A Multi-Agent System using Fuzzy Logic to Increase AGV Fleet Performance in Warehouses

Lucas Binhardi Branisso, Edilson Reis Rodrigues Kato, Emerson Carlos Pedrino, Orides Morandin Jr.
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, São Paulo
Email: lucasbinhardibranisso@gmail.com,
{kato, emerson, orides}@dc.ufscar.br

Roberto Hideaki Tsunaki
Universidade de São Paulo
São Carlos, São Paulo
Email: rtsunaki@sc.usp.br

***Abstract*—Market competition requires an ever increasing performance from warehouses. Coupled with information technologies, high automation levels are achieved. Such automation is seen in the use of AGVs for material handling. An important problem in AGV fleets is deciding what task should be assigned to each AGV. To tackle this problem, a multi-agent AGV system is proposed, which has three agents: an AGV agent, a Loading Point (LP) agent and a Storage Point (SP) agent. The AGV agent uses a Fuzzy system to decide what task it should take, and dispatch the AGV to the location of the task, using the A-star (A*) algorithm to find the shortest path to the task. The LP agent keeps a list of all available tasks in its corresponding loading point, such as a loading dock, and handles task requests from AGV agents. The SP agent manages a particular storage space, such as a rack section, and handles AGV requests for payloads stored in the rack or requests for free space. To validate the system, a warehouse operation was simulated and evaluated measuring the average task wait time, time to complete tasks and average jam time. Two other decision methods were used, First Come First Served (FCFS) and Contract Network (CNET), to compare with the Fuzzy method. Results show that the Fuzzy method enabled a greater average task wait time reduction than the other two decision methods, and also completed tasks in less time.**

I. Introduction

Warehouses are essential in any supply chain, storing materials to cope with variability caused by seasonality or batching in production, consolidating products from various suppliers for combined delivery to customers and value-added-processing such as pricing, labeling and product customization [1]. Market competition requires high performance from warehouses and philosophies such as Just-In-Time and Lean Production brings new challenges, including tighter inventory control, shorter response time and greater product variety. The implementation of new information technologies enables real time control of the warehouse and high automation levels [1].

Automated Guided Vehicles (AGVs) are used for material handling in several facilities such as manufacturing plants, warehouses and distribution centers [2], [3]. An important issue in AGV systems is deciding what task should be assigned to a particular AGV. There are several approaches to this problem. One approach is scheduling AGVs, which can be done once before operation (offline scheduling) or updated during operation (online scheduling) [2]. Another approach is dispatching, which can be seen as a scheduling with null planing horizon [2]. Dispatching is usually done using dispatching rules, which can be simple rules based on a single criterion, such as elapsed time, or more complex rules. These complex rules often consider several criteria, and to cope with this complexity, Fuzzy Logic has been used to develop dispatching rules [4], [5].

Fuzzy Logic has been used for dispatching rules successfully because rules often take into account multiple variables, *e.g.* distances, vehicle status, traffic condition and priorities, many of them are uncertain or vague, and specialists often formulate them in natural language, *e.g.* "if the AGV is near a transport and this transport deadline is soon, it must pick it up as soon as possible". These variables are called Linguistic Variables [6], and they assume linguistic values, *e.g.*, variable distance can assume "near" as a value, instead of a real number. Linguistic values are mapped to Fuzzy Sets [7], so that Fuzzy Logic can be processed in a computer. Dispatching rules must also cover all possible scenarios, otherwise an AGV might stall if it cannot decide what to do next. In this sense, Fuzzy Logic, thanks to Fuzzy Sets, can be used to summarize several scenarios into a single rule, because Fuzzy Sets are based on real intervals. Rules can be developed with the help of an expert [8] or using Genetic Algorithms [5], [9].

Each AGV can be seen as an entity independent from other parts of the system. AGVs are also physically distributed, and their locations change often. In this sense, the fleet of AGVs has a distributed nature, in which a Multi-Agent System sits well as a methodology for developing such system [10]. Also, the warehoused used in this study is highly dynamic, which also suggests a Multi-Agent System [10]: trucks can arrive at any time, and even if trucks are scheduled to arrive at a precise time, external traffic conditions often cause delays. Scheduling becomes impractical in such scenarios, because a schedule quickly becomes obsolete due to information and overall state constantly changing. Therefore, it is better to evaluate warehouse conditions and dispatch, *i.e.* decide the AGV's next move. According to Russel and Norvig [11], "an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators". In other words, it evaluates its environment and makes a decision. It offers a convenient abstraction layer for systems development, in which only the agent's behavior must be designed. There are several examples of vehicle fleets, including AGV fleets, modeled as MAS [12], [13], [14], [15], [16].

