electronics

Article

A Multi-Agent System for Parking Allocation: An Approach to Allocate Parking Spaces

Gabriel Icarte-Ahumada [1,*], Zhangyuan He [2,*], Victor Godoy [1], Francisco García [1] and Mauricio Oyarzún [1]

[1] Facultad de Ingeniería y Arquitectura, Universidad Arturo Prat, Iquique 1110939, Chile; francgar@unap.cl (F.G.); vngodoyr@unap.cl (V.G.); moyarzunsil@unap.cl (M.O.)
[2] School of Urban Planning and Design, Shenzhen University, Shenzhen 518060, China
* Correspondence: gicarte@unap.cl (G.I.-A.); he.zhangyuan@pku.edu.cn (Z.H.)

Abstract: Traffic congestion and the search for parking spaces have become significant challenges in modern urban development. These issues lead to increased fuel consumption and contribute to stressful lifestyles. To address this, efficient parking allocation has emerged as a challenge in urban environments, requiring innovative solutions to optimize the use of available parking spaces. Various approaches have been employed to tackle this problem, including mathematical programming and heuristic procedures, which typically follow a centralized model. Another promising solution involves the use of multi-agent systems (MASs), which adopt a decentralized approach. However, there is limited research on the application of MASs in parking allocation. This paper presents a multi-agent system where intelligent agents represent vehicles and parking spaces to efficiently allocate parking spaces to cars. The paper evaluates four coordination mechanisms based on the Contract Net Protocol for parking allocation. The results show that the concurrent use of the Contract Net Protocol with decommitment capacities is the most effective approach, as it reduces the time required for parking space allocation while optimizing the utilization of available parking spaces.

Keywords: multi-agent systems; smart cities; contract net protocol; intelligent parking systems

check for updates

Academic Editor: Fernando De la Prieta Pintado

Received: 6 February 2025
Revised: 12 February 2025
Accepted: 17 February 2025
Published: 21 February 2025

Citation: Icarte-Ahumada, G.; He, Z.; Godoy, V.; García, F.; Oyarzún, M. A Multi-Agent System for Parking Allocation: An Approach to Allocate Parking Spaces. *Electronics* **2025**, *14*, 840. https://doi.org/10.3390/electronics14050840

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Traffic congestion and the search for parking have become serious problems in current urban development. Studies reveal that drivers spend an average of more than eight minutes daily searching for parking, representing 30% of traffic in large cities [1]. Moreover, this issue is made worse by additional factors, such as increased fuel consumption and stressful lifestyles [2,3]. The continuous growth of population and vehicle mobility contributes to congestion, affecting the quality of life for residents and having a negative impact on the environment [4]. Hereafter, we refer to this as the parking allocation problem.

The solution to this problem requires an integrated approach, including a smart and efficient parking management system. This system should provide real-time information on available parking, offering the best option based on the driver's location and needs. It should consider various criteria, such as proximity to the destination, cost, safety, and traffic flow, and be able to handle multiple allocation requests simultaneously to avoid congestion and optimize the user experience [4].

To address the time spent searching for parking, researchers have explored various parking demand management strategies, such as dynamic parking pricing, negotiable parking permits, and parking time limits [5]. The study by [6] offers an extensive review of

modeling, solutions, and technologies related to smart parking, also identifying challenges and pending issues in this field. In addition, the rise of Artificial Intelligence applications in the automotive industry has driven an increasing demand for smart parking solutions.

One approach used to solve the parking allocation problem is multi-agent systems. Multi-agent systems are structures composed of multiple autonomous agents interacting with each other to solve complex problems that are difficult for a single agent to tackle. The benefits of MASs include the ability to manage decentralized control, flexibility to adapt to changes in the environment, and robustness against failures. These and other benefits make MASs a good alternative for applications in areas such as logistics, traffic management, electrical vehicles, and robotics, where collaboration and distributed decision-making are essential [7–10].

In this paper, we propose a MAS where intelligent agents represent vehicles and parking spaces from the real world. These agents interact with each other through a negotiation protocol, which allocates parking spaces to vehicles considering various criteria, such as passengers' final destinations, vehicle characteristics, and parking spaces features. We compare four different ways of applying the negotiation protocol: serial, serial with decommitment, concurrent, and concurrent with decommitment. The comparison aims to determine which of these protocols provides the fastest response time and the highest-quality allocations. A key innovation of this work is the in-depth analysis of communication mechanisms within the MASs, evaluating their impact on efficiency and allocation quality, which has been unexplored in previous studies.

The remainder of the research is structured as follows: Section 2 presents related works focusing on those using multi-agent systems. Section 3 shows the methodology applied. Section 4 provides a discussion. Finally, Section 5 concludes the paper.

2. Related Work

Smart parking solutions literature is very wide and diversified. In [6] the authors provide a large survey on smart parking modeling, solutions, and technologies as well as identify challenges and open issues. The article is a good starting point to understand the problem, the different solution strategies, and learn about some real applications. Next, and because our work is related to multi-agent systems, we present some work where this technology has been used to solve the parking allocation problem.

Murano et al. [3] examined the parking problem as a competitive multiplayer game in which each car acts as an interacting agent aiming to find an available parking that meets its specific constraints. The objective is to park as many cars as possible while meeting their requirements, such as parking within a certain timeframe and adhering to resilience limits, ultimately achieving a Nash Equilibrium for the game. However, this study only considers a scenario with a single gate and proposes a solution for this specific case, without considering that many parking areas (like those for hospitals, offices, and malls) have multiple gates.

