



HAL
open science

Simulation of dynamic allocation and routing in FMS using potential field

Nadine Zbib, Yves Sallez, Damien Trentesaux

► **To cite this version:**

Nadine Zbib, Yves Sallez, Damien Trentesaux. Simulation of dynamic allocation and routing in FMS using potential field. 10th International Middle Eastern Simulation Multiconference, Sep 2009, Beirut, Lebanon. hal-03088760

HAL Id: hal-03088760

<https://hal.science/hal-03088760v1>

Submitted on 27 Dec 2020

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.

SIMULATION OF DYNAMIC ALLOCATION AND ROUTING IN FMS USING POTENTIAL FIELD

Nadine Zbib
Yves Sallez
Damien Trentesaux
UVHC, LAMIH, CNRS – UMR 8530
Univ. Lille Nord de France
F-59313 Valenciennes, France
E-mail: Nadine.Zbib@univ-valenciennes.fr

KEYWORDS

Manufacturing, decision making, discrete simulation, resource management, dynamic system.

ABSTRACT

The aim of this paper is to point out the capabilities of potential fields for the dynamic control of flexible manufacturing systems. Products sense the fields emitted by resources according to the service they request. Resources adapt their emitted field according to their states and their workloads. Routing and allocation are then simultaneously solved dynamically. Simulations are made using Netlogo software. To illustrate the features of our model, we compare the results of our application to a concurrent dynamic allocation approach based upon the contract net protocol.

INTRODUCTION

The manufacturing industry must continually deal with the increasing complexity of product requirements, as well as with the desire to decrease costs and increase control system adaptability to environmental changes. In recent decades, scientific development in the field of production has provided greater equipment automation and defined new architectures, including the heterarchical/non-hierarchical architectures that play a prominent role in Flexible Manufacturing System (FMS), leading to entities (products & resources) that are more independent, adaptive and reactive.

Traditional approach usually leads to centralized or hierarchical control structures. Due to the complexity of manufacturing problems, the usual practice has been to split the overall problem into hierarchically-dependent functions that operate within decreasing time-ranges, such as planning, scheduling and/or monitoring. This traditional approach can hardly manage unpredictable events (e.g., machine breakdowns, urgent order) and mass customization (large amount of products, each of them being particularized to each customer). More adaptable and agile control must be designed.

In this context, the objective of our work is to show how the concept of potential field can provide very reactive and agile control of production systems.

The first section describes the different approaches that are usually proposed to control interactions among entities. The second section describes a modeling approach for potential fields. Then, a potential field model dedicated to production control is proposed. Finally, a simulation and a validation are presented on a flexible assembly cell.

STATE OF THE ART

Potential fields concern the interactions among several entities in a distributed way (in opposition to a centralized way). In distributed system, many interactions are used to ensure effective decisions by entities. The distributed architectures can be categorized in four types: bionic & bio-inspired, as proposed by (Okino 1993 or Dorigo and Stützle 2004), multi-agent, as proposed by (Maione and Naso 2003), holonic, as proposed by (Van Brussel et al. 1998), and heterarchical, as proposed by (Trentesaux et al. 1998). An analysis of the state-of-the-art has been recently published by (Trentesaux 2007).

The vast majority of agents and holonic models are based on the contract-net protocol (Smith 1980). The contract net protocol which is a widely-accepted problem solving model in distributed Artificial Intelligence presents the benefit of being general, intuitive and easily applicable to different contexts. This protocol has been developed in order to specify the resolution of problems by a group of entities. The distribution of tasks is made by using a negotiation process: a discussion done between the entities relating the tasks that should be executed and those which are capable to execute the candidate task ends in electing an entity that handles the task.

In the bionic or bio-inspired approaches, stigmergy is often used; see for example ACO (ant colony optimization) algorithms (Dorigo and Stützle 2004). Another example is that of the potential field (Ferber

1995), but this technique, applied widely for example in robotics has rarely been used in the context of production (Berger et al., 2009).

Our objective is to test and implement this concept in the context of production system. Historically in our research laboratory, the interactions among entities (resources only) were governed by the *contract-net*. More recently, we were interested to *stigmergy* for routing products (Sallez et al. 2009). The application of this third interaction mechanism (*potential field*) will give us a fairly comprehensive view on the advantages and disadvantages of each concept to propose a multi-model approach.