According to literature, material handling in warehouses is increasingly done by AGVs, and AGV systems (or any vehicle fleet system) are being modeled as Multi-Agent Systems, mainly due to their distributed nature. Fuzzy Logic has been used to dispatch AGVs in several scenarios, although examples of integration of Fuzzy Logic with Multi-Agent Systems are few [17] [18]. We propose a Multi-Agent System to control material handling in a warehouse, using Fuzzy Logic to control the behavior of the AGV agents. The other two agents are the Loading Point (LP) agent and the Storage Point (SP) agent. AGV agents coordinate with LP and SP agents to acquire transport tasks, although there is no coordination amongst AGV agents themselves. Therefore, it can be said that AGV agents exhibit a competitive behavior. The system is implemented and simulated in the MASON agent simulator tool [19].

The MAS is built in this fashion because of how the considered warehouse works. This simple warehouse has a number of docks (Loading Points) where trucks can park to load or unload and a storage area, where payloads are stored in racks (Storage Points). Therefore, it is naturally suitable to use a LP agent to represent a dock and a SP agent to represent a rack. Other works use different agent models, and that depends on where the MAS is applied. For example, in [16] the transports are represented by agents, in addition to the AGV agents. This is natural in that work's environment, since a single transport appears at a location and must be moved to another predefined location. In this work, several transports (or payloads) appear at once at a dock when a truck arrives. Because they are at the same dock and arrived at the same time, the AGV agent will perceive them in the same way. Therefore, there is no need to evaluate each one of them, but something they all have in common, which is their dock, and consequently, the related LP agent. This thought process is analogous for SP agents. This is a trend seen in the works cited here: instead of devising a general solution, authors focus on the application and try to exploit its particularities, developing simpler solutions that fit well in a particular applications. This was hinted in [20]. Agent methodology makes this easier, since one has to simply devise the agents and their interactions.

We compare the approach of using Fuzzy logic for AGV agent decision making against two other approaches. The first one is the First Come, First Served (FCFS), a simple decision making method based on one criterion, the elapsed time. The second one is an implementation of a simple Contract Net Protocol [21], or CNET, a well-known protocol for task assignment in distributed systems.

The system is described in Section II. Section III shows details of the simulation used to validate the system, and Section IV shows its results. In Section V, conclusions are drawn and future works presented.

II. The Multi-Agent System

To reduce the average task waiting time in a warehouse, it is proposed a multi-agent system to control an AGV fleet that handles materials in the warehouse. Fig. 1 shows an overview of the multi-agent system, as well as agents' most important internal structures. The agents are the AGV agent, the Loading Point agent and the Storage Point agent.

A. Agents

1) AGV Agent: The AGV agent represents a single AGV, responsible for the AGV dispatch. This agent perceives the AGV surroundings, via sensors and communication. Basically, the AGV agent has to deal with two situations: select a loading point to service and select a storage point to store or retrieve a payload. The agent uses some decision method to select a point and coordinate with it, *i.e*, request a task if the point is a LP and request a payload or free space if it is a SP. Decision methods studied are shown in Section II-B and Section II-C. LP agents assign tasks to the earliest requests, leading to a competitive behavior amongst AGV agents. Problems such as warehouse full or empty are out of this work's scope, therefore AGV agents are not designed to deal with them.

2) Loading Point Agent: The Loading Point agent represents a loading point, or dock, where trucks can park to load or unload. When a truck arrives, this agent creates tasks for each payload that must be loaded or unloaded, building a task buffer, and makes them available for AGVs. If a truck is unloading, it creates storage tasks. If a truck is loading, it creates retrieval tasks. AGV agents communicate with LP agents requesting a task. The agent searches the task buffer, looking for an unassigned task. If it finds one, it assigns the task to the AGV agent which contacted the LP agent, and informs that the task has been assigned. If all tasks are already assigned, the LP agent informs the AGV agent that no task is available. The LP agent also keeps useful information about the parked truck, such as its arrival time and task request ratio, which is the ratio between tasks requested over the total available tasks. This information is useful for AGV agents in their decision process.

