



# Comparative analysis of pick & place strategies for a multi-robot application

Gaël Humbert, Minh Tu Pham, Xavier Brun, Mady Guillemot, Didier Noterman

## ► To cite this version:

Gaël Humbert, Minh Tu Pham, Xavier Brun, Mady Guillemot, Didier Noterman. Comparative analysis of pick & place strategies for a multi-robot application. 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2015), Sep 2015, Luxembourg, Luxembourg. 10.1109/ETFA.2015.7301450 . hal-01225223

**HAL Id: hal-01225223**

**<https://hal.science/hal-01225223>**

Submitted on 2 May 2019

**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.

# Comparative analysis of pick & place strategies for a multi-robot application

G. Humbert, M.T. Pham, X. Brun, M. Guillemot and D. Noterman

**Abstract**—This paper deals with a comparative analysis of different pick & place strategies. The purpose is to give some rules to obtain a good sizing in terms of components (number of robots, conveyor speed) and control laws (individual scheduling rules of each robot, collaborative strategy of all the robots) of a multi-robot cell. This approach is validated by the use of a new simulation tool combining a behavioral simulation of multiple robots and the product flows. This simulation tool takes into account not only the end effector, but also the robots collaborative aspect to ensure the desired overall performance for a given task.

## I. INTRODUCTION

In recent years, the customers demand of productivity and flexibility for their production lines has largely increased. This is why robots and robotic pick & place cells are more and more present in some industrial fields such as the food industry. In high-performance applications, typical characteristics of a pick & place robot can reach the following values : velocity 10 m/s, acceleration 100 m/s<sup>2</sup> precision +/- 0.1mm, pick & place cycle 0.40s on average. To improve the performance of these applications it is necessary to improve the design of current production systems (number of robots, performances, etc.) whilst also improving the management of flows and workload management when several robots are used.

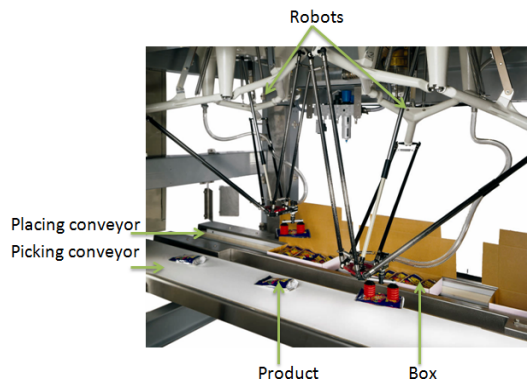


Fig. 1. Robotic cell with delta robots.

A pick & place application is usually composed of a series of several robots installed in a line one after the other taking products on a first conveyor and placing them in boxes located on a second conveyor, see figure 1 [23] [21].

On a multi-robot packaging cell, when there is no workflow optimization system, pick instructions are divided equally between the first robots. A final robot is added to try to recover the products that could not be taken by the previous robots. Products initially assigned to a robot may not be taken because they finally are out of the robot workspace because of a lack of boxes to fill, for example. This kind of flow management approach is widely used in industrial applications and has three disadvantages:

- 1) The first robots have an important individual workload close to 85%, which can lead to availability issues.
- 2) Contrarily, the last robot has an individual workload close to 25%.
- 3) A large number of products can not be picked and they are discarded. The average percentage product loss could reach 1% in steady state and 5% in the transient state.

In industrial applications, field experiences are commonly used for production line sizing. To the best of our knowledge, there are not efficient tools that can help sizing either the performance analysis or the workflow optimization for several reasons:

- The problem is complex: many parameters have to be taken into account (robots, conveyors, products, etc.).
- Different skills must be used: robotics and flow management.
- There are no tools dedicated to both simulation and implementation. Some tools such as online programming exist but they do not support high level programming. This kind of feature could be interesting for pick & place applications, product flow simulation and collaborative aspects between several robots.

However simulation approaches are beginning to be developed by some companies, but the performance of the software tools that they propose are difficult to assess:

- Staübli: LineManager software dynamically manages the workload [25].
- ABB: RobotStudio and PickMaster 3 software with the add-on Picking PowerPac provides a configurable environment to test different application [1]. Stumm *et al.* [26] have used this software to show a realistic representation of product flow and the arising dynamic

G. Humbert, M.T. Pham and X. Brun are with Université de Lyon, INSA-Lyon, Laboratoire Ampère, 20 Avenue Albert Einstein, Villeurbanne 69621, France (gael.humbert,minh-tu.pham,xavier.brun)@insa-lyon.fr

M. Guillemot is with INSA-Lyon, 8 rue de la physique, Villeurbanne 69621, France mady.guillemot@insa-lyon.fr

D. Noterman is with INSA-Lyon, Laboratoire DISP, 8 rue de la physique, Villeurbanne 69621, France didier.noterman@insa-lyon.fr

changes within product flow in pick & place applications.

- Keba: Real World Simulation Software Package software can test pick & place strategies [14].
- Bosch Rexroth: IndraWorks software dynamically manages the workload [14].
- Fanuc: Genkotsu Learning Control software improves the speed and the performances of their robots with learning [7].

