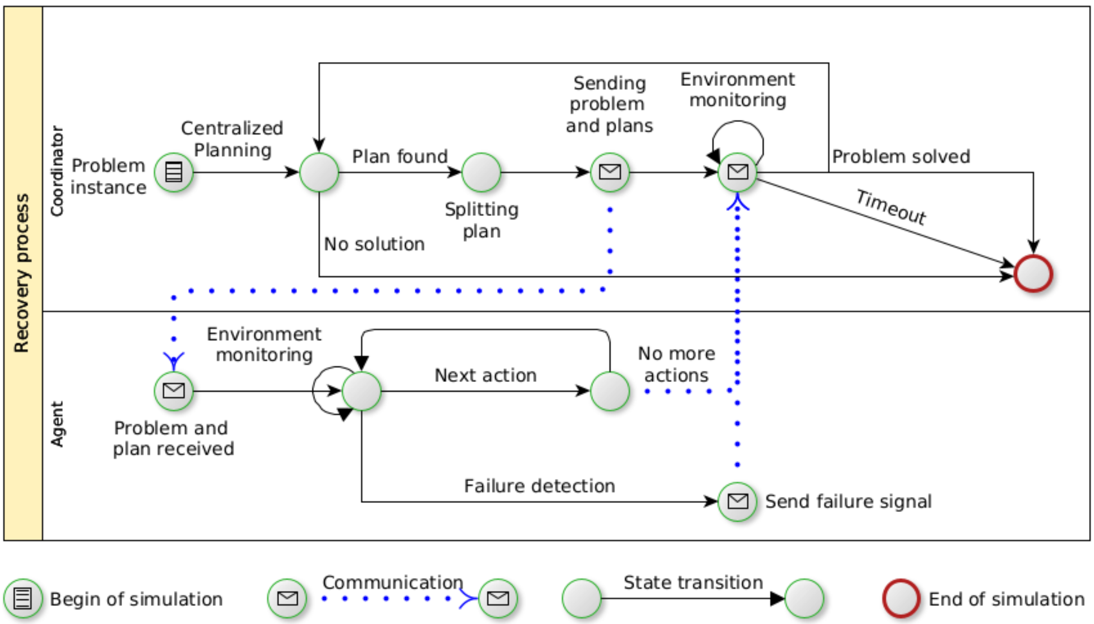


Figure (5.1) First configuration of recovery.

This setup involves the coordinator agent solely responsible for planning and implementing a single recovery strategy. However, this model needed more flexibility as it limits the available options to one strategy.

In the second configuration, the agents had a new responsibility: not just detecting and reporting failures but also planning and executing solutions for unexpected events. If an agent encountered an obstacle they could not handle, they would work with other agents to find a solution. If the failure remained, the agent in charge of coordination would take over the recovery process by creating a new plan involving all agents and coordinating the new sequence of actions.

A complex and decentralized design is necessary to simulate agents' plans, monitor the environment, and execute recovery processes. This requirement is because multiple agents may participate in the recovery process, as shown by the red arrows in Figure 5.2.

In the proposed process, there are two types of entities. The coordinator is responsible for handling a pair of files that represents the planning domain and problem. Those files are described according to MA-PDDL [Kovács, 2012b]. Then, the coordinator searches for a planning problem solution. This initial (and centralized) plan is transformed into single-agent plans. In such plans, actions are scheduled to a common step whether they can be carried out simultaneously by their executors. Then, these plans are sent to the coordinator to start an environment monitoring loop.

The second type is defined as agents that play planning and executing roles, which classifies the process, considering the agent distribution (Table 3.1), as planning by multiple agents. Therefore, each agent has autonomy to run a deliberation process, whenever it needs. Agents commit to executing the planned actions. Moreover, agents also perform coordination activities to guarantee an environment free of conflicts.

Regarding these premises, the dimensions are handled in staggered solutions when agents can try different strategies regarding their capabilities. The process design can be summarized as a three-phase sequence in that agents first try to recover from a failure using local planning. If in this phase is impossible to find a solution, the agent that detected the problem interacts with other agents asking for help. Whether some agents return positive answers, the caller will compare the solutions and choose the best (plan with the smallest number of actions) and coordinate with all agents the new execution condition. Otherwise, the caller agent triggers a centralized replanning process performed by a coordinator agent.

It is vital to highlight that the plan recovery process keeps the agent's privacy. Agents do not exchange their capabilities or their set of available actions. Indeed, they share the number of actions needed to solve the failure. Figure 5.2 presents the proposed plan recovery process. The pipeline of each entity type is described in individual lanes. Each process activity (i.e., the circle) is detailed in the sequence.

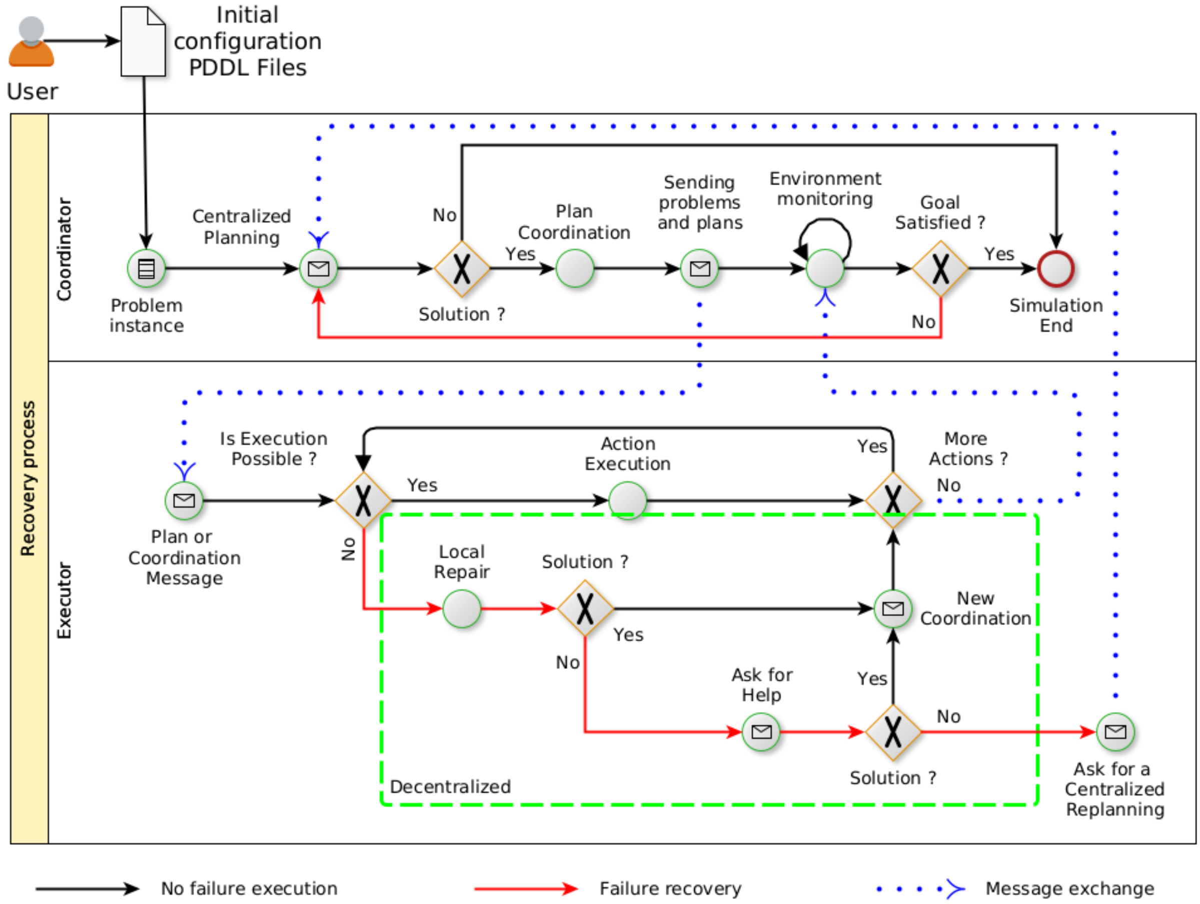


Figure (5.2) The plan recovery process.

The proposed method's connection between MAP and execution is related to Definition 5. After finding an initial plan, the actions are simulated by their executor agents in a parallel way, whenever possible, under the supervision of a central coordinator. So, after the execution of each action, the environment state is updated by the effects caused by those actions. For instance, let the initial state of the environment be I and a multi-agent plan π that meets the goals G, so the transitions between states will be as follows:

$$\pi = \begin{bmatrix} \alpha_1^1 & \alpha_2^1 & \varepsilon \\ \alpha_1^2 & \varepsilon & \alpha_2^2 \\ \varepsilon & \alpha_1^3 & \varepsilon \end{bmatrix}$$

$$run : I \xrightarrow{\alpha_1^1, \alpha_1^2} s_1 \xrightarrow{\alpha_2^1, \alpha_1^3} s_2 \xrightarrow{\alpha_2^2} s_n, \text{such that, } G \subseteq s_n$$

Problem instance

The planning domain and problem files are parsed to identify the initial state, goals, and operators. Then the available actions are computed, and the literals updated on the action's effects are listed. Those literals are decisive because they define the search space (relevant facts). The other literals are rigid facts because as not affected by action results. Both classifications are important since only relevant facts must remain after an encoding phase with the purpose of reducing the search space to exploration in the planning phase.

Centralized Planning

The first planning activity is carried out centralized by the coordinator. In this step, agents are viewed as resources. In addition, to compute the plan, this activity is crucial to select agents committed to execution. In this sense, centralized planning provides a solution that minimizes the number of actions required to turn the environment's initial state into the goal state. Thus, only the executors of those actions are granted to join further activities. If the planning problem has no solution, no execution is triggered, and no other activities are performed.

Plan Coordination

After defining the initial plan (centralized), the coordinator starts to build the multi-agent plan ρ. First, each agent action is split into individual lists. Then, a loop starts where the first action of each list is checked about the possibility to be carried out simultaneously from the initial state I. The actions that satisfy the conditions are popped from their agents' list. Otherwise, an empty action (idle state) is defined by the respective executor. Those actions are placed in a multi-agent plan and simulated to compute the next expected environment state (Equation 3.5). The loop finishes when every action of the initial plan is added to ρ.

Sending Problem and Plans

After the multi-agent plan definition, the coordinator sends the single-agent plans (ρ matrix rows) to their owners. Moreover, it sends a fragment of the planning problem formed only by the initial state, goals, and each agent's available actions. Thus, information privacy is preserved because no agent knows about other agents' capabilities.