Hassine et al. [2] proposed a dynamic parking charging system designed to optimize parking management demands. Their smart dynamic parking pricing system uses a multi-agent framework to adjust parking prices in real-time according to current supply and demand. This approach aims to reduce traffic congestion and ensure optimal utilization of parking, especially during peak hours. Drivers make reservations by receiving the current parking prices, but a notable limitation is the system's reliance on a central agent to communicate with parking agents. If this central agent fails, it risks the entire system's functionality. The negotiation between agents is managed through a Contract Net Protocol.

Khalidi et al. [4] propose a conceptual multi-agent system designed to handle all key functions within a smart parking system. Autonomous agents manage various services,

such as reservations, parking, and security, ensuring efficient communication and collaboration to provide optimized parking allocations to customers. Central agents are responsible for specific tasks; for instance, the reservation agent handles customer interactions to secure parking in designated areas. However, the system's reliability could be compromised if any of these central agents were to fail.

Zhang et al. [5] propose a multi-agent reinforcement learning (MARL) framework to address the optimal parking assignment (OPA) problem by reducing travel time for both connected vehicles (CVs) and non-connected vehicles (NCVs). The framework utilizes a modified exploration strategy to observe and dispatch CVs to suitable parking, ensuring efficient parking allocation at varying CV penetration rates. The model was tested through simulations based on real scenarios from Tongji University's campus, showing promising results. Additionally, the framework includes two agents: one dedicated to assessing the impact of NCVs, while the other focuses on exploring CV parking dynamics. Experimental results show that this approach improves parking resource allocation by up to 15% compared to existing benchmarks, demonstrating its potential for enhancing OPA strategies in urban environments.

Rizvi et al. [11] propose a Multi-Agent Auction-based Parking (MAPark) system for smart cities, utilizing IoT infrastructure. It introduces a unified reservation and demand-based pricing strategy incorporating practical parking preferences from both drivers and parking facility providers (PFPs). These preferences include parking type, maximum price, and maximum walking time for drivers, and passenger throughput, driver punctuality record, and monetary value for PFPs. The auction-based parking system positively impacts drivers' responses to pricing policies, reducing urban traffic congestion and maximizing parking revenue and passenger throughput. The proposed solution uses a request center to manage all the received requests from users. Nevertheless, although the authors ensure that individual rationality and budget balance are maintained, they do not elaborate on how their auction mechanism supports allocative efficiency.

Jioudi et al. [12] provide an examination of the evolution of the smart parking concept and conducts statistical analysis on published patent applications. Subsequently, a novel multi-agent architecture is introduced to oversee smart parking management. The proposed solution employs distinct agents for various tasks. For example, one agent oversees parking allocation, other handles bookings, a separate agent facilitates communication between agents, and a pricing agent regulates parking occupancy and updates parking prices at predetermined intervals.

Boudali and Ben [13] present a parking system that employs a multi-agent approach to offer real-time decision support for drivers based on their preferences and operational constraints. This is formulated as a multicriteria decision-making problem known as the multicriteria parking reservation problem. Parking allocation is updated to prevent reservation conflicts and ensure constraint satisfaction. Following reservation, a guidance module assists drivers in reaching their reserved parking via the shortest path. Simulation results demonstrate the system's effectiveness in various real-world scenarios.

In summary, the parking allocation problem has been addressed using multi-agent systems in various ways. However, many of these studies suffer from various shortcomings when applied to find appropriate solutions to this problem. The following research gaps are concluded based on the previous works that were analyzed in the literature:

1. The use of centralized agents to handle car requests or critical system tasks poses a vulnerability. If these agents fail, the systems could become unusable. Also, this centralized use of agents underutilized the parallel processing capacities of MASs.
2. Few studies consider open (ungated) and free parking. Most focus on closed parking lots and those with fees, some of which use entry gates.

3. There is a lack of in-depth studies on the interaction mechanisms used in the proposed MASs.
4. Most of the articles use agents in a functional way, meaning they perform specific tasks.

Our work covers the four points above but focuses on the last two gaps: studying and evaluating different ways to apply protocols to negotiate with each other on behalf of their real-world representation. In this way, we leverage some of the main strengths of MASs: parallel processing capabilities and high fault tolerance by not having agents that centralize certain tasks.

3. Methodology

The methodology employed in this study was structured around three key stages: defining the problem, designing and implementing a multi-agent system, and evaluating its performance. The following subsections provide a detailed description of this methodology.

3.1. Problem Statement

The parking allocation problem that we are facing involves allocating n cars to n parking spaces, where each car has a designated destination, a specific type (e.g., ambulance, disabled, police, etc.), and a certain size. Each parking space is restricted not only by the types of cars allowed but also by the size of the parking space, which must be appropriate for the car. There are two objectives to achieve this. The first one is to minimize the total distance traveled, considering a weighted distance: 0.4 for the distance from the car to the parking space, and 0.6 for the distance from the parking space to the car's destination. These values represent weighting coefficients. Their selection was arbitrary and reflects a slight preference for minimizing the walking distance for passengers over reducing the distance the car must travel to reach the parking lot.

The second objective is to minimize the distance traveled from the current position of a car to a parking space. Additionally, a car can only be allocated to a parking space if both the parking space type matches the car's type and the car physically fits in the parking space, adding a size compatibility constraint to the allocation process. The problem is formulated as multi-objective linear optimization problem, requiring optimization techniques to find the optimal allocation under these constraints. Table 1 shows the terms used in the formulation.

Table 1. Terms used in the formulation.

