Improved dynamic resource reservation-based AGV traffic control with optimized task allocation

Parikshit Verma
R&D Department
TAVIL IND, S.A.U.
S. Jaume de Llierca, Spain
parikshitverma@tavil.net

Josep M. Olm
Inst. of Indust. and Control Eng.
Univ. Politècnica de Catalunya
Barcelona, Spain
josep.olm@upc.edu

Raúl Suárez
Inst. of Indust. and Control Eng.
Univ. Politècnica de Catalunya
Barcelona, Spain
raul.suarez@upc.edu

Pol Toldrà
R&D Department
TAVIL IND S.A.U.
S. Jaume de Llierca, Spain
poltoldra@tavil.net

***Abstract*—Traffic management of multi-AGV fleets through improved dynamic resource reservation (IDRR) has been recently proven to be an interesting alternative within zone control-based approaches in terms of time completion efficiency in in-house transportation tasks. In this paper, preliminary results show that the combination of IDRR with PRIM allocation, a classical market-based task allocation strategy, yields a significant reduction of finishing times in a standard benchmark problem. Hence, further improvements are to be expected from deeper suitability analyses plus better tailored adaptations of optimal task allocation algorithms to zone control-based fleet management systems, and in particular to IDRR.**

***Index Terms*—AGVs, improved dynamic resource reservation-based traffic control, optimized task scheduling**

I. Introduction

In-house material handling and transportation with Automated Guided Vehicles (AGVs), see Figure 1, is undergoing a sustained growth due to efficiency and reliability reasons. This, in turn, attracts an increasing research interest aimed at improving current systems and also at adapting them to the new scenarios associated to the Industry 4.0 paradigm [1].

A key element of a multi-AGV (MAGV) system is the traffic management strategy, in charge of addressing a conflict-free evolution of the AGV fleet while completing the assigned tasks, i.e. carriages of load within the factory between pick-up and drop-off stations. Among the number of strategies available in the literature for this purpose [2], [3], a leading position is hold by zone control-based algorithms [4]. Essentially, the layout is divided into zones, and AGVs are granted access to them according to specific sets of rules specifically targeted for collision and deadlock avoidance. In fact, the main difference between zone control algorithms lies on the management of the access to the so-called shared route areas, which are common zones in the path to be followed by two or more AGVs while performing their assigned tasks.

P. Verma was supported by the Generalitat de Catalunya Project 2021 DI 16. J.M. Olm was partially supported by the Government of Spain Project PID2021-122821NB-I00 funded by MICIU/AEI/10.13039/501100011033 and ERDF/EU, and by the Generalitat de Catalunya Projects 2021 SGR 00376 and 2021 DI 16. R. Suárez was partially supported by the Government of Spain Project PID2020-114819GB-I00 funded by MICIU/AEI/10.13039/501100011033 and ERDF/EU, and by the Generalitat de Catalunya Project 2021 DI 16.

Fig. 1. A palletizer AGV.

Static strategies, as the Chain of Reservations (COR) [5], reserve the shared route of the entire task path for a single AGV and in a sequential manner. All zones in the shared route are considered busy until the AGV has fully traversed them, and it is only at this time that access to the next reserving AGV is allowed therein. Instead, the Dynamic Resource Reservation (DRR) strategy [6] computes the shared route in real time, and zones are freed not as a whole but one after the other, which definitely shortens waiting times.

Both COR and DRR are based on conflict prevention. i.e. AGVs are stopped at appropriate crossings to prevent collisions and deadlocks. However, the recently reported Improved Dynamic Resource Reservation (IDRR) [7] adds a conflict detection & resolution option that takes advantage of aisle crossing areas to resolve specific conflicts. This allows multiple access to shared routes under certain constraints, which reduces completion times w.r.t. DRR-commanded AGV fleets. Nevertheless, in the original version [7], tasks are allocated in a FIFO-like manner: AGVs are denoted as R_i, $i \geq 1$, and the list of idle AGVs is ordered according to the i-th labels; hence, any new task is assigned to the first AGV in such a list. Consequently, a natural path to explore when seeking for further improvements to IDRR performance is task allocation.

