

# Smartness Versus Embeddability: A Tradeoff for the Deployment of Smart AGVs in Industry

Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, Mohamed Djemai, André Thomas

▶ **To cite this version:**

Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, et al.. Smartness Versus Embeddability: A Tradeoff for the Deployment of Smart AGVs in Industry. Theodor Borangiu; Damien Trentesaux; André Thomas; Olivier Cardin. Service Orientation in Holonic and Multi-Agent Manufacturing. Proceedings of SOHOMA 2017, 762, Springer International Publishing, pp.395-406, 2018, Studies in Computational Intelligence, 978-3-319-89257-3. 10.1007/978-3-319-73751-5_30 . hal-03411175

**HAL Id: hal-03411175**
**https://uphf.hal.science/hal-03411175**

Submitted on 28 Apr 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Smartness versus embeddability: a tradeoff for the deployment of smart AGVs in industry.

Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, Mohamed Djemai, André Thomas

▶ **To cite this version:**

Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, et al.. Smartness versus embeddability: a tradeoff for the deployment of smart AGVs in industry.. 7th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, SOHOMA'17, Oct 2017, Nantes, France. hal-01643083

**HAL Id: hal-01643083**

**https://hal.archives-ouvertes.fr/hal-01643083**

Submitted on 21 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Smartness versus Embeddability: a tradeoff for the deployment of smart AGVs in industry

Guillaume Demesure[1,2], Damien Trentesaux[1], Michael Defoort[1], Abdelghani Bekrar[1], Hind Bril[2], Mohamed Djemaï[1], André Thomas[2]

[1] LAMIH, CNRS UMR 8201,UVHC, F-59313, Valenciennes, France, {damien.trentesaux, michael.defoort, abdelghani.bekrar, mohamed.djemai} @univ-valenciennes.fr

[2] Université de Lorraine, CRAN, CNRS UMR 7039, Campus Sciences, 54506 Nancy, France, {guillaume.demesure, hind.el-haouzi, andre.thomas}@univ-lorraine.fr

**Abstract** In order to deploy AGVs in industry, it is mandatory to consider the tradeoff between the smartness and the embeddability. This paper aims to make the manufacturing research community more sensitive about this tradeoff and its consequences. Nowadays, AGVs are widely chosen by manufacturers to implement flexible material-handling systems which are necessary to cover the industrial requirements. However, many issues, presented in this paper, must be tackled to deploy these AGVs. A tradeoff–oriented procedure is proposed by considering these issues in flexible manufacturing system applications. Then, an approach is proposed to illustrate this procedure by providing simulation and experimental results. This approach is also used to roughly describe the smartness/embeddability tradeoff.

**Key words:** Automated Guided Vehicles; flexible manufacturing systems; smartness; embeddability.

## 1 Introduction

Day after day, the industrial requirements evolve and become stricter [17], where reactivity in the short term and adaptability to the market in the long term become harder to obtain. This can lead the industrials to reconsider their manufacturing plants by applying new approaches to cover these requirements. In this context, flexible manufacturing systems have several interests to consider automated guided vehicles (AGVs). Indeed, since the technological evolution in mechatronics, computer science and Information and Communication Technologies (ICT) allows improving the use of AGV while limiting their cost [19], the scope of reactive behavior in the dynamic routing of AGVs is enlarged. In this paper, AGV-based flexible manufacturing systems are considered where products must be completed according to

some manufacturing specifications. The AGVs, used as product transportation systems, are assumed to navigate freely on the production floor.