Environment Monitoring

At this point, the coordinator starts monitoring the environment with a double concern. First, it controls the plan execution of every agent by receiving messages when they finish their tasks. When there are no more actions to perform, the coordinator runs its second verification, namely, goal satisfaction. This is a vital activity because of the possibility of exogenous events that impair the plan execution leading to failures. If the coordinator detects a problem, it starts a new centralized planning and follows with the next pipeline activities.

Plan or Coordination Message

The first activity in the agent's pipeline is receiving messages about plans and problems. Now, each agent knows its sequence of actions, which was checked and scheduled by to coordinator to provide an execution phase free of conflicts. This activity can also be triggered when another agent needs to update its plan (recovery) and then send a message to inform its efforts (number of actions) to overcome a failure. Hence, all the receivers adjust their plans, adding waiting steps (empty actions) as a new coordination phase. Note the coordinator is not warned by the sender as it only needs to control the agents' plans end.

Action Execution

The execution phase starts with the agent's evaluation of the conditions to run their actions. Each agent analyses the current environment state (s_t) and verifies the execution possibility of the following action. When the preconditions are held, the agent executes the action α, turning s_t to s_{t+1} according to Equation 3.1, and returns to the evaluation step. After the execution of the last action, agents send a message to the coordinator to inform him that the tasks are complete. At the evaluation step, the agent may detect a failure when the next action can not be carried out because of an error in the preconditions. Thus, it starts the recovery process.

Local Repair

The first step of the plan recovery process is performed by the agent that has just detected the failure. Then, it starts a local repair activity. The agent applies the repairing strategy and tries to find a solution that leads the current environment state to a condition where the preconditions of the failed actions are satisfied.

If the agent finds a possible solution, it updates its own plan by adding the actions at the beginning of its list, keeping the suffix of the plan from the failed action. In the sequence, the agent informs other executors that it needs to run more actions to bring the environment to the expected state. However, it is likely that the agent does not find a solution because of a lack of capabilities. Therefore, the next attempt is to ask other agents for help.

Ask for Help

When an agent asks for help, it shares the conditions the environment state needs to satisfy to guarantee the execution of the failed action. After other agents receive the message, they try a local repair to send back the results. To keep the information private, agents only share the number of actions they need to recover instead of sharing the actions themselves.

When just one agent returns a positive answer, this is the solution. However, in the presence of two or more answers, the agent evaluates the possibilities and chooses the best solution considering the smallest number of actions. Then, the selected executor is warned to update its plan by adding the solution. Other agents receive a coordination message with the number of actions the chosen agent needs to perform.

Ask for a Centralized Replanning

When the previous phases (Local Repair and Asking for Help) fail a solution, the agent that detected the failure sends a message asking the coordinator for centralized planning. While the earlier attempts applied the repairing strategy, now replanning is the solution.

As soon as the coordinator receives the message, it runs centralized replanning. However, different from the first round, the coordinator plans from the current state rather than the initial one. If it finds a solution to reach the goal from that state, it follows the pipeline (Plan Coordination - Sending Problem and Plans - Environment Monitoring). Otherwise, no further activity is carried out.

New Coordination

Agents may receive messages about a new coordination phase. These messages are sent in two conditions. First, when one agent runs a local repair activity and finds a solution. The other agents must adjust their plans regarding the new solution. Second, a local repair fails, but after asking for help, the agent receives one positive answer. In this case, all agents, but the chosen one, updates their plans to wait for the execution of that solution.

5.3 Experiments and Discussion

In this section, we detail the experiment setup and discuss the results published in different conference papers [Moreira and Ralha, 2021a,b]. To evaluate the plan recovery process, we used open-source software to build a simulation tool. As a solution for the parser and planning issues (problem instance, centralized planning, local repair, and ask for help), we decided to use the PDDL4J[1] JAVA library [Pellier and Fiorino, 2018]. The PDDL4J is based on the PDDL language and offers useful functions such as:

- parsing: the PDDL files are parsed and instantiated as objects of specific classes;
- pre-processing: operators are converted into ground actions based on the problem properties;
- pre-solving: checks whether the planning is solvable;
- immutable propositions are erased, keeping only the relevant facts;
- classical heuristics and planners.

Regarding the simulation engine, we chose the Repast Symphony[2] platform for supporting agent-based modeling and simulation [North et al., 2013]. The reason for using this tool was motivated by its (i) flexibility to code agents' behavior as JAVA methods; (ii) possibility to run batch simulations; (iii) ability to summarize results of the simulation in text files; and (iv) possibility to distribute simulation on different machines. The simulation tool is available in an online repository[3]

The case studies applied in the experiments are from two sources. First, we evaluated the domains and problems used in the CoDMAP, carried out with the workshop on Distributed and Multi-agent Planning (DMAP) at the ICAPS 2015. Second, we used an intralogistics scenario.

We carried all experiments out in a single computer with an Intel Core i7-10510U CPU and 16 GB RAM. The operational system was Ubuntu 20.04.1 LTS 64-bit. At last, the FF planner provided within the PDDL4J library was used in all experiments [Hoffmann and Nebel, 2001].

5.3.1 Domains and Experiment Setup

Regarding the most used case studies described in the related work, the domains chosen from CoDMAP were Satellite and Logistics as cited in Section 4.2.1 and Taxi:

- Taxi - related to transport issues in a city where agents can be passengers and taxis. Each taxi can transport only one passenger from the location it stays and only to a free drop-off location. A taxi can move only between connected locations.

[1]https://github.com/pellierd/pddl4j
[2]https://repast.github.io/
[3]https://gitlab.com/publicrepo/lcmap-de

The experiments were carried out under multiple conditions. The failure probabilities varied from 0.1 to 0.9 following a 0.1 step. Each case study had three different configurations simulated 30 times. We set the configurations according to the public actions ratio, simulating problems with diverse levels of agent coupling. Under those conditions, the experiments were run by 2,430 simulations. The setup description regarding the number of agents, goals, and actions in each domain is summarized in Table 5.1. The values in cells stand for the minimum and maximum values.

Table (5.1) Experiment setup description.

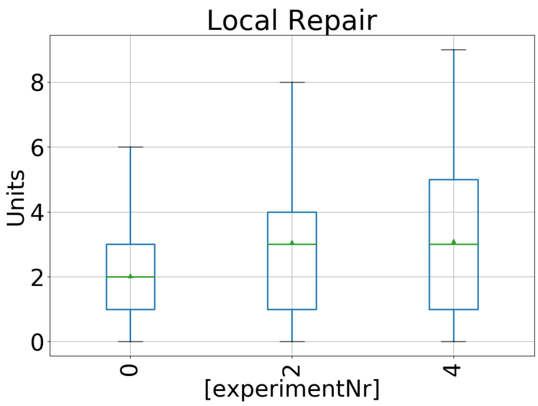
Domain	Agents	Goals	Actions	Public (%)
Satellite	3; 5	6; 10	497; 1473	26.2; 40.4
Logistics	3; 5	6; 10	78; 308	61.5; 62.3
Taxi	4; 7	4; 7	28; 126	100

Results and Discussion

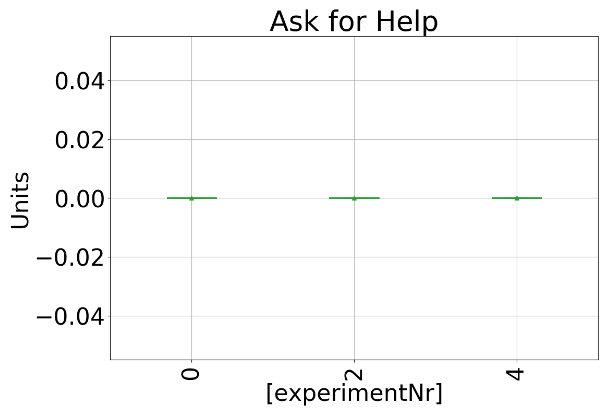
The plan recovery process was evaluated regarding three metrics: planning time, final plan length, and message exchange. The case studies are classified into three groups according to the agents' coupling level. The problems from the Satellite, Logistics, and Taxi were labeled as loosely, intermediate, and tightly coupled domains, respectively. The results of each group are discussed individually, and a global evaluation is presented.

The first important step towards the evaluation of the results is the definition of how many times each recovery activity (local repair, ask for help, and centralized planning) was performed. The information about recovery activities, planning time, final plan length, and the message is presented in Figures 5.3 to 5.5.

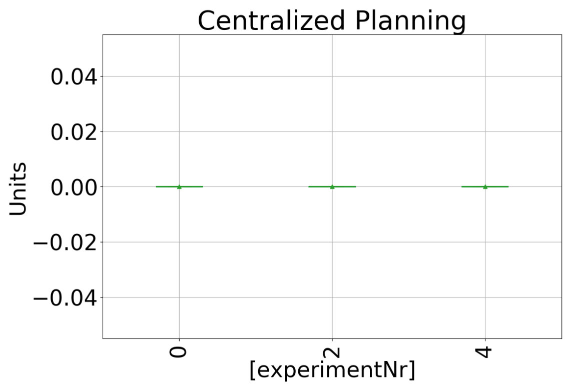
Regarding the loosely-coupled domain simulations, the recovery activity was restricted to local repair (Figure 5.3). This behavior is justified as agents carry out, at most, public actions. The highest coupling level is 40.2% (Table 5.1). Hence, agents do not depend on or affect other agents. Thus, agents do not need to interact to solve failures. Notably, the agents did not seek assistance from other agents or initiate a centralized replanning. This is evident from the zero values that are illustrated in Figures 5.3(b) and 5.3(c). Therefore, the motivation hypothesis that agents' autonomy in performing local repair is better explored in environments with low levels of interaction is accepted.



(a) Local repair.



(b) Ask for help.



(c) Centralized planning

Figure (5.3) Recovery in loosely-coupled domains.