Two main approaches are distinguished in Multi Agent Task Allocation (MATA) algorithms [1]: (i) optimization-based solutions, and (ii) market-based solutions. Most research done on the former uses global information and looks for an optimal solution according to different cost functions and criteria. Despite its accuracy, the associated computational burden scales up badly with the number of AGVs in the system. This is why the second option, which can be implemented in an easier and decentralized manner, is taken as a sound alternative for researchers and manufacturers despite the fact that they provide sub-optimal solutions.

In market-based approaches, an auctioneer (for instance: a central controller, a station or a robot) offers tasks to AGVs, which submit their bids according to a computation of individual costs of task executions usually based on local information. The task is finally assigned to the robot with better bid. In turn, auctioning processes can be classified as [1]: (i) sequential single-item, where one single task is auctioned at a time, (ii) parallel single-item, with AGVs bidding on a group of tasks, with a different bid for each one, but just one task is assigned per bidding round, and (iii) combinatorial, where bids are presented for a cluster of tasks, which is allocated to the bid winning AGV. Although combinatorial auctioning provides near-to-optimal results, the computational load scales up very quickly with the fleet size. Instead, sequential single-item has been reported to provide the best trade-off between optimality of the solution, computational burden, and implementation difficulty [8].

PRIM allocation [9], although formally falling within the parallel single-item category, is also easily implementable and takes partial advantage of sequential single-item, as multiple rounds are conducted for unallocated and unexecuted tasks. Besides, it has been reported to show superior performance than sequential single-item strategies in the event of a harsh communication scenario, because it enables more AGVs to participate in tasks and also tends to provide better quality solutions [10]. Hence, it becomes an interesting option for industrial fleets operating in large warehouses.

This manuscript explores the benefit of including an optimized task allocation loop, in particular PRIM allocation, to the IDRR traffic management policy. A preliminary validation in a benchmark problem borrowed from [11] shows promising improvements up to a 26% in task completion times, which encourages putting effort in this area of research.

The remainder of the paper is organised as follows. Basics of IDRR and PRIM allocation strategies are briefly introduced in Sections II and III, respectively. A numerical validation is carried out in Section IV, including a description of the experimental setup and the corresponding discussion. Finally, conclusions and further research goals are drawn in Section V.

II. The IDRR traffic management policy

As already mentioned in Section I, IDRR relies on conflict prevention plus detection & resolution. Namely, while some of the conflicts are prevented by banning the access to shared routes through waiting actions, in others cases multiple access into shared routes are allowed. Then, advantage is taken of the assumed manhattan-like geometry of the workspace to conduct

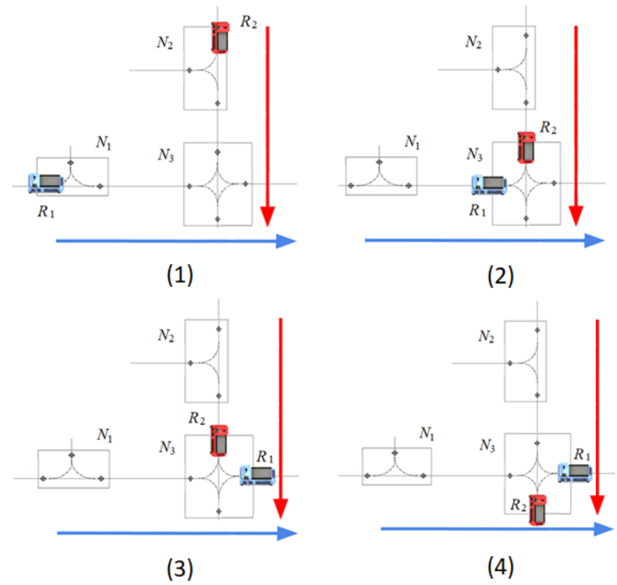

Fig. 2. Resolution of an intersection conflict with IDRR, from the initial (1) to the final position (4).

conflict resolutions in aisle crossings. The algorithm is briefly described below, while further details are available in [7].

IDRR uses a double layer architecture: a symbolic one, represented by the layout's graph model and essentially used in route planning, and a geometric one, constituted by points termed as Control Points (CPs), that effectively drive path executions. A single CP is associated to nodes representing pick-up, drop-off and parking stations, while crossings have as many CPs as aisles meet at the intersection. It is in crossing nodes where conflict resolutions may take place.