3) Storage Point Agent: The Storage Point agent represents a section of a storage rack. This agent keeps track of what payloads are stored in that section, the space available for payloads, what payloads are reserved for what AGVs and what free space is reserved for a payload. An AGV agent will contact a SP agent when it is looking for a transport to fulfill a retrieval task, or when they are looking for free space for storage. If an AGV agent accepts a transport from the SP agent, the latter reserves the transport to that AGV so no other can request it. Similarly, if an AGV agent decides to store a transport in the SP agent's rack, it must ask the SP agent to reserve space for it. The agent also keeps a "reserve ratio" information, which is the number of reserved spaces and payloads over total space.

B. Fuzzy Inference System

Each AGV agent uses a Fuzzy Inference System (FIS) to support decision making. The AGV agent collects information about its environment and provides this information to the FIS, which ranks the agent's options and helps it decide what transport the AGV will pickup. The FIS uses three variables as input (Distance, Waiting Time and Request Ratio), and one as output (Priority). Distance, in meters, refers to the shortest path to the destination, calculated using the A* (A-star) algorithm

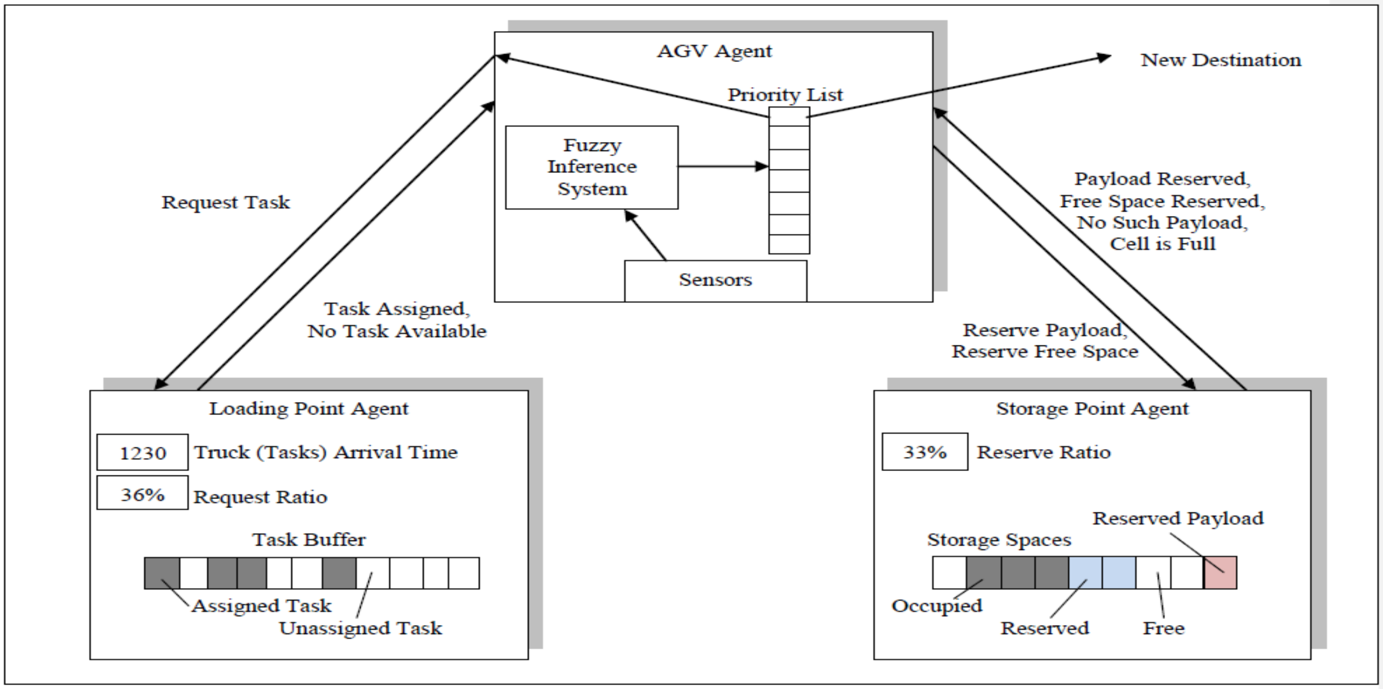

Fig. 1. Overview of the Multi-agent System

[22]. Waiting Time is the time elapsed since the truck arrived at the warehouse. Request Ratio is the ratio between number of requested tasks and number of total available tasks. This variable gives the AGV agent the notion of how congested a loading point might become. When evaluating a storage point, this variable is called Reserve Ratio, but its purpose is the same. Priority is the priority value assigned by the AGV agent to the point.

