Proposition d'une méthode d'apprentissage à base de produits intelligents pour les ateliers partiellement flexibles / dynamiques

Wassim Bouazza

▶ **To cite this version:**

Wassim Bouazza. Proposition d'une méthode d'apprentissage à base de produits intelligents pour les ateliers partiellement flexibles / dynamiques. Automatique. Université Oran 1 - Ahmed Ben Bella, 2020. Français. NNT : . tel-02568076

HAL Id: tel-02568076

https://theses.hal.science/tel-02568076v1

Submitted on 8 May 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.

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
كلية العلوم الدقيقة والتطبيقية
Faculté des sciences exactes et appliquées

THÈSE

En vue de l'obtention du diplôme de
Doctorat en Sciences en Informatique
Spécialité : Informatique et automatique

PROPOSITION D'UNE METHODE D'APPRENTISSAGE A BASE DE PRODUITS INTELLIGENTS POUR LES ATELIERS PARTIELLEMENT FLEXIBLES / DYNAMIQUES

Présentée par : **BOUAZZA Wassim**

Soutenue publiquement le 26 Janvier 2020 devant le jury composé de :

Pr. TAGHEZOUT Noria	Université Oran 1 Ahmed Ben Bella	Présidente
Pr. KHELFI Mohamed Fayçal	Université Oran 1 Ahmed Ben Bella	Examinateur
Pr. AHMED-FOITIH Zoubir	Université USTO Mohamed Boudiaf	Examinateur
Pr. FARAOUN Mohamed Kamel	Université Djilali Liabés, Sidi Bel Abbés	Examinateur
Pr. HAMDADOU Djamila	Université Oran 1 Ahmed Ben Bella	Directrice de thèse
Pr. SALLEZ Yves	Université Polytechnique des Hauts-de-France	Co-directeur de thèse
Pr. BELDJILALI Bouziane	Université Oran 1 Ahmed Ben Bella	Invité

« *Le savoir acquis dans un pays étranger peut être une patrie et l'ignorance peut être un exil vécu dans son propre pays.* »

IBN ROCH (AVERROES)

DEDICACES

À mes parents. À mon épouse et à mon fils. À ma sœur et à mes nièces. À ma famille et amis, ainsi qu'à tous ceux qui m'ont aidé et soutenu.

REMERCIEMENTS

Je tiens tout d'abord à remercier mes directeurs de thèse, le Pr. BELDJILALI Bouziane et le Pr. HAMDADOU Djamila de leur aide, de leur bienveillance et de la confiance qu'ils m'ont toujours témoigné. Je remercie tout particulièrement mon co-directeur le Pr. SALLEZ Yves qui n'a pas compté ses efforts pour que ce travail puisse aboutir. Je le remercie pour ses précieux conseils, pour la gentillesse et la patience qu'il a manifesté à mon égard. J'espère avoir été à la hauteur.

Mes remerciements vont également aux membres du jury, à sa présidente le Pr. TAGHEZOUT Noria*, ainsi qu'aux examinateurs le Pr. KHELFI Mohamed Fayçal, le Pr. AHMED-FOITIH Zoubir, ainsi que le Pr. FARAOUN Mohamed Kamel, pour l'honneur qu'ils m'ont fait en acceptant d'évaluer ce travail, pour leur temps et pour leur expertise.*

Je remercie l'équipe d'automatique du LAMIH de Valenciennes et les membres de l'IUT de Cambrai : le Pr. TRENTESAUX Damien, le Dr. BEKRAR Abdelghani, le Pr. DENEUX Dominique, le Dr CHAABANE Sondes, AUREGGI Corinne, GENEVIEVE Marlene mais aussi tous les autres, pour leur accueil chaleureux, leur aide et leur amitié

Je remercie mon ami le Dr. DIETSCH Damien pour ses lectures, ainsi que mon ami le Dr. BEKKAR Azzedine pour son aide précieuse et son soutien.

Il m'est incontestablement impossible de dire par des mots, toute la gratitude que j'ai pour mes parents. Sans leur amour et leurs sacrifices, je ne serais rien. Puissent-ils ressentir l'ineffable amour que je leur porte.

Je remercie mon épouse Narimane pour son indéfectible soutient, sa patience durant ces années parfois difficiles, ma sœur Narimane de son affection, elle qui m'a toujours aidé et soutenu, mon fils Samy et mes nièces Ines et Sarah qui (entre deux bêtises) me remplissent de joie, à mes tantes, cousins et cousines. À ma belle-famille. À mes amis d'Oran et de Valenciennes que j'ai eu la chance de croiser : Mohamed, Kamel, Sofiane, Sandro, Jocelene, Rym, Jose, Farid, Amandine ainsi qu'à tous les autres … Merci à vous tous

▪ **Résumé :** avec la quatrième révolution industrielle que nous vivons actuellement, un ordonnancement efficace des opérations de production est plus que jamais indispensable à la pérennité des industries 4.0. Si la flexibilité des FMS (Flexible Manufacturing System) procure des avantages face aux environnements incertains, elle implique un accroissement de la complexité des fonctions de pilotage. Les architectures distribuées semblent apporter l'agilité et la réactivité nécessaire pour remédier à cet inconvénient. Elles souffrent néanmoins de limitations, celles-ci étant essentiellement liées au phénomène de myopie décisionnelle. Ainsi, toute la difficulté du pilotage dynamique des systèmes de production cyber-physiques réside dans l'équilibre à trouver entre une réaction rapide face aux perturbations, et le maintien d'une performance globale satisfaisante. Cette thèse vise à proposer un modèle original applicable à un grand nombre de configurations d'atelier de production (i.e. flowshop, jobshop, etc.). En plus de la prise en charge d'évènements dynamiques, le modèle proposé intègre des contraintes majeures telles que les familles de produits, l'interopérabilité et les temps de configuration. L'architecture hétérarchique proposée assure un pilotage par les produits exclusivement. Ces derniers sont intelligents et autonomes. La contribution principale de ce travail de thèse réside dans le développement ? d'une hyper-heuristique qui repose principalement sur des règles d'ordonnancement rapides utilisées par les produits intelligents. Pour contrebalancer les performances globales faibles des produits intelligents, le concept de contexte décisionnel est développé pour améliorer la pertinence des règles sélectionnées. L'hyper-heuristique proposée est basée sur un modèle original d'encapsulation de règles combiné à un outil de simulation réduisant le phénomène de myopie. Des expérimentations sur quatre configurations d'atelier distinctes, 64 instances de scenarios dynamiques différentes, et plus de 30 400 simulations, viennent valider l'intérêt de l'approche proposée.

Mots clés : *ordonnancement manufacturier - FMS - produit intelligent - contrôle hétérarchique - pilotage réactif - hyper-heuristique - règles d'ordonnancement – optimisation-simulation.*

▪ **Abstract:** with the fourth industrial revolution that we are currently experiencing, efficient scheduling of production operations is more than ever necessary for the sustainability of 4.0 industries. While the flexibility of the Flexible Manufacturing System (FMS) provides advantages over uncertain environments, it implies an increasing complexity of the control functions. In this, distributed architectures seem to provide the agility and responsiveness needed. However, they suffer from limitations, mainly due to the myopia phenomenon in decision-making process. Thus, the main difficulty of dynamic control of cyber-physical production systems lies in the balance to be found between a rapid response to disturbances, and the upkeep of a satisfactory overall performance. This thesis proposes an original model that can be applied to a large types of production workshop configurations (flowshop, jobshop, etc.). In addition to the handling of dynamic events, it includes constraints such as multiple product families, interoperability constraints, configuration, etc. The heterarchical architecture thus proposed ensures a control by products exclusively. These are intelligent and autonomous. The main contribution of the thesis work is a hyper-heuristic based mostly on fast scheduling rules used by smart products. To counterbalance their weak overall performance, the concept of decisional context is developed to improve the relevance of selected rules. The proposed hyper-heuristic is based on an original model of encapsulation of rules combined with a simulation tool reducing the phenomenon of myopia. Experiments on four distinct workshop configurations, 64 instances of different dynamic scenarios and more than 30,400 simulations validate the interest of the

proposed approach.

Keywords: *Manufacturing scheduling - FMS - Intelligent product - heterarchical control - reactive control - hyper-heuristic - dispatching rules - simulation-optimization.*

▪ **ملخص:** مع الثورة الصناعية الرابعة التي نشهدها حاليًا، أصبحت عملية الجدولة الفعالة لعمليات الإنتاج ضرورية أكثر من أي وقت مضى لاستدامة 4.0 صناعات. في حين أن مرونة نظام التصنيع المرن (FMS) توفر مزايا على البيئات غير المؤكدة، فإنها تنطوي على زيادة في تعقيد وظائف التحكم. في هذا، يبدو أن الأبنية الموزعة توفر الرشاقة والاستجابة اللازمة. ومع ذلك، فإنها تعاني من القيود، ويرجع ذلك أساسا إلى ظاهرة قصر النظر في صنع القرار. وبالتالي، تكمن صعوبة التحكم الديناميكي في أنظمة الإنتاج الفيزيائي السيبراني في التوازن الذي يمكن العثور عليه بين الاستجابة السريعة للاضطرابات، والحفاظ على الأداء الكلي المرضي .تقترح هذه الرسالة نموذجًا أصليًا يمكن تطبيقه على عدد كبير من تكوينات ورشة الإنتاج (flowshop ، jobshop، إلخ (. بالإضافة إلى دعم الأحداث الديناميكية، فإنه يتضمن قيودًا مثل مجموعات المنتجات وقابلية التشغيل البيني وأوقات التهيئة والمزيد. تضمن البنية المتغايرة المقترحة بالتالي توجيه المنتجات بشكل حصري. هذه هي ذكية ومستقلة. المساهمة الرئيسية لهذا العمل هو أطروحة مفرطة الاستدلال تعتمد أساسا على قواعد جدولة سريعة تستخدمها المنتجات الذكية. لموازنة أدائها العام الضعيف، تم تطوير مفهوم سياق القرار لتحسين ملاءمة القواعد المختارة. يعتمد فرط الاستكشاف المقترح على نموذج أصلي لتغليف القواعد مع أداة محاكاة تقلل من ظاهرة قصر النظر. إن التجارب على أربع تكوينات مختلفة لورشة العمل و64 حالة من السيناريوهات الديناميكية المختلفة وأكثر من 30 400 محاكاة تثبت صحة النهج المقترح.

الكلمات الرئيسية: *جدولة التصنيع - FMS - المنتج الذكي - التحكم الموزع - التحكم التفاعلي - الاستدلال المفرط - قواعد الجدولة - تحسين المحاكاة.*

TABLE DES ACRONYMES

ACO	Ant Colony Optimization
ADACOR	ADAptive holonic COntrol aRchitecture
AG	Algorithme Génétique
ANN	Artificial Neural Network
AR	Assignement Rule
BRC	Best Rules Combination
CPPS	Cyber-Physical Production System
DMS	Dedicated Manufacturing System
DR	Dispatching Rule
FFSP	Flexible Flowshop Scheduling Problem
FJSP	Flexible Jobshop Scheduling Problem
FMS	Flexible Manufacturing System
FSP	Flowshop Scheduling Problem
GEP	Gene Expression Programming
HAR	Heuristique d'Allocation de Ressources
HFSP	Hybrid Flexible Flowshop Scheduling Problem
HH	Hyper-Heuristique
HMS	Holonic Manufacturing Systems
HS	Heuristique de Séquencement
IA	Intelligence Artificielle
ILS	Iterated Local Search
IoT	Internet of Things
JSP	Jobshop Scheduling Problem
MH	Méta-Heuristique
MILP	Mixed Integer Linear Programming
MSR	Machine Selection Rule
PA	Produit Actif
PAR	Politique d'allocation de ressource
PDSC	Product-Driven System Control
PF	Potential Field
PG	Programmation Génétique
PI	Produit Intelligent
PS	Politique de Séquencement
RMS	Reconfigurable Manufacturing System
RTS	Real-Time Scheduling
SMA	Système MultiAgents
SP	Système de Production
SR	Sequencing Rule
TAM	Temps d'Achèvement Moyen
VNS	Variable Neighborhood Search
WIP	Work-In-Process agent
VAP	Produit Actif Virtuel

TABLE DES MATIERES

TABLE DES ACRONYMES 6
TABLE DES MATIERES 7
TABLE DES FIGURES 9
TABLE DES TABLEAUX 12
INTRODUCTION GENERALE 13
1. CONTEXTE 13
2. PROBLEMATIQUE 14
3. CONTRIBUTIONS 15
4. PLAN DU DOCUMENT 16
PREMIERE PARTIE : SYNTHESE DE L'ETAT DE L'ART 18
CHAPITRE I : ORDONNANCEMENT DYNAMIQUE DES FMS 19
1. INTRODUCTION 19
2. FLEXIBILITES DES SYSTEMES DE PRODUCTION 19
2.2. LES BESOINS EN FLEXIBILITE 20
2.3. TYPOLOGIE DES FLEXIBILITES 20
2.4. LES SYSTEMES DE PRODUCTION FLEXIBLES (FMS) 23
3. PROBLEMES D'ORDONNANCEMENT « α|β|γ » 24
3.1. TYPE D'ATELIER « α » 24
3.2. CONTRAINTES « β » 27
3.3. INDICATEURS DE PERFORMANCE « γ » 30
4. TRAVAUX RELATIFS A L'OPTIMISATION DE L'ORDONNANCEMENT DYNAMIQUE 33
4.1. APPROCHES ANALYTIQUES (A) 34
4.2. HEURISTIQUES (H) 35
4.3. META-HEURISTIQUES (MH) 37
4.4. HYPER-HEURISTIQUES (HH) 39
4.5. SYNTHESE SUR L'OPTIMISATION DE L'ORDONNANCEMENT DYNAMIQUE 41
5. CONCLUSION 42
CHAPITRE II : PILOTAGE DES SYSTEMES DE PRODUCTION CYBER-PHYSIQUES (CPPS) 43
1. INTRODUCTION 43
2. PILOTAGE DES SYSTEMES DE PRODUCTION (SP) 43
2.1. VUES DECISIONNELLE ET FONCTIONNELLE DU SP 44
2.2. TYPES DE PILOTAGE DE LA PRODUCTION 45
2.3. ARCHITECTURES DES SYSTEMES DE PILOTAGE 47
3. CONDUITE DES SYSTEMES DE PRODUCTION CYBER-PHYSIQUES (CPPS) 49
3.1. PRINCIPES DES CPPS 49
3.2. DESAGREGATION DE LA PYRAMIDE D'AUTOMATISATION 50
3.3. PROBLEMATIQUE DE PILOTAGE DES CPPS EN ENVIRONNEMENT MANUFACTURIER COMPLEXE 51
4. APPORTS DES PRODUITS INTELLIGENTS AU PILOTAGE DES CPPS 54
4.1. CONCEPTS DE PRODUIT INTELLIGENT 54
4.2. LES SYSTEMES CONTROLES PILOTES PAR LE PRODUIT 58
5. TRAVAUX RELATIFS AUX SYSTEMES DE PRODUCTION PILOTES PAR LES PRODUITS 60
5.1. APPROCHES ORIENTEES PRODUITS INTELLIGENTS 60
5.2. APPROCHES MULTIAGENTS 61
5.3. APPROCHES HOLONIQUES 63
5.4. SYNTHESE DES TRAVAUX RELATIFS AUX SYSTEMES DE PRODUCTION PILOTES PAR LES PRODUITS 64
6. CONCLUSION 65

DEUXIEME PARTIE : CONTRIBUTION AUX CPPS PILOTES PAR LES PRODUITS 66

CHAPITRE III : APPROCHE PROPOSEE 67
1. INTRODUCTION 67
2. FORMULATION DU PROBLEME 67
2.1. HYPOTHESES 67
2.2. FORMALISATION DU PROBLEME D'OPTIMISATION 68
2.3. BESOINS ET SPECIFICATIONS 70
3. APPROCHE PROPOSEE 71
3.1. CHOIX ET MOTIVATIONS DE L'APPROCHE 71
3.2. MODELISATION DU CPPS 72
3.3. CONTEXTE DECISIONNEL 75
3.4. ENCAPSULATION DES REGLES DANS LES STRATEGIES DECISIONNELLES 75
3.5. COUPLAGE OPTIMISATION-SIMULATION 79
3.6. FONCTIONNEMENT GLOBAL DE L'HYPER-HEURISTIQUE 84
4. CONCLUSION 85

CHAPITRE IV : EXPERIMENTATIONS 86
1. INTRODUCTION 86
2. OUTIL DE SIMULATION A BASE DE SMA 86
2.1. FRAMEWORK MULTIAGENTS JADE 86
2.2. OUTIL DE SIMULATION DEVELOPPE 87
3. DONNEES EN ENTREE 89
3.1. LES DIFFERENTES CONFIGURATIONS D'ATELIERS 91
3.2. GENERATEUR DE SCENARIOS 94
3.3. PARAMETRAGE DES ALGORITHMES 97
4. RESULTATS EXPERIMENTAUX 99
4.1. FLOWSHOP FLEXIBLE HYBRIDE HF1 99
4.2. FLOWSHOP FLEXIBLE HYBRIDE HF2 103
4.3. JOBSHOP FLEXIBLE FJ1 107
4.4. JOBSHOP FLEXIBLE FJ2 111
5. DISCUSSION 115
5.1. ETUDE DE L'EVOLUTION DES BRC 115
5.2. PERFORMANCES DES HH 116
6. CONCLUSION 118

CONCLUSION GENERALE ET PERSPECTIVES 119

BIBLIOGRAPHIE 121

ANNEXE A : DISTRIBUTIONS DES PRODUITS 133
1. DISTRIBUTIONS HF1 133
2. DISTRIBUTIONS HF2 133
3. DISTRIBUTIONS FJ1 134
4. DISTRIBUTIONS FJ2 134

ANNEXE B : DETAILS DES SIMULATIONS 135
1. RESULTATS HF1 135
2. RESULTATS HF2 136
3. RESULTATS FJ1 137
4. RESULTATS FJ2 139

TABLE DES FIGURES

Figure 1 Positionnement de la thèse....14
Figure 2 Présentation du plan de la thèse....16
Figure 3 Liens entre les différentes flexibilités (Sethi et al. 1990)....21
Figure 4 Exemple de robot industriel à outils interchangeables....21
Figure 5 Exemples de systèmes de manutention automatisés....21
Figure 6 Classification des systèmes de production selon Variété/Volume (Mehrabi et al. 2002)....23
Figure 7 Exemple d'ateliers à machines parallèles....25
Figure 8 Exemples d'atelier flowshop et ses variantes....26
Figure 9 Exemples d'atelier jobshop et ses variantes....27
Figure 10 Fonctions dépendantes de la date d'échéance (Framinan et al. 2014)....31
Figure 11 Principe du couplage optimisation-simulation....34
Figure 12 Approches pour l'optimisation de l'ordonnancement dynamique....34
Figure 13 Vue décisionnelle et fonctionnelle d'un SP....44
Figure 14 Types de pilotages de la production....47
Figure 15 Organisation structurelle des systèmes de pilotages (Trentesaux 2009)....48
Figure 16 Les 4 révolutions industrielles....49
Figure 17 Convergence virtuelle et physique dans les CPPS....50
Figure 18 Désagrégation de la pyramide d'automatisation dans les CPPS....51
Figure 19 Métaphore de la myopie d'un agent décisionnel....53
Figure 20 Evolution du paradigme du produit intelligent (Filho et al. 2017)....54
Figure 21 Les différents vocables associés au produit « intelligent / actif » (Sallez 2012a)....55
Figure 22 Entité produit de Brun-Picard et al. (1997)....57
Figure 23 Produit actif en environnement d'intelligence ambiante (Dobre et al. 2009)....57
Figure 24 Les composants hardwares d'un produit intelligent (Quintanilla et al. 2013)....57
Figure 25 Exemple de « Boite de sauce spaghetti intelligente » (Wong et al. 2002)....58
Figure 26 Classification des produits intelligents....58
Figure 27 Architecture PDSC (Pétin et al. 2007; Gouyon 2007)....59
Figure 28 Integration des décisions « produits » dans la boucles de contrôle (McFarlane et al. 2013)....60
Figure 29 Exemple d'un VAP utlisant la stigmergie pour le routage (Sallez et al. 2009)....61
Figure 30 Exemple de négociation entre agents (Tehrani Niknejad et al. 2010)....62
Figure 31 Les composants de l'architecture PROSA (Van Brussel et al. 1998)....63
Figure 32 Formulation du probleme et besoins induits....70
Figure 33 Interaction entre les composants de l'approche proposée et adéquation avec les besoins....71
Figure 34 Exemple de représentation des cellules de production (Niveau physique)....72
Figure 35 Les couches informationelle et physique du CPPS modélisé....73
Figure 36 Diagramme d'états du PI....73
Figure 37 Architecture interne du produit intelligent....74
Figure 38 Diagramme d'états de l'entité informationelle « resource »....74
Figure 39 Performances de 2 règles VS heuristique combinant ces même règles....75
Figure 40 Modèle générique d'encapsulation des règles dans la stratégie décisionnelle....76
Figure 41 Exemple d'encapsulation des régles pour la stratégie proposée « ITV »....77
Figure 42 Classification de la stratégie proposée ITV et les variantes....77
Figure 43 Les différents schémas d'allocation de règles....78
Figure 44 Example d'encapsulation des régles pour la variante « CELL_ITV »....78
Figure 45 Example d'encapsulation des règles pour la variante « C+R »....79
Figure 46 Example d'encapsulation des règles pour la variante « CELL »....79
Figure 47 Example de règles MSR et SR dans une combinaison « RC »....79
Figure 48 Diagramme d'activité de la composante informationelle de la ressource....80

Figure 49 Diagramme d'activité générique du PI développé 81
Figure 50 Diagramme d'activité de la simulation du systéme de production 82
Figure 51 Exemple des procédures de croisement developpées 83
Figure 52 Exemple des procédures de mutation developpées 84
Figure 53 Diagramme d'activité globale de l'hyper-heuristique intégrant les simulations 85
Figure 54 Norme FIPA: services fournis par une plateforme (Bellifemine et al. 2003) 87
Figure 55 Fonctionnement global de l'outil de simulation 88
Figure 56 Diagramme de séquence simplifié d'une expérimentation type 89
Figure 57 Exemple simplifié d'une ligne de production de produits pharmaceutiques 89
Figure 58 Contrôle des paramétres experimentaux pour la génération d'intances 90
Figure 59 Configuration flowshop flexible hybride HF1 91
Figure 60 Configuration flowshop flexible hybride HF2 92
Figure 61 Configuration jobshop partiellement flexibles FJ1 93
Figure 62 Configuration jobshop partiellement flexibles FJ2 93
Figure 63 Outil de génération des scénarios 95
Figure 64 Cadences moyennes d'arrivée de produit pour la génération des scripts 95
Figure 65 Arrivées des produits pour les scripts de S1.1 à S1.8 pour HF1 96
Figure 66 Arrivées des produits pour les scripts S2.1 à S2.8 pour HF2 96
Figure 67 Arrivées des produits pour les scripts de S3.1 à S3.8 pour FJ1 96
Figure 68 Arrivées des produits pour les scripts S4.1 à S4.8 pour FJ2 97
Figure 69 Nombre d'opérations requis pour les différents scripts 97
Figure 70 Occurrences des pannes p=0.01 97
Figure 71 Tracé des principaux effets des paramètres des AG pour HF1 100
Figure 72 Evolutions du nombre de produits dans le FMS pour les scenarios S1.1a & S1.1.b 101
Figure 73 Moyennes des TAM (avec et sans pannes) pour la configuration HF1 101
Figure 74 Performances relatives des HH par rapport aux BRC pour HF1 101
Figure 75 Tracé des principaux effets des paramètres des AG pour HF2 103
Figure 76 Evolutions du nombre de produits dans le FMS scenarios S2.1a & S2.1b 104
Figure 77 Moyenne des TAM (avec et sans panne) pour la configuration HF2 104
Figure 78 Performances relatives des HH par rapport aux BRC pour HF2 105
Figure 79 Tracés des principaux effets des paramètres des AG pour FJ1 107
Figure 80 Evolutions du nombre de produits dans le FMS scenarios S3.1a et S3.1b 108
Figure 81 Moyenne des TAM (avec et sans panne) pour la configuration FJ1 108
Figure 82 Performances relatives des HH par rapport aux BRC pour FJ1 109
Figure 83 Tracé des principaux effets des paramètres des AG pour FJ2 111
Figure 84 Evolutions du nombre de produits dans le FMS scenarios S4.1a & S4.1b 112
Figure 85 Moyenne des TAM (avec et sans panne) pour la configuration FJ2 112
Figure 86 Performances relatives des HH par rapport aux BRC 113
Figure 87 Occurrences des règles MSR et SR dans les 64 BRC 115
Figure 88 Changement de règles dans les BRC pour les scenarios avec et sans pannes 116
Figure 89 Moyennes des performances relatives de chaque HH par configuration 116
Figure 90 Moyennes des Performances relatives des HH par cadences d'arrivée 117
Figure 91 Incidences des pannes 117
Figure 92 Ecart-Type des HH pour les 4 configurations 118
Figure 93 Distributions des familles de produits pour les 8 scripts pour HF1 133
Figure 94 Distributions des priorités des produits pour les 8 scripts pour HF1 133
Figure 95 Distributions des familles de produits pour les 8 scripts pour FJ1 133
Figure 96 Distributions des priorités des produits pour les 8 scripts pour FJ2 133
Figure 97 Distributions des familles de produits pour les 8 scripts pour FJ1 134
Figure 98 Distributions des priorités des produits pour les 8 scripts pour FJ1 134

Figure 99 Distributions des familles de produits pour les 8 scripts pour FJ2 134
Figure 100 Distributions des priorités des produits pour les 8 scripts pour FJ2 134
Figure 101 Evolutions du nombre de produits dans le FMS scenarios S1.3a & S1.3b.....135
Figure 102 Evolutions du nombre de produits dans le FMS scenarios S1.4a & S1.4b.....135
Figure 103 Evolutions du nombre de produits dans le FMS scenarios S1.5a & S1.5b.....135
Figure 104 Evolutions du nombre de produits dans le FMS scenarios S1.6a & S1.6b.....135
Figure 105 Evolutions du nombre de produits dans le FMS scenarios S1.7a & S1.7b.....136
Figure 106 Evolutions du nombre de produits dans le FMS scenarios S1.8a & S1.8b.....136
Figure 107 Evolutions du nombre de produits dans le FMS scenarios S2.2a & S2.2b.....136
Figure 108 Evolutions du nombre de produits dans le FMS scenarios S2.3a & S2.3b.....136
Figure 109 Evolutions du nombre de produits dans le FMS scenarios S2.6a & S2.6b.....137
Figure 110 Evolutions du nombre de produits dans le FMS scenarios S2.7a & S2.7b.....137
Figure 111 Evolutions du nombre de produits dans le FMS scenarios S2.8a & S2.8b.....137
Figure 112 Evolutions du nombre de produits dans le FMS scenarios S3.2a & S3.2b.....137
Figure 113 Evolutions du nombre de produits dans le FMS scenarios S3.3a & S3.3b.....138
Figure 114 Evolutions du nombre de produits dans le FMS scenarios S3.4a & S3.4b.....138
Figure 115 Evolutions du nombre de produits dans le FMS scenarios S3.5a & S3.5b.....138
Figure 116 Evolutions du nombre de produits dans le FMS scenarios S3.6a & S3.6b.....138
Figure 117 Evolutions du nombre de produits dans le FMS scenarios S3.8a & S3.8b.....139
Figure 118 Evolutions du nombre de produits dans le FMS scenarios S4.2a & S4.2b.....139
Figure 119 Evolutions du nombre de produits dans le FMS scenarios S4.3a & S4.3b.....139
Figure 120 Evolutions du nombre de produits dans le FMS scenarios S4.5a & S4.5b.....139
Figure 121 Evolutions du nombre de produits dans le FMS scenarios S4.6a & S4.6b.....140
Figure 122 Evolutions du nombre de produits dans le FMS scenarios S4.7a & S4.7b.....140
Figure 123 Evolutions du nombre de produits dans le FMS scenarios S4.8a & S4.8b.....140

TABLE DES TABLEAUX

Tableau 1 Facteurs liés aux besoins en flexibilité 20
Tableau 2 Classification des problèmes avec la représentation « α|β|γ » 32
Tableau 3 Travaux relatifs à l'optimisation de l'ordonnancement dynamique 41
Tableau 4 Classification des aspects de la complexité (Framinan et al. 2014) 52
Tableau 5 Synthése des travaux relatifs aux systemes de production pilotés par les produits 65
Tableau 6 Hypothèses et postulats du probléme abordé 67
Tableau 7 Principaux paramétres du modéle de stratégie. 77
Tableau 8 Contraintes de précédence pour la configuration HF1 91
Tableau 9 Contraintes de précédence pour la configuration HF2 92
Tableau 10 Contraintes de précédence pour la configuration FJ1 93
Tableau 11 Contraintes de précédence pour la configuration FJ2 94
Tableau 12 Variations des paramatres expérimentaux pour les configurations 94
Tableau 13 Les différentes règles « atomiques » utilisées 98
Tableau 14 Détail du nombre de simulations 99
Tableau 15 ANOVA de l'algorithme "ITV" pour HF1 99
Tableau 16 ANOVA de l'algorithme "CELL_ITV" pour HF1 99
Tableau 17 Paramètres des HH après ANOVA pour HF1 100
Tableau 18 Synthèse des résultats pour HF1 102
Tableau 19 ANOVA de l'algorithme "ITV" pour HF2 103
Tableau 20 ANOVA de l'algorithme "CELL_ITV" pour HF2 103
Tableau 21 ANOVA de l'algorithme "C+R" pour HF2 103
Tableau 22 ANOVA de l'algorithme "CELL" pour HF2 103
Tableau 23 Paramètres des HH après ANOVA pour HF2 104
Tableau 24 Synthèse des résultats pour HF2 106
Tableau 25 ANOVA de "ITV" pour FJ1 107
Tableau 26 ANOVA for "CELL_ITV" FJ1 107
Tableau 27 ANOVA for "CELL " FJ1 107
Tableau 28 ANOVA for "C+R " FJ1 107
Tableau 29 Paramètres des HH après ANOVA pour FJ1 108
Tableau 30 Synthèse des résultats pour FJ1 110
Tableau 31 ANOVA de l'algorithme "ITV" pour FJ2 111
Tableau 32 ANOVA de l'algorithme "CELL_ITV" pour FJ2 111
Tableau 33 ANOVA de l'algorithme "CELL" pour FJ2 111
Tableau 34 ANOVA de l'algorithme "C+R" pour FJ2 111
Tableau 35 Paramètres des HH après ANOVA pour FJ2 112
Tableau 36 Synthèse des résultats pour FJ2 114

INTRODUCTION GENERALE

1. CONTEXTE

L'évolution toujours plus rapide des technologies redessine perpétuellement le futur de nos sociétés. Le secteur industriel ne fait pas exception, ce dernier étant même souvent précurseur, car intégrant des technologies parfois à peine émergentes. Par exemple, les entreprises industrielles ont intégré très tôt les premiers ordinateurs pour la conception et la fabrication de nouveaux produits. Les robots ont eux aussi trouvé l'une de leurs premières applications réelles sur les chaines de montage. Cette évolution technologique des systèmes manufacturiers continue d'accélérer sans discontinuer. La quatrième révolution industrielle s'appuie sur l'essor de nouveaux paradigmes tels que les Systèmes Cyber-Physiques, l'Internet des objets (IoT), le Big Data ou encore les technologies Cloud. Elle se veut une réponse aux nouveaux défis de flexibilité, d'agilité et d'adaptabilité toujours plus pressants. L'industrie 4.0 définit alors une nouvelle organisation des usines, les transformant en *« Usines Intelligentes », « Smart Factories » ou CPPS (Cyber-Physical Production Systems).* Si la fin du XX^{ème} siècle était l'ère de l'échange d'informations grâce aux télécommunications, le début du XXI^{ème} est, sans conteste, celui du traitement de ces informations par des techniques d'intelligence artificielle. En effet, de nouvelles disciplines et techniques, telle que le Machine Learning, promettent aux entreprises une réponse pertinente et efficace aussi bien à la recherche de la performance commerciale sur des marchés de plus en plus globalisés et concurrentiels, qu'aux problématiques contemporaines de gestion des ressources ou de l'énergie, par exemple.

Ainsi, les systèmes de fabrication flexibles et/ou reconfigurables sont utilisés pour créer une grande variété de produits, notamment par assemblage combinatoire. Grâce notamment à la notion de famille de produits, les fabricants peuvent partager certains composants qui deviennent communs à différentes gammes produites. Ces systèmes de fabrication flexibles « FMS » (*Flexible Manufacturing System*) désignent un ensemble de technologies visant à rendre flexible l'ensemble de l'outil de production (Hu 2013). Leur but étant de permettre la réalisation du plus grand nombre possibles d'opérations en engageant le nombre le plus limité de ressources.

Processus décisionnel clé, l'ordonnancement des tâches de production consiste à allouer dans le temps un ensemble d'opérations à différentes ressources. Les approches classiques centralisent cette prise de décision. La planification qui en résulte est par la suite, diffusée verticalement vers les autres composants du système de production. Ces approches hiérarchiques offrent cependant uniquement dans de rares cas (et souvent théoriques), une optimisation sur le long terme. En effet, les environnements industriels étant loin d'être statiques, ces architectures centralisées ne sont que peu ou pas adaptées aux contraintes induites par des évènements dynamiques tels que les arrivées dynamiques de produits ou encore l'occurrence de pannes imprévues.

Afin d'apporter la réactivité indispensable pour faire face efficacement aux perturbations et aux incertitudes, des approches distribuées ont émergées. Bien que ces dernières confèrent davantage d'agilité, la décentralisation des décisions qui est en résulte font qu'elles souffrent de certaines limitations dont la plus importante est, sans doute, le phénomène de « myopie décisionnelle ». Celle-ci désigne le manque de visibilité des entités sur l'impact de leurs décisions sur la performance globale du système. Parmi les approches hétérarchiques, celles qui proposent un pilotage par le produit *(ou Product-Driven Manufacturing)* ont montré qu'elles étaient particulièrement efficaces en termes de réactivité. Néanmoins, nous avons constaté que,

dans la littérature, très peu de travaux proposent l'utilisation de techniques telles que l'apprentissage via simulation couplées aux systèmes pilotés par le produit. Ce constat fut le point de départ de notre réflexion. La Figure 1 met en évidence le positionnement de ce travail de thèse. Il se situe au croisement des problématiques d'optimisation combinatoire de l'ordonnancement, de la conduite des Systèmes de Production Cyber-Physiques (CPPS), et des techniques d'intelligence artificielle distribuées.

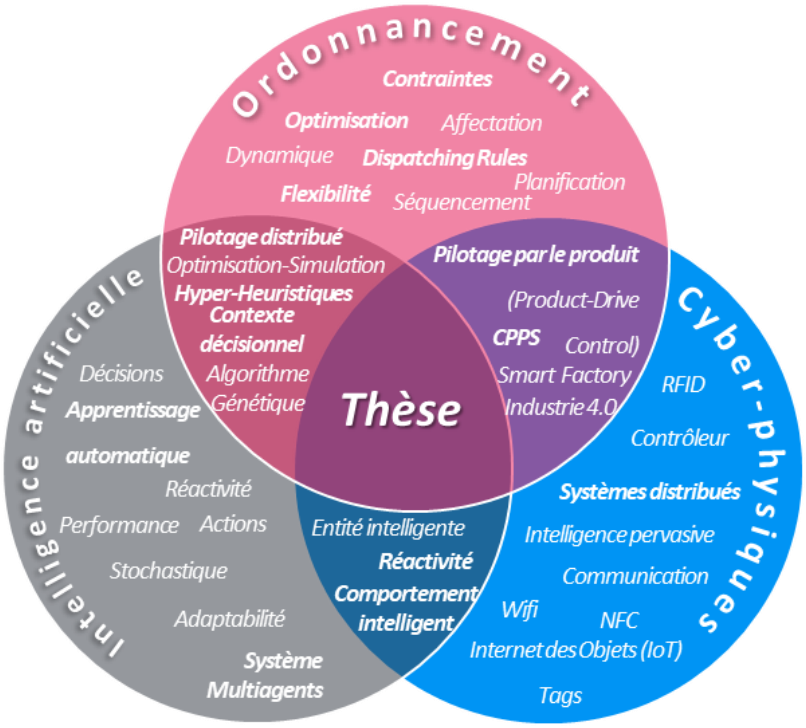


Figure 1 Positionnement de la thèse

2. PROBLEMATIQUE

La présente thèse est l'aboutissement d'un travail de longue haleine. Le mémoire de Magister (Bouazza 2013), intitulé *«Pilotage de cycle de vie à base de produits intelligents : application à l'industrie pharmaceutique»,* achevé en 2013, a permis de dégager une problématique à partir d'un cas industriel réel, à savoir l'ordonnancement dynamique d'ateliers avec « flexibilité partielle ». Celle-ci étant due notamment à des équipements hétérogènes au sein d'une même cellule de production. Ce premier cas démontra l'intérêt des approches pilotées par le produit et fit émerger des interrogations clés auxquelles cette thèse tentera d'apporter des réponses. Les voici :

Peut-on appliquer l'approche de pilotage par le produit à différents types de configurations d'atelier : flowshop flexible, jobshop flexible, etc. ?

L'approche pilotée par le produit peut-elle gérer plus de contraintes : des évènements dynamiques (p. ex. l'arrivée de nouveaux produits dans le système ou la survenance de pannes imprévues), les multiples familles de produits, les contraintes d'interopérabilité (i.e. la flexibilité totale ou partielle de cellules), les différents temps opératoires et temps de configuration ?

Quelle est l'influence du contexte décisionnel et comment peut-il améliorer la pertinence des décisions locales en assurant une bonne performance globale du système ?

Est-il possible d'exploiter la simplicité et la rapidité des règles d'ordonnancement tout en

réduisant le phénomène de myopie, notamment grâce au couplage Optimisation-Simulation ?

Les aspects de la problématique soulevée sont doubles : le premier aspect concerne le problème d'optimisation combinatoire, avec ses nombreuses contraintes, qu'est l'ordonnancement dans les FMS. Le second aspect est lié à la conduite dynamique des CPPS avec les problèmes sous-jacents qu'elle implique, comme la myopie décisionnelle. Toute la difficulté se résume à trouver un équilibre entre la nécessité de réagir rapidement face aux perturbations, et à maintenir l'objectif global de performance du système (Barbosa et al. 2011). De plus, minimiser la myopie n'équivaut pas toujours à maximiser les performances ; un système trop fortement « protégé » contre le comportement myope risque probablement de perdre les avantages d'un contrôle distribué (Blunck et al. 2016).

3. CONTRIBUTIONS

Face à la difficulté que représente la conduite de systèmes de production aussi dynamiques que complexes, une solution distribuée nous apparut comme la plus pertinente. Ce qui distingue les approches décentralisées des approches centralisées, c'est qu'elles sont bien plus agiles et plus réactives. Ainsi la solution proposée se doit d'apporter des réponses rapides et adaptées face à des environnements manufacturiers complexes en perpétuelle évolution.

Pour cela, nous nous sommes fixés comme objectif le développement d'un modèle d'architecture de contrôle hétérarchique pilotée par le produit. Celui-ci se devant d'apporter une optimisation efficace de l'ordonnancement tout en réduisant le phénomène de myopie décisionnelle inhérent à la distribution des nœuds décisionnels. L'approche proposée se veut également aussi générique que possible et applicable à un maximum de configurations d'ateliers différentes. L'aboutissement de nos travaux est une hyper-heuristique basée sur un modèle d'encapsulation des comportements des Produits Intelligents (PI), et la conception puis l'implémentation d'un outil de simulation de système de production (SP) piloté par les PI. L'originalité de cette contribution tient principalement en deux points :

Le premier consiste en la proposition d'un nouveau concept : le « contexte décisionnel » améliorant la pertinence des règles d'ordonnancement sélectionnées par les PI. L'espace de recherche est alors composé d'un ensemble de règles auquel s'ajoutent les différents contextes décisionnels qui conditionnent leur déclenchement. En effectuant le choix des règles en fonction d'une ou plusieurs variables objectives, la contextualisation permet de décomposer les stratégies adoptées par le PI en un schéma particulier d'allocation de règles d'ordonnancements.

Le second consiste en l'intégration d'un mécanisme de couplage Optimisation-Simulation assurant un apprentissage performant des stratégies suivies par les PI. En effet, ce point crucial consiste en l'utilisation de simulations pour réduire le phénomène de myopie en assurant une phase d'apprentissage. La simulation permet l'évaluation du comportement global du SP émergeant d'interactions complexes, dans le but de réduire le phénomène de myopie et le problème d'optimalité locale qui en résulte.

Le développement de la solution se fit de façon incrémentale. Il donna lieu à la création de différents modèles intermédiaires publiés dans différents articles de conférences (Bouazza et al. 2015; Bouazza et al. 2017), jusqu'à la solution, la plus aboutie, publiée dans l'article du journal *Manufacturing Letters* (Bouazza et al. 2019).

4. PLAN DU DOCUMENT

Comme illustré dans la Figure 2 ci-dessous, le présent document est composé de deux parties s'articulant en quatre chapitres :

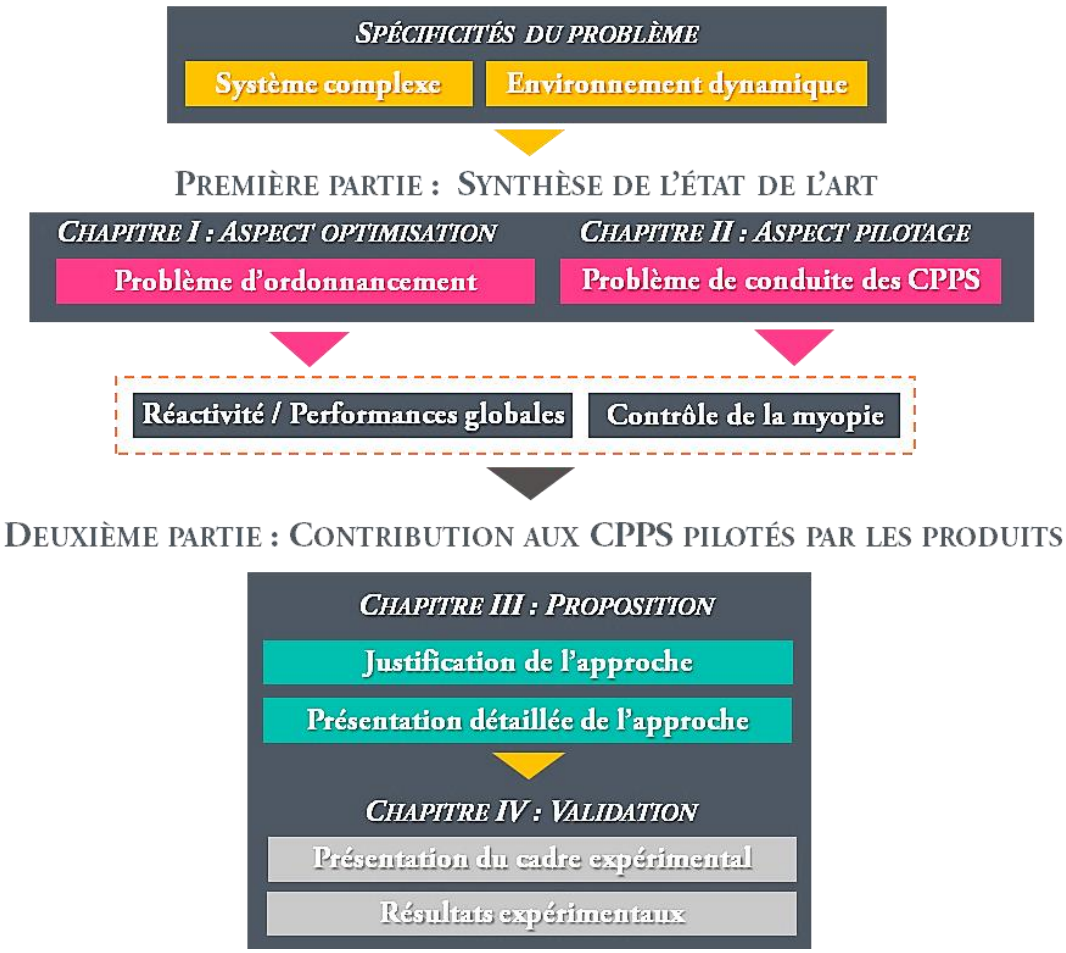


Figure 2 Présentation du plan de la thèse

▪ **PREMIERE PARTIE : SYNTHESE DE L'ETAT DE L'ART**

Cette première partie propose d'aborder les deux aspects de la problématique, à savoir le problème d'ordonnancement dans les FMS et le problème de conduite des CPPS. Pour chacun d'eux, un état de l'art de la littérature est proposé :

– **CHAPITRE I :** le premier chapitre est une introduction aux FMS, et plus précisément à l'ordonnancement dynamique des tâches de production dans ces systèmes particuliers. Une première section présente les différents aspects de la flexibilité des systèmes de production. La seconde section met l'emphase sur l'ordonnancement des tâches de production. Les 3 facteurs α, β et γ décrivant respectivement le type d'atelier de production, les contraintes du problème et les critères d'optimisation, sont successivement détaillés.

– **CHAPITRE II :** le second chapitre, s'intéresse aux fonctions de pilotage des systèmes de production cyber-physiques. Après avoir passé en revue les différentes fonctions du système de pilotage et les différentes architectures de contrôle les spécificités de la conduite des CPPS sont soulignées. Puis le focus est mis sur les « Systèmes de Contrôle pilotés par le Produit » ou « Product-Driven System Control » (PDSC). Les avantages et limites de ces approches sont ensuite explicités.

▪ **DEUXIEME PARTIE : CONTRIBUTION AUX CPPS PILOTES PAR LES PRODUITS**

La seconde partie de la thèse présente les contributions aux CPPS pilotés par les produits. Elle s'articule en deux chapitres :

– **CHAPITRE III** : le troisième chapitre détaille la contribution à l'ordonnancement dynamique d'ateliers partiellement flexibles. L'approche proposée vise à assurer un contrôle hétérarchique par les produits intelligents. L'utilisation d'une hyper-heuristique permet d'obtenir des stratégies décisionnelles efficaces.

– **CHAPITRE IV :** le dernier chapitre est dédié à la validation de l'approche précédente au travers de plusieurs cas d'études inspirés de l'industrie pharmaceutique.

▪ Enfin, une conclusion et un ensemble de travaux prospectifs sont proposés à la fin de ce mémoire.

PREMIERE PARTIE : SYNTHESE DE L'ETAT DE L'ART

CHAPITRE I : ORDONNANCEMENT DYNAMIQUE DES FMS

1. INTRODUCTION

La mise en place de moyens de production toujours plus polyvalents ainsi que l'accroissement de l'automatisation ont permis d'augmenter la flexibilité des systèmes de production (SP). Ce besoin en flexibilité répond à un impératif de réactivité face aux changements, qu'ils soient dus aux produits et aux technologies ou induits par les variations des marchés, mais aussi pour répondre à des besoins plus spécifiques telle la personnalisation de masse (*Mass-Customization*). A cela s'ajoutent les efforts de conception qui ont pour objectif la mise en commun de composants afin de réaliser des économies de gamme (ElMaraghy et al. 2006). Ainsi, les systèmes de production flexibles ou en anglais « *Flexible Manufacturing System* » (FMS), sont utilisés pour créer un maximum de variétés de produits en optimisant le pilotage d'un nombre limité de ressources. L'objectif ultime de ces FMS est de rendre l'ensemble de l'outil de production capable de gérer les incertitudes, qu'elles soient externes ou internes. Cet accroissement de la flexibilité engendre cependant un certain nombre de problèmes, notamment en termes de complexité de planification et de pilotage.

Ce premier chapitre est une introduction aux FMS, et plus précisément à l'ordonnancement dynamique des tâches de production dans ces systèmes particuliers. Une première section présente les différents aspects de la flexibilité des systèmes de production et met en évidence la manière dont elle contribue à faire face aux multiples incertitudes environnementales. La seconde section met l'emphase sur le processus décisionnel clé qu'est l'ordonnancement des tâches de production. Les 3 facteurs α, β et γ, communément utilisés pour décrire les problèmes d'ordonnancement et concernant respectivement le type d'atelier de production, les contraintes du problème et les critères d'optimisation, sont successivement détaillés.

2. FLEXIBILITES DES SYSTEMES DE PRODUCTION

La flexibilité est une notion dont la signification varie fortement en fonction du contexte dans lequel elle est employée. D'après le dictionnaire LAROUSSE[1], elle désigne la qualité de quelque chose qui peut s'adapter à des circonstances particulières. Selon Sethi et al. (1990), la flexibilité d'un système réside dans sa capacité à changer pour faire face à un environnement incertain et/ou changeant. Dans les systèmes de production, elle fait référence à la capacité à réagir ou à se transformer avec des pénalités minimales, que ce soit en termes de temps, de coût ou de performance (Upton 1997). Enfin, pour Hallgren et al. (2009), la flexibilité d'un SP est envisagée comme un « catalyseur » fournissant des propriétés améliorant d'autres capacités concurrentielles (p. ex. une meilleure adaptation à la variabilité de la demande ou une meilleure résilience aux pannes).

[1] https://www.larousse.fr/dictionnaires/francais/flexibilité/34138

2.2. LES BESOINS EN FLEXIBILITE

Une entreprise peut décider d'améliorer sa flexibilité pour diverses raisons, chacune amenant un besoin de flexibilité particulier. Malgré la grande diversité de ces facteurs, la flexibilité d'un système manufacturier demeure intrinsèquement un mécanisme permettant de faire face aux incertitudes de l'environnement, qu'il soit interne ou externe (Sawhney 2006). Selon Chen et al. (1991), les causes entrainant des besoins de flexibilité au sein d'une entreprise sont de deux ordres (Tableau 1) :

Les incertitudes environnementales : il s'agit des éléments provenant du macro-environnement d'une entreprise généralement liés à son marché et à sa position concurrentielle. Une entreprise peut être amenée à chercher à accroitre sa flexibilité, par exemple, dans le but de diversifier son catalogue de produits, contrer la concurrence en jouant sur les volumes, ou encore afin de s'inscrire dans une démarche de *« Mass Customisation »* qui vise à combiner une production sur mesure avec de faibles coûts de production unitaires.

Les incertitudes internes : il s'agit des éléments généralement liés au processus de fabrication. La flexibilité devient alors un outil permettant de gérer au mieux l'indisponibilité des machines (p. ex. liée aux pannes ou aux travaux de maintenance), faire face aux ruptures de stock (qu'elles soient d'origine interne ou résultant d'un problème d'approvisionnement externe), ou encore pour s'adapter aux problèmes de main d'œuvre.

Types	Facteurs	Exemples de sources
Incertitudes environnementales	Diversification du catalogue produit	Options, différenciation
	Variation de la demande (saisonnière ou aléatoire)	Production unitaire
	Cycle de vie du produit/de la technologie plus court	Produits nouveaux ou modifiés
	Augmentation de la personnalisation	Commandes sur mesure
	Raccourcissement des délais de livraison	Commandes urgentes
Incertitudes internes	Disponibilité des machines	Panne-machine
	Entrée des matières premières	Echecs du contrôle-qualité
	Approvisionnements	Ruptures de stock
	Variation de la main d'œuvre	Absentéisme

Tableau 1 Facteurs liés aux besoins en flexibilité

2.3. TYPOLOGIE DES FLEXIBILITES

Dans la littérature, il existe différentes typologies et classifications des flexibilités des systèmes de production. Toutefois, comme souligné par Pérez et al. (2016), il est fréquent que deux termes différents désignent deux concepts très similaires, voir identiques, entrainant une certaine ambiguïté. Même si aucune classification ne fait totalement consensus, nous retiendrons une typologie proche de celle proposée par Sethi et al. (1990) qui distinguent onze types de flexibilités réparties en trois catégories (Figure 3) :

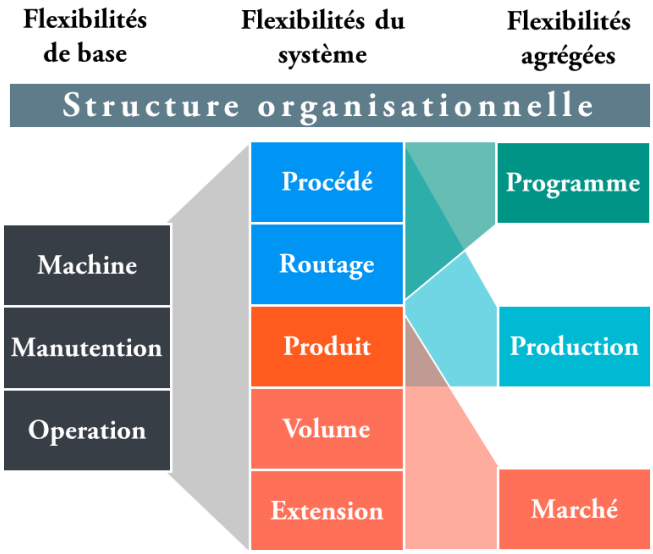


Figure 3 Liens entre les différentes flexibilités (Sethi et al. 1990)

2.3.1. FLEXIBILITES DE BASE OU DES COMPOSANTS

Flexibilité « machine » : fait référence aux passage d'une opération à une autre par une machine sans nécessiter d'effort excessif en temps, en coût ou en performance (Fanjul-Peyro et al. 2017). P. ex. via l'utilisation de machines ou de robots à outils interchangeables (Figure 4).

Figure 4 Exemple de robot industriel à outils interchangeables

Flexibilité de manutention : désigne la capacité d'un SP à déplacer les différentes pièces ou composants (Eloundou et al. 2015) et concerne aussi bien les opérations de chargement et de déchargement, de convoyage, que de stockage (Sarin et al. 1987; Stecke et al. 1981) (Figure 5).

Figure 5 Exemples de systèmes de manutention automatisés

Flexibilité d'opération : mesure la capacité du SP à modifier la séquence des opérations qui composent le processus industriel (Kaschel et al. 2006). Elle dépend du nombre de séquences différentes possibles pour la fabrication de chaque pièce (Phanden et al. 2015).

2.3.2. FLEXIBILITES DU SYSTEME

Flexibilité du processus : est définie par l'ensemble des types de pièces et de composants que le système peut produire sans (re)paramétrage important (Seok Shin et al. 2011).

Flexibilité de routage : désigne la capacité du SP à produire des trajectoires et chemins alternatifs (Joseph et al. 2011). Elément majeur de la flexibilité d'un SP, la flexibilité de routage dépend de l'existence de processus ou de machines alternatifs pour une même tâche (Eloundou 2016). Elle ne doit cependant pas être confondue avec la flexibilité d'opération, qui est indépendante des trajectoires dans le système, ni avec la flexibilité de la manutention, qui est une propriété de composants spécifiques du système (Melki et al. 2010).

Flexibilité du produit : représente la facilité (en termes de temps et de coût) avec laquelle de nouveaux produits peuvent être créés grâce à l'utilisation de composants existants. En d'autres termes, c'est la facilité avec laquelle le mélange de composants peut être modifié rapidement et à moindre coût en cours de production. La flexibilité du produit contribue à l'agilité du SP par la facilité d'introduction de nouveaux produits dans une gamme de produits existante (ElMaraghy et al. 2006).

Flexibilité du volume : correspond à la capacité du SP à répondre de manière rentable à des objectifs de niveau de production qui peuvent fluctuer. La disponibilité des ressources est cruciale afin de permettre une réponse en termes de quantités produites (Jack et al. 2002; Hallgren et al. 2009).

Flexibilité d'extension : désigne la facilité de remplacement ou d'ajout d'outils de production. Elle résulte souvent d'un effort de conception dès l'origine. La flexibilité d'extension inclut le coût nécessaire à la reconfiguration, mais aussi la rapidité avec laquelle l'extension du SP est réalisée (Karsak et al. 2002; Kumar 1995).

2.3.3. FLEXIBILITES AGREGEES

Flexibilité du programme de contrôle : bien que relativement imprécis, ce concept traduit la capacité d'un SP à fonctionner avec le moins d'interruptions possible et ce, de la façon la plus autonome possible. Cette flexibilité peut être atteinte grâce à l'utilisation de machines « intelligentes » et de logiciels de contrôle/pilotage efficaces, par exemple grâce à un ordonnancement en temps réel ou *Real-Time Scheduling* (RTS) (Ham et al. 2011; Park et al. 2014), à des approches prédictives (Zhang et al. 2014; Cardin et al. 2017) et/ou robustes et résilientes (Deep et al. 2015; Zhang et al. 2016) et, plus généralement, grâce à l'autonomie et à l'efficience des systèmes de pilotage (Yamamoto et al. 2001; Park et al. 2011)**.**

Flexibilité de la production : correspond à l'ensemble des produits pouvant être fabriqués sans ajout majeur d'équipements. Des ressources mineures telles que de nouveaux outils peuvent néanmoins être autorisées selon le cas. Elle doit être anticipée dès la phase de conception des produits (Kumar et al. 2009; Bateman et al. 2006), ainsi que durant les études technico-économiques (Schoute 2011; Eckel et al. 2010).

Flexibilité du marché : désigne la facilité avec laquelle le SP peut s'adapter à l'évolution des marchés. Elle résulte de la synergie entre les fonctions de production, celles de commercialisation et de marketing. Cette flexibilité correspond, par exemple, à la fabrication à la demande (Adamo et al. 2016), aux approches dites centrées sur le client/consommateur (Tu et

al. 2004; Tseng et al. 2003), ou encore aux approches pilotées par le marché (Ferguson et al. 1996; Baker 1991).

2.4. LES SYSTEMES DE PRODUCTION FLEXIBLES (FMS)

Contrairement aux systèmes de production dédiés (DMS) plus classiques, les systèmes flexibles et/ou reconfigurables sont utilisés pour créer une grande variété de produits, notamment par assemblage combinatoire. La notion de famille de produits permet aux fabricants de partager certains composants communs entre produits. Ainsi, les systèmes de production flexibles ou *Flexible Manufacturing System* (FMS), désignent un ensemble de technologies qui visent à rendre flexible l'ensemble de l'outil de production (Hu 2013). Le but premier est de permettre la réalisation du plus grand nombre d'opérations possibles sur un nombre limité de ressources. Pour Mehrabi et al. (2002) les FMS consistent en une configuration de système manufacturier qui comporte des logiciels permettant de gérer les modifications apportées aux ordres de fabrication, aux programmes de production et aux outils. Les FMS sont différents, tout en étant néanmoins très proches, du concept de systèmes de production reconfigurables ou RMS (*Reconfigurable Manufacturing System*) (Figure 6). Ces derniers sont conçus par l'intégration de modules (matériels et logiciels) pouvant être réorganisés ou remplacés rapidement et de manière fiable pour répondre au plus juste à la fluctuation de la demande (Koren 2006; Koren et al. 1999; ElMaraghy et al. 2006).

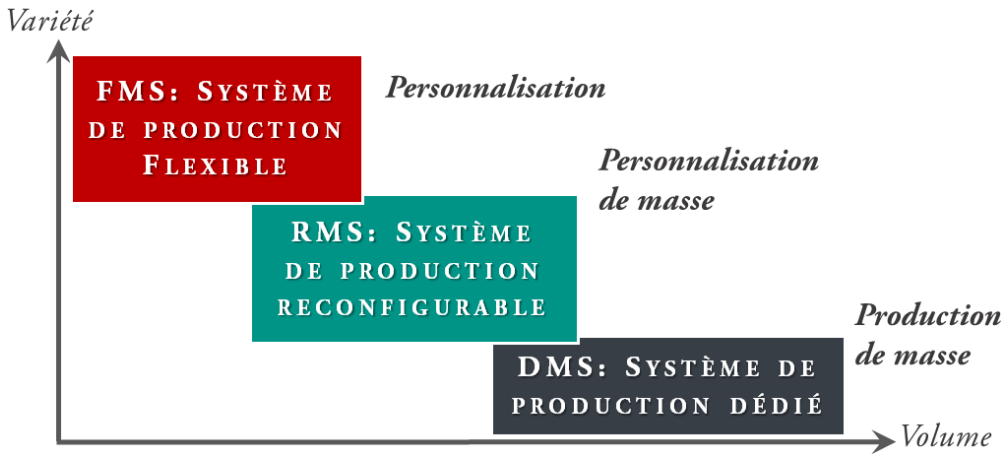


Figure 6 Classification des systèmes de production selon Variété/Volume (Mehrabi et al. 2002)

Le plus souvent, les FMS sont caractérisés par une flexibilité de routage et/ou par la possibilité d'utiliser différentes méthodes de fabrication pour les mêmes produits. L'utilisation des FMS présente plusieurs avantages. D'une part, ils permettent la production de petits lots de produits hautement personnalisés. Et, d'autre part, ils permettent à l'entreprise de réagir rapidement aux fluctuations des marchés. Les FMS contribuent ainsi à assurer un niveau de productivité élevé dans une environnement de pression concurrentielle toujours plus forte (Godinho Filho et al. 2014).

Cette augmentation de la flexibilité (notamment « machine » et de routage) engendre cependant un accroissement de la complexité en termes d'organisation et de planification des opérations. La section suivante traite de la formulation du problème d'ordonnancement dans les systèmes de production flexibles.

3. PROBLEMES D'ORDONNANCEMENT « α|β|γ »

L'ordonnancement industriel ou en anglais « *Manufacturing Scheduling* », est un processus décisionnel qui concerne l'affectation de ressources à des tâches sur des périodes de temps données , tout en optimisant une ou plusieurs fonctions « objectif » (Pinedo 2016). Chaque tâche *ou Job*, implique soit une entité physique indivisible appelée « produit », soit un « lot » lorsque plusieurs produits identiques sont regroupés (Lopez et al. 2010). Selon le nombre de machines candidates pouvant réaliser l'opération, le problème d'ordonnancement peut être décomposé en deux sous-problèmes que sont l'allocation de la machine et le séquencement des opérations (Chaudhry et al. 2016; Trentesaux 2013). L'allocation des machines consiste à allouer une machine au traitement de chaque opération, tandis que le séquencement des opérations consiste à organiser l'ordre dans lequel ces opérations vont être exécutées (Gao et al. 2015).

Dans notre démarche de présentation des problèmes d'ordonnancement mis en évidence dans la littérature, nous utiliserons la notation étendue dans (Allahverdi 2015; Allahverdi et al. 2009) basée sur les travaux de Graham et al. (1979). Un triplet de champs est utilisé et noté « α|β|γ » : le premier champ « α » décrit le type d'atelier ainsi que sa configuration, le second champ « β » précise les contraintes, et le dernier champ « γ » définit le ou les critères de performance à optimiser. Il est à noter que les deux derniers champs peuvent contenir plusieurs entrées (p. ex. les contraintes multiples et l'optimisation multi-objectifs).