To the best of our knowledge, in industrial and academic context, there are no simulation tools that take into account the four following aspects: a behavioral simulation of the robots, a simulation of the work environment (product flow, boxes flow), the collaborative work of several robots and finally the possibility to go from simulation to experimentation (the code developed for during the simulation tool need to be automatically generated for the industrial controllers of the robots).

The first contribution of this paper is to show the impact of the presence or lack of collaborative strategies between the robots on an oversized system and on an accurate sized system.

The second contribution is to show the impact of the individual scheduling rules assigned to each robot.

The third contribution of this paper is the development of a software interface that represents the robotic cell in a realistic 3D environment. The developed software is able to simulate realistic product and box flows, generate the trajectory of the end effector to reduce the picking time and the placing time, propose several collaboration strategies between robots. This software is also modular and allows the simulation of different configurations for a same application based on several parameters (number of robots, environment design, placement of the robots, etc.).

Section II presents the typical sizing that is done in some industrial applications. Pick & place strategies are shown and a new simulation tool dedicated to pick & place application is presented. Section III shows a comparative study of different pick & place strategies according to two sizing assumptions. Finally, a conclusion and future works are presented.

## II. SIZING AND SIMULATION OF PICK & PLACE APPLICATIONS

### A. SIZING OF INDUSTRY APPLICATION

As indicated in section I, the sizing of pick & place robotic systems is usually done empirically. However, this empirical approach follows a decision sequence, which can be summed up in the following:

- 1) An input product flow is chosen by the customer, in terms of products per second  $FPs$ .
- 2) The customer generally knows the average time for a robot to do a pick & place task. This information gives an estimation of the product cadence per second  $PPas$ .
- 3) The system is oversized using two possibilities:

- Either a margin of safety  $Mfp$  is added to the products flows (10-15%).
- Or another robot is added to the minimum number of robots.

- 4) The minimum of robot  $NRmin$  is calculated:  $NRmin = FPs * (1 + Mfp) / PPas$  rounded up to the nearest integer.
- 5) If the margin on the flow is chosen, the final number of robot  $NRf = NRmin$  otherwise  $NRf = NRmin + X$  where  $X$  can be 1 or 2 (depending on the application size).
- 6) The output box flow  $FBs$  is calculated based on the number of places  $NPb$  in one box:  $FBs = FPs / NPb$ .
- 7) The input  $Vi$  and output  $Vo$  speeds are either fixed by the customer or fixed arbitrarily, however, they need to depend on the flow and the spacing between the middle of the products  $EP$  and the boxes  $EB$ :  $FPs = Vi / EP$  and  $FBs = Vo / EB$ .
- 8) There is no collaborative strategy between the robots, the individual scheduling rules are the same for all the robots and are "all-you-can-pick".

This sizing works but is not optimal. The first robots work more than the last robots, which can cause premature wear.

### B. PICK & PLACE STRATEGIES

When a single robot is used, a queue or a basic sort in one direction is sufficient. Matton *et al.* [15] have proposed interesting online scheduling rules based on a queue. Erlandsson-Warvelin *et al.* [6] have filed a patent about control methods for machines which used a sort method in one direction.

If several robots are used, more complex algorithms than a queue are necessary. The aim is to size the number of robots with respect to the number of products, boxes and conveyors. To do this, optimization algorithms could be used. Research works related to optimization algorithms used in robotic applications are numerous. Huang *et al.* [11] have utilized a greedy randomized adaptive search procedure to search for the optimal scheduling rules. Slim *et al.* [24] have compared three metaheuristic methods: ant colony optimization, genetic algorithm and particle swarm optimization. The aim was to maximize the number of picking products and to take also into account the execution time. Bonert *et al.* [2] have searched a solution of the Travelling Salesperson Problem with a genetic algorithm. Fujimoto *et al.* [8] have used a genetic algorithm to seek the best combination of scheduling rules in order to obtain an appropriate flow. Zhu *et al.* [32] have studied a differential evolution algorithm for the pick & place problem of multi-head surface mounting machines. Mendelson *et al.* [17] have developed a decentralized palletizing multi-robot system using fuzzy logic. Edan *et al.* [5] have developed a fuzzy method and a simple "all-you-can-pick" algorithm. Isil Bozma *et al.* [3] have proposed an approach based on non-cooperative game theory to decide the robots actions in a multi-robots pick & place application. In all of

these works, only simulation is used and the translation from simulation to experimentation is unfortunately not tackled.

Usually the optimization techniques above are computationally intensive and are not appropriate to real time application where the computation cycle required is about a millisecond. Another alternative is to use this kind of off-line algorithms in simulation in order to schedule the best combination of rule then apply the result in practice.

Petri nets have been used for pick and place system modeling. The aim of these works is mainly to synchronize two robots relative each other. Yasuda [29], [30] has developed a Petri net for a production system where two robots can exchange pieces. Zhou *et al.* [31] have proposed a Petri net which allows the work of two robot arms inside the same workspace. However in our application, the robots are independent and their workspaces are not shared.

In the literature, a few patents are related to pick & place strategies. Izumi *et al.* [12] have filed a patent about conveyors sharing in order to balance the robots workload. Herzog [9] has filed a patent about a method of filling containers using the discretization of the system. He used quadratic programming to solve the problem. Wappling *et al.* [28] have filed a patent about a scheduling method for pick & place applications.