Fuzzy variables and associated Fuzzy sets are depicted in Fig. 2, Fig. 3, Fig. 4 and Fig. 5. Fuzzy rules used in the FIS are shown in Table I. First line of Table I should be read as "If distance is far and waiting time is short and request ratio is high then priority is low". The other lines are similar. Inference uses the Mamdani method.

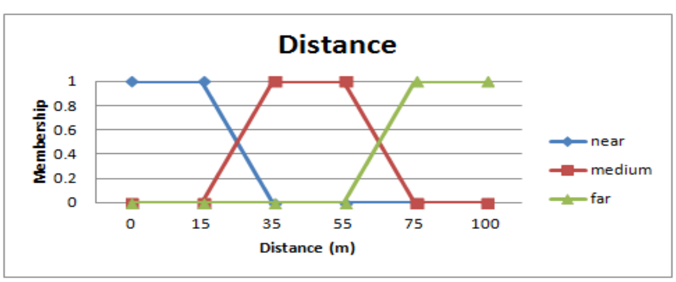

Fig. 2. Fuzzy variable Distance, in meters, and its associated Fuzzy sets.

The decision process is as follows: when the AGV is idle, its agent will check every LP agent for tasks. If the LP agent has tasks, the AGV agent will assign a priority to that point according to the Fuzzy rules. The AGV agent will calculate the distance to that LP and request the waiting time and request ratio from the LP agent. It will feed this information to the FIS, which will infer a priority. Note that

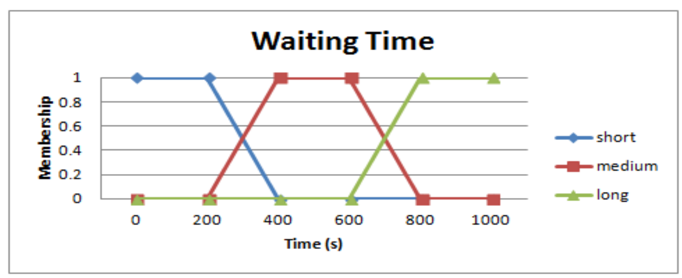

Fig. 3. Fuzzy variable Waiting Time, in seconds, and its associated Fuzzy sets.

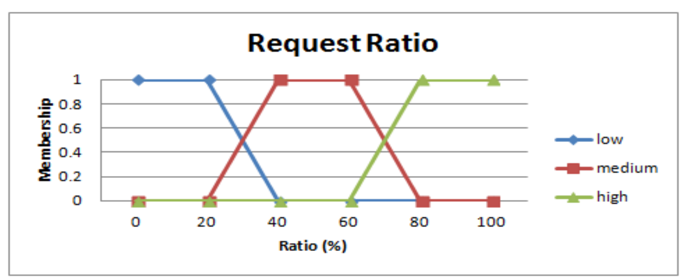

Fig. 4. Fuzzy variable Request Ratio and its associated Fuzzy sets.

this priority is defuzified, *i.e.* the value used is a real number, not the linguistic value used in the fuzzy rules. After every point has a priority, it will select the highest priority point and request a task from it. If two or more points have the same highest priority, the earliest contacted LP agent will be selected due to how the AGV agent sort its LP agents list. Priority assignment is slightly different between loading trucks and unloading trucks. If the truck is unloading, distance and request ratio are relative to the loading point. If the truck is loading,