POTENTIAL FIELD APPROACH

The concept of potential field is used in reactive systems to determine the behavior of entities. It is relatively a new technique that owns its roots in a radical critique of conventional planning techniques to determine a way forward. In the field of navigation of robots, searchers used this concept in the planning of movements, thanks to their ability to act in areas of continuous real time (Barraquand and Latombe 1989, Khatib, 1985 and Ferber 1995 and Mamei et al., 2006). Two types of potential fields are designed: a repulsive potential field to obstacles and an attractive potential field to the desired destination. A robot can then follow the path that has the greatest attractive potential field while avoiding obstacles.

This approach is used to solve the problem of positioning in a continuous environment. Moujahed (Moujahed et al., 2007), applied this approach for a better position to the bus stops on existing lines.

In manufacturing systems, (Ueda et al, 1998) have also used potential fields to ensure a dynamic allocation of products among a set of resources. They have applied this concept to a system line-less production where products (carried on AGVs) are treated by robots on mobile platforms. The field is the superposition of an attraction and a repulsion field: each machine generates an attraction force to obtain products. Each product senses the cumulated attraction field of the machines and moves accordingly.

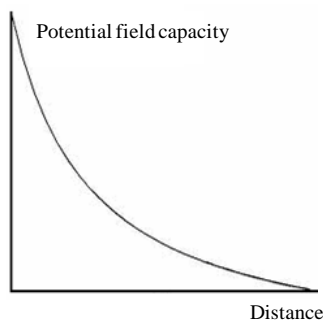


Figure 1: Variation of potential field

The capacity of potential field is inversely proportional to the Euclidean distance between the entity to attract and the center of attraction. The more the distance is small; the more the attraction is important (cf. Figure 1).

In all these potential field models, entities emit attractive potential fields propagating in a plane in all directions. Mobile entities move in 2D within a field, in any direction.

These characteristics are not relevant to typical production system and we must adapt the notion of distance according to the application context.

Figure 2 shows this type of potential field. 1D travels are rarely considered (which means that entities must move within a fixed routing network, such as pallets in a transport system, or vehicles in a city). In this paper:

- Only resources (R_j) emit fields, while products (P_i) sense and follow the gradient of these fields according to the services they request.
- Only attractive fields are considered, not repulsive ones.
- Potential fields are designed to take into consideration the workload of resources and the service they provide.

In future studies, more complex potential fields will be evaluated (e.g., both products & resources emit attractive & repulsive fields). The following section presents then our model.

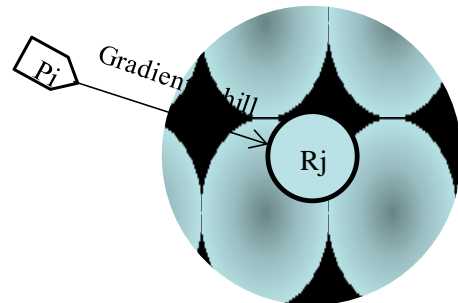


Figure 2: Product P_i moving to the resource R_j

PROPOSITION

Modeling approach

In our model, we consider active entities, which can be a resource or a product. An active entity is an entity that is able to inform, communicate, decide and act in order to reach its goals. For more detail about the concept of *active entity*, see (Zbib et al. 2007). Compared to, for example, contract net, which can be only used to dynamic allocation functionality (e.g., tasks to resources), potential field naturally enable dynamic allocation functionality simultaneously to dynamic routing functionality.

Figure 3 shows a general view of allocation and routing processes. In this figure, three resources are shown, each resource being able to satisfy services: R1 (S1, S2), R2 (S2, S3) and R3 (S1, S3). In this figure, product P1 needs to satisfy services S1, S2, S3 sequentially. More globally, this defines a flexible manufacturing system (FMS) which is the considered application case in this paper.

The topology of an FMS can usually be represented using graph theory. Resources, located on nodes, emit potential fields and active products follow a succession of arcs from node to node to complete its manufacturing. The moving of product is constrained by the topology of the FMS.

Other criteria other than the distance between product-resource should be taken into account to evaluate the intensity of the fields. In each resource, the capacity of the queue, the time of executing a service, the state of the resource (idle, busy, failed), the quality of service provided are to be considered.