### C. A NEW SIMULATION TOOL DEDICATED TO PICK & PLACE APPLICATIONS

There are several needs for this study. Firstly, the robots behavioral have to be simulate and the end effector has to be visualize to check if the end effector moves correctly and stops in the right position. Secondly, the work environment (product flow, boxes flow) has to be simulated in order to check the interaction between the robots and the environment and to verify the robot pick the right product. Thirdly, two level of strategies (collaborative strategies and individual scheduling rules) has to be taken into account. And fourthly, the possibility to go from simulation to experimentation have to be taken into account in order to have an implementation in-situ the easiest and fastest possible.

In the literature, there are several works dedicated to robotic pick & place simulation, which are only used for visualization, to verify the kinematics and dynamics. They are also used for robot design to validate its behavior, its movements and its interaction with the environment (collision detection). Johari *et al.* [13] have used Workspace5 to visualize an entire robotic application system in order to detect collisions between robots and their environment. Sam *et al.* [22] have designed a pick & place robotic system using Solidworks Softmotion software to study the motion of an articulated robot.

To improve the productivity of a pick & place multi-robots application, the flow management has to be improved. There are several programs that are able to simulate this. Meiyun *et al.* [16] have used Witness in order to find bottlenecks in a production system. Mirzapourrezaei *et al.* [18] have also used Witness to evaluate various aspect of manufacturing

systems. The objective was to enhance the productivity and efficiency of the line. Pegden *et al.* [20] and Sturrock *et al.* [27] have presented Simio, a software based on intelligent objects. Concannon *et al.* [4] have developed Simul8, a software which uses techniques of production planning and scheduling. Hindle *et al.* [10] have used Simul8 to answer the complex scheduling problem of sequencing part requirements through a composites manufacturing center. Nikakhtar *et al.* [19] have compared two simulation tools: Arena and Witness. However, all of these programs are mainly dedicated to flow simulation. Visualizations are very basic, and mainly focused on the flows. The displacement of products or boxes on conveyors can be visualized with Simio 3D or Witness but it can not have several products on the width of a conveyor and just the execution time of a robot is simulated and not the movement of the effector.

Unlike other works, the environment software developed in our work allows the creation of a virtual machine, in 3D and in real time, based on the CAD model of the real system. This simulated robotic cell has the same kinematics and the same dynamics as the real process, its environments can also be simulated: products arriving on a conveyor etc. Scenarios can be implemented to verify its behavior. A high-level layer can be used to implement a products pick & place strategy and a collaboration strategy between several robots if necessary. This software is also modular, it can configure the production system (robot, conveyor, etc.), its environment (products, boxes, etc.) and the different scenarios.

The tool developed incorporates two levels of strategies shown in Figure 2. Simple individual scheduling rules for a single robot can be:

- FIFO: First In First Out. The robot picks the first product arrived in its workspace.
- LIFO: Last In First Out. The robot picks the last product arrived in its workspace.
- SPT: Shortest Processing Time. The robot picks the nearest products of its end effector.

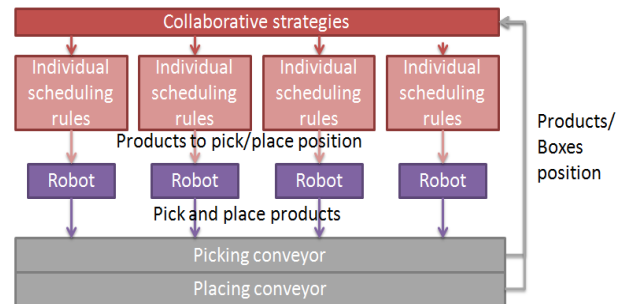


Fig. 2. Simulation architecture with the two levels of strategies (individual scheduling rules and collaborative strategies).

Collaborative strategies can also be used and the products can be assigned to the robots before they arrive in the different workspaces. An example of different collaborative strategies between four robots is given in figure 3. The red

products are assigned to the robot 1, the green products are assigned to the robot 2, the blue products are assigned to the robot 3 and the yellow products are assigned to the robot 4. If there is an assignment by clusters (Figures 3.c and 3.d), the last ones are sequentially treated. The lightest orange cluster is treated in the first row and the darkest orange cluster is treated at the end.

- Right / Left: Assign to the robots the products one by one from the right to the left (Figure 3.a).
- Up / Down: Assign to the robots the products one by one from up to down (Figure 3.b).
- Up / Down Cyclic: Assign to the robots the products one by one from up to down for a number of products equal to the number of robots. And then the order of robots is cyclically changed (Figure 3.c).
- Up / Down Alternate: Assign to the robots the products one by one from up to down for a number of products equal to the number of robots. Then the assignment is down to up and the assignment is alternate. (Figure 3.d).