The use of AGVs in manufacturing plants allows improving their flexibility in terms of material–handling or routing, aiming for greater responsiveness to industrial requirements. However, it needs considering several AGV aspects for an efficient functioning of their plants. For example, the embeddability and the feasibility of designers' approaches must be tackled before the AGV deployment in industry. The feasibility allows proving that an approach is technically feasible and economically profitable. The embeddability [11] refers to the capacity to embed enough communication, computational and energy devices in AGVs. Based on the holon paradigm, an AGV can be considered as a resource holon in a HMS since decisional capabilities are embedded [3]. The embeddability depends on the smartness level of their functions. Indeed, having a high–level of smartness means that the embedded functions have the capacity to deal with complex situations. However, the AGVs may not have the capacity to apply them correctly in short time, leading to computational overload which impacts the feasibility. Conversely, because of limited available processing capabilities, the AGVs may only be able to deploy simple functions where their lack of smartness prevents them from dealing with complex situations and being used at their full potential. Thus, one can see the AGV tradeoff between their smartness and their embeddability.

To deal with this tradeoff, the manufacturing research community and the industrial one design their approach differently. The industrial community deals more with the embeddability by designing simple embeddable functions with little attention being paid of the AGV smartness. Conversely, the manufacturing research community tends to design complex approaches by improving this smartness. However, the complexity to solve their smart approach is hardly compensated by the technical evolution. For example, the navigation is one of AGV functions which could be hard to embed. Indeed, the navigation tools are often complex since they need to be smart enough to prevent unexpected situations (e.g. deadlock [8], conflicts [7], local minima [3] ...). Therefore, this paper is focused on approaches which include a navigation function, especially due to its major impact on the tradeoff.

Among the several navigation tools, motion planners are widely used [13], allowing the generation of collision–free trajectories between two configurations. Since the AGV navigation is related to the production performances, mathematical programming seems to be an interesting tool [5]. However, the motion planning problems are NP-Hard [4], leading to reduce the computational time with meta-heuristic algorithms [14] or with discretization of the navigation area [16]. Even if the navigation may prevent the AGV deployment in industry, some approaches have been proposed in indoor environments. The Kiva system [20], which is currently used in Amazon warehouses [9], uses a standard implementation of $A^*$ algorithm to plan path. In [12], the AGV navigation is solved using a $D^*$ algorithm coupled with other functions.

The main purpose of this paper is to make the designer of AGV-based flexible manufacturing systems more sensitive about this existing tradeoff. Based on a design for "x" paradigm [10], designing for tradeoff is more suitable than only considering the smartness or the embeddability. Hence, the designers of AGV–based flexible manufacturing systems need to consider new approaches or reconsider their previous ones to deal with this tradeoff. Therefore, this article can be viewed as a basis where the main issues are presented and are related to a procedure representing the several design and test steps. An illustrative example is provided by proposing an approach tested in simulation and experimentally environment. This example highlights the consequences of the smartness/embeddability tradeoff.

The paper is organized as follows. Section 2 presents the main issues that designers may tackle using AGV functions. A design procedure, allowing to highlight the tradeoff and consequences, is proposed in Section 3. The illustrative example with results and discussions are provided in Section 4.

## 2 Issues which may impact the tradeoff

In order to deal with the mentioned tradeoff, it is necessary to introduce the main issues that AGV may solve using smart functions. Firstly, collision-free trajectories between resources are required since AGVs navigate freely on the production floor. These trajectories (path) have to be feasible due to the physical behaviors and constraints. The disturbances and discrepancies must be taken into account to prevent deviation from the planned trajectory. Furthermore, all issues related to localization, mapping and actuators/sensors must be tackled. Moreover, communication devices are required to send information in short time between AGVs and with the higher level of control (e.g. resources, supervisor...). The energy is another issue to tackle since the battery has limited capacities. Indeed, the battery charge/discharge may change the decisions that AGVs make according to the task they have to perform.

Secondly, several other issues are related to the AGV environment since they navigate in a flexible manufacturing system. For example, manufacturing disturbances such as machine breakdown may occur at any time which impacts the AGV decisions. As each AGV computes its own functions, the control architecture is fully or partially distributed. In spite of improving the reactivity, this architecture is more complex in structure and organization and may lead to myopic behaviors [17]. To remove these drawbacks, the control architecture must be adapted (by including a supervisor for example) and the interactions between the different entities (defined in [17]) must be tackled to prevent decision conflicts [2].