In the face of conflicts IDRR acts according to the detected issue. In case of pursuits, either one-to-one or closed loops of them, AGVs are forced to wait until the potential conflict ahead vanishes. Instead, for intersections and head-on conflicts AGVs are allowed to proceed into the shared route till they meet at a crossing, where the conflict is therefore solved. Besides, IDRR does not allow more than two AGVs at crossing nodes, which enables any of them to accommodate intersection or head-on resolutions, and it also bounds the occupancy in adjacent nodes to three AGVs, which prevents deadlocks due to path saturation. The resolution of an intersection conflict is illustrated in Figure 2.

A final distinct feature of IDRR with respect to other zone control-based approaches, in particular DRR, is that AGVs can undertake new tasks from the ending point of the previous one, with no need to return to its parking slot. The combined effect of multiple access to shared routes plus the parking policy renders excellent task completion times w.r.t. DRR [7].

III. PRIM allocation

PRIM allocation [9] is a multi-round single-item auction algorithm. Essentially, (i) a list of tasks is auctioned at every round; (ii) AGVs compute a distance-based bid using Prim's

Algorithm 1 PRIM Task allocation

1: **while** the system is active **do**
2: Create a set T_N of new tasks.
3: **if** $T_N \neq \emptyset$ **then**
4: Create a set T_U of assigned but unexecuted tasks.
5: Create the auction set $T_A = T_N \cup T_U$.
6: **while** $T_A \neq \emptyset$ **do**
7: Create a set of bids $B = \emptyset$.
8: **for** each AGV R_i **do**
9: Create a set of bids $B_i = \emptyset$.
10: **for** each task $T_k \in T_A$ **do**
11: Compute bid $B_{i,k}$.
12: $B_i = B_i \cup \{B_{i,k}\}$.
13: **end for**
14: Let $B_{i,l} = \min\{B_i\}$.
15: $B = B \cup \{B_{i,l}\}$.
16: **end for**
17: Let $B_{j,m} = \min\{B\}$,
18: Assign task T_m to AGV R_j.
19: $T_A = T_A \setminus \{T_m\}$.
20: **end while**
21: **end if**
22: **end while**

or Kruskal's algorithms for every task and place the lowest of all bids, i.e. each AGV actually bids for just one task; (iii) the auctioneer collects the bids and assigns the task with lowest bid to the corresponding AGVs, i.e. just one task is allocated at every round. Subsequent rounds are conducted as long as there exist unassigned tasks, with the specificity that tasks already allocated but not yet executed are also included in the auction list, thus giving room to further improve initial assignments.

Interestingly, AGVs compute bids using marginal costs [1]: the bid for a new task considers not only the distance from the AGV's current location, but also the distances from other assigned but not yet executed tasks the AGV may have in its queue list. This, in turn, renders a local optimization of the task sequencing in the to-do list of each AGV.

A pseudocode for PRIM's allocation is shown in Algorithm 1 which, for the sake of understanding, considers a centralized implementation. While the system is active (line 1), in each iteration a list T_N is created including all new tasks that have entered in the system and are ready to be assigned (line 2). If T_N is not empty (line 3), a list of already assigned but unexecuted tasks, T_U, is also created (line 4). The list of tasks to be auctioned, T_A, is created as the union of T_N and T_U (line 5), and while T_A is not empty (line 6) , an auxiliary empty list of bids, B, is created as well (line 7). Then, for each AGV R_i (line 8), an empty list of bids, B_i, is created (line 9), and a bid for each task in T_A is computed (lines 10-11), which is added to the list B_i (lines 12-13); in turn, the smallest bid in B_i is selected (line 14) and added to B (line 15). Once all the AGVs have added one bid to B (line 16), the smallest bid in B is selected (line 17), the corresponding assignment task-AGV is done (line 18), and such a task is removed from the auction task list, T_A (line 19). Then, subsequent bidding rounds are conducted till T_A is empty (line 20). Once at this point, and while the system is active, the process starts again creating the list of new tasks, T_N, that have lately entered and require allocation (line 2).