Term	Description
$A = \{a_1, a_2, \ldots, a_n\}$	Set of n cars.
$P = \{p_1, p_2, \ldots, p_n\}$	Set of n parking spaces.
$D = \{d_1, d_2, \ldots, d_n\}$	Set of n destinations, where each car a_i has a destination d_i.
$d_{ai,pj}$	Distance from car a_i to parking space p_j.
$d_{pj,ai}$	Distance from parking space p_j to the destination d_i of car a_i.
x_{ij}	Binary decision variable: $x_{ij} = 1$ if car a_i is allocated to parking space p_j, otherwise $x_{ij} = 0$.
$\tau(a_i)$	Type of car a_i (e.g., ambulance, police, normal).
$\tau(p_j)$	Type of parking space p_j allowed to accommodate certain types of cars.
$s(a_i)$	Size of car a_i.
$s(p_j)$	Size of parking space p_j, representing the maximum size of a car that can be parked there.

Table 1. *Cont.*

Term	Description
m_{ij}	Binary compatibility matrix, where $m_{ij} = 1$ if car a_i is allowed to park in space p_j based on type compatibility, otherwise $m_{ij} = 0$.
y_{ij}	Binary size compatibility matrix, where $y_{ij} = 1$ if car a_i fits in space p_j, otherwise $y_{ij} = 0$.
$w_1 = 0.4$	Weight for the distance from the car to the parking space.
$w_2 = 0.6$	Weight for the distance from the parking space to the car's destination.

Objective Functions

$$Min \sum_{i=1}^{n} \sum_{j=1}^{n} (w_1 \cdot d_{ai,pj} + w_2 \cdot d_{pj,\,ai}) \cdot x_{ij} \quad (1)$$

$$Min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{pj,ai} \cdot x_{ij} \quad (2)$$

Constraints

$$x_{ij} \leq m_{ij} \forall \mathrm{i,j} \quad (3)$$

$$x_{ij} \leq y_{ij} \forall \mathrm{i,j} \quad (4)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \forall \mathrm{i} = 1, 2, \ldots \mathrm{n} \quad (5)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \forall \mathrm{j} = 1, 2, \ldots \mathrm{n} \quad (6)$$

$$x_{ij} \in \{0, 1\} \ \forall \mathrm{i,j} \quad (7)$$

The first objective function (1) minimizes the total distance traveled for allocated cars to parking spaces, considering both the weighted distance from the car to the parking space (with weight $w_1 = 0.4$) and the distance from the parking space to the car's destination (with weight $w_2 = 0.6$). The second objective function (2) minimizes the distance traveled from the current position of a car to a parking space. The allocation is subject to several constraints. First, the type of compatibility constraint (3) ensures that a car can only be allocated to a parking space if the parking space allows that type of car. Second, the size compatibility constraint (4) ensures that a car can only be allocated to a parking space if the parking space is large enough to fit the car. Additionally, each car must be allocated to exactly one parking space (5), and each parking space can only accommodate one car (6). Finally, the decision variables x_{ij} are binary, indicating whether a car is allocated to a specific parking space (7). This formulation leads to a multi-objective linear optimization problem that seeks an optimal solution under these constraints.

3.2. The Multi-Agent System for Parking Allocation

The objective of the proposed multi-agent system for addressing the parking allocation problem is to provide a parking allocation considering drivers' preferences and car and parking features. Figure 1 shows an overall logical framework.

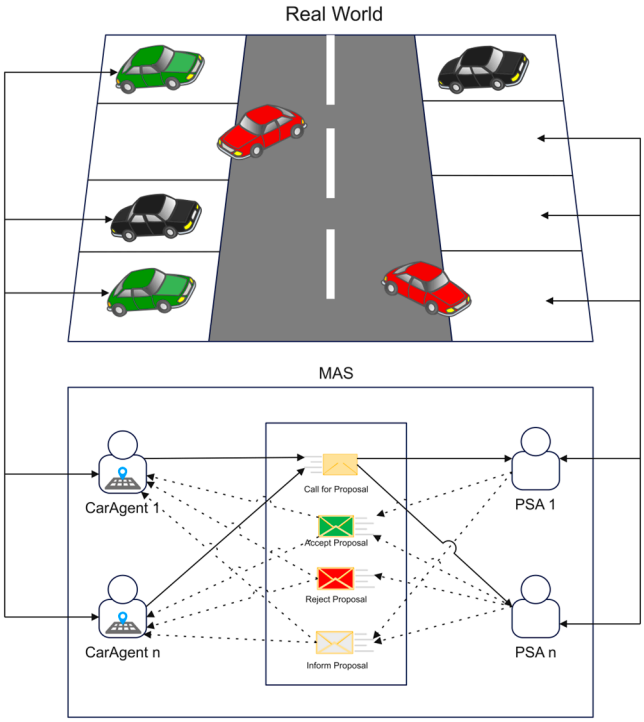


Figure 1. Visual representation of a real-world parking problem and the multi-agent system that represents it. The upper section illustrates the parking search by vehicles, while the lower section shows agents that represent entities from the real world interacting with each other for parking allocation. PSA means ParkingSpaceAgent.

The proposed MAS involves agents that represent real-world entities. The following subsections describe the agents, the coordination mechanism, and the agent decision making.

3.2.1. Agents

The agents implemented in the multi-agent system are the carAgents and the parkingAgents. A carAgent represents a car from the real world in search of a parking space. On the other hand, a parkingAgent represents a parking space from the real world. Table 2 provides a description of the agents and their roles.