The last issues are related to the industrial environment and the AGV deployment. Human factors must be taken into account because human workers may impact the AGV-based flexible manufacturing system. For example, hu-

mans may work on some resources or AGVs for maintenance purpose. The humans must be able to understand the AGV behaviors as well as to take control of the system in case of failure. Therefore, safety functions, allowing to prevent collisions with humans, and AGV/human cooperation mechanisms must be embedded in the AGVs.

All the issues mentioned above are related to the smartness of AGVs. However, the embeddability issue may prevent them to improve their smartness. Hence, the number of functions that they cover may be limited, due to technically and/or economically reasons. It may lead to reconsider their functions (e.g. the number of functions and their role) in order to reach a good tradeoff. This tradeoff can be seen as a balance where the smartness and the embeddability, respectively depending on the AGV functions and their device capabilities, are on each side of the balance as shown in Fig. 1.

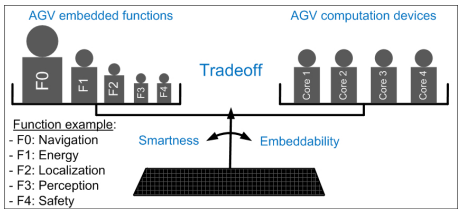



**Fig. 1** Tradeoff balance between smartness and embeddability

One can see on this figure that improving the smartness increases the weight on the left side of the balance. To compensate, the embeddability has to be improved by adding some weights in the right side. However, it impacts the deployment cost since AGV with better capabilities are more expensive. Thus, this cost becomes an important issue since their deployment needs to be economically profitable, allowing a good return on the investment. From these statements, the embeddability is limited by this cost, preventing the improvement on the smartness of AGV functions.

## 3 Proposed tradeoff–oriented procedure

The design of AGV approaches in flexible manufacturing system depends on production objectives and specifications. To cover these requirements, the AGVs must tackle one or several issues by using embedded functions. To keep the AGV benefits, it is often recommended to follow a design procedure while thinking about the next steps where some issues may occur. In Fig. 2, a tradeoff–oriented procedure is proposed by providing several design steps and tests until the AGV deployment. Each step has its own role but may depend on other steps. The role of each step is described as follows:

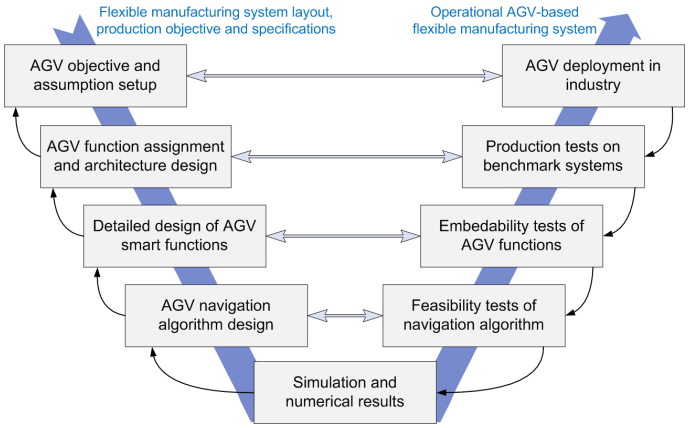



**Fig. 2** Main steps of AGV design procedure until the deployment in industry

- *AGV objective and assumption setup*: In this step, the problem is set for each AGV by defining their objective. Furthermore, several assumptions can be proposed to make the problem simpler. During the next steps, these assumptions may be reconsidered to take more issues into account.
- *AGV function assignment and architecture design*: This step is used to define the different functions of the AGV. The functions that AGV do not cover must be tackled by other entities (e.g. supervisor). Thus, the control architecture, where the interactions between the different entities are studied, must be designed to prevent conflicts between them.
- *Detailed design of AGV smart functions*: The AGV functions, including the navigation one, are detailed in this step. If the tradeoff between smartness and embeddability is not reached, some functions may be exchanged with the manufacturing high-level of control.
- *AGV navigation algorithm design*: After defining the functions, the navigation tools of AGVs must be designed such as the motion planner, the control strategy, the used algorithm (e.g. $A^*$, meta–heuristics, ...)
- *Simulation and numerical results*: These first results allow showing a preview about the feasibility, the smartness and the performances of the designed approach. When the results lead to infeasibility, an in-depth analysis must be done. This analysis allows finding the step(s) at which modifications are required. If the AGV smartness or the performances are not good enough, the functions both from the AGV and from other entities must be modified or improved.
- *Feasibility tests of navigation algorithms*: In this step, the navigation algorithms are experimentally tested to check if the trajectories, provided by the navigation tools, are feasible. If not, the AGV navigation algorithms must be improved by reconsidering the two previous steps.

- *Embeddability tests of AGV functions*: After coding and embedding all the AGV functions, the embeddability is checked. An experimental scenario must be designed to test the overall functions by letting the AGVs do their mission. If they do not have the capacity to do it, the smart functions of AGVs must be redesigned.
- *Production tests on benchmark systems*: When embeddability is checked, the AGV functions can be tested on higher instances. A full production scenario (e.g. benchmark [18]) is designed. However, some functions could be missing to cover other issues such as global optimization or energy. Thus, these functions need to be included in the AGVs or in the manufacturing high-level of control and previous steps must be tackled again.
- *AGV deployment in industry*: To carry out this step, the different issues related to the industrial environment must be tackled. It can lead to reconsider some assumptions, made to simplify the designed approach. Other systems, such as supervisor, have to be designed to take human factor into account. For example, user-guided interface can be used, allowing human workers to take control of the manufacturing plant or to understand the AGV behavior. Moreover, the cost must be evaluated, leading to reconsider previous steps to reduce the AGV deployment cost.

In order to reach a good tradeoff, the idea is not to follow the procedure step by step but rather to follow a design for tradeoff procedure, thinking about the AGV capability and the cost required to obtain these capabilities. Indeed, using the balance paradigm (i.e. Fig 1), some weights are positioned on each side of the balance. On one side, the weights correspond to the AGV functions, depending on their complexity. On the other side, the weights are related to the AGV computational capabilities where for example, one weight may correspond to one processor core. Heavier weight gives higher AGV capabilities but their cost are also higher. Therefore, to take the design for tradeoff aspect into account, it should be suitable to deal with the AGV capabilities and their consequence it terms of cost before starting designing their smart functions. After defining these capabilities, the AGV functions could be design until the tradeoff is respected. It can lead to reduce the function smartness or to remove some functions in AGVs. Hence, cooperation mechanisms [6] with other entities of the flexible manufacturing system become useful to outweigh the AGV smartness according to their capabilities.

## 4 Illustrative example

To illustrate the proposed tradeoff–oriented procedure, an example is proposed in this section where AGVs navigate on a manufacturing production floor. This illustrative example is tested by giving some simulation and experimental results. The steps of the procedure, in Fig. 2, are roughly described and the tradeoff is discussed according to the provided AGV capabilities.

**AGV design steps:** In the proposed approach, the production objective is to complete several products, transported by AGVs. Each AGV transports only one product from a resource to another one until the product is completed. For each product, transported by one AGV, an operational sequence must be followed where operation are tackled one by one. For each operation, some production specifications are provided to the AGV by the manufacturing operation and resource management (M.O.R.M.) level. These specifications include a due date for which the operation must be completed and a set of resources at which the operation may be performed. Moreover, the resources have queuing capacities, assumed infinite at first, and AGVs in queue must wait to perform their operation when resource is unavailable (e.g. performing an operation of other AGV). The resource disturbances and the energy issues are not tackled to make the problem simpler.