3.1. TYPE D'ATELIER « α »

Dans la littérature scientifique comme dans l'industrie, il existe une grande variété d'ateliers de production. Beaucoup d'éléments peuvent varier, par exemple, le nombre de machines, leur organisation (en parallèle, en série, en ilot, etc.) ou encore les procédés. Ainsi, les différentes machines et ressources peuvent être organisées en une ou plusieurs étapes ou niveaux (*multi-stage* en anglais*)*. Les principaux types d'ateliers sont listés ci-après :

▪ **Machine unique (1) :** l'environnement à machine unique, bien que simple de prime abord, constitue selon Pinedo (2016), un cas particulier qui se distingue des autres types d'environnements. Cette importance est d'autant plus grande que les modèles associés ont souvent des propriétés que ne possèdent pas ceux à plusieurs machines (qu'elles soient en parallèle ou série). L'auteur explique qu'une seule ressource « goulet », peut donner naissance à un modèle à machine unique.

▪ **Machines en parallèle :** le problème de planification dans des ateliers composés de machines parallèles est étudié depuis de nombreuses années. Dans ce type de problème, un job « *j* » nécessite une seule opération et peut être traité sur n'importe quelle machine parmi « *m* » ou sur une machine appartenant à « *n* », un sous-ensemble de *m* donné. Dans (Cheng et al. 1990), ce type de problème est divisé en trois catégories illustrées dans la Figure 7 :

– ***Machines parallèles identiques (P) :*** les machines sont répliquées à l'identique et semblables en tout point : vitesse, temps de configurations, etc. (Huang et al. 2011; Jans 2009).

– ***Machines parallèles uniformes (Q) :*** les machines sont considérées comme uniformes si leur vitesse de fonctionnement diffèrent, mais demeurent identiques sur tous les autres points dont notamment l'interopérabilité (K. Li et al. 2014; Tan et al. 2010).

– ***Machines parallèles hétérogènes (R) :*** cette configuration apparaît en industrie généralement quand des équipements hétérogènes coexistent dans l'atelier (Fanjul-Peyro et

al. 2017; Kim et al. 2002). Par exemple, deux machines à commande numérique, l'une capable de couper l'aluminium et l'acier, et une seconde, ne pouvant traiter que l'aluminium. Notons que cette différenciation peut apparaitre temporairement, par exemple en cas d'indisponibilité d'un outillage, de panne partielle ou encore de performance dégradée.

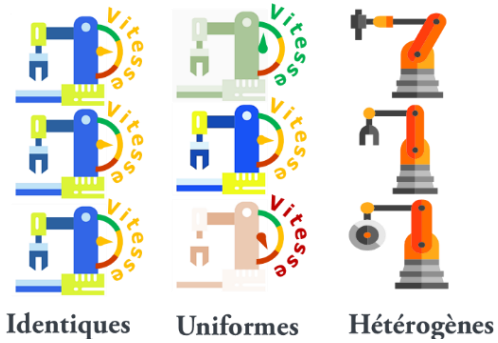


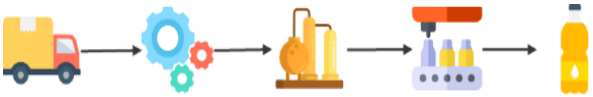
Figure 7 Exemple d'ateliers à machines parallèles

▪ **Ateliers à cheminement unique « flowshops » :** les ateliers à cheminement unique aussi appelés « flowshops », sont constitués de plusieurs machines organisées en ligne de production et regroupées en niveaux. Différentes variantes existent (Figure 8), en fonction notamment du nombre de machines par niveau et des différentes séquences possibles (p. ex. les flexibilités machine, de routage et d'opérations).

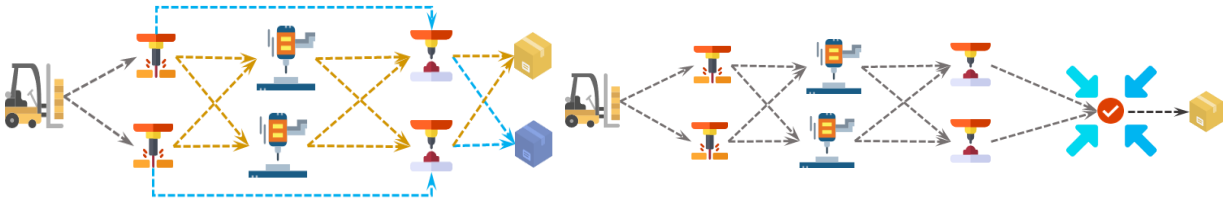
– ***Flowshop classique (Fm) :*** les flowshops classiques sont constitués de *m* niveau avec une seule machine par niveau. Tous les produits passent par tous les niveaux et dans le même ordre (Aggoune 2004; Wang et al. 2005). Dans les problèmes d'ordonnancement notés FSP (*Flowshop Scheduling Problem*), il n'y a qu'un problème de séquencement à résoudre, compte tenu du fait qu'une seule machine est possible pour chaque opération.

– ***Flowshop flexible hybride (FFm) :*** parmi les variantes du flowshop, il y a les flowshops hybrides flexibles. Le terme "hybride" fait référence à la présence de machines parallèles à certains niveaux. A la différence des FSP, une machine peut exister en plusieurs exemplaires identiques fonctionnant en parallèle (Boukef 2011). Le terme « flexible » signifie que le saut de certains niveaux est admissible sous certaines conditions, un produit ne visitant pas nécessairement l'ensemble des niveaux. Ce problème d'ordonnancement est fréquemment rencontré dans la production par lot (Gagné et al. 2014; Gupta et al. 2002).

– ***Flowshop d'assemblage (AFm) :*** ce type de ligne de production peut être considéré comme un cas particulier de flowshop flexible. En effet, la différence réside dans le dernier niveau de production qui est une phase d'assemblage. Celle-ci peut être composée d'une machine unique (Seidgar et al. 2014) ou de plusieurs machines identiques (Mozdgir et al. 2013).



Flowshop classique (Fm)



Flowshop flexible hybride (FFm) Flowshop d'assemblage (AFm)

Figure 8 Exemples d'atelier flowshop et ses variantes

Les problèmes d'ordonnancement associés sont notés de la manière suivante : FSP (Flowshop Scheduling Problem), FSP (Flexible Flowshop Scheduling Problem), ou encore HFSP (Hybrid Flexible Flowshop Scheduling Problem).

▪ **Ateliers à cheminements multiples « jobshops » :** les ateliers jobshops, appelés également ateliers à cheminements multiples sont des ateliers dans lesquels les opérations sont réalisées selon un ordre bien déterminé ; cette séquence d'opérations varie d'un produit à l'autre, ou le plus souvent, d'une famille de produits à l'autre (Figure 9).

– ***Jobshop classique (J) :*** le problème d'ordonnancement de jobshop classique noté JSP (*Jobshop Scheduling Problem)*, ne nécessite que le séquencement des opérations (tout comme pour le FSP) ; compte tenu du fait que les machines et les opérations qui leurs sont allouées sont fixées au préalable.

– ***Jobshop flexible (FJ) :*** le jobshop flexible est une extension du modèle jobshop classique ; sa particularité réside dans le fait que plusieurs machines sont potentiellement capables de réaliser un sous-ensemble d'opérations (Boukef 2011). Ainsi dans le FJSP (*Flexible Jobshop Scheduling Problem*), l'allocation des machines aux opérations n'est pas fixée à l'avance. On retrouve alors les deux sous-problèmes communs de l'ordonnancement : l'allocation et le séquencement. De plus, selon (Kacem et al. 2002), le FJSP peut, lui également, être décomposé en deux problèmes selon le degré de flexibilité de l'atelier :

  - *P-FJSP (Partitailly Flexible jobshop Scheduling Problem) :* au moins une opération ne peut pas être traitée sur toutes les machines.

  - *T-FJSP (Totally Flexible jobshop Scheduling Problem) :* chaque opération peut être traitée sur toutes les machines de l'atelier.

  Le T-FJSP est considérée comme un cas particulier du P-FJSP. Par conséquent, le P-FJSP est plus difficile à résoudre que le T-FJSP à facteurs identiques (Sullivan et al. 2009; Driss et al. 2015).

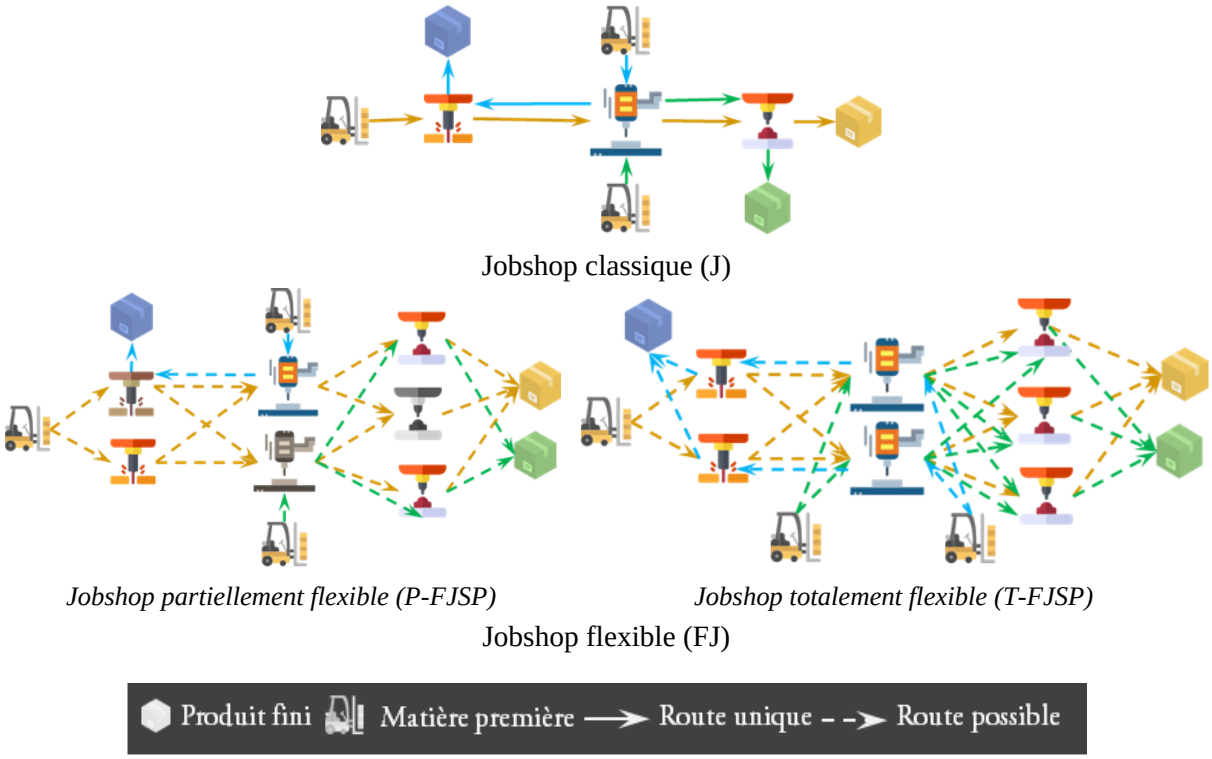


Figure 9 Exemples d'atelier jobshop et ses variantes

▪ **Ateliers à cheminements libres « openshops » (O) :** ce type d'atelier est moins contraint que les flowshops et jobshops. Ainsi, l'ordre des opérations n'est pas fixé a priori ; le problème d'ordonnancement consiste, d'une part, à déterminer le cheminement de chaque produit et, d'autre part, à ordonnancer les produits en tenant compte des gammes. Par rapport aux autres modèles d'ateliers, l'openshop est assez rarement utilisé dans les entreprises (Boukef 2011), et par extension dans l'industrie (Bai et al. 2016; Alcaide et al. 2006). Ce type d'atelier est plus proche de certains cas rencontrés dans le domaine hospitalier où différents examens peuvent éventuellement être effectués indépendamment de l'ordre (Fei et al. 2010; Paulussen et al. 2006).

3.2. CONTRAINTES « β »

En plus du type d'atelier indiqué dans le champ « α », il est nécessaire d'identifier clairement les différentes contraintes pouvant potentiellement complexifier les décisions liées à l'ordonnancement. Pour ce faire, le second champ nommé « β », peut contenir plusieurs entrées. Nous distinguons ci-après les contraintes liées aux ressources de celles liées aux produits.

3.2.1. CONTRAINTES PRODUITS

▪ **Familles de produits (fmls) :** cette contrainte implique que chaque produit appartient à une famille spécifique. Ces groupes sont définis comme un ensemble de produits partageant les mêmes caractéristiques de procédé (p. ex. les classes de médicaments dans l'industrie pharmaceutique ou les types d'huile dans l'industrie pétrochimique). Cette contrainte est très importante, voire cruciale puisqu'elle peut impliquer d'importantes variations des temps opératoires (Bokhorst et al. 2008), des temps de configuration (Chen et al. 2013), ou encore des incompatibilités entre familles de produits (Omar et al. 2006).

▪ **Précédence (prec) :** la contrainte de précédence est en général liée à la cohérence technologique. Elle décrit l'ordre relatif de certaines tâches par rapport à d'autres (Lopez et al.

2010). Elle implique de réaliser une ou plusieurs tâches avant de permettre à une autre de commencer. Cet impératif est notamment dû à l'ordre technique lié aux spécificités du procédé de fabrication. Dans certains cas, une tâche ne peut commencer que si toutes les tâches qui doivent lui précéder sont achevées. Dans d'autres, il suffit d'une partie pour autoriser son lancement. On parle alors de modèle de type *« AND/OR »* (Möhring et al. 2004; Moore et al. 2001).

▪ **Dates de libération (r_j) :** en anglais *« release dates »* elles correspondent aux dates « au plus tôt » pour débuter des tâches. La date de libération est l'heure à laquelle le produit arrive dans le système et peut potentiellement commencer son traitement (Arroyo et al. 2019). Cette contrainte peut concerner un produit unique (Lu et al. 2014), ou l'ensemble d'un lot (Xiong et al. 2017).

▪ **Autres contraintes produit :** il existe de très nombreuses autres contraintes liées aux produits. Parmi lesquelles nous citerons :

– ***Echéances (D_j) :*** en anglais *« Due Dates »*, elles désignent les instants où les travaux doivent être terminés de manière idéale (p. ex. les dates de livraison convenues avec les clients). La dernière opération doit alors être achevée avant la date d'échéance. Habituellement, les dates d'échéance ne sont pas complètement rigides et un dépassement est toléré. Mais le plus souvent, ce retard implique une pénalité ou un coût supplémentaire (p. ex. la réduction des gains ou la baisse de la satisfaction client).

– ***Préemption (prmp) :*** dans le cas de tâches préemptives, il est possible d'interrompre le traitement, et de le reprendre par la suite pour achever le reste de l'opération (Haleh et al. 2017). Il est par exemple possible d'interrompre le fraisage d'une pièce à moitié et de terminer la moitié restante plus tard.

– ***Traitement par lots (batch) :*** une machine peut être en mesure de traiter un certain nombre de tâches simultanément regroupées par lot de produits ; c'est-à-dire qu'elle peut traiter l'ensemble du lot de produits en même temps (p. ex. un lot de vêtements dans l'industrie textile ou un lot de produits dans l'industrie agroalimentaire). Le temps d'achèvement d'un lot correspond au moment où la dernière tâche du dernier élément du lot est achevée (Li et al. 2008).

3.2.2. CONTRAINTES RESSOURCES

▪ **Temps/coûts de configuration (setup) :** dans l'industrie, en plus du temps opératoire, une ressource peut nécessiter une opération de configuration supplémentaire. Elle peut avoir lieu antérieurement au traitement de la tâche (p. ex. pour adapter des outils ou préchauffer une machine), ou après une opération (p. ex. le nettoyage ou la stérilisation). L'intérêt de considérer explicitement ces opérations est relativement récent dans la littérature (Weng et al. 2001). Cette configuration peut impliquer un temps ou « *Setup Time* » (ST), mais également un coût ou *« Setup Cost »* (SC). De plus, la durée de ces configurations, et/ou les coûts qu'elles engendrent peuvent varier. Les travaux de (Allahverdi et al. 1999; Allahverdi et al. 2008) proposent de classer les temps/coûts de configuration selon qu'ils soient dépendants ou indépendants de la famille de produits. Pour chaque classe, on distingue également deux sous-classes selon que les temps/coûts soient dépendants ou indépendants de la séquence (Kianfar et al. 2012; Rossi 2014). Une autre variante a également été introduite dans (Koulamas et al. 2008) et concerne la variation de la durée des opérations de configuration dans le temps. On parle alors de dépendance à la séquence passée. Les trois classes relatives au setup sont détaillées ci-après.

– ***Dépendant de la famille :*** le temps/coût de configuration est dit dépendant de la famille si

sa valeur varie selon la famille de produits. De plus, selon la séquence d'opérations, il y a deux sous-catégories :

- *Dépendant de la séquence :* si la contrainte concerne les temps de configuration, elle est notée « $ST_{SD,F}$ ». Si elle concerne les coûts, elle est alors notée «$SC_{SD,F}$» (Liou et al. 2013; Monkman et al. 2008)
- *Indépendant de la séquence :* si le temps de configuration est indépendant de la séquence, la notation « $ST_{SI,F}$ » est utilisée. Si ce sont les coûts qui sont impactés, la notation est «$SC_{SI,F}$» (Wu et al. 2008; Schaller et al. 2013).

– ***Indépendant de la famille :*** le temps/coût de configuration est dit indépendant de la famille si sa valeur ne varie pas selon la famille de produits. Là aussi, on distingue deux sous-catégories :

- *Dépendant de la* séquence : si la contrainte concerne les temps de configuration, elle est notée « ST_{SD} », et «SC_{SD}» si c'est un coût (Ebrahimi et al. 2014; Q. Li et al. 2014).
- *Indépendant de la séquence :* notée « ST_{SI} » si c'est un temps indépendant de la séquence (Agrawal et al. 2012; Gupta et al. 2013), et « SC_{SI} » si c'est un coût indépendant de la séquence.

– ***Dépendant de la séquence passée :*** Koulamas et al. (2008) ont introduit le concept de temps de configuration dépendant d'une séquence passée *ou Past-sequence dependent setup time/cost*. Le temps/coût de configuration est proportionnel à la somme des temps de traitement des travaux déjà effectués. Le temps/coût de configuration peut ainsi diminuer ou augmenter suivant différents phénomènes, par exemple, la réduction du temps de configuration grâce à un phénomène d'apprentissage (Mani et al. 2011), ou au contraire son augmentation à cause d'un phénomène de fatigue de l'opérateur (Biskup et al. 2008). Selon sa nature, la contrainte est notée « ST_{PSD} » ou « SC_{PSD} » (Soroush 2013).

▪ **Eligibilité machine (M_j) :** cette contrainte implique que toutes les machines ne sont pas capables de traiter n'importe quelle tâche ; chaque tâche a un sous-ensemble de machines prédéfini sur lequel elle peut être traitée. Cet ensemble est appelé ensemble éligible (Lee et al. 2011). Dans les SP, cette contrainte est souvent due à la modernisation des outils de production. Ainsi, plusieurs machines aux caractéristiques différentes (souvent améliorées au fur et à mesure), peuvent coexister au sein d'un même atelier (Arroyo et al. 2019). Ainsi, les coûts de production différent parfois selon la machine et ne dépendent pas toujours exclusivement des durées de traitement (Wang et al. 2018). Cette contrainte peut concerner des produits uniques (Nguyen et al. 2017), aussi bien que des lots (Li 2017).

▪ **Panne machine (brk) :** elle implique qu'une machine peut être indisponible pendant un certain temps. Ces périodes d'indisponibilité peuvent être connues (p. ex. dans le cas d'une maintenance planifiée dans le cadre d'une maintenance préventive), ou inconnue (p. ex. des pannes aléatoires). Aussi désignée comme indisponibilité machine, la panne machine constitue l'interruption la plus couramment constatée dans la planification d'ateliers flexibles (Sun et al. 2014). Lors d'une panne machine, l'opération en cours est suspendue. Lorsque l'incident est terminé, deux cas se présentent : soit l'opération peut continuer (la tâche est dite récupérable ou *« resumable »*), soit l'opération recommence depuis le début (la tâche est dite alors irrécupérable ou *« non-resumable »*) (Lei 2011).

▪ **Autres contraintes ressources :** comme pour les contraintes liées aux produits et matières premières, de nombreuses autres contraintes liées aux ressources existent, dont les suivantes :

- ***Blocage (block) :*** ce phénomène peut se produire dans des ateliers aussi bien de type flowshop (Zhang et al. 2018) que de type jobshop (Hansmann et al. 2014). Il survient lorsqu'un stock intermédiaire (ou tampon) entre deux machines consécutives arrive à saturation. Dans ce cas, la machine en amont ne peut libérer une tâche même si celle-ci est achevée.
- ***No-wait (nwt) :*** ce phénomène peut se produire si les jobs ne sont pas autorisés à attendre entre deux opérations successives (Allahverdi 2016; Allahverdi et al. 2014). Par exemple, un laminoir d'acier dans lequel le bloc de métal ne doit pas attendre après l'opération de chauffe qui précède. En effet, le bloc se refroidirait et ne serait plus façonnable.
- ***Recirculation (rcrc) :*** cette contrainte peut apparaitre lorsqu'une tâche peut visiter une machine ou un poste de travail plus d'une fois (Boudhar et al. 2010; Ahonen et al. 2017). Par exemple, un traitement chimique qui doit être effectué plusieurs fois sur un même produit à différentes étapes de son processus de fabrication.

3.3. INDICATEURS DE PERFORMANCE « γ »

Dans les problèmes d'ordonnancement, différents indicateurs peuvent constituer l'objectif de l'optimisation. Nous les classons ici en 5 catégories selon le critère de base. En effet, à partir d'un même critère, il est possible de calculer différents indicateurs de performance, par exemple, la valeur maximale, la valeur moyenne, ou encore la moyenne pondérée.

▪ **Temps d'achèvement (C_j) :** le temps d'achèvement d'un job *j* est communément noté C_j. A partir de ce critère, différents indicateurs de performance peuvent être calculés. Le C_{max} aussi appelé *Makespan*, est le plus répandu dans la littérature et correspond au temps auquel le dernier job a achevé sa dernière opération. Toutefois, dans certains cas, notamment lors d'arrivées dynamiques de produits, il peut être plus approprié de se concentrer sur le temps d'achèvement moyen (TAM), aussi noté $\bar{c}$ ou MCT *(Mean Complétion Time)*. En effet, dans de nombreux domaines tels que l'électronique ou l'industrie pharmaceutique, minimiser le TAM peut permettre de réduire les encours et/ou d'augmenter la satisfaction client (Aydilek et al. 2012). De plus, parmi les moyens permettant de mesurer la variation des temps d'achèvement, un indicateur consiste à mesurer les différences absolues entre les paires des temps d'achèvement. Cette mesure notée *TADC (Total Absolute Differences in Completion Time)* correspond au total des différences absolues des temps d'achèvement (Kanet 1981; Huang et al. 2011).

▪ **Flowtime (F_j) :** aussi appelé temps de transit et noté F_j, il correspond au temps passé par le produit dans le système. Si la date de libération de tous les produits est égale à zéro, le flowtime est égal au temps d'achèvement $F_j=C_j$. La minimisation du temps d'achèvement (qu'il soit total ou moyen, pondéré ou non) est équivalant à la minimisation du flowtime (Ribas et al. 2015).

▪ **Temps d'attente (w_j):** le temps d'attente d'un job noté w_j, correspond au temps cumulé par le produit dans le système à attendre la disponibilité des ressources, c'est-à-dire hors opération de production. Dans certains cas, il peut être important d'axer l'optimisation sur ces temps d'attente (Bagchi 1989).

▪ **Configurations :** les opérations de configuration peuvent avoir une grande importance dans les problèmes d'optimisation de l'ordonnancement (cf. I-3.2.2). On peut alors chercher à optimiser leur nombre *TNS* (*Total number of setups*), leur durée totale *TST* (*Total Setup Time*), mais également leur coût *TSC* (*Total Setup Cost*).

▪ **Date d'échéance (D_j) :** la date d'échéance notée D_j, peut avoir une incidence importante et

souvent négative, sur la performance du SP. Ainsi, selon les cas et en fonction de cette contrainte (voir Figure 10), on peut calculer trois principaux types d'indicateurs :

- ***Précocité (E_j) :*** en anglais « *earliness* », il peut arriver dans la pratique qu'un produit soit achevé avant sa date d'échéance. Cette précocité, engendre une pénalité (p. ex. des coûts de stockage ou d'encours ou des produits périssables). On peut alors chercher à minimiser la précocité maximale E_{max}, cumulée $\sum w_j E_j$ ou encore le coût des pénalités $\sum h(E_j)$.
- ***Lenteur (T_j) :*** la lenteur d'un job ou Tardiness, est notée T_j. Elle concerne les travaux qui se terminent au-delà de leur date échéance. Elle ne peut être négative et les jobs qui sont achevés avant ne sont pas considérés (Framinan et al. 2014).
- ***Retard (L_j) :*** tout comme la fonction de lenteur, le retard noté L_j représente la différence entre la date d'achèvement d'un produit et la date buttoir ou échéance de ce dernier ; à la différence que le retard peut avoir une valeur négative. Ainsi, dans ce cas, le fait de produire en avance est considéré comme souhaitable.

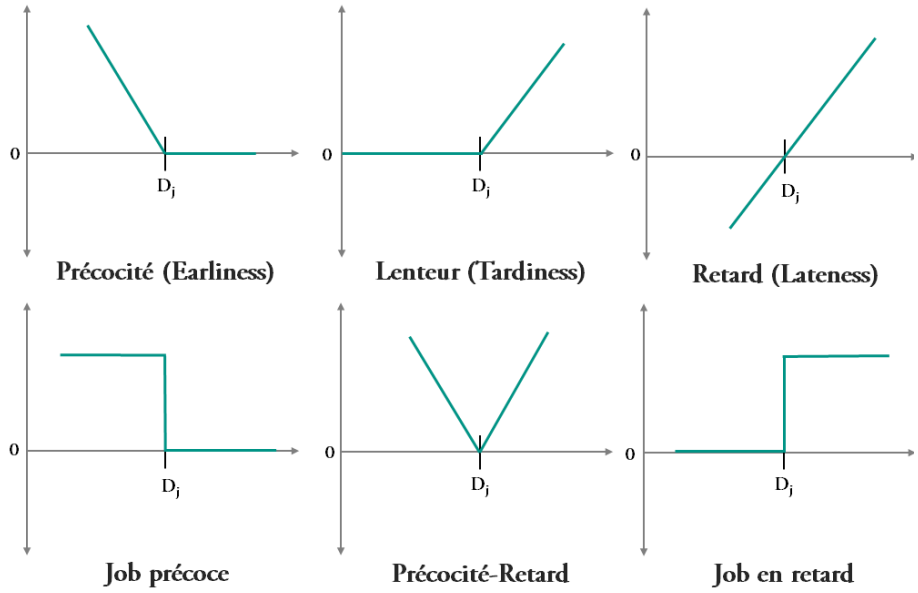


Figure 10 Fonctions dépendantes de la date d'échéance (Framinan et al. 2014)

La Table 2 détaille l'ensemble des notations associées aux trois champs α|β|γ introduites.

	Notation	Signification
Champ α	1	Machine unique
	P	Machines parallèles identiques
	Q	Machines parallèles homogènes
	R	Machines parallèles hétérogènes
	Fm	Flowshop classique
	FFm	Flowshop flexible à plusieurs étapes ou niveaux *(m-stage flexible flowshop)*
	J	Jobshop classique
	FJ	Jobshop flexible
	O	Atelier libre « openshop »
Champ β	fmls	Famille de produit
	prec	Contrainte de précédence
	ST_{SD} / SC_{SD}	Temps/coût de configuration dépendant de la séquence
	ST_{Si} / SC_{Si}	Temps/coût de configuration indépendant de la séquence
	ST_F / SC_F	Temps/coût de configuration dépendant de la famille de produits
	M_j	Contrainte d'interopérabilité
	r_j	Date de libération
	D_j	Date d'échéance
	brk	Pannes
Champ γ	C_{max}	Makespan Temps d'achèvement maximal
	$\sum C_j$	Total des C_j *(Total Completion Times)* aussi noté TCT
	$\sum w_j C_j$	Total pondéré des C_j *(Total Weighted Completion Time)* aussi noté TWCT
	TAM	Temps d'achèvement moyen *(Mean Complétion Time)* aussi noté $\bar{c}$ ou MCT
	MWCT	Temps d'achèvement moyen pondéré *(Mean Weighted Completion Time)*
	TADC	Total des différences absolues des C_j *(Total Absolute Differences in Completion Time)*
	$\sum F_j$	Total des Flowtime F_j
	$\sum w_j F_j$	Total pondéré des flowtime
	$\bar{F}$	Flowtime moyen
	$\sum W_j$	Total des temps d'attente (Total Waiting Time)
	$\sum w_j W_j$	Total pondéré des temps d'attente (Total Weighted Waiting Time)
	TST	Total des durées des opérations de configuration *(Total setup time)*
	TSC	Coût total dû aux configurations *(Total setup time)*
	TNS	Nombre d'opérations de configuration *(Total number of setups)*
	Précocité	E_{max} Précocité maximale
		$\sum E_j$ Total des précocités *(Total earliness)*
		$\sum w_j E_j$ Total pondéré des précocités *(Total weighted earliness)*
		$\sum h(E_j)$ Somme des pénalités de précocité *(Total earliness penalties)*
	Retard	L_{max} Retard maximal
		$\sum L_j$ Total des retards *(Total Lateness)*
		$\sum w_j L_j$ Total pondéré des retards *(Total weighted Lateness)*
		$\bar{L}$ Retard moyen
		$\sum U_j$ Nombre de jobs en retard *(Number of late [tardy] jobs)*
	Lenteur	T_{max} Lenteur maximale
		$\sum T_j$ Total des lenteurs (*Total tardiness*)
		$\sum w_j T_j$ Total pondéré des lenteurs (*Total Weighted Tardiness*)
		$\bar{T}$ Lenteur moyenne
		$\sum U_j$ Nombre de jobs lents (Number of tardy [late] jobs)
		$\%U_j$ Taux de jobs lents
		$\sum h(T_j)$ Somme des retards dus aux lenteurs (Total tardiness)

Tableau 2 Classification des problèmes avec la représentation « α|β|γ »

Dans la section suivante sont présentées les contributions de la littérature à l'ordonnancement dynamique, et plus particulièrement les solutions proposées pour gérer la flexibilité des outils de production.

4. TRAVAUX RELATIFS A L'OPTIMISATION DE L'ORDONNANCEMENT DYNAMIQUE

Dans la littérature, les nombreuses approches visant à résoudre le problème d'ordonnancement sont souvent statiques. De ce fait, elles ne sont pas, dans leur grande majorité, applicables dans des environnements réels. Les perturbations, telles que les pannes machines et les arrivées dynamiques de produits, peuvent invalider la planification à tout moment (Ouelhadj et al. 2009). L'ordonnancement dynamique, parfois noté (ré)ordonnancement ou en anglais *(re)Scheduling* permet, quant à lui, de répondre aux spécificités et aux variations de l'environnement du SP. Une définition de l'ordonnancement dynamique est donnée dans (Fahmy et al. 2014). Il est considéré comme permettant à la fois de générer le programme de production initial, mais aussi de réagir aux événements en temps réel. En cas de perturbation, les systèmes de planification dynamiques (aussi appelés « en ligne ») peuvent procéder aussi bien à de petits ajustements du plan initial qu'à des réparations plus conséquentes, qu'à un réordonnancement complet. Une seconde définition axée sur les aspects d'incertitudes est donnée dans (Cottet et al. 2002) ; l'ordonnancement dynamique est défini par des caractéristiques de tâches (délais, périodes, temps de calcul, etc.) qui ne sont pas connues à l'avance. Les détails relatifs à la tâche ne sont connus qu'au moment où elle doit être exécutée pour la première fois. Enfin, une dernière définition de Lopez et al. (2010) met l'accent, sur l'absence de plan clairement spécifié à l'avance. Les auteurs indiquent que l'ordonnancement dynamique est caractérisé par des décisions de planification prises individuellement à chaque libération d'une machine. Ceci permet donc de prendre en compte des aléas de production. En d'autres termes, cette définition s'appuie sur l'absence de plan spécifié à l'avance, mais une série de décisions prises au moment où une opération à exécuter sur une machine inactive doit être choisie. Bien que divergeant sur l'absence ou non de plan initial, ces définitions présentent le dénominateur commun suivant : les méthodes d'ordonnancement dynamiques prennent en charge des événements imprévus qui rendent difficile voire impossible l'application de méthodes statiques plus classiques. En plus de l'aspect dynamique de l'environnement, la plupart des algorithmes classiques ne permettent pas de traiter l'ensemble des contraintes induites par la flexibilité des FMS. Il est alors nécessaire d'élaborer des méthodes spécifiques qui tiennent compte de toutes les hypothèses et contraintes intrinsèques au FMS (Godinho Filho et al. 2014).

De nombreux travaux ont été proposés dans la littérature pour tenter de répondre aux problèmes de (ré)ordonnancement. Ils peuvent utiliser une méthode unique ou en combiner plusieurs. Parmi ces méthodes les plus significatives sont celles dites « d'optimisation-simulation ». Ce couplage aussi nommé optimisation via simulation, est composé de deux modules distincts : l'un d'optimisation et le second de simulation (Figure 11). Cette organisation permet au premier module de proposer des solutions au module de simulation, se voulant de plus en plus performantes au fil des itérations (Chaari 2010).

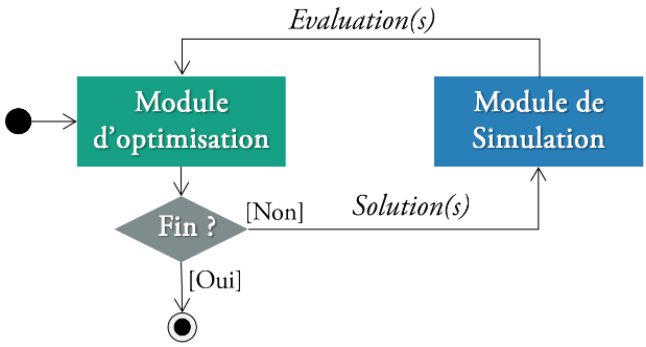


Figure 11 Principe du couplage optimisation-simulation

Nous classerons les différentes approches en 4 types illustrés dans la Figure 12 et présentés ci-après.

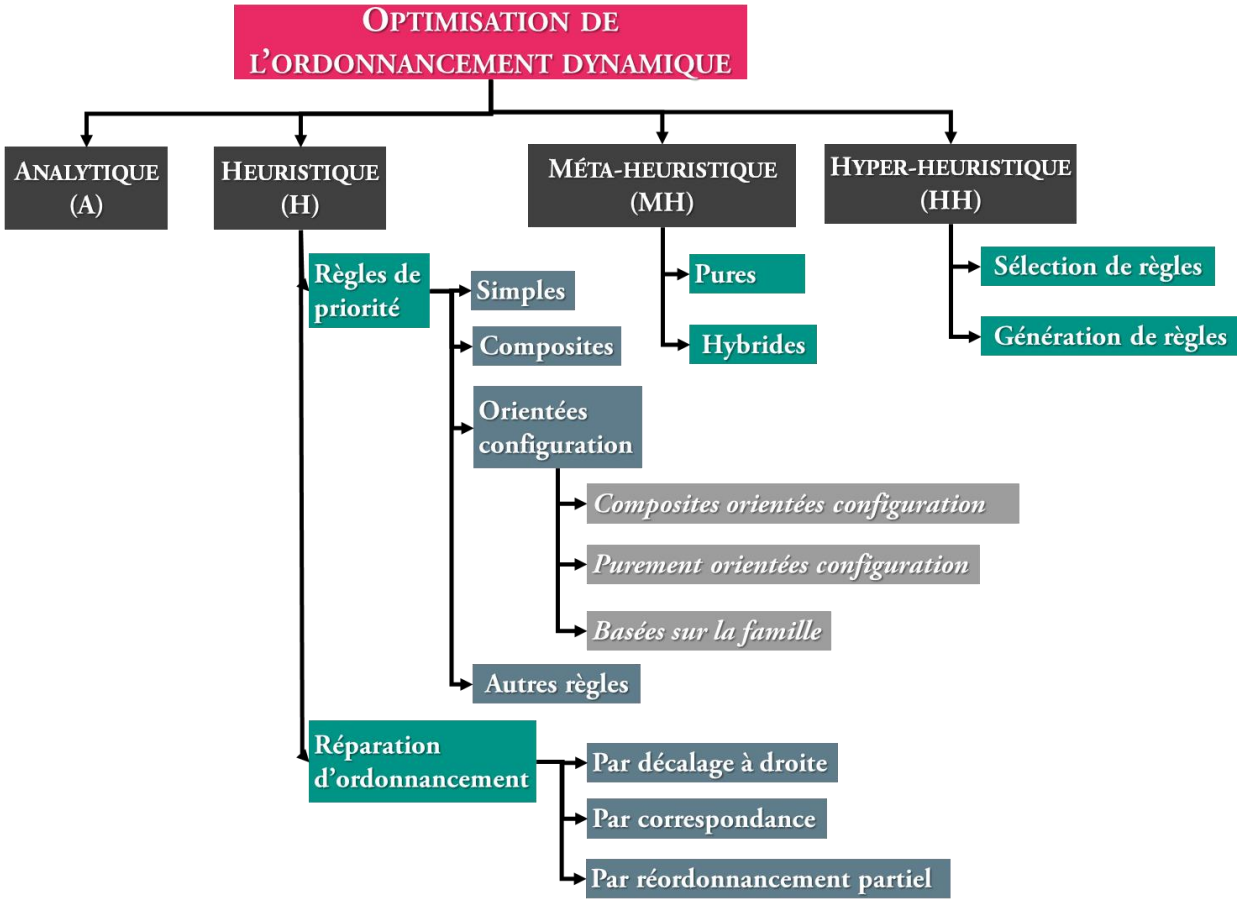


Figure 12 Approches pour l'optimisation de l'ordonnancement dynamique

4.1. APPROCHES ANALYTIQUES (A)

La plupart des problèmes d'ordonnancement d'ateliers flexibles sont difficiles à résoudre en un temps raisonnable (i.e. NP-difficile). Le plus souvent, les chercheurs ne traitent que les versions déterministes et statiques de ces problèmes par les méthodes analytiques classiques (Sabuncuoglu et al. 2003). De plus, dans le cas d'arrivées dynamiques de produits, l'indisponibilité d'informations sur les jobs avant leur arrivée rend difficile, voire impossible, la création d'un modèle fiable. A cela s'ajoute la survenue d'événements dynamiques pouvant invalider l'ordonnancent et nécessitant un réordonnancement complet. La modélisation

mathématique des SP dynamiques est une tâche ardue (Corning 2002). Pannequin et al. (2009) soulignent que leur résolution par des méthodes analytiques classiques implique des modèles bien plus complexes que la théorie ne peut résoudre.

Néanmoins, de nombreux travaux utilisent des modèles mathématiques pour calculer une limite (notamment par des techniques de relaxation de contraintes). Ces bornes servent par la suite à évaluer des méthodes plus sophistiquées. Par exemple, dans (Mousakhani 2013), afin de minimiser le retard total $\sum T_j$ dans un FJSP, un modèle mathématique sous la forme d'un MILP (*Mixed Integer Linear Programming*) est tout d'abord proposé. Ensuite, les résultats obtenus sont comparés avec des méta-heuristiques telle que l'*Iterated Local Search* (ILS).

4.2. HEURISTIQUES (H)

Compte tenu des difficultés à appliquer des méthodes exactes pour les problèmes de (re)ordonnancement, les méthodes approchées sont souvent préférées aux méthodes analytiques. Les heuristiques sont destinées à fournir une réponse rapide bien qu'elles ne soient pas nécessairement optimales.

▪ **Règles d'ordonnancement :** cette catégorie particulière d'heuristiques, en anglais « *Dispatching Rules* » (DR), calculent une priorité pour chaque job à traiter en fonction de ses attributs. Du fait de leur simplicité et leur faible besoin en informations et en calculs, les DR peuvent être appliquées à des problèmes complexes de planification (Pickardt et al. 2012). En 1977 déjà, Panwalkar recensait 113 DR (Panwalkar et al. 1977). Depuis, la littérature s'est enrichie de nombreuses nouvelles règles (Sarin et al. 2011; Blackstone et al. 1982; Fan et al. 2016). Elles peuvent être classées en quatre catégories :

– ***Règles de priorité simples :*** elles constituent les règles généralistes classiques, comme la règle du « premier arrivé, premier servi » FIFO *(First In / First Out)*, la règle du « dernier arrivé, premier servi » LIFO *(Last In / First Out)*, la règle SPT *(Shortest Processing Time)* qui consiste à faire passer la tâche la plus courte en premier, ou encore la date d'échéance la plus proche « *earliest due-date* » (EDD).

– ***Règles composites :*** elles sont le résultat de la combinaison de plusieurs DR simples, comme la règle FIFO / SPT proposée dans (Conway et al. 1962). Cette règle composite implique que pour les travaux qui dépassent une durée d'attente spécifique, la règle FIFO est utilisée. Pour tous les autres travaux la règle SPT est employée.

– ***Règles orientées configuration*** : la prise en compte explicite de facteurs comme les temps de configuration peut considérablement améliorer les performances des DR. Dans (Allahverdi 2015), l'auteur souligne la nécessité de prendre en compte explicitement les temps de configuration afin d'accroître la productivité, d'éliminer le gaspillage, d'améliorer l'utilisation des ressources et le respect des délais. Dans (Pickardt et al. 2012), les DR orientées configuration, ou « *setup-oriented* », sont classées en trois catégories :

  - *Règles purement orientées configuration* : la première règle SST (*Shortest Setup Time*) a été proposée dans (Gavett 1965). Cette règle simple sélectionne en priorité le job avec le temps de setup le plus faible.
  - *Règles composites orientées configuration :* ces règles intègrent dans le calcul des priorités, un terme pour réduire le nombre d'opérations de configuration. Un des premiers exemples est donné dans (Wilbrecht et al. 1969). Les auteurs ajoutent le temps de configuration en modifiant la règle SPT.

- *Règles orientées configuration basées sur la famille :* ces règles basées sur la famille de produits suivent une approche de hiérarchisation des tâches. Elles cherchent à réduire les temps de configuration en considérant explicitement les informations relatives aux familles et en regroupant un maximum d'opérations de même famille. Dans (Mosier et al. 1984), un modèle de logique de prise de décision de ces règles est proposé en tant que processus en plusieurs niveaux prenant en compte trois décisions interdépendantes : (1) Quand changer de famille (i.e. quand effectuer l'opération de configuration) ? ; (2) quelle famille choisir ? ; (3) quelle opération choisir parmi les jobs de la famille de produits sélectionnée ?

– **Autres règles :** une autre règle plus sophistiquée existe. Contrairement aux règles traditionnelles, les heuristiques dites « têtes levées » (ou *« Look Ahead »*) ne prennent pas en compte uniquement les jobs déjà en file d'attente mais tentent de prendre en compte l'effet des tâches qui arriveront avant l'achèvement des tâches en cours de planification.

Parmi les nombreux travaux proposant des DR, nous pouvons citer Huq et al. (2009) s'intéressent à l'optimisation multi-objectif d'un FJSP avec des coûts de configuration « non nuls » et différents taux d'arrivée de produits. Les règles SPT, LPT, FIFO et EDD sont évaluées selon plusieurs critères. Les résultats montrent que le choix optimal de la DR dépend non seulement de paramètres dont le taux d'arrivée, mais également du choix des objectifs.

Dans (Kia et al. 2010), pour résoudre un FFSP dynamique avec $ST_{SD,F}$, six heuristiques sont proposées. Diverses conditions telles que le ratio de temps d'installation et le taux d'utilisation (%u) de l'atelier sont testés par des simulations à événements discrets. Les résultats montrent qu'une des règles proposées donne de meilleurs résultats pour la mesure du flowtime moyen $\overline{F}$ et une autre pour le retard moyen $\overline{L}$.

Dans (Krishnan et al. 2012), les auteurs proposent une règle nommée RIT/TPT pour *« Ratio of Idle Time to Total Processing Time »*. Cette règle signifie littéralement « ratio temps d'inactivité / temps total de traitement » et est proposée dans le contexte d'un FJSP avec arrivées dynamiques de produits et occurrences de pannes. Les résultats démontrent l'amélioration du taux d'utilisation des machines induite par la règle RIT/TPT pour un FJSP avec pannes.

Une autre application plus récente est présentée dans (Joo et al. 2013) dans laquelle des tâches de plusieurs types arrivent dans FFSP de manière dynamique. Une opération de configuration indépendante de la séquence $ST_{SI,F}$ peut être nécessaire pour changer de type de job. Deux solutions basées sur des règles de priorité intégrant une mesure de la qualité ainsi que des informations sur l'atelier en temps réel sont proposées. L'objectif étant la maximisation du taux de qualité et la minimisation du retard moyen $\overline{L}$.

En plus des règles de séquencement, les règles de sélection des machines ou *« Assignement Rules »* (AR) peuvent elles aussi grandement améliorer les performances de l'ordonnancement. Subramaniam et al. (2000) proposent trois règles pour optimiser la lenteur moyenne $\overline{T}$ ainsi que les coûts dans un FJSP avec dates d'échéance et indisponibilité machine. Celles-ci sont évaluées sous trois conditions dynamiques, à savoir le respect des dates d'échéance, la flexibilité des itinéraires et la fiabilité des machines. Les résultats des simulations indiquent que les performances des DR simples sont considérablement améliorées lorsqu'elles sont utilisées avec des AR adéquates.

▪ **Heuristiques de réparation d'ordonnancement :** la réparation d'ordonnancement est très différente du réordonnancement complet impliquant une régénération à partir de zéro. Le principe est de procéder à des ajustements locaux pour prendre en compte les nouveaux événements. Du fait de leur simplicité, ces heuristiques sont le plus souvent couplées à d'autres

méthodes. Selon (Gholami et al. 2009), trois types de méthodes basées sur des heuristiques de réparation d'ordonnancement peuvent être distinguées :

- ***Réparation par décalage à droite :*** en anglais « *right-shift schedule repair* », cette heuristique consiste à décaler à droite les opérations restantes après la survenue d'une indisponibilité machine, proportionnellement à la durée de celle-ci.
- ***Réparation par correspondance :*** en anglais « *match-up schedule repair* », cette stratégie consiste à réordonnancer les opérations pour atteindre un point de correspondance avec l'ordonnancement initial quelque part dans le futur.
- ***Réparation par réordonnancement partiel :*** elle implique que seule la ou les opérations concernées par la panne sont réordonnancées.

Ces heuristiques, souvent très spécialisées, offrent dans certains cas bien définis des solutions intéressantes. La prochaine section est consacrée aux solutions basées sur des métaheuristiques plus généralistes.

4.3. META-HEURISTIQUES (MH)

Des méthodes exactes telles que le Branch & Bound (B&B) garantissent de trouver une solution optimale mais au prix de longs calculs. D'autres approches telles que la recherche aléatoire ou la descente de gradient sont des méthodes approchées plus rapides mais n'évaluant qu'une solution à la fois. Ceci les rend inefficaces pour des problèmes de grande taille. En comparaison, certaines méta-heuristiques telles que les algorithmes à population rendent le processus de recherche beaucoup plus efficace en calculant plusieurs solutions simultanément (Jorapur et al. 2016). Nous pouvons ainsi distinguer les approches métaheuristiques pures des approches hybrides (plusieurs métaheuristiques ou avec des heuristiques).

▪ **Méta-heuristiques pures :** un exemple d'ordonnancement dynamique est présenté dans (Bierwirth et al. 1999). Le modèle est appliqué à un JSP dans un environnement non déterministe. Les produits sont libérés au fur et à mesure. Un Algorithme Génétique (AG) est proposé et intègre une procédure de décodage afin d'améliorer la qualité des ordonnancements générés. Il est testé dans différentes situations de charge de travail afin d'améliorer le flowtime moyen $\bar{F}$. Un autre exemple utilisant un AG est présenté dans (Fattahi et al. 2010). Deux objectifs sont considérés simultanément pour un FJSP avec arrivées dynamiques. Les auteurs cherchent à établir un équilibre entre performance et stabilité. Les résultats des expérimentations montrent que l'AG proposé est capable de trouver les solutions optimales pour les problèmes de petite taille et des solutions quasi-optimales pour les problèmes de taille moyenne. D'autres algorithmes évolutionnaires proposent un fonctionnement différent des AG. Par exemple, l'Estimation de l'Algorithme de Distribution (EDA) permet d'estimer les relations entre les différentes variables d'un problème d'optimisation. Il n'emploie donc pas d'opérateurs de croisement ou de mutation. La construction de l'échantillon se fait directement à partir des paramètres de distribution issus de l'itération précédente. Une application est proposée dans (Liu et al. 2015) et plus précisément une variante d'EDA nommée *Fast Estimation of Distribution Algorithm* (FEDA). Les auteurs s'intéressent à un problème de planification dynamique d'ateliers flexibles avec temps de traitement flou. L'algorithme FEDA proposé permet d'améliorer la vitesse de convergence et la précision de calcul, en comparaison à l'EDA d'origine. De plus, d'autres travaux ont proposé d'appliquer des approches de l'intelligence artificielle (IA) au problème d'ordonnancement dynamique. Parmi ces techniques d'IA, se positionnent les réseaux de neurones artificiel (ANN pour Artificial Neural Network), généralement constitués

d'un grand nombre d'unités (neurones artificiels) reliées entre elles selon un schéma de connexion particulier (Kotsiantis 2007). Un exemple d'application de ANN aux problèmes d'ordonnancement dynamiques peut être trouvé dans (Eguchi et al. 1999). Le but est d'obtenir un programme qui, à défaut d'être optimal, est satisfaisant et généré en temps réel. Le ANN est entrainé par apprentissage de problèmes de petite taille. Après cette étape préliminaire, le ANN est utilisé pour calculer la priorité de chaque job en temps réel. Des expérimentations montrent la faisabilité de l'approche pour résoudre des problèmes de taille supérieure.

▪ **Méta-heuristiques hybrides :** ces dernières années, un nombre croissant de solutions basées sur des techniques hybrides sont apparues. Contrairement aux approches purement heuristiques ou purement méta-heuristiques, l'idée est de tirer parti de plusieurs méthodes.

Dans (He et al. 2013), les auteurs abordent un problème d'ordonnancement d'un FJSP soumis à des pannes machine. Une première phase de pré-ordonnancement avec un AG est d'abord effectuée. Puis plusieurs heuristiques sont utilisées pour gérer les pannes, à savoir une stratégie d'insertion de temps mort, une stratégie de changement d'itinéraire et une heuristique de décalage à droite. Ainsi, l'approche combinant différentes stratégies tente de maintenir le réordonnancement aussi stable et robuste que possible. Les résultats montrent l'efficacité de cette hybridation entre plusieurs méthodes pour deux indicateurs globaux appelés « mesures de robustesse relatives moyennes » (ARRM) et « mesures de stabilité moyennes » (ASM).

Un autre exemple d'hybridation entre méta-heuristique et heuristiques est proposé dans (Halat et al. 2015), les auteurs abordent un FJSP avec famille de produits afin de minimiser le C_{max}. Un modèle de programmation linéaire est développé en prenant notamment en compte les temps de transport intercellulaires et les temps de configuration dépendants de la séquence et de la famille de produit $ST_{SD,F}$. Afin de résoudre efficacement les problèmes de taille réelle dans un délai raisonnable, une approche heuristique basée sur un AG est proposée et complétée par une heuristique de réparation d'ordonnancent. Les résultats démontrent que le modèle réduit considérablement le temps de calcul.

Enfin, nous citerons un dernier exemple d'hybridation entre méta-heuristique et heuristique présenté dans (Logendran et al. 2007). Les auteurs s'adressent à un problème de minimisation des retards pondérés $\sum w_j L_j$ dans atelier composé de machines parallèles hétérogènes avec ST_{sd}. Des contraintes dynamiques de libération des jobs et de disponibilité des machines sont aussi considérées. Six algorithmes de recherche différents basés sur la recherche taboue (TS) sont développés pour générer des solutions initiales. Puis, pour améliorer l'efficacité et l'efficience des algorithmes de recherche, quatre mécanismes différents de recherche de solution initiale, basés sur des DR, sont développés spécifiquement.

D'autres approches proposent d'hybrider plusieurs méta-heuristiques. Dans (Zhang et al. 2017), les auteurs s'intéressent à l'ordonnancement dynamique d'un FJSP. L'objectif est la minimisation du total pondéré des lenteurs $\sum w_j T_j$ avec différentes contraintes telles que des pannes machine, des annulations et des rushs de commandes. Pour cela, les auteurs utilisent à la fois la négociation du système multiagents (MAS) et l'Optimisation des Colonies de Fourmis (ACO). La solution hybride qui en résulte est appelée HMA pour *« Hybrid MAS/ACO »*.

Un autre exemple d'hybridation est proposé dans (Zandieh et al. 2010). Le problème abordé concerne la planification dynamique avec arrivées d'ordres de fabrication aléatoires et pannes de machines. Pour cela, les auteurs proposent une solution basée sur la recherche à voisinage variable (VNS). L'algorithme VNS est constitué de deux phases : une phase déterministe au sein de laquelle une recherche locale permet de converger vers un optimum local, et une seconde

phase stochastique pour s'en échapper (Mjirda 2015). L'originalité de l'approche réside dans un ANN doté d'un algorithme d'apprentissage de type « back-propagation » utilisé pour mettre à jour les paramètres du VNS. Les résultats comparatifs avec des règles de répartition tirées de la littérature montrent l'efficacité de la méthode proposée sous diverses conditions de fonctionnement d'ateliers.

De manière complémentaire, des travaux plus détaillés sur les applications de l'IA à l'ordonnancement peuvent être cités dont ceux de (Kotsiantis 2007) qui étudient les méthodes d'apprentissage automatique, telles que les ANN, les arbres décisionnels et l'apprentissage bayésien, ou encore (Priore et al. 2006) qui traitent des méthodes d'apprentissage automatique pour les appliquer à l'ordonnancement dynamique dans les systèmes de fabrication flexibles.

4.4. HYPER-HEURISTIQUES (HH)

Le néologisme « Hyper- Heuristique » (HH) a été introduit pour la première fois dans un document de conférence (Cowling et al. 2001). Bien que l'idée d'utiliser une combinaison d'heuristiques ne soit pas nouvelle, ce n'est que dans ce document que fut clairement identifié ce groupe de méthodologies. Plusieurs définitions peuvent ainsi être trouvées dans la littérature (Burke et al. 2019; Swiercz 2017; Pillay et al. 2018). Cependant, la caractéristique commune des HH est qu'elles fonctionnent sur un espace de recherche composé d'heuristiques en essayant de sélectionner les plus performantes ou d'en générer de nouvelles. Ainsi, dans (Ochoa et al. 2009), il est question *d'heuristiques pour sélectionner d'autres heuristiques*. Les auteurs définissent les HH comme des méthodologies de recherche visant à automatiser le processus de sélection ou de combinaison d'heuristiques simples afin de résoudre des problèmes complexes. De plus, Burke et al. (2010) soulignent un aspect intéressant qu'est la boucle d'apprentissage automatique que les HH peuvent comporter. Les auteurs désignent une HH comme un mécanisme d'apprentissage permettant de sélectionner ou de générer des heuristiques pour résoudre des problèmes de recherche informatique. S'il existe un feed-back issu d'une phase d'apprentissage, ce dernier peut être soit « en ligne », c'est-à-dire que l'apprentissage a lieu pendant que les algorithmes résolvent une instance de problème, soit « hors-ligne » lorsque l'algorithme recueille des connaissances avant d'essayer de les généraliser plus tard lors de l'exécution du système. Etant donné que les DR sont limitées de par leur horizon d'informations local, aucune règle ne surpasse les autres règles pour tous les objectifs, scénarios et conditions du système (Heger et al. 2016). L'intérêt des HH est de combiner plusieurs règles spécifiques à un problème ou d'en générer de nouvelles pour améliorer la performance globale.

- **HH pour la sélection de règles (HH-S) :** comme vu précédemment, rechercher une combinaison parmi un ensemble de règles est antérieur au terme d'hyper-heuristique. Par exemple, dès 1975 dans Hershauer et al. (1975), les auteurs ont mis au point une approche normalisée pour la sélection de règles de séquencement (SR). La SR est considérée comme une combinaison linéaire de facteurs pondérés. La règle est utilisée pour déterminer la priorité de chaque job dans les files d'attente. Chaque règle génère des coûts calculés à l'aide de la simulation.

Dans (Min Hee Kim et al. 1994) un RTS (Real Time Scheduling) est appliqué à un SP flexible. La solution proposée vise à obtenir un RTS par le biais de la simulation de DR. En fonction des informations issues de la simulation à événements discrets qui évaluent un ensemble de DR candidates, la meilleure DR est sélectionnée. Le système de contrôle vérifie périodiquement la performance du système. Dans (Vázquez Rodríguez et al. 2007), une HH est ainsi développée

pour minimiser le C_{max} dans un atelier flowshop hybride avec ST_{sd}. La solution consiste alors en une séquence de plus d'une douzaine de règles qui sont alors appelées l'une après l'autre.

Dans (Zhou et al. 2013), les auteurs proposent l'utilisation de techniques issues de l'IA pour assurer une sélection de règles efficace. Le problème concerne un problème de planification des opérations de photolithographie dans l'industrie des semi-conducteurs. Le but est d'obtenir un ordonnancement dynamique basé sur le réseau de neurones de Kohonen (KNN). La méthode utilise le KNN pour la sélection des meilleures règles d'ordonnancement. Les résultats des expériences indiquent que la méthode proposée est non seulement faisable, mais aussi efficace pour le RTS d'un système de fabrication de semi-conducteurs.

Un dernier exemple de d'HH pour le sélection de règles est donné dans (Heger et al. 2016). L'approche proposée permet d'ajuster de manière dynamique les paramètres d'une DR en fonction des conditions d'un FSP dynamique avec ST_{sd}. L'influence de différents paramètres de la règle choisie sur la performance du système est estimée par une méthode d'apprentissage automatique. Ces données d'apprentissage sont générées par des simulations préliminaires. L'approche permet ainsi de réduire les retards moyens $\bar{L}$.

HH pour la génération de règles (HH-G) : les HH pour la génération de règles sont également populaires. La raison principale en est qu'elles permettent la génération de fonctions de priorité de format et de longueur variables. Et cela, sans qu'il soit nécessaire de définir un format de base à l'avance (Branke et al. 2016). Contrairement aux HH de sélection, elles sont à même de générer des DR entièrement nouvelles. Par exemple, Tay et al. (2008) traitent un FJSP avec plusieurs fonctions objectives, à savoir le makespan minimum, le retard moyen $\bar{L}$ et le flowtime moyen $\bar{F}$. L'espace de recherche est constitué de paramètres et d'opérateurs servant à faire évoluer des DR composites à l'aide de la Programmation Génétique (GP). Les résultats expérimentaux montrent que les DR composites ainsi générées surpassent les DR simples mais aussi un certain nombre de DR composites issues de la littérature.

Un autre exemple d'HH pour la génération de règles est donné dans (Nie et al. 2011). Le problème abordé est un JSP dynamique. Aucune information n'est disponible sur les tâches avant leur arrivée et elles ne peuvent donc pas être ordonnancées à l'avance. Les auteurs appliquent la programmation par expression génétique (GEP) pour construire automatiquement des règles de planification efficaces. Les résultats de simulations informatiques montrent que la méthode de GEP a permis de construire des DR efficaces.

Freitag et al. (2016) s'intéressent à un problème d'ordonnancement avec dates d'échéances. L'algorithme utilisé se base sur la programmation génétique (PG) multi-objectifs utilisant un module de simulation, pour à la fois réduire la précocité, mais également les retards. Une application à l'industrie des semi-conducteurs montre que les DR résultantes surpassent significativement les DR de la littérature.

Enfin, un exemple d'apprentissage de nouvelles règles en combinant plusieurs techniques est proposé dans (Shahzad et al. 2016). L'approche d'ordonnancement se veut être le résultat d'une synergie entre optimisation combinatoire, apprentissage supervisé et simulation à événements discrets. L'extraction de DR hors-ligne utilise des arbres de décision pour générer des règles simples de type « If-Then » à partir d'une base de connaissances. Ces règles sont ensuite utilisées pour l'ordonnancement en ligne. Les expérimentations montrent l'efficacité de la méthode pour la réduction du retard maximal L_{max}.

4.5. SYNTHESE SUR L'OPTIMISATION DE L'ORDONNANCEMENT DYNAMIQUE

Le Tableau 3 propose une synthèse des travaux les plus significatifs, traitant de l'optimisation de l'ordonnancement dynamique.

Source	Atelier (α)	Contraintes (β)		Indicateurs (γ)	Approche	
		Ress.	Prod.		Type	Algo.
Min Hee Kim et al. (1994)	FJ	-	-	C_{max}	HH-S	DR + Simu.
Bierwirth et al. (1999)	J	-	r_j	$\overline{F}$	MH	AG
Subramaniam et al. (2000)	FJ	brk	D_j	$\overline{T}$	H	AR
Logendran et al. (2007)	R	ST_{SD} + brk	r_j	$\sum w_j L_j$	MH/H	TS + DR
Vázquez Rodríguez et al. (2007)	FFm	ST_{SD}	D_j	C_{max}	HH-S	DR
Tay et al. (2008)	FJ		D_j	$C_{max} + \overline{F} + \overline{L}$	HH-G	GP
Huq et al. (2009)	FJ	SC_{SD}	D_j	$\overline{T} + T_{max} + \sum U_j$	H	DR
Kia et al. (2010)	FJ	$ST_{SD,F}$	a_j	$\overline{F}, \overline{T}$	H	DR
Fattahi et al. (2010)	FJ		r_j	C_{max}	MH	AG
Zandieh et al. (2010)	FJ	brk	a_j	C_{max}	MH/IA	VNS + ANN
Nie et al. (2011)	J	-	a_j	C_{max}	HH-G	GEP
Mousakhani (2013)	FJ	-	-	$\sum T_j$	MH/A	ILS / MILP
Zhou et al. (2013)	FJ	-	a_j	$\overline{F}$	HH-S	KNN + DR
Joo et al. (2013)	FFm	$ST_{SI,F}$	D_j, a_j	$\overline{L}$	H	DR
He et al. (2013)	FJ	brk	D_j,	ARRM, ASM	MH/H	AG + 3 H
Halat et al. (2015)	FJ	$ST_{SD,F}$	fmls	C_{max}	MH/H	AG + H
Heger et al. (2016)	Fm	ST_{SD}	a_j	$\overline{L}$	HH-S	Simu + ML
Freitag et al. (2016)	FJ		D_j	C_{max}	HH-G	DR+Simu
Shahzad et al. (2016)	FJ	-	fmls	L_{max}	HH-G	DR + Simu
Kia et al. (2017)	Fm	ST_{SD}	r_j	$\overline{F} + \overline{L}$	H	6 H
Zhang et al. (2017)	FJ		a_j	$\sum w_j T_j$	MH/MH	SMA + ACO

Tableau 3 Travaux relatifs à l'optimisation de l'ordonnancement dynamique

Parmi les solutions présentées dans la littérature, les règles d'ordonnancement et les hyper-heuristiques se démarquent par leur grande réactivité. Les méthodes d'hybridation ou de couplage telle que l'optimisation-simulation semblent aussi intéressantes pour l'ordonnancement dans des systèmes aussi complexes que dynamiques. Cet état de l'art nous permet de mettre en évidence les deux points suivants :

La majorité des travaux traitant d'ordonnancements dynamiques n'abordent pas de cas avec une absence totale d'information sur les produits avant leur arrivée. En effet, dans la plupart des travaux, l'aspect dynamique du problème se limite le plus souvent à des dates de libération de produit ou encore à l'ajout de pannes machines.