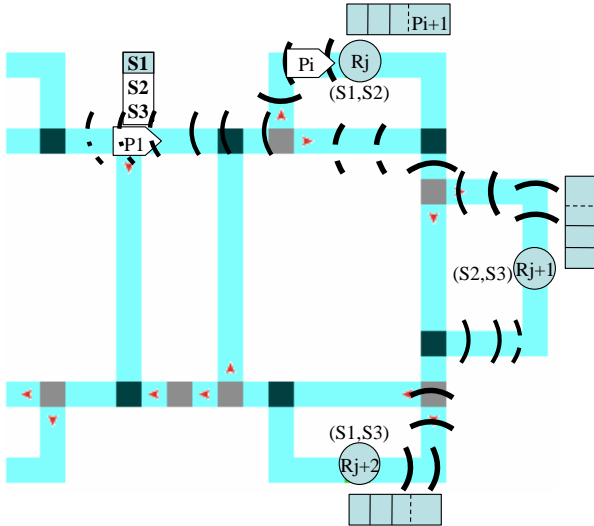


Figure 3: model of our graph

Notations

Notations are:

Let $R = (R_j)$ the set of resources located on nodes, and $P = (P_i)$ the set of active products that move in a manufacturing system to get a list of requested services $S = (S_k)$. Each R_j diffuses an attractive potential field to attract active product P_i and provide one service S_k .

A product P_i , seeking to satisfy a service S_k , goes to the resource emitting the more intense potential field for S_k .

The potential field is emitted in 1D according the topology of the graph (cf. Figure 3) and its value decreases by moving away from the resource.

In this context, the distance between two points is not the Euclidean distance but is the sum of the lengths of

arcs joining these two points. Thus the example in Figure 4, the distance between the starting node A and destination node E, is expressed as $\text{dist}(A, E) = d1 + d2 + d3 + d4$, d_i is the length of each arc i traveled.

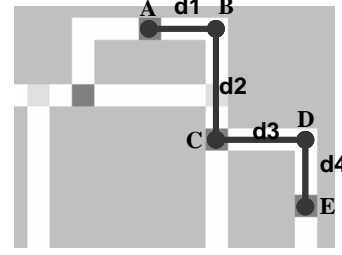


Figure 4: formula of distance

To each resource a queue is defined, $Ta(P_i, R_j)$ denotes the waiting time of a product P_i in the queue of a resource R_j .

$Tm(P_i, R_j)$ denotes the movement time of a product P_i to the resource R_j . V_{pi} is the speed of P_i during its movement.

The execution time for a service S_k to a product P_i in the resource R_j is $Ts(R_j, P_i, S_k)$. Total processing time T for a product P_i , is the time a product requires his service S_k on R_j . The form of T is therefore expressed in the formula:

$$T(R_j, P_i, S_k) = Ta(P_i, R_j) + Ts(R_j, P_i, S_k)$$

Potential field intensity is then defined as:

$$Cattr(R_j, P_i, S_k)_t = Cattr(R_j, P_i, S_k)_0 *$$

$$\left(\frac{1}{1 + (V_{pi} * Tm_{Pi, R_j})} * \frac{1}{1 + T(R_j, P_i, S_k)} \right)$$

This formula takes into consideration the velocity of products, movement times, number of products in the queue of a resource and the total processing time.

The initial value of the intensity of the potential field is $Cattr(R_j, P_i, S_k)_0$. At time 0 this value is the maximal

capacity of potential field of each resource, when none product is in the system and all resources are idle. After that, at time t , the availability of resource will decrease when it becomes busy delivering a service S_k to P_i (and / or) when some products are waiting in its queue. This value decreases also if the product has an important movement time, i.e. the distance between P_i and R_j is high.

Active product decisional behaviour

The state graph of Figure 5 describes the successive states of a product.

A product must obtain all services from its list in order to be considered finished. First, the product is moving to reach a resource that gives him a requested service. The product is on the way, it is blocked by an event that disrupts its displacement, or it reaches its resource destination.