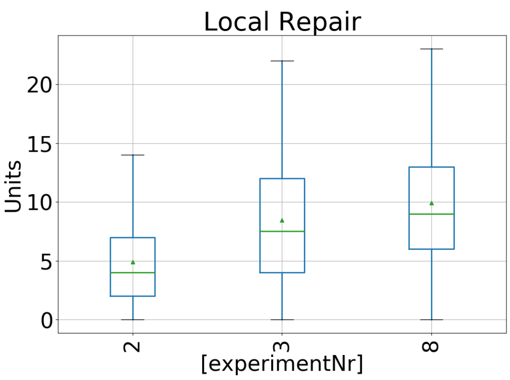
Regarding the intermediate-coupled domain simulations, agents could not solve the failures using local repair activities. Indeed, they need to interact by asking for help (Section 5.2, Ask for Help). Sometimes, they also need to request centralized planning. Those recovery strategies are shown in Figure 5.4. The reason for this different behavior in the Logistics domain simulations is inherited from the set of available actions. Thus, agents do not have all the capabilities required to solve a problem. Hence, they need to cooperate in the search for a solution. It is worth noting that as the level of interaction between agents grows, there is a corresponding rise in requests for mutual assistance among agents (Ask for Help) and centralized replanning. The need for such interaction is emphasized by the levels charts depicted in Figure 5.4, which surpasses the levels outlined in Figure 5.3.

Regarding the tightly-coupled domain simulations, agents must request more often for centralized replanning, as shown in Figure 5.5. The ratio of public actions in the available actions was 100% (Table 5.1). Thus, every action either depends on or affects other actions. As expected, the failure solution is achieved by a centralized replanning activity where all actions are available in a common process. Sometimes, certain failures can be fixed by local repairs, however, there are situations where a complete system reorganization is necessary. In scenarios where there is a high level of coupling, individual agents may not have the ability to resolve failures on their own because of their limited set of available actions. They must collaborate to find solutions. In these cases, a centralized replanning approach is recommended as a suitable strategy.

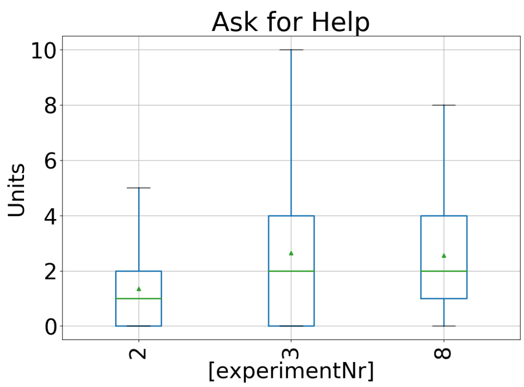
The execution of higher recovery strategies increases from loose to tight domains. The analysis of the averages of the results from each strategy is detailed in Table 5.2 by domains, which demonstrates that: (i) the frequency of local repair calls in tightly-coupled domains is 3.11× bigger than in loosely; (ii) ask for help and centralized replanning activities are carried out 1.73× and 8.68× more often in tightly than in intermediate domains. Therefore, the coupling level among agents increases the complexity of the recovery process.

Table (5.2) Strategies calls (means) by domains.

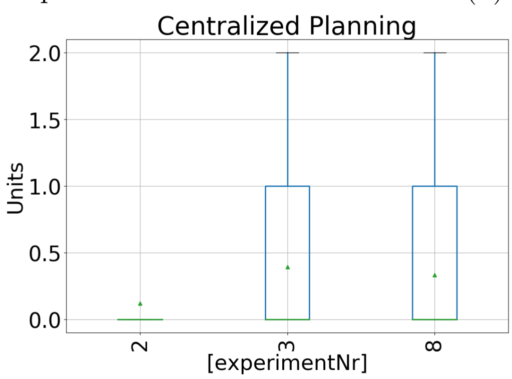
Strategy	Loosely	Intermediate	Tightly
Local Repair	2.73	7.76	8.5
Ask for Help	0	2.18	3.78
Centralized Planning	0	0.28	2.43



(a) Local Repair.



(b) Ask for Help.

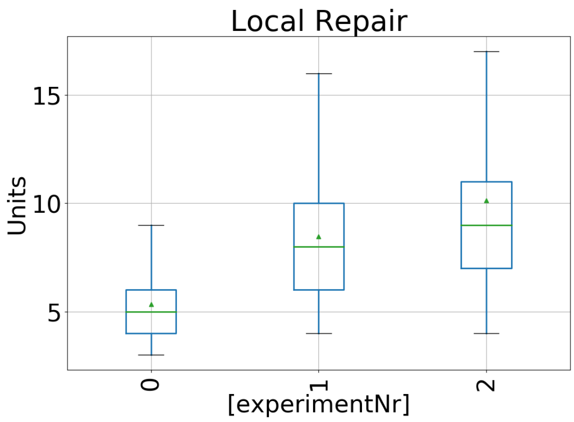


(c) Centralized Planning

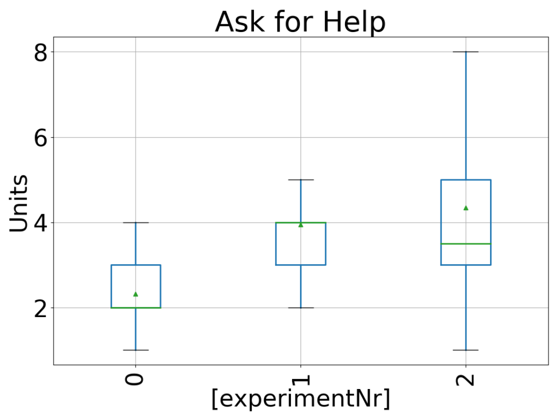
Figure (5.4) Recovery in intermediate-coupled domains.

Conclusions can be drawn from evaluating the metrics shown in Figures 5.6. Regarding the final plan length (Figure 5.6(a)), although the intermediate-coupled domain simulations have complexity levels (number of public actions) lower than the Taxi domain, they presented final plans with a higher number of actions. The Logistics domain simulation applied to repair (Local Repair and Ask for Help) more than other simulations, which is the reason for bigger plans. The main drawback of this strategy derives from the fact that repairing tends to build bigger plans. The Taxi domains highlighted plans with fewer actions because of replanning strategy [Komenda et al., 2014].

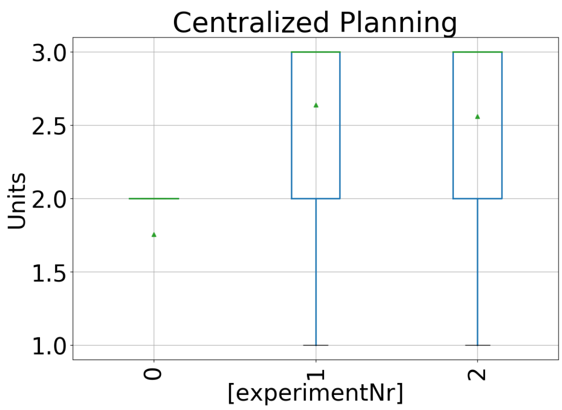
Regarding the planning time analysis (Figure 5.6(b)), loose domains showed irrelevant and small values when compared to the other groups where only local repair was needed and repairing strategy tends to be faster. The final planning time in Logistics was higher than the values of Taxi domains since the plans were bigger. Hence, more actions were likely to fail, and more recovery activities had to be performed.



(a) Local repair.



(b) Ask for help.



(c) Centralized planning

Figure (5.5) Recovery in tightly-coupled domains.

In situations where agents were partially connected, the logistics domain, recovering a plan took longer due to their attempts to solve issues independently. If they were unsuccessful, they triggered the centralized replanning. Both methods had their advantages and disadvantages. Agents added more actions to their plans if the issue could be repaired locally. However, if replanning was required, the planning time increased while the number of actions decreased. Therefore, the results of logistics experiments, where planning time and final plan length were the highest, can be attributed to the limitations of both strategies.

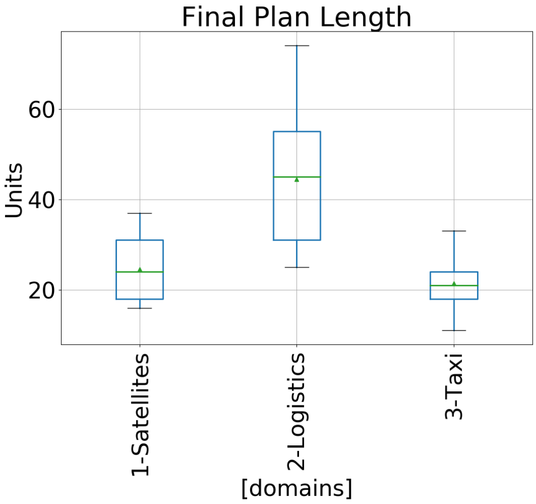
The message exchange (Figure 5.6(c)) highlights the interaction of agents after facing a failure. Since the loosely-coupled domains handle the problem with a local repair strategy, the agents only need to exchange messages to provide new execution coordination. However, in the intermediate and tightly-coupled domains, the number of exchanged messages tends to be higher than the first because agents ask for help and request centralized planning with more messages sent. Therefore, this is another evidence that the hypothesis that agents' autonomy is better explored in environments with low levels of interaction can be accepted.

5.3.2 Intralogistics Scenarios

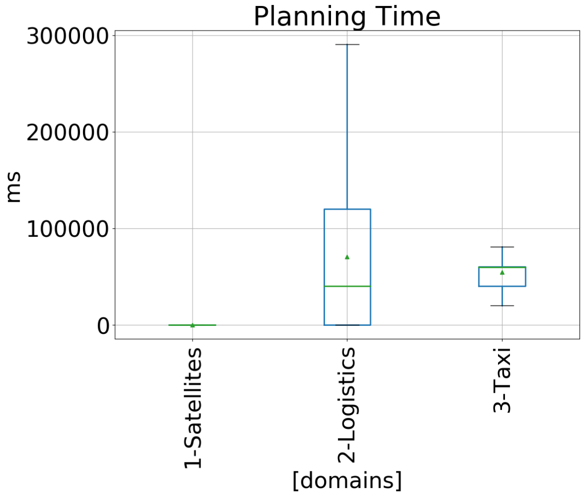
In intralogistics scenarios, a group of robots runs a sequence of actions. The ability to perform those actions may be familiar to all robots or restricted to a subset. Namely, AMR can be part of homogeneous or heterogeneous fleets, respectively [Lee and Murray, 2019].

An intralogistics scenario is illustrated in Figure 5.7 where robots run warehousing operations. A picker robot selects an object in the inventory area, and a carrier independently loads and transports that object to a delivery area where a packer boxes the product to be sent to the customer [Azadeh et al., 2019].