Le second point est que le plus souvent, les approches sont destinées à être supportées par des systèmes de contrôle centralisés « monolithiques ». Ce qui limite fortement la réactivité du pilotage alors que celle-ci est justement, plus que nécessaire.

5. CONCLUSION

Ce premier chapitre a permis, dans un premier temps, de présenter les systèmes manufacturiers flexibles. À travers les différents aspects de la flexibilité, il apparait que cette caractéristique des FMS procure un avantage concurrentiel mais implique aussi un accroissement de la complexité. Dans la seconde partie de ce chapitre, le processus décisionnel clé qu'est l'ordonnancement des opérations de production a été introduit. Les trois champs $\alpha|\beta|\gamma$ communément utilisés dans la littérature ont permis de passer en revue les principaux aspects des problèmes d'ordonnancement. Une dernière partie a permis de dresser un état de l'art des travaux relatifs à l'optimisation de l'ordonnancement dynamique. Celui-ci a permis de mettre en lumière la grande diversité des problèmes, mais aussi des solutions présentées dans la littérature. Nous avons pu souligner l'intérêt des approches réactives comme les règles d'ordonnancement et les hyper-heuristiques. De plus, il a été montré que l'immense majorité des approches sont centralisées ce qui limite la réactivité du pilotage.

Afin de mieux comprendre ces besoins spécifiques à la conduite des systèmes de production modernes, le prochain chapitre est consacré au pilotage des SP dans le contexte cyber-physique. Ces systèmes, portés par les récentes évolutions technologiques de l'industrie 4.0, offrent de nouvelles possibilités. Cependant, leur conduite constitue aussi un nouveau défi.

CHAPITRE II : PILOTAGE DES SYSTEMES DE PRODUCTION CYBER-PHYSIQUES (CPPS)

1. INTRODUCTION

Le chapitre précèdent a permis d'introduire les systèmes particuliers que sont les FMS et de présenter leur principal avantage qui réside en leur flexibilité. Cette flexibilité présente néanmoins un inconvénient dans la mesure où elle engendre une complexification conséquente de l'ordonnancement de la production. Si des solutions pour l'ordonnancement des FMS plus ou moins performantes existent, elles demeurent souvent cantonnées à des problèmes statiques, difficilement déployables dans des industries réelles. Les évolutions des environnements industriels modernes redéfinissent les impératifs de pilotage de la production et exigent, plus que jamais, une réactivité et une agilité des systèmes de contrôle, ce qui nécessite de nouvelles approches plus adaptées aux contraintes actuelles et futures (p. ex. la mass-customisation ou la rationalisation de la consommation énergétique).

Portée par les Systèmes Cyber-Physiques, l'industrie 4.0 transforme progressivement les systèmes manufacturiers classiques en Systèmes de Production Cyber-Physiques, en anglais *Cyber-Physical Production Systems* (CPPS). Ce second chapitre, s'intéresse aux fonctions de pilotage dans ces systèmes qui sont issus de la convergence du monde informatique et du monde physique. Une première section définit, tout d'abord, les fonctions du système de pilotage et passe en revue les différentes architectures de contrôle, des classiques centralisées aux plus décentralisées. La section suivante détaille les spécificités de la conduite des CPPS. Dans l'objectif de gagner en réactivité, indispensable dans des environnements dynamiques, l'intérêt se porte sur les systèmes distribués, et notamment sur les « Systèmes Contrôlés par les Produits » ou « *Product-Driven System Control* » (PDSC). Toutefois, un certain nombre de problèmes et de difficultés dus à la distribution des nœuds décisionnels existent, le principal étant la « myopie décisionnelle » (cf. II-3.3.3). Un état de l'art qui suit s'attache à mettre en évidence les différentes applications des PDSC dans la littérature ainsi que l'apport du concept de produits dit « intelligents » à l'ordonnancement de la production.

2. PILOTAGE DES SYSTEMES DE PRODUCTION (SP)

S'il existe des décisions qui sont cruciales au sein d'une entreprise manufacturière, ce sont bien celles ayant trait à la planification des tâches de production. D'après (Le Moigne 1994), « décider » consiste à identifier et à résoudre les problèmes que rencontre une organisation. En ce qui concerne les SP, (Pach 2013) définit le pilotage comme le fait de *« décider dynamiquement des commandes pertinentes à donner à un système soumis à l'incertitude pour atteindre un objectif donné décrit en termes de maîtrise de performances »*. Cette définition est une évolution de celle proposée initialement par (Trentesaux et al. 2002) dans laquelle le terme « incertitude » remplace celui de « perturbation ». Le terme de perturbation désigne une perte partielle ou totale des capacités de production (p. ex. le ralentissement d'une machine CNC, ou la panne d'un robot manipulateur) alors que le terme d'incertitude inclut, en plus des perturbations dites classiques, d'autres variations qu'il est nécessaire de prendre en considération (p. ex. la non-connaissance à

l'avance des OF). En effet, si ces dernières ne constituent pas des perturbations au sens strict, elles demeurent des facteurs à prendre en compte dans la conduite du SP. Afin de mieux appréhender toute la difficulté du pilotage de la production, la section suivante s'intéresse aux différentes fonctions de conduite.

2.1. VUES DECISIONNELLE ET FONCTIONNELLE DU SP

Les représentations classiques des fonctions de pilotage du SP comptent plusieurs niveaux. La Figure 13 propose une synthèse de plusieurs modèles et contributions. Elle permet de mettre en parallèle les trois niveaux décisionnels (Gudehus et al. 2012; Caramia et al. 2006) avec les niveaux fonctionnelles d'un SP (Pach 2013; Zambrano Rey 2014), mettant ainsi en lumière les nombreux flux d'informations.

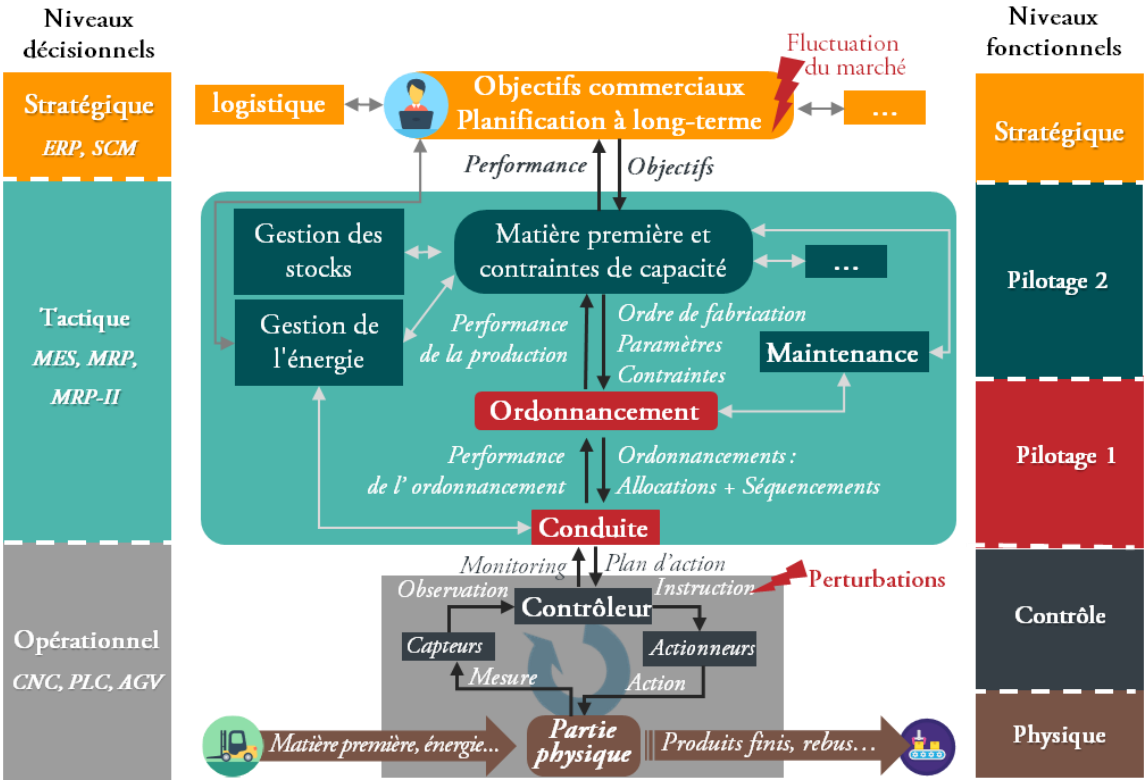


Figure 13 Vue décisionnelle et fonctionnelle d'un SP

▪ **Niveau stratégique :** à ce niveau, les décisions concernent les orientations de l'entreprise sur le long terme. Il est ici question de faire des choix, par exemple liés à une anticipation de l'évolution du marché, amenant à l'implantation de nouvelles unités, l'extension des lignes de production, ou encore le développement du catalogue produits. Le rapprochement des activités commerciales avec celles de production est une évolution nécessaire pour les entreprises manufacturières. Elle nécessite de disposer d'informations correctes, précises et à jour (Brandl 2002) sur des outils et logiciels tels que les ERP (*Entreprise Ressource Planning*), le SCM (*Supply Chain Management*) ou encore le PLM (*Product Life-cycle Management*).

▪ **Niveau tactique :** ce niveau est en charge de la mise en œuvre des décisions prises au niveau stratégique, sur une période de temps plus courte. En tenant compte d'objectifs à long terme, mais aussi de paramètres et contraintes tels que la nomenclature des produits et l'état des stocks, des OF sont créés selon un type particulier de flux de production (p. ex. poussé, tiré ou tendu). Cette planification à moyen terme définit les besoins en matières premières, les commandes aux fournisseurs, les opérations de maintenance, etc. Dans (Pach 2013), les fonctions « conduite » et

« ordonnancement » ne sont plus supportées comme dans un MES (*Manufacturing Execution System*) classique, construit autour d'une base de données centralisée. Les décisions peuvent provenir d'éléments extérieurs au MES, ce qui est plus en adéquation avec le caractère distribué des CPPS. Les deux niveaux fonctionnels considérés sont :

– ***Niveau pilotage 1 :*** il intègre les fonctions « ordonnancement » et « conduite ». Celles-ci ont une incidence directe sur la performance et la réactivité. A partir des OF (et selon les paramètres et contraintes), l'ordonnancement va pouvoir affecter les ressources de production à des tâches (cf. I-3).

– ***Niveau pilotage 2 :*** il regroupe les autres fonctions de pilotage plus classiques comme la gestion des stocks, la gestion de l'énergie ou encore la maintenance. A partir des objectifs, sont définis les besoins en articles MRP (*Materials Requirements Planning*), la planification des ressources de fabrication MRP-II (*Manufacturing Resources Planning*), etc.

▪ **Niveau opérationnel :** ce niveau comprend, par exemple, les machines CNC, les robots, les tapis convoyeurs, les véhicules autoguidés (AGV) et les Automates Programmables Industriels PLC (*Programmable Logic Controller*). A ce niveau, le système tente d'appliquer le plan d'action prévu au niveau tactique. Il comprend deux niveaux fonctionnels :

– ***Niveau contrôle :*** le terme « Contrôle » inclut l'ensemble de la boucle permettant de contrôler un processus ou un système, des capteurs aux actionneurs. Une boucle fermée peut alors être définie comme quelque chose qui existe entre un système qui contrôle et un système qui est contrôlé (Wiener 1961; Trentesaux 2009). Le niveau « contrôle » est en lien direct avec la fonction « Conduite » du niveau « Pilotage 1 ». Il mesure l'état du système physique à l'aide de capteurs. Ainsi, il peut surveiller l'exécution du plan d'actions et agir par le biais d'actionneurs pour satisfaire les exigences imposées par le niveau pilotage 1 (p. ex. en prenant des mesures correctives en cas de perturbations).

– ***Niveau physique :*** il désigne la partie concrète et tangible du système composé de flux, depuis les matières premières jusqu'aux produits finis (p. ex. les composants, les pièces, les produits semi-finis ou intermédiaires), ainsi que les rebuts, déchets, etc.

2.2. TYPES DE PILOTAGE DE LA PRODUCTION

Dans la littérature, les systèmes de pilotage peuvent être classés selon différents critères tels que le lancement du système (p. ex. la temporalité), l'approche appliquée, l'optimalité de la solution, ou encore selon les flux d'échange d'informations avec le système opérant. Ainsi, nous pouvons considérer six grands types de pilotage de la production (Figure 14) :

▪ **Pilotage prédictif :** le pilotage prédictif est destiné à préparer le lancement d'un système existant. Les paramètres de pilotage (déterministes et/ou probabilistes) sont définis avant l'exécution sur le système réel (p. ex. a priori). Au prix d'un important coût en calculs, des méthodes souvent issues de la recherche opérationnelle tels les MILP, la programmation non linéaire ou les AG (Algorithmes Génétiques) (Trentesaux et al. 2002; Birgin et al. 2014) déterminent une solution optimale ou quasi-optimale, supposant un comportement « idéal » du système. Néanmoins, le manque de réactivité les rend souvent irréalisables dans la plupart des environnements. Car en effet, les SP réels nécessitent la planification d'un grand nombre d'OF complexes, avec de nombreuses contraintes, et des objectifs tout aussi complexes. A cela s'ajoutent les perturbations qui peuvent survenir tout moment et invalider la planification.

▪ **Pilotage proactif :** tout comme le pilotage prédictif, le pilotage proactif est utilisé en phase de

préparation qui précède le lancement du processus de production, à la différence, ici, que l'on essaie d'anticiper un certain nombre d'aléas. L'objectif est de déterminer a priori, la meilleure réponse possible à apporter si l'aléa survient. Pour ce faire, les perturbations les plus probables sont tout d'abord déterminées (p. ex. en utilisant les données d'indicateurs de performance préexistantes ou bien extraites d'une base de données des événements passés). Puis sont estimés les risques de déviation des objectifs prévisionnels. Des outils tels que la simulation hors ligne sont utilisés afin d'anticiper les perturbations et de calculer les risques de non-respect de l'objectif visé (Qu et al. 2015; Lamas et al. 2016).

▪ **Pilotage réactif :** le pilotage réactif intervient pendant l'exécution de la production, une fois le lancement effectué. Il a pour but de corriger le plus rapidement possible les variables de décision pour faire face à une perturbation. La principale caractéristique de ces approches est qu'elles fournissent des réponses très rapides. Cependant, cette même réactivité du processus de décision implique de privilégier la rapidité à la qualité. Ce qui donne des réponses le plus souvent non-optimales. On distingue deux cas nécessitant une correction :

– Dans le premier cas, une perturbation survient (p. ex. une panne machine ou une commande urgente) et le système de pilotage tente d'estimer les conséquences sur l'objectif initial. Dans ces circonstances idéales, l'événement a été anticipé à une phase proactive antérieure. La réponse à apporter est donc déjà connue et est alors appliquée.
– Le second cas est plus courant. L'événement n'a pu être anticipé. Le système de pilotage tente alors de corriger au mieux les paramètres afin de minimiser l'impact de cette perturbation.

▪ **Pilotage correctif :** dans le cas où l'on ne peut maîtriser une dérive et ramener les paramètres dans des valeurs normales, il est parfois trop tard pour adopter une démarche préventive. Les données alors collectées et pourront servir lors d'une modélisation future (Armetta 2006). La solution consistera à modifier la trajectoire de référence, et parfois même l'objectif de production (Berchet et al. 1999). Le pilotage correctif s'appuie souvent sur des systèmes à base de connaissances (Davenport 2010; Smith 1995; Fox 1994). L'atteinte de certains seuils de dérive déclenchent une démarche corrective, par exemple, une opération de maintenance sur la machine concernée (Wang et al. 2014; Kajko-Mattsson et al. 2001).

▪ **Pilotage prédictif-réactif :** ce pilotage hybride est conçu pour tirer parti de plusieurs techniques, tout en tentant de limiter leurs inconvénients respectifs. En général, une première phase hors-ligne utilise une technique prédictive pour définir des instructions générales visant une optimalité globale. Puis, lors de l'exécution, et en cas de perturbations, une méthode réactive est en charge d'apporter une réponse à celles-ci (Baños et al. 2011; Jimenez et al. 2016) par exemple par des approches basées sur les systèmes de contrôle holoniques ou multiagents (Cardin et al. 2017; Zambrano Rey 2014).

▪ **Pilotage proactif-réactif** : comme le prédictif-réactif, cette méthode essaye de cumuler les avantages de plusieurs méthodes. En général, on distingue 3 phases (Demeulemeester 2009) :

– Une première phase génère une planification dans un scénario idéal (non perturbé), tout en intégrant les différentes contraintes.
– Une seconde phase anticipe les perturbations les plus susceptibles de se produire pendant l'exécution. Les différentes réponses possibles sont ainsi prédéfinies pour créer un plan robuste pourvu d'un ensemble d'alternatives.
– Une dernière phase consiste en une procédure réactive pour réparer la planification de base

en cas de perturbation. L'objectif est ainsi de rectifier de manière dynamique le planning selon la méthode définie par la phase proactive initiale (Lou et al. 2012; Wu et al. 2009).

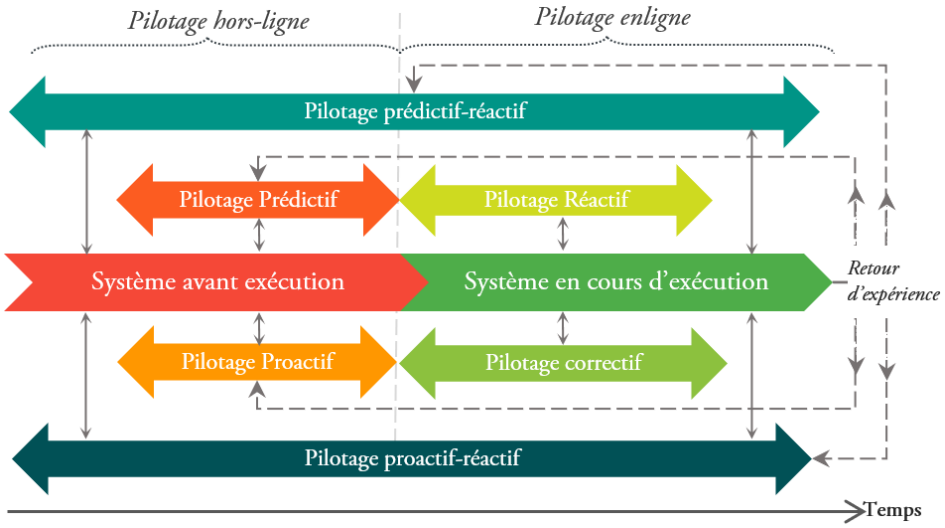


Figure 14 Types de pilotages de la production

2.3. ARCHITECTURES DES SYSTEMES DE PILOTAGE

De nombreux travaux se sont intéressés à l'organisation structurelle des systèmes de pilotage (Jiménez 2017; Trentesaux 2009). Cette architecture permet de spécifier : le nombre de composants décisionnels, les niveaux qui les regroupent mais également les types de relations qui les lient. Le choix d'une architecture appropriée de système de pilotage peut potentiellement, fortement améliorer les performances du SP (Dhaevers et al. 2007).

2.3.1. COMPOSANTS DE L'ARCHITECTURE

La structure de pilotage est une organisation composée d'un certain nombre de centres de décision reliés par des liens de dépendance :

▪ **Les centres de décision** : ces composants sont des structures organisées et autonomes, ayant une faculté décisionnelle associée. Ils dépendent aussi fortement de la stratégie de l'entreprise (Berchet et al. 1999). Selon les architectures, ils peuvent être uniques ou multiples, répartis sur un ou plusieurs niveaux décisionnels différents. Les autres éléments importants permettant de caractériser l'architecture de pilotage sont les liens de dépendances.

▪ **Liens de dépendances :** de natures très diverses, ils relient les centres de décision au sein de l'architecture (Clivillé 2004; Pujo et al. 2002). Qu'ils soient verticaux ou horizontaux, ils jouent un rôle majeur dans la cohérence du système de pilotage et ont donc, de facto, un impact non négligeable sur l'efficacité et la performance de ce dernier (Dhaevers et al. 2007). Dans la littérature, trois catégories de liens sont communément répertoriées :

– ***Liens de subordination :*** ce sont des liens verticaux entre des éléments de niveaux différents. Ils sont de deux types : soit coercitifs dans le cas de décisions imposées à un ou plusieurs éléments de niveau inférieur (p. ex. relation maître-esclave), soit il s'agit de retours d'informations sur les objectifs, actions enclenchées et indicateurs issus du niveau inférieur.
– ***Liens de coordination :*** ces liens horizontaux permettent aux centres de décisions de s'informer mutuellement, de se consulter ou de décider conjointement d'actions à mener pour atteindre les objectifs. Cette coordination est une action collective orientée vers un but commun pour éviter la prise de décisions contradictoires.

– ***Liens de communication :*** s'ils ne constituent que de simples échanges d'informations entre les différents composants du système de pilotage, ils peuvent néanmoins être cruciaux pour la mise en œuvre et le suivi des actions.

A partir de ces deux éléments (i.e. entités et liens) et selon l'organisation choisie, différentes architectures sont possibles.

2.3.2. CLASSIFICATION DES ARCHITECTURE DES SYSTEMES DE PILOTAGE

Les quatre principales architectures furent introduites initialement dans (Dilts et al. 1991) et développées par la suite dans de nombreux travaux (Trentesaux 2009; Pujo et al. 2002; Jiménez 2017) :

Centralisée : les approches traditionnelles allouent l'autonomie décisionnelle à un composant unique exerçant des relations coercitives de type maître-esclave sur l'ensemble du système.

Totalement hiérarchique : afin de diminuer la complexité, une pratique courante consiste à le diviser selon les dépendances hiérarchiques. Le système de contrôle est alors organisé en une structure pyramidale hiérarchisée. Le but est de répartir l'autonomie décisionnelle sur plusieurs niveaux. Un composant domine alors pleinement les actions d'un ou plusieurs autres composants d'un niveau inférieur (Pach 2013).

Totalement hétérarchique : cette architecture ne comprend qu'un seul niveau composé de plusieurs composants interconnectés. Ils se partagent l'autonomie décisionnelle dans le but d'augmenter la réactivité du système par la collaboration. Il n'y a aucune relation maître–esclave mais des relations de coopération dans lesquelles ils collaborent et partagent la responsabilité de l'exécution de leurs actions et de la réalisation de leurs propres objectifs.

Semi-hétérarchique : les approches traditionnelles centralisées et hiérarchisées ne permettent pas de gérer efficacement des événements imprévisibles tels que des perturbations ou encore des contraintes complexes. Pour cela, des architectures hybrides comme les architectures semi-hétérarchiques, proposent de répartir les décisions entre plusieurs composants interconnectés par des relations soit hiérarchiques, soit hétérarchiques (Figure 15). Le contrôle peut donc être divisé par le biais de ces différentes relations pour s'adapter aux spécificités du système.

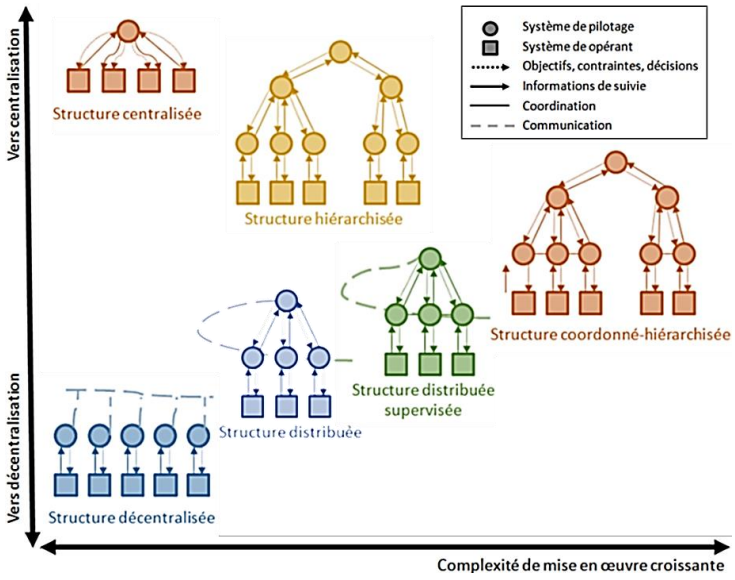


Figure 15 Organisation structurelle des systèmes de pilotages (Trentesaux 2009)

La section suivante présente la conduite des SP dans le nouveau contexte de l'industrie 4.0 et des technologies cyber-physiques.

3. CONDUITE DES SYSTEMES DE PRODUCTION CYBER-PHYSIQUES (CPPS)

Nous vivons actuellement les prémices d'une quatrième révolution industrielle (Figure 16). Celle-ci se veut une réponse aux bouleversements causés par l'irruption du numérique à tous les niveaux des chaînes de valeur industrielles. Cette mutation prend la forme de grands projets nationaux : « Industrie 4.0 » en Allemagne, « l'industrie du futur » en France ou encore « Made In China 2025 » en Chine. Ces ambitions technologiques visent avant tout l'apprentissage de nouveaux processus de conception et de fabrication. Cette 4éme révolution exploite de nouveaux modes d'interaction entre les acteurs économiques et leurs clients, aussi bien à l'intérieur qu'à l'extérieur de la chaîne de valeur (Kohler et al. 2016).

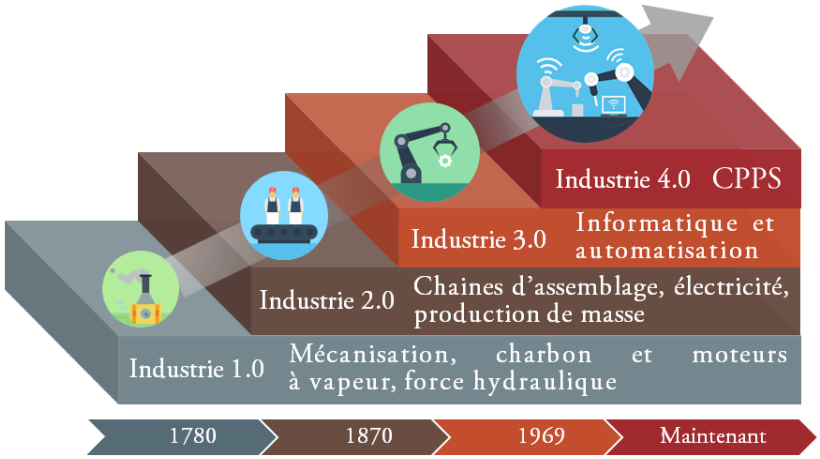


Figure 16 Les 4 révolutions industrielles

D'après (Rüßmann et al. 2015), l'industrie 4.0 repose sur neuf piliers fondateurs, à savoir les robots autonomes, les outils de simulation, le Big Data et analyses de données, la réalité augmentée, l'intégration horizontale et verticale des systèmes, la fabrication additive, les technologies Cloud, l'Internet industriel des objets (IIoT) et enfin la cybersécurité. Bien que ces avancées technologiques soient en partie déjà utilisées dans le secteur industriel, leur généralisation, leur combinaison et leur montée en puissance promettent de transformer en profondeur les entreprises manufacturières.

3.1. PRINCIPES DES CPPS

Bien que les composantes du SP n'aient pas fondamentalement changé, leurs interactions ont elles d'ores et déjà entamé une profonde mutation. Le nouveau paradigme novateur des CPPS (*Cyber-Physical Production Systems*) fait de l'usine, un ensemble d'éléments technologiques (logiciels, capteurs, systèmes embarqués, etc.) en interaction permanente. Cette transformation fait progressivement évoluer les systèmes de production en « Usines Intelligentes ». Ce concept ne possède pas de définition claire qui fasse consensus. Selon (Vinet et al. 2011), plusieurs termes peuvent être utilisés de manière interchangeable : A l'Usine Intelligente (Hozdić 2015), s'ajoute «l'usine ubiquitaire» (Dubey et al. 2017) ou encore «l'usine des objets» (FoT) ou *Factory Of Things* (Zuehlke 2010). Sa mise en œuvre est caractérisée par l'intégration de systèmes cyber-physiques (CPS) dans la fabrication. Les équipements au sein des unités de fabrication sont un ensemble de capteurs et d'actionneurs, auxquels est associé un modèle virtuel de l'usine. Ce dernier peut être utilisé pour la prise de décision automatisée pour améliorer les

performances (Arvind et al. 2016).

Ainsi, les CPPS sont le résultat de la convergence de deux mondes (Figure 17). L'un virtuel (ou cybernétique), composé d'un ensemble d'unités de traitement logiques, de commande et de capteurs. L'autre physique, constitué d'un ensemble d'actionneurs (Dong et al. 2012). Afin de permettre des interactions entre les différents éléments constituant le CPPS, de nouvelles méthodologies se développent, visant à gérer les échanges informationnels entre la partie physique et l'espace cyber-informatique (Bagheri et al. 2015).

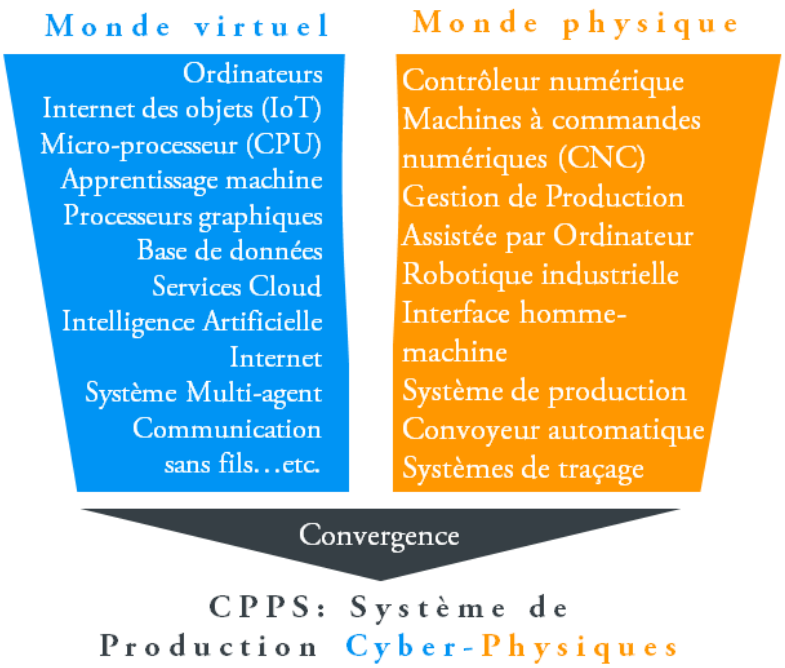


Figure 17 Convergence virtuelle et physique dans les CPPS

Dans (Monostori 2015), l'accent est mis sur l'enchevêtrement de différentes technologies au sein des CPPS. L'auteur met en lumière comment l'évolution des CPPS s'appuie sur un ensemble de technologies. En effet, le développement des ordinateurs, et de l'électronique en général, a abouti à l'essor de machines à commandes numériques (CNC) et de robots. Des algorithmes de vision par ordinateur appliqués à la robotique ont permis le développement de cellules robotisées efficientes. Les réseaux informatiques ont permis la transmission des données générées par les CIM. L'Internet a, lui aussi, considérablement révolutionné la coopération entre les êtres humains et les systèmes (p. ex. les entreprises étendues, la gestion de la chaîne logistique ou les réseaux de production). La communication sans fil, les réseaux de capteurs et l'Internet des objets ont rendu possible le développement de systèmes de production aux rendements toujours plus élevés (Främling et al. 2013). Les données stockées sont exploitées par des techniques d'IA et d'apprentissage automatique ou Machine Learning, ce qui a largement contribué à la réalisation de systèmes de production intelligents.

3.2. DESAGREGATION DE LA PYRAMIDE D'AUTOMATISATION

Comme nous l'avons vu précédemment (cf. II-2.3.2), parmi les différentes architectures proposées, les plus courantes sont les approches centralisées et hiérarchiques. Dans un environnement industriel stable, elles permettent d'obtenir d'excellentes performances (Thomas et al. 2012). Néanmoins, les temps de propagation des informations (vers le haut) et des décisions (vers le bas) génèrent décalages et instabilités. De ce fait, elles sont généralement incapables de faire face à des événements inattendus. De plus, et afin de d'opérer efficacement dans un

environnement réaliste, un système de pilotage doit être à même de réagir de la façon la plus rapide et la plus appropriée possible face aux perturbations.

C'est pour combler ce manque de réactivité et d'agilité que les approches décentralisées et distribuées virent le jour plus récemment. Elles cherchent à répartir l'autonomie décisionnelle sur un ensemble d'entités. Dans (Bettenhausen et al. 2013), les auteurs désignent cette distribution graduelle des centres de décisions dans un CPPS comme étant la « désagrégation progressive de la pyramide d'automatisation ». Comme illustré par la Figure 18, la pyramide est peu à peu remplacée par un ensemble de nœuds et services en réseau, organisés de manière décentralisée. Les services, les données et les composants matériels peuvent être liés à n'importe quel nœud du réseau émergent. Ils forment des modules fonctionnels abstraits qui constituent le système de pilotage du CPPS.

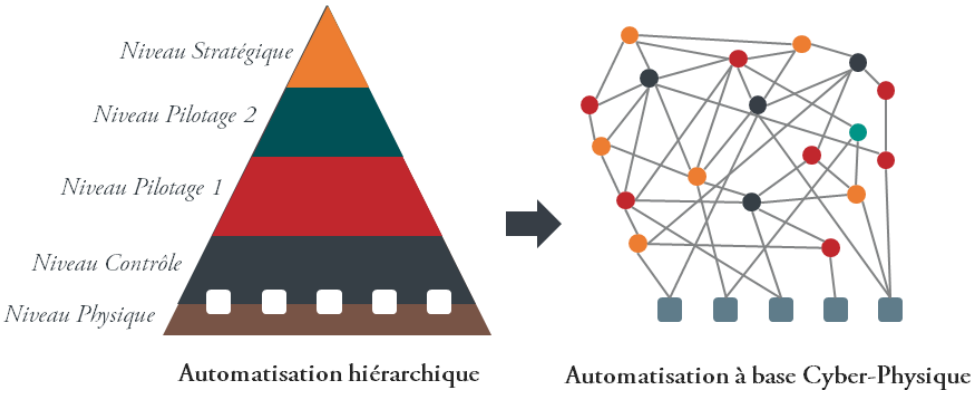


Figure 18 Désagrégation de la pyramide d'automatisation dans les CPPS

La distribution des centres de décision dans les CPPS peut « théoriquement », permettre de réduire la complexité du problème global, de diminuer les incertitudes et de hiérarchiser certains objectifs. Cependant, un certain nombre de difficultés et de limitations peuvent apparaitre.

3.3. PROBLEMATIQUE DE PILOTAGE DES CPPS EN ENVIRONNEMENT MANUFACTURIER COMPLEXE

Intrinsèquement, les CPPS hébergent de plus en plus d'entités de traitement et de nœuds décisionnels, ce qui affecte positivement leur performance globale. Le système de pilotage devient alors plus flexible, gérant mieux les perturbations que les architectures centralisées classiques. Parmi les bénéfices attendus de l'approche CPPS, nous pouvons citer l'optimisation des processus de production, l'amélioration de la personnalisation des produits, la minimisation des coûts de production indirects, ou encore la possibilité de « centrer » les approches sur l'humain (Rudtsch et al. 2014). Néanmoins, un certain nombre de problèmes peuvent apparaitre pour le pilotage dans des environnements complexes. Ici, nous retiendrons trois grandes sources complexifiant le pilotage des CPPS à savoir la dynamicité de l'environnement, les aspects inhérents à la complexité des systèmes, et les phénomènes de myopie. Ces problèmes sont explicités dans les sections suivantes.

3.3.1. ENVIRONNEMENTS DYNAMIQUES

Dans le cadre de la planification dynamique, les événements arrivent au fur et à mesure et sont traités en continu. Ces événements souvent imprévisibles tels que des pannes ou des arrivées intempestives de produits, peuvent survenir et perturber le comportement normal du système (Hamzadayi et al. 2016). Dans (Vieira et al. 2003), les auteurs classent ces événements en deux catégories :

- **Evènements liés aux produits :** de nouveaux produits peuvent arriver à tout instant dans le futur. Ces dates d'arrivées peuvent être connues ou non. Cela contraste fortement avec les problèmes de planification statiques, dans lesquels tous les travaux sont connus à l'instant 0. Une distinction doit cependant être faite. Les incertitudes liées aux arrivées dynamiques impliquent l'indisponibilité d'informations sur les produits avant leur arrivée. L'autre concept est plus proche des « dates de libération » (cf. I-3.2.1). Dans ce second cas, les informations sont disponibles avant la libération des produits. Ce point fondamental rend possible des approches prédictives ou proactives (cf. II-2.2)
- **Evènements liés aux ressources :** ils correspondent pannes machine, aux temps de traitement aléatoires, aux limites de chargement, etc. (Sun et al. 2014).

3.3.2. ASPECTS DE LA COMPLEXITE

En ce qui concerne la complexité, même la définition d'un « système complexe » demeure ambiguë (Whitesides et al. 1999). Néanmoins, un consensus se dégage pour caractériser de tels systèmes au travers de propriétés singulières dont ils sont dotés telles que la non-linéarité, l'émergence, l'adaptation ou encore les boucles de rétroaction. Dans (Framinan et al. 2014), une relation est établie entre différents aspects issus de la théorie de la complexité plus abstraite, et les problèmes de planification de la production. Comme explicité dans le Tableau 4, on distingue deux principaux aspects :

- **Aspects de masse :** ils résultent du nombre de jobs, des étapes de production, de processus de fabrication multiples, etc.
- **Aspects chaotiques :** ils sont relatifs aux durées stochastiques, aux arrivées de jobs dynamiques, aux pannes de machines, etc.

Aspects de la complexité		Aspects des systèmes de production
Aspects de masse	Multiplicité	Jobs, étapes, etc.
	Variété	Temps opération, operations, etc.
Aspects chaotiques	Ambiguité	Temps opératoire non-deterministes, etc.
	Incertitude	Arrivées de nouveaux produits, pannes, etc.

Tableau 4 Classification des aspects de la complexité (Framinan et al. 2014)

Les CPPS contiennent de nombreuses entités décisionnelles et la complexité des interactions les reliant favorise l'apparition de comportements émergents (cf. II-3.3.2). Ce phénomène désigne l'apparition d'une relation non triviale et d'un comportement entre des éléments à la suite de plusieurs interactions au sein du système (Sayama 2015). Il en résulte, entre autres, une forte imprévisibilité du système. Ainsi, ces aspects de la complexité ont une incidence importante sur la difficulté du problème de pilotage.

3.3.3. PHENOMENE DE MYOPIE

Dans (Leitão 2009), les entités d'un SP réactif sont décrites comme «autonomes, possédant leurs propres objectifs et dénués de vision globale». Un des « effets de bord » de la distribution décisionnelle est connue sous le nom de « myopie ». Si, à l'origine, elle désigne une pathologie de la vision (p. ex. une personne myope distingue des objets proches, mais ne peut distinguer ceux éloignés), elle a depuis été étendue et adaptée à d'autres domaines, avec autant de définitions différentes.

En ce qui concerne les SP, (Adam et al. 2011) dénombre quatre principaux types de myopie :

Myopie temporelle : ce premier type de myopie concerne la temporalité des décisions. Dans (Trentesaux 2009), la myopie correspond aux conditions qui génèrent un manque de visibilité précise sur les états futurs du système.

Myopie spatiale : correspond au fait qu'une entité a une vision limitée de son environnement. Il en résulte une capacité de prise de décisions limitée du fait du manque d'informations sur les autres entités du système, comme illustré dans la Figure 19.

Myopie de capacité : il s'agit de l'aptitude d'une entité à reconfigurer et à adapter ses capacités pour faire face à des environnements en mutation rapide.

Myopie périphérique : désigne le fait qu'une entité communique et travaille toujours avec le même groupe d'entités.

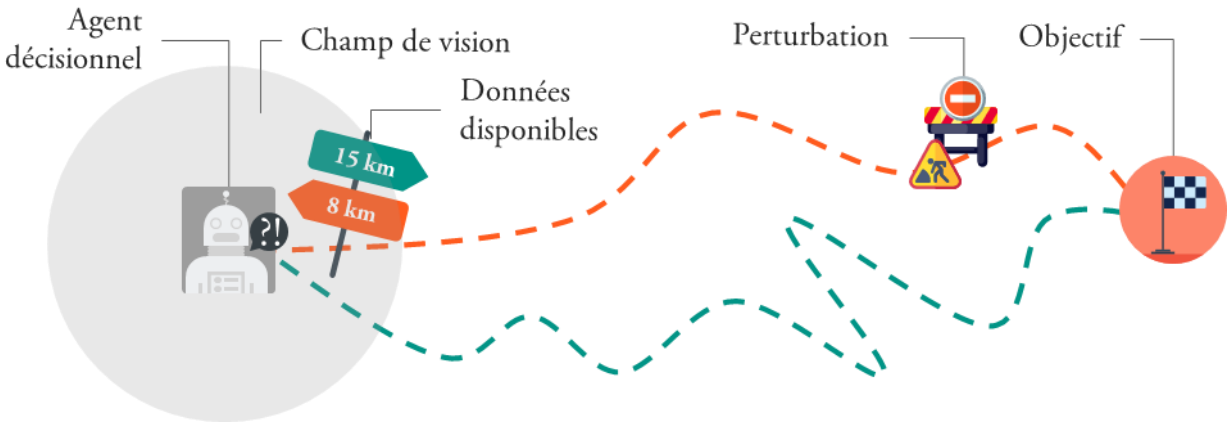


Figure 19 Métaphore de la myopie d'un agent décisionnel

Le comportement myopique est l'un des principaux obstacles au déploiement d'architectures hétérarchiques pour le pilotage de la production. Dans (Zambrano Rey 2014), trois principaux effets indésirables sont recensés :

Optimalité locale : souvent, les décisions d'entités myopes entraînent une optimalité locale, car il est difficile de prévoir et de garantir un certain niveau de performance globale en ayant uniquement une vision locale partielle. De plus, des contradictions peuvent apparaître entre objectifs locaux et objectifs globaux (Dilts et al. 1991), du fait de l'absence de moyen explicite pour relier l'effet de ces actions locales sur la performance globale du système. Elle est très sensible aux algorithmes de décision myope sous-jacentes (Maione et al. 2003).

Imprévisibilité : La complexité des CPPS a pour conséquence une imprévisibilité liée aux algorithmes décisionnels (p. ex. non-linéarité du système). Il est donc difficile de déterminer à l'avance si une perturbation sera gérée efficacement ou non. Il n'est pas impossible que les réactions du système engendrent des pertes pires que celles causées par la perturbation initiale (Zambrano Rey 2014).

Nervosité du système : ce phénomène apparait lorsque les entités décisionnelles réagissent brusquement face à une perturbation. Le système devient alors « nerveux » et risque de déclencher des réactions en chaîne, perturbant d'autres activités telles que la maintenance, l'inventaire ou l'approvisionnement. Cette instabilité de la planification peut se propager dans tout le système (Pujawan 2004). Elle provoque des effets indésirables aux niveau tactique et opérationnel, et induit souvent une augmentation des coûts, ce qui impacte négativement la productivité globale (Herrera et al. 2009).

Le paradigme des CPPS renferme un fort potentiel, aussi bien en termes de gestion optimisée

des processus de production, que de conduite agile et réactive dans des environnements dynamiques et complexes. Il est cependant indispensable de lever, entièrement ou partiellement, certaines limitations énoncées plus haut (par exemple celles liées à la dynamicité, aux aspects de la complexité, et au phénomène de myopie). Dans cette optique, la section suivante décrit l'apport du concept de produit « intelligent » pour le pilotage des CPPS.

4. APPORTS DES PRODUITS INTELLIGENTS AU PILOTAGE DES CPPS

Les Systèmes de Contrôle Pilotés par le Produit ou « Product-Driven System Control » (PDSC) ont récemment attiré l'attention du monde académique et industriel en tant que modèle émergent et prometteur pour le développement de nouvelles générations de systèmes de production intelligents. Ce passage d'une planification centralisée à un contrôle décentralisé, et idéalement en temps réel, offre de nombreuses possibilités pour gérer des systèmes dynamiques et complexes (Windt et al. 2010). Ils semblent également adaptés aux modèles de CPPS dans lesquels la désagrégation de la pyramide décisionnelle (cf. II-3.2) peut être soutenue par la mise en œuvre d'approches centrées sur le produit. La section suivante aborde les concepts liés à la notion de produits « intelligents » et met en évidence les apports potentiels d'une intelligence associée au produit.

4.1. CONCEPTS DE PRODUIT INTELLIGENT

Les progrès des technologies tels que les systèmes embarqués et l'IoT ont créé les bases de l'émergence de produits « intelligents », capables non seulement d'identification et de stockage d'informations, mais capables aussi de comportements complexes via des processus décisionnels autonomes (Morariu et al. 2014). Comme le montre la figure 20, les concepts et terminologies associés aux « Produits Intelligents » (PI) évoluent continuellement. Après plus de deux décennies, le terme reste imprécis et relativement ambigu. Différentes définitions, se concentrant sur certaines aspects, ont été suggérées, discutées et modifiées tout au long du développement de cet axe de recherche (Filho et al. 2017).

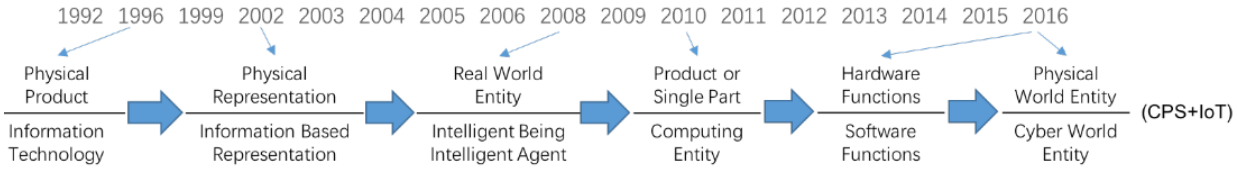


Figure 20 Evolution du paradigme du produit intelligent (Filho et al. 2017)

Comme explicité en figure 21, une analyse de la littérature fait apparaître les termes d'« Intelligent Product » (Sallez et al. 2010; Leitão et al. 2015; Dabrowski et al. 2012), de « Smart Product » (Anderl et al. 2013; Filho et al. 2017; Cronin 2010) ou encore d'« Active Product » (Dobre et al. 2009; Woo et al. 2009; Sallez et al. 2009). Seuls les termes émergeants des domaines de l'automatique et de la productique tels que recensés dans (Sallez 2012a), seront abordés dans la section suivante.

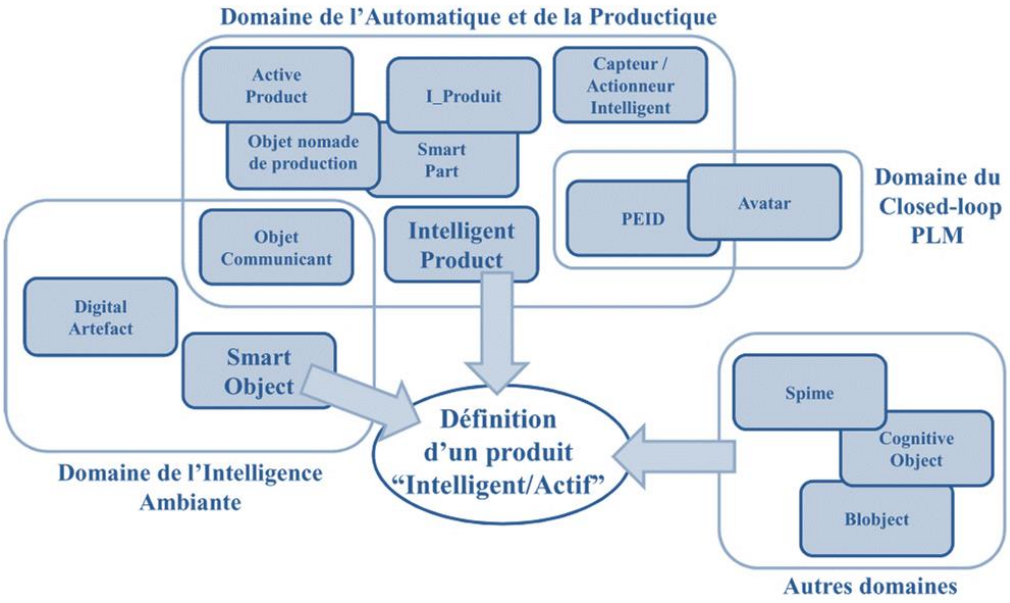


Figure 21 Les différents vocables associés au produit « intelligent / actif » (Sallez 2012a)

4.1.1. DEFINITIONS ET PRINCIPES

Avant de tenter d'apporter une définition au concept de « Produit Intelligent », il nous semble intéressant d'aborder la démarche proposée dans (McFarlane et al. 2013). Les auteurs y posent trois questions permettant de définir les contours du PI dans un contexte manufacturier. Elles interrogent sur la nature du PI, son environnement ainsi que sur ses capacités décisionnelles :

Qu'est-ce qui constitue le Produit Intelligent ? *:* le déploiement de l'intelligence du produit peut se faire au travers de l'intelligence de ses sous-parties (par exemple les composants à assembler), des conteneurs destinés à l'accueillir, des commandes (groupes de produits) ou des expéditions (groupes de commandes).

Avec quel environnement le produit va-t-il interagir ? *:* c'est l'environnement physique, les organisations qui agissent sur le produit, leurs systèmes d'information ou même d'autres produits.

De quelle capacité décisionnelle dispose le PI ? *:* un processus de prise de décision se compose généralement de plusieurs étapes. Ainsi, il convient de déterminer clairement le niveau d'autonomie d'un produit impactant la planification de la production.

Parmi les différentes définitions de la littérature, nous retiendrons celles recensées dans (Meyer et al. 2009) à partir des travaux de (McFarlane et al. 2003), (Kärkkäinen et al. 2003) et (Ventä 2007). Ces définitions décrivent les PI en identifiant un certain nombre de propriétés et caractéristiques.

▪ **Définition 1 :** dans (McFarlane et al. 2003), le PI est caractérisé par cinq propriétés :

– Possède une identification unique.
– Est capable de communiquer efficacement avec son environnement.
– Peut conserver ou stocker des données sur lui-même.
– Déploie un « langage » pour afficher ses fonctionnalités, ses exigences de production, etc.
– Est capable de participer ou de prendre des décisions en rapport avec son propre destin.

▪ **Définition 2** : énoncée dans (Kärkkäinen et al. 2003), cette seconde définition met l'accent sur le contrôle interne de la chaîne tout au long du cycle de vie des produits. Pour que cela soit possible, le PI doit présenter quatre propriétés :

- Posséder un code d'identification unique.
- Posséder un lien vers des sources d'informations au-delà des frontières de l'organisation. Ce lien est soit inclus dans le code d'identification lui-même, soit accessible par un mécanisme de recherche.
- Posséder la capacité de communiquer avec le système d'information mais aussi avec les différents utilisateurs du système.
- Posséder différents niveaux d'intelligence, du niveau le plus basique « identification unique », jusqu'au plus sophistiqué des produits décisionnels.

▪ **Définition 3 :** une troisième définition donnée dans (Ventä 2007), propose une vision plus contextualisée, plus en lien avec le SP qui accueille le PI. Quatre capacités sont énoncées :

- Surveille continuellement son statut et son environnement.
- Peut réagir et s'adapter aux conditions environnementales et opérationnelles.
- Peut maintenir des performances optimales dans des circonstances variables, même dans des cas exceptionnels.
- Est capable de communiquer activement avec l'utilisateur, l'environnement ou avec d'autres produits et systèmes.

A partir de ces définitions et des propriétés qu'elles sous-entendent, il est possible de regrouper les PI selon différents critères détaillés dans la section qui suit.

4.1.2. CLASSIFICATION DES PI

Afin de classer les nombreux concepts existants, nous utiliserons une typologie largement inspirée des travaux de Zbib Nadine (2010), Meyer et al., (2009) et Le Moigne (1994). Comme l'illustre la Figure 26, la typologie est basée sur trois axes qui décrivent respectivement, le degré d'intelligence, la localisation de celle-ci, et le niveau d'agrégation du PI.

▪ **Degrés d'intelligence :** ce premier axe relatif au niveau d'intelligence du PI compte cinq degrés.

- ***Produit passif « ppf » :*** ce type de produit est le plus minimaliste mais sans doute le plus répandu. Il se caractérise par un identificateur unique (par exemple le code barre), dont peuvent être déduites des informations statiques. Les produits de cette classe sont considérés comme passifs puisqu'ils ne peuvent ni répondre aux requêtes et encore moins prendre l'initiative d'en envoyer. Sa seule faculté est donc de mémoriser des informations de base accessibles depuis l'extérieur.
- ***Produit informationnel « pil » :*** ce type de produit est caractérisé par une capacité de mémorisation des informations. Cette mémoire peut être utilisée pour lire et y inscrire des données dynamiques. Ces données peuvent alors être exploitées à différents moment du cycle de vie du produit.
- ***Produit déclencheur « pdr » :*** ce produit est capable d'envoyer des signaux sous forme de message (p.ex. en déclenchant des alarmes sonores pour avertir d'un dépassement de date de péremption). Ces produits déclencheurs sont considérés comme actifs puisqu'ils « décident » de communiquer avec leur environnement sur leur état. Toutefois, il faut nuancer l'influence qu'ils peuvent avoir sur le processus décisionnel. Celui-ci reste très limité.
- ***Produit décisionnel « pdl » :*** doté de capacités de traitement de l'information, le produit est apte à « décider » partiellement ou entièrement de la suite de son cycle de vie (par exemple en décidant de son cheminement sur les chaines de montage). Ce produit peut être considéré

à la fois comme demandeur mais aussi comme fournisseur de services pour d'autres entités. Un premier modèle d'agent produit qualifié « d'actif » est proposé dans (Brun-Picard et al. 1997) et illustré dans la Figure 22.

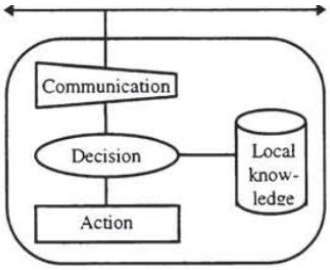


Figure 22 Entité produit de Brun-Picard et al. (1997)

Un autre exemple est proposé dans (Dobre et al. 2009) et est nommé « Produit actif dans un environnement d'intelligence ambiante ». Il propose une application du principe de produit actif aux activités de stockage de matériaux dangereux (Figure 23). Le modèle de comportement interne du PI est dédié à la prise en charge de la sécurité industrielle.

Figure 23 Produit actif en environnement d'intelligence ambiante (Dobre et al. 2009)

– ***Produit acteur physique « pap » :*** cette classe regroupe les produits dotés du plus haut niveau d'intelligence. Ces derniers ont les mêmes propriétés de traitement d'informations que ceux de la classe « décisionnel » mais sont en plus dotés de capteurs et d'actionneurs leur permettant d'agir physiquement sur leur environnement.

▪ **Localisation de l'intelligence :** deux cas peuvent se présenter.

– ***Intelligence embarquée « ie » :*** au produit est associé physiquement une unité de traitement de données pouvant exécuter le process informatique supporté par le produit (Figure 24). Ce type de déploiement de l'intelligence est notamment utilisé dans le cadre de produits dit « augmentés » (Sallez 2012a) ou « autonomes » (Quintanilla et al. 2013).

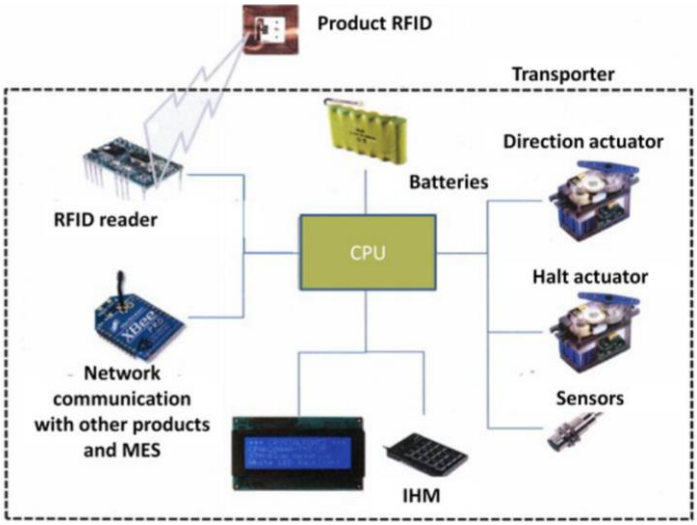


Figure 24 Les composants hardwares d'un produit intelligent (Quintanilla et al. 2013)

– ***Intelligence distante :*** dans ce deuxième cas, l'intelligence est « débarquée » et se situe à distance du produit physique. C'est le cas le plus couramment rencontré (p. ex. le process informatique associé au produit s'exécute sur un ordinateur distant). Un exemple nommé « Objet pointeur vers un système d'information » est présenté dans (Wong et al. 2002). Une image virtuelle représentant le produit physique existe au sein du système d'information. Une etiquette RFID permet alors de pointer sur cette image virtuelle sans pour autant permettre d'interagir avec d'autres entités (Figure 25).

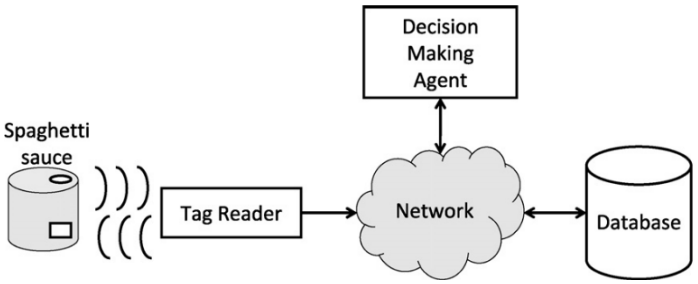


Figure 25 Exemple de « Boite de sauce spaghetti intelligente » (Wong et al. 2002)

▪ **Niveau d'agrégation « id » :** pour ce troisième et dernier axe, deux types d'agrégation sont considérés selon la nature récursive ou non du produit.

– ***Item intelligent « ii » :*** le produit ne gère que ses propres informations et décisions. S'il est lui-même composé d'autres éléments, ces derniers ne peuvent à leur tour être considérés comme des produits intelligents.

– ***Conteneur intelligent « ci » :*** les produits considérés comme étant des conteneurs intelligents sont des produits actifs pouvant eux-mêmes être récursivement composés de produits actifs ou passifs. Cette caractéristique peut être modélisée en utilisant les concepts de l'approche holonique (cf. II-5.3).

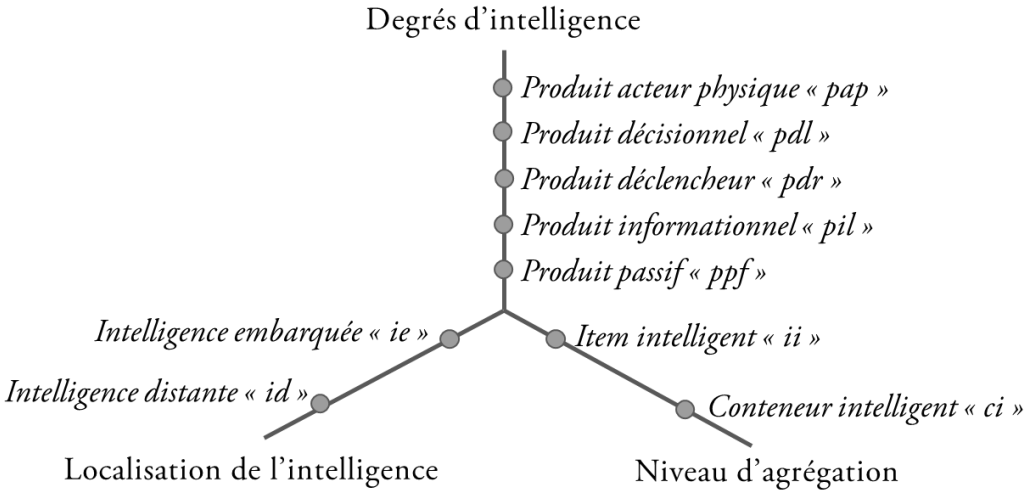


Figure 26 Classification des produits intelligents

La prochaine section s'intéresse à l'apport des capacités décisionnelles des PI à la conduite des systèmes de production.

4.2. LES SYSTEMES CONTROLES PILOTES PAR LE PRODUIT

Dans un premier temps, les approches distribuées se sont d'abord intéressées aux ressources. Les nœuds décisionnels ont alors été liés à des machines dites « intelligentes ». Celles-ci

choisissent les tâches à exécuter et les produits sont « passifs » (Leitão 2009). D'autres travaux, plus récents, proposent de répartir les capacités décisionnelles au niveau des produits eux-mêmes (Keddis et al. 2015). Duffie (1990) fut l'un des pionniers du concept de contrôle hétérarchique dans les systèmes de production. Dès l'émergence des premiers réseaux informatiques, il imagina un système où les pièces s'acheminaient elles-mêmes dans un système de fabrication, un peu à la manière du trafic automobile. Ces produits « intelligents» sont des entités intelligentes capables de franchir les différentes étapes de fabrication et ce, de manière autonome (Sallez 2012b).

4.2.1. DEFINITION ET PRINCIPE

Dans les démarches de systèmes contrôlés par les produits ou *Product-Driven Systems Control (PDSC)*, l'automatisation de la production repose principalement sur des produits dotés de capacités de traitement de l'information, de décision et de communication. Les produits ont alors la faculté d'agir directement sur l'évolution de leur environnement en prenant des décisions sur leurs cheminements au sein du SP. Une définition des systèmes de contrôle par les produits est formulée par (Trentesaux et al. 2012). Le PDSC y est défini comme étant « *le moyen d'optimiser l'ensemble du cycle de vie en traitant des produits dont le contenu informationnel lie de façon permanente leur contenu virtuel et matériel. Ce lien est alors utilisé pour influencer les prises de décisions les concernant* ». Le PI devient actif pour la planification de l'ordonnancement mais aussi tout au long de l'exécution des opérations de production.