Table 2. Agent description.

Agent	Real-World Representation	Objective	Properties
carAgent	Car	Finding a parking space as close as possible to its destination and as close as possible to its current position	- current position - destination - size - type (e.g., normal, ambulance, fire truck)
parkingAgent	Parking	Being used by the nearest vehicle that requires it	- position - size - type (e.g., normal, for disabled, pregnancy)

3.2.2. Coordination Mechanisms in the Multi-Agent System

The agents interact with each other by applying the well-known Contract Net Protocol (CNP) [14]. In the context of the proposed MAS for the parking allocation problem, the CNP works as follows: a carAgent initiates a negotiation with parkingAgents sending them a call-for-proposals message (CFP). The CFP points out the characteristics of the car and the destination of the passengers. When a parkingAgent receives a CFP, it evaluates and decide to send a proposal message. The proposal points out the distance from the parking space to the destination of the passengers and the location of the parking space. When the carAgent receives all the proposal messages from the parkingAgents, or when the deadline is expired, it looks for the best proposal. The best proposal is the one that proposes the shortest distance to the destination of the passengers and the nearest parking space to its current position. Then, the carAgent sends an accept message to the parkingAgent that proposed the best proposal and sends a rejection message to the other parkingAgents that sent a proposal. After this, the car that is represented by the carAgent that initiated the negotiation starts the trip to the parking space and the negotiation is then finished. Figure 2 depicts the interactions among the agents applying the Contract Net Protocol.

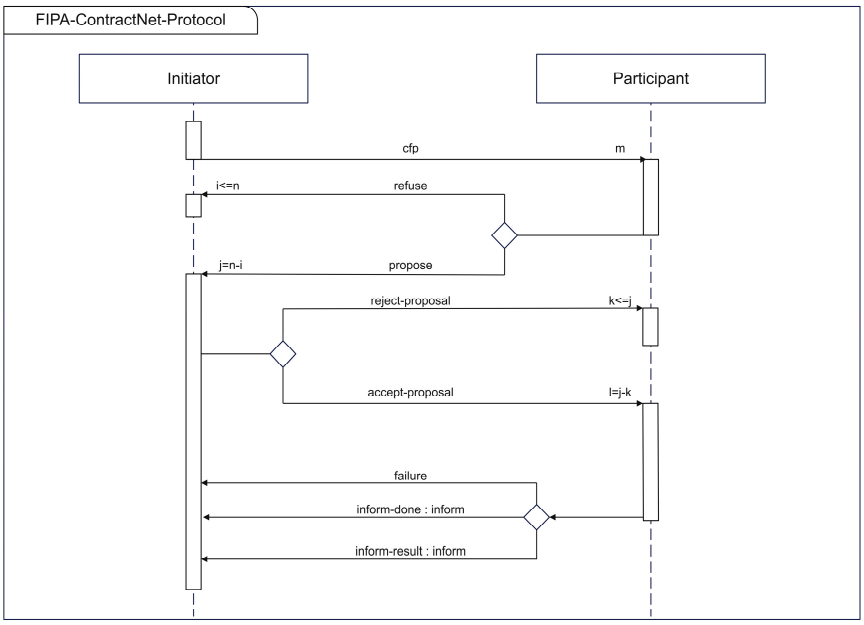


Figure 2. The Contract Net Protocol [14]. The diagram depicts the interaction between an initiator and a participant, detailing the message exchange process, including call-for-proposals (CFP), proposals, acceptance, rejection, and completion notifications.

The CNP can be applied in different ways. For instance, one way is a serial way. Another way is a concurrent application, this means that several CNPs work in parallel. To know which application of the CNP is more appropriate, we implemented four different variations of the application of the CNP: serial, serial with decommitment, concurrent, and concurrent with decommitment.

Serial: This is simplest way. One carAgent starts a negotiation and the other carAgents wait until the negotiation is finished. After the negotiation is finished, another carAgent starts a negotiation and so on. In this case, if a parkingAgent receives an accept-message (it means that the parking space will be reserved for a car), this parkingAgent is not able to take part in another negotiation.

Serial with decommitment: This case is a variation of the previous one. Unlike before, if a parking space has been reserved for a car, the parkingAgent it represents can still engage in other negotiations. If another negotiation is more favorable, it withdraws from

the previous negotiation and accepts the new one. This means that the carAgent affected by the withdrawal must initiate a new negotiation to find a new parking space.

Concurrent: In this case, carAgents can request parking spaces in parallel, resulting in parkingAgents receiving multiple CFPs. The parkingAgents send their proposals, and upon receiving an acceptance, they reject all other acceptances and do not engage in further negotiations. In this context, it is possible for parkingAgents to reject a CFP that is a better option than a previously answered CFP. This issue is also known as "the eager bidder problem" [15].

Concurrent with decommitment: This case is a variation of the previous one. This variation faces with "the eager bidder problem" adding a decommitment stage. Unlike before, if a parking space has been allocated to a car, the parkingAgent it represents can still engage in other concurrent negotiations. If another negotiation is more favorable, it withdraws from the previous negotiation and accepts the new one. This means that the carAgent affected by the withdrawal must initiate a new negotiation to find a new parking space.