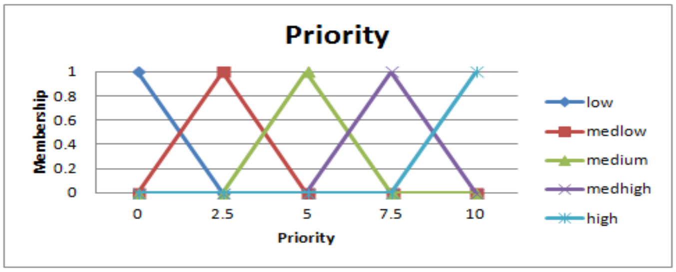

Fig. 5. Fuzzy variable Priority and its associated Fuzzy sets.

TABLE I. FUZZY RULE BASE USED IN THE CONTROLLER

Rule	Distance	Waiting Time	Request Ratio	Priority
1	Far	Short	High	Low
2	Far	Short	Medium	Medlow
3	Far	Short	Low	Medium
4	Far	Medium	High	Low
5	Far	Medium	Medium	Low
6	Far	Medium	Low	Medium
7	Far	Long	High	Medlow
8	Far	Long	Medium	Medium
9	Far	Long	Low	Medhigh
10	Medium	Short	High	Medium
11	Medium	Short	Medium	Medium
12	Medium	Short	Low	Medhigh
13	Medium	Medium	High	Medlow
14	Medium	Medium	Medium	Medlow
15	Medium	Medium	Low	Medium
16	Medium	Long	High	Medium
17	Medium	Long	Medium	Medhigh
18	Medium	Long	Low	High
19	Near	Short	High	Medium
20	Near	Short	Medium	Medhigh
21	Near	Short	Low	High
22	Near	Medium	High	Medlow
23	Near	Medium	Medium	Medhigh
24	Near	Medium	Low	High
25	Near	Long	High	Medium
26	Near	Long	Medium	High
27	Near	Long	Low	High

distance and request ratio are relative to a storage point that contains the requested transport. This avoids the undesirable scenario where an unloading truck receives a higher priority than a loading truck that is farther away and the transport it is requesting is just beside the AGV. Quite often, several storage points contain the requested transport, therefore the AGV will rank them, assigning a priority to each one, much like it does with loading points, and select the top priority storage point. This priority will also be assigned to the loading point which is requesting the transport.

C. FCFS and CNET

In addition to the FIS for AGV decision making, two other methods where tested: First Come, First Served (FCFS) and Contract Net Protocol (CNET). FCFS was chosen because of its simplicity: the AGV agent will choose the Loading Point that waited the most. It shows the importance of defining an agent behavior which is suitable to the application, *i.e.* the system is structured as a MAS, but this is not enough. CNET is a well-know protocol for task assignment in distributed systems [23]. In CNET, LP agents create an auction for each task and receive bids from AGV agents. AGV agents use the distance to the Loading Point as the bid's cost. LP agents declare the lowest cost bid as the winner.

III. SIMULATION

The warehouse in which the AGVs operate has a layout as Fig. 6. It has five loading points in which trucks can park to load or unload transports, depicted as the black squares. The storage points are depicted as gray squares, and each one can store up to five payloads. AGVs can position themselves out of paths when working in a storage point, so they do not block other AGVs. However, only one AGV can serve a point at a time and all others trying to serve that same point must wait, therefore forming up lines and these lines might block paths.

There are 11 AGVs operating material handling in the warehouse. All AGVs move at an average speed of 2 m/s, can only pick one load at a time, and perform pickup and drop operations in 30 seconds. Decision process takes one simulation step, as well as message transmission and path finding. Therefore, computational performance of these actions have no impact on the results, and, in fact, are not in the scope of this work. Message transmission details are also not in the scope of this work. Each AGV has a maintenance station. When there is no task available in the warehouse, the AGV will move to its maintenance station.

Deadlock is an important issue in AGVs routing, and it is not dealt with in this work, but avoided by designing a path network that prevents deadlocks. The path network is comprised of nodes linked by segments. Segments are unidirectional and every path loop has more nodes than AGVs to avoid cyclic deadlocks.