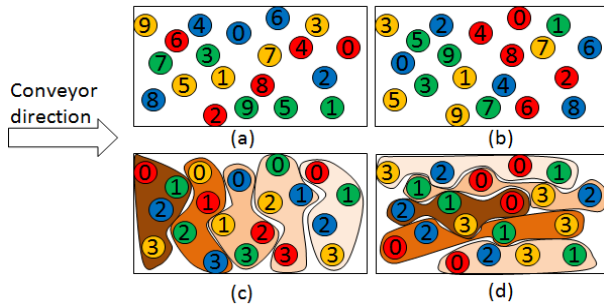


Fig. 3. Example of different collaborative strategies

The simulator also includes the management of field bus and a real-time kernel. The user can verify the behaviour of different pick & place strategies on his/her host computer with a refined simulation where the behaviour of a real industrial controller is included. The idea is to run the simulations either on an instruction set simulator, with a behaviour close to the real production system, or on a hardware target.

The pick & place application creation consists of several steps, first the definition of the simulation model is carried out with a 3D software. At this stage the graphic objects and the kinematic behaviors of the application objects are defined (robots, conveyors, etc.) in C++. The second step is the development of collaborative strategies between robots in simulation in C++. Finally, simulations could be run with the software, see figure 4, to test the model behavior and analyze the results of the different pick & place strategies. Once the parameters of simulation are optimized, experimental tests can be done to check the algorithms operation and test their performance.

### III. COMPARATIVE STUDY

A comparative analysis has been carried out using the new software tool presented section II-C. This study compares



Fig. 4. Simulation example with four robots and two conveyors co-current with the Right/Left collaborative strategy.

different collaborative strategies between robots and various individual scheduling rules for each robot presented section II-C: Left/Right, Up/Down, Up/Down Cyclic and Up/Down Alternate collaborative strategies and FIFO, LIFO and SPT individuals scheduling rules. Firstly this study was conducted for two sizing assumptions: an oversizing of the system similar to those that can be found in industrial applications, and an accurate sizing of the system, where the number of robots and the parameters (speeds of the conveyors) are chosen to obtain the desired product flows. In both cases, the sizing is performed such that there are not any lost products and unfilled boxes. Undersized systems are not studied in the following analysis. During the simulation with the oversizing assumptions, a study was carried out to compare the impact of a placing priority rule. This rule stops the output conveyor if an unfilled box reaches the second half of the workspace of the last robot.

Different parameters are analyzed and compared such as the percentage of lost products, unfilled boxes and the balance of the workload of the robots and the total workload of the system. The workload is defined by the following equation (1):

$$Workload = \frac{T_{Pick} + T_{Place}}{T_{Pick} + T_{Place} + T_{Wait}} \quad (1)$$

where  $T_{Pick}$ ,  $T_{Place}$  and  $T_{Wait}$  are respectively the picking, placing and waiting time in seconds. The balance of the workload is defined as the difference of the workload of the robots and calculated with the equation (2):

$$BW = \sum_{k=1}^{NR_f-1} |W_k - W_{k-1}| \quad (2)$$

where  $W_k$  is the workload of the  $k_{th}$  robot. The more  $BW$  is close to 0 the more the system is balanced. The total workload of the system is defined by the sum of the workload of all the robots and calculated with the equation (3):

$$TW = \sum_{k=1}^{NR_f} W_k \quad (3)$$



where  $W_k$  is the workload of the  $k_{th}$  robot. The index  $TW$  gives an idea of the operating rate of the entire system. The higher this index is the more the robots work.

Secondly for an accurate sized system, the influence of the speed of the input and output conveyors is studied for the same input and output flow.

In the following, general assumptions on the pick & place application are considered:

- All the robots are identical.
- A pick & place cycle for one robot takes an average 0.383 sec. The robot can pick  $PPas = 1/0.383 = 2.61$  products per second.
- The robots work at a speed of 10 m/s and an acceleration of 100 m/s<sup>2</sup> and pick & place one product at the same time.
- The conveyors are parallel with co-current flows.
- Products in the input flow are positioned randomly.
- The boxes measure 0.1 m \* 0.1 m and contain 9 products ( $NPb = 9$ ).

#### A. OVERSIZED SYSTEM

Additional assumptions on the pick & place application in this section:

- The input conveyor speed is chosen arbitrarily equal to 0.2 m/s ( $Vi = 0.2$ ).
- The distance between two boxes is arbitrarily set to 0.05 m ( $EB = 0.1 + 0.05 = 0.15$ ).

According to the method described in section II.A, an input product flow is chosen  $FPs = 7.83$  products per second. To oversize the system, a margin of safety  $Mfp = 25\%$  is added. The minimum of robot is  $NRmin = FPs * (1 + Mfp)/PPas = 7.83 * (1 + 0.25)/2.61 = 3.75$  rounded to 4. Since a margin of safety was used, the final number of robot is  $NRf = NRmin = 4$ . Based on the number of products in a box, the output flow is equal to  $FBs = FPs/NPb = 7.83/9 = 0.87$  boxes per second. Using the distance between the boxes, the output conveyor speed needs to be equal to  $Vo = FBs * EB = 0.87 * 0.15 = 0.13$  m/s.

The simulation is performed with the following arbitrary parameters:

- Pick and place individual scheduling rule for all the robots: FIFO.
- Simulation time: 1h.