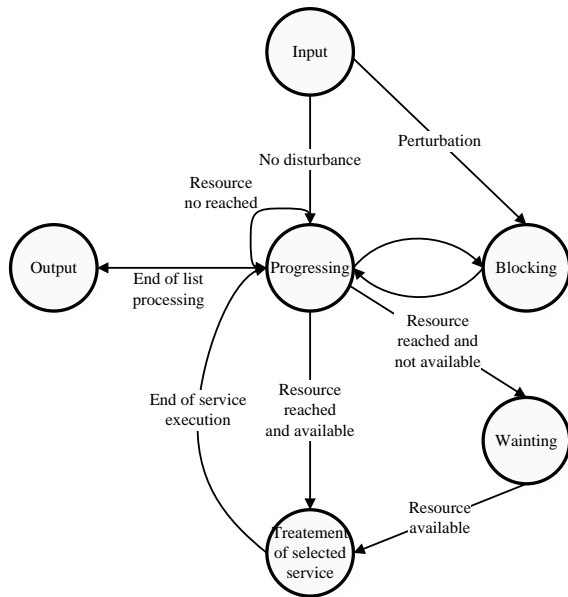


Figure 5: Product states

When the resource is reached and available, the product will be processed, but if resource is not available the product waits for the treatment; possibly, several products are in the queue of the resource, the product remains there waiting until it is selected. When the product is selected by the resource, it is processed. Then, once the list of services is completed, the product leaves the system; otherwise it continues its progress towards other resources.

SIMULATION

The proposed model is naturally distributed, meaning there is no central memorization and processing system. This property influenced our choice of an agent-based parallel modeling and simulation environment. With NetLogo (Wilensky 1999), each modeled entity can be described as an independent agent interacting with its environment. All agents operate in parallel on a grid of patches (i.e., a cellular world), and each agent can read and modify some of the attributes linked to the patches in its proximity. The behavioral rules defined for the agents make it possible to describe agent-environment interaction, which is very important when simulating the potential field concept. Netlogo was chosen because it supports easily all our desired functionalities and provides an intuitive and well-documented programming language with an elegant graphical interface.

Netlogo is based upon two major concepts: turtles (moving entity able to inform and communicate with its

environment) and patches (fixed locations on which turtles moves). In our simulation, a product is a turtle moving in his environment, and a resource is a fixed turtle able to emit potential field.

CASE STUDY AND VALIDATION

An existing benchmark developed in our laboratory has been used purpose (Trentesaux et al, 2000). This benchmark used multi-agent simulation where products and resources negotiate using the contract-net protocol to optimize dynamically allocation of tasks. Routing was simplified (not considered or roughly modeled as a constant time delay). We note that in this approach, resources are the only decisional entities for the treatment of tasks. We used the same data to compare our approach to this benchmark.

Experimental context

This case study concerns a FMS system inspired from the flexible assembly cell at the Valenciennes AIP-Primeca pole. This cell is composed of workstations placed around a flexible conveyor system based on Montrac technology (Montech 2008). This system is a monorail transport system using self-propelled shuttles to transport products. Topological graph is composed of three types of nodes (cf. Figure 6):

- Service or resource nodes: resources where products obtain services (R1, R2, R3, R4, R5, and R6).
- 11 decisional nodes, where a routing choice between two adjacent arcs must be made (gray node): when a product arrives on a decisional node, it checks the potential field value of neighbours patches to know the direction it must take to reach the best resource.
- 11 convergent nodes (black node).

A product, represented as an arrow (gray), enters and leaves on "I" node of the system.

We implemented in our simulation two types of agents with for each one private variables and parameters: resources – as fixed agents – and products – as a mobile agents –. In this simulation potential field values can be sensed by the products everywhere in the system.

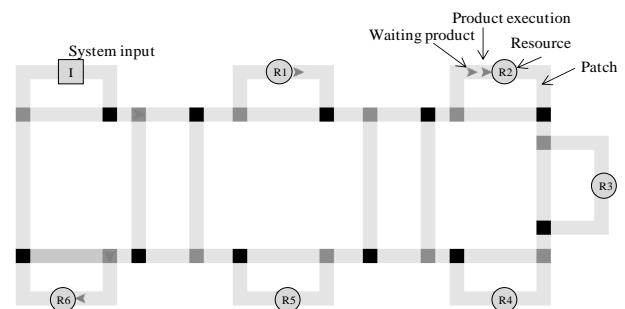


Figure 6: Topology of the FMS

The variables and parameters for a resource agent are given in Figure 7.

patch -2 17	
pxcor	-2
pycor	17
plabel	""
plabel-color	9.9
num_noeud	1
type_noeud	3
nb_neighbors	1
neighbors	[2]
dir	[1]
ch_resource	[400 376.59060017856075 384]
nb_precedents	1
precedent	[32]
dir_pre	[3]
num_resource	1

Figure 7: Variables and parameters for a fixed entity (resource agent)

For each resource, the main parameters and variables are: its location ($pxcor$, $pycor$), the reference number of the node on which the resource is located (num_noeud), the node type (1 = convergent node, 2 = decision node, 3 = service node), the number of node neighbors ($nb_neighbors$), the direction to be taken to reach the neighboring node (dir), the potential field value ($ch_resource$), this list contains values of potential field for each type of service. Another variable is the precedent nodes, and finally we can show the identifier of the resource ($num_resource$).