3.2.3. Agent Decision-Making

A carAgent is the agent that initiates a negotiation process using the CNP. To decide what is the best proposal received from the parkingAgents, the carAgents evaluate all the proposals received using a utility function. This function promotes those proposals that minimize the distance from the parking space to the passengers' destination and are closer to the cars. Then, the function to select the best proposal is:

$$p_{best} = \min\left(0.4 \cdot d_{ai,pj} + 0.6 \cdot d_{pj,ai}\right) , \forall\, i \in \{1, 2, \ldots, n\} \tag{8}$$

In the case of a parkingAgent must decide whether to send a proposal. If the parking space is already allocated, or its type does not match the car's type, or if the parking space is smaller than the car, it will not send a proposal. If it decides to send a proposal, it will indicate its position and the distance to the passengers' destination. Additionally, in the case of protocols with decommitment, the parkingAgent may decide to decommit and to take part in another negotiation even if the parking space is allocated to another car. This will happen if it receives a CFP message that improves its chances of fulfilling its objective. This could occur if a car is closer to the parking space.

3.2.4. MAS Implementation

The MAS was developed using the Java Agent DEvelopment framework (JADE 4.6.0) platform [16]. JADE is an appropriate toolkit for implementing multi-agent systems as it offers various interaction protocols, behaviors, and graphical tools for analyzing agent behavior and system performance. Since JADE is built in JAVA 1.8, it is compatible with all operating systems. All experiments were conducted on a MacBook Pro equipped with an Apple M2 Pro chip, with 16 GB of RAM.

3.3. Evaluation

To evaluate the different applications of the CNP, we simulated a hypothetical scenario. The aim of the simulation was to determine the required time by the MAS to allocate parking spaces to cars and the quality of the allocation. The proposed MAS and the simulation were run on the JADE platform. The following subsections describe the simulation setup and the results.

3.3.1. Simulation Setup

We evaluated the performance of the MAS in a scenario with 1000 cars and 1000 parking spaces, representing small city centers without gates or restricted entries. The distances between cars and parking spaces range from 1 to 1000 m, with 500 distinct destinations. Cars are randomly distributed across various locations and each car is assigned a specific destination. All 1000 cars simultaneously request parking spaces in a simulation lasting 5 min.

An important factor in the simulation is the protocol deadline, which impacts negotiation delays. A longer deadline allows messages to be received and processed more thoroughly but increases negotiation time. Conversely, a shorter deadline speeds up negotiations but risks lower allocation quality, as late-arriving messages may be discarded. For this experiment, the protocol deadline was set to 4000 ms to balance negotiation speed and allocation efficiency.

3.3.2. Results

The aim of this study was to assess the effectiveness of a multi-agent system for optimizing parking allocation in smart cities, with a particular focus on negotiation protocols. The analysis of the results is presented from two perspectives: negotiation protocol efficiency and the quality of parking allocations. Table 3 provides the results.

Table 3. Results for serial, concurrent, serial decommit, and concurrent decommit.

Criteria	Serial	Concurrent	Serial Decommit	Concurrent Decommit
Number of negotiations	493	1020	324	1919
Successful negotiations	493	1000	301	1060
Failed negotiations	0	20	23	859
Number of decommits	0	0	21	271
Number of allocated cars	493	1000	280	789
Max duration (ms)	1638	4057	4360	7126
Min duration (ms)	87	39	206	396
Total distance to parking (km)	15,064	45,340	6944	17,802
Quality of allocations *	30.55	45.34	24.8	22.57

* Total distance to parking/number of cars allocated.

Negotiation Results

Successful negotiations are those that end with a parkingAgent that confirm the accept message received from a carAgent, while failed negotiations are those when the carAgent did not receive proposals, or because a possible parkingAgent did not confirm the accept message received from a carAgent. The latter happens when the parkingAgent has confirmed the accept message received in another negotiation. The latter occurs in negotiations with decommitment.

The results show that serial negotiations are those that achieve fewer successful negotiations compared to concurrent negotiations. This is clearly explained by the nature of the negotiation: parallel negotiations allow for more negotiations. In the case of the serial negotiation, it achieves 493 successful negotiations and the serial decommit negotiation achieves 301. The difference is explained because in the serial decommit negotiation some negotiations must be repeated due to the non-commitment of some stationary agents. Therefore, in the time that the simulation lasts, it achieves fewer successful negotiations. In contrast, concurrent negotiations achieve over 1000 successful negotiations.

Figure 3 shows the distribution of the negotiation times. It can be observed that the serial, serial decommitment, and concurrent negotiations have similar average duration

times and are under 1000 ms. On the other hand, the negotiations in the concurrent decommitment mostly reach the established deadline (4000 ms). The difference between concurrent decommitment and concurrent is explained by the fact that in concurrent the number of agents participating in the negotiations decreases, and as there are fewer agents participating, the negotiation is faster. On the other hand, in concurrent decommitment, the number of agents does not decrease, since they are always participating in other negotiations with the objective of finding a better agreement in another negotiation. This results in more messages being sent and therefore more processing.