Tables I and II show the simulation results for different collaboration strategies applied on an oversized system. The table I shows the simulation not using the placing priority rule and the table II shows the simulation using the placing priority rule. First we notice that the placing priority rule allows to have all the boxes filled. Without, there are unfilled boxes. This is only due to the rounded errors in the calculations. In addition, the total workload is slightly lower with the placing priority rule. However, this rule unbalanced the workload between the robots.

In these two cases, the worst results are obtained without any collaborative strategy because there is a percentage of

lost products. In addition, there is a strong imbalance between the robots, which can lead to premature wear of the first robots. However, the total workload is better than the others (around 297%). With the others strategies the total workload is bigger (around 335%) but there is a better balance (around 20%). There are two kind of strategies, Left/Right and Up/Down cyclic strategies which have a better balance (around 12.8%) and Up/Down and Up/Down Alternate which have a better total workload (around 329%).

#### B. ACCURATE SIZED SYSTEM

In addition to the general assumptions, further assumptions on the pick & place application are made:

- The input conveyor speed is chosen arbitrarily equal to 0.2 m/s ( $Vi = 0.2$ ).
- The distance between two boxes is arbitrarily set to 0.05 m ( $EB = 0.1 + 0.05 = 0.15$ ). A placing priority rule is added. The output conveyor is stoped if an unfilled box reaches the second part of the workspace of the last robot.

According to the method described in section II.A, an input product flow is chosen  $FPs = 10.18$  products per second. To oversize the system, a margin of safety  $Mfp = 2.5\%$  is added. The minimum of robot is  $NRmin = FPs * (1 + Mfp)/PPas = 10.18 * (1 + 0.025)/2.61 = 4$ . Since a margin of safety was used, the final number of robot is  $NRf = NRmin = 4$ . Based on the number of products in a box, the output flow is equal to  $FBs = FPs/NPb = 10.18/9 = 1.13$  boxes per second. Using the distance between the boxes, the output conveyor speed needs to be equal to  $Vo = FBs * EB = 1.13 * 0.15 = 0.17$  m/s.

The simulation is performed with the following arbitrary parameters:

- Picking and placing individual rules: FIFO.
- Using of a placing priority rule.
- Simulation time: 1h.

Table III shows the simulation results where different collaboration strategies on an accurate sized system. The best results are obtained without any strategies and with the Up/Down Cyclic strategy. All the products are picked and all the boxes are filled thanks to the placing priority rule. However, without any strategies the flow can be increased slightly because the total workload is smaller than 400%. The other strategies are bad because some products are lost even though the number is low.

Compared to an oversized system this result is quite unexpected but this is due to the collaborative strategy. In fact putting a collaborative strategy adds a constraint to the pick & place system. Since there is no need of collaborative strategies between robots, a comparison of the four different individual scheduling rules of each robot is presented in the following. The possibilities are numerous ( $3^4 = 81$ ) and the simulation time of each of them takes 10 min.

Tables IV and V show the simulation results of the effect of the individual scheduling rules of each robots on an accurate

TABLE I  
RESULTS OF THE COLLABORATIVE STRATEGIES FOR AN OVERSIZED SYSTEM WITHOUT PLACING PRIORITY RULE.

	Lost products (%)	Unfilled boxes (%)	Balance workload (%)	Total Workload (%)
Without any strategy	0.0467	5.01	93.7	297
Left/Right	0	3.90	5.00	352
Up/Down	0	3.68	6.21	346
Up/Down Cyclic	0	4.20	4.58	351
Up/Down Alternate	0	3.91	6.36	346

TABLE II  
RESULTS OF THE COLLABORATIVE STRATEGIES FOR AN OVERSIZED SYSTEM WITH PLACING PRIORITY RULE.

	Lost products (%)	Unfilled boxes (%)	Balance workload (%)	Total Workload (%)
Without any strategy	4.48	0	96	288
Left/Right	0	0	12.8	344
Up/Down	0	0	28.8	329
Up/Down Cyclic	0	0	12.8	343
Up/Down Alternate	0	0	28.8	329

TABLE III  
RESULTS OF THE COLLABORATIVE STRATEGIES FOR AN ACCURATE SIZED SYSTEM WITH PRIORITY RULE.

	Lost products (%)	Unfilled boxes (%)	Balance workload (%)	Total Workload (%)
Without any strategy	0	0	5.92	385
Left/Right	0.225	0	1.09	390
Up/Down	0.721	0	0.04	391
Up/Down Cyclic	0	0	1.46	389
Up/Down Alternate	0.529	0	0.12	391

TABLE IV  
INDIVIDUAL SCHEDULING RULES WITH LESS THAN 1% OF LOSS

R1	R2	R3	R4	Products	Boxes
0	0	0	0	0	0
0	0	1	0	0.0236	0
0	0	2	0	0.1535	0
0	1	0	0	0	0
0	1	1	0	0.0472	0
0	1	2	0	0.2481	0
0	2	0	0	0.8677	0
1	0	0	0	0	0
1	0	1	0	0	0
1	0	2	0	0.3786	0
1	1	0	0	0	0
1	1	1	0	0.0118	0
1	2	0	0	0.9168	0

TABLE V  
SUMMARY OF THE BEST INDIVIDUAL SCHEDULING RULES.