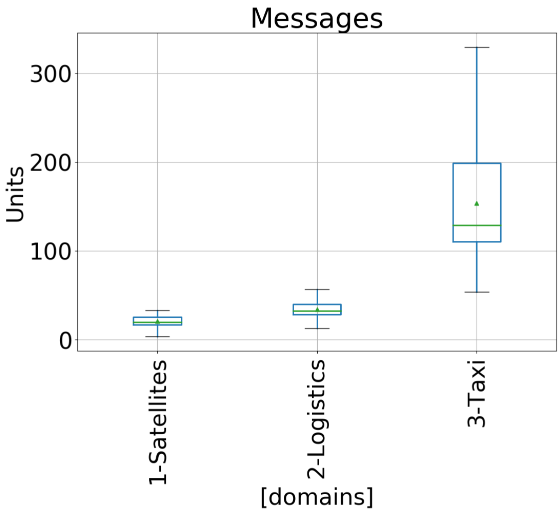
The organization of the AMRs highlights three attributes in robotic applications: decentralized control, scalability, and robustness. Robustness is required in real-world and dynamic environments because exogenous events may happen, and AMRs must be able to recover after failures caused by an object's accidental drop or by an error during loading operations. Thus, the level of control decentralization, especially in environments where robots might be affected by failures, is an important decision because it determines which parts of a system should be controlled in a centralized or decentralized and how AMRs will react after an exogenous event.



(a) Final plan length.



(b) Planning time.



(c) Message exchange.

Figure (5.6) Message exchange.

There were two scenarios to simulate and evaluate under which conditions AMRs might run a centralized or decentralized decision-making process. In the centralized process, the coordinator runs the replanning recovery strategy. In the decentralized, each AMR tries to repair its plan before asking for assistance from other agents.

The experiments consider multiple conditions. The failures were simulated in the following way. The effects of an action α could be undone following a continuous uniform distribution after its execution. So, whether an action failed, the number of propositions $eff(\alpha)$ that might be affected were returned to the same condition before the execution of the action, namely $pre(\alpha)$. In this way, we avoid inconsistencies in the state of the environment. The uniform distribution used in the failure simulation was 0.05.

We design two problems: one for homogeneous robots and another for heterogeneous ones. Heterogeneous robots are distributed equally among pickers, carriers, and packers (Figure 5.7). There were five different configurations for each type of problem, with a different number of robots. Table 5.3 presents the setup description, where the column labels have the following meanings. The type describes AMRs' nature regarding their set of actions. The experiment identifies the problem file used to select the simulation inputs and designate the experiments with unique values for each domain. AMRs, items, and actions detail the amount of AMRs, items to be delivered, and actions instantiated from the planning problem definition, respectively. The public stands for the percentage of public actions according to Definition 13.

Table (5.3) Setup description.

Type	Experiment	AMRs	Items	Actions	Public
Heterogeneous	0	3	5	39	0.82%
	1	6	5	88	0.954%
	2	9	5	147	0.959%
	3	12	5	216	0.962%
	4	15	5	295	0.966%
Homogeneous	0	3	5	162	0.963%
	1	6	5	414	0.971%
	2	9	5	738	0.976%
	3	12	5	1164	0.98%
	4	15	5	1680	0.982%

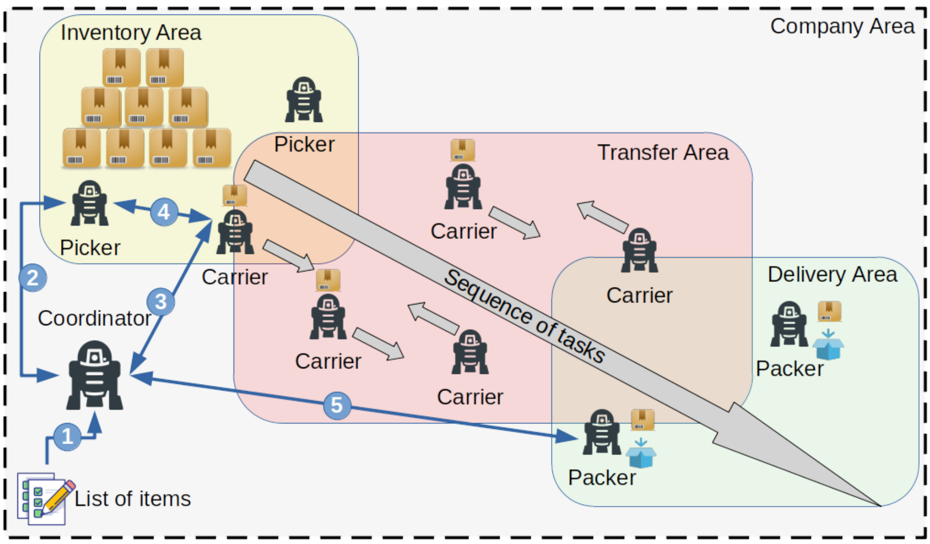


Figure (5.7) Intralogistics scenario. Blue arrows stand for message exchange. Numbers detail message to order.

The experiments were designed considering different levels of public actions because, from this subset of planning domains and problems, it was possible to run simulations with distinct coordination complexity because the interaction among AMRs is defined by the dependencies and effects of their actions.

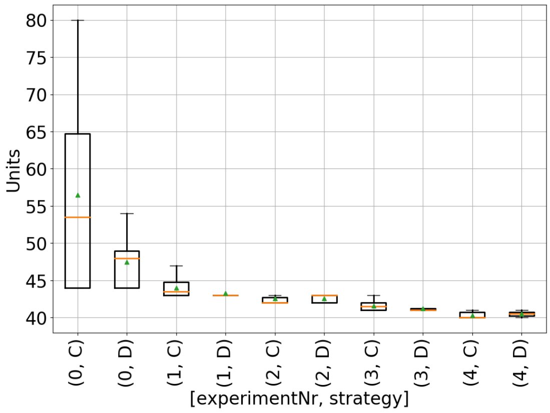
Results and Discussion

The decision-making process was evaluated regarding three metrics. The final plan length represents the number of actions that robots carried until all items were delivered. The planning time represents the sum of all time spent by the coordinator to build each AMR's initial plan and the AMRs to recover after a failure. At last, the message exchange summarizes the communication activities among robots to receive plans or coordinate their actions.

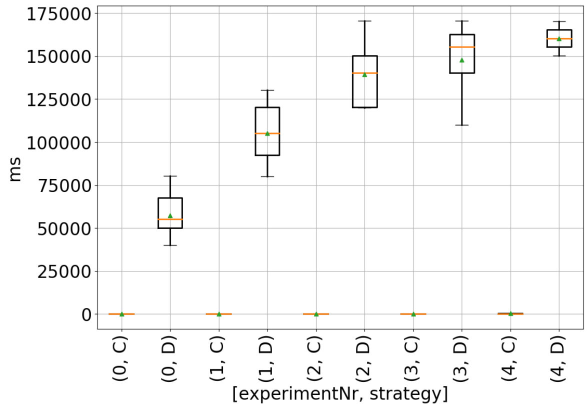
The experiments were sorted according to the agents' coupling level (Definition 13), which was represented in the column labeled as Public in Table 5.3. The results of each group are discussed individually, and a global evaluation is presented. The information about the evaluation metric is shown in Figures 5.8 to 5.10 using boxplot charts. The values in the horizontal axis follow the notation: numbers represent different configurations of each experiment; (C) centralized and (D) stand for decentralized decision-making strategy; (Het) and (Hom) indicate heterogeneous and homogeneous robots, respectively.

Regarding the heterogeneous problems, there was a balance between the final plan length (Figure 5.8a) and the planning time (Figure 5.8b). The centralized strategy required the AMRs to carry out more actions than the decentralized to fulfill their goals in all the experiments. This performance was caused by the centralized strategy coordinator favoring a quick response to failures instead of a better, more minor recovery sequence. On the other hand, the robot, under a decentralized strategy, was able to find better solutions, however, spending more time planning. The number of messages was more significant in the decentralized decision-making process because robots were required to coordinate more intensely than in the other strategy (Figure 5.8c). Moreover, the final plan length experimented with decreased because more actions could be carried out in parallel by more robots.

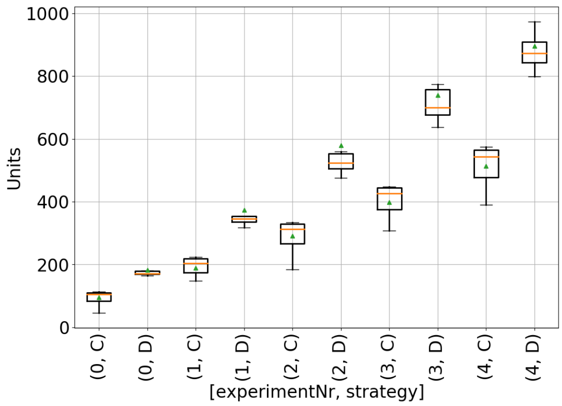
In the homogeneous problems, all AMRs had the same set of abilities and the same set of available actions. Hence, the ratio of public actions was higher than in the heterogeneous problems in every configuration because the robots could affect other AMRs more often after the execution of their actions (Figure 5.9a). Thus, the homogeneity among the robots changed the performance of the decision-making process. First, the planning time (Figure 5.9b) had lower levels than the values illustrated in heterogeneous problems (Figure 5.8a) because every AMR had all the conditions to recover after a failure by itself without requiring aid for other robots. Consequently, the AMRs also had a lower level of message exchange because they could solve their problems locally, and the interactions were required only to coordinate new actions (Figure 5.9c). It is also important to highlight that the highest number of messages was similar to the smallest amount in the heterogeneous problems. Therefore, environments with homogeneous AMRs tend to be more economical regarding messages. That is important in scenarios under bad communication conditions.



(a) Final plan length.



(b) Planning time.



(c) Message exchange.

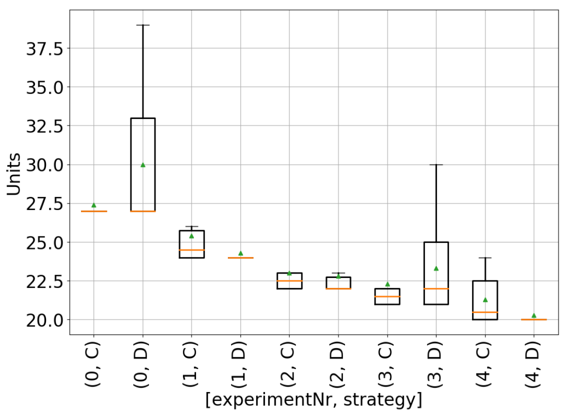
Figure (5.8) Heterogeneous AMRs.