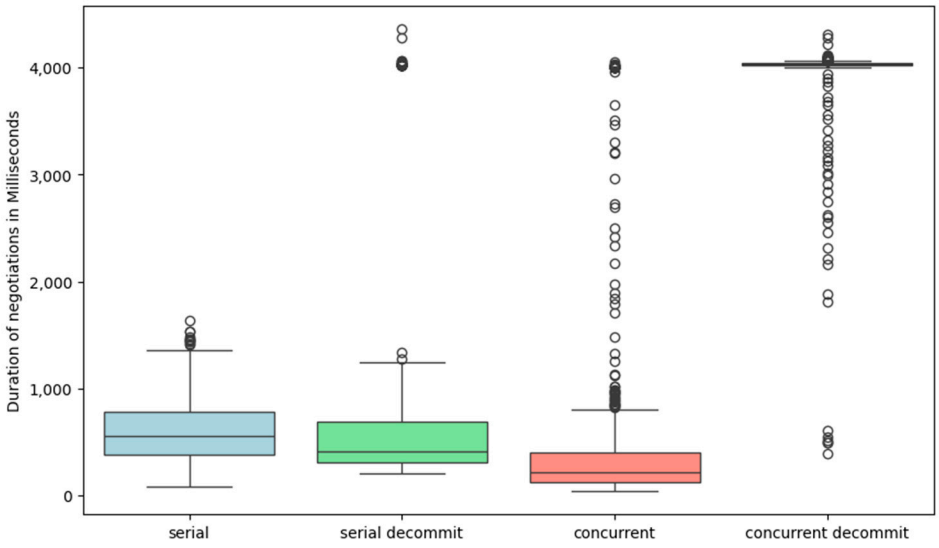


Figure 3. Boxplot showing the duration of negotiations in milliseconds across four configurations: serial, serial with decommit, concurrent, and concurrent with decommit.

Figure 4 shows the time that the MASs take to allocate all the cars. The graph illustrates the performance of the MASs in assigning 1000 cars to parking spaces over time applying the concurrent with decommitment application of the CNP. The x-axis represents time in milliseconds and the y-axis indicates the number of cars successfully assigned. The upward trend shows a progressive allocation process. Initially, the allocation rate is steep, indicating quick initial assignments. However, as time progresses, the slope decreases, suggesting a slower allocation rate as fewer parking spaces remain available. By approximately 300,000 milliseconds, the system approaches full allocation, successfully assigning all 1000 cars. The results obtained are influenced by the fact that the simulation was executed on a single computer, leading to high processing demands. In a real-world system, where agents operate on distributed devices with their own processing capabilities, the performance would likely improve. Each device would independently handle its respective agent's operations, reducing computational bottlenecks and improving the overall efficiency of the allocation process. This decentralized execution would result in faster responses and better scalability, optimizing resource utilization in real-time parking scenarios.

Quality of Allocations

The quality of allocation is evaluated by considering the number of cars allocated to parking spaces and the total number that cars must travel to reach their allocated parking space. Here, negotiations that allow dynamic adjustments, such as serial decommit and concurrent decommit, stand out. In the case of concurrent decommit, the best quality is achieved with an average distance of 22.57, followed by serial decommit with 24.8. This is

because decommits allow agents to improve their allocations, resulting in closer parking spaces for cars.

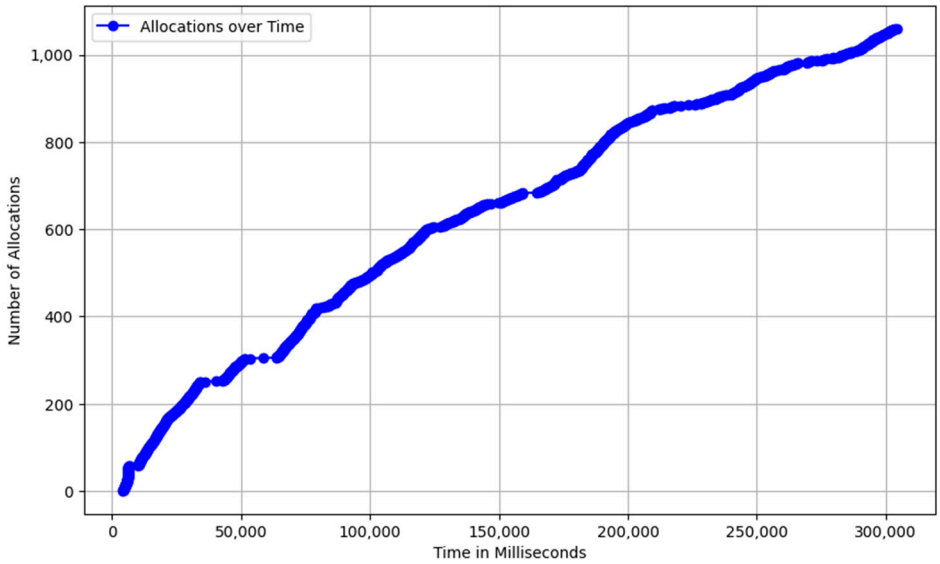


Figure 4. Time to allocate 1000 cars using the concurrent decommitment application of the CNP.

On the other hand, negotiations without decommits, such as serial and concurrent, prioritize the number of allocated cars over individual optimization. Concurrent allocates 1000 cars with a higher average distance (45.34), indicating less optimization in allocations due to the lack of renegotiations for optimizing. Serial, although efficient and with an average distance of 30.55, does not reach the quality levels observed in strategies with decommits.

In general terms, serial decommit and concurrent decommit offer better results when the priority is to maximize the quality of the allocations, while serial and concurrent are ideal for situations where it is more important to allocate a larger number of cars quickly and stably. Figure 5 shows how the quality of the allocations progresses over time.