Since the flexible manufacturing system layout, the production objective and the different specifications are given, the AGV design steps (i.e. left side of the procedure provided in Fig. 2) can be described. The AGV objective is both to compute the best resource and to generate a collision–free trajectory towards this resource. The best resource is chosen by minimizing the time to complete the ongoing operation and depends on the transportation time, the waiting time and the processing time of the resource. Minimizing the completion time is equivalent to complete the operation at the soonest date. In terms of assumptions, all AGVs have the same physical behaviors where their velocity are bounded and they are supposed to know their position at any time as well as the position of the resource they are moving towards. They have a limited communication range and are called neighbors when they are able to communicate. Moreover, they are able to communicate with the higher level of control, to receive help or specifications and transmit feedbacks about their configuration.

The different functions assigned to the AGV are mainly focused on their navigation. The motion planning function, combined with the scheduling one, allows selecting a resource at which the operation will be performed while generating a collision–free trajectory towards this resource. The tracking strategy function is in charge of following the planned trajectory in spite of inherent discrepancies and disturbances. The control architecture, where all AGV functions are highlighted, is given in Fig. 3. The AGV may interact with a supervisor which allows helping their navigation by solving the conflicts using performance–based priority negotiation. Moreover, it allows preventing their myopic behaviors since the AGVs only communicate with their neighbors. The supervisor has the role to solve the different conflicts. At first, it defines the arrival order of AGVs if they are moving towards the same resource. Furthermore, the AGV applies the scheduling function only if the supervisor allows it. The last supervisor function allows AGV to anticipate the collision by providing them some variable areas where big conflict may occur. It allows AGV to avoid these areas before reaching them, reducing the complexity of the AGV motion planner since big conflict areas are prevented.

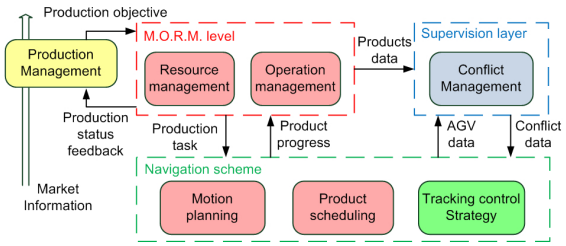



**Fig. 3** Control and decision architecture of the illustrative example

For the detailed design of AGV functions, the motion planner uses physical constraint (e.g. velocity bound) and temporal constraint (e.g. due date) to compute a collision–free trajectory towards a selecting resource. This planner is applied gradually over time and is divided into two steps. The first step, where the resource is chosen, is used as a global planner where AGVs must decide their intention by planning a presumed trajectory to avoid the conflicts given by the supervisor. The second step is local since each AGV uses the presumed trajectories, both its own and its neighbors' ones, to compute its final collision–free trajectory. One can notice that each AGV only avoids its neighbors having higher priority. It means that the AGV with the highest priority does not need to avoid others.

In terms of algorithm design, mathematical programming is used for each part (global and local) of the motion planner since it is related to the manufacturing performances. For embeddability purpose, the solving algorithms have to provide the trajectories in short time since they are computed over time. To reduce the computational costs, a Particle Swarm Optimization (PSO) is used due to its relatively fast convergence and global search character [15].

**AGV test steps:** Since the design steps have been given, the approach has to be tested to follow the procedure given in Fig. 2. To provide the simulation and numerical results, the AGV physical constraints are required. The AGVs are here represented by Lego Mindstorms robots where their maximum allowed velocity is $0.15m/s$. Their communication range and safety distance are respectively set to $0.4m$ and $0.2m$. The number of robots (called $AGV\ i$) is limited to 4 (i.e. $i = 1, 2, 3, 4$) and they navigate in a reduced area where three virtual resources (called $R_c$, $c = 1, 2, 3$) are positioned (as shown in Fig. 4). The positions $(x_c, y_c)$ of resources $R_c$ are set to $(1.85, 0.2)$, $(1.9, 0.9)$ and $(1.85, 0.65)$, respectively. For this scenario, the several AGV initial parameters can be found in Tab. 1. When the scenario starts, AGV 3 has the highest priority while AGV 4 has the lowest one.