Truck arrivals are generated by a random generator. For every simulation run, it creates a random truck arrival schedule and this schedule is used in simulation runs for all decision methods for comparison purposes. It is important to note that the warehouse is not aware of this schedule, it is used for simulation control purposes. Every schedule contains 200 truck arrivals. At least one truck arrival occurs at time 0 (always an unloading truck, because the warehouse starts empty), and the remaining arrivals can be created until 40000 seconds of simulation, *i.e.*, the last truck arrival can occur at most at the 40000 seconds mark. There are 8 different types of payloads and trucks always carry 20 payloads of the same type. The random generator always creates feasible schedules, *i.e.*, it won't create a loading truck arrival for an empty warehouse nor an unloading truck arrival if there is no space for the truck's load.

The simulation was written in Java, on top of the multi-agent simulation tool MASON [19]. For data gathering, 100 simulation runs were done for each decision method, for a total of 300 simulation runs. For each run, a schedule of 200 trucks is generated, each with 20 tasks, for a total of 4000 tasks per simulation run. All simulations were run for 100000 seconds. The Fuzzy Inferece System was developed with the jFuzzyLogic library [24].

To measure how well AGVs can handle tasks, the average task wait time, total completed tasks and average jam time are used as metrics. Total completed tasks is the number of tasks completed in a given time through the simulation. Average jam time is the amount of time AGVs waste stuck in traffic jams. It is measured as a proportion, relative to the total operational time of the AGV fleet. Average Task wait time is calculated as in (1).

$$W = \frac{\sum T_C}{N_T} \quad (1)$$

In (1), T_C is the time for a given task to be completed, and N_T is the number of completed tasks. For an unloading truck, a task is completed when an AGV picks up one of its transports. For a loading truck, a task is completed when an AGV delivers a transport requested by the truck.

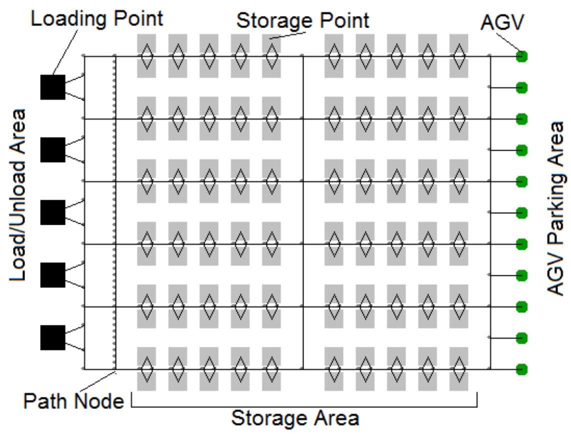

Fig. 6. Warehouse layout. Loading bays, or docks, are depicted as black squares. Storage cells, such as a rack, are gray squares. Green dots are AGVs.

IV. RESULTS

In Fig. 7, it is shown the average amount of completed tasks through simulation time. The average task waiting time through time is shown in Fig. 8. Table II summarizes the results of time to complete all tasks, maximum waiting time and throughput for both Fuzzy and CNET methods. Table III shows the average jam time for each method.

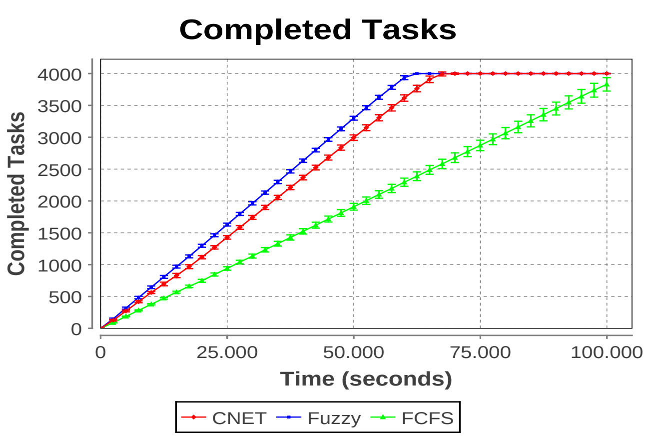

Fig. 7. Mean number of completed tasks through the simulation run

The Fuzzy decision method performs better overall than the other two methods. The FCFS rule performance is due to high traffic congestion near the loading points. Since AGVs request tasks from the loading point that waited the most, all AGVs will request tasks from the same loading point at once. Space near loading points is limited and, coupled with the time it takes for an AGV to actually pickup and drop the transport, long lines of AGVs waiting to service the loading point are formed. This rule suggests that one of the main bottlenecks of the warehouse is congestion, and this information was used to develop the fuzzy decision method.