At last, centralized and decentralized strategies were evaluated regarding a global approach that guided the evaluation of the hypothesis that motivated this work.

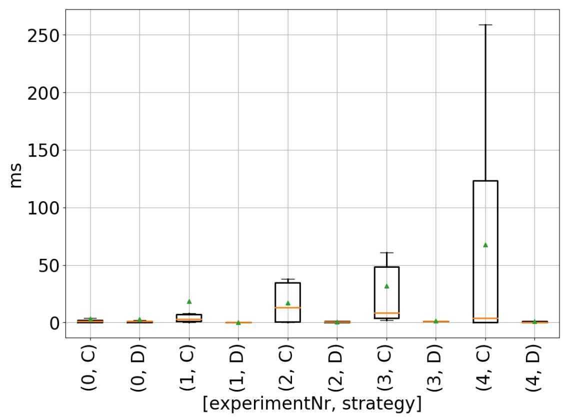
The final plan length (Figure 5.10a) was more strongly affected by the nature of the AMRs, heterogeneous or homogeneous, than by the level of public actions. In other words, heterogeneous robots had the worst performance, nevertheless, the strategy applied. However, homogeneous AMRs could fulfill their tasks with fewer actions under a decentralized strategy than the centralized. Thus, the lower level of public action is not relevant to the choice of the decision-making process. Furthermore, this level is important when robots must be concerned about their resources impaired by long-term operations, such as battery and fuel.

In the meantime, the decision-making process was affected differently by the strategy regarding the planning time and message metrics. The centralized strategy performed quicker (Figure 5.10b) in both types of problems than the decentralized. Moreover, the latter was strongly impaired in scenarios with heterogeneous robots. The same behavior was noticed regarding the number of messages (Figure 5.10c). Heterogeneous robots require more coordination messages than homogeneous AMRs.

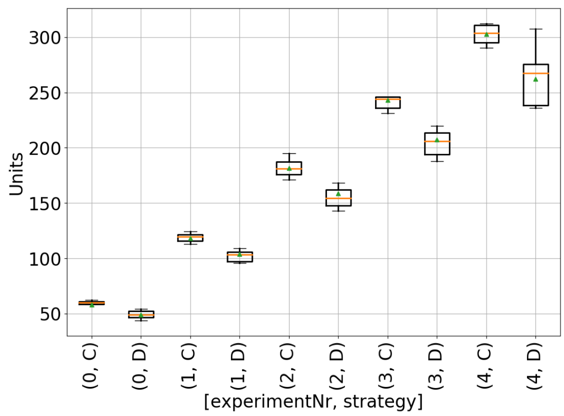
Therefore, the motivation hypothesis that AMR's autonomy in performing local decisions is better explored in environments with low levels of interaction is accepted. The homogeneous AMRs had higher levels of autonomy than the heterogeneous because of the equality of their abilities. Hence, scenarios where homogeneous robots worked under a decentralization decision-making process highlighted the best metric values of the experiments.



(a) Final plan length.



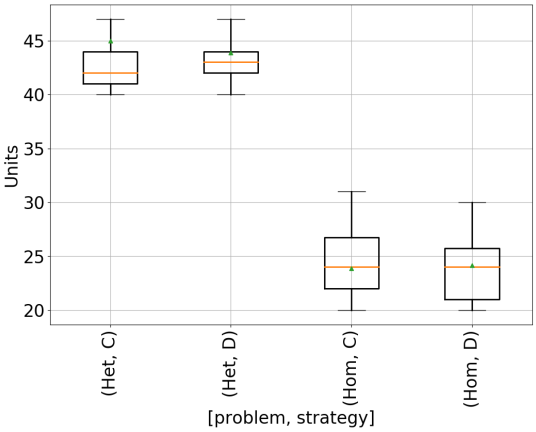
(b) Planning time.



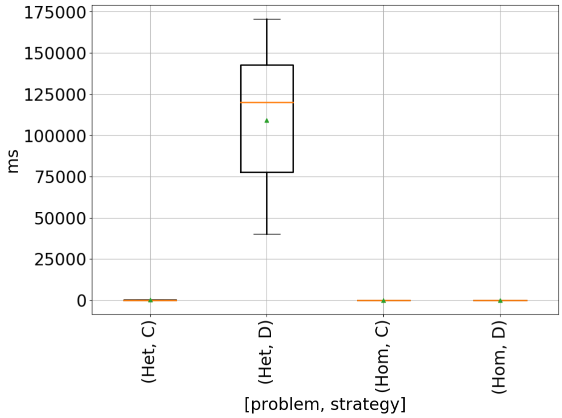
(c) Message exchange.

Figure (5.9) Homogeneous AMRs.

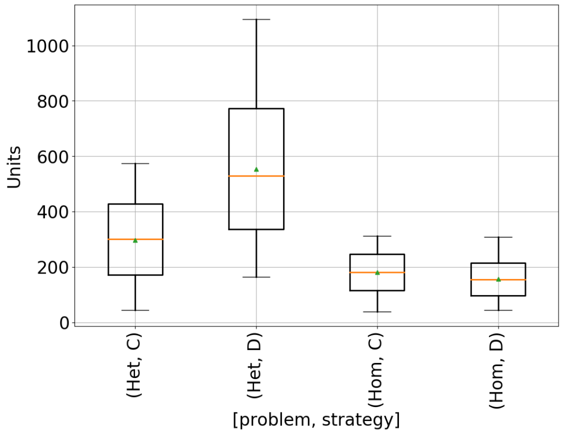
When making decisions in intralogistics, the type of AMRs used is meaningful. The performance of these robots can be significantly affected by whether they are homogeneous or heterogeneous. It is also possible to determine the level of interaction between the robots by calculating the ratio of public actions in their set of actions.



(a) Final plan length.



(b) Planning time.



(c) Message exchange.

Figure (5.10) Global evaluation.

5.3.3 RQ3 Investigation

The experiments yielded evidence to address the third research question (RQ3): How can replanning and repairing be combined to enhance the performance of MAP models?

We aim to design a model that offers agents fast and local solutions whenever they encounter a problem. To achieve this, we created a model that exclusively focuses on using repair strategies during individual recovery. In addition, if an agent faces a problem it cannot solve, it will seek help from another agent. The assisting agent will then initiate a repair process to address the issue as quickly as possible. As all agents in the environment may be asked for help, the requesting agent may receive multiple solutions. It will choose the solution that requires the fewest new actions. Furthermore, we ensured the model could handle complex failures requiring multiple agent updates. To achieve this, we implemented a centralized replanning model that provides solutions for these failures.

These decisions were based on the performance evaluation discussed in Chapter 4. Our unique approach combines replanning and repairing techniques, resulting in a staggered solution. The solution involves local repairs, seeking assistance, and centralized replanning, making it stand out from similar methods. We believe this configuration is valid based on the results discussed in Section 5.3.1. Our experiments on the CoDMAP domains show that the repairing strategy maintains planning time while resulting in plans with more actions. However, it minimizes the need for coordination messages.

In Section 5.3.2, we found that the centralized strategy outperformed the decentralized strategy in problems with homogeneous and heterogeneous agents. Additionally, we observed that heterogeneous agents require more coordination messages than homogeneous ones, indicating a greater need for coordination. These results suggest that nature significantly impacts agents' performance when evaluated from the perspective of homogeneity or heterogeneity.

Thus, to answer RQ3 we had to weigh the advantages and disadvantages of different recovery strategies. As a result, we created a model that uses local repairs to save time on planning. We also chose a centralized approach for replanning to avoid the need for message exchange.

5.4 Final Discussion

Through case studies with varying levels of interaction, we analyzed metrics related to final length, planning time, and message exchange in simulations. Our findings suggest that agents' autonomy in performing local repair is better explored in low interaction environments and that the coupling level among agents increases complexity in recovery and planning. In a real-world intralogistics scenario, we evaluated the decision-making process of robots with different levels of interaction. We found that the nature of the AMRs is more critical to performance than the level of coupling inherited from the ratio of public actions.

The contributions include a three-phase plan recovery process, a benchmark simulation tool, and a statistically robust evaluation method for MAP in dynamic environments.

Chapter 6

Conclusion

This thesis conducted a comprehensive literature review of the MAP research area. By examining various works, the review identifies key authors and organizations that have substantially contributed to the field and the most cited documents. This analysis provides valuable information on the key figures and institutions driving research in MAP, guiding future collaborations, identifying potential research partners, and helping researchers stay updated with the latest advancements in the field.

Moreover, the literature review provides insights into MAP concepts, techniques, and challenges, making it valuable for researchers and practitioners interested in the field. Readers can understand the current state-of-the-art in the field through this review.

In this thesis, we achieved the proposed objective of presenting a comprehensive analysis of plan recovery strategies in dynamic environments proposing a model that explores replan and repair in a complementary way. We proposed a method that extrapolated average values and a MAP model that provides synergy between repairing and replanning.

The first contribution was evaluating the performance of plan recovery strategies under different scenarios and varying coupling levels. Unlike other approaches, we proposed a method that used standard deviation, T-Test inferential hypothesis test, and correlation analysis to check the performance of recovery strategies. Using various evaluation methods in our case studies, which were grounded in relevant research and problem areas, we have concluded, backed by statistically significant evidence, that repairing is a faster solution than replanning. However, it is worth noting that the latter approach does result in better plans overall. Additionally, we have demonstrated that the coupling level between agents, influenced by the ratio of public action, notably impacts the relationship between variables.

We have drawn some conclusions regarding RQ1 and RQ2 that address the missing analysis in the literature on the correlation between the metrics that affect the performance of recovery strategies. In simpler terms, we have discussed how the coupling level affects the correlations between these metrics.

Based on the results and discussions, the replanning strategy is slower when dealing with tightly coupled problems. This approach leads to better plans, however, with longer planning times. On the other hand, in loosely coupled scenarios, like those seen in Satellite domains, the strategies showed similar performance.

We found that changes in the ratio of public action can affect the relationships between variables, mainly when there is an intermediate coupling level. Therefore, we checked how the coupling level among agents affects the correlation values. In this sense, we compared the relationship between final plan length $\times$ fails and planning time $\times$ fails.

The evolution of the correlation values derived from both techniques was discussed, considering different domains and levels of coupling. The final plan length metric was more strongly related to failure when applying the repairing strategy. Regarding the planning time metric, the replanning strategy presented a higher value in the extreme edges, from domains with fewer public actions or with a more significant level of them. Explicitly considering the logistics domain (public actions = 0.6), the relationship between variables had a precise pattern. Values from the repairing strategy were consistently higher than the ones from replanning.