4.2.2. MISES EN ŒUVRE

Un exemple de mise en œuvre de PDSC est proposé dans (Pétin et al. 2007) et illustré à par la Figure 27. Le modèle proposé repose sur deux types distincts de contrôleurs :

Les superviseurs de produits qui contrôlent les routages via le système de fabrication en fonction de spécifications de produits et de capacités de ressources de façon dynamique.

Les superviseurs de ressources qui gèrent l'exécution des opérations requises par les superviseurs de produits.

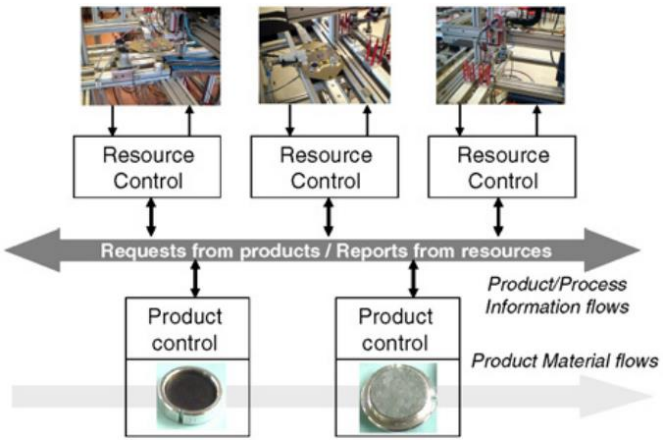


Figure 27 Architecture PDSC (Pétin et al. 2007; Gouyon 2007)

L'approche PDSC s'intéresse au rôle potentiel du produit dans la boucle de contrôle cybernétique (El Haouzi et al. 2009). Dans (McFarlane et al. 2013), la mise en œuvre de PDSC est abordée via l'intégration des décisions des PI dans les boucles de régulation (Figure 28). A cet effet, un ensemble d'algorithmes doivent être développés spécifiquement pour être automatiquement exécutés par les PI. Il est à noter que ces algorithmes diffèrent des

fonctionnalités internes définissant les PI (cf. II-4.1.1), en ce sens qu'ils ne définissent pas de modèles architecturaux généraux, mais des outils spécifiques à l'application aptes à prendre en charge le processus décisionnel dans un contexte de pilotage de la production.

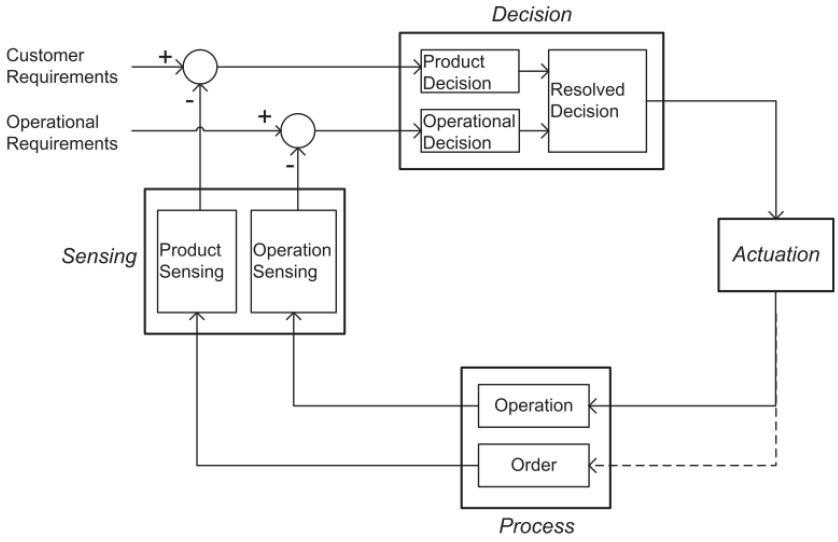


Figure 28 Integration des décisions « produits » dans la boucles de contrôle (McFarlane et al. 2013)

La section suivante présente quelques contributions dans le domaine de la conduite des systèmes de production centrée sur les PI.

5. TRAVAUX RELATIFS AUX SYSTEMES DE PRODUCTION PILOTES PAR LES PRODUITS

Différentes classifications de ces approches existent (Bousbia et al. 2002; Cardin 2019). Dans ce qui suit, le critère de classement retenu est le paradigme utilisé pour construire les entités décisionnelles liées aux produits. Sont successivement analysées, les approches purement orientées Produits Intelligents, les approches multiagents et les approches holoniques utilisant les concepts dérivés *Holonic Manufacturing Systems* (HMS). En réalité, la plupart des développements de systèmes holoniques à ce jour ont déployé des solutions de type multiagents afin de résoudre les problèmes de planification et de contrôle en atelier. Cependant, alors que les systèmes multiagents représentent des environnements purement logiciels, un système holonique) encapsule les aspects physiques et informationnels de l'environnement de fabrication (McFarlane et al. 2002).

5.1. APPROCHES ORIENTEES PRODUITS INTELLIGENTS

Ces approches sont caractérisées par la définition d'une entité Produit indépendamment des concepts SMA ou HMS**.** Un premier exemple d'une telle entité, nommé « Produit Actif Virtuel » (VAP), est proposé dans (Sallez et al. 2009). Les auteurs proposent un modèle de produit actif capable de trouver automatiquement des trajectoires de routage efficaces dans un FMS soumis à des perturbations (Figure 29). L'architecture hétérarchique repose sur deux niveaux : un niveau virtuel dans lequel les produits actifs virtuels évoluent de manière stochastique dans un temps accéléré, et un niveau physique dans lequel les produits physiquement actifs évoluent de manière déterministe en temps réel. Les produits physiquement actifs suivent alors les meilleurs chemins déterminés au niveau virtuel.

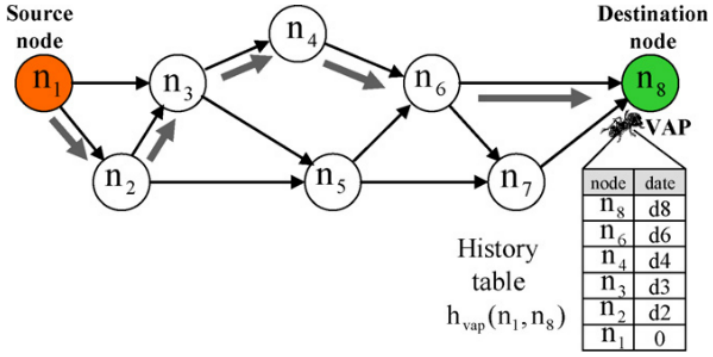


Figure 29 Exemple d'un VAP utlisant la stigmergie pour le routage (Sallez et al. 2009)

Dans la continuité des travaux précédents, un exemple de routage stigmergique au sein d'un FMS est proposé dans (Berger et al. 2010). La structure de contrôle semi-hétérarchique repose sur deux processus : le DAP (processus d'allocation dynamique) et le DRP (processus de routage dynamique). Le DRP est pris en charge par les Produits Actifs (PA). Comme dans le premier exemple, le modèle repose sur deux niveaux : VL de niveau virtuel dans laquelle les produits actifs virtuels évoluent de manière stochastique dans un temps accéléré, et un niveau physique PL dans lequel les produits physiquement actifs évoluent de manière déterministe en temps réel). Les PA utilisent le routage stigmergique pour trouver automatiquement des chemins de routage efficaces dans le système FMS soumis à des perturbations. Dans (Reinhart et al. 2011), une solution construite autour de la technologie RFID permettant de déployer un pilotage prédictif-réactif construit autour de l'entité nommée « Smart Product ».

Le modèle à deux niveaux de (Berger et al. 2010) est étendu dans (Zambrano et al. 2011) pour une planification prédictive-réactive. Un produit actif virtuel est chargé de déployer une planification prédictive-réactive, tandis qu'un produit actif physique est chargé de percevoir un champ de potentiel. Il peut ainsi décider, soit de suivre la VAP, soit de passer en mode autonome pour réagir à un événement incertain. Le but est de réduire l'impact négatif de la myopie temporelle, sans utiliser d'entités centralisées ni créer d'organisations complexes.

Dans (Zbib et al. 2012), les auteurs proposent d'utiliser une méthode basée sur les champs de potentiel ou *Potential Fields* (PF) pour le routage dynamique dans un FMS. L'approche se base sur l'augmentation des capacités d'interaction des différentes entités (par exemple ressources ou produits). Les produits demandant des services aux ressources, ils détectent les champs émis par les ressources et sélectionnent le champ qui répond le mieux à la demande de service. L'architecture de contrôle retenue est totalement hétérarchique et temps réel. Des simulations réalisées sur la plateforme multiagents NETLOGO, ont montré que les produits sont en mesure de choisir la meilleure ressource sans utiliser de mécanismes de négociation. Une autre architecture nommée ORCA-FMS est proposée dans (Pach et al. 2014). Cette fois, l'architecture de contrôle est hybride : elle utilise à la fois une organisation centralisée et décentralisée. Cette hybridation permet de basculer dynamiquement entre une architecture prédictive hiérarchique et une architecture réactive hétérarchique. Le pilotage est donc ici Prédictif-réactif. Il suppose une phase préalable générant l'ordonnancement optimal. Les PI basculent ensuite en mode autonome lorsqu'une perturbation survient, et tentent de l'absorber.

5.2. APPROCHES MULTIAGENTS

Un agent est une entité (physique ou abstraite) caractérisée par le fait qu'elle est autonome au niveau prise de décision, par ses connaissances sur elle-même et sur les autres, et par sa

capacité d'agir (Finnveden et al. 2009). Plusieurs travaux proposent l'application du paradigme des Systèmes Multiagents (SMA) aux approches PDSC. Dans ces approches SMA, l'architecture de contrôle repose sur un réseau d'agents qui, en interagissant entre eux, sont capables de produire un comportement collectif complexe cohérent par rapport aux objectifs fixés (Sallez 2012a). Dans (Parunak et al. 1998), les auteurs proposent une architecture SMA, nommée AARIA (agents autonomes pour Rock Island Arsenal), pour le pilotage de l'ordonnancement. L'approche utilise ainsi une négociation par contrat entre agents afin de répartir l'allocation Produits-Ressources.

Un autre exemple d'application de SMA aux FMS est proposé dans (Maione et al. 2003). Le système de contrôle se compose d'agents « produits » et d'agents « ressources » tous deux dotés de capacités décisionnelles. Ils sont capables de communiquer entre eux et d'utiliser une approche multicritère pour réaliser l'allocation Produits-Ressources.

Dans (Vrba et al. 2006), les auteurs proposent une architecture de pilotage à base d'agents utilisant la technologie RFID, les produits étant représentés par un ensemble d'agents « pièces ». Ceux-ci négocient avec les ressources de production pour assurer l'exécution de leurs listes de services à obtenir.

Dans (Tehrani Niknejad et al. 2010), deux types d'agents sont considérés pour le pilotage temps-réel d'un FMS : les agents physiques représentant les tâches à réaliser, les ressources et les procédés de fabrication ; et les agents informationnels. Ces derniers régissent les protocoles de négociation et de prise de décision pour l'allocation des opérations de production (Figure 30).

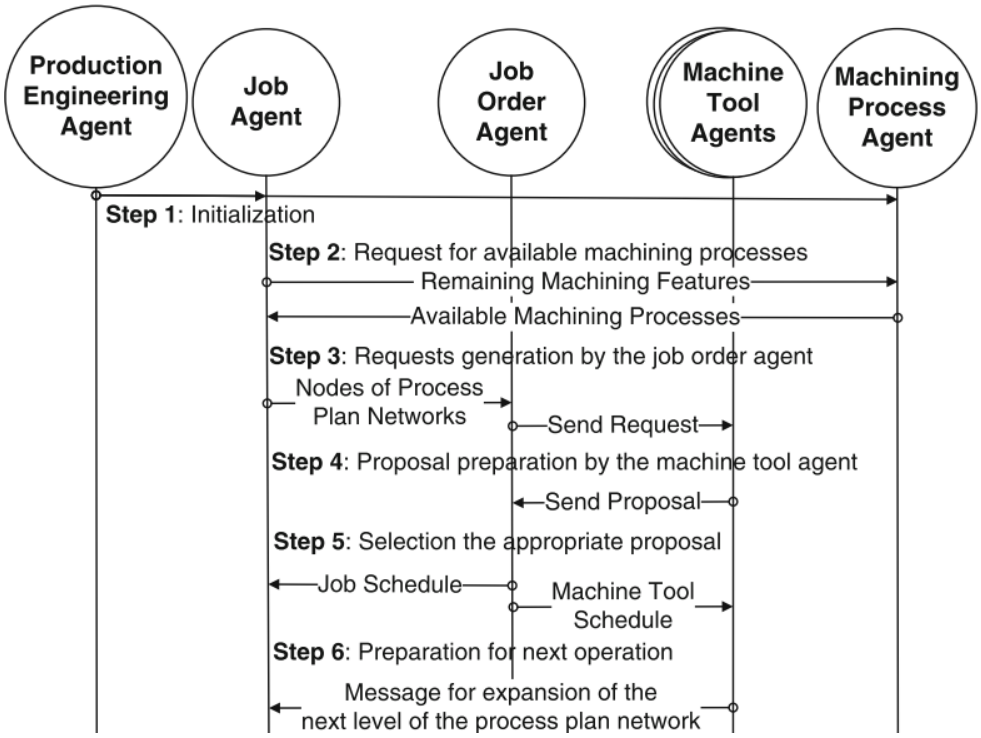


Figure 30 Exemple de négociation entre agents (Tehrani Niknejad et al. 2010)

Plus récemment, dans (Morariu et al. 2014), les auteurs proposent un système multiagents utilisant la plateforme JADE (*Java Agent Development Environment*), couplé à un réseau de neurones artificiels (ANN). L'agent WIP (Work-in-Process agent) contient les informations d'identification du produit, son statut, sa localisation dans l'atelier, et la liste des opérations à effectuer. L'ANN est utilisé dans le cadre d'une prise de décision locale pour la sélection de la prochaine ressource (allocation).

5.3. APPROCHES HOLONIQUES

Les systèmes holoniques définis par (Koestler 1967) considèrent un « holon » comme une entité considérée à la fois comme un « tout », constitué d'autres entités, et une partie d'un ensemble, apte à coopérer avec d'autres holons. Les concepts associés à l'architecture de référence HMS (Holonic Manufacturing Systems) furent proposés dans (Van Brussel et al. 1998) et débouchèrent sur l'architecture PROSA (Figure 31), construite à partir de trois principaux types de holons : les holons « Order » (i.e. ordre de fabrication), les holons produits ou « Product », et les holons « Ressource ». Chacun d'entre eux est responsable d'un aspect du contrôle de la production. Deux principales caractéristiques peuvent être associées aux HMS. La première est l'intégration des produits ou des pièces de fabrication (ou produits semi-finis), en tant qu'entités de contrôle au sein du système décisionnel distribué. La seconde est l'association permanente du produit physique à son équivalent informationnel.

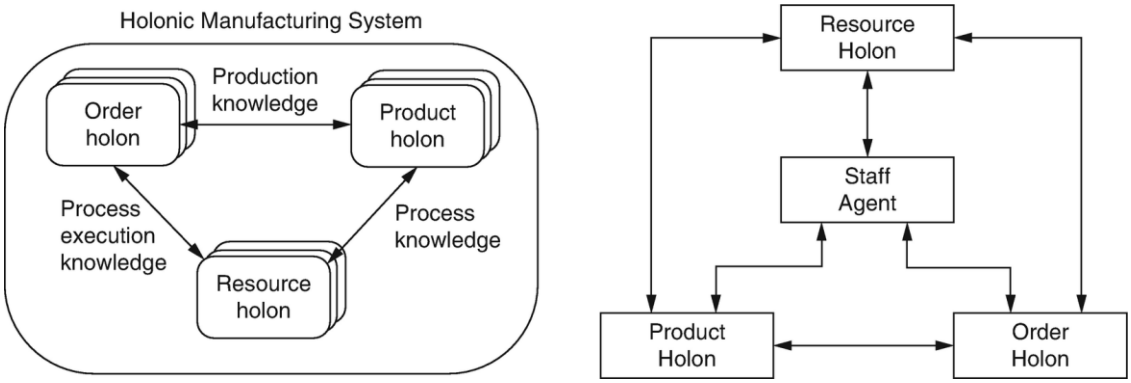


Figure 31 Les composants de l'architecture PROSA (Van Brussel et al. 1998)

Dans (Chirn et al. 2000), une Architecture (HCBA) est créée à partir de « blocs de construction intelligents ». Ces éléments sont utilisés pour les ressources et les produits formant un contrôleur virtuel au travers d'un réseau informatique. Un agent WIP (Work in Progress) est chargé de négocier avec les ressources. Les exigences, telles que la qualité ou un débit élevé de produits, sont définies par négociation avec les ressources.

Une autre contribution (Valckenaers et al. 2005) propose d'augmenter le modèle initial PROSA. L'architecture HMES (*Holonic Manufacturing Execution Syst*ems) propose d'ajouter des mécanismes de coordination et de contrôle bio-inspirés, issus par exemple du comportement des colonies de fourmis.

Dans (Leitao et al. 2006), les auteurs proposent l'architecture ADACOR (ADAptive holonic COntrol aRchitecture) appliquée à l'ordonnancement d'ateliers de type jobshop. La particularité de la solution est l'utilisation des holons « *task* » et « *operational* » pour supporter respectivement le comportement des produits et des ressources. Le pilotage prédictif-réactif est supporté par une architecture hybride (centralisée-décentralisée). Deux modes de fonctionnement sont supportés : l'un stable nommé « *Stationary state* » et un second, nommé « Transient state » autorisant un pilotage plus réactif.

Un autre HMS est proposé dans (Verstraete et al. 2008). Dans cette approche, les Holons « Order » utilisent la stigmergie : des agents « intention » sont envoyés à la recherche de ressources disponibles comme le feraient des fourmis exploratrices. Les plans de productions locaux sont actualisés continuellement. En cas perturbation, les agents fourmis la découvrent et diffusent les informations disponibles.

Une approche proposant un ordonnancement dynamique complétement distribué est

proposée dans (Leitão et al. 2008). Elle exploite un mécanisme d'ordonnancement dynamique pour associer stratégies centralisées et réparties, améliorant ainsi sa réactivité pour maîtriser l'émergence, plutôt que des algorithmes d'ordonnancement complexes et optimisés utilisés dans les approches traditionnelles.

Un modèle constitué d'un FMS avec des entités de produits et de ressources actives est présenté dans (Adam et al. 2009). Basé sur une architecture holonique, le système de pilotage proposé utilise un contrôle classique dit « explicite », ainsi qu'un second type de contrôle appelé « implicite », permettant d'influencer les entités du système via un mécanisme d'optimisation.

Dans (Pujo et al. 2009), une méthode holonique de contrôle de la production nommée PROSIS est présentée. Elle se base sur le paradigme social et le concept d'isoarchie. L'objectif de l'architecture décentralisée est d'améliorer la productivité, la flexibilité, et en particulier la réactivité. Un exemple permet d'apporter une première validation du comportement réactif pour le contrôle d'un FMS avec pannes et commandes urgentes.

Dans (Simão et al. 2009), un métamodèle est présenté pour le contrôle à l'aide de holons nommés « Smart-Product-HL ». Ces entités sont capables de négocier et d'interagir avec d'autres holons. Ainsi, ils reçoivent des informations sur les ressources et négocient pour décider des actions à exécuter. Une comparaison avec d'autres pilotage holoniques autres que le PDSC permettent de démontrer l'intérêt de l'approche.

Dans (Herrera et al. 2011), un cadre générique est présenté pour la planification et le contrôle de la production. L'approche se base sur le Modèle de Système Viable (VSM) introduit pour les systèmes de production intelligents. Basé sur des propriétés telles que l'autonomie, l'auto-organisation et l'adaptabilité, le VSM permet de modéliser et d'appliquer ces propriétés dans le cadre des PDSC. Le principal avantage de cette application est de parvenir à une cohérence entre les différents niveaux de décision.

Enfin, dans (Borangiu et al. 2015), les auteurs proposent une solution de contrôle semi-hétérarchique de la production. L'originalité de la solution réside dans un mécanisme de commutation bidirectionnelle *« switching mechanism »* du mode de fonctionnement (ordonnancement, routage), entre mode centralisé et mode décentralisé en présence de perturbations.

5.4. SYNTHESE DES TRAVAUX RELATIFS AUX SYSTEMES DE PRODUCTION PILOTES PAR LES PRODUITS

Le Tableau 5 propose une synthèse des travaux relatifs aux systèmes de production pilotés par les produits, Ils sont classés selon le concept utilisé, l'architecture du système de contrôle, la technique de résolution du problème d'ordonnancement, ainsi que du type de pilotage.

Source	Concept	Architecture	Allocation / Routage	Pilotage
Parunak et al. (1998)	SMA	Semi-hétérarchique	Négociation	Réactif
Chirn et al. (2000)	SMA/HMS	Hétérarchique	Négociation	Réactif
Maione et al. (2003)	SMA	Hétérarchique	Négociation	Réactif
Vrba et al. (2006)	SMA	Hétérarchique	Négociation	Réactif
Leitao et al. (2006)	HMS	Hybride	Négociation	
Verstraete et al. (2008)	HMS	Centralisée-Décentralisée	Négociation	Prédictif-réactif
Adam et al. (2009)	HMS	Hétérarchique	Heuristique	Réactif
Simão et al. (2009)	HMS	Hétérarchique	Négociation /Heuristique	Réactif
Sallez et al. (2009)	PDSC	Hétérarchique	Stigmergique	Réactif

Tehrani Niknejad et al. (2010)	SMA	Hétérarchique	Négociation	Réactif
Berger et al. (2010)	PDSC	Semi-hétérarchique	Stigmergique	Prédictif-réactif
Zambrano et al. (2011)	HMS	Hétérarchique	PF	Prédictif-réactif
Herrera et al. (2011)	HMS		-	-
Zbib et al. (2012)	PDSC	Hétérarchique	PF	Réactif
Reinhart et al. (2011)	PDSC	Centralisée-Décentralisée		
Pach et al. (2014)	PDSC	Centralisée-Décentralisée	PF	
Morariu et al. (2014)	SMA	Hétérarchique	ANN	Réactif
Borangiu et al. (2015)	HMS	Semi-hétérarchique	CP /Négociation	Prédictif-réactif
Gaham et al. (2017)	SMA	Hétérarchique	Contract-Net	Réactif

Tableau 5 Synthése des travaux relatifs aux systemes de production pilotés par les produits

Au travers des différentes contributions au pilotage par le produit, il est intéressant de souligner certains points importants ainsi qu'un certain nombre de tendances :

Le plus souvent, le pilotage réactif n'est utilisé qu'en cas de perturbation pour tenter d'absorber celle-ci.

La résolution du problème d'ordonnancement se limite le plus souvent au sous-problème d'allocation Produit-Ressource. L'aspect séquencement est soit ignoré, soit supposé résolu à une étape antérieure, souvent de manière centralisée (Berger et al. 2010; Pach 2013).

Les approches utilisant un pilotage prédictif-réactif supposent que suffisamment d'informations soient disponibles pour générer une planification préliminaire. Cependant, ceci n'est pas le cas dans des systèmes comprenant de réelles arrivées dynamiques de produits.

Généralement, le mécanisme réactif se limite à des négociations multiagents entre entités décisionnelles (par exemple (Parunak et al. 1998; Tehrani Niknejad et al. 2010)), ou à l'application d'heuristiques simples telles que les remontées de gradient.

Bien que ces différentes approches soient appliquées à des systèmes de production flexibles, aucun ne prend explicitement en charge les contraintes telles que les temps de configuration.

6. CONCLUSION

Ce second chapitre a permis, tout d'abord, d'introduire les CPPS, ainsi que les principes et technologies sur lesquels ils reposent. Les profondes mutations qu'ils induisent, notamment par la multiplication des nœuds décisionnels, ont été présentés. Bien que les architectures de contrôle distribuées apportent des avantages en termes de flexibilité et d'agilité, elles souffrent également de limitations dues notamment au phénomène de myopie décisionnelle. De plus, le pilotage de systèmes complexes dans des environnements dynamiques et incertains, pose des problèmes supplémentaires, comme les phénomènes d'émergence qui les rendent quasi-imprévisibles. Après avoir introduit le concept de produits « intelligents », un ensemble de travaux relatifs aux systèmes de production pilotés par les produits a été présenté.

Le prochain chapitre, présente notre contribution aux pilotages des CPPS par les produits. L'objectif est de prendre en charge à la fois des systèmes complexes et dynamiques, tout en atténuant le phénomène de myopie.

DEUXIEME PARTIE : CONTRIBUTION AUX CPPS PILOTES PAR LES PRODUITS

CHAPITRE III : APPROCHE PROPOSEE

1. INTRODUCTION

Comme nous l'avons vu précédemment, une allocation performante des tâches de production est cruciale. Si la flexibilité des FMS procure des avantages face aux environnements incertains, elle implique un accroissement de la complexité des fonctions de pilotage. En cela, les architectures distribuées semblent apporter l'agilité et la réactivité nécessaire. Toutefois, elles souffrent de limitations, celles-ci étant essentiellement dues à la myopie décisionnelle. Toute la difficulté du pilotage dynamique des CPPS réside dans l'équilibre à trouver entre une réaction rapide face aux perturbations, et le maintien d'une performance globale satisfaisante.

Ce troisième chapitre détaille la contribution à l'ordonnancement dynamique d'ateliers partiellement flexibles. La première section présentera le problème d'optimisation combinatoire qu'est l'ordonnancement des opérations de production, ainsi que l'approche qui permet un contrôle hétérarchique grâce à des produits intelligents dotés de capacités de perception et de décision. Les concepts originaux d'encapsulation, de contextes décisionnels et de stratégies décisionnelles sont également explicités.

2. FORMULATION DU PROBLEME

Cette section a pour objectif d'énoncer une formulation précise du problème abordé dans la présente thèse. Tout d'abord, et sur la base de différentes hypothèses et postulats, une formalisation mathématique du problème d'optimisation est réalisée. Puis, les besoins et spécifications qui en résultent sont recensés pour définir les contours de la solution.

2.1. HYPOTHESES

Afin de formaliser le problème, il est indispensable de recenser de manière exhaustive les hypothèses qui définissent ce dernier. Ils sont au nombre de seize :

#	Hypothèses
H_1	Tous les produits arrivent dynamiquement à partir de t_0=0
H_2	Aucune information sur les futurs produits n'est disponible avant leur arrivée
H_3	Chaque produit appartient à une et une seule famille fixée a priori
H_4	Chaque produit est supposé avoir une priorité (ou criticité) qui est fixée a priori
H_5	Chaque produit demande un ensemble d'opérations (un à la fois)
H_6	Le nombre et l'ordre des opérations demandées dépendent de la famille de produits
$H7$	Chaque ressource appartient à une et une seule cellule de production
H_8	Chaque ressource ne peut traiter qu'un seul produit à la fois
H_9	Chaque ressource peut traiter une ou plusieurs familles de produits (i.e. interopérabilité)
H_{10}	La durée des temps opératoires peut varier selon la famille du produit traité
H_{11}	Un temps de configuration supplémentaire peut être nécessaire lors d'un changement de famille
H_{12}	La durée du temps de configuration peut varier selon la famille du futur produit à traiter
H_{13}	Aucune information sur les pannes n'est disponible avant leur survenue
H_{14}	A la fin de la panne, la tâche qui était en cours au moment de l'interruption est récupérée
H_{15}	Les temps de transport inter-ressources et inter-cellules ne sont pas pris en compte
H_{16}	Chaque ressource a une zone de file d'attente d'entrée, qui est supposée être infinie

Tableau 6 Hypothèses et postulats du probléme abordé

Cette énumération des hypothèses permet de mieux appréhender la complexité du système (i.e. H_9 et H_{11}), mais aussi la nature très changeante de l'environnement considéré (i.e. H_2 et H_{13}). De manière complémentaire à cette liste d'hypothèses, il est important de proposer une formulation mathématique du problème.

2.2. FORMALISATION DU PROBLEME D'OPTIMISATION

Afin de proposer une formalisation mathématique du problème d'optimisation, un certain nombre de notations doivent être introduites.

2.2.1. PARAMETRES

P	Ensemble de produits (ou job), P = {1, 2, ..., |P|}	
F	Ensemble des familles de produits, F = {1, 2, ..., |F|}, F⊂ P	
R	Ensemble des ressources, R = {1, 2..., |R|}	
L	Ensemble des cellules de production, L = {1, 2, ..., |L|}, L ⊂ R	*(H_7)*
f_j	Famille du produit j, $f_j \in F$ et $j \in P$	*(H_3)*
p_j	Priorité du produit j, $p_j \in \mathbb{N}^*$, $j \in P$	*(H_4)*
I_j	Ensemble ordonné des opérations du produit j, $I_j = \{1,2, \ldots, \|I_j\|\}$, $j \in P$	*(H_5)*
O_{ij}	Opération i du produit j, $i \in I_j$, $j \in P$	
t_{ijr}	Durée de traitement de l'opération O_{ij} sur la ressource r, $t_{ijr} \in \mathbb{N}^*$, $i \in I_j$, $j \in P$, $r \in R$	*(H_{10})*
s_{ijr}	Durée de configuration (*setup time*) à l'opération O_{ij} sur la ressource r, $S_{ijr} \in N$	*(H_{12})*
δ_r	Ensemble des familles que peut traiter la ressource r, $\delta_r \subset F$	*(H_9)*
b_{rt}	Durée de la panne de la ressource r, à l'instant t, $b_{rt} \in \mathbb{N}$, $r \in r$, $t \in \mathbb{N}$	*(H_{13})*

2.2.2. VARIABLES

a_{ij}	Temps d'arrivée du produit j dans la cellule i, $j \in P$
w_{ij}	Temps d'attente w_{ij} avant de débuter la configuration ou le traitement de O_{ij}, $j \in P$ et $i \in I_j$
b_{ij}	Temps de début de l'opération i du produit j, $j \in P$ et $i \in I_j$
c_{ij}	Temps d'achèvement de l'opération i du produit j, $j \in P$ et $i \in I_j$
C_j	Temps d'achèvement du job j, $j \in P$
S_r	Famille actuelle qui peut être traitée sur la ressource r sans configuration, $S_r \in F$ *(H_{11})*

2.2.3. CONTRAINTES

Un certain nombre de contraintes, introduites précédemment dans la section I-3.2, est considéré ici :

H_1⇒ Arrivées dynamiques à partir de t_0=0

$$0 \leq a_{ij} \leq b_{ij} \; \forall \; O_{ij} \in I_j \tag{1}$$

H_3⇒ (fmls) Famille de produits

$$\exists! \; f_j \,/\, j \in f_j \quad \forall j \in P, \; f_j \in F \tag{2}$$

H_5⇒ (preced) Operations successives : On notera la relation d'ordre croissant entre les opérations avec le symbole « ≺ »

$$O_{ij} \prec O_{i+1j} \quad \forall \; j \in P \tag{3}$$

$$b_{i+1j} \geq c_{ij} \quad \forall \; j \in P, \forall \; i \in I_j \tag{4}$$

H6⇒ (fmls) Le nombre et l'ordre des opérations demandées dépend de la famille de produit

$$\forall j1, j2 \in R, f_{j1} = f_{j2} \Rightarrow I_{j1} = I_{j2} \quad (5)$$

H7⇒ Chaque ressource appartient à une et une seule cellule de production

$$\exists! L \subset R / r \in L \forall r \in R \quad (6)$$

H8⇒ Contraintes disjonctives : une machine ne peut traiter qu'une seule opération

$$\nexists O_{ij1}, O_{ij2} / b_{ij1} < b_{ij2} < c_{ij2} \quad \forall i \in |I_{j1}|, \forall j1, j2 \in P \quad (7)$$

H9⇒ (M_j) chaque ressource peut traiter une ou plusieurs familles de produits

$$\forall r \in R \quad |\delta_r| > 0 \quad (8)$$

H10⇒ La durée des temps opératoires peut varier selon la famille du produit traité

$$\forall j1, j2 \in R, f_{j1} = f_{j2} \Rightarrow t_{ij1r} = t_{ij2r}, \forall i \in O_f \forall r \in R \quad (9)$$

H12⇒ (ST) Un temps de configuration supplémentaire peut être nécessaire lors d'un changement de famille

$$S_r \neq f_j \Rightarrow c_{ij} > \varepsilon_{jr} * St_{ij} \quad \forall r \in R, \forall j \in P, \forall i \in I_j \quad (10)$$

H12 ⇒ ($ST_{SD, F}$) La durée du temps de configuration peut varier selon la famille du futur produit à traiter

$$\forall j1, j2 \in R, f_{j1} = f_{j2} \Rightarrow s_{ij1r} = s_{ij2r}, \forall i \in O_f \forall r \in R \quad (11)$$

2.2.4. FONCTION OBJECTIVE

Comme indiqué dans la section I-3.3, lorsqu'il s'agit de problèmes avec arrivées dynamiques, il est plus intéressant de focaliser les efforts sur le Temps d'Achèvement Moyen (TAM). Afin de calculer cette fonction objective, il convient tout d'abord définir la variable b_{ij} qui correspond au début de l'opération i du produit j :

$$b_{ij} = a_{ij} + w_{ij} \quad (12)$$

Par soucis de simplification nous définirons d_{min} comme suit :

$$d_{min} = b_{ij} + (\varepsilon_{jr} * St_{ij}) + t_{ijr} \quad (13)$$

En tenant notamment compte de H_{14}, il faut tenir compte des pannes éventuelles :

$$c_{ij} = d_{min} + \sum_{t=b_{ij}}^{t<d_{min}} b_{rt} \quad (14)$$

Conformement à H_5, le temps d'achevement est alors :

$$C_j = c_{|Ij|r} \quad (15)$$

Et le Temps d'Achèvement Moyen se calcule comme suit :

$$TAM = \sum C_j / |P| \quad (16)$$

A partir de cette formalisation du problème et des différentes hypothèses, il est possible d'en déduire un certain nombre de spécifications pour la future solution.

2.3. BESOINS ET SPECIFICATIONS

Les éléments de la figure 34 permettent de tracer les contours de la future solution.

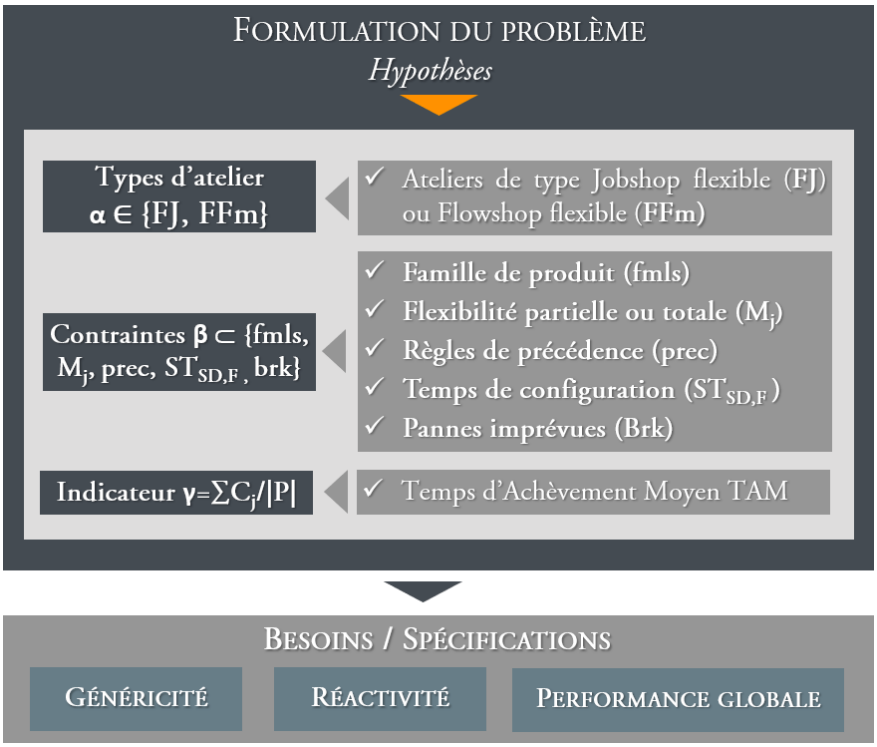


Figure 32 Formulation du probleme et besoins induits

La future solution doit nécessairement satisfaire trois grandes exigences :

1. **Besoin de généricité :** la configuration du système de production peut fortement varier (i.e. H_6 et H_7) ; la solution doit pouvoir alors être appliquée à différents types d'ateliers de production (p. ex. Flowshop ou jobshop) avec notamment, des spécificités telles que la flexibilité partielle (i.e. H_9). En plus des différents cheminements, il doit être possible, selon le cas, d'intégrer des contraintes telles que les familles de produits, les variations des temps opératoires et les temps de configuration qu'elles impliquent. Le modèle doit donc être aussi générique et indépendant que possible de l'organisation du système de production à piloter.
2. **Besoin de réactivité :** l'une des principales caractéristiques de l'environnement considéré, est sa nature fortement changeante. En particulier l'indisponibilité d'informations sur les futurs produits avant leur arrivée dans le système (i.e. H_1 et H_2), mais aussi les pannes imprévisibles (i.e. H_{13}), rend quasi-impossible toute anticipation par les méthodes classiques (i.e. proactive et prédictive). La solution doit être alors suffisamment réactive pour réagir efficacement aux nombreuses perturbations dans un environnement particulièrement incertain.
3. **Besoin de performance globale :** en plus des besoins de généricité et de réactivité, la solution doit aussi être performante. Cet objectif global se traduit dans notre cas, par une minimisation des Temps d'Achèvement Moyens TAM. Ce besoin implique cependant de maitriser le comportement émergent d'interactions aussi complexes que dynamiques.

L'approche proposée est décrite dans la section suivante. Les différents choix ayant conduit à l'élaboration de notre approche sont explicités. Une présentation détaillée permet d'expliquer comment la contribution tente de répondre simultanément et efficacement aux impératifs de généricité, de réactivité et de maintien de la performance globale.

3. APPROCHE PROPOSEE

3.1. CHOIX ET MOTIVATIONS DE L'APPROCHE

Comme le montre la Figure 33, les différents outils, méthodes et concepts originaux développés sont intimement liés. La synergie qui résulte de leur combinaison permet d'obtenir une solution globale.

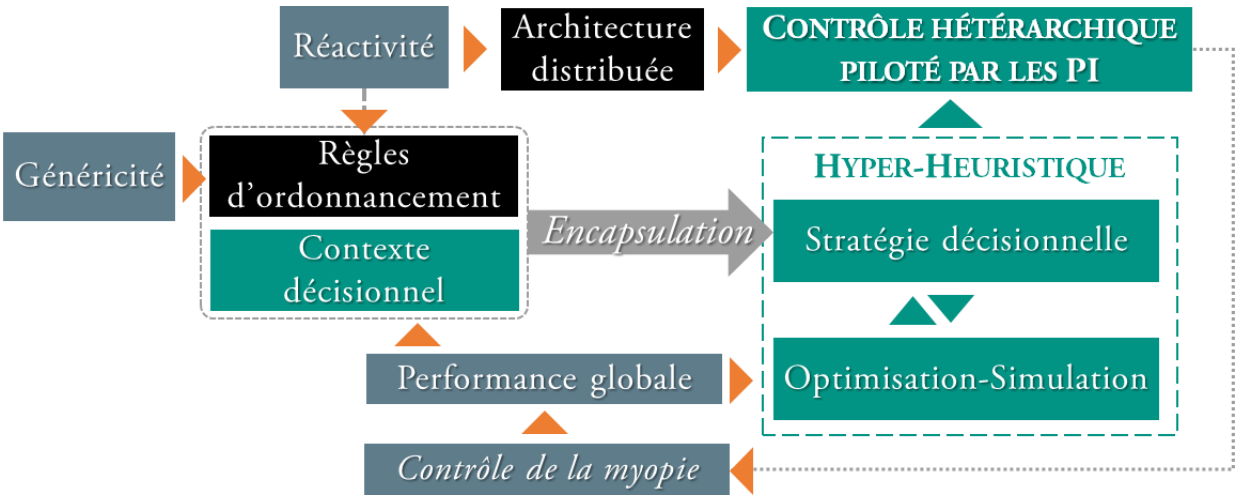


Figure 33 Interaction entre les composants de l'approche proposée et adéquation avec les besoins

La solution proposée apporte une réponse aux besoins explicités dans la section III-0 :

1. **Réactivité de la solution :** l'aspect fortement dynamique et incertain de l'environnement a conduit à privilégier un pilotage réactif. Celui-ci est supporté par une architecture de contrôle purement distribuée. Le pilotage hétérarchique repose ainsi exclusivement sur des PI dotés de capacités de perception et de décision. Ce pilotage par les produits permet d'augmenter considérablement la flexibilité et l'agilité du système par rapport à d'autres modèles de contrôle (cf. II-5). Afin de résoudre chacun des deux sous-problèmes que sont, d'une part, l'allocation et, d'autre part, le séquencement, le processus décisionnel consiste essentiellement à sélectionner puis à appliquer des règles d'ordonnancement (i.e. d'allocation et de séquencement) en tirant parti de la grande réactivité de ces heuristiques (cf. I-4.2 et I-4.4).
2. **Généricité de la solution :** un autre avantage des règles d'ordonnancement est leur généricité. En effet, bien que relativement simples, elles peuvent être appliquées dans un grand nombre de situations. Les PI développés sont des entités autonomes cherchant à obtenir une succession d'opérations et ce, relativement indépendamment de l'organisation des ateliers. Le concept de stratégie décisionnelle développé ici, permet d'obtenir un PI qui utilise un ensemble de règles dans le but de résoudre une succession de problèmes d'ordonnancement.
   Enfin, Il convient également noter la présence de nombreux paramètres permettant d'adapter l'approche en agissant directement sur le répertoire de règles (p. ex. en en implémentant de nouvelles). Le comportement du PI est ainsi suffisamment générique et paramétrable, pour permettre d'explorer un très grand nombre de schémas d'allocation de règles selon différentes configurations et caractéristiques de SP.
3. **Performance globale :** comme évoqué précédemment, le cœur de la solution repose sur la combinaison de plusieurs règles.

Une première solution pour améliorer la faible performance globale des DR est d'agir sur le mécanisme opérant leur sélection. Pour cela, les PI se doivent de considérer le contexte induit par l'environnement lors de la sélection d'une règle (cf. III-3.2.1). Plusieurs travaux ont en effet démontré que l'autonomie et l'intelligence de la prise de décision locale sont particulièrement importantes dans une approche de contrôle par le produit (Pannequin et al. 2009). Dans cette optique, le concept de « contexte décisionnel » développé permet de conditionner la sélection des règles à l'état d'une ou plusieurs variables spécifiques. Cette prise en compte d'un ou plusieurs indicateurs, permet d'agir sur la pertinence des décisions prisent par les PI (cf. III-3.4.1). Des « stratégies décisionnelles » permettent d'encapsuler les contextes décisionnels et les règles d'ordonnancement. Une hyper-heuristique permet alors la génération de la meilleure stratégie décisionnelle possible.

L'autre voie permettant d'améliorer la performance globale est un couplage original optimisation-simulation. En effet, l'utilisation de simulations permet de réduire le phénomène sous-jacent de myopie (II-3.3.3) lié au choix d'une architecture distribuée. Ce couplage assure une évaluation du comportement émergeant des interactions entre produits et ressources, et permet de s'affranchir des problèmes d'optimalité locale (cf. II-3.3.3).

La modélisation du système de production cyber-physique destiné à soutenir notre solution est présentée dans la section suivante.

3.2. MODELISATION DU CPPS

Afin de modéliser le CPPS, il convient, tout d'abord, d'identifier les principaux composants du niveau physique du SP à piloter. La Figure 34 permet d'illustrer un exemple de processus de production constitué de successions d'étapes ordonnées. A chaque étape une cellule regroupe, une ou plusieurs ressources en fonction des similitudes entre les opérations qu'elles peuvent traiter. Chaque cellule possède ainsi une zone d'attente, à laquelle s'ajoutent les différentes files d'attente propres aux ressources.

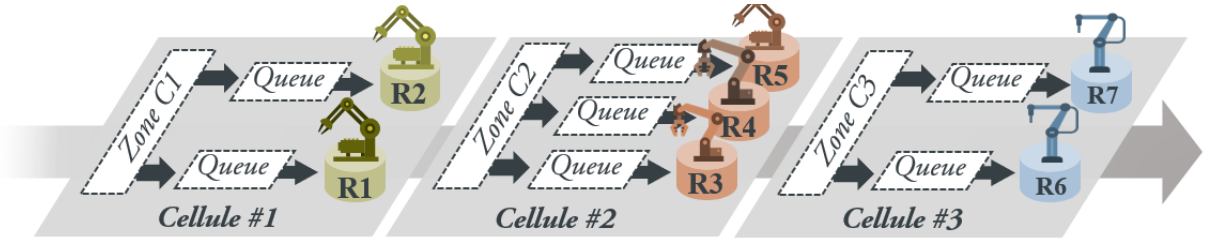


Figure 34 Exemple de représentation des cellules de production (Niveau physique)

L'architecture de pilotage étant purement hétérarchique, deux entités sont considérées : les produits et les ressources. Les entités informationnelles, associées aux produits et ressources, sont accueillies par un niveau logiciel, comme illustré par la Figure 35. Ce niveau logique est en quelque sorte superposé au niveau physique.

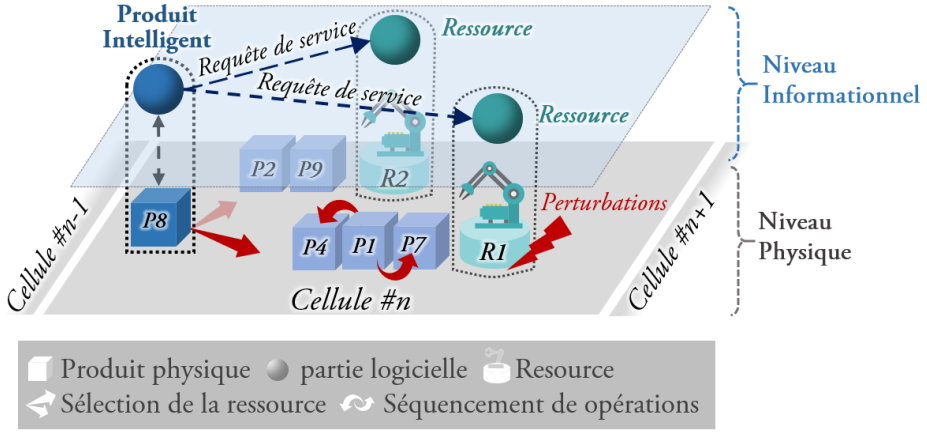


Figure 35 Les couches informationelle et physique du CPPS modélisé

Il faut noter que le formalisme retenu pour les différents graphes (i.e. d'états, de classe et d'activité) est le Langage UML (*Unified Modeling Language*) dans sa version 2.0. Cette spécification[2] UML 2.0 est de loin la plus riche (plus de 600 pages de documentation), mais aussi la plus claire et la plus compacte des versions (Dan et al. 2005). Elle est particulièrement adaptée à la description du fonctionnement dynamique des composants de notre modèle.

3.2.1. PI AVEC PERCEPTION DU CONTEXTE DECISIONNEL

Chaque produit intelligent peut communiquer et prendre des décisions afin d'accomplir son processus de fabrication (p. ex. en cherchant à obtenir successivement 3 opérations O_1, O_2, puis O_3). Pendant son cheminement à travers le système de production, le PI peut être dans quatre états différents (Figure 36) :

Recherche : indique que le processus de fabrication du produit est en cours. Le produit est alors en train d'accomplir le processus décisionnel pour ordonnancer sa prochaine opération. Il peut être soit dans la zone de chargement de la cellule (i.e. sous-problème d'allocation), soit dans la file d'attente d'une ressource (i.e. sous-problème de séquencement).

En attente : signifie que le produit est à présent dans la file d'attente d'une ressource en attente d'une opération.

En traitement : Indique que le produit est en cours de traitement par une ressource.

Achevé : signale que le produit a obtenu a achevé toutes les opérations de son processus de fabrication.

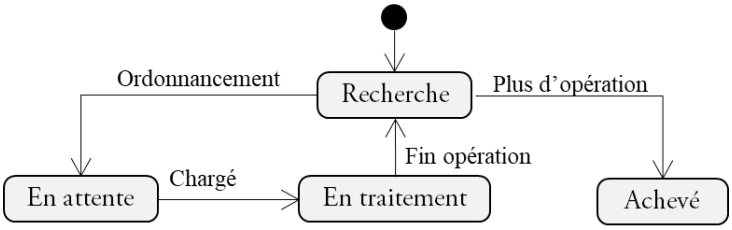


Figure 36 Diagramme d'états du PI

Afin de rendre le PI autonome et réactif, une architecture interne décrite à la Figure 37 est proposée. Elle se compose de quatre modules :

Communication : permet de gérer les interactions pour permettre au PI de communiquer et

[2] https://www.omg.org/spec/UML/

d'interagir avec son environnement.

Perception : traite les données entrantes afin d'identifier le contexte décisionnel actuel.

Décision : à partir du contexte actuel, choisit la règle d'ordonnancement adéquate.

Action : traduit la règle sélectionnée par le module décision en un ensemble d'actions (p. ex. requêtes ou envoi d'instructions).

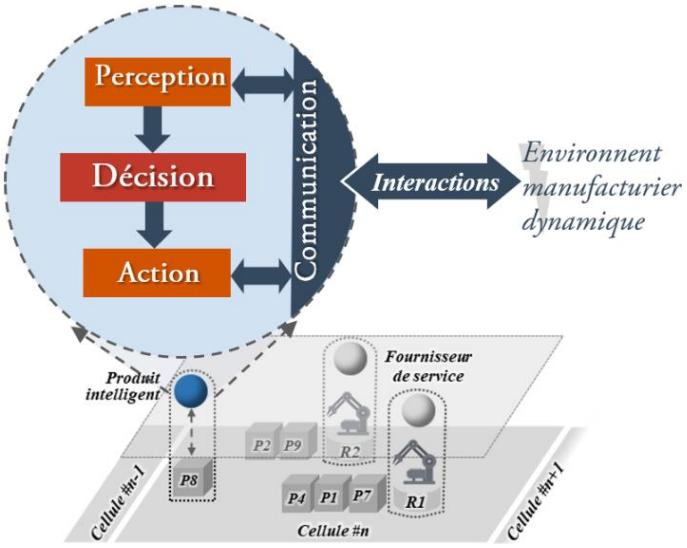


Figure 37 Architecture interne du produit intelligent

3.2.2. ENTITE INFORMATIONNELLE « RESOURCE »

Comme explicité précédemment, seuls les produits peuvent prendre des décisions. Les ressources sont considérées comme des entités sans capacité décisionnelle. Chaque ressource peut réaliser une opération sollicitée par un produit selon la contrainte d'interopérabilité, avec un temps opératoire, et éventuellement de configuration, spécifiques. Toutefois, la ressource répond aussi aux requêtes des produits en communicant des informations telles que son état, le contenu de sa file d'attente ou encore ses caractéristiques. Elles suivent les instructions fixées par les PI. En effet, le contenu de la file d'attente (allocation de la ressource), ainsi que l'ordre de ces opérations (séquencement des opérations) sont explicitement ordonnés par les PI. En fonction des différents évènements qui peuvent survenir, une ressource R peut prendre 4 états :

Libre : indique que la ressource est disponible et opérationnelle.

Setup : signale que la ressource effectue une opération de configuration en vue de préparer le traitement du prochain produit.

Process : signifie que la ressource a chargé un produit et est en train de le traiter.

En panne : indique une indisponibilité machine en cours. Le pseudo état « H » mémorise l'état de la ressource au moment de la panne, permettant ainsi une future récupération.

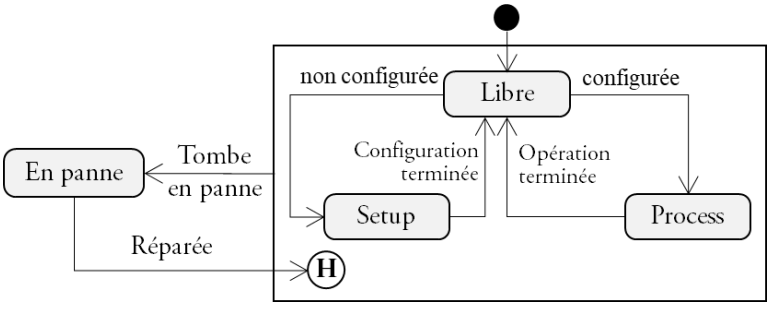


Figure 38 Diagramme d'états de l'entité informationelle « resource »

La prochaine section décrit en détails les concepts de contexte décisionnel implémenté au niveau du PI.

3.3. CONTEXTE DECISIONNEL

Pour permettre le contrôle hétérarchique, les PI se voient dotés de capacités de perception et de décision. Le PI doit être capable de percevoir le contexte décisionnel et de sélectionner dynamiquement les règles d'ordonnancement adhoc. Parmi les multiples définitions du dictionnaire[3] pour les termes « contexte » et « décision », nous retiendrons les suivantes :

Contexte : « ensemble des circonstances dans lesquelles se produit un événement, se situe une action ».

Décision : « acte par lequel quelqu'un opte pour une solution, décide quelque chose ; résolution, choix ».

Davantage en lien avec le domaine industriel qui nous intéresse ici, le contexte décisionnel permet de caractériser l'état de l'environnement manufacturier à un instant donné. Il doit être pris en compte dans le processus de prise de décision pour l'allocation et / ou le séquencement des travaux et, plus globalement, pour améliorer le comportement du système de production (Bouazza et al. 2019). A cette définition, nous ajouterons la notion de « variable discriminante » dont la variation de la valeur est calculée objectivement et permet de différencier deux contextes décisionnels distincts. L'exemple illustré par la Figure 39, montre qu'il peut être, sous certaines conditions, plus intéressant de combiner deux règles en passant de l'une à l'autre. La variable discriminante permet ainsi de discerner quand ce changement doit être opéré.

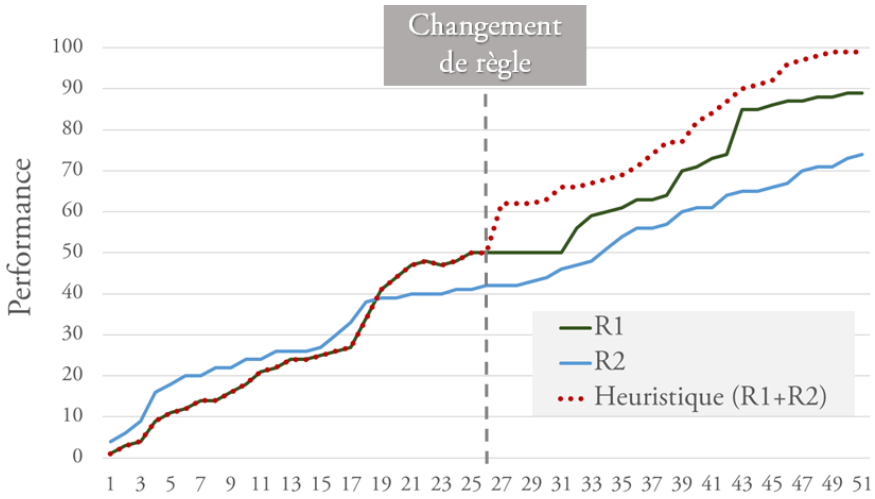


Figure 39 Performances de 2 règles VS heuristique combinant ces même règles

3.4. ENCAPSULATION DES REGLES DANS LES STRATEGIES DECISIONNELLES

Afin de permettre aux PI de prendre en charge explicitement les différents contextes décisionnels, le concept de « stratégie décisionnelle » est ici proposé. Basé sur un modèle original d'encapsulation des règles d'ordonnancement, il permet de déterminer les décisions et comportements du produit. Comme illustré par la Figure 40, le modèle générique d'encapsulation compte 4 niveaux :

▪ **Niveau-0 :** c'est le niveau le plus bas. Il est composé des règles d'ordonnancement dites « atomiques ».

[3] https://www.larousse.fr/

– ***Règle d'allocation de ressource (MSR) :*** ou « Machine Selection Rule », permet au PI de sélectionner une ressource parmi un ensemble de machines candidates (p. ex. LQE pour « *Less Queued Elements* »).

– ***Règle de séquencement (SR)*** : ou « *Sequencing Rule* », est utilisée pour définir l'ordre de passage des opérations dans une ressource donnée (p. ex. FIFO pour « *First in First Out* »).

▪ **Niveau-I :** ce niveau est composé de règles, auxquelles on ajoute des intervalles.

– ***Heuristique d'allocation de ressources (HAR)*** : composée d'une MSR avec deux bornes limites. Ces intervalles correspondent aux plages de valeurs de la variable discriminante allouant l'utilisation de la règle. Par exemple, si v1 ∈]0,5] utiliser MSR#4.

– ***Heuristique de séquencement (HS) :*** similairement à l'HAR, elle est composée d'une règle, dans ce cas des SR, auxquelles s'ajoutent un intervalle de valeurs conditionnant le déclenchement de la règle. Par exemple, si v2 ∈]3,7] utiliser SR#2.

▪ **Niveau-II :** ce niveau permet de regrouper les différentes heuristiques selon le sous-problème d'ordonnancement traité par la politique.

– Politique d'allocation de ressource (PAR) : composée d'une ou de plusieurs heuristiques. Selon la valeur de la variable discriminante nommée v1, la MSR correspondante est appelée (p. ex. si v1 ∈]0,5] utiliser MSR#4, si v1 ∈] 5, 8] utiliser MSR#2, etc).

– Politique de séquencement (PS) : composée d'une ou plusieurs HS. Une seconde variable discriminante v2 permet de définir la règle à appeler (p. ex. si v2 ∈]0,3] utiliser SR#3, si v2 ∈]3,7] utiliser SR#1, etc).

A noter que dans le cas de plusieurs heuristiques au sein d'une même politique, les intervalles doivent être contigus pour couvrir l'ensemble des valeurs que peut prendre la variable discriminante.

▪ **Niveau-III :** c'est le niveau d'encapsulation le plus élevé. Il représente la Stratégie décisionnelle dans sa totalité. La stratégie décisionnelle regroupe deux politiques, une politique pour chacun des sous-problèmes d'ordonnancement, à savoir l'allocation de ressources et le séquencement des files d'attentes.

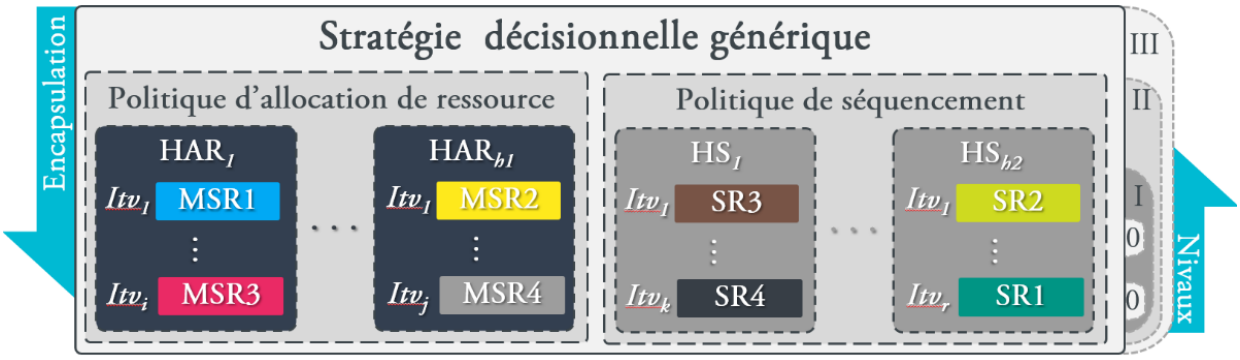


Figure 40 Modèle générique d'encapsulation des règles dans la stratégie décisionnelle

Selon ce modèle générique, on peut distinguer 4 paramètres principaux (Tableau 7). Ils permettent de définir le nombre d'heuristiques dans chaque politique, les variables discriminantes utilisées pour le déclenchement de celles-ci, et également les espaces à discrétiser.

Paramètre	Signification
h_1	Nombre d'heuristiques HAR dans la politique PAR
v_1	Variable discriminante pour la sélection des heuristiques HAR
h_2	Nombre d'heuristiques SR dans la politique PS
v_2	Variable discriminante pour la sélection des heuristiques HS

Tableau 7 Principaux paramétres du modéle de stratégie.

3.4.1. STRATEGIE PROPOSEE BASEE SUR LES INTERVALLES "ITV"

La contribution principale de cette thèse est une stratégie basée sur les intervalles et nommée « ITV » (Figure 41). Cette stratégie permet de décrire le comportement du PI avec un maximum de finesse en implémentant un grand nombre de contextes décisionnels. Pour cela, il existe une HAR pour chaque cellule (h_1=|L|), et une HS pour chaque ressource (h_2=|R|) (voir Figure 42). Pour les variables discriminantes, on considère le nombre de produits dans la cellule pour HAR, $v_1 \in]0, +\infty[$. Pour les HS, c'est le nombre de produits dans la ressource qui est considéré, $v_2 \in]0, +\infty[$. Ainsi, le produit peut adapter son comportement non seulement en fonction de la cellule et de la ressource dans laquelle il se trouve, mais aussi selon le nombre de produits qu'elle contient. Le tout s'effectuant de façon quasi-instantanée.

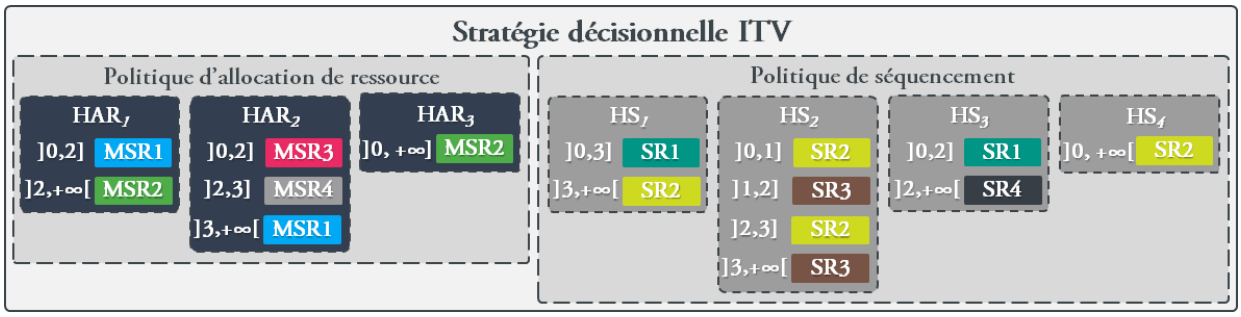


Figure 41 Exemple d'encapsulation des régles pour la stratégie proposée « ITV »

3.4.2. AUTRES VARIANTES

En plus de la stratégie ITV, un certain nombre de variantes est également proposé (Figure 42), allant de la version à peine altérée CELL-ITV, à la plus basique RC combinaison de deux règles. Le but est ici de permettre de comparer l'approche proposée avec d'autres schémas d'allocation de règles, et ainsi de mieux juger de la pertinence du modèle proposé.