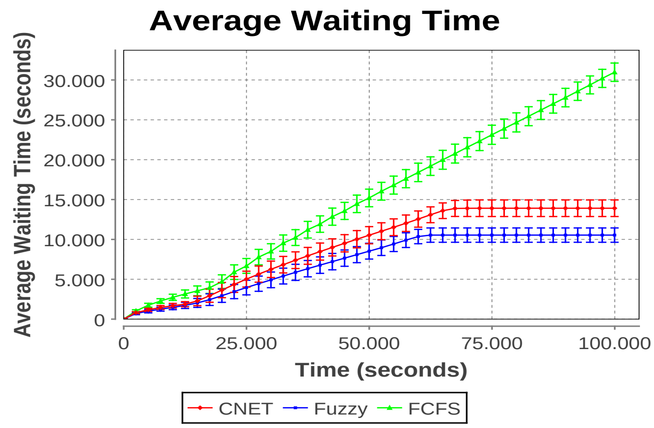

Fig. 8. Mean Average Waiting Time through the simulation run

TABLE II. SUMMARY OF SIMULATION RESULTS

Method	Time to Complete All Tasks (s)	Throughput (tasks/hour)	Maximum Average Waiting Time (s)
Fuzzy	62449	230,6	10540
CNET	71958	200,1	13909

Using the CNET method, traffic jam problem is greatly reduced to the same level as the Fuzzy method, as seen in Table III, although the latter can choose AGVs that are better suited for the task, as seen in Fig. 7 and Fig. 8. From Table II, the Fuzzy method took 13.2% less time to complete all tasks and achieved a 24.2% reduction in maximum average waiting time. Another source of wasted time in CNET is the auction itself. The LP agent has to wait a reasonable amount of time for more bids, because not all AGV agents can respond at the same time. If it waits less time, tasks will tend to have less bids. If it waits more time, more bids can be evaluated, but this time will be wasted.

The Fuzzy dispatch method was developed such that congestion, distance and time are considered together in the Fuzzy system. Therefore the variables Request Ratio, Distance and Waiting Time. Request Ratio gives the AGV a notion of how congested the area near a loading point can become. Distance lets the AGV agent account for traveling times and Waiting Time lets AGV agents give more priority to trucks that already waited long times. Decision time is kept to a minimum, compared to the time an auction takes to complete. Instead of forming a group and "discuss" what action take (the auction process), AGV agents with the Fuzzy system evaluate what they can see about their environment and choose an action (task assignment) which they think best suits the system as a whole. One problem with this approach is that, once the AGV

TABLE III. AVERAGE JAM TIME

Fuzzy	CNET	FCFS
7,8%	7,8%	47,9%

agent chooses a task, it does not revise its decision. Because of this, opportunities that may appear due to an ever-changing environment may be missed.

V. Conclusion

In this work, a multi-agent system to control a fleet of AGVs in a warehouse was successfully developed. Because of the distributed and high dynamic nature of the problem, multi-agent system methodology proved to be a natural and convenient way to model the system. It permitted the problem to be divided into sub-problems (agents and their behaviors), and then integrated (inter-agent interactions).

The multi-agent AGV system using a Fuzzy dispatching rule base successfully reduced the average task wait time, compared to a simple rule, FCFS, and a well-know task assignment method, the CNET protocol. This reduction is because of better AGVs utilization, dispatching them in a way that congestion in the warehouse is reduced (an important bottleneck because of path restrictions) and better distributing AGVs amongst loading points. The downside of the Fuzzy dispatching rule is that it requests information from all LP and SP agents, which can significantly load the network infrastructure in case of actual system deployment. This problem must be considered in future works.

Also in future works, a dynamic fuzzy decision module is being studied for development and testing. Current AGV agent decision selects a loading point, request a task from it and carry on the transportation task. The dynamic fuzzy decision also ranks loading points and request a task from the best loading point, but reconsiders its decision on its way to the loading point. If some other loading point becomes more attractive to the AGV, it gives up its current task, informing the former loading point so it can release the AGV from the task and requesting the new task from the new loading point.