Regarding RQ3, we designed a three-phase multi-agent model to work in dynamic environments. This model focuses on planning, acting, and monitoring the environment while combining repairing and replanning.

In this sense, we examined the recovery strategies' benefits and drawbacks. Our findings led us to develop a model that utilizes local repairs to save time in the planning process. Additionally, we opted for a centralized approach to replanning to eliminate the need for message exchange. In contrast to other models, it examines the autonomy level of agents to enhance the recovery process in the event of failures. The model has undergone evaluation in different scenarios, including centralized and distributed planning, with varying coupling levels.

To summarize, we highlight the following contributions of this thesis:

- a comprehensive literature review of MAP research area;
- statistically significant analysis of the performance of replanning and repairing strategies;
- study of the impact of variables that define the environment over performance metrics;
- combination of replanning and repairing in a MAP model applicable to dynamic environments.

Future Work

As upcoming research, we suggest investigating additional failures, including removing agents. This type of failure requires investigation beyond the analysis of recovery strategies. Since the coupling level is affected by an agent removal, the environment's conditions can change, making a recovery strategy more efficient than it was at the beginning of the plan execution.

Therefore, it might be interesting to evaluate the performance of strategies at runtime. To achieve this, regression models to predict and compare the performance of strategies in run-time based on the current environment is desirable. This approach can lead to a more efficient and effective recovery strategy that adapts to environmental changes. In this sense, one option is to gather a collection of regression models created from previous executions and apply them during runtime.

Another suggestion for future work is to update the plan even if no failure is detected. In this sense, exploring the concept of opportunity proposed by [Borrajo and Fernández, 2019] might be interesting. In dynamic environments, some changes might occur and turn into possible new updates to improve the current plans. Therefore, those changes might introduce conditions to update the plan to a better configuration than at the beginning of its execution.

Furthermore, there is room for improvement in evaluating the plan recovery process in more complex scenarios. The Command Control applications provide challenging situations ideal for experimentation and evaluation. The planning, coordination, and execution components of the MAP, combined with environmental monitoring, can assist in guiding the Observe-Orient-Act-Decide cycle.

References

G. Alabi. Bradford's law and its application. *International Library Review*, 11(1):151–158, 1979. 11

F. Amato, F. Moscato, V. Moscato, F. Pascale, and A. Picariello. An agent-based approach for recommending cultural tours. *Pattern Recognition Letters*, 131, 03 2020. 27

K. Azadeh, D. Roy, and R. De Koster. Design, modeling, and analysis of vertical robotic storage and retrieval systems. *Transportation Science*, 53, 2019. 83

M. Babli, Óscar Sapena, and E. Onaindia. Plan commitment: Replanning versus plan repair. *Engineering Applications of Artificial Intelligence*, 123:106275, 2023. x, 31, 35

C. Backstrom. Computational aspects of reordering plans. *Journal of Artificial Intelligence Research*, 9:99–137, 1998. 1

J. Banfi and M. Campbell. High-level path planning in hostile dynamic environments. In *Proceedings of the 18^{th} International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, page 1799–1801. IFAAMAS, 2019. 33, 35

D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27 (4):819–840, 2002. ISSN 0364-765X. 24

S.-O. Bezrucav, G. Canal, A. Coles, M. Cashmore, and B. Corves. Towards Automatic State Recovery for Replanning. In *Proceedings of the 32^{nd} International Conference on Automated Planning and Scheduling (ICAPS) Workshop on Integrated Planning, Acting, and Execution (IntEx)*, 2022. 31, 35

D. Borrajo and S. Fernández. Efficient approaches for multi-agent planning. *Knowledge and Information Systems*, 58(2):425–479, Feb 2019. 45, 93

D. Borrajo and M. Veloso. Computing opportunities to augment plans for novel replanning during execution. In *Proceedings of the 31^{st} International Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, pages 51–55, 2021. 32, 35

D. Borrajo, S. Kambhampati, A. Oddi, and S. Fratini, editors. *Proceedings of the 23^{rd} International Conference on Automated Planning and Scheduling (ICAPS)*, 2013. AAAI. 24

C. Boutilier and R. I. Brafman. Partial-order planning with concurrent interacting actions. *Journal of Artificial Intelligence Research*, 14:105–136, 2001. 24

R. I. Brafman. A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the* 24^{th} *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1530–1536. AAAI Press, 2015. 23, 29

R. I. Brafman and C. Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *Proceedings of the* 18^{th} *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 28–35. AAAI, 2008. 24, 42

R. I. Brafman and C. Domshlak. On the complexity of planning for agent teams and its implications for single agent planning. *Artificial Intelligence.*, 198:52–71, 2013. 24

M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19:297–331, 12 2009. 24

T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence.*, 69(1-2):165–204, 1994. 24

R. Cardoso and A. Ferrando. A review of agent-based programming for multi-agent systems. *Computers*, 10:16, 2021. 29

M. Cashmore, A. Coles, B. Cserna, E. Karpas, D. Magazzeni, and W. Ruml. Replanning for situated robots. In *Proceedings of the* 21^{st}*International Conference on Automated Planning and Scheduling (ICAPS)*, pages 665–673. AAAI Press, 2019. x, 33, 35, 45, 50, 58

S. S. Chouhan and R. Niyogi. MAPJA: Multi-agent planning with joint actions. *Applied Intelligence*, 47(4):1044–1058, 2017. ix, 26, 50

L. Chrpa, J. Gemrot, and M. Pilat. Planning and acting with non-deterministic events: Navigating between safe states. *Proceedings of the* 34^{th} *Conference on Artificial Intelligence*, 34(06):9802–9809, 2020a. 50

L. Chrpa, J. Gemrot, and M. Pilát. Planning and acting with non-deterministic events: Navigating between safe states. In *Proceedings of the* 34^{th} *Conference on Artificial Intelligence*, pages 9802–9809. AAAI Press, 2020b. 33, 35

I. Cil and M. Mala. A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare. *Expert Systems with Applications*, 37(2): 1331 – 1343, 2010. 3

Clarivate. Web of science platform - clarivate. `https://clarivate.com/products/scientific-and-academic-research/research-discovery-and-workflow-solutions/webofscience-platform/`, 2023. Accessed: 2023-04-20. 9

M. C. Cooper, A. Herzig, F. Maffre, F. Maris, and P. Régnier. A simple account of multi-agent epistemic planning. In *Proceedings of the* 22^{nd} *European Conference on Artificial Intelligence (ECAI)*, pages 193–201, 2016. 29

M. Crosby, M. Rovatsos, and R. P. A. Petrick. Automated agent decomposition for classical planning. In *Proceedings of the* 23^{rd} *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2013. 24

M. Crosby, A. Jonsson, and M. Rovatsos. A Single-Agent Approach to Multiagent Planning. In *Proceedings of the* 21^{st} *European Conference on Artificial Intelligence (ECAI)*, pages 237–242, 2014. 24, 27

M. De Ryck, M. Versteyhe, and F. Debrouwere. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, 54: 152–173, 2020. 36, 71

N. E. H. Dehimi, G. Tahar, T. Zakaria, and F. Mokhati. A novel distributed mic planning approach based on constraint satisfaction. *Multiagent and Grid Systems*, 14:243–261, 2018. 33, 35

G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemai. Decentralized motion planning and scheduling of agvs in an fms. *IEEE Transactions on Industrial Informatics*, 14(4):1744–1752, 2017. 36, 71

J. Dibangoye, A. Doniec, H. Fakham, F. Colas, and X. Guillaud. Distributed economic dispatch of embedded generation in smart grids. *Engineering Applications of Artificial Intelligence*, 44:64–78, 2015. 29

A. Dukeman and J. A. Adams. Hybrid mission planning with coalition formation. *Autonomous Agents and Multi-Agent Systems*, 31(6):1424–1466, 2017. 27

B. Dunin-Keplicz and R. Verbrugge. *Teamwork in multi-agent systems: A formal approach*, volume 21. John Wiley & Sons, 2011. 3

Elsevier. About scopus - abstract and citation database | elsevier. `https://www.elsevier.com/solutions/scopus`, 2023. Accessed: 2023-04-20. 9

T. Engesser, T. Bolander, R. Mattmüller, and B. Nebel. Cooperative epistemic multi-agent planning for implicit coordination. *Electronic Proceedings in Theoretical Computer Science*, 243, 03 2017. 29

R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3):189–208, 1971. 24

M. Fox, A. Gerevini, D. Long, and I. Serina. Plan stability: Replanning versus plan repair. In *Proceedings of the* 5^{th} *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 6, pages 212–221, 2006. 32

G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen. Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Annals of operations research*, pages 1–19, 2020. 36, 71

G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, 2021. 36

B. P. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004. 24

M. Ghallab, D. Nau, and P. Traverso. The actor's view of automated planning and acting: a position paper. *Artificial Intelligence*, 208:1–17, 2014. viii, 1, 31, 49, 50, 51

M. Ghallab, D. Nau, and P. Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. viii, xiv, 37, 45, 56

F. Gouidis, T. Patkos, G. Flouris, and D. Plexousakis. Dynamic repairing A*: a plan-repairing algorithm for dynamic domains. In *Proceedings of the* 17^{th} *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 363–370, 2018. x, 33, 35, 50, 58

K. Gue, K. Furmans, Z. Seibold, and O. Uludag. Gridstore: A puzzle-based storage system with decentralized control. *Automation Science and Engineering, IEEE Transactions on*, 11:429–438, 04 2014. 36

B. Hayes and B. Scassellati. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5469–5476, 2016. 30

W. Hellmann, D. Marino, M. Megahed, M. Suggs, J. Borowski, and A. Negahban. Human, agv or aiv? an integrated framework for material handling system selection with real-world application in an injection molding facility. *The International Journal of Advanced Manufacturing Technology*, 101, 03 2019. 71

J. Hoffmann and B. Nebel. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001. 24, 58, 78

J. Hrnčíř, M. Rovatsos, and M. Jakob. Ridesharing on timetabled transport services: A multiagent planning approach. *Journal of Intelligent Transportation Systems*, 19(1): 89–105, 2015. 27

L. Igual, S. Segu, J. Vitri, E. Puertas, P. Radeva, O. Pujol, S. Escalera, F. Dant, and L. Garrido. *Introduction to Data Science: a Python Approach to Concepts, Techniques and Applications*. Springer Publishing Company, Incorporated, 1^{st} edition, 2017. 48, 51, 58