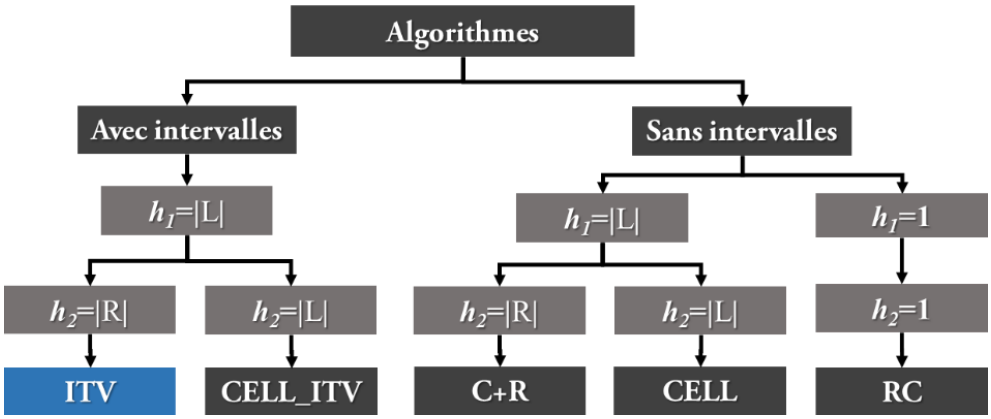


Figure 42 Classification de la stratégie proposée ITV et les variantes

Notons que tous les schémas d'allocation des variantes sont inclus dans les solutions d'ITV ; en effet, cette stratégie peut « coder » n'importe quel schéma des variantes. Cette inclusion des espaces de solutions est illustrée par la Figure 43. Par exemple, pour coder RC avec ITV, il suffit pour cela de choisir deux règles MSR+SR, chacune avec un seul intervalle de]0, +∞ [. Ainsi, les règles seront toujours appelées quelles que soient les valeurs des variables discriminantes.

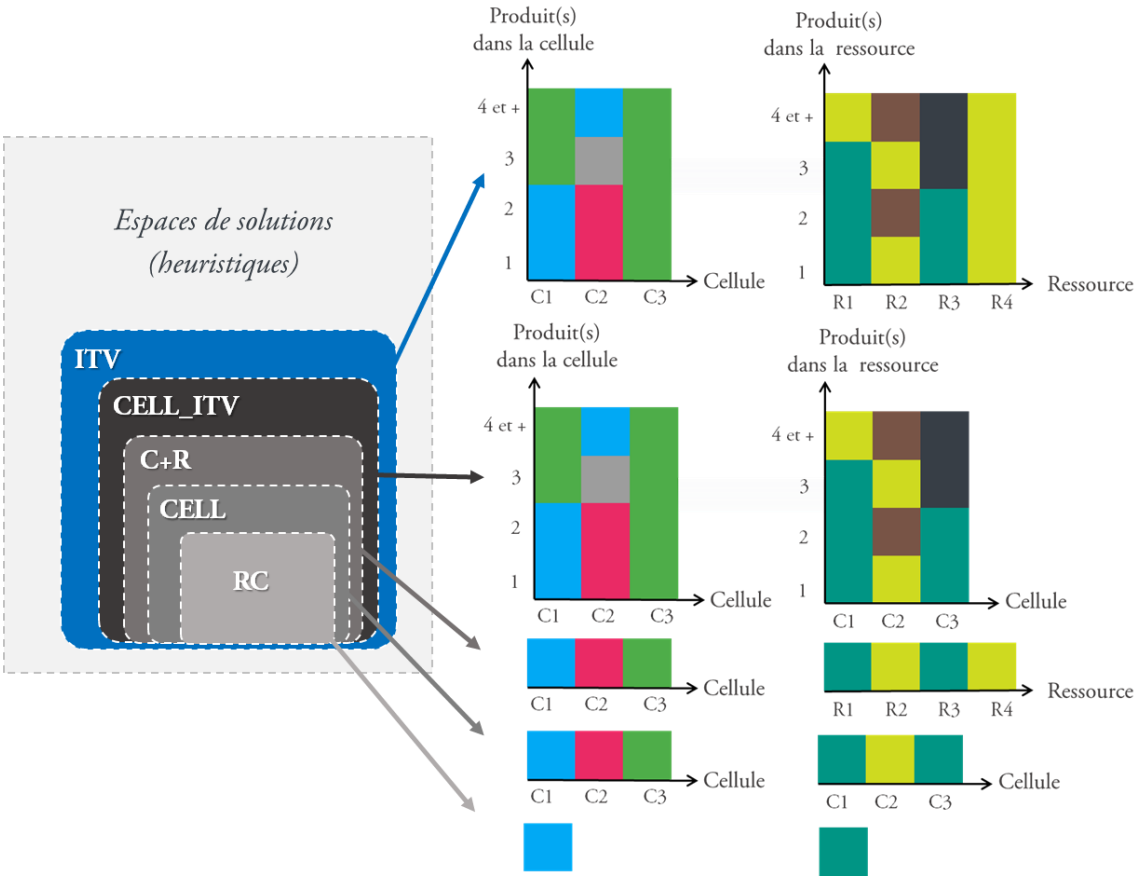


Figure 43 Les différents schémas d'allocation de règles

▪ ***Stratégie « CELL_**ITV **» :*** cette première variante est assez similaire à ITV (Figure 44). Elle est également basée sur des intervalles décomposant les valeurs de deux variables discriminantes v_1 et v_2. La politique d'allocation de ressources est rigoureusement la même que pour ITV (h_1=|L|), La différence réside dans la seconde politique ; pour CELL-ITV, il existe une HS par cellule (h_2=|L|), et non pas par ressource comme c'est le cas pour ITV.

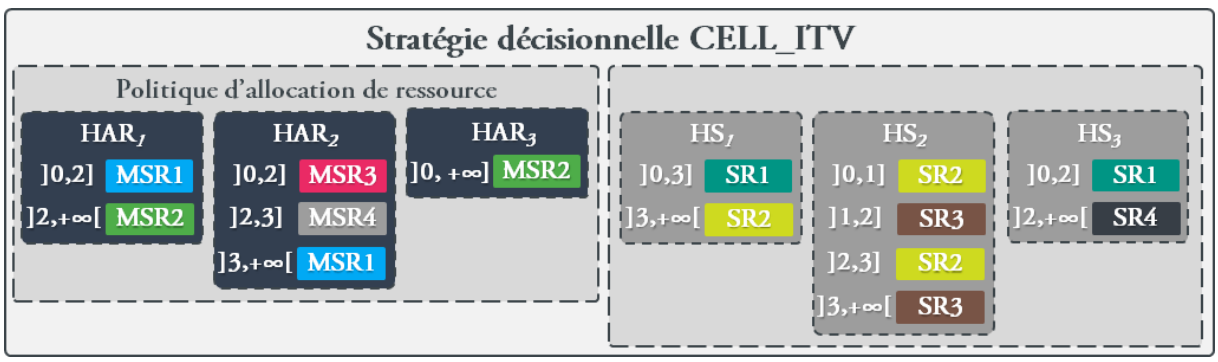


Figure 44 Example d'encapsulation des régles pour la variante « CELL_ITV »

▪ ***Stratégie « C+R » :*** cette seconde variante n'utilise pas de variable discriminante (i.e. sans

intervalle). Une règle d'allocation unique MSR est utilisée par le produit selon la cellule où il se trouve (h_1=|L|). Une règle SR distincte est allouée à chaque ressource (h_2=|R|). Ainsi, la résolution de l'allocation se fait en fonction de la cellule dans laquelle se trouve le PI, tandis que le problème de séquencement se fait selon la ressource (i.e. *« Cell + Ressource »*).

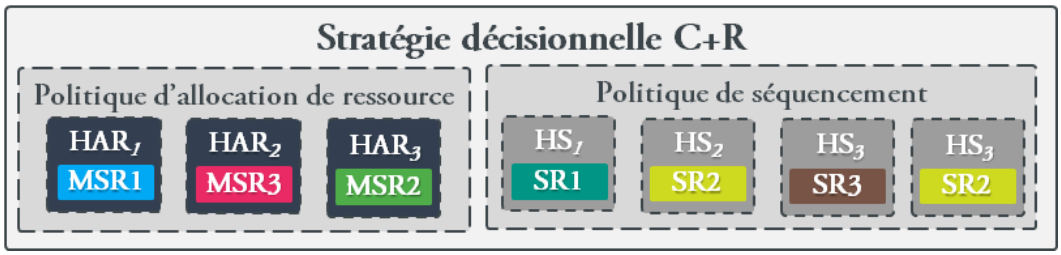


Figure 45 Example d'encapsulation des règles pour la variante « C+R »

▪ ***Stratégie « CELL » :*** tout comme la stratégie C+R, cette variante n'utilise pas de variable discriminante (i.e. aussi sans intervalle). La politique d'allocation est la même que pour C+R (h_1=|L|). En revanche, les règles SR sont propres à chaque cellule (h_2=|L|). De ce fait, le comportement est défini uniquement par la cellule dans laquelle se trouve le PI.

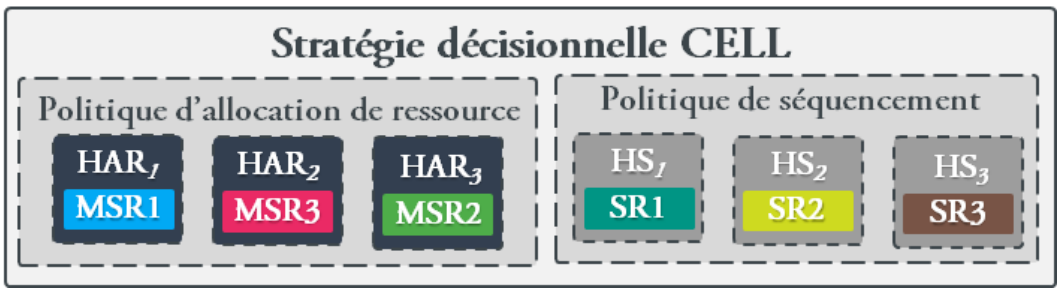


Figure 46 Example d'encapsulation des règles pour la variante « CELL »

▪ ***Combinaison de règles « RC » :*** notée RC pour *« Rules Combination »*, cette dernière variante est la plus élémentaire. Elle consiste en une simple combinaison de deux règles, h_1=1 et h_2=1. Elles sont systématiquement utilisées par les PI indépendamment de toute notion de contexte décisionnel.

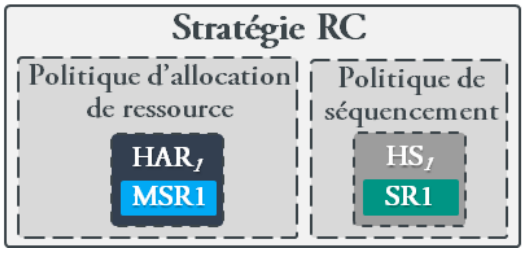


Figure 47 Example de règles MSR et SR dans une combinaison « RC »

3.5. COUPLAGE OPTIMISATION-SIMULATION

Comme expliqué précédemment, l'espace de recherche est composé des différentes stratégies décisionnelles qui déterminent le comportement des PI. Etant donné le grand nombre de combinaisons possibles, l'hyper-heuristique proposée combine l'optimisation des schémas d'allocation de règles avec un outil de simulation pour évaluer la performance globale de ces stratégies.

3.5.1. MODULE DE SIMULATION DU SYSTEME DE PRODUCTION DISTRIBUE

La simulation du système de production est un élément crucial de la solution proposée. Elle permet d'évaluer le comportement des PI selon différentes stratégies décisionnelles.

▪ **Données en entrée :** La simulation permet d'évaluer la performance d'une stratégie donnée, sur une configuration d'un système de production donné, avec un scenario spécifique. Pour cela, le module simulation comporte en entrée, en plus de la stratégie décisionnelle, les deux éléments suivants :

– ***Données de configuration :*** elles contiennent les informations relatives aux ressources (organisation des cellules, temps opératoires, temps de configuration, etc.), mais aussi certaines données relatives aux produits, essentiellement en lien avec les différents processus de fabrication.

– ***Scenario de production :*** il décrit les événements liés aux arrivées des produits (p. ex. famille, priorité, etc.) auxquels peuvent s'ajouter des événements liés aux pannes (p. ex. ressource concernée et durée de la perturbation).

▪ **Simulation des composantes informationnelles :** afin de permettre l'évaluation des stratégies décisionnelles, les comportements des deux composantes informationnelles que sont Ressource et PI sont simulés et décrits ci-après (cf. III-3.2).

– ***Comportement de la ressource :*** il consiste essentiellement en le traitement des produits dans la file d'attente. La ressource entame une opération de configuration si la famille du produit le nécessite. Dans le cas contraire, elle commence aussitôt l'opération sur le produit en tête de file et envoie un message au PI concerné à la fin de l'opération. Si une panne intervient, elle interrompt toute opération et se met en attente d'un message indiquant la fin de la panne. A l'arrivée du message indiquant la réparation, la ressource peut récupérer son état antérieur à la survenue de la panne et poursuivre. A noter qu'indépendamment des opérations de configuration et de traitement, la ressource met à jour sa file d'attente selon les instructions des PI (Figure 48).

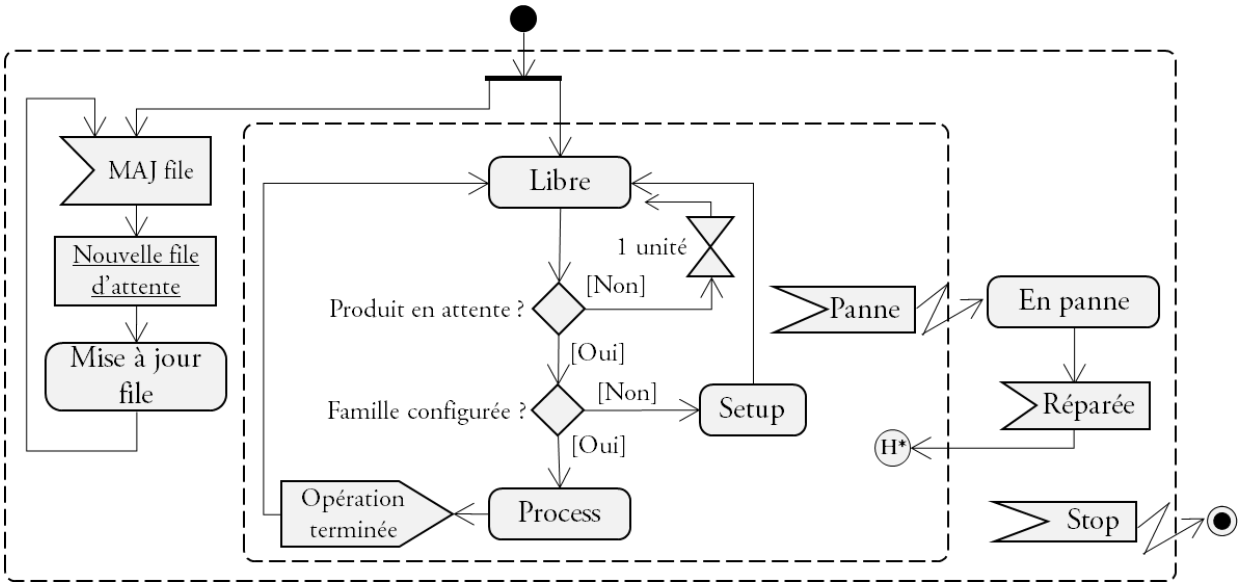


Figure 48 Diagramme d'activité de la composante informationelle de la ressource

– ***Comportement du PI :*** compte tenu de ses capacités décisionnelles, le comportement du PI

est plus complexe que celui de la ressource (Figure 49). L'enchainement des opérations à obtenir est défini au préalable et conditionné par la famille de produits à laquelle il appartient. En fonction de la progression dans cette liste d'opérations, il rejoint la cellule correspondante. Après avoir requis puis collecté les données de la cellule, il recherche une ressource en fonction de la politique d'allocation de ressources de sa stratégie décisionnelle. Selon le contexte courant, une règle d'allocation de ressources est appliquée pour sélectionner une ressource parmi l'ensemble des ressources candidates. Après avoir rejoint la ressource sélectionnée, la seconde politique relative au séquencement est utilisée. La règle de séquencement correspondante au second contexte est appliquée. Le PI indique la nouvelle file d'attente (i.e. nouvelle séquence d'opérations à traiter) à la Ressource. Le produit se met alors en attente de la fin de l'opération ; une fois l'opération terminée, s'il reste des opérations à réaliser, le PI rejoint la nouvelle cellule et réitère le processus décisionnel. Dans le cas contraire, le produit est considéré achevé et passe dans son état final.

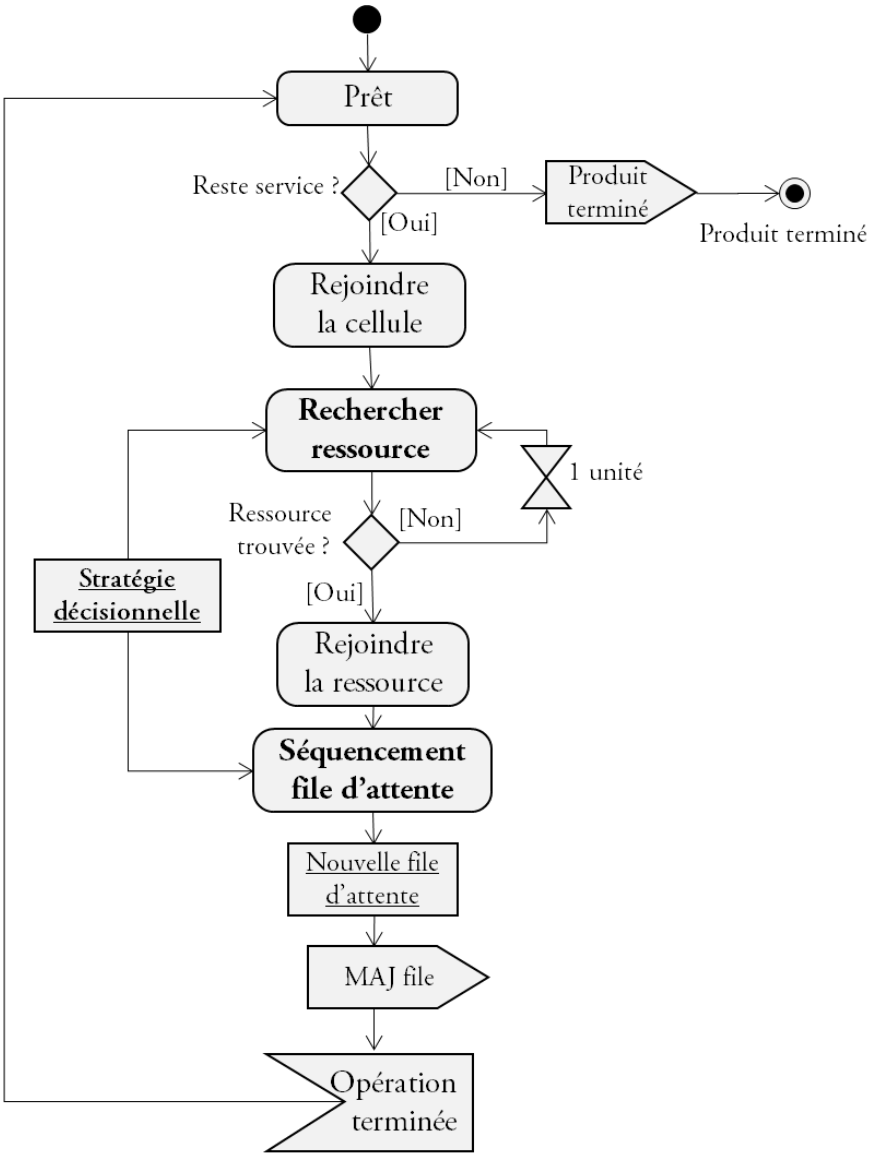


Figure 49 Diagramme d'activité générique du PI développé

▪ **Déroulement du scenario de production :** La Figure 50 montre les principales étapes du déroulement de la simulation. Elle se décompose en 3 phases :

- ***Phase I :*** c'est la phase d'initialisation. Après lecture des données de configuration, les ressources qui constituent le système de production sont paramétrées puis lancées p. ex. les temps opératoires, les temps de configuration, ou encore les contraintes d'interopérabilité).
- ***Phase II :*** c'est la phase d'exécution du scenario de production. Les différents événements séquentiels qui le composent sont ainsi déclenchés au fur et à mesure de l'avancement de la simulation. Soit l'évènement concerne une arrivée de produit et un PI est créé, paramétré puis lancé. Soit, l'événement est une panne machine et un message informant la ressource concernée est envoyé en indiquant la durée de l'indisponibilité.
- ***Phase III :*** cette dernière phase consiste à attendre que tous les PI aient indiqué qu'ils ont achevé leur processus de production. Les données (par exemple, le nombre d'opérations de configuration, le temps d'exécution), mais également la fonction objective TAM, sont calculés et la simulation est terminée.

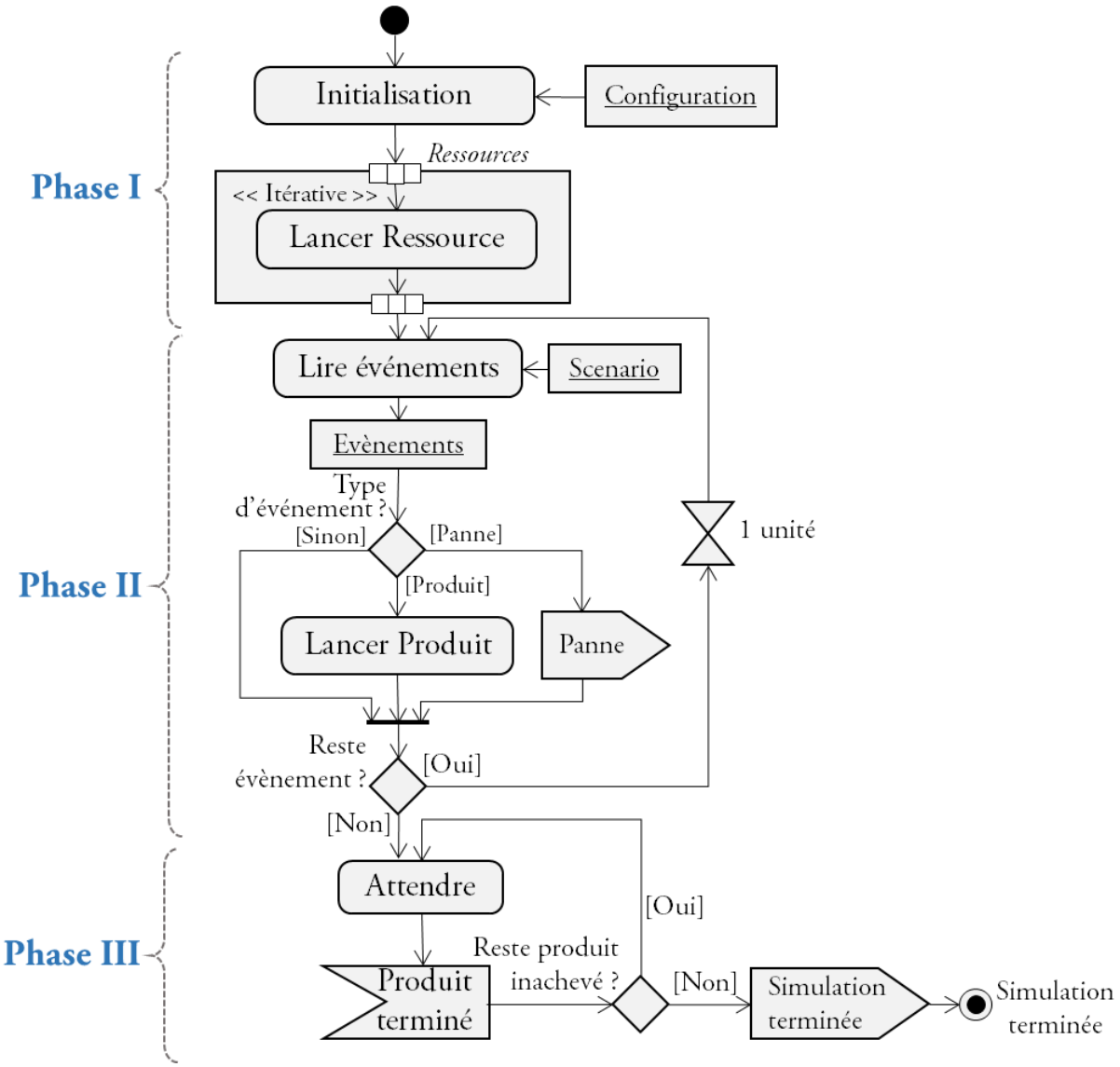


Figure 50 Diagramme d'activité de la simulation du systéme de production

3.5.2. MODULE D'OPTIMISATION DES STRATEGIES DECISIONNELLES

Un algorithme génétique est utilisé pour parcourir l'espace de recherche constitué de l'ensemble des stratégies possibles, la simulation servant à évaluer chaque individu d'une

population de stratégies. L'optimisation se fait progressivement par croissement et mutation des stratégies avec les meilleures évaluations. Compte tenu de la nature particulière d'encodage des solutions (cf. III-3.4), l'optimisation est obtenue progressivement par différentes procédures de croisement et de mutation adaptées au modèle d'encapsulation. Le but étant de maintenir la contrainte d'intégrité pour les nouvelles stratégies. Ces procédures peuvent également être combinées de différentes manières pour guider l'exploration et contrôler l'équilibre entre exploration et convergence :

▪ **Procédures de croisement :** comme le montre la Figure 51, trois procédures de croisement ont été développées :

– ***Cross_S1 :*** croise deux stratégies (Niveau-III) en prenant une politique de chacune des deux stratégies parentes.

– ***Cross_S2 :*** permet d'hybrider plus finement deux stratégies (Niveau-III). Les politiques sont fusionnées en utilisant la procédure de croisement de Niveau-II inférieur CrossP.

– ***Cross_P :*** permet de croiser deux politiques (Niveau-II), en choisissant, selon une règle probabiliste, une heuristique des deux politiques parentes, formant ainsi une nouvelle politique.

Il est à noter que, compte tenu de la contrainte de contiguïté des intervalles, il n'est pas possible de sélectionner au hasard des heuristiques (Niveau-I), au risque de ne plus garantir la continuité des intervalles de la politique. Il est alors impératif de garantir cette intégrité à tout moment.

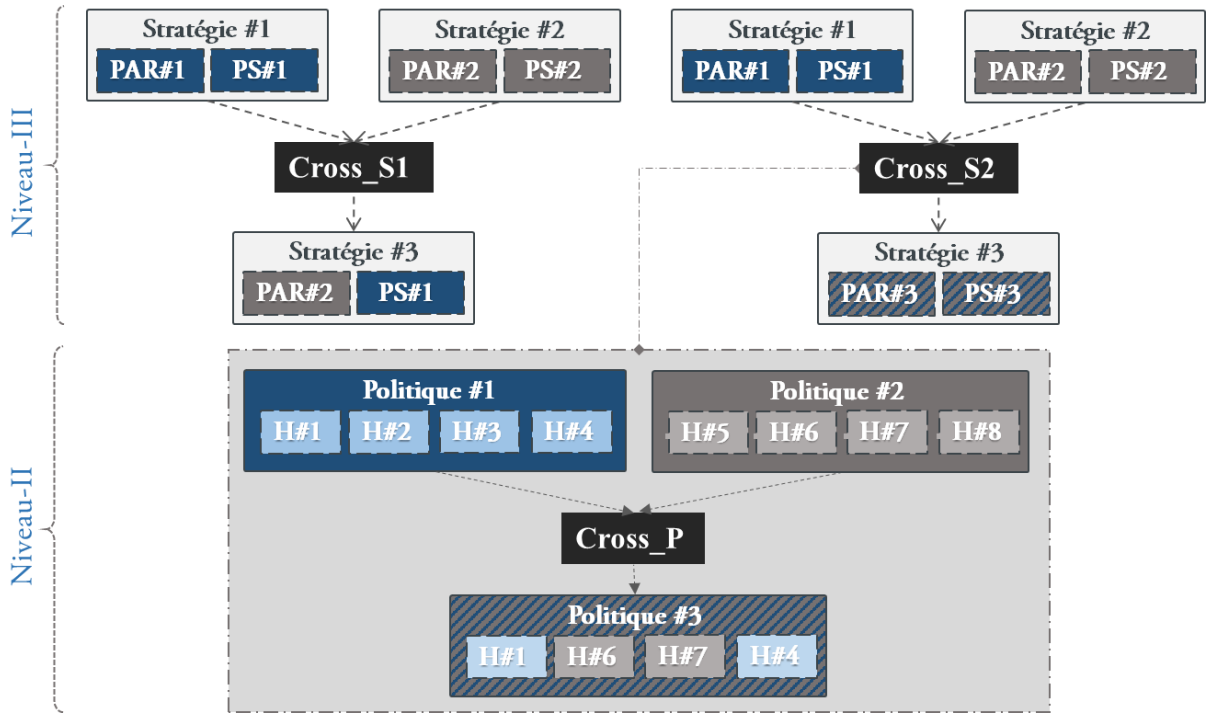


Figure 51 Exemple des procédures de croisement developpées

▪ **Procédures de mutation :** tout comme dans le cas les procédures de croisement, des procédures de mutations adaptées au modèle d'encapsulation ont été développées. Comme illustré dans la Figure 52, elles sont liées :

– ***Mut_S1 :*** permet de modifier une stratégie (Niveau-III) en modifiant intégralement une de ses politiques.

- ***Mut_S2 :*** permet de modifier une stratégie (Niveau-III) en modifiant intégralement une de ses politiques.
- ***Mut_P1 :*** fait muter une politique (Niveau-II) en modifiant une ou plusieurs de ses heuristiques.
- ***Mut_P2 :*** fait muter une politique (Niveau-II) en modifiant une ou plusieurs de ses heuristiques.
- ***Mut_H :*** permet de faire muter une heuristique (Niveau-I) en modifiant une ou plusieurs de ses règles atomiques.

A noter que, dans ce cas également, la contrainte de contiguïté des intervalles limite les mutations de Niveau-I seulement aux règles et non aux bornes.

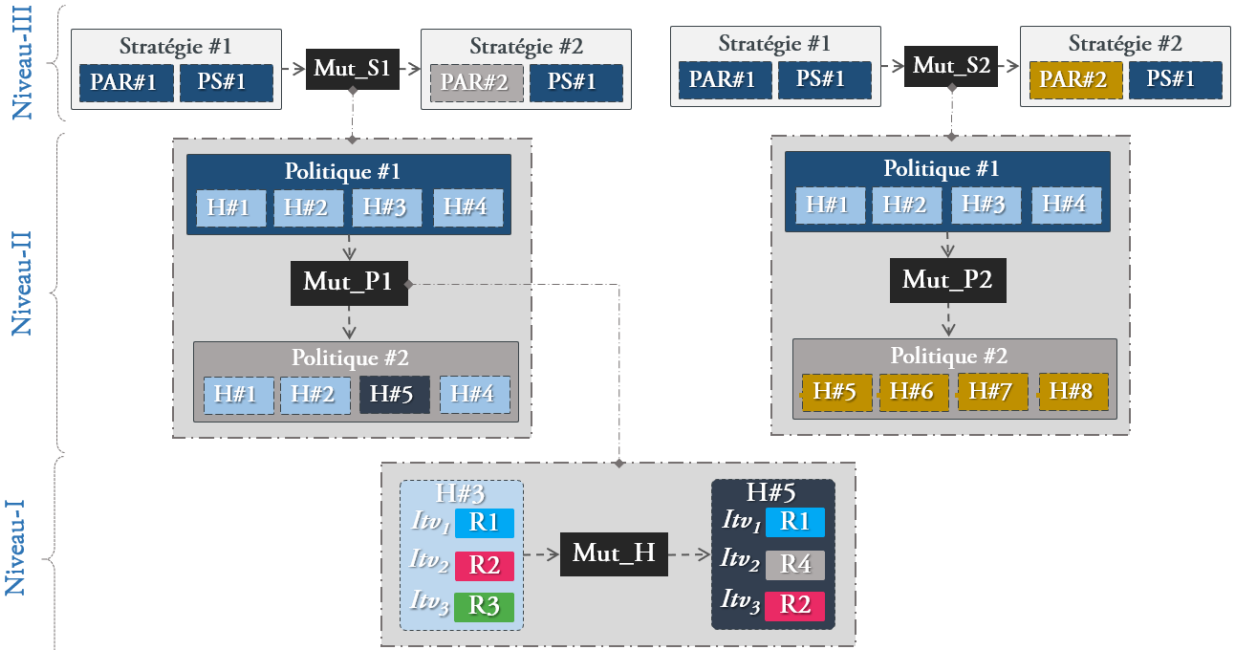


Figure 52 Exemple des procédures de mutation developpées

3.6. FONCTIONNEMENT GLOBAL DE L'HYPER-HEURISTIQUE

La Figure 53 montre les principales étapes du fonctionnement de l'hyper-heuristique et le fonctionnement de l'optimisation via simulation :

1. Les règles atomiques implémentées dans le « répertoire de règles » sont utilisées pour générer une population initiale de stratégies décisionnelles.
2. Pour chaque stratégie décisionnelle, une simulation associée est créée et initialisée avec les données de configuration.
3. Les différentes composantes informationnelles « produits » et « ressources » sont ensuite lancées conformément au scénario de production. A noter que les simulations sont indépendantes. Elles peuvent, selon l'implémentation choisie, être lancées parallèlement pour réduire les temps de calcul.
4. Une fois les lancements terminés, les indicateurs de performance sont collectés au fur et à mesure de l'achèvement des simulations.
5. A la fin de toutes les simulations, les fonctions « fitness » sont recueillies. La population de stratégies décisionnelles est alors triée selon les évaluations obtenues par les simulations.

6. Si la ou les conditions de fin ne sont pas encore réalisées, une nouvelle population de stratégies décisionnelles est obtenue par croisement et mutation. Un nouveau cycle de simulation peut alors commencer avec la nouvelle population.

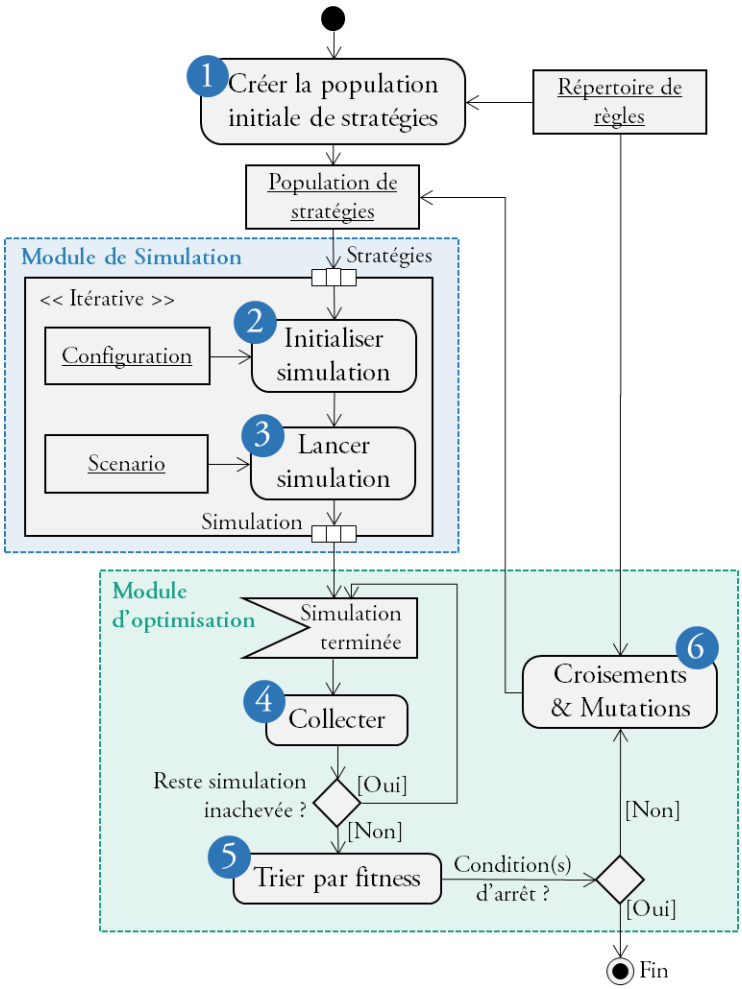


Figure 53 Diagramme d'activité globale de l'hyper-heuristique intégrant les simulations

4. CONCLUSION

Après une présentation des hypothèses de travail et un recensement des besoins, le présent chapitre a permis de :

Modéliser le CPPS en décrivant le comportement des différentes entités produits et ressources.

Proposer une approche basée sur un modèle original d'encapsulation de règles. Ce concept permet d'inclure à la fois des règles d'ordonnancement des contextes décisionnels dans des « stratégies décisionnelles ». En suivant ces stratégies, un PI autonome est à même de mener à bien son processus de production de façon efficace.

Une hyper-heuristique nommée « ITV », ainsi que trois variantes ont été proposées de façon à prendre en compte différents niveaux de finesse des contextes perçus. Un couplage optimisation-simulation est utilisé afin de réduire le phénomène de myopie inhérent à l'approche de pilotage distribué retenue.

L'approche proposée est validée dans le prochain chapitre au travers de plusieurs cas d'étude caractérisés par diverses contraintes inspirées de l'industrie pharmaceutique : différentes configurations d'ateliers (i.e. flowshop flexible hybride et jobshop flexible), multiples familles de produits, variations des temps opératoires, variations des temps de configuration, etc.

CHAPITRE IV : EXPERIMENTATIONS

1. INTRODUCTION

Le chapitre précèdent a détaillé la contribution de la présente thèse à savoir l'hyper-heuristique nommée ITV, dont l'objectif est le pilotage des FMS « par les produits ». Ce quatrième chapitre propose de valider la pertinence de cette approche au travers d'un certain nombre d'expérimentations. Afin de pouvoir comparer les stratégies générées par d'ITV avec différents comportements de PI, les variantes d'HH présentées dans la section III-3.4.2 (i.e. *CELL_ITV, C+R, CELL et RC*) ont été implémentées puis testées.

La première section présente l'outil de simulation, implémenté en Java et basé sur le Framework multiagents JADE. Sur la base de simulations à événements discrets, il permet d'évaluer les différentes stratégies décisionnelles. La seconde section présente les données expérimentales. Largement inspirées de l'industrie pharmaceutique, quatre configurations d'ateliers sont proposées (i.e. deux Flowshops flexibles hybrides et deux Jobshops partiellement flexibles) permettant de tester différents niveaux de flexibilité de routage et « machine ». L'outil ayant permis la génération des 64 scénarios de production est introduit, accompagné des paramètres de l'analyse de variance ANOVA utilisée pour paramétrer l'AG des HH. La troisième section présente les résultats commentés des quelques 30 400 simulations effectuées. La quatrième et dernière section consiste en une discussion plus détaillée selon différentes grilles de lecture.

2. OUTIL DE SIMULATION A BASE DE SMA

Afin de simuler le comportement du FMS piloté par les produits, une solution exploitant un système multiagents a été retenue, celui-ci permettant une implémentation facile des composantes informationnelles des PI et des ressources.

2.1. FRAMEWORK MULTIAGENTS JADE

Le choix du Framework s'est porté sur JADE (Java Agent DEvelopment Framework). Ce logiciel open-source est codé en langage Java. Il simplifie la mise en œuvre de SMA grâce à un middleware conforme aux spécifications FIPA (*Foundation of Intelligent Physical Agents*) destiné à promouvoir l'interopérabilité d'agents hétérogènes et de services (Figure 54). Outre le concept d'abstraction d'agent, JADE fournit un modèle simple mais puissant d'exécution et de composition de tâches, un service de pages jaunes prenant en charge le mécanisme de découverte par abonnement/publication, ainsi que de nombreuses autres fonctionnalités avancées facilitant le développement de systèmes distribués.

La communication entre agents est basée sur le paradigme de transmission asynchrone des messages (Bellifemine et al. 2003). Les agents JADE peuvent communiquer de façon simplifiée grâce au protocole ACL (Agent Communication Language). La structure d'un message ACL, tel que défini par la FIPA, comprend des champs avec des variables indiquant le contexte auquel un message fait référence et un délai d'attente de la réponse. Le protocole prend aussi en charge des interactions complexes comme les conversations parallèles[4]. De plus, il comporte un ensemble

[4] https://jade.tilab.com

d'outils graphiques prenant en charge les phases de débogage et de déploiement. Pour toutes ces raisons, JADE simplifie le développement d'applications distribuées composées d'entités autonomes ayant besoin de communiquer et de collaborer pour assurer le fonctionnement de l'ensemble du système. Ces différentes fonctionnalités correspondent en tous points aux besoins de l'architecture de contrôle hétérarchique proposée dans ce mémoire.

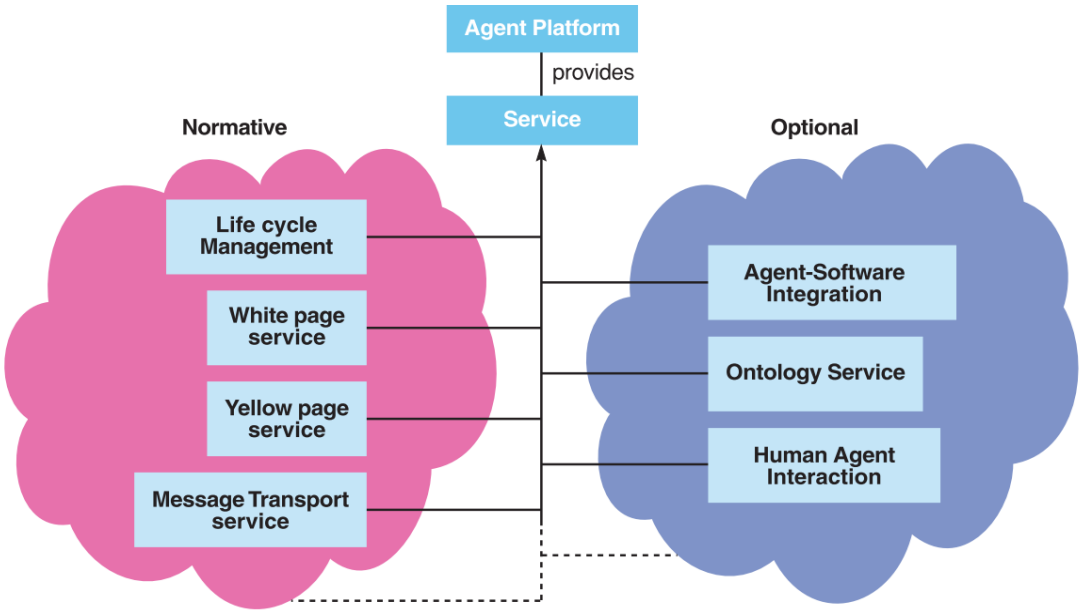


Figure 54 Norme FIPA: services fournis par une plateforme (Bellifemine et al. 2003)

La section suivante présente l'outil de simulation réalisé.

2.2. OUTIL DE SIMULATION DEVELOPPE

Comme illustré par la Figure 55, l'outil de simulation repose sur un certain nombre de containeurs JADE. Les principales étapes du processus de simulation sont les suivantes :

1. Tout d'abord, l'application JAVA instancie le module de simulation supporté par la plateforme JADE. La plateforme est principalement destinée à accueillir les différents containeurs. Pour ce faire, l'application initialise un premier container en fonction des différents paramètres (p. ex. la taille de la population, le nombre d'itération). Puis, elle y crée l'agent principal « *Main_Agent* ».
2. Cet agent principal gère les itérations de l'AG dans sa globalité, en créant une population de stratégies à évaluer. Pour chaque stratégie, il instancie un container JADE et un agent « Simu » qui supervise l'évaluation par simulation. L'agent principal est également en charge de la phase d'optimisation via différentes procédures de croisement et de mutation (cf. III-3.5.2). Grâce à ses fonctionnalités, cet agent correspond au module d'optimisation de notre HH (cf. Figure 53). Les simulations sont lancées sur des containers fonctionnant en parallèle. Cette caractéristique améliore considérablement les temps d'exécution des simulations en réduisant drastiquement les durées d'évaluation d'une population de stratégies décisionnelles.
3. Chaque agent « Simu » supervise une unique simulation afin d'évaluer une stratégie donnée, fournie en entrée par l'agent principal. Selon les données de configuration et le scenario de production fournis, il initialise un ensemble d'agents « Ressource » avec les caractéristiques appropriées. Suite à cette phase d'initialisation des ressources, il entame la lecture du

scénario de production. Les différents événements (p. ex. l'arrivée de nouveaux produits ou pannes) sont déclenchés au fur et à mesure de l'avancement de la simulation. Cette dernière se poursuit jusqu'à l'achèvement de tous les produits. Les indicateurs nécessaires à l'évaluation sont alors calculés et remontés à l'agent principal.

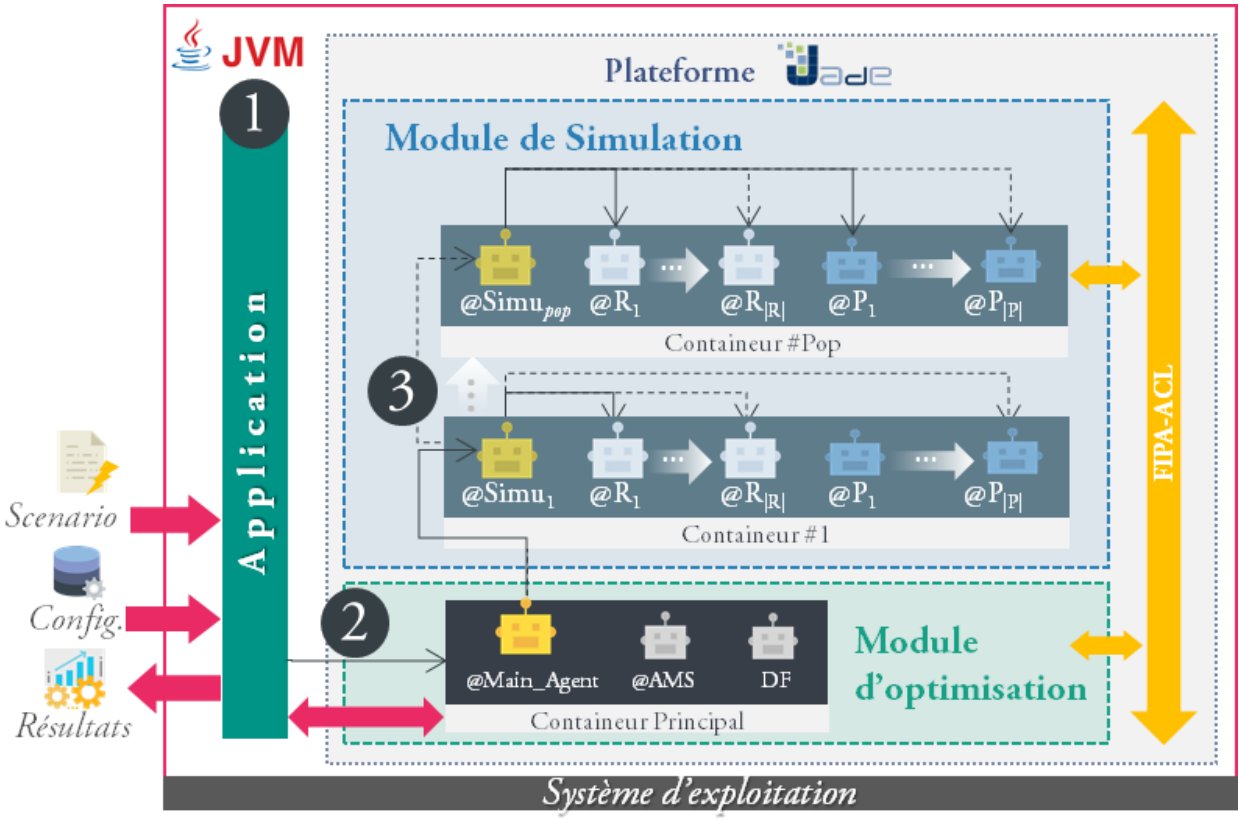


Figure 55 Fonctionnement global de l'outil de simulation

La Figure 56 illustre la séquence des différentes interactions entre agents dans un cas simple (i.e. une seule stratégie, un seul produit et une seule opération). Ce diagramme de séquence décrit les échanges de messages entre les différents agents :

1. Les différents lancements sont effectués en cascade pour initialiser la simulation (i.e. Main_Agent, puis $Simu_1$ puis ressources (R_1, R_2 et R_3) et enfin le produit P_1).
2. Une panne est déclenchée sur la ressource R_2, conformément au scenario.
3. Le produit envoie des demandes d'informations afin d'identifier les ressources susceptibles de réaliser l'opération dont il a besoin (selon l'interopérabilité).
4. A partir des informations reçues, il identifie le contexte décisionnel courant et applique sa politique d'allocation pour sélectionner une règle.
5. Une fois la ressource R_1 sélectionnée, le produit rejoint la file d'attente. En appliquant la politique de séquencement, il met à jour cette file. Le produit attend ensuite l'achèvement de l'opération.
6. Une fois tous les produits achevés, des messages de fin en cascade permettent d'arrêter les simulations et de remonter les informations utiles via les différents agents, jusqu'à l'application.

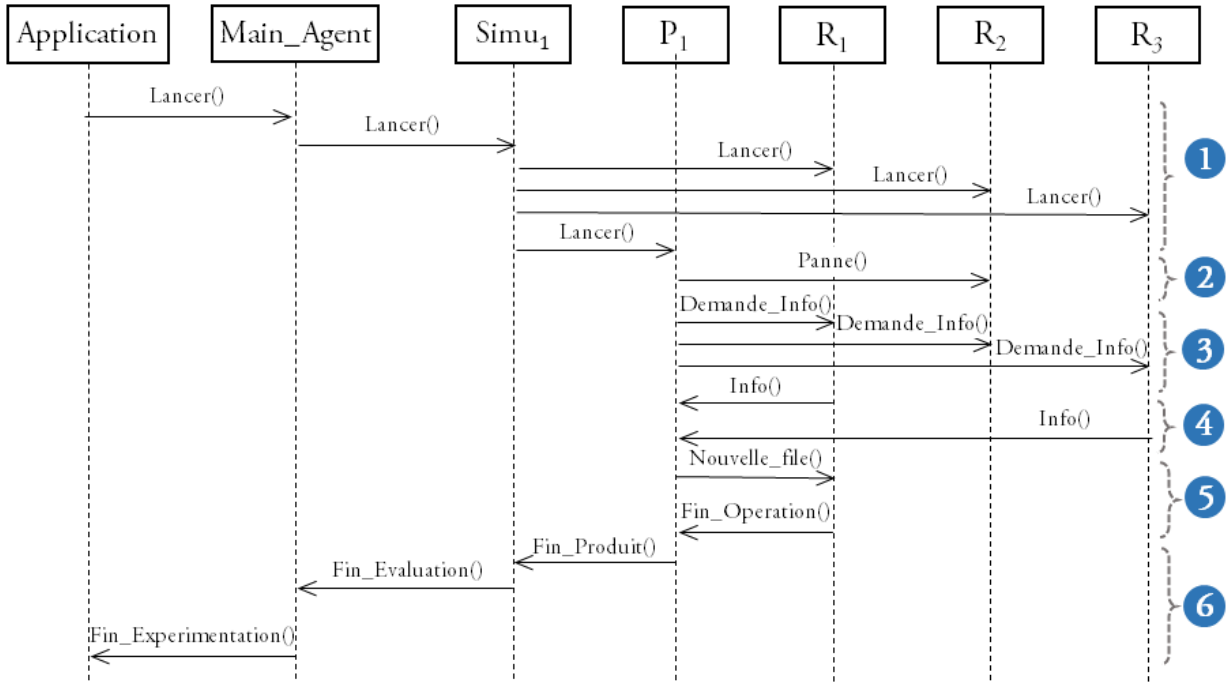


Figure 56 Diagramme de séquence simplifié d'une expérimentation type

3. DONNEES EN ENTREE

Les expérimentations réalisées sont inspirées de l'industrie pharmaceutique, secteur dans lequel ont porté les travaux réalisés dans le cadre du magistère et dans lequel une expérience professionnelle a été acquise. La figure 59 fournit un exemple typique de ligne de production de médicaments composée de quatre cellules (mélange, granulation, enrobage et ensachage). Elle met en lumière certaines caractéristiques fréquemment rencontrées dans ce secteur industriel telles que les multiples familles de produits et l'hétérogénéité des ressources.

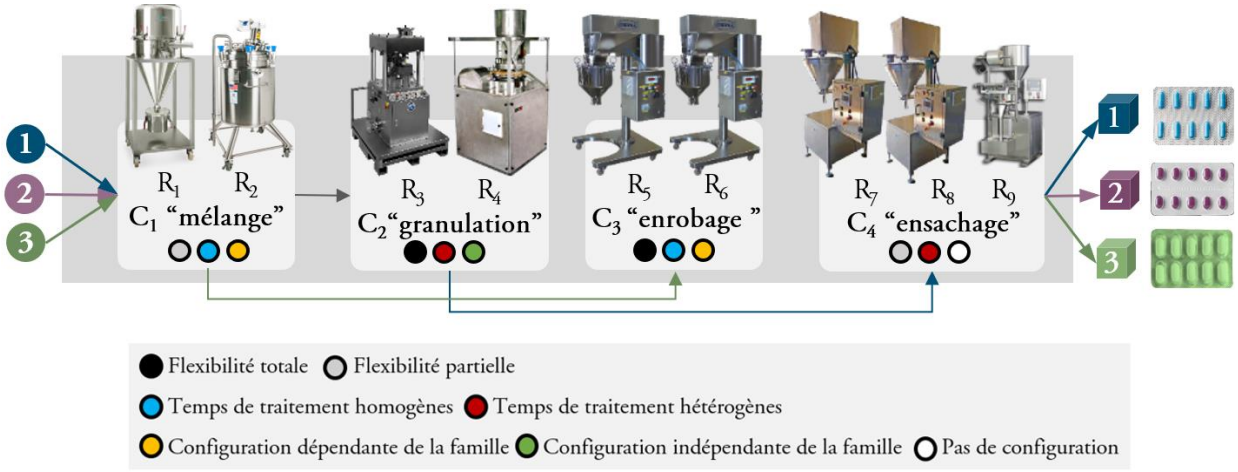


Figure 57 Exemple simplifié d'une ligne de production de produits pharmaceutiques

Ce type de système de production est intéressant en termes de flexibilité et de complexité, et présente les spécificités suivantes :

▪ **Variété des procédés :** facteur essentiel de la flexibilité du SP, elle peut résulter de la variété des produits et/ou de la variété des ressources.

- ***Variété due aux produits :*** si les lignes de production sont le plus souvent dédiées à une classe pharmaceutique particulière (p. ex. antalgique, anti-inflammatoire, ou encore antibiotique), y cohabitent néanmoins de multiples familles de produits. Le plus souvent, une même unité de production peut produire différentes molécules (p. ex. ibuprofène, kétoprofène ou diclofénac) et une même molécule peut être produite sous différentes formes galéniques (p. ex. comprimé, gélule, injectables…etc.). Ces particularités impactent fortement le nombre et l'ordre des opérations constituant les procédés de fabrication.
- ***Variété due aux ressources :*** du fait de l'évolution rapide des procédés de fabrication mais également des outils de production, il n'est pas rare de trouver au sein d'une même cellule des machines hétérogènes utilisant des technologies très différentes. S'il y a incompatibilité entre au moins une ressource et une famille de produits, la cellule devient alors partiellement flexible.

La variété des produits, de par leur nombre, mais aussi de par leurs natures diverses, implique des variations de temps opératoires et de temps de configuration. Ces opérations de configurations sont fréquentes et obligatoires (p. ex. calibration ou changement d'outils, opérations de nettoyage et/ou de stérilisation). A noter que ces temps de configuration peuvent s'avérer particulièrement longs.

▪ **Variation des ordres de fabrication :** la production pharmaceutique est sujette à de nombreuses variations des ordres de fabrication principalement dues à :

- L'incertitude liée aux fluctuations du marché (pénurie, épidémie…etc.).
- La composition parfois complexe des produits, et leur nature périssable, pouvant entrainer une rupture de la chaîne d'approvisionnement.
- Les nombreuses contraintes technico-règlementaires régissant le fonctionnement des industries pharmaceutiques. En effet, la production est conditionnée à l'approbation des services internes d'assurance-qualité et d'organismes de contrôle des autorités sanitaires.

Ces variations se traduisent alors par des arrivées imprévues de produits à ordonnancer.

▪ **Pannes :** à ces spécificités, s'ajoutent la prise en compte des pannes relativement fréquentes, impliquant des arrêts de sécurisation afin d'éviter tout risque sur le produit.

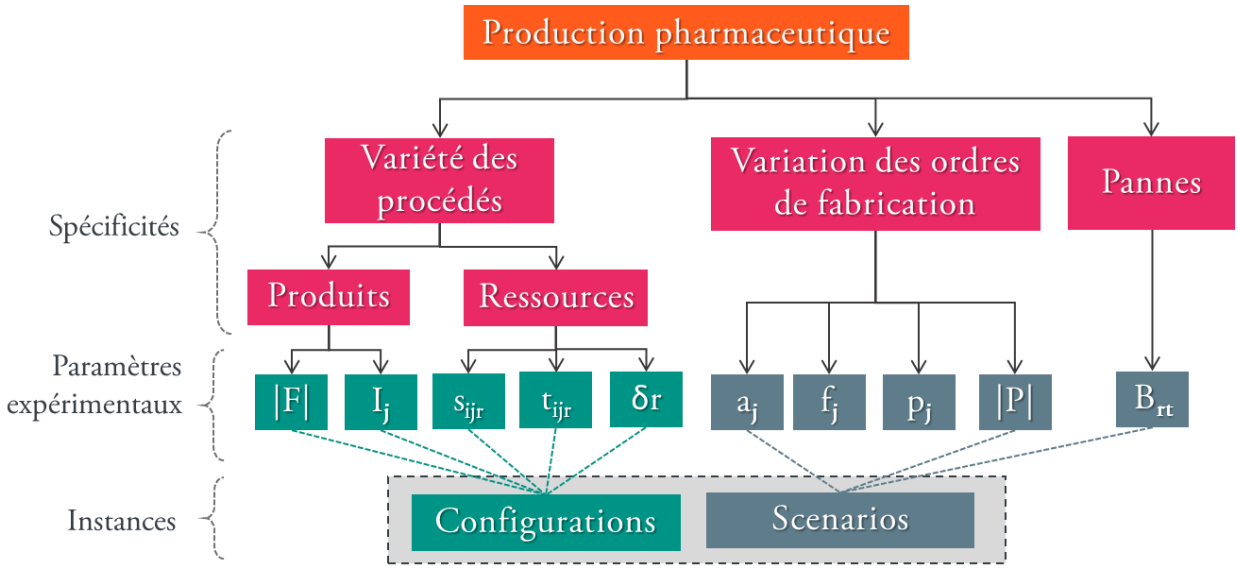


Figure 58 Contrôle des paramétres experimentaux pour la génération d'intances

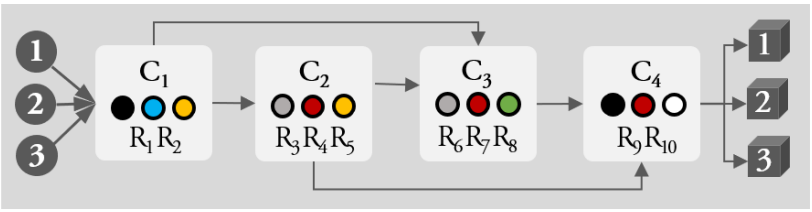
La Figure 58 présente ces contraintes en termes de variation des différents paramètres expérimentaux et d'impacts sur les configurations d'ateliers et les scenarios de production. Ces derniers, qui constituent les différentes instances utilisées lors des expérimentations, sont détaillés dans les deux sections suivantes.

3.1. LES DIFFERENTES CONFIGURATIONS D'ATELIERS

Représentatives de l'industrie pharmaceutique, quatre configurations ont été générées : deux ateliers à cheminement unique flexibles hybrides, ainsi que deux ateliers à cheminement multiples partiellement flexibles. Les différentes ressources et cellules sont paramétrées en faisant varier : le nombre de ressources |R|, le nombre de cellules |L|, le nombre de familles de produits |F|, le nombre et l'ordre des opérations de chaque famille $|I_j|$, les temps opératoires t_{ijr} et de configuration s_{ijr}, ainsi que différents niveaux de flexibilité des ressources δr ,et par conséquence des cellules.

3.1.1. FLOWSHOP FLEXIBLE HYBRIDE HF1

Cette première configuration notée HF1, est composée de 4 cellules de production organisées en ligne. La première et la dernière cellule sont composées de 2 machines tandis que les cellules 2 et 3 sont composées de 3 machines. Comme indiqué Figure 59, les cellules 1 et 4 sont totalement flexibles tandis que les deux autres cellules sont partiellement flexibles. Les temps de traitement sont hétérogènes, sauf pour la première cellule où ils sont identiques, quelle que soit la machine. Les temps de configuration varient selon la famille pour les deux premières cellules, tandis qu'elles varient selon la ressource pour la troisième et sont nuls pour la dernière cellule.



Flexibilité totale ○ Flexibilité partielle
Temps de traitement homogènes ● Temps de traitement hétérogènes
Configuration dépendante de la famille ● Configuration indépendante de la famille ○ Pas de configuration

Figure 59 Configuration flowshop flexible hybride HF1

Le FMS HF1 traite 3 familles de produits. Chaque famille a un certain nombre d'opérations à obtenir dans un ordre bien défini (Tableau 8). Alors que les produits de la famille f_1 transitent par toutes les cellules, les produits des deux autres familles n'en visitent que 3 lors de leur processus de production.