The AGVs studied in this work carry only a single load at a time. There are examples of multi-load AGVs [20], and research is planned in the direction of this kind of AGV. We expect that research with multi-load AGVs can also benefit other transportation systems, such as taxis.

Acknowledgment

The authors would like to thank INCT-SEC (Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the financial support.

References

[1] J. Gu, M. Goetschalckx, and L. F. McGinnis, "Research on warehouse operation: A comprehensive review," *European Journal of Operational Research*, vol. 177, no. 1, pp. 1–21, Feb. 2007.

[2] T. Le-Anh and M. De Koster, "A review of design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 171, no. 1, pp. 1–23, May 2006.

[3] I. F. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, May 2006.

[4] D. Naso and B. Turchiano, "Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence," pp. 208–226, 2005.

[5] O. Morandin Jr, V. F. Carida, E. R. R. Kato, and C. C. M. Tuma, "Adaptive genetic fuzzy, predictive and multiobjective approach for AGVs dispatching," in *Industrial Electronics Conference*, 2011, pp. 2317–2322.

[6] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, Apr. 1988.

[7] ——, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[8] A. X. Benincasa, O. Morandin Jr, and E. R. R. Kato, "Reactive fuzzy dispatching rule for automated guided vehicles," in *Systems, Man and Cybernetics*, 2003, pp. 4375–4380.

[9] P. A. D. Castro, O. Morandin Jr, E. R. R. Kato, and H. A. Camargo, "A Genetic Fuzzy System for Defining a Reactive Dispatching Rule for AGVs," in *IEEE international Conference on Systems, Man and Cybernetics*, 2007, pp. 56–61.

[10] M. Wooldridge, *An introduction to Multi-Agent Systems*, 2002.

[11] S. J. Russel and P. Norvig, *Artificial Intelligence - A modern Approach*, 3rd ed. Pearson Education, 2010.

[12] K. T. Seow, N. H. Dang, and D.-h. Lee, "A Collaborative Multiagent Taxi-Dispatch System," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 607–616, Jul. 2010.

[13] S. Ibri, M. Nourelfath, and H. Drias, "A multi-agent approach for integrated emergency vehicle dispatching and covering problem," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 554–565, Apr. 2012.

[14] P. Farahvash and T. O. Boucher, "A multi-agent architecture for control of AGV systems," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 6, pp. 473–483, Dec. 2004.

[15] O. Gaci and H. Mathieu, "A Multi-Agent System to simulate storage activities in warehouse of chemical substances," in *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*. Ieee, Jun. 2012, pp. 435–440.

[16] D. Weyns and T. Holvoet, "Architectural design of a situated multiagent system for controlling automatic guided vehicles," *International Journal of Agent-Oriented Software Engineering*, vol. 2, no. 1, p. 90, 2008.

[17] P. G. Balagi, D. Srinivasan, and C.-K. Than, "Coordination in distributed multi-agent system using type-2 fuzzy decision systems," in *International Conference on Fuzzy Systems*, 2008, pp. 2291–2298.

[18] Y. Li, J.-Q. Zheng, and S.-L. Yang, "Multi-agent-based fuzzy scheduling for shop floor," *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 689–695, Nov. 2009.

[19] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, "MASON: A Multiagent Simulation Environment," *Simulation*, vol. 81, no. 7, pp. 517–527, Jul. 2005.

[20] N. Singh, P. V. Sarngadharan, and P. K. Pal, "AGV scheduling for automated material distribution: a case study," *Journal of Intelligent Manufacturing*, vol. 22, no. 2, pp. 219–228, Jul. 2009.

[21] R. G. Smith, "The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of Systems Science and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107, 1968.

[23] D. Weyns, N. Boucké, T. Holvoet, and B. Demarsin, "Dyncnet: A protocol for dynamic task assignment in multiagent systems," in *Conference on Self-Adaptive and Self-Organizing Systems*, no. Saso, 2007, pp. 281–284.

[24] P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation," in *2012 IEEE International Conference on Fuzzy Systems*. Ieee, Jun. 2012, pp. 1–8.