For the simulation results of the proposed scenario, the AGV planned trajectories are provided in Fig. 4. One can see that AGV 3 does not avoid any other AGVs due to its highest priority. Conversely, AGVs 1 and 4 must adapt their trajectories to avoid others and AGV 2 needs to avoid AGV 3.

**Table 1** Agent initial parameters of the proposed scenario

| | Initial position | Operation due date | Processing time | Resource possibilities |
|---|---|---|---|---|
| AGV 1 | $[0.1, 0.125]$ | $76.1s$ | $5.4s$ | $\{R_3\}$ |
| AGV 2 | $[0.1, 0.625]$ | $30.8s$ | $4.3s$ | $\{R_1\}$ |
| AGV 3 | $[0.1, 1.125]$ | $23.8s$ | $5s$ | $\{R_2\}$ |
| AGV 4 | $[0.1, 1.625]$ | 60.2 | $4.7s$ | $\{R_2, R_3\}$ |

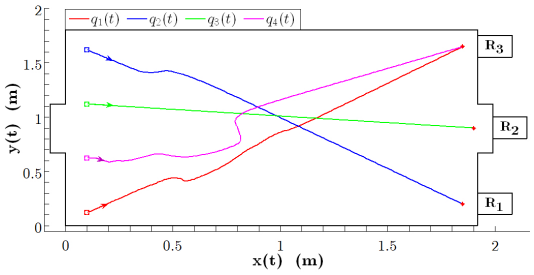



**Fig. 4** Simulation of the illustrative example: AGV planned trajectories

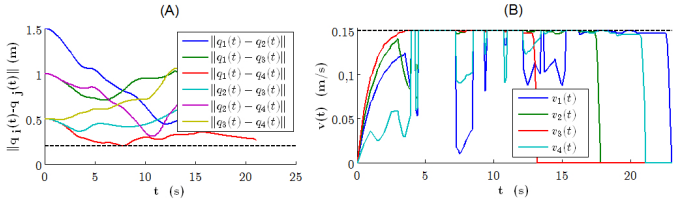



**Fig. 5** Simulation of the illustrative example: AGV distances (A) and velocities (B)

The approach is feasible in simulation since the velocity and collision avoidance constraints are fulfilled according to Fig. 5. Moreover, Table 2 shows that the operations of products transported by AGVs are completed before the provided due dates. During the navigation, one can see that AGV 4 changes its chosen resource. It implies that AGV 1 has lower priority than AGV 4. Therefore, the AGV smartness is proved since they are able to apply their function in spite of their specifications they have to fulfill. Indeed, collision-free trajectories are planned by the AGVs and the scheduling function is applied (*AGV* 4) to improve its product performances. Moreover, the supervisor allows them to prevent unsolvable conflict arisen from myopic behaviors.

The next step of the procedure is focused on the feasibility tests of navigation functions. Thus, the trajectories are tested by the Lego Mindstorms robots for this scenario. The feasibility of the trajectories is demonstrated experimentally as shown in the video[1] where all explanations are provided.