The variables and parameters for a product are shown Figure 8. A product with its identifier ($prod$) enters the system with a list of ordered services ($liste$) [S1, S2, S3] to be satisfied, and a list for execution time ($temps_trait$) [T1, T2, T3] for each service. Each P_i selects the first service S1 ($demande$), and moves to find the “best” resource (according to the potential flow values). When it arrives to a gray node, it checks the neighbours patch to choose the highest value of potential field corresponding to the required service. Product uphill the gradient to the chosen patch, and moves then, following the light grey patches (cf. figure 6), to reach the targeted resource/service node.

On this node, the product waits during a time $temps$ to have ($demande$). When its execution is achieved ($finish_time$), it selects the next service and moves again. When $prod$ is achieved, ($traitement_fini$) is set to true, and $prod$ goes out of the system.

turtle 6	
en_attente	false
demande	1
prod	0
liste	[1 2 0]
traitement_fini	false
avancement	0
fini	false
partenaire	nobody
temps_trait	[5 28 10]
avancement_temps	0
temps	5
temps_attente	0
temps_mouvement	11
start_time	0
finish_time	0

Figure 8: Variables and parameters for a mobile entity (product agent)

In the case study used, flexible capabilities are the following: Resource #1 is able to execute service task S1, resource #2 and resource #3 are able to execute service task S2 and resources #4, #5 and #6 are able to execute service task S3. Each manufacturing order (a set of ordered services to complete and their durations) is composed of three tasks/services, where each task is one among the three types.

Table 1: Data of simulation

Manufacturing orders	List	Processing times		
1	1-2-3	5	28	10
2	2-3-1	20	11	12
3	3-1-2	18	11	14
4	1-2-3	9	19	23
5	3-1-2	18	10	14
6	2-3-1	15	26	15
7	2-1-3	29	10	10
8	1-3-2	10	15	18
9	2-1-3	14	22	29
10	3-2-1	27	17	9
11	1-2-3	25	11	30
12	2-1-3	14	7	35
13	1-3-2	18	38	23
14	1-2-3	6	20	38
15	2-3-1	18	43	5
16	3-2-1	35	21	8
17	1-3-2	7	35	16
18	2-3-1	12	30	10
19	3-2-1	37	22	13
20	2-3-1	18	32	6
21	1-2-3	7	15	37
22	2-1-3	18	15	32
23	1-2-3	9	12	32
24	3-2-1	27	13	8
25	1-3-2	9	13	28
26	1-2-3	8	17	21
27	3-2-1	25	10	9
28	2-1-3	19	13	21
29	1-2-3	14	31	21
30	3-1-2	26	11	13

30 manufacturing orders have been defined. The list of the 30 manufacturing order is given in table 1.

Results

Scenario 1: optimality

The comparison indicator is the classical makespan value that is the final date, when all tasks have been done.

We propose 6 different simulations according to the growing number of manufacturing orders to schedule: Simulation #1 (5 manufacturing orders), Simulation #2 (10 manufacturing orders), Simulation #3 (15 manufacturing orders), Simulation #4 (20 manufacturing orders), Simulation #5 (25 manufacturing orders), Simulation #6 (30 manufacturing orders).



Figure 9: Gantt for simulation #6 (scenario 1)

Figure 9 shows an example of the obtained Gantt for simulation #6, starting and finished times for each product is provided (time is measured by Time Unit). We note that, the resource 1 is overloaded, since it is the only resource that handles the service S1, the treatment of service S2 is correctly balanced between the resource 2 and 3, while the S3 is correctly balanced among R4, R5 and R6.