Famille	Séquence d'opération
f_1	$O_1 \prec O_2 \prec O_3 \prec O_4$
f_2	$O_1 \prec O_3 \prec O_4$
f_3	$O_1 \prec O_2 \prec O_4$

Tableau 8 Contraintes de précédence pour la configuration HF1

3.1.2. FLOWSHOP FLEXIBLES HYBRIDE HF2

La seconde configuration notée HF2 est composée de 4 cellules de production organisées elles aussi en ligne (Figure 60). Tout comme pour la configuration HF1, la première et la dernière cellule sont composées de 2 machines tandis que les cellules 2 et 3 sont composées de 3 machines. De même, les cellules 1 et 4 sont totalement flexibles alors que les cellules 2 et 3 ne sont que partiellement flexibles. Les temps de traitement sont homogènes pour les cellules 2 et 3, et hétérogènes pour les cellules 1 et 4. Les temps de configuration sont, quant à eux, indépendants des familles pour la première cellule, nuls pour la deuxième cellule, et dépendants des familles de produits pour les cellules 3 et 4.

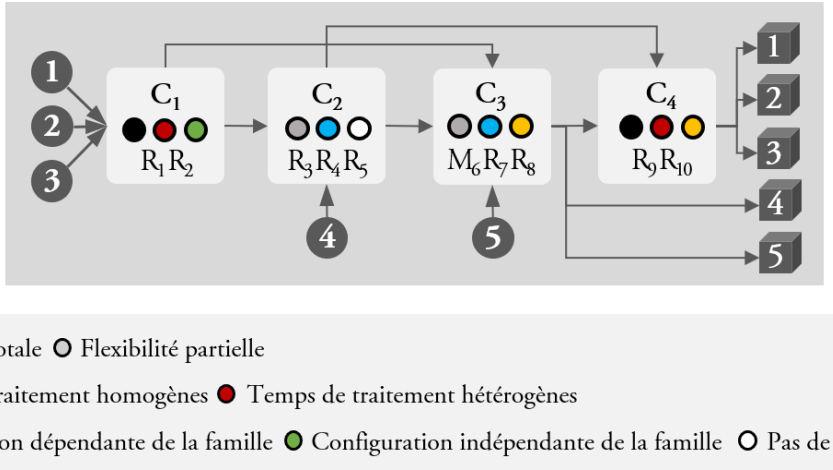


Figure 60 Configuration flowshop flexible hybride HF2

Cinq familles de produits sont associées à la configuration HF2 (Tableau 9). La première famille de produits f_1 doit compléter 4 opérations pour achever la production d'un produit. La dernière famille f_5 n'en requiert que 2, alors que les 3 autres familles nécessitent de transiter par 3 cellules.

Famille	Séquence d'opération
f1	$O_{1j} \prec O_{2j} \prec O_{3j} \prec O_{4j}$
f2	$O_{1j} \prec O_{3j} \prec O_{4j}$
f3	$O_{1j} \prec O_{2j} \prec O_{4j}$
f4	$O_{2j} \prec O_{3j} \prec O_{4j}$
f5	$O_{3j} \prec O_{4j}$

Tableau 9 Contraintes de précédence pour la configuration HF2

3.1.3. JOBSHOP PARTIELLEMENT FLEXIBLE FJ1

La troisième configuration FJ1 correspond à un Jobshop partiellement flexible composé de 5 cellules de production (Figure 61). Les cellules 1, 2 et 5 sont composées de 2 machines, tandis que les cellules 3 et 4 comptent une machine de plus. Seules les cellules 1 et 3 sont caractérisées par une flexibilité totale, les autres cellules étant partiellement flexibles. Les temps de traitement sont homogènes pour les cellules 1, 2 et 5, mais hétérogènes pour les cellules 3 et 4. En ce qui concerne les temps de configuration, ils sont dépendants des familles pour les deux premières cellules, nuls pour la cellule 4, et indépendants des familles de produits pour les cellules 3 et 5.

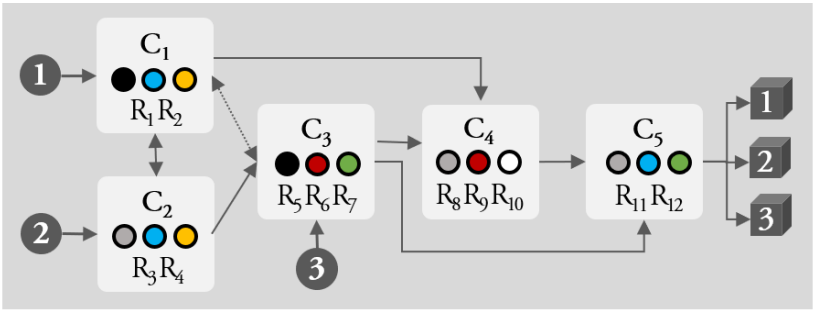


Figure 61 Configuration jobshop partiellement flexibles FJ1

Trois familles de produits sont traitées par la configuration FJ1 (Tableau 10). La première famille de produits f_1, doit cheminer à travers la totalité des 5 cellules. Les Familles f_2 et f_3 achèvent leur production en sollicitant 4 opérations consécutives.

Famille	Séquence d'opération
f_1	$O_{1j} \prec O_{2j} \prec O_{3j} \prec O_{4j} \prec O_{5j}$
f_2	$O_{2j} \prec O_{1j} \prec O_{3j} \prec O_{5j}$
f_3	$O_{3j} \prec O_{1j} \prec O_{4j} \prec O_{5j}$

Tableau 10 Contraintes de précédence pour la configuration FJ1

3.1.4. JOBSHOP PARTIELLEMENT FLEXIBLE FJ2

La quatrième et dernière configuration FJ2 est représentative d'un jobshop partiellement flexible. Elle se compose, tout comme le FMS FJ1, de 5 cellules (Figure 62). Les cellules 2, 3 et 5 comportent chacune 2 machines, tandis que les cellules 1 et 4 sont composées de 3 machines. Les cellules 1 et 3 sont partiellement flexibles alors que les autres sont totalement flexibles. Les temps de traitement sont hétérogènes pour les cellules 1, 2 et 4, et homogènes pour les cellules 3 et 5. Les temps de configuration sont indépendants des familles pour les 3 premières cellules, nuls pour la cellule 4, et dépendants des familles pour la cellule 5.

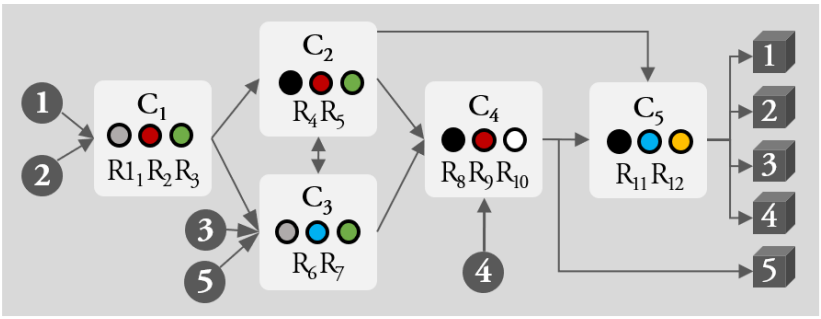


Figure 62 Configuration jobshop partiellement flexibles FJ2

Cette dernière configuration traite 5 familles de produits (Tableau 11). Les deux premières familles doivent cheminer par les 5 cellules pour achever leur production, alors que les familles f_3 et f_4 transitent uniquement par 3 cellules. La dernière famille f_5 requiert, quant à elle, 4 opérations.

Famille	Séquence d'opération
f_1	$O_{1j} \prec O_{2j} \prec O_{3j} \prec O_{4j} \prec O_{5j}$
f_2	$O_{1j} \prec O_{3j} \prec O_{2j} \prec O_{4j} \prec O_{5j}$
f_3	$O_{3j} \prec O_{2j} \prec O_{5j}$
f_4	$O_{4j} \prec O_{2j} \prec O_{5j}$
f_5	$O_{3j} \prec O_{2j} \prec O_{4j} \prec O_{5j}$

Tableau 11 Contraintes de précédence pour la configuration FJ2

Comme présenté dans le Tableau 12, les différentes valeurs des paramètres expérimentaux permettent d'obtenir différents niveaux de flexibilité de « routage » et « machine ». Cette variété au niveau des paramètres est indispensable à une bonne évaluation des performances des algorithmes proposés.

Config.	|R|	|L|	|F|	t_{ijr}	s_{ijr}	Dépendance à la famille				Flexibilité partielle (δr)		Flexibilité	
						L		R		L	R	« Routage »	« Machine »
						t_{ijr}	s_{ijr}	t_{ijr}	s_{ijr}				
HF1	10	4	3	[1..4]	[1..3]	3/4	5/4	3/10	5/10	2/4	5/10	+	+++
HF2			5	[1..5]	[1..3]	4/4	5/4	4/10	5/10	2/4	6/10	++	+
FJ1	12	5	3	[2..5]	[1..5]	0/5	1/5	0/12	1/12	3/5	7/12	+++	++
FJ2			5	[1..7]	[1..6]	7/5	5/5	7/12	5/12	2/5	5/12	++++	+++

Tableau 12 Variations des paramatres expérimentaux pour les configurations

La prochaine section décrit l'outil développé pour permettre de générer les 64 scénarios dynamiques de production.

3.2. GENERATEUR DE SCENARIOS

Après une description assez statique du système de production, réalisée via les configurations d'ateliers, cette section traite de l'aspect dynamique au travers du générateur de scénarios. Cet outil développé en JAVA permet, grâce à des modèles stochastiques, de faire varier des paramètres tels que le nombre de produits, leurs fréquences d'arrivée, leurs priorités, leurs familles, ainsi que le nombre et les durées des perturbations (i.e. pannes machines). La Figure 63 présente le processus de génération des scénarios. Dans un premier temps, des fichiers nommés scripts décrivent |P| produits (id, famille f_j et priorité p_j) ainsi que leurs temps d'arrivée a_{1j}. Chaque script est nommé sous la forme x.y : x représentant la configuration (de 1 à 4), et y le numéro du script (de 1 à 8). Dans un second temps, chaque script permet de générer deux scénarios :

- Un premier scénario, désigné par le suffixe « a », reprend le script tel quel, sans perturbations ;
- Le second scénario désigné par le suffixe « b », reprend le script en y ajoutant un certain

nombre de perturbations B_{rt}. Ces dernières sont générées en fonction de deux paramètres : la probabilité d'occurrence ainsi que la probabilité fixant la durée (dans notre cas, une règle uniforme).

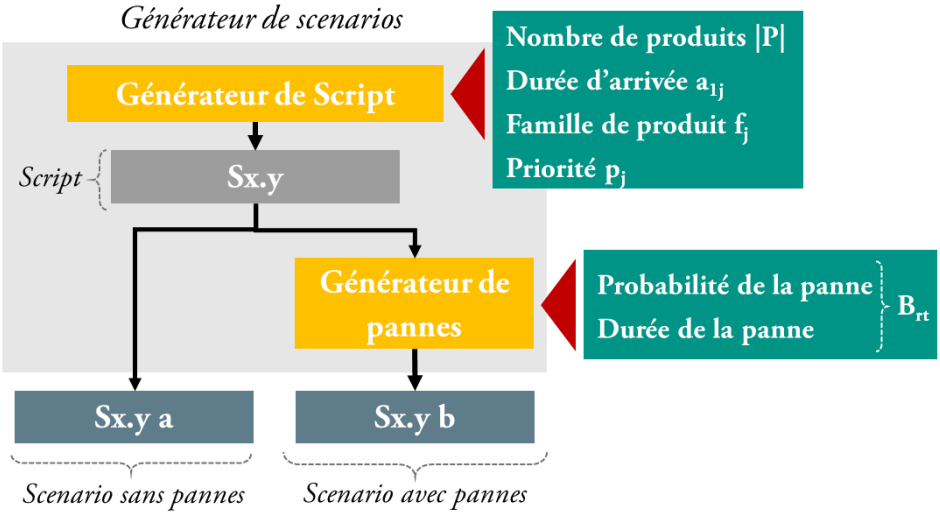


Figure 63 Outil de génération des scénarios

3.2.1. GENERATION DES SCRIPTS

En utilisant deux valeurs pour le nombre de produits (i.e. |P|=60 et |P|=120) et 4 pour la durée d'arrivée (i.e. $a_{11} \leq 60$, $a_{11} \leq 120$, $a_{11} \leq 180$ et $a_{11} \leq 240$), un ensemble de 8 scripts est généré pour chaque configuration. Les familles de produits ainsi que les priorités sont générées en suivant des lois uniformes. Ainsi, différentes cadences d'arrivée sont obtenues pour chaque script (Figure 64), allant de 0.25 produit/minute à 2 produits/minute.

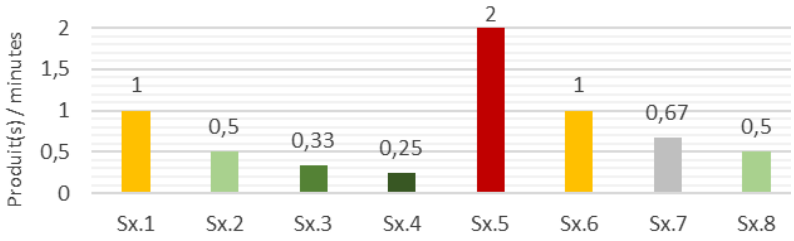


Figure 64 Cadences moyennes d'arrivée de produit pour la génération des scripts

Les figures Figure 65 à Figure 68 décrivent les arrivées des produits a_{1j} (i.e. $j \in [1 .. |P|]$) pour chacun des 8 scripts des quatre configurations (soit un total de 32 scripts différents). Les distributions des familles, ainsi que les priorités des produits sont présentées en Annexe A.

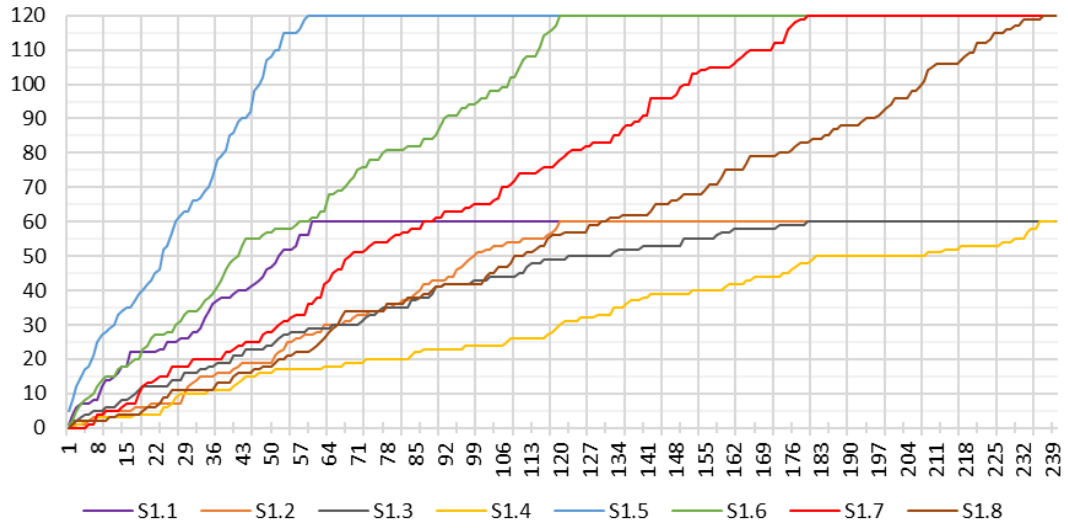


Figure 65 Arrivées des produits pour les scripts de S1.1 à S1.8 pour HF1

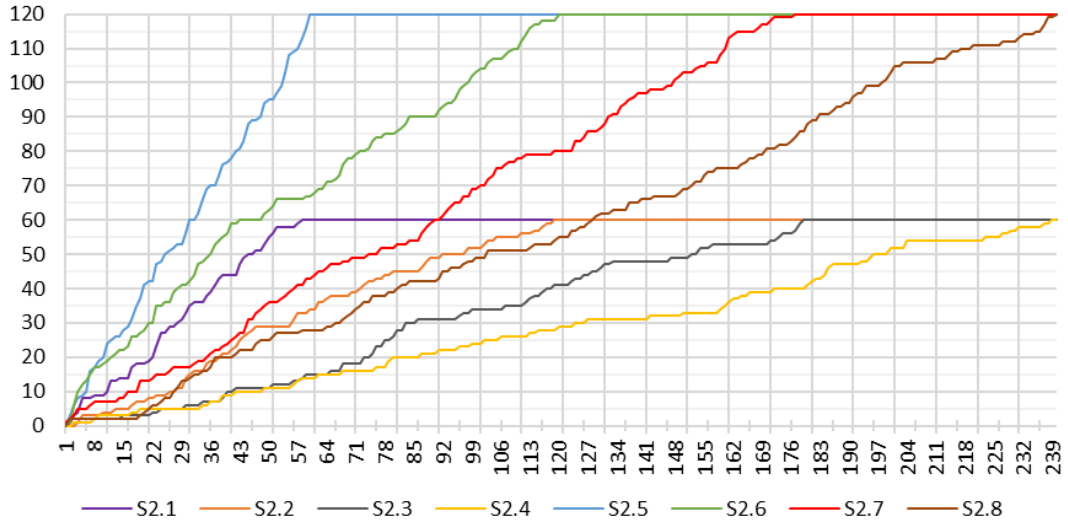


Figure 66 Arrivées des produits pour les scripts S2.1 à S2.8 pour HF2

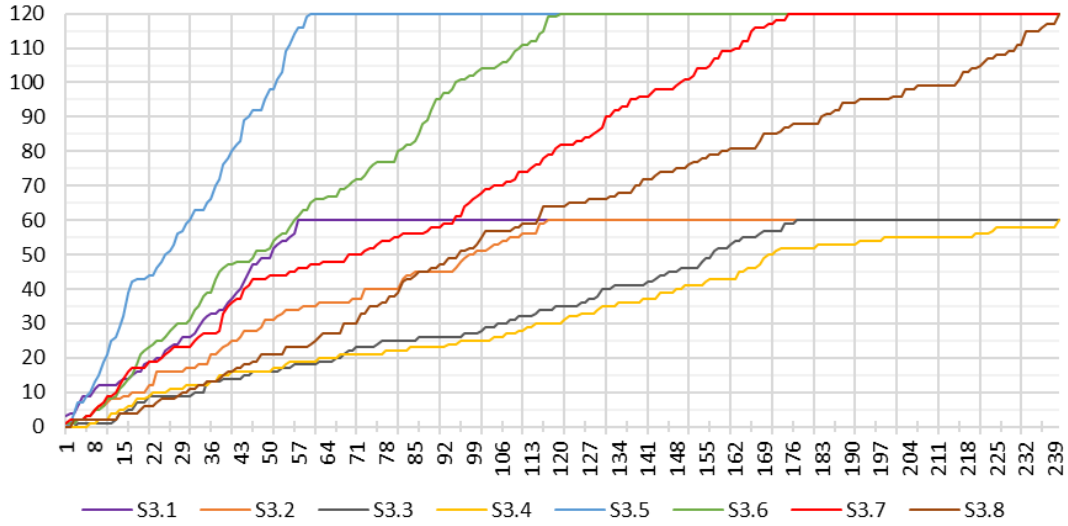


Figure 67 Arrivées des produits pour les scripts de S3.1 à S3.8 pour FJ1

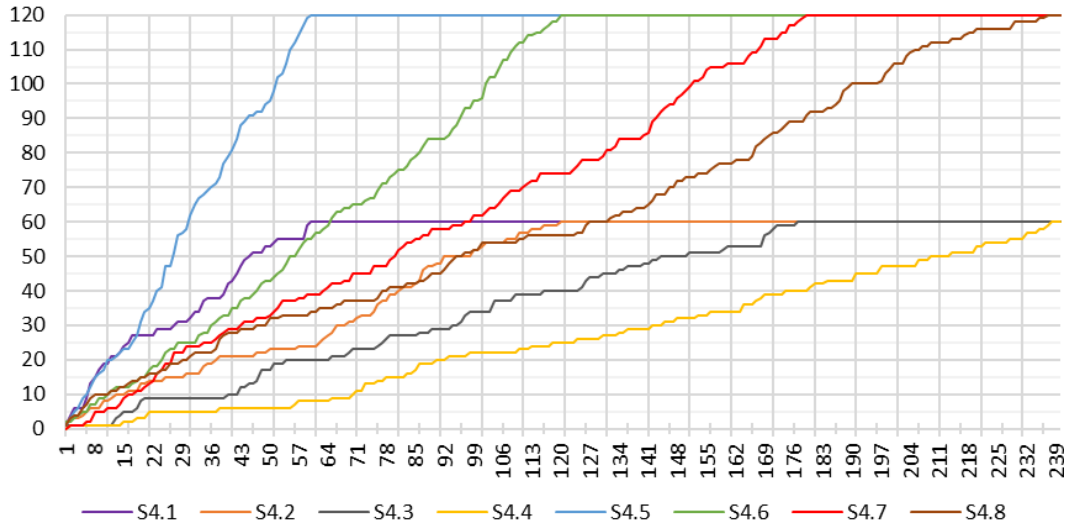


Figure 68 Arrivées des produits pour les scripts S4.1 à S4.8 pour FJ2

Comme explicité précédemment, le nombre d'opérations requis par un produit variant d'une famille à l'autre, le nombre d'opérations O_{ij} varie d'un script à l'autre de 175 à 525 (Figure 69).

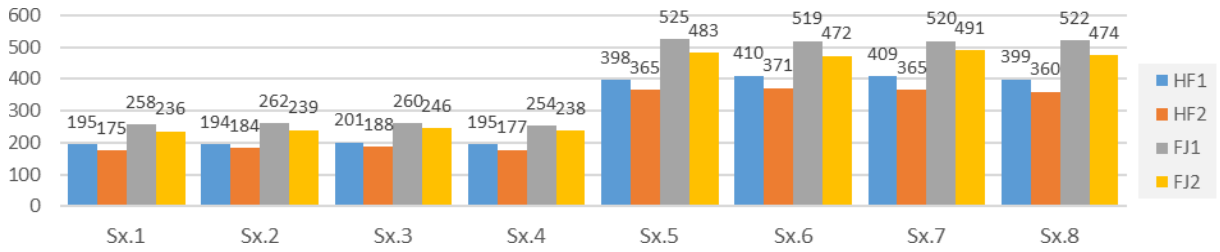


Figure 69 Nombre d'opérations requis pour les différents scripts

3.2.2. GENERATIONS DE PANNES

Les deux paramètres régissant le générateur de pannes sont : la probabilité d'occurrence d'une panne fixée à p=0.01 et la durée de celle-ci suivant une loi uniforme entre 3 et 10 minutes. Le nombre de pannes pour chaque scénario est présenté Figure 70.

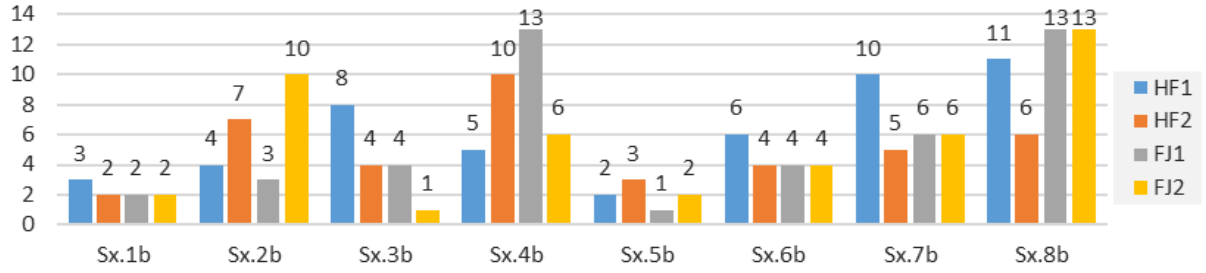


Figure 70 Occurrences des pannes p=0.01

3.3. PARAMETRAGE DES ALGORITHMES

Le but est double :

Evaluer la performance de l'HH ITV selon différentes configurations d'ateliers et scénarios.

Comparer cette performance avec celles des combinaisons de règles d'ordonnancement classiques (i.e. RC) et des variantes de l'HH (i.e. CELL_ITV, C+R et CELL).

Dans ce contexte, des analyses de variances (ANOVA) sont conduites systématiquement

pour chaque configuration afin de paramétrer au mieux chaque HH.

3.3.1. REGLES ATOMIQUES

Pour chaque scénario, les combinaisons de toutes les règles atomiques (14 MSR et 12 SR) sont évaluées. Ces règles relativement simples, issues de la littérature et classiquement employées, sont les règles atomiques utilisées par l'HH proposée « ITV » (Tableau 13), ainsi que par ses différentes variantes. Toutes les combinaisons de règles sont évaluées pour chaque scénario, ce qui correspond à 168 simulations à chaque fois. Une comparaison de la performance des HH avec la meilleure combinaison de règle notée BRC (Best Rules Combination) est également réalisée.

MSR (Machine Selection Rules)			**SR (Sequencing Rules)**		
#	Nom	Signification	#	Nom	Signification
1	LQE	Less Queued Elements	1	FIFO	First In First Out
2	MQE	Most Queued Elements	2	LIFO	Last In First Out
3	STPT	Shortest Total PT[1]	3	FASFO	First At Stage First Out
4	LTPT	Longest Total PT	4	LASFO	Last At Stage First Out
5	STST	Shortest Total ST[2]	5	HPF	Highest Priority First
6	LTST	Longest Total ST	6	LPF	Lowest Priority First
7	STG	Shortest Total Gross	7	SJF	Shortest Job First
8	LTG	Longest Total Gross	8	LJF	Longest Job First
9	QPT	Quickest PT	9	SST	Shortest ST first
10	LPT	Longest PT	10	LST	Longest ST first
11	QST	Quickest ST	11	SGOS	Shortest Gross first
12	LST	Longest ST	12	LGOS	Longest Gross first
13	QG	Quickest Gross			
14	LG	Longest Gross			

[1] Processing Time [2] Setup Time

Tableau 13 Les différentes règles « atomiques » utilisées

3.3.2. ANOVA ET TUNING DES ALGORITHMES

Afin de paramétrer au mieux l'AG de chaque HH, des ANOVA sont conduites pour chaque configuration. L'ANOVA considère deux facteurs : la taille de la population « Pop », et le taux de remplacement de l'AG « Rate ». Ces deux facteurs peuvent prendre 5 valeurs : 100, 250, 500, 750 et 1000 pour « Pop », et 98%, 95%, 90%, 80% et 50% pour « Rate ». Les 25 combinaisons de facteurs possibles sont évaluées pour chacune des 4 HH (i.e. ITV, CELL-ITV, C+R et CELL).

La prochaine section est dédiée à la présentation et à une brève discussion des résultats expérimentaux.

4. RESULTATS EXPERIMENTAUX

Cette section présente les résultats des 30 432 simulations (Tableau 14) exécutées sur une station de travail équipée d'un CPU Intel® Xeon® E3-1230 v3 @ 3.30 GHz 3.30 GHz et de 8Go de RAM. Pour chaque configuration d'atelier, les 7 609 simulations ont nécessité entre 60 à 120 minutes de temps de calcul selon la complexité des configurations et des scénarios. Ces temps d'exécution assez faibles ont été rendus possibles par la parallélisation des containers multiagents (voir section IV-2.2).

	HF1	**HF2**	**FJ1**	**FJ2**	**Total**
ANOVA	30 x 25 x 4	30 x 25 x 4	30 x 25 x 4	30 x 25 x 4	12 000
BRC	168 x 16	168 x 16	168 x 16	168 x 16	10 752
HH	30 x 4 x 16	30 x 4 x 16	30 x 4 x 16	30 x 4 x 16	7 680
Total	7 609	7 609	7 609	7 609	**30 432**

Tableau 14 Détail du nombre de simulations

4.1. FLOWSHOP FLEXIBLE HYBRIDE HF1

4.1.1. ANOVA HF1

Pour l'HH ITV, l'étude ANOVA montre que la taille de la population Pop a un impact statistiquement significatif avec une p-value < 0.05 (Tableau 15). Le taux de remplacement Rate n'a lui que peu d'incidence sur sa performance. En revanche, ces deux facteurs sont statistiquement significatifs pour la seconde HH CELL_ITV (Tableau 16). Enfin, ces deux facteurs n'ont pas d'impact sur les HH CELL et C+R.

Variations	**Sommes des carrés**	**Degré de liberté**	**Moyennes**	**F**	**Probabilité**
Pop	4,91487556	4	1,22871889	22,0007759	**4,7777E-13**
Rate	0,16107556	4	0,04026889	0,72103295	0,57954637
Pop*Rate	0,95521333	16	0,05970083	1,06897083	0,39445038

Tableau 15 ANOVA de l'algorithme "ITV" pour HF1

Variations	**Sommes des carrés**	**Degré de liberté**	**Moyennes**	**F**	**Probabilité**
Pop	0,84106667	4	0,21026667	8,88909766	**3,4969E-06**
Rate	0,37128889	4	0,09282222	3,92409225	**0,0053111**
Pop*Rate	0,53664444	16	0,03354028	1,4179271	0,14865686

Tableau 16 ANOVA de l'algorithme "CELL_ITV" pour HF1

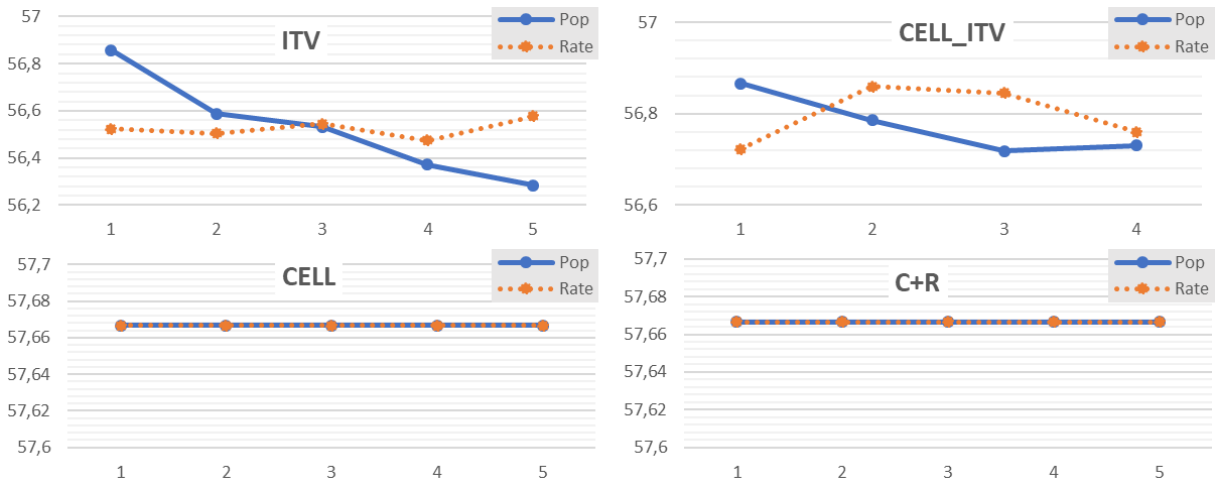


Figure 71 Tracé des principaux effets des paramètres des AG pour HF1

En considérant les résultats présentés en Figure 71, les valeurs retenues pour le tuning des AG sont décrites dans le Tableau 17 ci-dessous. Il convient de noter que pour les HH CELL et C+R, la sélection des paramètres a été faite dans le but de maintenir un temps d'exécution raisonnable.

Paramètres	ITV	CELL_ITV	CELL	C+R
Pop	1000	750	750	750
Rate	80%	90%	80%	80%

Tableau 17 Paramètres des HH après ANOVA pour HF1

4.1.2. RESULTATS HF1

Le Tableau 18 présente une synthèse des résultats issus des 4 608 simulations pour les scénarios de S1.1a à S1.8b. Les scénarios 1.1a, 1.1b sont pris comme exemples illustratifs, et sont détaillés ci-après.

Pour le premier scénario S1.1a, les meilleures combinaisons sont constituées de la règle LQE, qu'elle soit associée à la règle LJF ou LST (i.e. LQE+LJF et LQE+LST). La production des 60 produits est effectuée avec un TAM de 67,93min.

L'HH la plus performante est ITV qui parvient à abaisser le TAM à 59,48min., puis vient CELL_ITV avec un TAM= 59,80min., et enfin C+R et CELL avec toutes deux des TAM de 66,67min.

Le second scénario S1.1b contient 3 pannes dont les durées respectives sont 3, 2 et 1 minute. La meilleure combinaison de règles est LQE+LJF avec un TAM= 74,20min.

L'HH ITV abaisse ce dernier à 66,32min. CELL_ITV l'abaisse à 66,38 min. et enfin CELL et C+R ont un TAM=73,10min.

La Figure 72 montre les évolutions du nombre de produits dans le FMS pour les scénarios S1.1a & S1.1.b HF1. D'autres graphiques complémentaires montrant l'évolution du nombre de produits dans le FMS sont disponibles en annexe B.1.

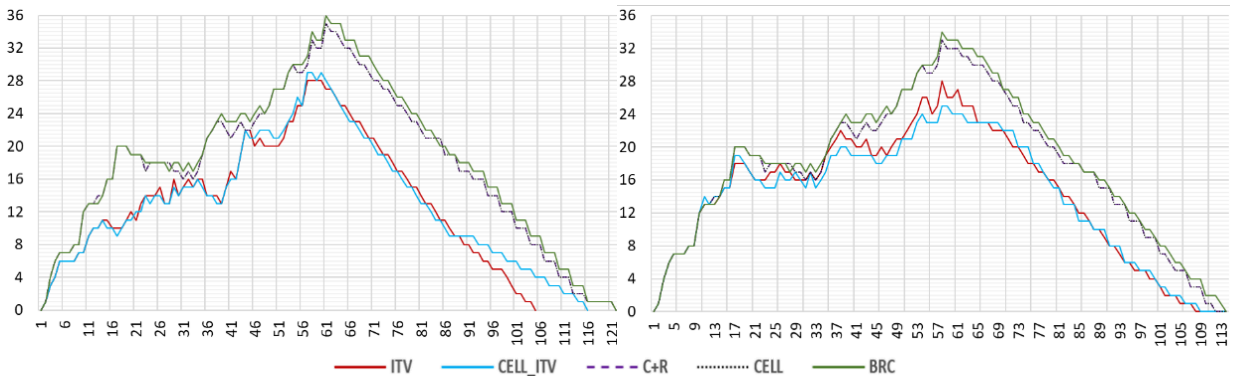


Figure 72 Evolutions du nombre de produits dans le FMS pour les scenarios S1.1a & S1.1.b

On remarque une corrélation entre les performances des différents algorithmes avec les schémas d'allocation de règles. En d'autres termes, plus la stratégie permet de décomposer finement le comportement des PI, plus ceux-ci peuvent combiner des règles efficacement.

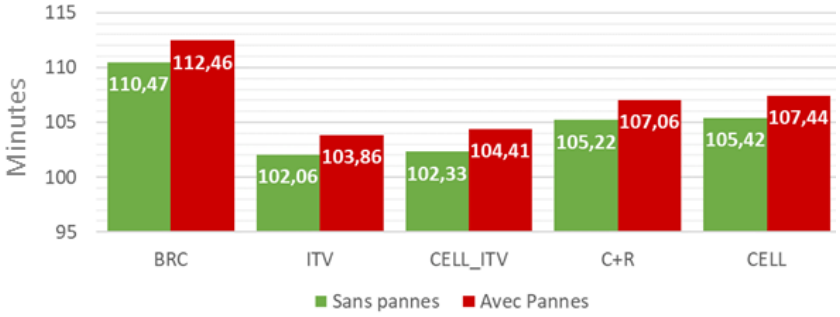


Figure 73 Moyennes des TAM (avec et sans pannes) pour la configuration HF1

Un autre aspect apparait dans la Figure 74 qui présente les moyennes des performances relatives au BRC pour les 4 HH pour chacun des 16 scénarios de cette configuration. Il apparait clairement qu'il existe une autre corrélation forte entre les performances des HH et les cadences d'arrivées : plus les cadences sont élevées, plus les HH surpassent les combinaisons BRC. Ceci est particulièrement vrai pour les deux HH « ITV » et « CELL_ITV ».

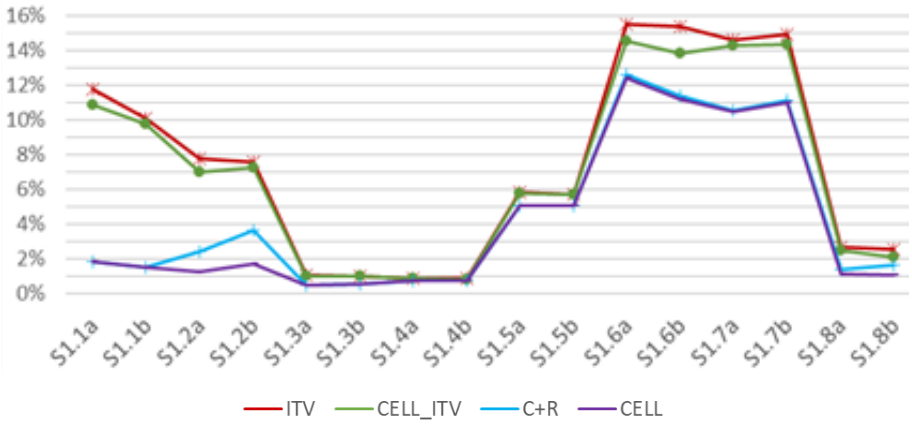


Figure 74 Performances relatives des HH par rapport aux BRC pour HF1

#	Scenario	Arrivées	Pannes	BRC		ITV			CELL_ITV			C+R			CELL		
				MSR+SR	TAM	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max
1	S1.1a	60/60	-	LQE+LJF	67,93	59,48	59,94	60,78	59,80	60,55	61,28	66,67	66,67	66,67	66,67	66,67	66,67
2	S1.1b	60/60	3	LQE+LJF	74,20	66,32	66,71	67,90	66,38	66,97	67,17	73,10	73,10	73,10	73,10	73,10	73,10
3	S1.2a	60/120	-	STPT+LJF	86,33	79,33	79,66	81,10	79,58	80,30	81,10	84,23	84,23	84,23	85,23	85,23	85,23
4	S1.2b	60/120	4	STG+LJF	92,62	85,47	85,61	85,83	85,65	85,92	86,53	89,25	89,25	89,25	91,05	91,05	91,05
5	S1.3a	60/180	-	LQE+FIFO	83,33	82,47	82,47	82,52	82,52	82,52	82,52	82,92	82,92	82,92	82,92	82,92	82,92
6	S1.3b	60/180	8	LQE+FIFO	83,33	82,47	82,48	82,53	82,47	82,51	82,53	82,90	82,90	82,90	82,90	82,90	82,90
7	S1.4a	60/240	-	LQE+FIFO	129,42	128,28	128,29	128,30	128,30	128,30	128,30	128,50	128,50	128,50	128,50	128,50	128,50
8	S1.4b	60/240	5	LQE+FIFO	129,43	128,28	128,30	128,37	128,35	128,36	128,37	128,50	128,50	128,50	128,50	128,50	128,50
9	S1.5a	120/60	-	STG+LJF	108,27	101,96	101,97	102,03	101,99	101,99	101,99	102,83	102,83	102,83	102,83	102,83	102,83
10	S1.5b	120/60	2	STG+LJF	108,27	101,96	102,08	102,33	101,98	102,09	102,35	102,83	102,83	102,83	102,83	102,83	102,83
11	S1.6a	120/120	-	LQE+LJF	121,73	102,50	102,85	103,59	103,68	104,03	104,56	106,38	106,38	106,38	106,61	106,61	106,61
12	S1.6b	120/120	6	LQE+LJF	123,49	103,32	104,55	106,55	105,95	106,45	106,94	109,45	109,45	109,45	109,71	109,71	109,71
13	S1.7a	120/180	-	STPT+LJF	137,85	117,59	117,76	118,00	117,91	118,15	118,48	123,35	123,35	123,35	123,38	123,38	123,38
14	S1.7b	120/180	10	STG+LJF	139,18	118,29	118,43	119,01	118,63	119,22	120,41	123,74	123,74	123,74	123,86	123,86	123,86
15	S1.8a	120/240	-	STG+LJF	148,93	144,85	144,99	145,41	144,88	145,19	145,60	146,88	146,88	146,88	147,25	147,25	147,25
16	S1.8b	120/240	11	STG+LJF	149,16	144,73	145,39	146,32	145,89	146,08	146,23	146,67	146,67	146,67	147,54	147,54	147,54

Tableau 18 Synthèse des résultats pour HF1

4.2. FLOWSHOP FLEXIBLE HYBRIDE HF2

4.2.1. ANOVA HF2

L'ANOVA montre que le facteur Pop a toujours une incidence statistiquement significative sur les HH à l'exception de C+R. Le second facteur Rate n'a cependant une incidence que pour l'HH CELL.

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	44,6935688	4	11,17339222	14,93041566	**1,33714E-09**
Rate	1,99450222	4	0,498625556	0,666287073	0,616871206
Pop*Rate	9,95125333	16	0,621953333	0,831083489	0,647968171

Tableau 19 ANOVA de l'algorithme "ITV" pour HF2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	34,54624444	4	8,636561111	13,08100799	**1,32147E-08**
Rate	3,146777778	4	0,786694444	1,191534012	0,319241396
Pop*Rate	8,441311111	16	0,527581944	0,799080044	0,683436914

Tableau 20 ANOVA de l'algorithme "CELL_ITV" pour HF2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	0,127831111	4	0,031957778	2,275834784	0,066335655
Rate	0,050564444	4	0,012641111	0,900221554	0,466915662
Pop*Rate	0,195702222	16	0,012231389	0,871043678	0,603344278

Tableau 21 ANOVA de l'algorithme "C+R" pour HF2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	1,665368889	4	0,416342222	4,624197848	**0,001817139**
Rate	1,346746667	4	0,336686667	3,739485635	**0,007054232**
Pop*Rate	1,831297778	16	0,114456111	1,271232353	0,23063242

Tableau 22 ANOVA de l'algorithme "CELL" pour HF2

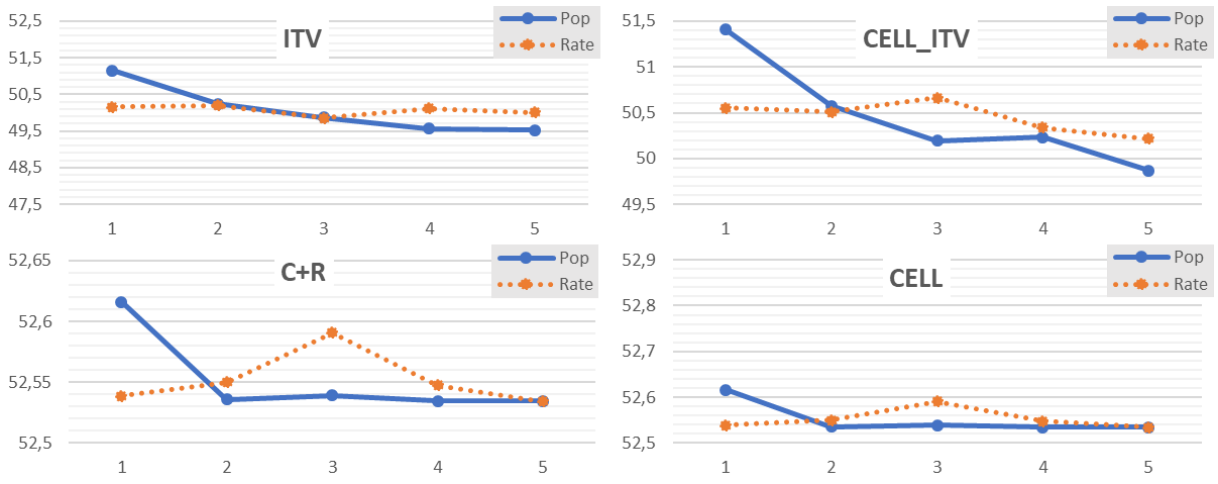


Figure 75 Tracé des principaux effets des paramètres des AG pour HF2

En considérant les effets des paramètres des AG présentés dans la Figure 75, les paramètres retenus pour paramétrer les HH sont donnés dans le Tableau 23.

Paramètres	ITV	CELL_ITV	CELL	C+R
Pop	1000	1000	1000	1000
Rate	90%	50%	50%	50%

Tableau 23 Paramètres des HH après ANOVA pour HF2

4.2.2.RESULTATS HF2

Comme pour HF1, ce sont 4 608 simulations qui sont synthétisées dans le Tableau 24. Les scénarios 2.1a, 2.1b sont pris comme exemples illustratifs et détaillés ci-après. D'autres figures supplémentaires sont disponibles en annexe B.2.

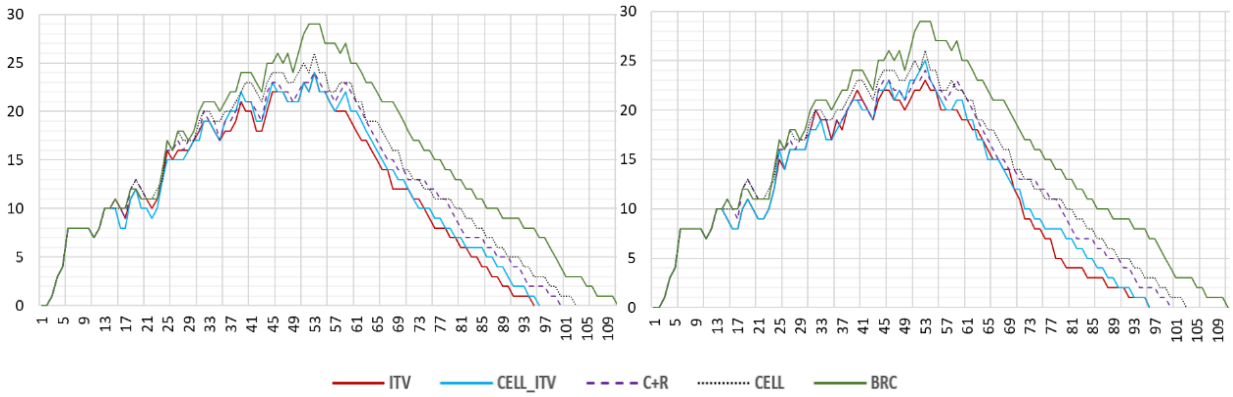


Figure 76 Evolutions du nombre de produits dans le FMS scenarios S2.1a & S2.1b

Pour le scénario S2.1a, la BRC est la combinaison LPT+LJF avec un TAM= 55,35min. ITV produit lui un TAM de 48,10min., CELL_ITV un TAM de 48,22min., C+R un TAM de 50,20min, et enfin l'HH CELL un TAM de 51,43min.

Le scénario S2.1b, inclut 2 pannes. Tout comme pour le scénario S2.1a, la BRC est LPT+LJF avec un TAM= 55,35min. ITV produit, quant à lui, un TAM= 47,70min. CELL_ITV un TAM= 47,87min. C+R et CELL restent inchangées avec des TAM respectifs de 50,20min. et 51,43min. Le faible impact des pannes trouve ici deux explications possibles : soit les ressources en panne ne sont pas utilisées et la panne n'a donc aucune incidence. Soit la panne touche une ressource dont une réplique identique est disponible.

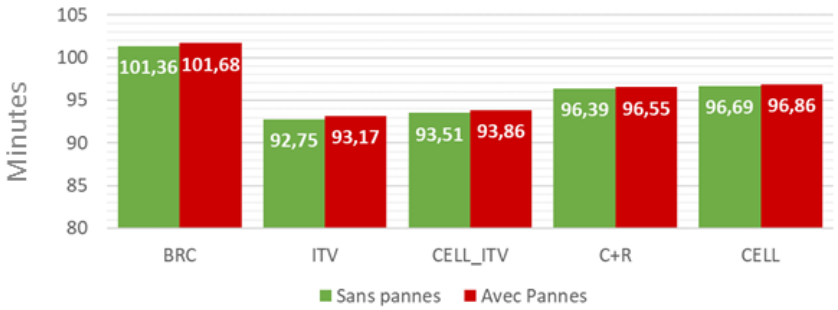


Figure 77 Moyenne des TAM (avec et sans panne) pour la configuration HF2

La Figure 77 montre les moyennes des TAM, avec et sans panne, pour la configuration HF2. Les algorithmes par ordre de performance sont ITV, CELL_ITV, C+R, CELL et enfin les BRC, avec respectivement des TAM de 92,96min., 93,69min., 96,47min., 96,78min. et 101,52min.

Dans ce cas également, comme pour HF1, il existe une corrélation entre les différents

schémas d'allocation de règles et les performances obtenues. Cependant, il existe tout de même une différence notable, comme le montre la Figure 78. En effet, pour cette configuration qui présente une flexibilité de routage plus importante et une flexibilité machine faible, les gains apportés par les HH avec intervalles atteignent respectivement 25% et 23%. Ceci traduit une efficacité accrue comparativement à HF1.

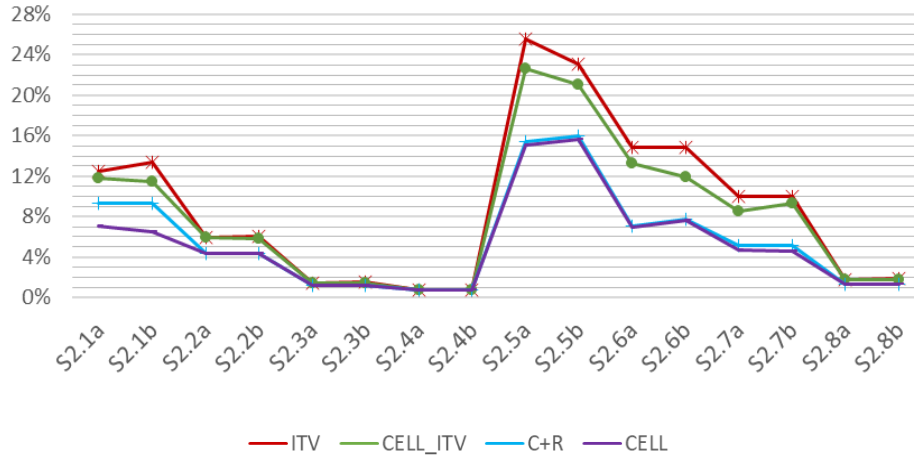


Figure 78 Performances relatives des HH par rapport aux BRC pour HF2

#	Scenario	Arrivées	Pannes	BRC		ITV			CELL_ITV			C+R			CELL		
				MSR+SR	TAM	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max
1	S2.1a	60/60	-	LPT+LJF	55,35	48,10	48,45	49,52	48,22	48,82	49,65	50,20	50,20	50,20	51,43	51,43	51,43
2	S2.1b	60/60	2	LPT+LJF	55,35	47,70	47,94	48,77	47,87	48,99	50,03	50,20	50,20	50,20	51,43	51,76	51,80
3	S2.2a	60/120	-	STG+LJF	71,55	67,22	67,26	67,42	67,27	67,27	67,28	68,45	68,45	68,45	68,45	68,45	68,45
4	S2.2b	60/120	7	STG+LJF	71,58	67,25	67,25	67,27	67,32	67,38	67,58	68,50	68,50	68,50	68,50	68,50	68,50
5	S2.3a	60/180	-	LPT+LIFO	106,23	104,67	104,67	104,68	104,68	104,70	104,77	104,92	104,92	104,92	104,92	104,92	104,92
6	S2.3b	60/180	4	LPT+LIFO	106,23	104,63	104,65	104,67	104,67	104,69	104,72	104,92	104,92	104,92	104,98	104,98	104,98
7	S2.4a	60/240	-	LPT+FASFO	135,61	134,57	134,57	134,57	134,58	134,58	134,58	134,60	134,60	134,60	134,62	134,62	134,62
8	S2.4b	60/240	10	LPT+FASFO	135,61	134,57	134,57	134,57	134,58	134,58	134,58	134,60	134,60	134,60	134,62	134,62	134,62
9	S2.5a	120/60	-	LPT+LST	93,95	69,02	69,96	71,08	71,68	72,65	73,14	79,30	79,51	79,78	79,84	79,84	79,84
10	S2.5b	120/60	3	LPT+LASFO	94,26	70,88	72,47	73,71	72,70	74,44	76,27	79,07	79,26	79,43	79,52	79,52	79,52
11	S2.6a	120/120	-	LPT+LGOS	94,71	80,33	80,67	81,08	81,76	82,14	83,18	88,03	88,04	88,04	88,13	88,13	88,13
12	S2.6b	120/120	4	LPT+LGOS	96,92	81,74	82,55	83,34	84,42	85,38	86,83	89,45	89,46	89,46	89,55	89,55	89,55
13	S2.7a	120/180	-	LPT+LST	117,1	104,28	105,41	107,28	105,95	107,10	108,08	111,08	111,08	111,13	111,57	111,57	111,57
14	S2.7b	120/180	5	LPT+SST	117,1	104,73	105,42	106,68	105,38	106,24	108,32	111,10	111,10	111,10	111,76	111,76	111,76
15	S2.8a	120/240	-	LPT+LIFO	136,39	133,83	133,91	134,02	133,94	134,00	134,03	134,55	134,55	134,55	134,55	134,55	134,55
16	S2.8b	120/240	6	LPT+LIFO	136,40	133,83	133,89	134,02	133,94	134,00	134,02	134,55	134,55	134,55	134,55	134,55	134,55

Tableau 24 Synthèse des résultats pour HF2

4.3. JOBSHOP FLEXIBLE FJ1

4.3.1. ANOVA FJ1

L'étude des variances démontre que les deux facteurs sélectionnés ont un impact toujours statistiquement significatif avec une exception, à savoir le taux de remplacement sur l'algorithme CELL.

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	84,0478089	4	21,0119522	12,5267662	**2,6772E-08**
Rate	17,4644089	4	4,36610222	2,60295384	**0,04038667**
Pop*Rate	31,7471467	16	1,98419667	1,1829252	0,29474479

Tableau 25 ANOVA de "ITV" pour FJ1

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	81,1492667	4	20,2873167	11,8815379	**6,1623E-08**
Rate	17,1712889	4	4,29282222	2,51414865	**0,04623388**
Pop*Rate	58,6704444	16	3,66690278	2,14757057	0,01152657

Tableau 26 ANOVA for "CELL_ITV" FJ1

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	70,6595022	4	17,6648756	7,67104572	**1,9679E-05**
Rate	8,21290222	4	2,05322556	0,89162174	0,47195377
Pop*Rate	22,3621867	16	1,39763667	0,60692954	0,87167211

Tableau 27 ANOVA for "CELL " FJ1

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	27,527015	4	6,8817538	5,30647809	**0,0006448**
Rate	15,509369	4	3,8773424	2,98979495	**0,0223441**
Pop*Rate	30,918724	16	1,9324202	1,49007741	0,1182952

Tableau 28 ANOVA for "C+R " FJ1

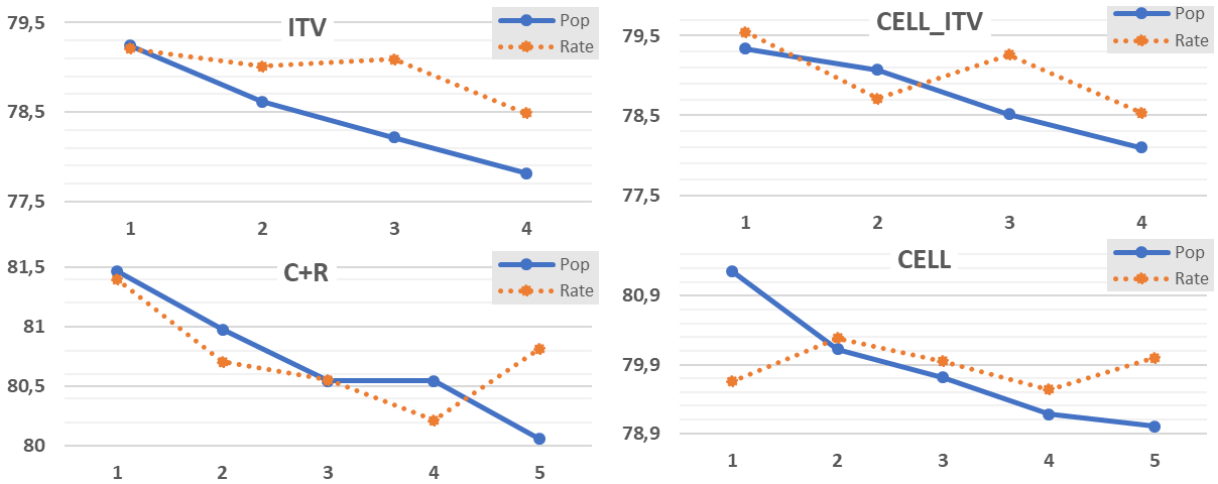


Figure 79 Tracés des principaux effets des paramètres des AG pour FJ1

Si, comme illustré dans la Figure 79, la performance est corrélée avec la taille de la population, la variable Rate prend différentes valeurs données dans le Tableau 29.

Paramètres	ITV	CELL_ITV	CELL	C+R
Pop	1000	1000	1000	1000
Rate	50%	50%	80%	80%

Tableau 29 Paramètres des HH après ANOVA pour FJ1

4.3.2.RESULTATS FJ1

La synthèse des résultats des 4 608 simulations est donnée dans le Tableau 24. Les scénarios 3.1a, 3.1b sont pris comme exemple illustratifs et détaillés ci-après. Des graphiques additionnels montrant l'évolution du nombre de produits dans le FMS sont disponibles en annexe B.3.

Pour le scénario S3.1a, la meilleure combinaison BRC est LQE+SGOS avec un TAM= 80,98min. ITV produit, quant à lui, un TAM= 73,15min., CELL_ITV un TAM= 73,50min., C+R un TAM= 76,42 min., et en dernier CELL un TAM= 78,47min.

La BRC reste LQE+SGOS pour S3.1b incluant 2 pannes mais son TAM est dégradé et augmente de 55,2 secondes, et passe à 81,90min. En ce qui concerne les HH, à l'exception d'ITV qui voit sont TAM légèrement amélioré à 72,93min., CELL_ITV augmente à 74,50min., de même que C+R à 76,63min., et CELL à 79,18min.

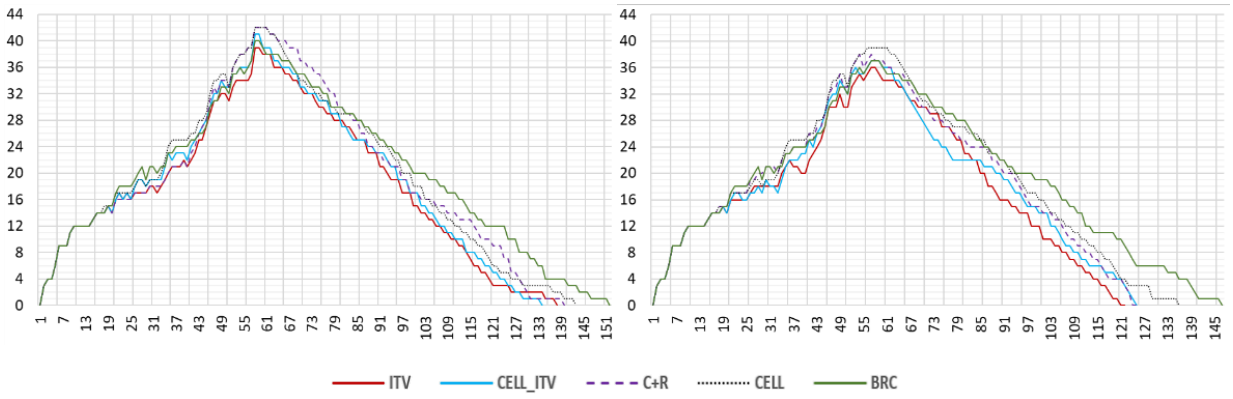


Figure 80 Evolutions du nombre de produits dans le FMS scenarios S3.1a et S3.1b

La Figure 81 montre les moyennes des TAM, avec et sans panne. En considérant la moyenne des TAM pour les 16 scenarios de FJ1, ITV arrive en premier avec 117,32min., suivie de CELL_ITV avec 118,57min., C+R avec 121,16min., puis CELL avec 122,18min., et enfin les combinaisons BRC avec un TAM moyen de 127,65 min.

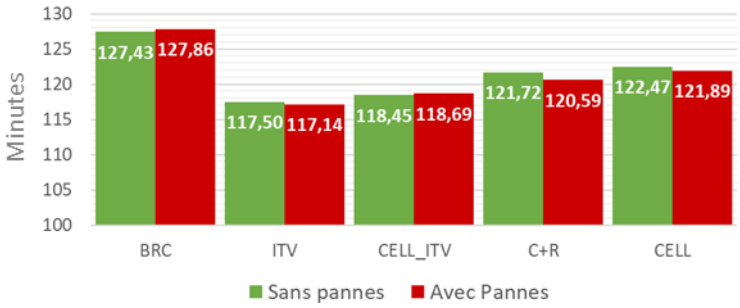


Figure 81 Moyenne des TAM (avec et sans panne) pour la configuration FJ1

Comme le montre la Figure 82, les performances des HH restent toujours assez fortement corrélées à la cadence des arrivées de produits. Cependant, contrairement à la configuration HF2, ITV ne culmine plus qu'à 16%.