	R1	R2	R3	R4
Lost products (%)	0	0	0	0
Unfilled boxes (%)	0	0	0	0
Balance workload (%)	4.08	1	1	1
Total workload (%)	384	1	2	0

sized system. For the sake of simplicity the rules are denoted: 0 for the FIFO rule, 1 to the LIFO rule and 2 for the SPT rule. Table IV gathers only the combinations that allow to obtain less than 1% of lost products and unfilled boxes. First, we note that the worst rule is the SPT rule in every combination. The lost rate is greater than 1%, or very close to this value. This can be explained by the fact that the pick and place time increases from the first robot to the last robot. It is noteworthy that the FIFO rule must be used by the last robot. When several products arrive in the workspace of the last robot, if the individual scheduling rule of this robot is LIFO or SPT rule, the last or the nearest product is picked while the other products will continue. If the penultimate robot uses a LIFO rule there are few losses. For the other robots, the choice between FIFO and LIFO rules do not have a significant impact on the results.

Table V shows the combinations that allow to have the best balance or total workload. The best combinations for balance and total workload causes lost products.

For an accurate sized system, the customer has to use the robot without a collaborative strategy to have no lost products and no unfilled boxes. This adds a constraint to the system. The latest robots have to use the FIFO individual scheduling rules. For the other robots, the individual scheduling FIFO or LIFO rules seem to be better. Since a collaborative strategy is not required when the system is well sized, the last aspect of our study concern the influence of the conveyors.

### C. IMPACT OF THE SPEED OF THE CONVEYORS

According to the method described in section II.A, an input product flow is chosen  $FPs = 10.44$  products per second. No margin  $Mfp$  is added. The final number of robot is  $NRf = FPs/PPas = 10.44/2.61 = 4$ . Based on the number of products in a box, the output flow is equal to  $FBs = FPs/NPb = 10.44/9 = 1.16$  boxes per second.

The simulation is performed with the following arbitrary parameters:

- Pick and place individual scheduling rules for all the robots: FIFO.
- Simulation time: 20 min for each configuration.

**Input conveyor speed:** During this simulation, the speed of the input conveyor varies between 0.1 and 0.8 m / s but with the same product flow. The spacing between boxes is arbitrarily equal to 0.5 m, which corresponds to an output conveyor speed of 0.170 m/s.

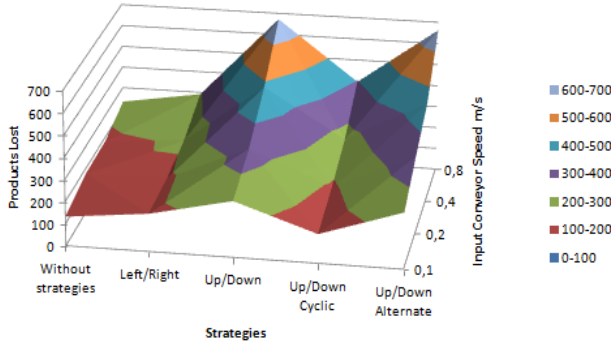


Fig. 5. Lost Products depending of the input conveyor speed.

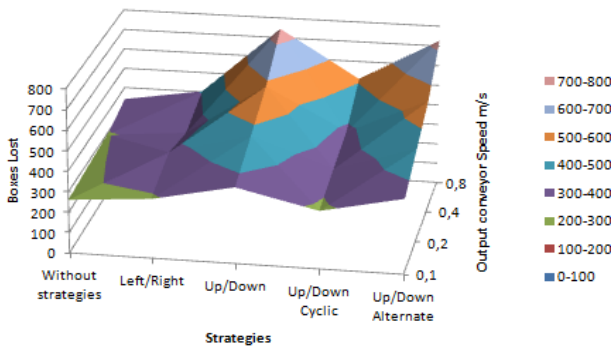


Fig. 6. Unfilled Boxes depending of the input conveyor speed.

Figure 5 and 6 show the impact of the input conveyor speeds on the number of lost products and unfilled boxes. It is noteworthy that more the input conveyor speed is high more products and boxes are lost. This happens for all the strategies.

**Output conveyor speed:** During this simulation, the speed of the output conveyor varies between 0.1169 m/s (which corresponds to a space between boxes equal to 0m)

and 0.4 m/s (space equal to 0.1566m) with the same boxes flow. The input conveyor speed is arbitrarily fixed to 0.2 m/s.

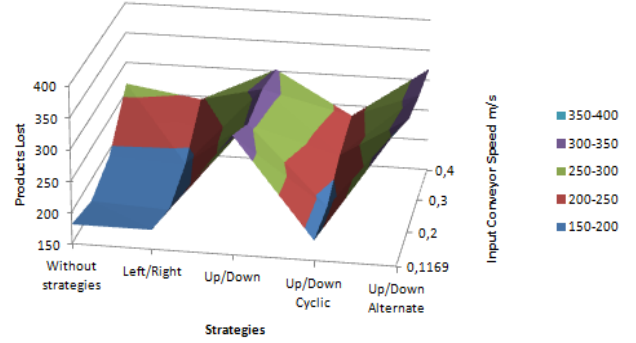


Fig. 7. Lost products depending of the output conveyor speed.