[1] Available at: https://www.youtube.com/watch?v=KUY7cBRx9vI&feature=youtu.be

**Table 2** Numerical results of the proposed scenario

| | AGV 1 | AGV 2 | AGV 3 | AGV 4 |
|---|---|---|---|---|
| Initial chosen resource | $R_3$ | $R_1$ | $R_2$ | $R_2$ |
| Final time | $25.44s$ | $17.76s$ | $13.14s$ | $21.02s$ |
| Completion time | $31.12s$ | $22.07s$ | $18.14s$ | $25.72s$ |
| Chosen resource | $R_3$ | $R_1$ | $R_2$ | $R_3$ |

To test the embeddability of AGVs, it is mandatory to consider the provided robots capabilities. These robots do not have the same capabilities as industrial AGVs, which are able to cover different functions [12]. Thus, all of designed functions cannot be embedded in AGVs to prevent bad tradeoffs as shown Fig. 6–(A) and (B). To reach a good tradeoff, the AGV functions must be reduced to the tracking one (see Fig. 6–(C)). It means that the combined motion planning/scheduling functions are computed by another entity, such as the supervisor, and then transmitting to the AGVs.

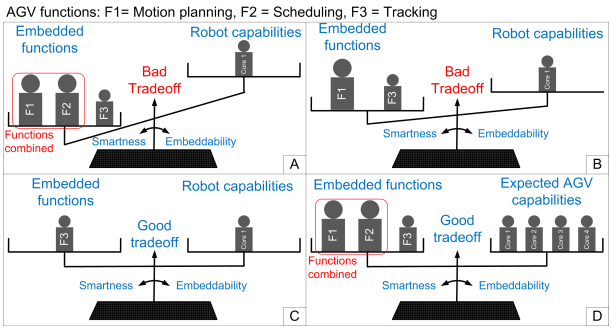



**Fig. 6** Tradeoff balance for the proposed approach: (A) all AGV functions, (B) scheduling function removed, (C) tracking function only (D) all functions with better capabilities

Using the Lego Mindstorms robots, the production tests are hard to obtain since they do not have to capabilities to plan their own trajectory. Indeed, planning a trajectory gradually over time for each AGV is computationally expensive, preventing to deal with manufacturing uncertainties such as machine breakdown. To perform these production tests, reconsidering the used AGVs seems to be more appropriate. Therefore, the AGV capabilities have to be improved to use their functions at full potential while maintaining a good tradeoff. Thus, the weights of the tradeoff balance changes, as shown in Fig. 6–(D), by proposing some expectations about the required AGV capabilities.

In order to deal with our proposed approach, the AGVs must be able to compute their own functions simultaneously. Thus, improving their capabilities is required by using multi-core processors for example. Using these processors, the computational time could be reduced and other tools, such as parallel meta–heuristic [1], may be proposed to solve our motion planning

problem. Indeed, since a resource must be selected, a trajectory for each resource could be computed in parallel. Moreover, the AGV capabilities must consider other simple functions such as the safety or energy ones which are not tackled in the proposed approach. Therefore, using multi-core processors is more suitable where each core can be represented as a weight in the embeddability side of the balance (i.e. Fig. 1). Nowadays, quad–core processors are easily embedded (e.g. in smart phones) and become less expensive than the last decades. Thus, their use as AGV device may be economically profitable.

## 5 Conclusion

In this paper, a tradeoff between smartness and embeddability is discussed for the use of AGVs in manufacturing plants. Several issues, which may be solved using AGV functions and have a possible influence on this tradeoff, are presented. The objective allows making the designers of AGV–based flexible manufacturing systems more sensitive about this tradeoff and its consequences. A tradeoff–oriented procedure is advisable to balance the smartness according to the AGV functions and its capabilities. An illustrative example on an AGV-based flexible manufacturing system is proposed where simulation and feasibility tests are provided to highlight the importance of this tradeoff for the AGV deployment in industry. To quantify and ease the analysis of this tradeoff, depending on company culture and methods, the designer could use some methods from the quality domain, like Value Analysis Method, Kano Model or Quality Function Deployment (QFD). Consequently, this first work leads to many perspectives from i) methodological view: how to choose efficient AGVs depending of requirements and how adapt them to deals with organization or requirements changes and ii) domain applications view: the introduction of AGVs on manufacturing processes opens new ways to organize the shop floor. It could be interesting, for example, to evaluate which physical element associate to AGV: the product or the resource?