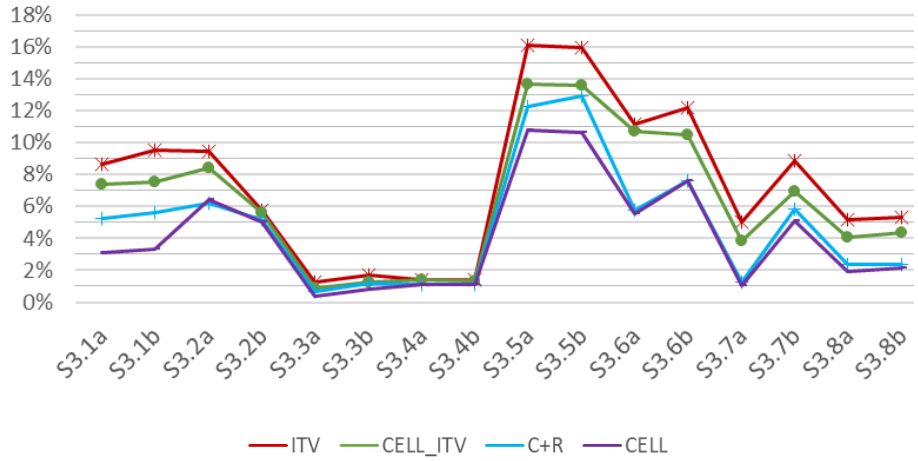


Figure 82 Performances relatives des HH par rapport aux BRC pour FJ1

#	Scenario	Arrivées	Pannes	BRC		ITV			CELL_ITV			C+R			CELL		
				MSR+SR	TAM	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max
1	S3.1a	60/60	-	LQE+SGOS	80,98	73,15	74,00	74,80	73,50	75,02	76,92	76,42	76,74	78,07	78,47	78,47	78,47
2	S3.1b	60/60	2	LQE+SGOS	81,90	72,93	74,08	75,72	74,50	75,71	77,00	76,63	77,33	77,75	79,18	79,18	79,18
3	S3.2a	60/120	-	LQE+LJF	92,15	82,77	83,42	84,12	82,98	84,38	85,88	85,98	86,44	86,78	86,22	86,25	86,30
4	S3.2b	60/120	3	LQE+SJF	85,57	79,85	80,66	82,45	80,20	80,78	82,07	80,68	81,17	84,70	81,30	81,30	81,30
5	S3.3a	60/180	-	STPT+LPF	117,08	115,58	115,62	115,63	115,90	116,04	116,12	116,28	116,28	116,28	116,63	116,63	116,63
6	S3.3b	60/180	4	LQE+HPF	117,68	115,53	115,67	116,12	116,12	116,17	116,22	116,28	116,28	116,28	116,68	116,68	116,68
7	S3.4a	60/240	-	STG+FIFO	127,95	126,13	126,15	126,18	126,17	126,20	126,23	126,48	126,48	126,48	126,48	126,48	126,48
8	S3.4b	60/240	13	STG+FIFO	127,95	126,17	126,18	126,20	126,18	126,27	126,40	126,48	126,48	126,48	126,48	126,48	126,48
9	S3.5a	120/60	-	STPT+FASFO	147,11	121,55	123,42	125,19	126,89	127,04	127,15	128,94	129,09	129,44	131,26	131,26	131,26
10	S3.5b	120/60	1	STPT+FASFO	147,13	123,14	123,65	124,54	125,37	127,14	128,13	128,08	128,08	128,08	131,34	131,48	132,78
11	S3.6a	120/120	-	STG+LJF	147,90	130,98	131,42	132,23	131,22	132,10	133,52	139,33	139,40	139,64	139,73	139,73	139,73
12	S3.6b	120/120	4	STG+LJF	149,32	130,48	131,09	132,08	132,69	133,68	134,73	138,01	138,01	138,01	138,01	138,01	138,01
13	S3.7a	120/180	-	LQE+LJF	142,93	135,31	135,72	136,20	136,26	137,47	138,77	141,11	141,11	141,11	141,48	141,48	141,48
14	S3.7b	120/180	6	STPT+LJF	149,85	135,55	136,52	138,23	138,47	139,46	140,19	141,01	141,08	141,72	142,17	142,17	142,17
15	S3.8a	120/240	-	STPT+LPF	163,35	154,50	154,94	155,28	154,69	156,74	157,54	159,19	159,53	159,76	159,50	160,23	161,08
16	S3.8b	120/240	13	STPT+LPF	163,49	153,43	154,85	156,10	156,00	156,34	156,69	157,58	159,57	159,79	159,99	159,99	159,99

Tableau 30 Synthèse des résultats pour FJ1

4.4. JOBSHOP FLEXIBLE FJ2

4.4.1. ANOVA FJ2

L'analyse ANOVA montre que dans 7 tests sur 8, les facteurs ont une incidence sur les performances des HH. L'exception étant la variable Rate sur l'HH CELL.

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	101,167822	4	25,2919556	24,0241225	**6,0686E-14**
Rate	15,4074444	4	3,85186111	3,65877534	**0,00798679**
Pop*Rate	12,5065111	16	0,78165694	0,74247411	0,74448278

Tableau 31 ANOVA de l'algorithme "ITV" pour FJ2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	65,1365689	4	16,2841422	20,3161257	**2,8436E-12**
Rate	13,6393911	4	3,40984778	4,25413233	**0,00320035**
Pop*Rate	13,5981867	16	0,84988667	1,06032016	0,40275805

Tableau 32 ANOVA de l'algorithme "CELL_ITV" pour FJ2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	32,6678089	4	8,166952222	23,9177421	**6,7502E-14**
Rate	2,14714222	4	0,536785556	1,57203056	0,1876053
Pop*Rate	3,16061333	16	0,197538333	0,5785109	0,89331825

Tableau 33 ANOVA de l'algorithme "CELL" pour FJ2

Variations	Sommes des carrés	Degré de liberté	Moyennes	F	Probabilité
Pop	29,4743689	4	7,36859222	15,5972516	**5,9942E-10**
Rate	9,60663556	4	2,40165889	5,08364104	**0,00090339**
Pop*Rate	6,99694222	16	0,43730889	0,92566077	0,54276584

Tableau 34 ANOVA de l'algorithme "C+R" pour FJ2

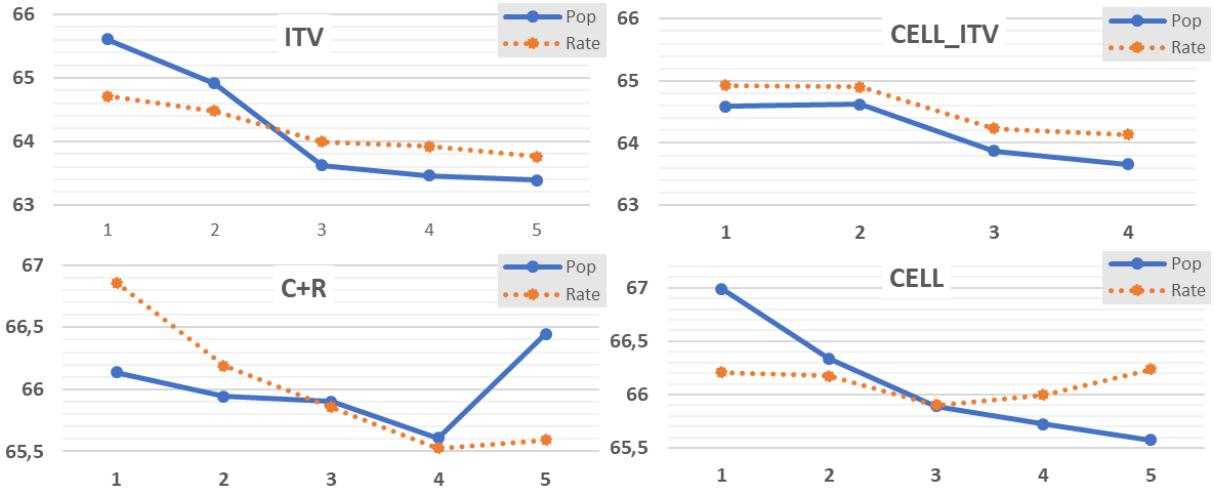


Figure 83 Tracé des principaux effets des paramètres des AG pour FJ2

Comme le montre la Figure 83, il existe une assez forte disparité dans les valeurs des paramètres. Ceux-ci sont indiqués dans le Tableau 35 ci-dessous.

Paramètres	ITV	CELL_ITV	CELL	C+R
Pop	1000	1000	750	1000
Rate	50%	50%	80%	90%

Tableau 35 Paramètres des HH après ANOVA pour FJ2

4.4.2.RESULTATS FJ2

A l'instar des autres configurations, 4 608 simulations ont été conduites et synthétisées dans le Tableau 36. Les scénarios 4.1a, 4.1b sont pris comme exemples illustratifs, et sont détaillés ci-après. Le complément des figures présentées ici est disponible en annexe B.4.

Pour l'instance S4.1a, la BRC est STPT+FASFO avec un TAM=69,75min. ITV est plus performante avec une TAM=54,95min. CELL_ITV est légèrement moins performante avec un TAM= 55,45min. C+R arrive en troisième position avec un TAM=56,35min. CELL se positionne en dernier avec un TAM= 57,07min.

Malgré les 2 pannes du scénario S4.1b, la BRC reste STPT+FASFO avec un TAM = 69,75min. Les performances des HH varient légèrement avec des TAM de 55,40min. pour ITV, de 56,13min. pour CELL_ITV, de 56,07min. pour C+R et de 57,07min. pour CELL.

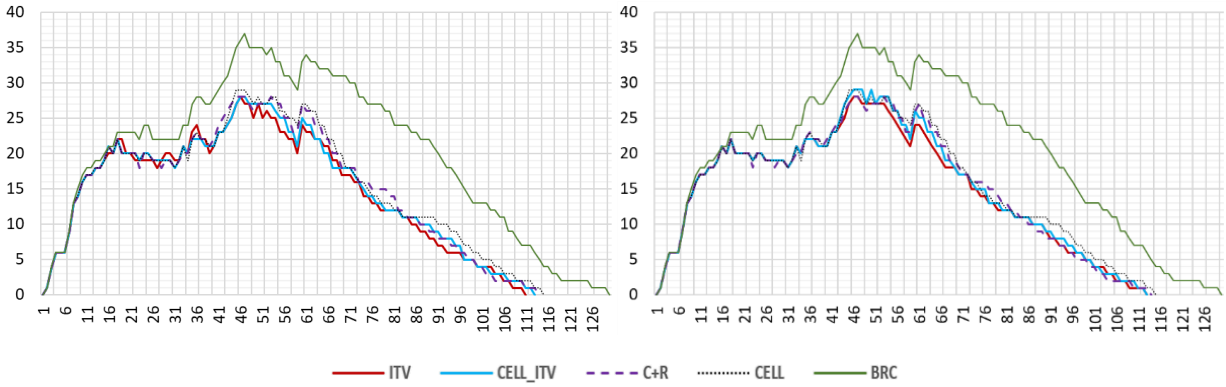


Figure 84 Evolutions du nombre de produits dans le FMS scenarios S4.1a & S4.1b

La Figure 85 montre les moyennes des TAM des 16 scénarios (avec et sans panne) pour la configuration FJ2. Les BRC demeurent les moins performantes avec un TAM moyen de 120,22min. ITV est la plus performante avec un TAM moyen=107,93min. En deuxième position se situe CELL_ITV avec un TAM de 108,05min. Et, enfin, se positionnent en dernier les deux HH sans intervalles C+R et CELL, avec respectivement des TAM de 109,14min. et de 109,49min.

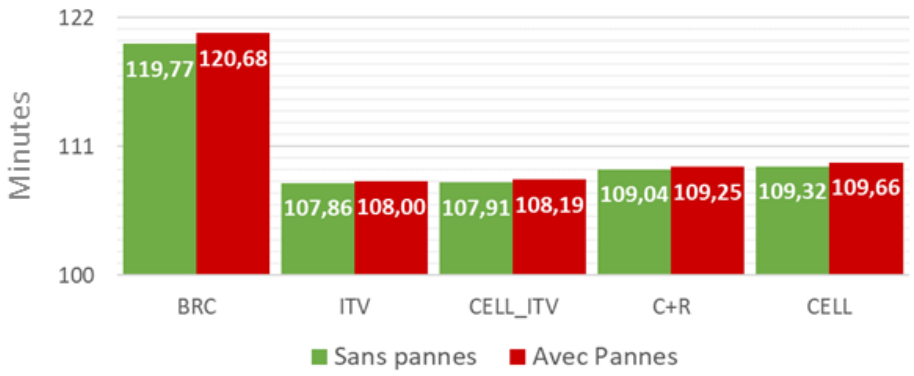


Figure 85 Moyenne des TAM (avec et sans panne) pour la configuration FJ2

De plus, comme le montre la Figure 86, bien que la corrélation avec la cadence des arrivées de produits soit toujours visible, on remarque que les performances des 4 HH sont particulièrement proches. Ceci traduit l'apport moindre des HH avec intervalles comparativement aux configurations d'ateliers précédentes.

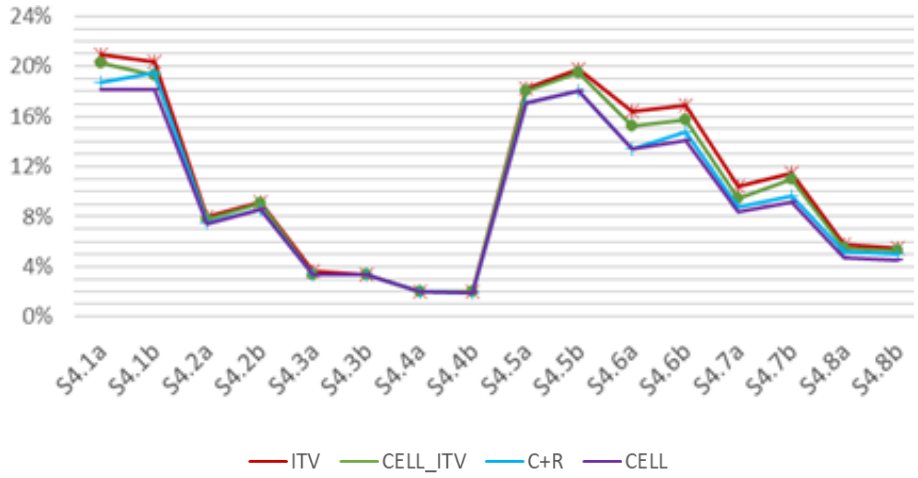


Figure 86 Performances relatives des HH par rapport aux BRC

#	Scenario	Arrivées	Pannes	BRC		ITV			CELL_ITV			C+R			CELL		
				MSR+SR	TAM	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max	Min	Moy.	Max
1	S4.1a	60/60	-	LQE+SGOS	69,75	55,40	55,58	55,73	54,95	55,37	55,70	56,35	56,58	56,70	57,07	57,08	57,10
2	S4.1b	60/60	2	LQE+SGOS	69,75	55,40	55,69	55,85	56,13	56,24	56,43	56,07	56,47	56,70	57,07	57,08	57,10
3	S4.2a	60/120	-	LQE+LJF	85,63	78,77	78,80	78,82	78,72	78,86	79,02	79,23	79,24	79,27	79,25	79,26	79,27
4	S4.2b	60/120	10	LQE+SJF	86,75	78,82	78,85	78,92	78,78	78,85	78,98	79,30	79,31	79,33	79,33	79,33	79,33
5	S4.3a	60/180	-	STPT+LPF	110,12	106,30	106,35	106,40	105,97	106,26	106,40	106,45	106,45	106,45	106,45	106,45	106,45
6	S4.3b	60/180	1	LQE+HPF	110,12	106,40	106,40	106,40	106,40	106,40	106,40	106,45	106,45	106,45	106,45	106,45	106,45
7	S4.4a	60/240	-	STG+FIFO	153,20	150,10	150,10	150,10	150,10	150,10	150,10	150,17	150,17	150,17	150,17	150,17	150,17
8	S4.4b	60/240	6	STG+FIFO	153,22	150,17	150,17	150,18	150,18	150,20	150,23	150,25	150,25	150,25	150,25	150,25	150,25
9	S4.5a	120/60	-	STPT+FASFO	126,33	103,38	103,51	103,68	103,24	103,47	103,63	104,74	104,75	104,75	104,75	104,75	104,75
10	S4.5b	120/60	2	STPT+FASFO	128,68	103,33	103,47	103,61	103,18	103,41	103,83	105,44	105,44	105,44	105,44	105,44	105,44
11	S4.6a	120/120	-	STG+LJF	124,43	104,02	104,46	105,18	104,18	105,11	106,18	107,38	107,95	108,33	107,43	107,94	108,97
12	S4.6b	120/120	4	STG+LJF	126,22	104,55	104,98	105,42	104,96	105,89	106,64	107,41	107,68	108,03	108,18	108,48	108,69
13	S4.7a	120/180	-	LQE+LJF	142,79	128,51	129,43	129,88	127,57	128,07	128,43	129,81	130,27	130,64	130,46	130,76	131,07
14	S4.7b	120/180	6	STPT+LJF	144,73	127,82	128,35	128,83	127,93	128,64	129,10	130,40	130,84	131,13	131,31	131,51	131,68
15	S4.8a	120/240	-	STPT+LPF	145,89	137,23	137,49	138,03	137,59	137,95	138,38	138,17	138,29	138,73	138,94	138,97	139,00
16	S4.8b	120/240	13	STPT+LPF	145,97	137,58	137,99	138,31	137,89	138,21	138,40	138,64	138,73	139,05	139,24	139,35	139,44

Tableau 36 Synthèse des résultats pour FJ2

5. DISCUSSION

Cette section se propose de synthétiser et de discuter les différents résultats des quelques 30 400 simulations décrites précédemment. Les changements de règles à travers l'évolution des BRC sont analysés dans un premier temps. Puis, dans un second temps, les performances des différentes HH.

5.1. ETUDE DE L'EVOLUTION DES BRC

Cette section permet d'étudier l'évolution des BRC selon l'incidence des configurations et des pannes.

5.1.1. INCIDENCE DES CONFIGURATIONS

En observant les meilleures combinaisons BRC sur l'ensemble des 4 configurations d'atelier (Figure 87), on observe que seules 5 règles MSR sur les 14 implémentées apparaissent dans les BRC des 64 instances : 3 règles pour HF1 et FJ1, 2 règles pour HF2 et 4 règles pour FJ2.

Les résultats relatifs aux règles de séquencement montrent que toutes les règles apparaissent au moins une fois dans les combinaisons BRC. Si la configuration HF1 fait office d'exception avec 2 règles sur 12, les 3 autres configurations HF2, FJ1 et FJ2 utilisent à chaque fois 7 règles différentes.

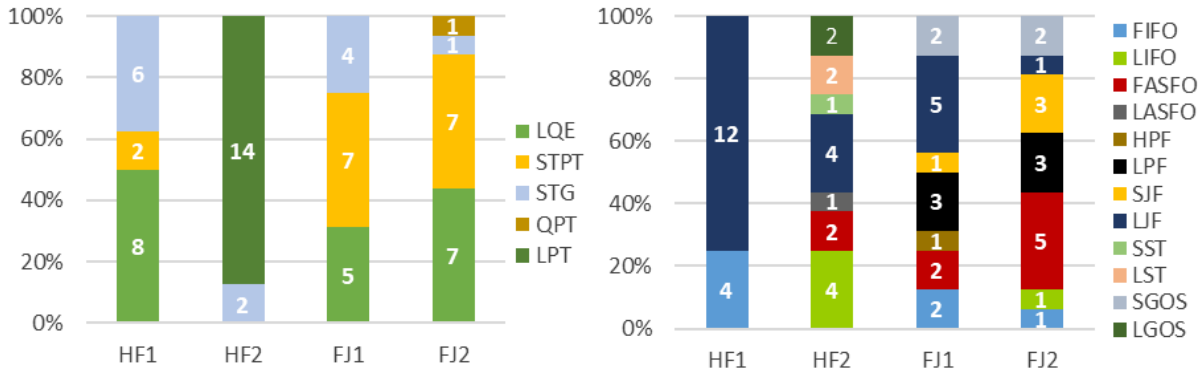


Figure 87 Occurrences des règles MSR et SR dans les 64 BRC

5.1.2. INCIDENCE DES PANNES

A script identique (i.e. même configuration et même arrivées de produits), l'occurrence de pannes induit un changement dans la BRC. Ainsi, une combinaison de règles la plus performante pour un scénario sans perturbations est supplantée par une autre combinaison en présence de pannes. Cette nouvelle combinaison BRC peut se traduire par un changement de règle d'allocation (MSR), de séquencement (SR) ou les deux. Comme le montre Figure 88, cette incidence des pannes sur l'évolution des BRC est de plus en plus forte en fonction de la complexité de la configuration de l'atelier. Si le changement de règles est relativement limité pour les configurations Flowshops (Une seule règle change dans 25% des cas), il est nettement plus marqué pour FJ2 (au moins une règle change dans 75% des cas).

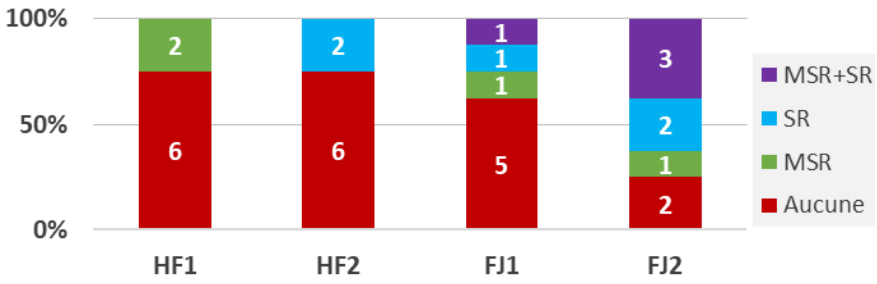


Figure 88 Changement de règles dans les BRC pour les scenarios avec et sans pannes

Ces résultats montrent la nécessité de changement de règles fréquent (au niveau SR, MSR ou les deux), de façon à s'adapter à des configurations de plus en plus complexes et à l'occurrence de perturbations (i.e. variation des arrivées et des pannes).

5.2. PERFORMANCES DES HH

Comme le montre la Figure 89, quel que soit la configuration d'atelier traitée, les HH apportent un gain de performance par rapport à l'approche BRC. Pour toutes les configurations, le classement des HH reste inchangé. Les HH avec intervalles se démarquent toujours des HH sans intervalles. L'approche proposée ITV se révèle toujours la plus performante.

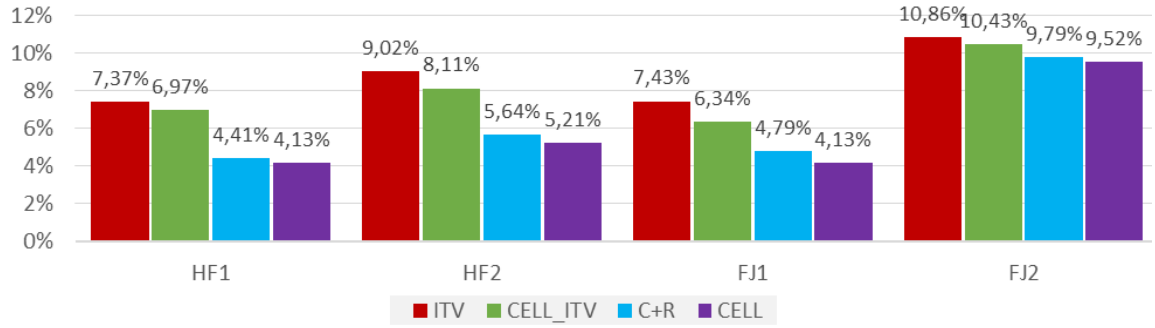


Figure 89 Moyennes des performances relatives de chaque HH par configuration

Les trois prochaines sections sont dédiées à l'étude des incidences respectives des cadences d'arrivées de produits et de pannes, et de la complexité des stratégies décisionnelles sur les performances des HH.

5.2.1. INCIDENCE DES CADENCES D'ARRIVEES

Comme vu précédemment, la performance des différentes HH est fortement corrélée à la cadence des arrivées des produits. Cependant, comme le montre la Figure 90, les HH se révèlent particulièrement performantes dans le cas de FMS fortement sollicités ; la cadence importante des arrivées entrainant des goulets d'étranglement. La capacité des PI à combiner plusieurs règles dans le cadre des HH, permet de fluidifier plus efficacement le flux de produits comparativement à une utilisation de règles uniques. Grâce à la prise en compte du contexte, les PI sont à même de sélectionner et d'appliquer les règles les plus pertinente pour le choix de la machine et le séquencement de la file d'attente. Comparativement au comportement « myope » exhibé par BRC, une perception de plus en plus fine du contexte est réalisée au travers des différentes HH (de CELL à ITV). Ceci se traduit par une amélioration graduelle des performances des HH liée à une réduction progressive de la myopie.

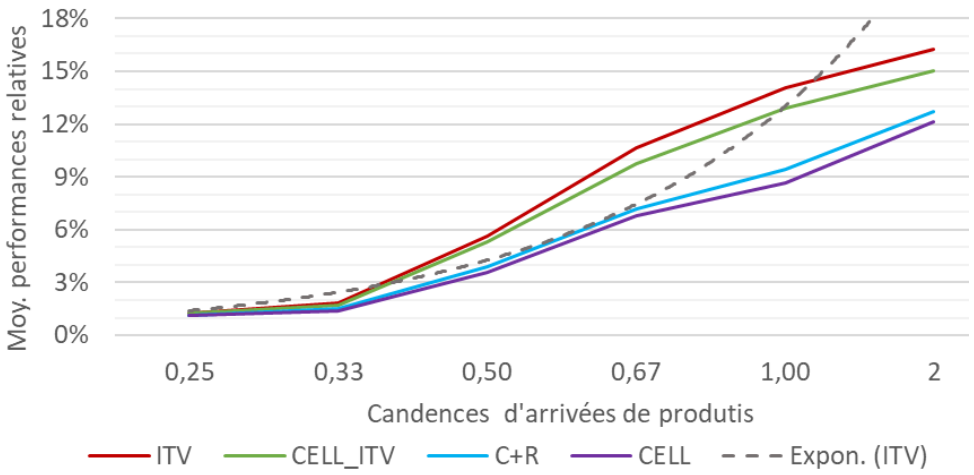


Figure 90 Moyennes des Performances relatives des HH par cadences d'arrivée

5.2.2. INCIDENCE DES PANNES

Comme le montre la Figure 91, en plus d'être plus performantes, les HH sont plus résilientes aux pannes que les BRC.

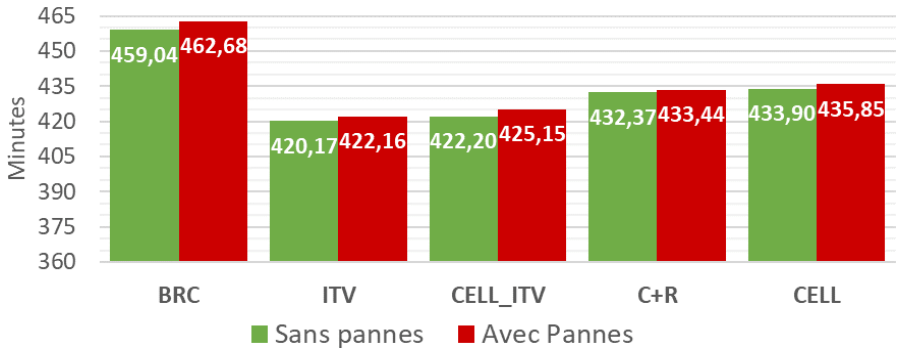


Figure 91 Incidences des pannes

Un phénomène contre-intuitif apparait néanmoins. Dans certains cas, les performances obtenues en présence de pannes peuvent être légèrement meilleures (de l'ordre de quelques secondes) que celles obtenues sans panne. Ceci peut s'expliquer par la sélection d'une ressource peu performante dans un scénario sans panne. Quel que soit l'algorithme, les PI ignorent des ressources quand celles-ci sont en pannes. De ce fait, si une panne intervient au moment « propice », elle aura pour effet de corriger le choix du PI.

La présence de tels phénomènes démontre toute la complexité du pilotage d'un FMS dans un environnement fortement perturbé.

5.2.3. INCIDENCE DE LA COMPLEXITE DES STRATEGIES DECISIONNELLES

Concernant la convergence des HH, une corrélation est observée entre les écarts types des performances et la complexité des stratégies décisionnelles associées aux HH. Bien que les écarts soient relativement faibles, on observe une différence entre les HH avec intervalles (i.e. ITV et CELL_ITV) et celles sans intervalles (i.e. C+R et CELL). Ceci s'explique par la taille de plus en plus importante de l'espace de recherche parcouru par l'AG, induite par la complexité progressive des stratégies décisionnelles. Même si ITV et CELL_ITV finissent par converger vers des solutions plus performantes, on remarque une plus grande difficulté de convergence. Cet effet nous laisse entrevoir les limites de ce type d'approche dans la prise en compte d'un

nombre de contextes toujours plus important. Ceci n'est pas sans rappeler le problème d'overfitting induit par la prise en compte excessive de termes dans les modèles et/ou d'approches s'avérant plus complexes que nécessaire (Hawkins 2004).

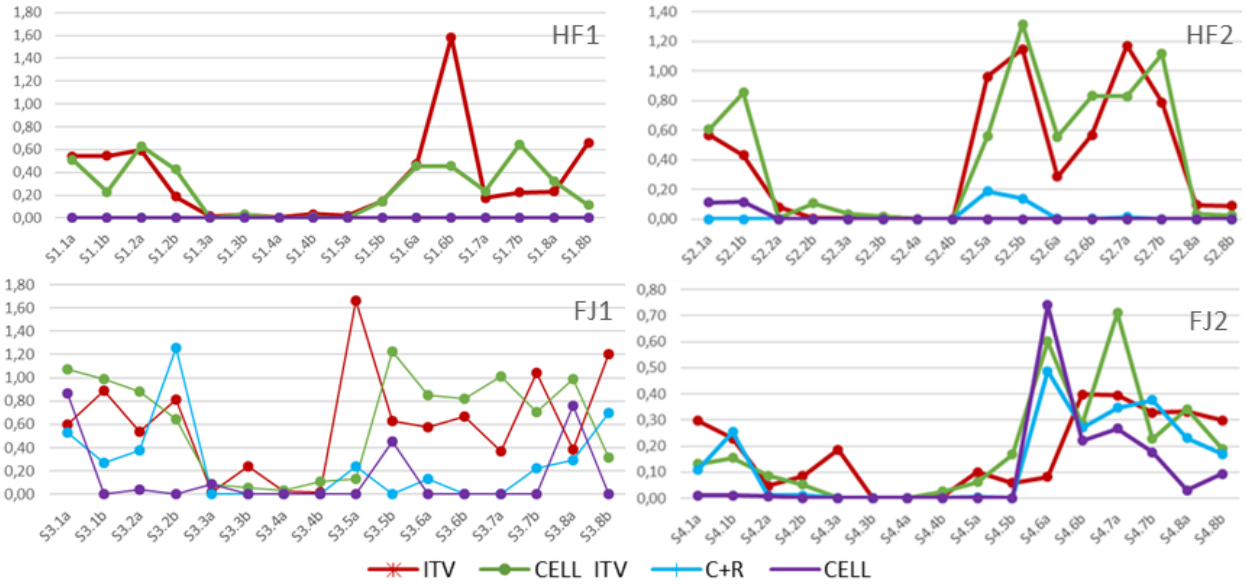


Figure 92 Ecart-Type des HH pour les 4 configurations

6. CONCLUSION

Ce quatrième chapitre a permis de synthétiser et de discuter les résultats des quelques 30 400 simulations effectuées à l'aide de l'outil de simulation multiagents. L'étude de quatre ateliers inspirés de l'industrie pharmaceutique (deux flowshop hybrides flexibles, ainsi que deux jobshop flexibles) a permis d'éprouver un grand nombre de paramètres expérimentaux. Le générateur de scénarios a permis de créer 64 instances de problèmes avec différentes cadences d'arrivées de produits et d'occurrences de pannes aléatoires.

L'étude de l'évolution des BRC a d'abord démontré la nécessité d'un changement de règles fréquent de façon à s'adapter à des configurations de plus en plus complexes et à l'occurrence de perturbations (i.e. variation des arrivées et des pannes).

Les résultats comparatifs avec les règles d'ordonnancement classiques de la littérature, mais aussi avec les différentes variantes d'HH ont permis de mettre en lumière la supériorité du modèle de stratégie ITV proposé. De plus, une amélioration graduelle des performances des HH liée à une réduction progressive de la myopie est observée, démontrant ainsi l'efficacité de la contextualisation proposée pour améliorer les comportements des PI.

CONCLUSION GENERALE ET PERSPECTIVES

Cette thèse est une contribution au pilotage par le produit des systèmes de production manufacturiers *(ou Product-Driven Manufacturing)*. Avec les prémices de la quatrième révolution industrielle, les systèmes de productions deviennent « cyber-physiques ». Les nouveaux paradigmes de l'industrie 4.0 posent de nouveaux défis aussi bien technologiques que concurrentiels. Dans ce contexte, un ordonnancement efficace des opérations de production est, plus que jamais, indispensable à la pérennité des entreprises manufacturières.

La problématique abordée concerne des systèmes de production inspirés de l'industrie pharmaceutique. Ces systèmes se distinguent essentiellement par deux aspects ; le premier concerne la flexibilité partielle des cellules de productions. Due généralement à des ressources hétérogènes, elle impacte fortement la complexité du problème d'ordonnancement. Le second aspect concerne la nature dynamique inhérente à la plupart des environnements industriels. En effet, les incertitudes liées aux évènements imprévus, tels que les arrivées dynamiques de produits et les pannes machines, accroissent la complexité.

L'approche proposée se veut aussi générique que possible et donc applicable à un maximum de configurations d'ateliers différentes. Elle souhaite apporter des réponses rapides et adaptées à des environnements complexes en perpétuelle évolution. Dans un tel contexte, une architecture de contrôle distribuée parait naturellement la plus appropriée. Comparée aux approches centralisées, les approches décentralisées s'avèrent plus agiles et plus réactives. Néanmoins, elles sont limitées par le phénomène de myopie due à des prises de décision locales. Une étude de la littérature relative à l'ordonnancement, bien que prolifique, a permis de constater que très peu de travaux proposent l'utilisation de techniques telle que l'apprentissage via simulation couplé aux systèmes pilotés par le produit.

L'approche proposée dans le cadre de ce mémoire repose sur l'utilisation d'une hyper-heuristique couplée à une technique d'optimisation-simulation. Elle met à profit des règles d'ordonnancement connues pour être rapides et extrêmement réactives. Cet ensemble de règles est utilisé par des produits intelligents et totalement autonomes au niveau décisionnel. Le mécanisme d'optimisation-simulation associé permet de rechercher progressivement les stratégies les plus réactives, tout en veillant à une bonne performance globale. Après expérimentation sur 4 configurations d'ateliers, les résultats obtenus par simulation confirment l'intérêt de l'approche proposée.

Les principales contributions peuvent être synthétisées comme suit :

- Proposition d'une architecture de pilotage totalement hétérarchique, basée sur des produits intelligents et autonomes au niveau décisionnel.
- Développement du concept de contexte décisionnel permettant au PI d'effectuer une sélection efficace de règles d'ordonnancement.
- Développement d'une approche à base d'hyper-heuristique couplée à un mécanisme d'optimisation-simulation.

Le développement des concepts proposés ayant été réalisé de manière incrémentale, les différents résultats ont fait l'objet de plusieurs publications durant ce travail de thèse :

- Une première publication (Bouazza et al. 2015) dans SOHOMA a proposé une approche de

pilotage par le produit utilisant une technique d'apprentissage par renforcement

– Un article présenté dans le cadre de l'IFAC Word Congress (Bouazza et al. 2017) a traité de l'application d'un algorithme Q-learning au pilotage d'un jobshop partiellement flexible.
– Un article dans la revue *Manufacturing Letters* (Bouazza et al. 2019) a présenté l'hyper-heuristique développée dans le cadre de cette thèse.

Ces travaux présentent des résultats intéressants qui ouvrent d'importantes perspectives de recherche, dont les suivantes.

A court terme :

- D'explorer d'autres voies pour permettre une meilleure prise en compte de la contextualisation (par exemple, par la prise en compte d'autres variables discriminantes).
- De répondre au problème de convergence mis en évidence dans la section IV-5.2.3, par une étude approfondie des facteurs influents.
- De prendre en compte des contraintes supplémentaires, par exemple, en considérant des temps de transport non nuls ou encore des files à capacités finies.

A moyen terme :

- D'appliquer l'approche dans un environnement réel. Il est notamment prévu un déploiement de l'approche sur la cellule flexible de production du pôle S-mart de Valenciennes. Cette cellule s'articule autour d'un système de convoyage flexible et comprend trois postes de travail robotisés, un poste d'inspection par vision, ainsi qu'un poste de travail manuel.
- D'expérimenter un autre type d'hyper-heuristiques permettant la génération de nouvelles règles d'ordonnancement, notamment des HH basées sur les grammaires.
- De tester l'approche dans d'autres domaines applicatifs, tels que le secteur hospitalier ou la logistique.

BIBLIOGRAPHIE

Adam E., Berger T., Sallez Y., Trentesaux D. 2009. An Open-Control Concept for a Holonic Multiagent System. In V. Mařík, T. Strasser, & A. Zoitl (eds) *Lecture Notes in Computer Science*, 145–154.

Adam E., Zambrano G., Pach C., Berger T., Trentesaux D. 2011. Myopic behaviour in holonic multiagent systems for distributed control of FMS. In *Advances in Intelligent and Soft Computing*, 91–98.

Adamo A. et al. 2016. On-demand continuous-flow production of pharmaceuticals in a compact, reconfigurable system. *Science (80-.).* (352) p.61–67.

Aggoune R. 2004. Minimizing the makespan for the flow shop scheduling problem with availability constraints. *Eur. J. Oper. Res.* (153) p.534–543.

Agrawal S., Gautam A., Chauhan D.K., Tiwari L.M., Kapoor S. 2012. A Flow Shop Scheduling Problem with Transportation Time and Separated Setup Time of Jobs. *Procedia Eng.* (38) p.1327–1332.

Ahonen H., de Alvarenga A.G. 2017. Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: formulation and solution procedures. *Int. J. Adv. Manuf. Technol.* (89) p.765–777.

Alcaide D., Rodriguez-Gonzalez A., Sicilia J. 2006. A heuristic approach to minimize expected makespan in open shops subject to stochastic processing times and failures. *Int. J. Flex. Manuf. Syst.* (17) p.201–226.

Allahverdi A. 2015. The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* (246) p.345–378.

Allahverdi A. 2016. A survey of scheduling problems with no-wait in process. *Eur. J. Oper. Res.* (255) p.665–686.

Allahverdi A., Al-Anzi F.S. 2009. The two-stage assembly scheduling problem to minimize total completion time with setup times. *Comput. Oper. Res.* (36) p.2740–2747.

Allahverdi A., Aydilek H. 2014. Total completion time with makespan constraint in no-wait flowshops with setup times. *Eur. J. Oper. Res.* (238) p.724–734.

Allahverdi A., Gupta J.N.., Aldowaisan T. 1999. A review of scheduling research involving setup considerations. *Omega* (27) p.219–239.

Allahverdi A., Ng C.T.T., Cheng T.C.E.C.E., Kovalyov M.Y. 2008. A survey of scheduling problems with setup times or costs. *Eur. J. Oper. Res.* (187) p.985–1032.

Anderl R., Picard A., Albrecht K. 2013. Smart Product Engineering. In M. Abramovici & R. Stark (eds) *23rd CIRP Design Conference*.Lecture Notes in Production Engineering, 1–10.

Armetta F. 2006. *Proposition d'une approche auto-organisationnelle pour le partage de ressources critiques*.

Arroyo J.E.C., Leung J.Y.T., Tavares R.G. 2019. An iterated greedy algorithm for total flow time minimization in unrelated parallel batch machines with unequal job release times. *Eng. Appl. Artif. Intell.* (77) p.239–254.

Arvind A., Smith S., Rubinstein Z., Bourne D., Chuang C.-Y. 2016. *Architecture for Industry 4.0-based Manufacturing Systems*.

Aydilek H., Allahverdi A. 2012. Heuristics for no-wait flowshops with makespan subject to mean completion time. *Appl. Math. Comput.* (219) p.351–359.

Bagchi U. 1989. Simultaneous Minimization of Mean and Variation of Flow Time and Waiting Time in Single Machine Systems. *Oper. Res.* (37) p.118–125.

Bagheri B., Yang S., Kao H.A., Lee J. 2015. Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. (28) p.1622–1627.

Bai D., Zhang Z.H., Zhang Q. 2016. Flexible open shop scheduling problem to minimize

makespan. *Comput. Oper. Res.* (67) p.207–215.
Baker A. 1991. *Manufacturing Control with a Market-Driven Contract Net.*
Baños R. et al. 2011. Optimization methods applied to renewable and sustainable energy: A review. *Renew. Sustain. Energy Rev.* (15) p.1753–1766.
Barbosa J., Leitao P., Pereira A.I. 2011. Combining adaptation and optimization in bio-inspired multi-agent manufacturing systems. In *2011 IEEE International Symposium on Industrial Electronics*, 1773–1778.
Bateman R.J., Cheng K. 2006. Extending the product portfolio with 'devolved manufacturing': Methodology and case studies. *Int. J. Prod. Res.* (44) p.3325–3343.
Bellifemine F., Caire G., Poggi A., Rimassa G. 2003. Jade - a white paper. In *Telecom Lab Italia*, 6--19.
Berchet C., Habchi G., Courtois A. 1999. Integration Du Processus De Pilotage a La Simulation Des Systemes De Production. In *2ème conférence Francophone de MOdélisation et SIMulation (MOSIM'99)*, 337–343.
Berger T., Sallez Y., Valli B., Gibaud A., Trentesaux D. 2010. Semi-heterarchical Allocation and Routing Processe. In *Journal of Intelligent and Robotic Systems: Theory and Applications*, 17–45.
Bettenhausen K.D., Kowalewski S. 2013. *Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation - Thesen und Handlungsfelder*.
Bierwirth C., Mattfeld D.C. 1999. Production Scheduling and Rescheduling with Genetic Algorithms. *Evol. Comput.* (7) p.1–17.
Birgin E.G. et al. 2014. A MILP model for an extended version of the Flexible Job Shop Problem. *Optim. Lett.* (8) p.1417–1431.
Biskup D., Herrmann J. 2008. Single-machine scheduling against due dates with past-sequence-dependent setup times. *Eur. J. Oper. Res.* (191) p.587–592.
Blackstone J.H., Phillips D.T., Hogg G.L. 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *Int. J. Prod. Res.* (20) p.27–45.
Blunck H., Bendul J. 2016. Controlling Myopic Behavior in Distributed Production Systems - A Classification of Design Choices. *Procedia CIRP* (57) p.158–163.
Bokhorst J.A.C., Nomden G., Slomp J. 2008. Performance evaluation of family-based dispatching in small manufacturing cells. *Int. J. Prod. Res.* (46) p.6305–6321.
Borangiu T., Rəileanu S., Berger T., Trentesaux D. 2015. Switching mode control strategy in manufacturing execution systems. *Int. J. Prod. Res.* (53) p.1950–1963.
Bouazza W. 2013. *Pilotage de cycle de vie à base de produits intelligents : application à l'industrie pharmaceutique*.
Bouazza W., Hamdadou D., Sallez Y., Trentesaux D. 2019. Effective dynamic selection of smart products scheduling rules in FMS. *Manuf. Lett.* (20) p.45–48.
Bouazza W., Sallez Y., Aissani N., Beldjilali B. 2015. A model for manufacturing scheduling optimization through learning intelligent products. In *Studies in Computational Intelligence*, 233–241.
Bouazza W., Sallez Y., Beldjilali B. 2017. A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect. In *IFAC-PapersOnLine*, 15890–15895.
Boudhar M., Meziani N. 2010. Two-stage hybrid flow shop with recirculation. *Int. Trans. Oper. Res.* (17) p.239–255.
Boukef H. 2011. *Sur l'ordonnancement d'ateliers job-shop flexibles et flow-shop en industries pharmaceutiques: optomisation par algorithmes génétiques et essaims particulaires*.
Bousbia S., Trentesaux D. 2002. Self-organization in distributed manufacturing control: State-of-the-art and future trends. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 661–666.
Brandl D. 2002. Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95. *Rev. l'Electricité l'Electronique* p.46.
Branke J., Nguyen S., Pickardt C.W., Zhang M. 2016. Automated Design of Production

Scheduling Heuristics: A Review. *IEEE Trans. Evol. Comput.* (20) p.110–124.
Brun-Picard D., Bouvet H., Baboli H., Binder Z. 1997. The Product as an Active Element of Distributed Production Control. *IFAC Proc. Vol.* (30) p.1359–1365.
Van Brussel H., Wyns J., Valckenaers P., Bongaerts L., Peeters P. 1998. Reference architecture for holonic manufacturing systems: PROSA. *Comput. Ind.* (37) p.255–274.
Burke E., Hyde M., Kendall G. 2010. A classification of hyper-heuristic approaches. *Handb. ...* p.1–21.
Burke E.K. et al. 2019. A classification of hyper-heuristic approaches: Revisited. In M. G. J.-Y. Potvin (ed) *International Series in Operations Research and Management Science*, 453–477.
Caramia M., Dell'Olmo P. 2006. *Effective Resource Management in Manufacturing Systems*.
Cardin O. et al. 2017. Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. *J. Intell. Manuf.* (28) p.1503–1517.
Cardin O. 2019. Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Comput. Ind.* (104) p.11–21.
Chaari T. 2010. *Un algorithme génétique pour l'ordonnancement robuste: application au problème du flow shop hybride*.
Chaudhry I.A., Khan A.A. 2016. A research survey: Review of flexible job shop scheduling techniques. *Int. Trans. Oper. Res.* (23) p.551–591.
Chen C.F., Wu M.C., Li Y.H., Tai P.H., Chiou C.W. 2013. A comparison of two chromosome representation schemes used in solving a family-based scheduling problem. *Robot. Comput. Integr. Manuf.* (29) p.21–30.
Chen F.F., Adam E.E. 1991. The Impact of Flexible Manufacturing Systems on Productivity and Quality. *IEEE Trans. Eng. Manag.* (38) p.33–45.
Cheng T.C.E., Sin C.C.S. 1990. A state-of-the-art review of parallel-machine scheduling research. *Eur. J. Oper. Res.* (47) p.271–292.
Chirn J.L., McFarlane D.C. 2000. A holonic component-based approach to reconfigurable manufacturing control architecture. *Proc. - Int. Work. Database Expert Syst. Appl. DEXA* (2000-Janua) p.219–223.
Clivillé V. 2004. *Approche systémique et méthode multicritère pour la définition d'un système d'indicateurs de performance*.
Conway R.W., Maxwell W.L. 1962. *Network Dispatching by the Shortest-Operation Discipline*.
Corning P.A. 2002. The re-emergence of "emergence": A venerable concept in search of a theory. *Complexity* (7) p.18–30.
Cottet F., Delacroix J., Kaiser C., Mammeri Z. 2002. *Scheduling in Real-Time Systems*.
Cowling P., Kendall G., Soubeiga E. 2001. A Hyperheuristic Approach to Scheduling a Sales Summit. *Pract. Theory Autom. Timetabling III* p.176–190.
Cronin M.J. 2010. Smart products, smarter services: Strategies for embedded control. *Smart Prod. Smarter Serv. Strateg. Embed. Control* p.1–337.
Dabrowski M., Acton T., van der Heijden H. 2012. Intelligent product search with soft-boundary preference relaxation. *Expert Syst. Appl.* (39) p.8279–8287.
Dan P., Pitman N., Pilone D., Pitman N. 2005. *UML 2.0 in a Nutshell*.
Davenport A. 2010. Integrated maintenance scheduling for semiconductor manufacturing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 92–96.
Deep K., Singh P.K. 2015. Design of robust cellular manufacturing system for dynamic part population considering multiple processing routes using genetic algorithm. *J. Manuf. Syst.* (35) p.155–163.
Demeulemeester E. 2009. Robust Project Scheduling. *Found. Trends® Technol. Inf. Oper. Manag.* (3) p.201–376.
Dhaevers V., Riane F., Duvivier D. 2007. *Vers un pilotage souple de la performance des*

systèmes de production.
Dilts D.M., Boyd N.P., Whorms H.H. 1991. The evolution of control architectures for automated manufacturing systems. *J. Manuf. Syst.* (10) p.79–93.
Dobre D., Bajic E., Zouinkhi A. 2009. Active product modeling based on smart object concept: Application to chemical security management. In *Journal Europeen des Systemes Automatises*, 561–579.
Dong J., Xiao T., Zhang L. 2012. A Prototype Architecture for Assembly-Oriented Cyber-Physical Systems. In *Communications in Computer and Information Science*, 199–204.
Driss I., Mouss K.N., Laggoun A. 2015. A new genetic algorithm for flexible job-shop scheduling problems. *J. Mech. Sci. Technol.* (29) p.1273–1281.
Dubey R., Gunasekaran A., Chakrabarty A. 2017. Ubiquitous manufacturing: overview, framework and further research directions. *Int. J. Comput. Integr. Manuf.* (30) p.381–394.
Duffie N.A. 1990. Synthesis of Heterarchical manufacturing systems. *Comput. Ind.* (14) p.167–174.
Ebrahimi M., Fatemi Ghomi S.M.T., Karimi B. 2014. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Appl. Math. Model.* (38) p.2490–2504.
Eckel C., Neary P.P. 2010. Multi-product firms and flexible manufacturing in the global economy. *Rev. Econ. Stud.* (77) p.188–217.
Eguchi T., Oba F., Hirai T. 1999. A neural network approach to dynamic job shop scheduling K. Mertins, O. Krause, & and B. Schallock (eds). *Glob. Prod. Manag.* (24) p.152–159.
ElMaraghy H.A., El Maraghy H.A. 2006. Flexible and reconfigurable manufacturing systems paradigms. *Flex. Serv. Manuf. J.* (17) p.261–276.
Eloundou J. 2016. *Modélisation multi-contraintes d'unsys eme de production flexible*.
Eloundou J., Baudry D., Bensrhair A. 2015. Evaluation Of The Routing Flexibility Of Flexible Manufacturing System. *Xième Conférence Int. Concept. Prod. Intégrées* p.2–4.
Fahmy A., Hassan T., Bassioni H. 2014. What is Dynamic Scheduling? *PM World J.* (III) p.1–9.
Fan H.L., Xiong H.G., Jiang G.Z., Li G.F. 2016. Survey of the selection and evaluation for dispatching rules in dynamic job shop scheduling problem. In *Proceedings - 2015 Chinese Automation Congress, CAC 2015*, 1926–1931.
Fanjul-Peyro L., Perea F., Ruiz R. 2017. Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *Eur. J. Oper. Res.* (260) p.482–493.
Fattahi P., Fallahi A. 2010. Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP J. Manuf. Sci. Technol.* (2) p.114–123.
Fei H., Meskens N., Chu C. 2010. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Comput. Ind. Eng.* (58) p.221–230.
Ferguson D.F., Nikolaou C., Sairamesh J., Yemini Y. 1996. Market-Based Control: A Paradigm for Distributed Resource Allocation (Scott H. Clearwater, editor). In 311.
Filho M.F., Liao Y., Loures E.R., Canciglieri O. 2017. Self-Aware Smart Products: Systematic Literature Review, Conceptual Design and Prototype Implementation. *Procedia Manuf.* (11) p.1471–1480.
Finnveden G. et al. 2009. Recent developments in Life Cycle Assessment. *J. Environ. Manage.* (91) p.1–21.
Fox M.S. 1994. ISIS: A retrospective. *Intell. Sched.* (1) p.3–28.
Framinan J.M., Leisten R., Ruiz García R. 2014. *Manufacturing Scheduling Systems*.
Främling K., Holmström J., Loukkola J., Nyman J., Kaustell A. 2013. Sustainable PLM through Intelligent Products. *Eng. Appl. Artif. Intell.* (26) p.789–799.
Freitag M., Hildebrandt T. 2016. Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. *CIRP Ann. - Manuf. Technol.* (65) p.433–436.
Gagné C., Sioud A., Gravel M. 2014. Recherches Dans Le Voisinage Pour La Résolution D'Un

Problème De Flowshop Flexible Et Hybride Avec Temps De Réglages Dépendants De La Séquence.

Gaham M., Achour N., Bouzouia B. 2017. Multi-Agent Intelligent Product Driven Manufacturing Control A JADE-based experimentation framework.

Gao K.Z. et al. 2015. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. *Expert Syst. Appl.* (42) p.7652–7663.

Gavett J. 1965. Three heuristic rules for sequencing jobs to a single production facility. *Manage. Sci.* (11) p.166–176.

Gholami M., Zandieh M. 2009. Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop. *J. Intell. Manuf.* (20) p.481–498.

Godinho Filho M., Barco C.F., Tavares Neto R.F. 2014. Using Genetic Algorithms to solve scheduling problems on flexible manufacturing systems (FMS): a literature survey, classification and analysis. *Flex. Serv. Manuf. J.* (26) p.408–431.

Gouyon D. 2007. A product-driven reconfigurable control for shop floor systems. *Stud. Informatics Control* (16) p.??-??

Graham R.L., Lawler E.L., Lenstra J.K., Kan A.H.G.R. 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* (5) p.287–326.

Gudehus T., Kotzab H. 2012. *Comprehensive Logistics*.

Gupta D., Sharma S., Aggarwal S. 2013. Flow shop scheduling on 2-machines with setup time and single transport facility under fuzzy environment. (50) p.14–24.

Gupta J.N.D., Krüger K., Lauff V., Werner F., Sotskov Y.N. 2002. Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Comput. Oper. Res.* (29) p.1417–1439.

Halat K., Bashirzadeh R. 2015. Concurrent scheduling of manufacturing cells considering sequence-dependent family setup times and intercellular transportation times. *Int. J. Adv. Manuf. Technol.* (77) p.1907–1915.

Haleh H., Maghsoudlou H., Hadipour H., Nabovati H. 2017. Scheduling single machine with random breakdown and preemptive jobs. *J. Ind. Prod. Eng.* (34) p.289–299.

Hallgren M., Olhager J. 2009. Flexibility configurations: Empirical analysis of volume and product mix flexibility. *Omega* (37) p.746–756.

Ham M., Lee Y.H., Kim S.H. 2011. Real-time scheduling of multi-stage flexible job shop floor. *Int. J. Prod. Res.* (49) p.3715–3730.

Hamzadayi A., Yildiz G. 2016. Hybrid strategy based complete rescheduling approaches for dynamic m identical parallel machines scheduling problem with a common server. *Simul. Model. Pract. Theory* (63) p.104–132.

Hansmann R.S., Rieger T., Zimmermann U.T. 2014. Flexible job shop scheduling with blockages. *Math. Methods Oper. Res.* (79) p.135–161.

El Haouzi H., Petin J.F., Thomas A. 2009. Design and validation of a product-driven control system based on a six sigma methodology and discrete event simulation. *Prod. Plan. Control* (20) p.510–524.

Hawkins D.M. 2004. The Problem of Overfitting. *J. Chem. Inf. Comput. Sci.* (44) p.1–12.

He W., Sun D.H. 2013. Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies. *Int. J. Adv. Manuf. Technol.* (66) p.501–514.

Heger J., Branke J., Hildebrandt T., Scholz-Reiter B. 2016. Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times. *Int. J. Prod. Res.* (54) p.6812–6824.

Herrera C., Berraf S.B. 2011. Viable System Model approach for holonic product-driven manufacturing systems. In *Service Orientation in Holonic and Multi-Agent Manufacturing Control*, 1–10.

Herrera C., Thomas A. 2009. Simulation of Less Master Production Schedule Nervousness Model. *IFAC Proc. Vol.* (42) p.1585–1590.