F. Ingrand and M. Ghallab. Deliberation for autonomous robots: A survey. *Artificial Intelligence*, 247:10–44, 2017. 50

G. Jayaputera, A. Zaslavsky, and S. Loke. Enabling run-time composition and support for heterogeneous pervasive multi-agent systems. *Journal of Systems and Software*, 80 (12):2039–2062, 2007. 40, 44

A. Jonsson and M. Rovatsos. Scaling Up Multiagent Planning: A Best-Response Approach. In *Proceedings of the* 10^{th} *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 114–121. AAAI Press, 2011. 37

T. Kalmár-Nagy, G. Giardini, and B. D. Bak. The multiagent planning problem. *Complexity*, 2017:3813912, Feb 2017. 1

B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004. 8

B. A. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University Joint Report, 2007. URL `https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf`. 8

L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2006. 24

A. Komenda, P. Novák, and M. Pěchouček. Decentralized multi-agent plan repair in dynamic environments. In *Proceedings of the* 11^{st} *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 1239–1240. IFAAMAS, 2012. x, 50

A. Komenda, P. Novák, and M. Pchouček. Domain-Independent Multi-Agent Plan Repair. *Journal of Network and Computer Applications*, 37:76–88, 2014. x, 25, 35, 42, 45, 50, 57, 58, 82

A. Komenda, M. Štolba, and D. L. Kovacs. The international competition of distributed and multiagent planners (codmap). *AI Magazine*, 37(3):109–115, 2016. 24, 27

N. Kousi, S. Koukas, G. Michalos, and S. Makris. Scheduling of smart intra – factory material supply operations using mobile robots. *International Journal of Production Research*, 57(3):801–814, 2019. 36, 71

D. L. Kovács. A multi-agent extension of pddl3.1. *Proceedings of International Conference on Automated Planning and Scheduling*, 2012a. 24

D. L. Kovács. A multi-agent extension of PDDL3.1. In *Proceedings of the* 3^{rd} *Workshop on the International Planning Competition (IPC)*, pages 19–27. AAAI Press, 2012b. 56, 73

L. Kraemer and B. Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016. 17, 28

A. Kumar, S. Zilberstein, and M. Toussaint. Probabilistic inference techniques for scalable multiagent decision making. *Journal of Artificial Intelligence Research*, 53:223–270, 2015. 28

H.-Y. Lee and C. C. Murray. Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research*, 57(18):5821–5841, 2019. 83

S. Maliah, G. Shani, and R. Stern. Privacy preserving landmark detection. In *Proceedings of the* 21^{st} *European Conference on Artificial Intelligence (ECAI)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 597–602. IOS Press, 2014. 23

S. Maliah, G. Shani, and R. Stern. Stronger privacy preserving projections for multi-agent planning. In *Proceedings of the* 26^{th} *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 221–229. AAAI Press, 2016. 27

S. Maliah, G. Shani, and R. Stern. Collaborative privacy preserving multi-agent planning. *Autonomous Agents and Multi-Agent Systems*, 31(3):493–530, 2017. 27

K. D. Maniya and M. G. Bhatt. A multi-attribute selection of automated guided vehicle using the ahp/m-gra technique. *International Journal of Production Research*, 49(20): 6107–6124, 2011. 36

A. M. Mariano and M. S. Rocha. Revisão da literatura: apresentação de uma abordagem integradora. In *Proceedings of the* 26^{th} *European Academy of Management and Business Economics (AEDEM)*, volume 18, pages 427–442, 2017. 8

A. M. Mariano, A. C. B. Reis, L. d. S. Althoff, and L. B. Barros. A bibliographic review of software metrics: applying the consolidated meta-analytic approach. *Industrial Engineering and Operations Management I: XXIV IJCIEOM, Lisbon, Portugal, July 18–20 24*, pages 243–256, 2019. 9

S. Matsuoka and T. Sawaragi. Recovery planning of industrial robots based on semantic information of failures and time-dependent utility. *Advanced Engineering Informatics*, 51:101507, 2022. 32, 35

R. Micalizio and P. Torasso. On-line monitoring of plan execution: a distributed approach. *Knowledge-Based Systems*, 20(2):134–142, 2007. 51

S. K. Mohalik, M. B. Jayaraman, R. Badrinath, and A. V. Feljan. HIPR: an architecture for iterative plan repair in hierarchical multi-agent systems. *Journal of Computers*, 13 (3):351–359, 2018. x, 34, 35, 45, 50

L. H. Moreira and C. G. Ralha. Evaluation of decision-making strategies for robots in intralogistics problems using multi-agent planning. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1272–1279. IEEE, 2021a. 49, 69, 71, 77

L. H. Moreira and C. G. Ralha. Plan recovery process in multi-agent dynamic environments. In *Proceedings of the* 18^{th} *International Conference on Informatics in Control, Automation and Robotics(ICINCO)*, pages 187–194. SCITEPRESS, 2021b. 69, 71, 77

L. H. Moreira and C. G. Ralha. An efficient lightweight coordination model to multi-agent planning. *Knowledge and Information Systems.*, 64(2):415–439, 2022. 32, 35

R. Nissim and R. Brafman. Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research*, 51:293–332, 2014. 22, 23, 29

R. Nissim, R. I. Brafman, and C. Domshlak. A general, fully distributed multi-agent planning algorithm. In *Proceedings of the* 9^{th} *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1323–1330. IFAAMAS, 2010. 24

M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 2013. 54, 78

Pandas development team. Python data analysis library. `https://pandas.pydata.org/`, 2021. Accessed on: 2021-03-22. 58

D. Pellier and H. Fiorino. PDDL4J: a planning domain description library for java. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(1):143–176, 2018. 53, 78

A. Queffelec, O. Sankur, and F. Schwarzentruber. Complexity of planning for connected agents in a partially known environment. *Theoretical Computer Science*, 941:202–220, 2023. 31, 35

S. Richter and M. Westphal. The lama planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010. 24

Y. Rizk, M. Awad, and E. W. Tunstel. Cooperative heterogeneous multi-robot systems. *ACM Computing Surveys (CSUR)*, 52:1 – 31, 2019. 15, 16, 19, 28, 30

A. Saetti and E. Scala. Optimising the stability in plan repair via compilation. In *Proceedings of the* 32^{nd} *International Conference on Automated Planning and Scheduling (ICAPS)*, volume 32, pages 316–320, 2022. 32, 35

Y. E. Sahin, P. Nilsson, and N. Ozay. Provably-correct coordination of large collections of agents with counting temporal logic constraints. In *Proceedings of the* 8^{th} *International Conference on Cyber-Physical Systems (ICCPS)*, pages 249–258, 2017. 28

P. Schillinger, M. Bürger, and D. V. Dimarogonas. *Decomposition of Finite LTL Specifications for Efficient Multi-agent Planning*, pages 253–267. Springer International Publishing, 2018a. 28

P. Schillinger, M. Bürger, and D. V. Dimarogonas. Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. *The International Journal of Robotics Research*, 37(7):818–838, 2018b. 28

P. Skobelev, E. Simonova, A. Zhilyaev, and V. Travin. Application of multi-agent technology in the scheduling system of swarm of earth remote sensing satellites. *Procedia Computer Science*, 103:396–402, 2017. 29

M. Štolba and A. Komenda. MADLA: planning with distributed and local search. *Competition of Distributed and Multi-Agent Planners (CoDMAP 2015)*, page 21, 2015. ix, 45, 50

A. Torreño, E. Onaindia, and Ó. Sapena. An approach to multi-agent planning with incomplete information. In *Proceedings of the* 20^{th} *European Conference on Artificial Intelligence (ECAI)*, pages 762–767. IOS Press, 2012a. 45

A. Torreño, E. Onaindia, and O. Sapena. An approach to multi-agent planning with incomplete information. In *Proceedings of the* 20^{th} *European Conference on Artificial Intelligence (ECAI)*, volume 242, 08 2012b. 24

A. Torreño, E. Onaindia, and Ó. Sapena. FMAP: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606–626, 2014. ix, 25, 27, 29, 45, 50

A. Torreño, E. Onaindia, A. Komenda, and M. Štolba. Cooperative Multi-Agent Planning: A Survey. *ACM Computing Surveys*, pages 1–32, 2017. xvi, 15, 23, 27, 29, 31, 37, 45, 51

J. ToźIăźKa, J. JakubźV, and A. Komenda. Generating multi-agent plans by distributed intersection of finite state machines. In *Proceedings of the* 21^{st} *European Conference on Artificial Intelligence (ECAI)*, 2014. 26

J. ToźIăźKa, J. JakubźV, A. Komenda, and M. PundefinedchouăźEk. Privacy-concerned multiagent planning. *Knowledge and Information Systems*, 48(3):581–618, 2016. 26

P. Traverso, M. Ghallab, and D. Nau. *Automated Planning: Theory and Practice*. M. Kauffman, 2004. 24

J. Tumova and D. V. Dimarogonas. A receding horizon approach to multi-agent planning from local LTL specifications. *American Control Conference*, abs/1403.4174, 2014. 27

J. Tumova and D. V. Dimarogonas. Decomposition of multi-agent planning under distributed motion and task LTL specifications. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7448–7453. IEEE, 2015. 27

J. Tumova and D. V. Dimarogonas. Multi-agent planning under local ltl specifications and event-based synchronization. *Automatica*, 70:239–248, 2016. 27

R. Vogel and W. H. Güttel. The dynamic capability view in strategic management: A bibliometric review. *International Journal of Management Reviews*, 15(4):426–446, 2013. 12

J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, and J. Lloret. Context-aware cloud robotics for material handling in cognitive industrial internet of things. *IEEE Internet of Things Journal*, 5(4):2272–2281, 2017. 71

M. Weerdt and B. Clement. Introduction to planning in multiagent systems. *Multiagent and Grid Systems*, 5:345–355, 12 2009. 24

G. Weiss. *Multiagent Systems*. The MIT Press, 2^{nd} edition, 2013. 37

M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd, 2^{nd} edition, 2009. 3, 37

Y. Zhang, Z. Zhu, and J. Lv. Cps-based smart control model for shopfloor material handling. *IEEE Transactions on Industrial Informatics*, 14(4):1764–1775, 2018. 71

M. Štolba and A. Komenda. The madla planner: Multi-agent planning by combination of distributed and local heuristic search. *Artificial Intelligence*, 252:175–210, 2017. 26

M. Štolba, D. Fišer, and A. Komenda. Admissible landmark heuristic for multi-agent planning. *Proceedings of International Conference on Automated Planning and Scheduling*, 25(1):211–219, 2015. 25

Appendix A

Domain Categorization

We based our case studies on the CoDMAP domains on choosing appropriate experiments for evaluation. It is crucial to ensure that our selection covers all relevant issues, including the level of coupling displayed by each domain.