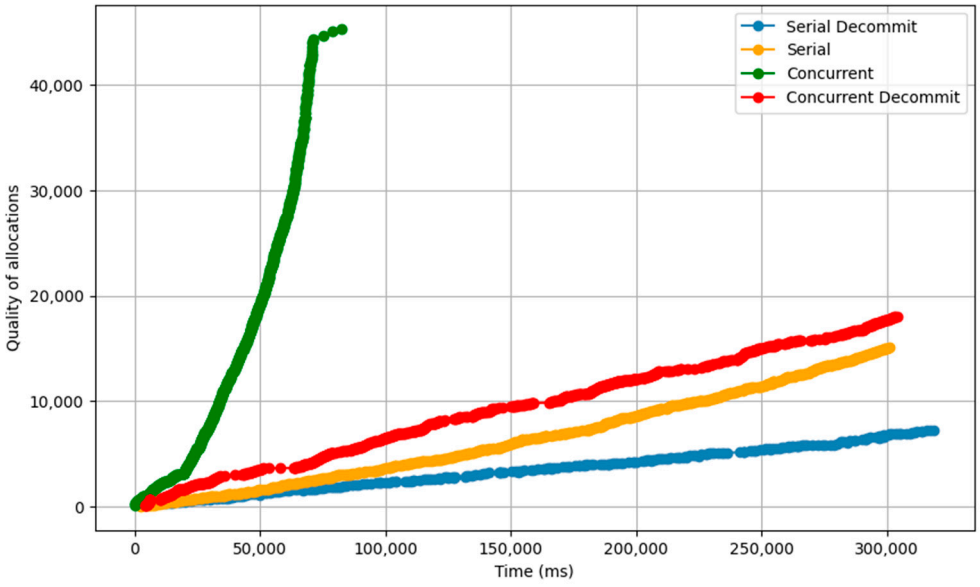


Figure 5. Quality of the allocations.

4. Discussion

Serial negotiations tend to take longer than concurrent negotiations due to the sequential nature of the process. In serial negotiations, each agent must wait for the previous one to complete its negotiation before starting its own, which increases waiting times and total allocation time. Concurrent negotiations, on the other hand, allow all agents to negotiate simultaneously, significantly reducing allocation times. The main advantage of concurrent negotiations is that they leverage parallelism, enabling the system to complete multiple allocations in a shorter period. This approach aligns with strategies discussed in multi-agent systems, where managing several negotiation processes in parallel can optimize the length of negotiations among agents [17].

Negotiations with decommitment, while adding flexibility and allowing for more optimal reallocations, often lengthen negotiation time compared to negotiations without decommitment. With decommitment, agents can change their decision if they find a more favorable option, necessitating re-negotiation. This introduces more complexity and additional processes, extending the total time. Conversely, non-commitment negotiations are faster because once an agent has made an allocation, it remains fixed, with no option for subsequent changes. This also implies that the agent does not participate in future negotiations, allowing negotiations with fewer participants and is therefore faster, as the initiator has fewer proposals to evaluate. The trade-off between flexibility and efficiency in agent negotiations has been explored in others application of MASs, highlighting the need for strategies that balance these aspects [18].

Negotiations with decommitment tend to produce higher-quality allocations, as they allow agents to change their allocation if a more favorable option arises. This leads to dynamic optimization, where agents can search for and secure the most suitable parking space based on changing circumstances. However, negotiations without decommitment, although faster, can result in suboptimal allocations, since agents are forced to stick to their first decision, even if a better option appears later. Therefore, decommitment improves quality but at the cost of increased time and complexity in the negotiation. The concept of decommitment in competitive multi-agent environments has been studied to understand its impact on negotiation outcomes.

Despite the positive results obtained, the proposed multi-agent system presents some limitations that must be considered for implementation in large-scale scenarios. One of the main limitations is scalability; increasing the number of agents (vehicles and parking spaces) and the volume of exchanged messages can significantly increase negotiation time and computational resource consumption. This issue has been previously highlighted in large-scale MAS applications, where communication overhead can impair system responsiveness [19,20]. Additionally, practical challenges include technological infrastructure constraints, such as the availability of stable, low-latency communication networks and the deployment of IoT devices that enable continuous interaction between agents and the physical environment. The integration of IoT and machine learning methods has been proposed to enhance parking management systems, effectively managing parking spaces by using real-time data and predictive algorithms [21]. To address these limitations in future research, it is suggested to integrate predictive algorithms based on machine learning that anticipate parking availability and reduce the need for real-time negotiations. Techniques such as proximity filtering and batch message aggregation could also optimize communication between agents, minimizing processing delays. Finally, the use of agent clustering and the allocation of specific roles to leader agents can enhance negotiation efficiency in highly congested urban environments, ensuring that the system remains robust and scalable. Implementing a decentralized strategy for multi-agent systems can further improve scalability and efficiency in such settings.

5. Conclusions and Future Work

Throughout this research, an analysis of concurrent and serial negotiation strategies in multi-agent systems has been carried out, with and without the implementation of decommits. The results obtained have revealed that the concurrent strategy tends to have a better performance in terms of the number of accepted agreements and the time of allocation of parking spaces, particularly when decommitment is allowed, which suggests greater flexibility and adaptability.

Future work will focus on simulating dynamic scenarios with continuous vehicle flow while also introducing enhancements to the Contract Net Protocol in its concurrent version with decommitment. These modifications aim to reduce negotiation times by optimizing the coordination process between agents. By streamlining communication and decision-making steps, the system can achieve faster convergence on parking allocations, even in complex and high-demand environments. Evaluating these improvements in simulated dynamic scenarios will provide insights into the protocol's efficiency and scalability, ensuring its applicability to real-world urban challenges. Additionally, future research should consider dynamically adjusting the weighting coefficients used in the decision-making process based on the state of the system. By adapting these values in real time, the system can prioritize either minimizing walking distance for passengers or reducing vehicle travel distance depending on traffic conditions, parking demand, and other environmental factors. This adaptive approach would enhance the flexibility and responsiveness of the multi-agent system, leading to more efficient and context-aware parking allocations.