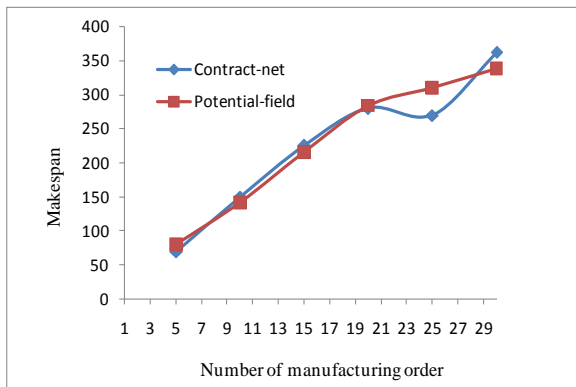


Figure 10: Evolution of the makespan values

Makespan results are given in Figure 10. Most of the makespan values provided by potential field concept are better than those provided by the contract-net based control.

In our results only product is a decisional entity that makes decisions of routing according the potential field value. We show that the execution time increases gradually with increasing number of manufacturing orders, which is logical with the concept of potential field

The decrease of execution time with contract-net in the simulation from #23 to #27 according to the authors depends on the following reason: "The local and dynamic decisions implying a restriction of visibility induced by a distributed- based approach".

Scenario 2: reactivity

Reactivity can easily be evaluated by for example, allowing or forbidding replies of resources: when a resource is unavailable (e.g., under maintenance), it will no longer reply to any request. Figure 11 illustrates a situation where resource #2 is available at date 100 and resource #4, at date 200. This implies logically an increase of the makespan value (354). This makespan is also increased with contract-net protocol (407).

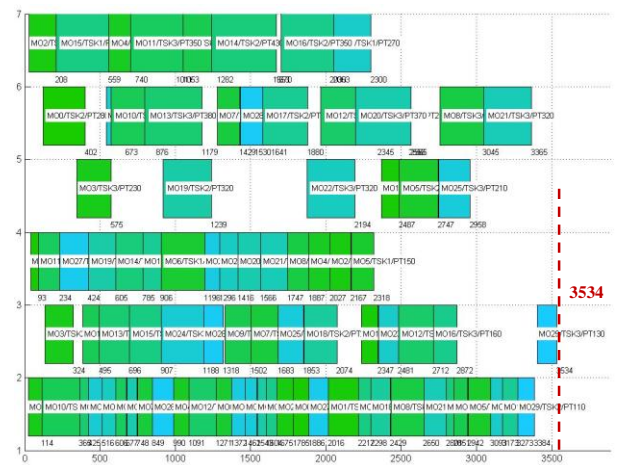


Figure 11: Impact of availability of resources

Scenario 3: scalable & realistic behavior

We extend the complexity of the study by introducing a greater number of products (60 products), to study the ability of the simulation to be scalable. We faced no issue in this case (cp. Figure 12). The number of entities does not imply complex exponential interactions, in contrast to approaches such as contract-net.

Other studies not reported in this paper show the capability of our simulation model to take into account other realistic assumptions often not considered in the literature, such as limited capacity of production stocks, jamming of the routing system, re-routing of products due to quality problems, etc.

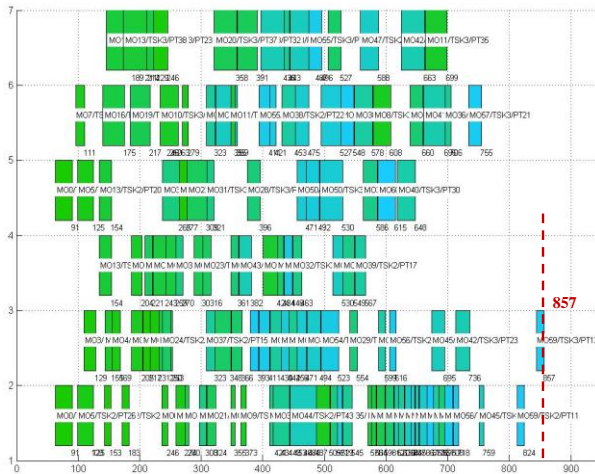


Figure 12: Gantt for 60 tasks

CONCLUSION AND PERSPECTIVES

In this paper, we have presented the promising application of the concept of potential field to the dynamic routing and allocation of products in FMS. Our models have been validated using Netlogo simulation tool. A comparison with an existing benchmark has been made.

Validating this model in a real implementation taking part for the flexible assembly cell of the AIP-PRIMECA Valenciennes will be the next challenge. Potential field will be managed by information systems. The objective will be to compare our results with the real implementation made by (Raileanu 2009) in the same assembly cell.