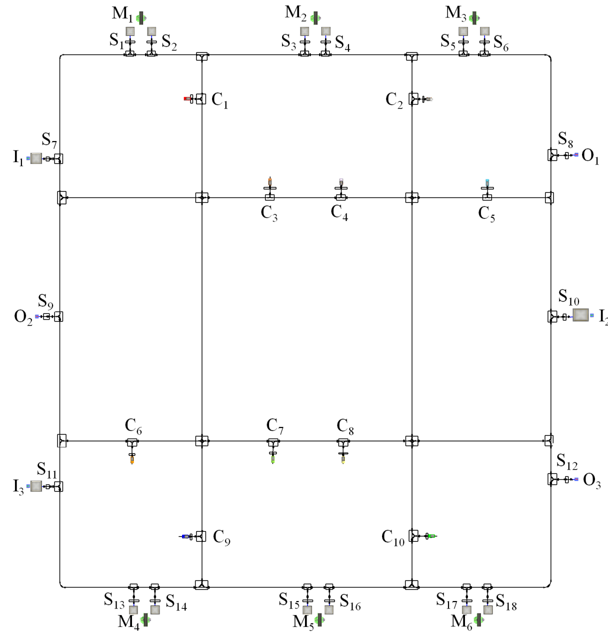

Fig. 3. The benchmark layout.

IV. Numerical results

In-silico experiments were run using the simulation software Flexsim for the benchmark layout and task rationale borrowed from [11] and also used in [7], see Figure 3. The 50 m $\times$ 40 m square workspace includes (i) three source stations for work pieces, I_1-I_3; (ii) three sinks, O_1-O_3; six machines, M_1-M_6; nine pick-up points, S_2, S_4, S_6, S_7, S_{10}, S_{11}, S_{13}, S_{15}, S_{17}; nine drop-off stations, S_1, S_3, S_5, S_8, S_9, S_{12}, S_{14}, S_{16}, S_{18}, and ten parking stations, C_1-C_{10}. Each source station has 20 pieces, which have to be dropped at the corresponding output station after going through two different machines, where a 20 s processing time takes place in each one. This amounts a total of 180 tasks to be performed by the system. The AGVs commence from the parking stations, move at 1 m/s, and the simulation concludes when all the pieces are at the sinks and the fleet is fully parked.

Simulations were carried out for a number of AGVs ranging from 1 to 10 using IDRR plus PRIM allocation (PIDRR) and compared to IDRR plus FIFO allocation (FIDRR). In turn, two metrics were considered to assess the performance: ending times and total travelled distance. The results, available in Table I left and right, respectively, indicate that FIDRR is significantly outperformed by PIDRR. The percentage improvements, shown in Figure 4, lie between 12% and 26% with an average of 19% for the ending time, and between 18% and 25% with an average of 23% for the total travelled distance.

TABLE I

ENDING TIMES [S]

AGVs	FIDRR	PIDRR
1	20139	15145
2	10668	9047
3	7818	6265
4	6022	5266
5	5249	4598
6	4732	4003
7	4404	3512
8	4068	3257
9	3979	2962
10	3957	3062

TOTAL TRAVELLED DISTANCE [M]

AGVs	FIDRR	PIDRR
1	18923	14177
2	18541	15164
3	19808	15116
4	19480	15404
5	19705	15673
6	19722	15396
7	20619	15258
8	20306	15542
9	20448	15440
10	20578	16129

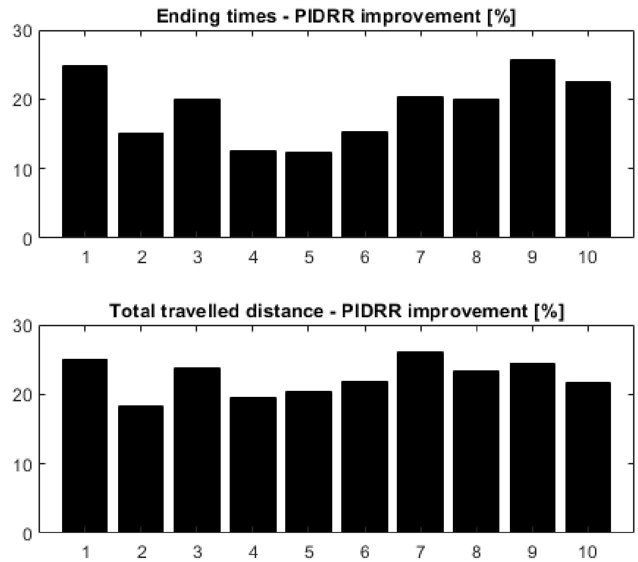

Fig. 4. Percentage improvement of PIDRR w.r.t. FIDRR in ending times (top) and total travelled distance (bottom).