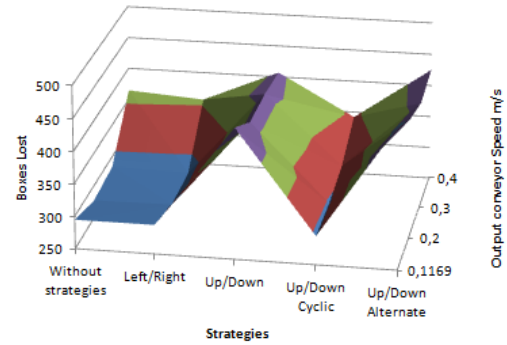


Fig. 8. Unfilled boxes depending of the output conveyor speed.

Figure 7 and 8 show the impact of the output conveyor speed on the number of lost products and unfilled boxes. It is noteworthy that the best strategies lose less products and boxes with a speed equal to 0.2m/s. Up/Down and Up/Down Alternate strategies lose less products and boxes when the speed increase but still higher than the other strategies.

To minimize the number of lost products and unfilled boxes, the input conveyor speed has to be reduced to its minimum. For this system, the output conveyor speed has to be close to 0.2 m/s (space equal to 0.14m).

## IV. CONCLUSION

In this paper, we presented a comparative study of different collaborative strategies between robots and various individual scheduling rules for each robot under two sizing assumptions: oversizing and accurate sizing. To validate our study a new simulation tool has been developed. According to the customer demand, a collaborative strategy between the robots and the individual scheduling rules of each robot will not be the same. For an oversized system, it is preferably to use a collaboration strategy between robots. This allows to pick all the products and fill all the boxes. This allows also to balance the workload on all the robots and thus to reduce the premature wear of a robot. The study shows that if the



customer wants an accurate sized system to be the most efficient, it is better to not use a collaborative strategy because it adds a constraint to the system. We also have shown that it is better to use an individual scheduling rule FIFO for the last robots and FIFO or LIFO rules for the others robots. If the customer wishes to improve workload balance, the use of the FIFO rule for the last robot and LIFO rule for the others robots is a good combination. However, some products will be lost. Finally, the last section of this paper shows it seems to be better to have an input and output conveyor speed the lowest possible.

Future work concerns the study and the analysis in simulation with different configurations (number of robots, product flow, simulation time) to reinforce the results. Another work is to improve the capability of the application to be autonomous in order to cope of an input products flow change for example. Another work is to validate these results on a benchmark composed of three robots and two independent conveyors.

## ACKNOWLEDGEMENT

This work was partially supported by Schneider Electric with the CIFRE 158/2013.