Hershauer J.C., Ebert R.J. 1975. Search and Simulation Selection of a Job-Shop Sequencing

Rule. *Manage. Sci.* (21) p.833–843.
Hozdić E. 2015. SMART FACTORY FOR INDUSTRY 40 A REVIEW Hozdic_Elvis - Hozdić 2015 .pdf. *Int. J. Mod. Manuf. Technol.* (VII).
Hu S.J. 2013. Evolving paradigms of manufacturing: From mass production to mass customization and personalization. *Procedia CIRP* (7) p.3–8.
Huang X., Wang M.-Z. 2011. Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties. *Appl. Math. Model.* (35) p.1349–1353.
Huq F., Cutright K., Bhutta M.K.S. 2009. The analysis of scheduling rules in a flexible job shop with non-zero setup costs. *Int. J. Technol. Policy Manag.* (9) p.126.
Jack E.P., Raturi A. 2002. Sources of volume flexibility and their impact on performance. *J. Oper. Manag.* (20) p.519–548.
Jans R. 2009. Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints. *INFORMS J. Comput.* (21) p.123–136.
Jiménez J.-F. 2017. *Dynamic and hybrid architecture for the optimal reconfiguration of control systems : application to manufacturing control*.
Jimenez J.-F.F., Bekrar A., Trentesaux D., Leitão P. 2016. A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures. *Int. J. Prod. Res.* (54) p.7027–7042.
Joo B.J., Choi Y.C., Xirouchakis P. 2013. Dispatching Rule-based Algorithms for a Dynamic Flexible Flow Shop Scheduling Problem with Time-dependent Process Defect Rate and Quality Feedback. *Procedia CIRP* (7) p.163–168.
Jorapur V.S., Puranik V.S., Deshpande A.S., Sharma M. 2016. A Promising Initial Population Based Genetic Algorithm for Job Shop Scheduling Problem. *J. Softw. Eng. Appl.* (09) p.208–214.
Joseph O.A., Sridharan R. 2011. Evaluation of routing flexibility of a flexible manufacturing system using simulation modelling and analysis. *Int. J. Adv. Manuf. Technol.* (56) p.273–289.
Kacem I., Hammadi S., Borne P. 2002. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Math. Comput. Simul.* (60) p.245–276.
Kajko-Mattsson M., Forssander S., Olsson U. 2001. Corrective maintenance maturity model (CM/sup 3/): maintainer's education and training. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, 610–619.
Kanet J.J. 1981. Minimizing Variation of Flow Time in Single Machine Systems. *Manage. Sci.* (27) p.1453–1459.
Kärkkäinen M., Holmström J., Främling K., Artto K. 2003. Intelligent products - A step towards a more effective project delivery chain. *Comput. Ind.* (50) p.141–151.
Karsak E.E., Özogul C.O. 2002. An options approach to valuing expansion flexibility in flexible manufacturing system investments. *Eng. Econ.* (47) p.169–193.
Kaschel H., Bernal L.Y. 2006. Importance of flexibility in manufacturing systems. *Int. J. Comput. …* (I) p.53–60.
Keddis N., Kainz G., Zoitl A. 2015. Product-driven generation of action sequences for adaptable manufacturing systems. (28) p.1502–1508.
Kia H., Ghodsypour S.H., Davoudpour H. 2017. New scheduling rules for a dynamic flexible flow line problem with sequence-dependent setup times. *J. Ind. Eng. Int.* (13) p.297–306.
Kia H.R., Davoudpour H., Zandieh M. 2010. Scheduling a dynamic flexible flow line with sequence-dependent setup times: a simulation analysis. *Int. J. Prod. Res.* (48) p.4019–4042.
Kianfar K., Fatemi Ghomi S.M.T., Oroojlooy Jadid A. 2012. Study of stochastic sequence-dependent flexible flow shop via developing a dispatching rule and a hybrid GA. *Eng. Appl. Artif. Intell.* (25) p.494–506.
Kim D.-W.D.-W., Kim K.-H.K.-H., Jang W., Frank Chen F. 2002. Unrelated parallel machine scheduling with setup times using simulated annealing. *Robot. Comput. Integr. Manuf.* (18)

p.223–231.
Koestler A. 1967. *The ghost in the machine*.
Kohler D., Weisz J.-D., Dorothée kohler J.-D. weiz kohler, Dorothée K., Jean-Daniel weisz. 2016. Industrie 4.0 : comment caractériser cette quatrième révolution industrielle et ses enjeux ? *Ann. des Mines - Réalités Ind.* (4) p.51–56.
Koren Y. et al. 1999. Reconfigurable Manufacturing Systems. *CIRP Ann.* (48) p.527–540.
Koren Y. 2006. General RMS characteristics. comparison with dedicated and flexible systems. *Reconfigurable Manuf. Syst. Transform. Factories* p.27–45.
Kotsiantis S.B. 2007. Supervised Machine Learning: A Review of Classification Techniques. *Inform.* (31) p.249–268.
Koulamas C., Kyparisis G.J. 2008. Single-machine scheduling problems with past-sequence-dependent setup times. *Eur. J. Oper. Res.* (187) p.1045–1049.
Krishnan M., Chinnusamy T.R., Karthikeyan T. 2012. Performance study of flexible manufacturing system scheduling using dispatching rules in dynamic environment. *Procedia Eng.* (38) p.2793–2798.
Kumar D., Chen W., Simpson T.W. 2009. A market-driven approach to product family design. *Int. J. Prod. Res.* (47) p.71–104.
Kumar R.L. 1995. An options view of investments in expansion-flexible manufacturing systems. *Int. J. Prod. Econ.* (38) p.281–291.
Lamas P., Demeulemeester E. 2016. A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *J. Sched.* (19) p.409–428.
Lee K., Leung J.Y.T., Pinedo M.L. 2011. Scheduling jobs with equal processing times subject to machine eligibility constraints. *J. Sched.* (14) p.27–38.
Lei D. 2011. Scheduling stochastic job shop subject to random breakdown to minimize makespan. *Int. J. Adv. Manuf. Technol.* (55) p.1183–1192.
Leitão P. 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. *Eng. Appl. Artif. Intell.* (22) p.979–991.
Leitao P., Colombo A.W., Restivo F. 2006. A formal specification approach for holonic control systems: the ADACOR case. *Int. J. Manuf. Technol. Manag.* (8) p.37.
Leitão P., Restivo F. 2008. A holonic approach to dynamic manufacturing scheduling. *Robot. Comput. Integr. Manuf.* (24) p.625–634.
Leitão P., Rodrigues N., Barbosa J., Turrin C., Pagani A. 2015. Intelligent products: The grace experience. *Control Eng. Pract.* (42) p.95–105.
Li K., Leung J.Y.T., Cheng B.Y. 2014. An agent-based intelligent algorithm for uniform machine scheduling to minimize total completion time. *Appl. Soft Comput. J.* (25) p.277–284.
Li L., Qiao F. 2008. ACO-based scheduling for a single batch processing machine in semiconductor manufacturing. *4th IEEE Conf. Autom. Sci. Eng. CASE 2008* p.85–90.
Li Q., Milne R.J. 2014. A production scheduling problem with sequence-dependent changeover costs. *Int. J. Prod. Res.* (52) p.4093–4102.
Li S. 2017. Parallel batch scheduling with inclusive processing set restrictions and non-identical capacities to minimize makespan. *Eur. J. Oper. Res.* (260) p.12–20.
Liou C.-D., Hsieh Y.-C., Chen Y.-Y. 2013. A new encoding scheme-based hybrid algorithm for minimising two-machine flow-shop group scheduling problem. *Int. J. Syst. Sci.* (44) p.77–93.
Liu B., Fan Y., Liu Y. 2015. A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem. *Comput. Ind. Eng.* (87) p.193–201.
Logendran R., McDonell B., Smucker B. 2007. Scheduling unrelated parallel machines with sequence-dependent setups. *Comput. Oper. Res.* (34) p.3420–3438.
Lopez P., Roubellat F. 2010. *Production Scheduling*.
Lou P., Liu Q., Zhou Z., Wang H., Sun S.X. 2012. Multi-agent-based proactive–reactive

scheduling for a job shop. *Int. J. Adv. Manuf. Technol.* (59) p.311–324.
Lu Y.-Y., Wang J.-J., Wang J.-B. 2014. Single machine group scheduling with decreasing time-dependent processing times subject to release dates. *Appl. Math. Comput.* (234) p.286–292.
Maione G., Naso D. 2003. A soft computing approach for task contracting in multi-agent manufacturing control. *Comput. Ind.* (52) p.199–219.
Mani V., Chang P.-C., Chen S.-H. 2011. Single-machine scheduling with past-sequence-dependent setup times and learning effects: a parametric analysis. *Int. J. Syst. Sci.* (42) p.2097–2102.
McFarlane D., Giannikas V., Wong A.C.Y., Harrison M. 2013. Product intelligence in industrial control. *Annu. Rev. Control* (37) p.69–88.
McFarlane D., Sarma S., Chirn J.L., Wong C.., Ashton K. 2003. Auto ID systems and intelligent manufacturing control. *Eng. Appl. Artif. Intell.* (16) p.365–376.
McFarlane D., Sarma S., Chirn J.L., Wong C.Y., Ashton K. 2002. *the Intelligent Product in Manufacturing Control and Management*.
Mehrabi M.G., Ulsoy A.G., Koren Y., Heytler P. 2002. Trends and perspectives in flexible and reconfigurable manufacturing systems. *J. Intell. Manuf.* (13) p.135–146.
Melki A., Hammadi S., Sallez Y., Berger T., Tahon C. 2010. Advanced approach for the public transportation regulation system based on cybercars. *RAIRO - Oper. Res.* (44) p.85–105.
Meyer G.G., Främling K., Holmström J. 2009. Intelligent Products. *Comput. Ind.* (60) p.137–148.
Min Hee Kim, Kim Y.D. 1994. Simulation-based real-time scheduling in a flexible manufacturing system. *J. Manuf. Syst.* (13) p.85–93.
Mjirda A. 2015. *Recherche à voisinage variable pour des problèmes de routage avec ou sans gestion de stock*.
Möhring R.H., Skutella M., Stork F. 2004. Scheduling with AND/OR Precedence Constraints. *SIAM J. Comput.* (33) p.393–415.
Le Moigne J.-L. 1994. *La théorie du système général*.
Monkman S.K., Morrice D.J., Bard J.F. 2008. A production scheduling heuristic for an electronics manufacturer with sequence-dependent setup costs. *Eur. J. Oper. Res.* (187) p.1100–1114.
Monostori L. 2015. Cyber-physical production systems: roots from manufacturing science and technology. - *Autom.* (63) p.766–776.
Moore K.E., Güngör A., Gupta S.M. 2001. Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships. *Eur. J. Oper. Res.* (135) p.428–449.
Morariu O., Raileanu S., Morariu C., Borangiu T. 2014. Multi-agent system for heterarchical product-driven manufacturing. *Proc. 2014 IEEE Int. Conf. Autom. Qual. Testing, Robot. AQTR 2014*.
Mosier C.T., Elvers D.A., Kelly D. 1984. Analysis of group technology scheduling heuristics. *Int. J. Prod. Res.* (22) p.857–875.
Mousakhani M. 2013. Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. *Int. J. Prod. Res.* (51) p.3476–3487.
Mozdgir A., Fatemi Ghomi S.M.T., Jolai F., Navaei J. 2013. Two-stage assembly flow-shop scheduling problem with non-identical assembly machines considering setup times. *Int. J. Prod. Res.* (51) p.3625–3642.
Nguyen S., Mei Y., Zhang M. 2017. Genetic programming for production scheduling: a survey with a unified framework. *Complex Intell. Syst.* (3) p.41–66.
Nie L., Gao L., Li P., Zhang L. 2011. Application of gene expression programming on dynamic job shop scheduling problem. *Proc. 2011 15th Int. Conf. Comput. Support. Coop. Work Des.* p.291–295.
Ochoa G., Qu R., Burke E.K. 2009. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. *Proc. 11th Annu. Conf. Genet. Evol. Comput. - GECCO '09*

p.341–348.
Omar M.K., Teo S.C. 2006. Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach. *Appl. Math. Comput.* (181) p.1008–1017.
Ouelhadj D., Petrovic S. 2009. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* (12) p.417–431.
Pach C. 2013. *ORCA : architecture hybride pour le contrôle de la myopie dans le cadre du pilotage des systèmes flexibles de production*.
Pach C., Berger T., Bonte T., Trentesaux D. 2014. ORCA-FMS: A dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind.* (65) p.706–720.
Pannequin R., Morel G., Thomas A. 2009. The performance of product-driven manufacturing co. *Comput. Ind.* (60) p.195–203.
Panwalkar S.S., Iskander W. 1977. A Survey of Scheduling Rules. *Oper. Res.* (25) p.45–61.
Park H.S., Tran N.H. 2011. An autonomous manufacturing system for adapting to disturbances. *Int. J. Adv. Manuf. Technol.* (56) p.1159–1165.
Park S., Kim J.H., Fox G. 2014. Effective real-time scheduling algorithm for cyber physical systems society. *Futur. Gener. Comput. Syst.* (32) p.253–259.
Parunak H.V.D., Baker A.D., Clark S.J. 1998. *AARIA agent architecture: from manufacturing requirements to agent-based system design*.
Paulussen T.O. et al. 2006. Agent-Based Patient Scheduling in Hospitals. In *Multiagent Engineering*, 255–275.
Pérez M.P., Bedia A.M.S., López Fernández M.C. 2016. A review of manufacturing flexibility: Systematising the concept. *Int. J. Prod. Res.* (54) p.3133–3148.
Pétin J.F., Gouyon D., Morel G. 2007. Supervisory synthesis for product-driven automation and its application to a flexible assembly cell. *Control Eng. Pract.* (15) p.595–614.
Phanden R.K., Jain A. 2015. Assessment of makespan performance for flexible process plans in job shop scheduling. (28) p.1948–1953.
Pickardt C.W., Branke J. 2012. Setup-oriented dispatching rules - A survey. *Int. J. Prod. Res.* (50) p.5823–5842.
Pillay N., Qu R. 2018. *Hyper-Heuristics: Theory and Applications*.
Pinedo M.L. 2016. *Scheduling*.
Priore P., De La Fuente D., Puente J., Parreño J. 2006. A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. *Eng. Appl. Artif. Intell.* (19) p.247–255.
Pujawan I.N. 2004. Schedule nervousness in a manufacturing system: a case study. *Prod. Plan. Control* (15) p.515–524.
Pujo P., Broissin N., Ounnar F. 2009. PROSIS: An isoarchic structure for HMS control. *Eng. Appl. Artif. Intell.* (22) p.1034–1045.
Pujo P., Kieffer J.-P. 2002. Concepts fondamentaux du pilotage des systèmes de production.
Qu S., Chu T., Wang J., Leckie J., Jian W. 2015. A centralized reinforcement learning approach for proactive scheduling in manufacturing. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–8.
Quintanilla F.G., Cardin O., Castagna P. 2013. Evolution of a flexible manufacturing system: From communicating to autonomous product. In *Studies in Computational Intelligence*, 167–180.
Reinhart G., Engelhardt P., Ostgathe M. 2011. Modular Configuration of an Rfid-Based Hybrid Control Architecture. *Ind. Eng.* (1) p.31–39.
Ribas I., Companys R. 2015. Efficient heuristic algorithms for the blocking flow shop scheduling problem with total flow time minimization. *Comput. Ind. Eng.* (87) p.30–39.
Rossi A. 2014. Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. *Int. J. Prod. Econ.* (153)

p.253–267.
Rudtsch V., Gausemeier J., Gesing J., Mittag T., Peter S. 2014. Pattern-based Business Model Development for Cyber-Physical Production Systems. *Procedia CIRP* (25) p.313–319.
Rüßmann M. et al. 2015. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. *Bost. Consult.* (62) p.40–41.
Sabuncuoglu I., Kizilisik O.B. 2003. Reactive scheduling in a dynamic and stochastic FMS environment. *Int. J. Prod. Res.* (41) p.4211–4231.
Sallez Y. 2012a. *Produit 'actif' tout au long de son cycle de vie*.
Sallez Y. 2012b. The augmentation concept. In T. Borangiu, A. Thomas, & D. Trentesaux (eds) *Studies in Computational Intelligence*.Studies in Computational Intelligence, 35–48.
Sallez Y., Berger T., Deneux D., Trentesaux D. 2010. The lifecycle of active and intelligent products. *Int. J. Comput. Integr. Manuf.* (23) p.905–924.
Sallez Y., Berger T., Trentesaux D. 2009. A stigmergic approach for dynamic routing of active products in FMS. *Comput. Ind.* (60) p.204–216.
Sarin S.C., Chen C.S. 1987. The machine loading and tool allocation problem in a flexible manufacturing system. *Int. J. Prod. Res.* (25) p.1081–1094.
Sarin S.C., Varadarajan A., Wang L. 2011. A survey of dispatching rules for operational control in wafer fabrication. *Prod. Plan. Control* (22) p.4–24.
Sawhney R. 2006. Interplay between uncertainty and flexibility across the value-chain: Towards a transformation model of manufacturing flexibility. *J. Oper. Manag.* (24) p.476–493.
Sayama H. 2015. *Introduction to the modeling and analysis of complex systems*.
Schaller J., Valente J.M.S. 2013. An evaluation of heuristics for scheduling a non-delay permutation flow shop with family setups to minimize total earliness and tardiness. *J. Oper. Res. Soc.* (64) p.805–816.
Schoute M. 2011. The relationship between product diversity, usage of advanced manufacturing technologies and activity-based costing adoption. *Br. Account. Rev.* (43) p.120–134.
Seidgar H., Kiani M., Abedi M., Fazlollahtabar H. 2014. An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flow shop problem. *Int. J. Prod. Res.* (52) p.1240–1256.
Seok Shin K., Park J.-O., Keun Kim Y. 2011. Multi-objective FMS process planning with various flexibilities using a symbiotic evolutionary algorithm. *Comput. Oper. Res.* (38) p.702–712.
Sethi A.K., Sethi S.P. 1990. Flexibility in manufacturing: A survey. *Int. J. Flex. Manuf. Syst.* (2) p.289–328.
Shahzad A., Mebarki N. 2016. Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. (5) p.3.
Simão J.M., Tacla C.A., Stadzisz P.C. 2009. Holonic control metamodel. *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans* (39) p.1126–1139.
Smith S.F. 1995. Reactive Scheduling Systems. In *Intelligent Scheduling Systems*, 155–192.
Soroush H.M. 2013. On the scheduling with past-sequence-dependent setup times and learning effects on a single machine. *Int. J. Adv. Manuf. Technol.* (68) p.2483–2487.
Stecke K.E., Solberg J.J. 1981. Loading and control policies for a flexible manufacturing system. *Int. J. Prod. Res.* (19) p.481–490.
Subramaniam V., Lee G.K., Ramesh T., Hong G.S., Wong Y.S. 2000. Machine selection rules in a dynamic job shop. *Int. J. Adv. Manuf. Technol.* (16) p.902–908.
Sullivan S. et al. 2009. Downregulation of sodium transporters and NHERF proteins in IBD patients and mouse colitis models: Potential contributors to IBD-associated diarrhea. *Inflamm. Bowel Dis.* (15) p.261–274.
Sun D.-H., He W., Zheng L.-J., Liao X.-Y. 2014. Scheduling flexible job shop problem subject to machine breakdown with game theory. *Int. J. Prod. Res.* (52) p.3858–3876.
Swiercz A. 2017. Hyper-Heuristics and Metaheuristics for Selected Bio-Inspired Combinatorial Optimization Problems. In *Heuristics and Hyper-Heuristics - Principles and Applications*,

Tan Z., Zhang A. 2010. A note on hierarchical scheduling on two uniform machines. *J. Comb. Optim.* (20) p.85–95.

Tay J.C., Ho N.B. 2008. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Comput. Ind. Eng.* (54) p.453–473.

Tehrani Niknejad H., Sugimura N., Iwamura K., Tanimizu Y. 2010. Multi agent architecture for dynamic incremental process planning in the flexible manufacturing system. *J. Intell. Manuf.* (21) p.487–499.

Thomas A., Trentesaux D., Valckenaers P. 2012. Intelligent distributed production control. *J. Intell. Manuf.* (23) p.2507–2512.

Trentesaux D. 2009. Distributed control of production systems. *Eng. Appl. Artif. Intell.* (22) p.971–978.

Trentesaux D. 2013. Are intelligent manufacturing systems sustainable ? To cite this version : Are Intelligent Manufacturing Systems Sustainable ?

Trentesaux D., Borne P.P., Lille E.C. De, Erschler P.J., Ladet P.P. 2002. *Pilotage hétérarchique des systèmes de production*.

Trentesaux D., Thomas A. 2012. Product-driven control: A state of the art and future trends. *IFAC Proc. Vol.* (14) p.716–721.

Tseng M.M., Piller F.T. 2003. Cust. Centric Enterp. In *The Customer Centric Enterprise*, 3–16.

Tu Q., Vonderembse M.A., Ragu-Nathan T.S., Ragu-Nathan B. 2004. Measuring Modularity-Based Manufacturing Practices and Their Impact on Mass Customization Capability: A Customer-Driven Perspective. *Decis. Sci.* (35) p.147–168.

Upton D.M. 1997. Process Range in Manufacturing: An Empirical Study of Flexibility. *Manage. Sci.* (43) p.1079–1092.

Valckenaers P., Van Brussel H. 2005. Holonic manufacturing execution systems. *CIRP Ann. - Manuf. Technol.* (54) p.427–432.

Vázquez Rodríguez J.A., Petrovic S., Salhi A. 2007. A Combined Meta-Heuristic with Hyper-Heuristic Approach to the Scheduling of the Hybrid Flow Shop with Sequence Dependent Setup Times and Uniform Machines. *Proc. 3rd Multidiscip. Int. Conf. Sched. Theory Appl.* p.506–513.

Ventä O. 2007. Intelligent products and systems: Technology theme-final report. *VTT Publ.* (635) p.3–304.

Verstraete P. et al. 2008. Towards robust and efficient planning execution. *Eng. Appl. Artif. Intell.* (21) p.304–314.

Vieira G.E., Herrmann J.W., Lin E. 2003. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *J. Sched.* (6) p.39–62.

Vinet L., Zhedanov A. 2011. A 'missing' family of classical orthogonal polynomials. *J. Phys. A Math. Theor.* (44) p.085201.

Vrba P., Macůrek F., Mařík V.A. 2006. Using Radio Frequency Identification in Agent-Based Control Systems for Industrial Applications. *Inf. Control Probl. Manuf. 2006* (21) p.437–442.

Wang B., Wang H. 2018. Multiobjective Order Acceptance and Scheduling on Unrelated Parallel Machines with Machine Eligibility Constraints. *Math. Probl. Eng.* (2018) p.1–12.

Wang J.-B., Xia Z.-Q. 2005. Flow-shop scheduling with a learning effect. *J. Oper. Res. Soc.* (56) p.1325–1330.

Wang Yuanhang, Deng C., Wu J., Wang Yingchun, Xiong Y. 2014. A corrective maintenance scheme for engineering equipment. *Eng. Fail. Anal.* (36) p.269–283.

Weng M.X., Lu J., Ren H. 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *Int. J. Prod. Econ.* (70) p.215–226.

Whitesides G.M., Ismagilov R.F. 1999. Complexity in chemistry. *Science (80-.).* (284) p.89–92.

Wiener N. 1961. *Cybernetics; or, Control and communication in the animal and the machine.*

Wilbrecht J.K., Prescott W.B. 1969. The Influence of Setup Time on Job Shop Performance.

Manage. Sci. (16) p.B274–B280.

Windt K., Becker T., Jeken O., Gelessus A. 2010. A classification pattern for autonomous control methods in logistics. *Logist. Res.* (2) p.109–120.

Wong C.Y., McFarlane D., Zaharudin A.A., Agarwal V. 2002. The intelligent product driven supply chain. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 393–398.

Woo S.H., Choi J.Y., Kwak C., Kim C.O. 2009. An active product state tracking architecture in logistics sensor networks. *Comput. Ind.* (60) p.149–160.

Wu C.-C., Lee W.-C. 2008. Single-machine group-scheduling problems with deteriorating setup times and job-processing times. *Int. J. Prod. Econ.* (115) p.128–133.

Wu L.H., Chen X., Chen X.D., Chen Q.X. 2009. The research on proactive-reactive scheduling framework based on real-time manufacturing information. In *Materials Science Forum*, 789–794.

Xiong H., Fan H., Jiang G., Li G. 2017. A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. *Eur. J. Oper. Res.* (257) p.13–24.

Yamamoto H., Marui E. 2001. Decentralized Autonomous FMS Control by Hypothetical Reasoning Including Discrete Simulator. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 571–581.

Zambrano G., Pach C., Aissani N., Berger T., Trentesaux D. 2011. An approach for temporal myopia reduction in Heterarchical Control Architectures. *Proc. - ISIE 2011 2011 IEEE Int. Symp. Ind. Electron.* p.1767–1772.

Zambrano Rey G. 2014. Reducing myopic behavior in FMS control: A semi-heterarchical simulation-optimization approach. *Simul. Model. Pract. Theory* (46) p.53–75.

Zandieh M., Adibi M.A. 2010. Dynamic job shop scheduling using variable neighbourhood search. *Int. J. Prod. Res.* (48) p.2449–2458.

Zbib N., Pach C., Sallez Y., Trentesaux D. 2012. Heterarchical production control in manufacturing systems using the potential fields concept. *J. Intell. Manuf.* (23) p.1649–1670.

Zbib Nadine. 2010. *Allocation et routage dynamique dans un FMS basés sur le concept de champ de potentiel*.

Zhang D., Liu C. 2014. Predictive coding with context as a model of image saliency map. In F. Sun, D. Hu, & H. Liu (eds) *Advances in Intelligent Systems and Computing*.Advances in Intelligent Systems and Computing, 15–23.

Zhang G., Xing K., Cao F. 2018. Discrete differential evolution algorithm for distributed blocking flowshop scheduling with makespan criterion. *Eng. Appl. Artif. Intell.* (76) p.96–107.

Zhang J., Yang J., Zhou Y. 2016. Robust scheduling for multi-objective flexible job-shop problems with flexible workdays. *Eng. Optim.* (48) p.1973–1989.

Zhang S., Wong T.N. 2017. Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid MAS/ACO approach. *Int. J. Prod. Res.* (55) p.3173–3196.

Zhou B. hai, Li X., Fung R.Y.K. 2013. Dynamic scheduling of photolithography process based on Kohonen neural network. *J. Intell. Manuf.* (26) p.73–85.

Zuehlke D. 2010. SmartFactory—Towards a factory-of-things. *Annu. Rev. Control* (34) p.129–138.

ANNEXE A : DISTRIBUTIONS DES PRODUITS

1. DISTRIBUTIONS HF1

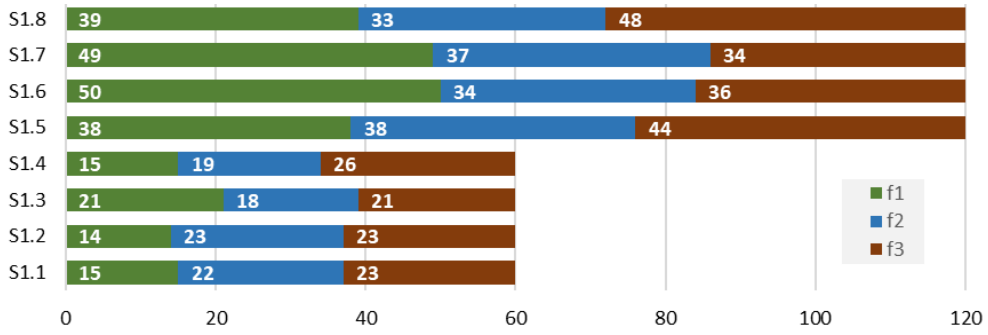


Figure 93 Distributions des familles de produits pour les 8 scripts pour HF1

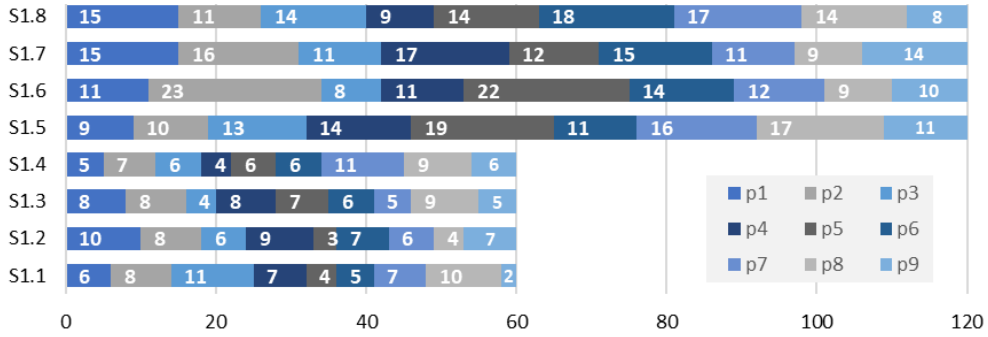


Figure 94 Distributions des priorités des produits pour les 8 scripts pour HF1

2. DISTRIBUTIONS HF2

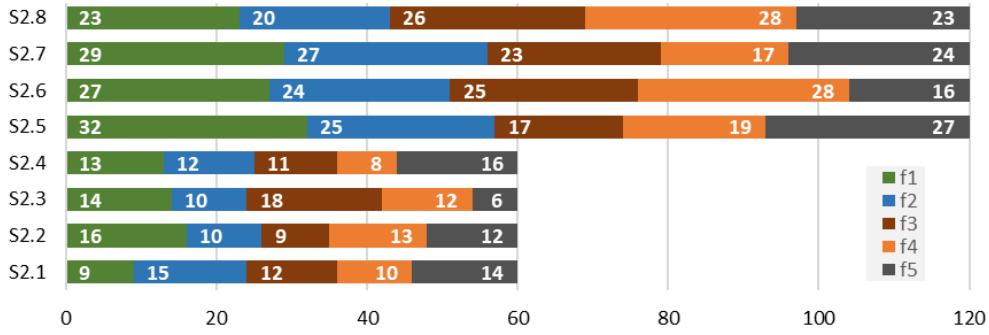


Figure 95 Distributions des familles de produits pour les 8 scripts pour FJ1

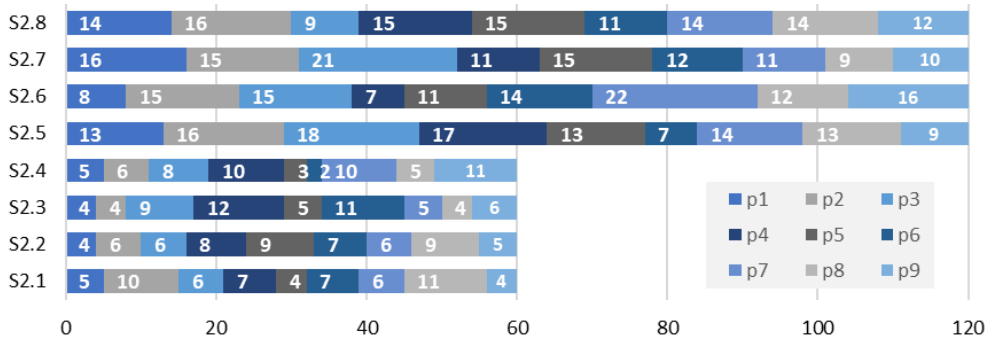


Figure 96 Distributions des priorités des produits pour les 8 scripts pour FJ2

3. DISTRIBUTIONS FJ1

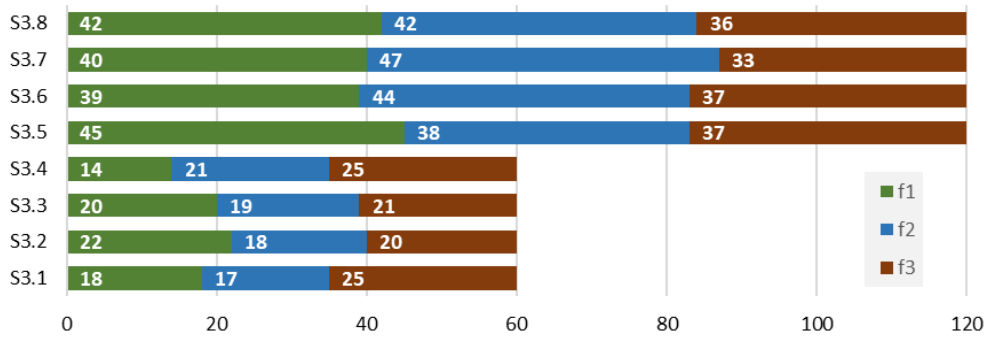


Figure 97 Distributions des familles de produits pour les 8 scripts pour FJ1

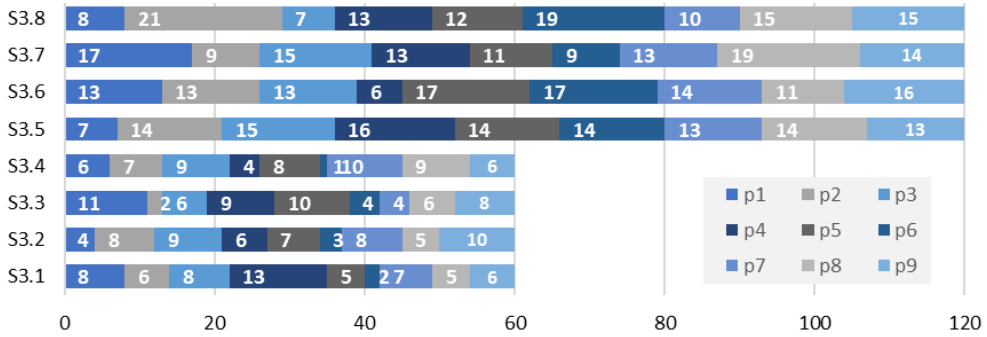


Figure 98 Distributions des priorités des produits pour les 8 scripts pour FJ1

4. DISTRIBUTIONS FJ2

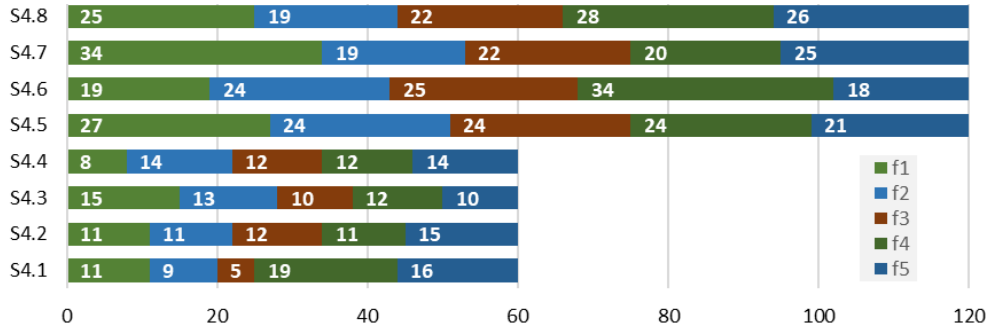


Figure 99 Distributions des familles de produits pour les 8 scripts pour FJ2

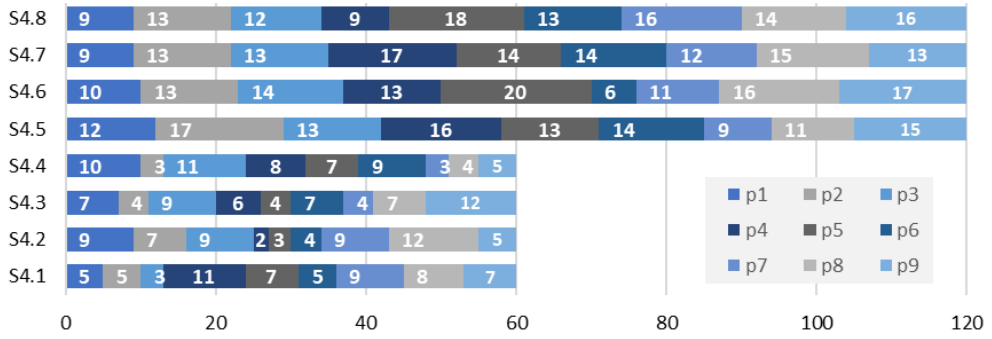


Figure 100 Distributions des priorités des produits pour les 8 scripts pour FJ2

ANNEXE B : DETAILS DES SIMULATIONS

1. RESULTATS HF1

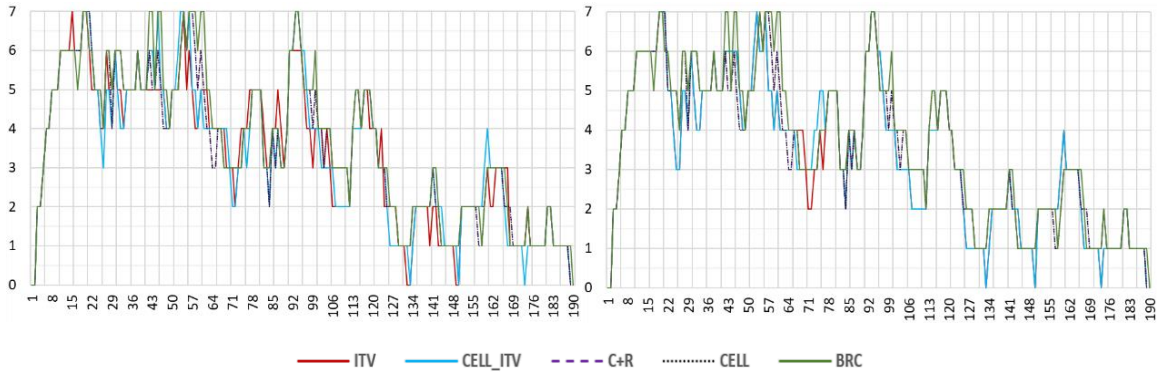


Figure 101 Evolutions du nombre de produits dans le FMS scenarios S1.3a & S1.3b

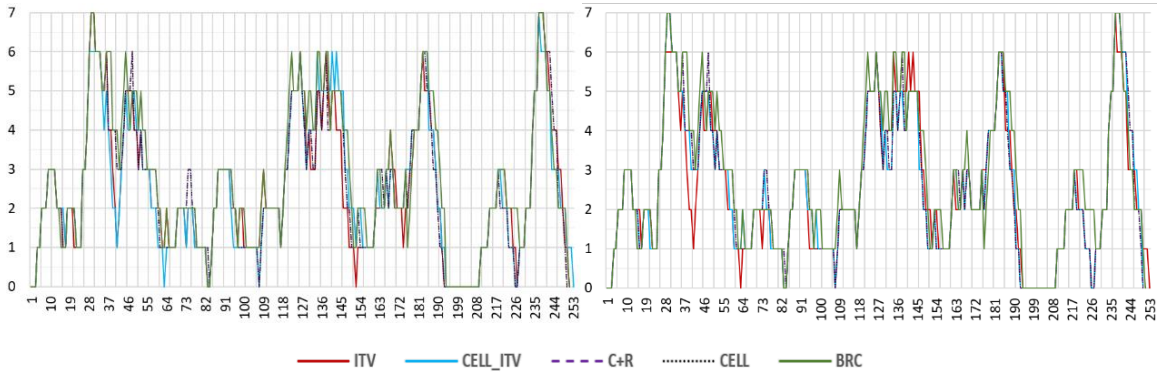


Figure 102 Evolutions du nombre de produits dans le FMS scenarios S1.4a & S1.4b

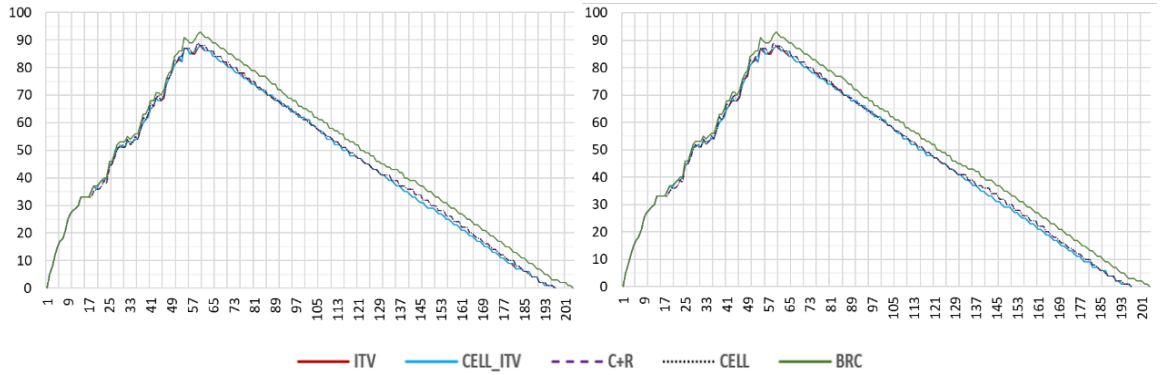


Figure 103 Evolutions du nombre de produits dans le FMS scenarios S1.5a & S1.5b

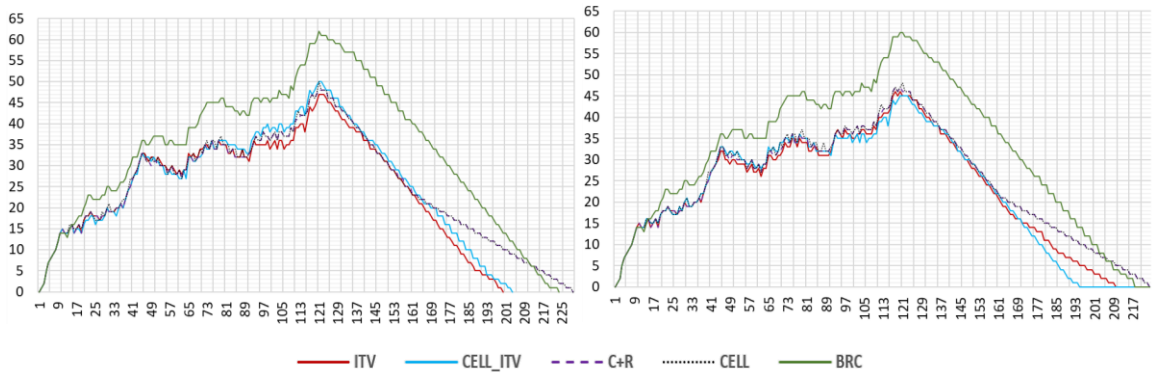


Figure 104 Evolutions du nombre de produits dans le FMS scenarios S1.6a & S1.6b

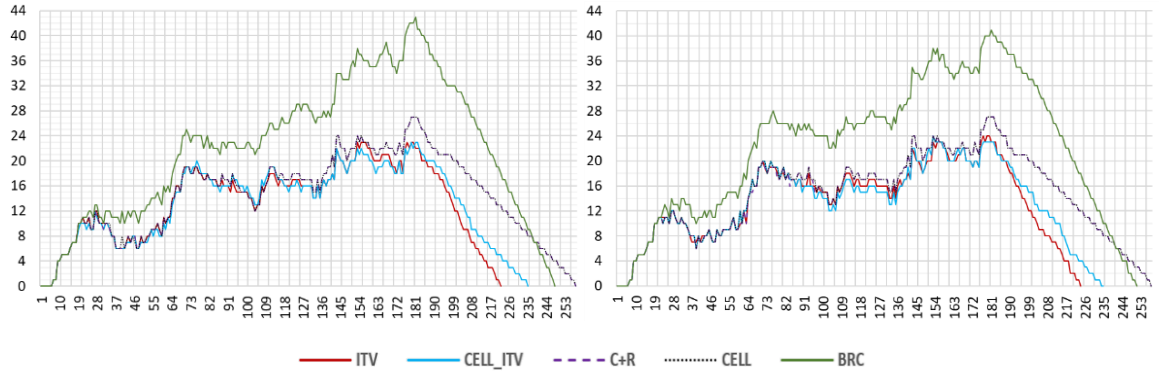


Figure 105 Evolutions du nombre de produits dans le FMS scenarios S1.7a & S1.7b

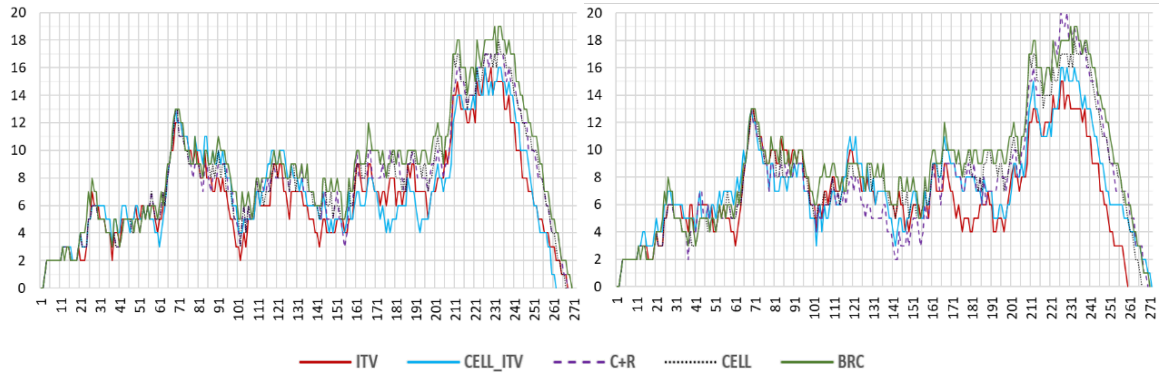


Figure 106 Evolutions du nombre de produits dans le FMS scenarios S1.8a & S1.8b

2. RESULTATS HF2

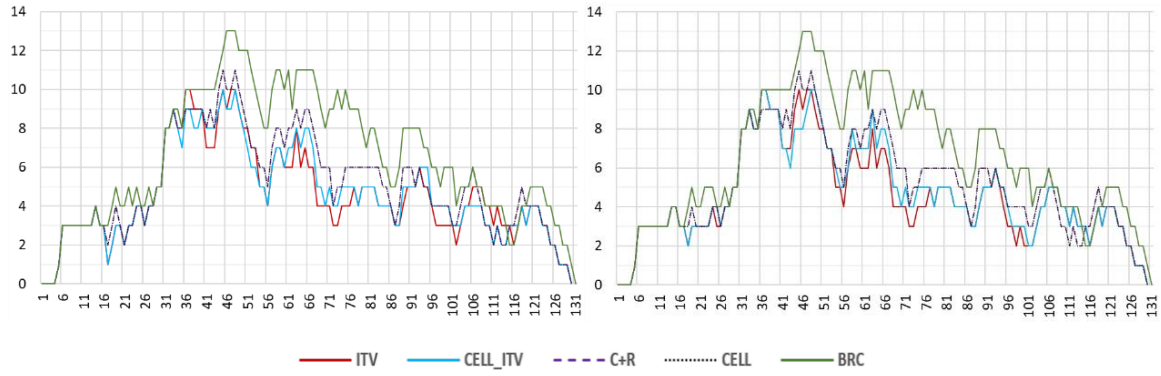


Figure 107 Evolutions du nombre de produits dans le FMS scenarios S2.2a & S2.2b

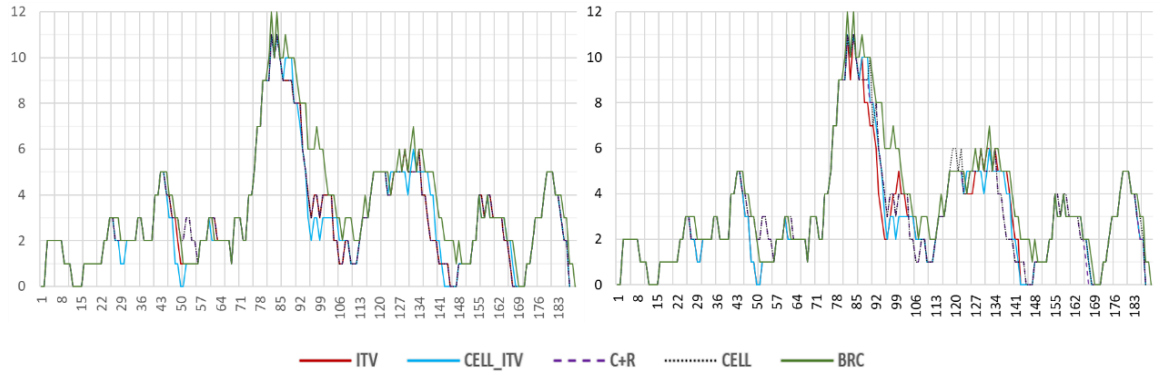


Figure 108 Evolutions du nombre de produits dans le FMS scenarios S2.3a & S2.3b

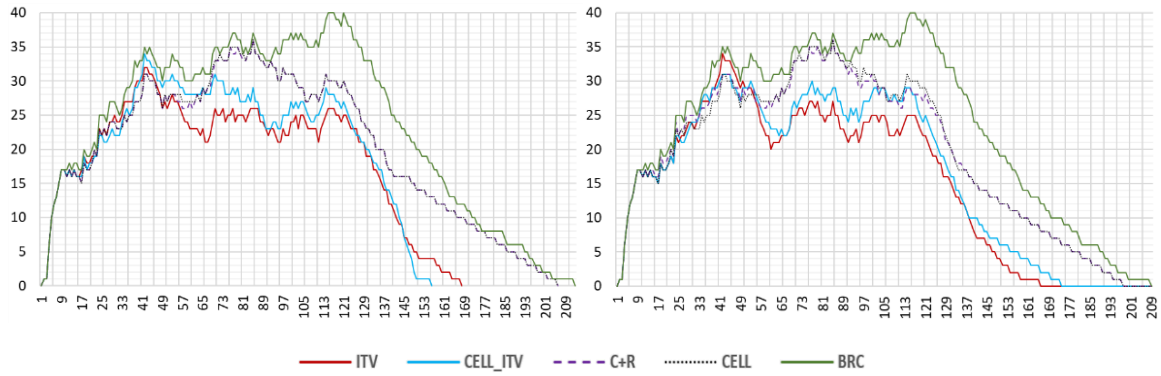


Figure 109 Evolutions du nombre de produits dans le FMS scenarios S2.6a & S2.6b

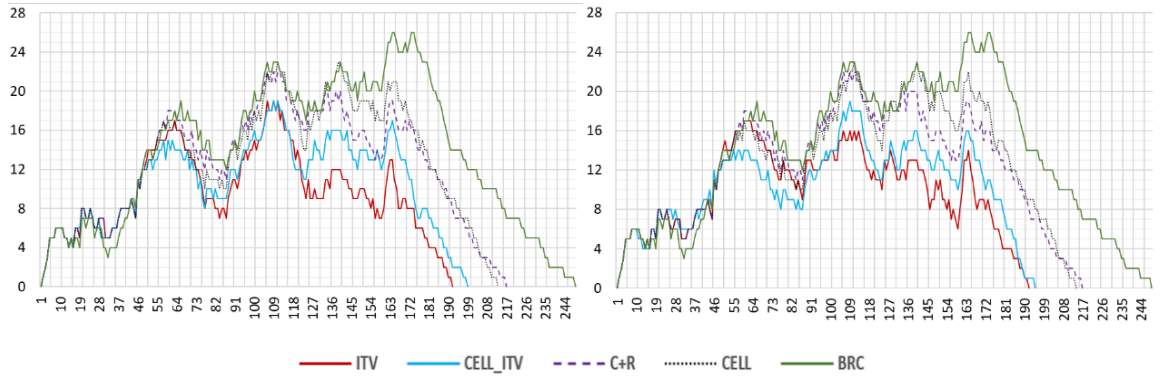


Figure 110 Evolutions du nombre de produits dans le FMS scenarios S2.7a & S2.7b

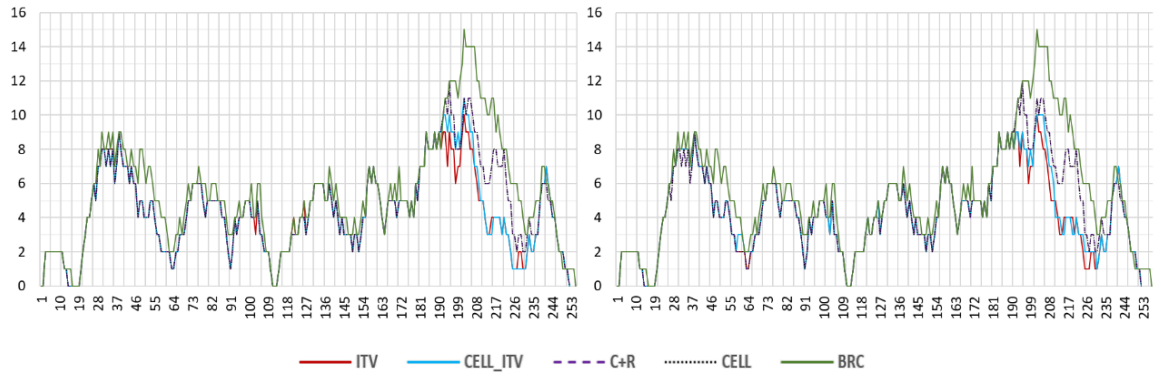


Figure 111 Evolutions du nombre de produits dans le FMS scenarios S2.8a & S2.8b

3. RESULTATS FJ1

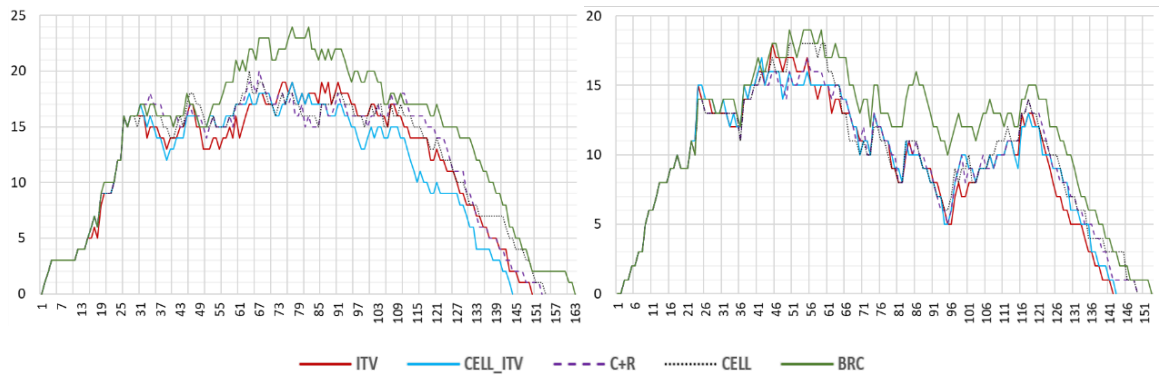


Figure 112 Evolutions du nombre de produits dans le FMS scenarios S3.2a & S3.2b

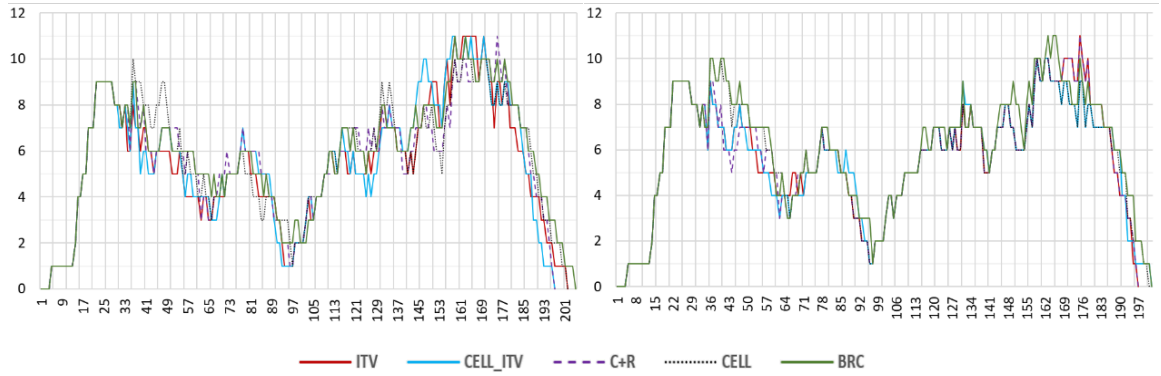


Figure 113 Evolutions du nombre de produits dans le FMS scenarios S3.3a & S3.3b

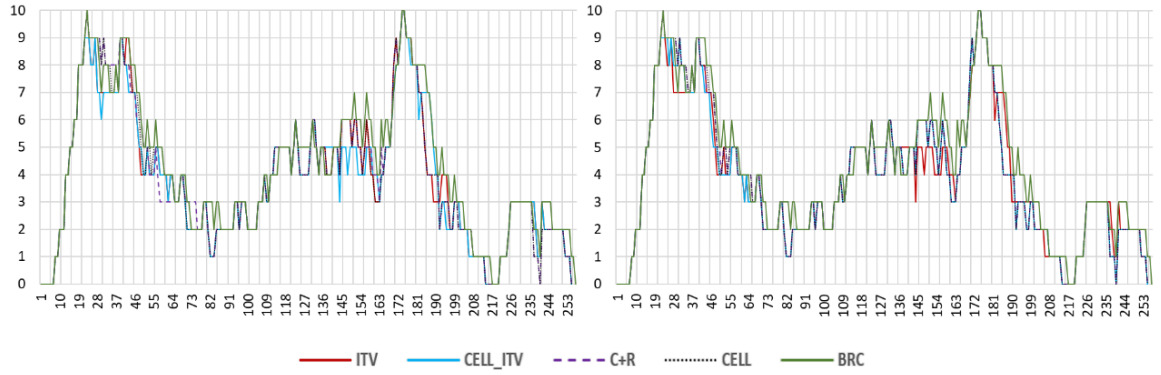


Figure 114 Evolutions du nombre de produits dans le FMS scenarios S3.4a & S3.4b

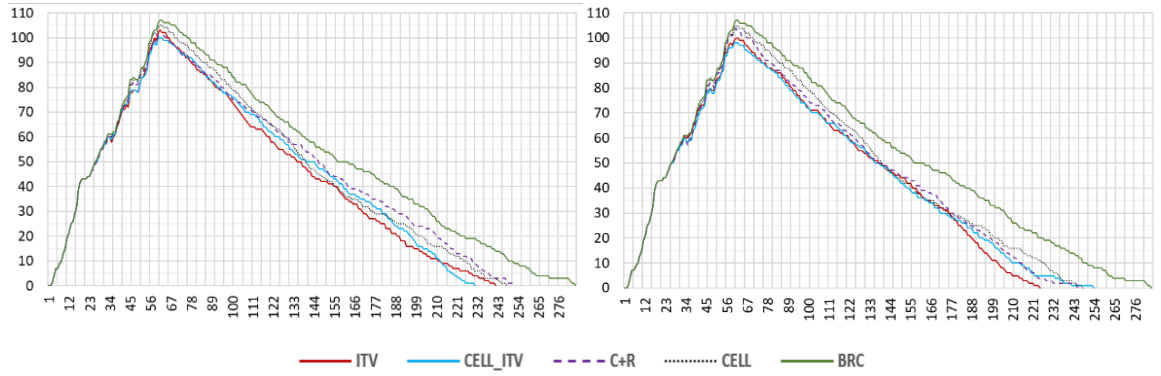


Figure 115 Evolutions du nombre de produits dans le FMS scenarios S3.5a & S3.5b

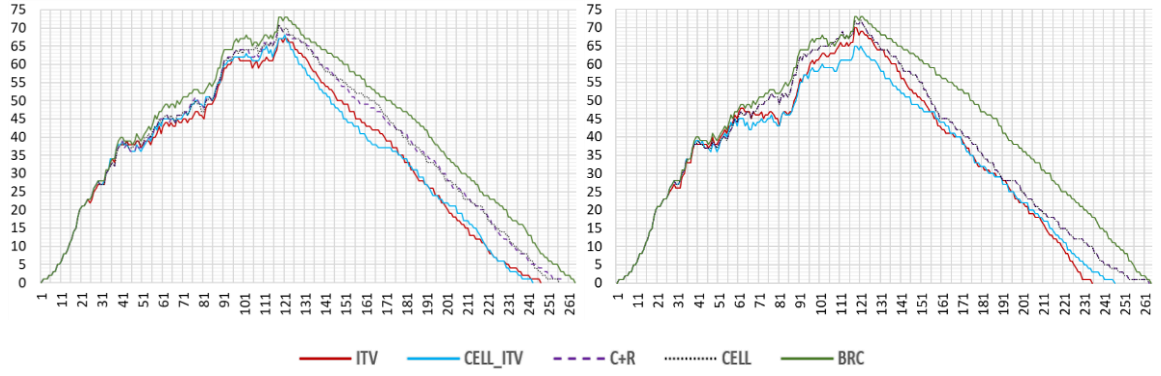


Figure 116 Evolutions du nombre de produits dans le FMS scenarios S3.6a & S3.6b

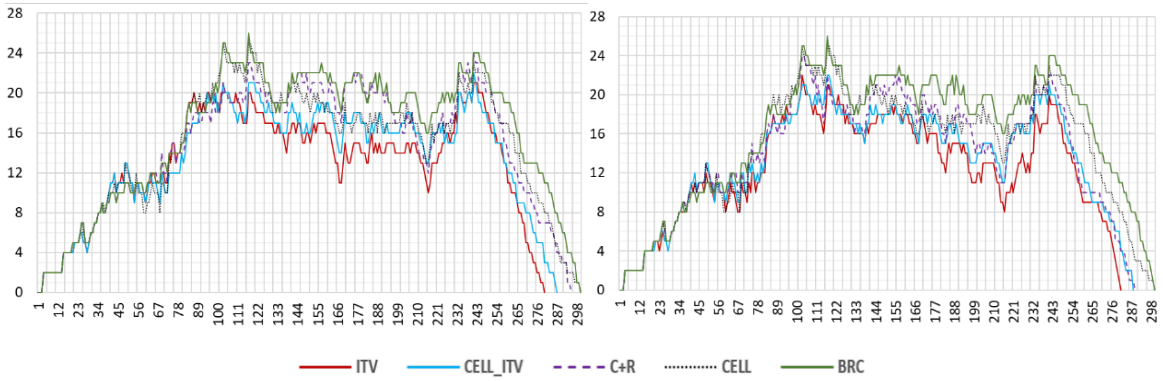


Figure 117 Evolutions du nombre de produits dans le FMS scenarios S3.8a & S3.8b

4. RESULTATS FJ2

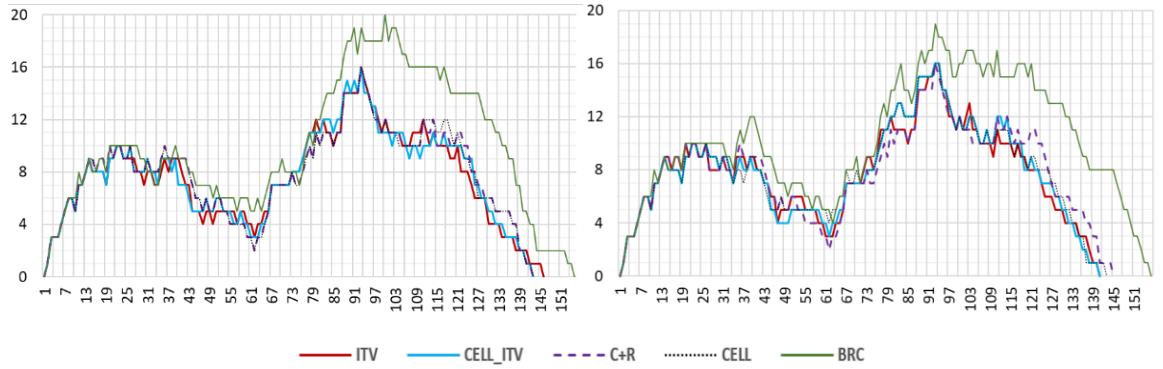


Figure 118 Evolutions du nombre de produits dans le FMS scenarios S4.2a & S4.2b

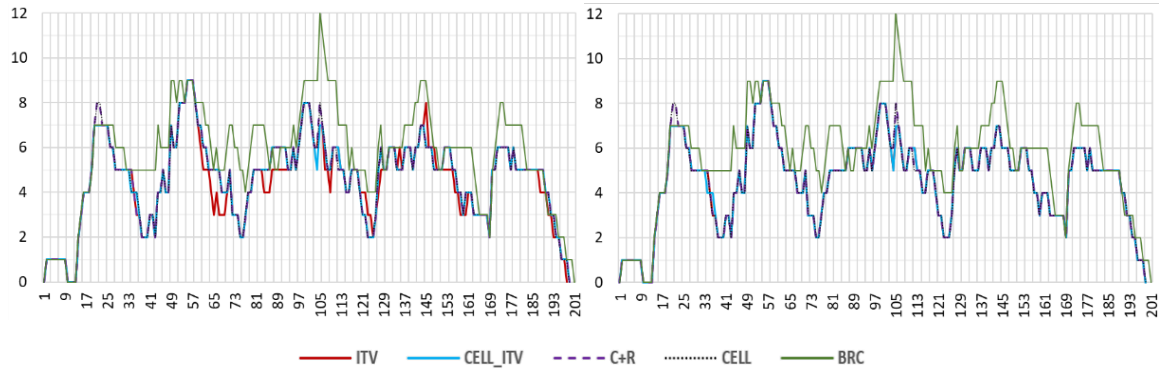


Figure 119 Evolutions du nombre de produits dans le FMS scenarios S4.3a & S4.3b

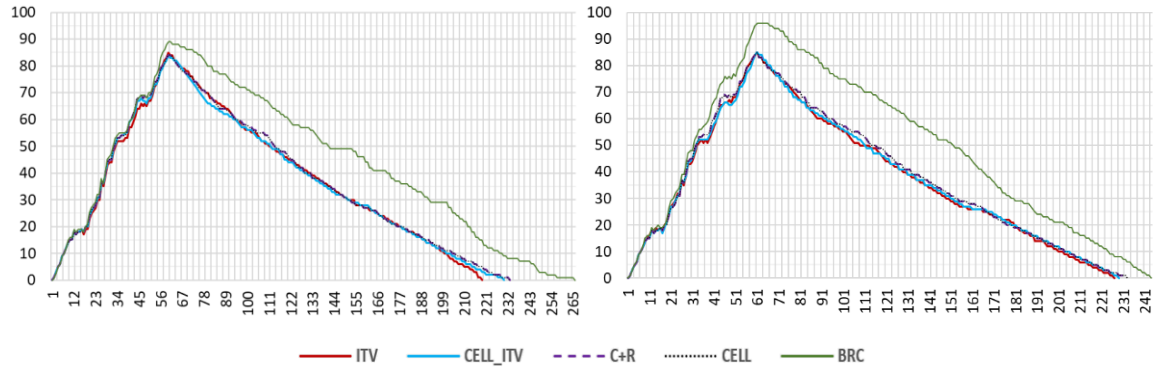


Figure 120 Evolutions du nombre de produits dans le FMS scenarios S4.5a & S4.5b

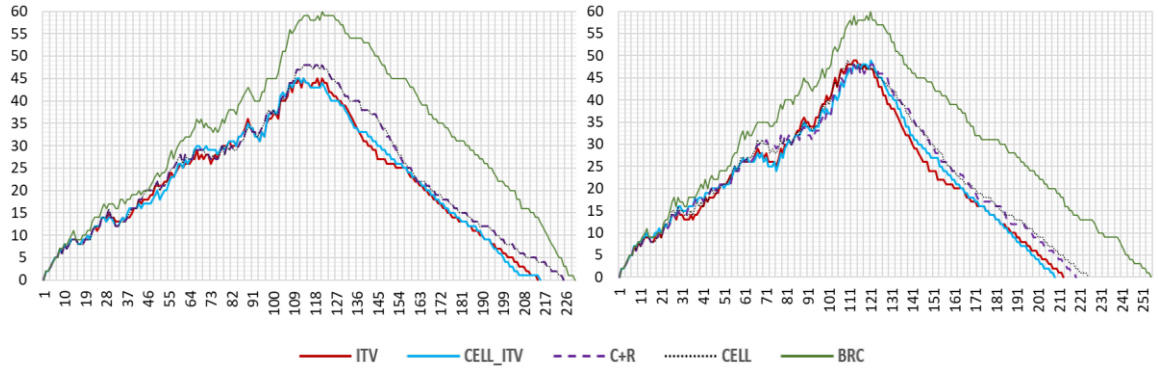


Figure 121 Evolutions du nombre de produits dans le FMS scenarios S4.6a & S4.6b

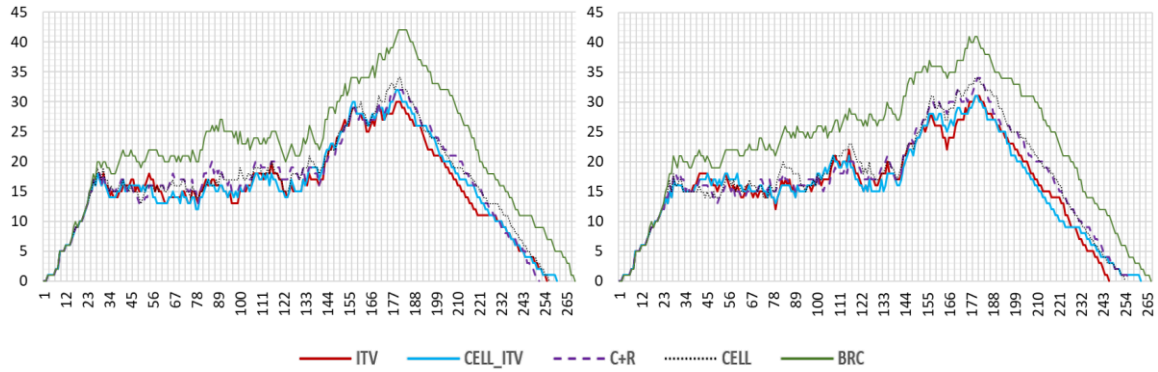


Figure 122 Evolutions du nombre de produits dans le FMS scenarios S4.7a & S4.7b

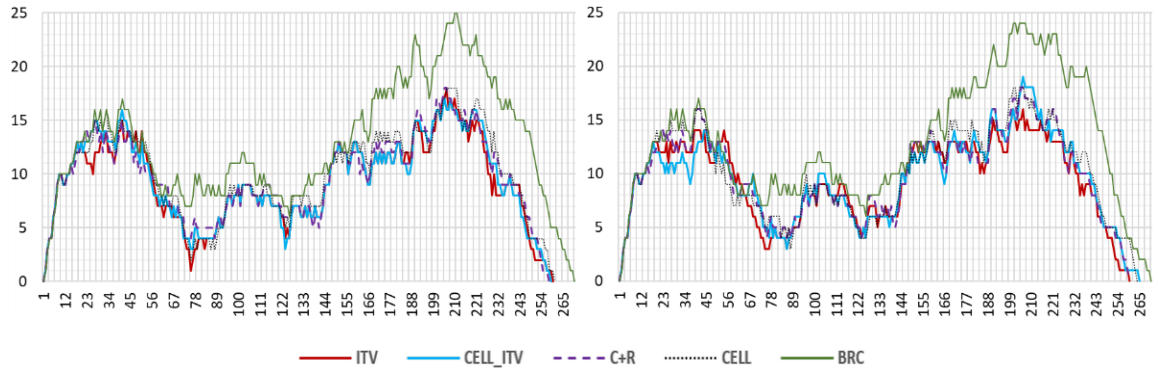


Figure 123 Evolutions du nombre de produits dans le FMS scenarios S4.8a & S4.8b