In our model, decisions of product must be made at decision nodes and potential fields are emitted in every point of the cell. To support this, in our on-going real implementation, a decision node will be composed of a node controller and a data communication system. The node controller will be a 750-841 (Wago 2007). It will oversee the transfer gate, supporting the routing functionalities. Product will communicate to the controller with IrDA system (Clarinet system 2007). In this node controller, the update of potential field values will be done (as a consequence, in a discrete geographical space instead of a continuous one).

REFERENCES

Barraquand J. and Latombe J. 1989, "Robot Motion Planning: A Distributed Representation Approach". Research Report, STAN-CS-89-1257, Department of Computer Science, Stanford University, 1989.

Berger T., Sallez Y., Trentesaux D., (2009) "Open control of FMS and its application to potential field", *CIRP09, 42nd Conference on Manufacturing Systems*, Wed. 3 - Fri. 5, June 2009, Grenoble, France.

Clarinet System, Network Connectivity for Mobile Devices, <http://www.clarinetsys.com>, 2007.

Dorigo, M., and Stützle, T. 2004, *Ant Colony optimization*. The MIT Press, 2004.

Ferber J. 1995, *Les systèmes multi- agents – vers une intelligence collective*, InterEditions, Paris (ISBN 2-7296-0572- X), 1995.

Khatib, O. 1985, Real-time obstacle avoidance for manipulators and mobile robots. *In IEEE international conference on robotics and automation*, pages 500-505, 1985.

Maione, G., and Naso, D. 2003, "A soft computing approach for task contracting in multi-agent manufacturing control". *Computers in Industry*, 52, 199–219, 2003.

Mamei M., Menezes R., Tolksdorf R., Zambonelli F. 2006, "Case studies for self-organization in computer science". *Journal of Systems Architecture* 52, 443-460, 2006.

Montech Technology, 2008, Conveyor systems, <http://www.montech.com>, 2008.

Moujahed S., *Approche multi-agents auto-organisée pour la résolution des contraintes spatiales dans les problèmes de positionnement mono et multi-niveaux*. Thèse de l'Université de Franche-Comté et de l'Université de Technologie de Belfort-Montbéliard, 2007.

Okino, N. 1993, "Bionic Manufacturing System in Flexible Manufacturing System: past – present – future". J. Peklenik (ed), CIRP, Paris, 73-95, 1993.

Raileanu S., Sallez Y., Beger T., Borangiu T., Trentesaux D., "Holonc implementation of the open-control paradigm". *International conference on Industrial Engineering and Systems Management*, Montreal, 2009.

Sallez Y., Berger T., Trentesaux D., (2009) "A stigmergic approach for dynamic routing of active products in FMS", *Computer In Industry*, 60(3), pp 204-216.

Smith R.G. 1980, The contract net protocol: High level communication and control in a distributed problem solver, *IEEE Transactions on Computer*, vol. C29, n°12, pp. 1104-1113, 1980.

Trentesaux, D., Dindeleux, R. and Tahon, C. 1998, "A MultiCriteria Decision Support System for Dynamic task Allocation in a Distributed Production Activity Control Structure". *Int. J. of Computer Integrated Manufacturing*, 11 (1), 3-17, 1998.

Trentesaux D., Pesin P. and Tahon C. 2000, "Distributed artificial intelligence for FMS scheduling, control and design support". *Journal of Intelligent Manufacturing* 11, 573–589, 2000.

Trentesaux, D. 2007, Les systèmes de pilotage hétérarchiques : innovations réelles ou modèles stériles ?. *Journal Européen des Systèmes Automatisés*, 41 (9-10), 1165-1202, 2007.

Ueda K., Vaario J. 1998, "An emergent modelling method for dynamic scheduling". *Journal of Intelligent Manufacturing*, pp. 129-140, 1998.

Van Brussel, H., Wyls, J., Valckenaers, P., Bongaerts, L. and Peeters, L. 1998, "Reference architecture for holonic manufacturing systems: PROSA". *Computers in Industry*, 37 (3), 255–274, 1998.

Wago system, innovative connections, <http://www.wago.com>, 2007.

Wilensky U. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, 1999.

Zbib N., Raileanu S., Sallez Y., Berger T., Trentesaux D. 2008, From "Passive Products to Intelligent Products: The Augmentation Module Concept". *5th int. CIRP-sponsored DET conference*, Nantes, France, 22-24th October 2008.