## REFERENCES

- [1] ABB. Abb robotstudio picking powerpac simplifies the design and programming of complex robotic pick-and-pack systems, April 2014. [http://www.robotics.org/content-detail.cfm/Industrial-Robotics-News/ABB-RobotStudio-Picking-PowerPac-simplifies-the-design-and-programming-of-complex-robotic-pick-and-pack-systems/content\\_id/4780](http://www.robotics.org/content-detail.cfm/Industrial-Robotics-News/ABB-RobotStudio-Picking-PowerPac-simplifies-the-design-and-programming-of-complex-robotic-pick-and-pack-systems/content_id/4780).
- [2] M. Bonert, L.H. Shu, and B. Benhabib. Motion planning for multi-robot assembly systems. *International Journal of Computer Integrated Manufacturing*, 13(4):301–310, Nov 2010.
- [3] H. I. Bozma and M.E. Kalalioglu. Multirobot coordination in pick-and-place tasks on a moving conveyor. *Robotics and Computer-Integrated Manufacturing*, 28:530–538, Aug 2012.
- [4] K.H. Concannon, K.I. Hunter, and J.M. Tremble. Simul8-planner simulation-based planning and scheduling. In *Proceedings of the 2003 Winter Simulation Conference, 2003.*, number 2, pages 1488 – 1493. IEEE, Dec 2003.
- [5] Y. Edan, S. Berman, E. Boteach, and E. Mendelson. Distributed multi-robot assembly-packaging algorithms. *Intelligent Automation and Soft Computing*, 10(4):349–357, Mar 2004.
- [6] Mats Erlandsson-Warvelin and Henrik Knoble. Control method for machines, including a system, computer program, data signal and gui, 01 2012. Patent EP 1 635 995 B1.
- [7] Fanuc. Fanuc america demonstrates learning capabilities with m-3ia/6s robot at pack expo 2013, 2013. <http://www.fanucamerica.com/fanucamerica-news/press-releases/pressreleasedetails.aspx?id=27>.
- [8] H. Fujimoto, I. Tanigawa, K. Yasuda, and K. Iwahashi. Applications of genetic algorithm and simulation to dispatching rule-based fms scheduling. In *Proceedings 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 190–195. IEEE, May 1995.
- [9] Florian Herzog. Method and apparatus for filling containers with piece goods, Feb 2003. Patent US 2003/0037515 A1.
- [10] K. Hindle and M. Duffin. Simul8-planner for composites manufacturing. In *Proceedings of the Winter Simulation Conference, 2006. WSC 06.*, pages 1779 – 1784. IEEE, Dec 2006.
- [11] Yanjiang Huang, R. Chiba, T. Arai, T. Ueyama, and J. Ota. Part dispatching rule-based multi-robot coordination in pick-and-place task. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1887 – 1892. IEEE, Dec 2012.
- [12] Tetsuro Izumi, Kenichi Koyanagi, Kenji Matsukuma, and Yukio Hashiguchi. Robot system, May 2013. Patent EP 2 586 576 A1.
- [13] N.A.M. Johari, H. Haron, and A.S.M. Jaya. Robotic modeling and simulation of palletizer robot using workspace5. In *4th International Conference on Computer Graphics, Imaging and Visualization (CGIV 2007), August 14-16, 2007, Bangkok, Thailand*, pages 217 – 222. IEEE, Aug 2007.
- [14] Keba. Real world simulation package, 2014. [http://www.keba.com/fileadmin/user\\_upload/downloads/kemotion/Datenblatt\\_KeMotion\\_Real\\_World\\_Simulation\\_Package\\_EN\\_RZ.pdf](http://www.keba.com/fileadmin/user_upload/downloads/kemotion/Datenblatt_KeMotion_Real_World_Simulation_Package_EN_RZ.pdf).
- [15] R. Mattone, L. Adduci, and A. Wolf. Online scheduling algorithms for improving performance of pick-and-place operations on a moving conveyor belt. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA-98, Leuven, Belgium, May 16-20, 1998*, volume 3, pages 2099 – 2105 vol.3. IEEE, May 1998.
- [16] Zhao Meiyun and Xu Heguo. Research on the production system of an enterprise simulation base on witness. In *International Conference on E-Business and E-Government (ICEE), 2011*, pages 1 – 4. IEEE, May 2011.
- [17] E. Mendelson, O. Nayer, S. Berman, and Y. Edan. Behavior-based control of multi-robot assembly palletizing systems. In *Proceedings of the 5th Biannual World Automation Congress, 2002*, volume 14, pages 1–6. IEEE, 2002.
- [18] S.A. Mirzapourzeaei, M. Lalmazloumian, A. Dargi, and Kuan Yew Wong. Simulation of a manufacturing assembly line based on witness. In *Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2011*, pages 132 – 137, July 2011.
- [19] A. Nikakhtar, Kuan Yew Wong, M.H. Zarei, and A. Memari. Comparison of two simulation software for modeling a construction process. In *Third International Conference on Computational Intelligence, Modelling and Simulation (CIMSIM), 2011*, pages 200 – 205. IEEE, Sept 2011.
- [20] C.D. Pegden and D.T. Sturrock. Introduction to simio. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 314 – 321. IEEE, Dec 2009.
- [21] Hakan Sahin. *Design of a secondary packaging robotic system*. PhD thesis, Middle est technical university, Dec 2005.
- [22] R. Sam, K. Arrifin, and N. Buniyamin. Simulation of pick and place robotics system using solidworks softmotion. In *International Conference on System Engineering and Technology (ICSET), 2012*, pages 1 – 6. IEEE, Sept 2012.
- [23] Ralf Schubert. Process and apparatus for introducing products into containers, September 2000. Patent 6.122.895.
- [24] Daoud Slim, Chehade Hicham, Yalaoui Farouk, and Amodeo Lionel. Efficient metaheuristics for pick and place robotic systems optimization. *Journal of Intelligent Manufacturing*, 25:27–41, Feb 2014.
- [25] Staubli. Robot linemanager system, 2013. [http://pmmi.files.cms-plus.com/packexpo2013/Monday\\_11am\\_Stage2\\_RobertRochelle.pdf](http://pmmi.files.cms-plus.com/packexpo2013/Monday_11am_Stage2_RobertRochelle.pdf).
- [26] Sven Stumm, Daniel Pintar, and Bernd Kuhlencoetter. A novel concept for realistic simulation of industrial pick and place applications. In *Proceedings of ISR/Robotik 2014; 41st International Symposium on Robotics*, pages 1–7. VDE, June 2014.
- [27] D.T. Sturrock and C.D. Pegden. Recent innovations in simio. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 52 – 62. IEEE, Dec 2011.
- [28] Daniel Wappling and Steve Murphy. Pick and place, Jun 2012. Patent US 2012/0165972 A1.
- [29] Gen'ichi Yasuda. An object-oriented multitasking control environment for multirobot system programming and execution with 3d graphic simulation. In *International Journal of Production Economics*, volume 60-61, pages 241–250. Elsevier, 1999.
- [30] Gen'ichi Yasuda. Implementation of distributed control architecture for multiple robot systems using petri nets. In *Petri Nets - Manufacturing and Computer Science, Dr. Pawel Pawlewski*, pages 75–94. InTech, 2012.
- [31] MengChu Zhou and Ming-Chuan Leu. Petri net modeling of a flexible assembly station for printed circuit boards. In *Proceeding IEEE International Conference on Robotics and Automation*, volume 3, pages 2530–2535. IEEE, April 1991.
- [32] Guang-Yu Zhu and Zhi-Jin Chen. A differential evolution optimization approach to solve the pick-and-placing problem. In *Fifth International Conference on Natural Computation, 2009. ICNC '09.*, volume 4, pages 66–70. IEEE, Aug 2009.