## References

[1] Alba, E.: Parallel metaheuristics: a new class of algorithms, vol. 47. John Wiley & Sons (2005)
[2] Bader, M., Richtsfeld, A., Suchi, M., Todoran, G., Holl, W., Vincze, M.: Balancing centralised control with vehicle autonomy in agv systems for industrial acceptance. In: 11th Int. Conf. on Aut. and Aut. Sys. (2015)
[3] Berger, T., Sallez, Y., Trentesaux, D.: Open control of FMS and its application to potential field. In: 42nd Conf. on Man. Syst. (2009)
[4] Canny, J.: The complexity of robot motion planning. MIT press (1988)

[5] Defoort, M., Veluvolu, K.C.: A motion planning framework with connectivity management for multiple cooperative robots. Journal of Int. & Rob. Sys. **75**(2), 343–357 (2014)
[6] Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., Djemaï, M.: Cooperation mechanisms in multi-agent robotic systems and their use in distributed manufacturing control: Issues and literature review. In: IECON 2014-40th Conf. of the IEEE, pp. 2538–2543. IEEE (2014)
[7] Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., Djemaï, M.: Navigation scheme with priority–based scheduling of agents: Application to AGV-based FMS. Journal of Int. & Rob. Syst. pp. 1–18 (2015)
[8] Erol, R., Sahin, C., Baykasoglu, A., Kaplanoglu, V.: A multi-agent based approach to dynamic scheduling of machines and AGVs in manufacturing systems. App. Soft Comp. **12**(6), 1720–1732 (2012)
[9] Gilmour, K.: Amazon warehouse, amazon adventure. Int. Mag. (2003)
[10] Kuo, T.C., Huang, S.H., Zhang, H.C.: Design for manufacture and design for 'x': concepts, applications, and perspectives. Comp. & Ind. Eng. **41**(3), 241–260 (2001)
[11] Malinowski, A., Yu, H.: Comparison of embedded system design for industrial applications. IEEE Trans. on Ind. Inf. **7**(2), 244–254 (2011)
[12] Martínez-Barberá, H., Herrero-Pérez, D.: Autonomous navigation of an automated guided vehicle in industrial environments. Rob. and Comp.-Int. Man. **26**(4), 296–311 (2010)
[13] Masehian, E., Sedighizadeh, D.: Classic and heuristic approaches in robot motion planning-a chronological review. Wld. Acad. of Sc., Eng. and Tech. **23**, 101–106 (2007)
[14] Masehian, E., Sedighizadeh, D.: A multi-objective PSO-based algorithm for robot path planning. In: IEEE Int. Conf. on Ind. Tech. (ICIT), pp. 465–470. IEEE (2010)
[15] Saska, M., Macaš, M., Přeučil, L., Lhotska, L.: Robot path planning using particle swarm optimization of ferguson splines. In: IEEE Conf. on Emerg. Tech. and Fact. Aut., pp. 833–839. IEEE (2006)
[16] Thyagarajan, R.: A motion control algorithm for steering an agv in an outdoor environment. Ph.D. thesis, College of Engineering, Osmania University, India (2000)
[17] Trentesaux, D.: Distributed control of production systems. Eng. Appl.of Art. Int. **22**(7), 971–978 (2009)
[18] Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., Barbosa, J.: Benchmarking flexible job-shop scheduling and control systems. Cont. Eng. Pract. **21**(9), 1204–1225 (2013)
[19] Ueda, K.: Emergent synthesis approaches to biological manufacturing systems. In: Digital Enterprise Technology, pp. 25–34. Springer (2007)
[20] Wurman, P.R., D'Andrea, R., Mountz, M.: Coordinating hundreds of autonomous vehicles in warehouses. AI magazine **29**(1), 9 (2008)