Finally, the task distribution between AGVs dispatched by FIDRR and PIDRR is shown in Table II, where rows represent the AGV IDs R_i, $i = 1, \ldots, 10$, and columns display the total number of AGVs working in the system for each simulation. Notice that both algorithms achieve a similar balance, with slightly higher differences in the PIDRR strategy.

V. CONCLUSIONS

A market-based algorithm for task scheduling optimization in an AGV fleet, namely PRIM allocation, was added to the IDRR traffic management policy. Preliminary results showed a significant improvement in ending times and total travelled distance, up to a 26%, w.r.t. a plain IDRR strategy. This not only confirms the effectiveness of task optimization loops in improving the overall efficiency of AGV fleet control architectures, but also fosters the research in the area seeking for even better results.

Further improvements are expected by exploring the combination of IDRR with alternative, probably sequential single-item-based auction methods, analysing the potential influence of the layout geometry and fleet scalability in the overall effectiveness of the optimization algorithm, proposing more accurated bid calculation functions, and extending the scope of the study to other zone control-based traffic management algorithms.

TABLE II

TASK DISTRIBUTION

TOP: FIDRR BOTTOM: PIDRR

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
R_1	180	87	59	45	36	31	26	22	21	18
R_2		93	59	48	35	30	26	21	19	18
R_3			62	43	38	29	27	23	21	19
R_4				44	36	31	25	22	19	17
R_5					35	29	25	22	20	19
R_6						30	26	24	20	17
R_7							25	23	21	17
R_8								23	20	19
R_9									19	19
R_{10}										17

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
R_1	180	93	61	49	40	35	30	32	27	16
R_2		87	61	45	43	31	27	17	20	24
R_3			58	42	40	30	26	21	20	16
R_4				44	35	26	17	22	18	20
R_5					22	26	27	24	18	23
R_6						32	27	28	27	22
R_7							26	19	18	17
R_8								17	18	16
R_9									14	14
R_{10}										12

REFERENCES

[1] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *Journal of Manufacturing Systems*, vol. 54, pp. 152–173, 2020.

[2] Z. Yunlong, L. Xiaoping, W. Shaobo, and W. Gang, "Spare zone based hierarchical motion coordination for multi-AGV systems," *Simulation Modelling Practice and Theory*, vol. 109, 2021, 102294.

[3] J. Chen, X. Zhang, X. Peng, D. Xu, and J. Peng, "Efficient routing for multi-AGV based on optimized Ant-agent," *Computers & Industrial Engineering*, vol. 167, 2022, 108042.

[4] Q. Li, J. T. Udding, and A. Pogromsky, "Zone-control-based traffic control of automated guided vehicles," *Lecture Notes in Control and Information Sciences*, vol. 456, pp. 53–60, 2015.

[5] W. Małopolski, "A sustainable and conflict-free operation of AGVs in a square topology," *Computers & Industrial Engineering*, vol. 126, pp. 472–481, 2018.

[6] Y. Zhao, X. Liu, W. Gang, S. Wu, and S. Han, "Dynamic resource reservation based collision and deadlock prevention for multi-AGVs," *IEEE Access*, vol. 8, pp. 82 120–82 130, 2020.

[7] P. Verma, J. M. Olm, and R. Suárez, "Traffic management of multi-AGV systems by improved dynamic resource reservation," *IEEE Access*, vol. 12, pp. 19 790–19 805, 2024.

[8] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain, "The power of sequential single-item auctions for agent coordination," in *Proc. 21st National Conf. on Artificial Intelligence*, vol. 2, 2006, p. 1625–1629.

[9] M. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2004, pp. 698–705.

[10] M. Otte, M. Kuhlman, and D. Sofge, "Multi-robot task allocation with auctions in harsh communication environments," in *Proc. Internat. Symp. on Multi-Robot and Multi-Agent Systems (MRS)*, 2017, pp. 32–39.

[11] J. Zając and W. Małopolski, "Structural on-line control policy for collision and deadlock resolution in multi-AGV systems," *Journal of Manufacturing Systems*, vol. 60, pp. 80–92, 2021.