Regarding the most used case studies described in the experiments of Chapters 4 and 5 (Sections 4.3 and 5.3), the domains chosen from CoDMAP were:

- Satellite - each agent represents a satellite characterized by its specific attributes, such as position, direction, and available instruments. Despite having distinct qualities, this characteristic makes it easier to solve the original problem as the agents can work independently without the need for collaboration. The main challenge is to collect and store data using various instruments to observe multiple targets while scaling satellite observations;
- Zeno-travel - agents deal with transportation, where individuals board planes, travel between locations, and exit. The fuel consumption of airplanes varies based on their speed;
- Rovers - this domain focuses on the exploration of Mars. It involves using a group of robots to collect samples and transmit data back to the base by traveling to different points on the planet. The problem considers the limitations of the base's visibility from different positions and each robot's ability to travel between points. Each robot has unique equipment to perform tasks, and they only need to cooperate during the mission when one collects a sample, making it unavailable to the others;
- Logistics - airplanes and trucks transport items. Sometimes, both vehicles are needed to deliver a package - airplanes handle trips between cities, while trucks are responsible for transportation within the same city;

- Driverlog - similar to logistics but involves drivers and trucks moving between locations. The drivers walk on paths different from those used by the trucks for driving. Additionally, there is always an intermediate location on a footpath between two road junctions. The trucks can be loaded or unloaded with packages, and the objective is to transport these packages;
- Taxi - this domain deals with transportation in a city involving passengers and taxis. Each taxi can transport only one passenger at a time from its current location to an available drop-off point, and taxis can travel only between connected locations.

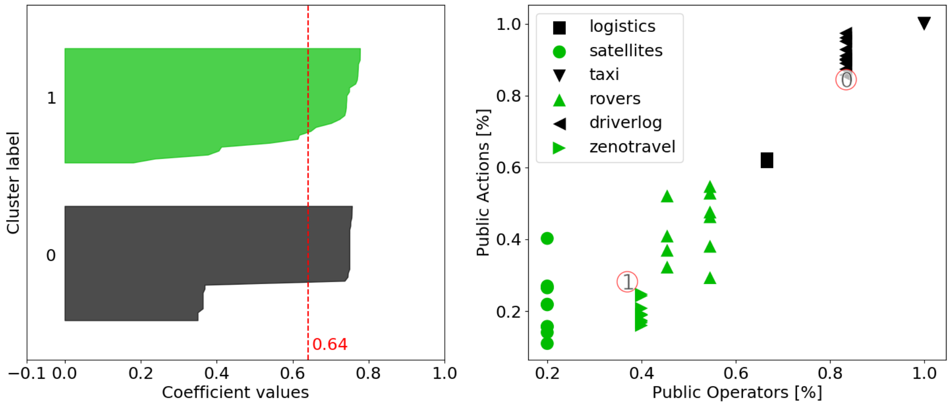
We sorted various domains into types based on the level of public actions. We aimed to gather domains with similar characteristics and believed each group could represent the other domains. We decided to cluster the domains and categorize the experiments to accomplish this.

We utilized silhouette[1] analysis to group domains into clusters, which measures the distance between clusters. The silhouette plot shows the proximity of each point in a cluster to those in adjacent clusters. This analysis allows a visual assessment of parameters like the number of clusters, with a range of -1 to 1. If a coefficient is near +1, the sample is far from neighboring clusters. A value of 0 suggests that the sample is on or very close to the decision boundary between two neighboring clusters. In contrast, negative values indicate that the samples may have been assigned to the wrong cluster.

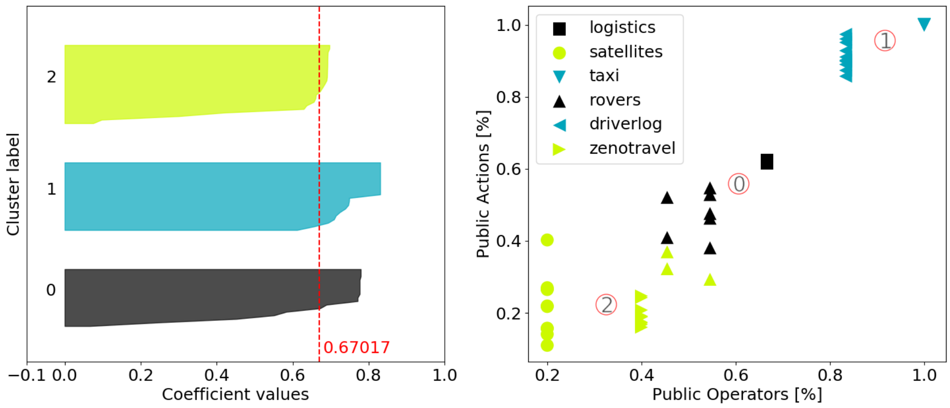
To determine the ideal number of clusters, each should have a silhouette value more significant than the average. We conducted a Silhouette analysis on experiments grouped from two to six (number of domains), as shown in Figures A.1 and A.2. Results indicated that the configuration of two and three clusters (left column of Figure A.1(a and b)) surpassed the average value (red text at the bottom of each chart). However, the best number of clusters is three as they have a more uniform shape than the first (left graphs in Figure A.1). The configurations of four, five, and six clusters presented in Figure A.1(c) and Figure A.2 were disqualified because some clusters did not reach the average value.

We evaluated each domain in ten different configurations to check the ideal number of clusters. In Figure A.1(b - right column), 60 experiments were divided into three clusters with red circles indicating the centers of each cluster. Experiments in the same cluster share a common color. This categorization not only defines the coordination complexity (Definition 15) of the experiment but also the level of coupling (Definition 13) between agents. The cluster centers (circles) form a crescent line representing the complexities of the problems. Table A.1 groups the case studies according to the clusters used throughout the evaluations.

[1] `http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html`



(a) Two clusters.

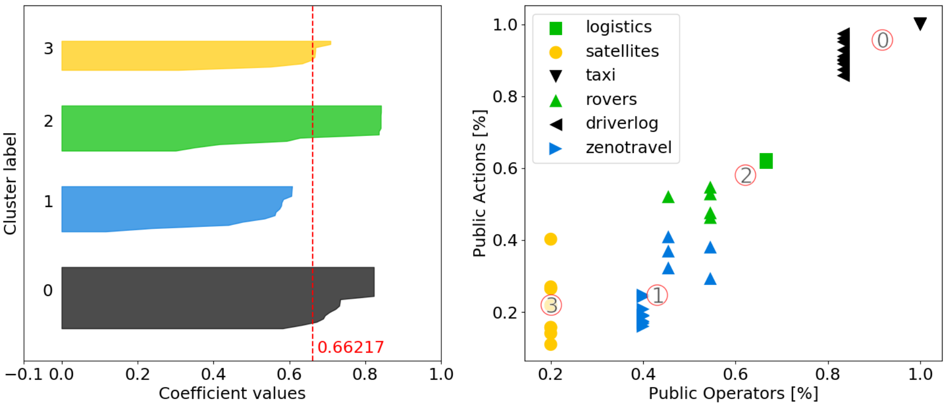


(b) Three clusters.

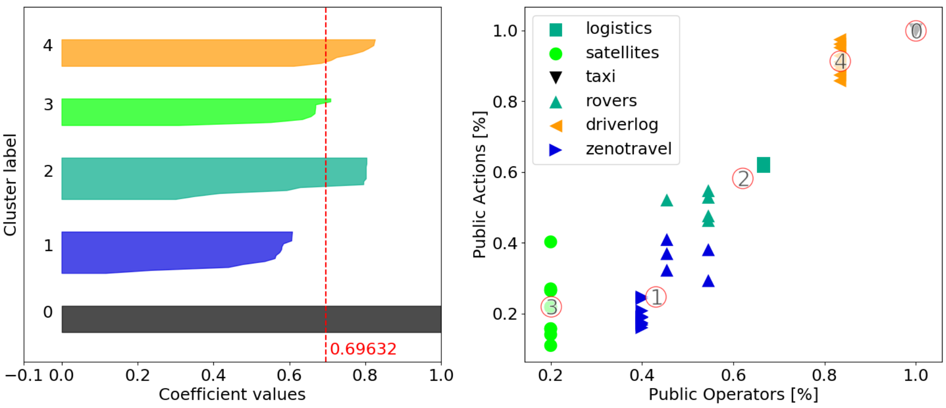
Figure (A.1) Cluster evaluation.

Table (A.1) Clusters of experiments.

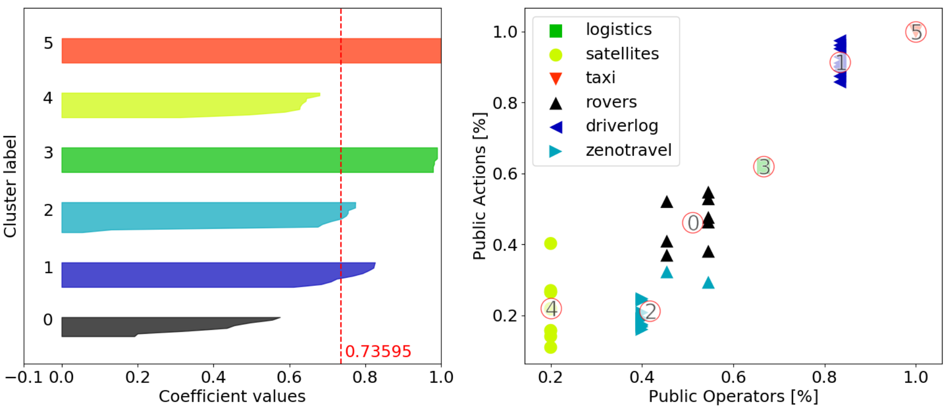
Cluster	Domains	Level of coupling
Cluster 1	Satellite and Zeno-travel	Loose
Cluster 2	Rovers and Logistics	Intermediate
Cluster 3	Driverlog and Taxi	Tight



(a) Four clusters.



(b) Five clusters.



(c) Six clusters.

Figure (A.2) Cluster evaluation.