Author Contributions: Conceptualization, G.I.-A. and Z.H.; methodology, G.I.-A.; software, V.G.; validation, F.G., M.O. and V.G.; formal analysis, M.O.; investigation, G.I.-A.; resources, V.G.; data curation, V.G.; writing—original draft preparation, G.I.-A.; writing—review and editing, Z.H., F.G. and M.O.; visualization, V.G.; supervision, G.I.-A.; project administration, G.I.-A.; funding acquisition, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MAS	Multi-agent system
CO2	Carbon dioxide
CNP	Contract Net Protocol
JADE	Java Agent Development Framework
CFP	Call for proposals
MARL	Multi-agent reinforcement learning
OPA	Optimal parking assignment
CVs	Connected vehicles
NCVs	Non-connected vehicles
MAPark	Multi-Agent Auction-based Parking
PFPs	Parking facility providers

References

1. Ayala, D.; Wolfson, O.; Xu, B.; Dasgupta, B.; Lin, J. Parking slot assignment games. In Proceedings of the 19th International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL), Chicago, IL, USA, 1–4 November 2011; pp. 299–308. [CrossRef]
2. Hassine, S.B.; Kooli, E.; Mraihi, R. Multi-agent Smart Parking System with Dynamic Pricing. In *Lecture Notes in Networks and Systems, Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*; Springer: Cham, Switzerland, 2023; pp. 781–790. [CrossRef]
3. Murano, A.; Stranieri, S.; Mittelmann, M. Multi-agent parking problem with sequential allocation. In Proceedings of the 15th International Conference on Agents and Artificial Intelligence—Volume 3: ICAART, Lisbon, Portugal, 22–24 February 2023; Science and Technology Publications, Lda: Setúbal, Portugalia, 2023; pp. 484–492. [CrossRef]
4. El Khalidi, N.; Benabbou, F.; Sael, N. Distributed parking management architecture based on multi-agent systems. *IAES Int. J. Artif. Intell.* **2021**, *10*, 801–809. [CrossRef]
5. Zhang, X.; Zhao, C.; Liao, F.; Li, X.; Du, Y. Online parking assignment in an environment of partially connected vehicles: A multi-agent deep reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2022**, *138*, 103624. [CrossRef]
6. Lin, T.; Rivano, H.; Le Mouël, F. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [CrossRef]
7. Jennings, N.R.; Sycara, K.; Wooldridge, M. A roadmap of agent research and development. *Auton Agents Multi-Agent Syst.* **1998**, *1*, 7–38. [CrossRef]
8. Wooldridge, M.J. *An Introduction to Multiagent Systems*, 2nd ed.; John Wiley & Sons: Sussex, UK, 2009; p. 461.
9. Liang, J.; Lu, Y.; Yin, G.; Fang, Z.; Zhuang, W.; Ren, Y.; Xu, L.; Li, Y. A distributed integrated control architecture of AFS and DYC based on MAS for distributed drive electric vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5565–5577. [CrossRef]
10. Liang, J.; Lu, Y.; Wang, F.; Yin, G.; Zhu, X.; Li, Y. A robust dynamic game-based control framework for integrated torque vectoring and active front-wheel steering system. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7328–7341. [CrossRef]
11. Rizvi, S.R.; Zehra, S.; Olariu, S. MAPark: A multi-agent auction-based parking system in internet of things. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 104–115. [CrossRef]
12. Jioudi, B.; Amari, A.; Moutaouakkil, F.; Medromi, H. e-Parking: Multi-agent smart parking platform for dynamic pricing and reservation sharing service. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*. [CrossRef]
13. Boudali, I.; Ouada, M.B. Smart parking reservation system based on distributed multicriteria approach. *Appl. Artif. Intell.* **2017**, *31*, 518–537. [CrossRef]
14. Smith, R.G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* **1980**, *29*, 1104–1113. [CrossRef]
15. Schillo, M.; Kray, C.; Fischer, K. The eager bidder problem: A fundamental problem of DAI and selected solutions. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, Bologna, Italy, 15–19 July 2002; Association for Computing Machinery: New York, NY, USA, 2002; pp. 599–606. [CrossRef]
16. Bellifemine, F.; Caire, G.; Greenwood, D. *Developing Multi-Agent Systems with JADE*; John Wiley & Sons: Chichester, UK, 2007. [CrossRef]
17. Aknine, S.; Pinson, S.; Shakun, M.F. An extended multi-agent negotiation protocol. *Auton. Agent. Multi Agent Syst.* **2004**, *8*, 5–45. [CrossRef]
18. Kraus, S. Automated negotiation and decision making in multiagent environments. In *Multi-Agent Systems and Applications*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 104, pp. 150–172. [CrossRef]
19. Nezamoddini, N.; Gholami, A. A survey of adaptive multi-agent networks and their applications in smart cities. *Smart Cities* **2022**, *5*, 318–347. [CrossRef]
20. Li, L.; Liu, L.; Huang, S.; Lv, S.; Lin, K.; Zhu, S. Agent-based multi-tier sla negotiation for intercloud. *J. Cloud Comput.* **2022**, *11*, 16. [CrossRef]
21. Chmaj, G.; Lazeroff, M. IoT machine learning based parking management system with anticipated prediction of available parking spots. In *Advances in Intelligent Systems and Computing, Proceedings of the ITNG 2022 19th International Conference on Information Technology-New Generations. Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2022; Volume 1421. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.