Contribution à l'ordonnancement post-pronostic de plates-formes hétérogènes et distribuées : approches discrète et continue.

Nathalie Herr

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UFC UNIVERSITÉ DE FRANCHE-COMTÉ

Contribution à l'ordonnancement post-pronostic de plates-formes hétérogènes et distribuées : approches discrète et continue

Décision post-pronostic

Nathalie Herr

SPIM

Thèse de Doctorat

UFC

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

présentée à L'U.F.R. DES SCIENCES ET TECHNIQUES DE L'UNIVERSITÉ DE FRANCHE-COMTÉ pour obtenir LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE FRANCHE-COMTÉ
Spécialité : Informatique

Contribution à l'ordonnancement post-pronostic de plates-formes hétérogènes et distribuées : approches discrète et continue

par **Nathalie Herr**
Soutenue le 19 Novembre 2015 devant la Commission d'Examen :

Rapporteurs	Olivier Beaumont	Directeur de Recherche, Institut National de Recherche en Informatique et en Automatique (INRIA), LaBRI, Bordeaux
	Chengbin Chu	Professeur des Universités, Ecole CentraleSupélec, LGI, Paris
Membres du jury	Stéphane Chrétien	Maître de Conférences HDR, National Physical Laboratory, Londres, Angleterre
	Alix Munier	Professeur des Universités, Université Pierre et Marie Curie (UPMC), LIP6, Paris
	Jean-Marc Nicod	Professeur des Universités, Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM), FEMTO-ST, Besançon – Directeur
	Kévin Subrin	Docteur Ingénieur R&D, ALTRAN Research, Lyon
	Denis Trystram	Professeur des Universités, Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliquées (ENSIMAG), LIG, Grenoble
	Christophe Varnier	Maître de Conférences HDR, Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM), FEMTO-ST, Besançon – Co-Directeur

Remerciements

Au moment où on croit en avoir fini avec la rédaction, vient la partie qui n'est, de loin, pas la plus simple à écrire...

Je tiens tout d'abord à remercier mes deux directeurs de thèse, Jean-Marc Nicod et Christophe Varnier. Merci de m'avoir fait confiance et de m'avoir donné l'opportunité d'effectuer ce travail de recherche. J'aurais difficilement pu souhaiter de meilleures conditions de travail : vous avez été d'un soutien et surtout d'une disponibilité sans faille tout au long de ces trois ans. Travailler avec vous a été formateur et très motivant.

Un grand merci à Stéphane Chrétien, qui m'a énormément aidée pour toute la partie mathématique de ce travail. Tu as été très pédagogue et surtout très patient avec moi, alors que je n'étais jamais satisfaite des résultats. Trouver un terrain d'entente entre les maths, Matlab et les contraintes de l'application n'a pas été facile, mais on y est finalement arrivé !

Je remercie ensuite les membres de mon jury pour avoir accepté d'évaluer mes travaux et d'avoir fait le déplacement pour assister à ma soutenance. Merci à M. Olivier Beaumont et au Professeur Chengbin Chu d'avoir pris le temps de lire mon manuscrit et de rapporter mes travaux. La lecture de vos rapports m'a reboostée pour la préparation de ma soutenance.

J'ai effectué mes travaux de thèse au sein du département AS2M [1] de l'institut FEMTO-ST [2] de Besançon, au sein duquel règne une ambiance de travail très agréable et très conviviale. Merci à tous mes collègues pour leur accueil et les nombreux éclats de rire à la cafèt' ! Je tiens aussi à remercier mes collègues des équipes d'enseignement de l'ENSMM [3] et de l'UFC [4], qui m'ont permis d'assurer mes heures d'enseignement dans d'excellentes conditions.

Un merci particulier va à mes collègues de bureau, qui ont participé tous les jours à une ambiance studieuse mais détendue et amicale : Baptiste, Kamran, Bassem. Un grand merci à Amélie et Élodie, sans qui la période de stress qui clôt la thèse ne se serait pas passée aussi bien ! Merci aussi à Elvia, qui a partagé, même de loin, les périodes de doute aussi bien que celles de joie. À ton tour maintenant !

Je tiens aussi à remercier Marc Barth et David Damand, qui m'ont ouvert les portes de la recherche académique et m'ont encouragée à me lancer dans l'aventure de la thèse.

Je voudrais finalement remercier ma famille et mes amis d'Alsace pour avoir cru en moi et m'avoir toujours encouragée et soutenue tout au long de ces trois dernières années. Merci d'avoir fait le déplacement pour ma soutenance, votre présence m'a fait énormément plaisir. Comme quoi, Besançon n'est pas si loin ! ;)

Nathalie Herr.

1. Automatique et Systèmes Micro-Mécatroniques
2. Franche-Comté Électronique, Mécanique, Thermique et Optique – Sciences et Technologies
3. École Nationale Supérieure de Mécanique et des Microtechniques
4. Université de Franche-Comté

Table des matières

Introduction

Quel que soit le type de production ou le champs d'application, l'ordonnancement de la production est un problème complexe lié à de nombreux enjeux (économiques, écologiques ou sociétaux par exemple). Cela est confirmé par le grand nombre d'études menées sur ce sujet. De nombreuses hypothèses différentes pouvant être prises en considération et combinées de différentes manières, le nombre de possibilités est très grand. Les contributions dans le domaine de l'ordonnancement partagent toutefois un but global commun, qui est l'utilisation optimale des ressources.

Les travaux menés au cours de cette thèse concernent l'ordonnancement de la production d'une plate-forme composée de plusieurs machines indépendantes et de même type, utilisées en parallèle pour fournir un service global. La production totale correspond à chaque instant à la somme des contributions de chaque machine et doit atteindre au minimum un certain niveau de service, basé sur un besoin. La plate-forme de machines considérée peut être vue comme un environnement distribué, dans lequel les machines contribuent toutes de manière indépendante à une tâche commune. Toutes les machines ne sont pas supposées être utilisées en même temps à chaque instant. Le débit total demandé peut en effet dans certains cas être atteint en n'utilisant qu'un sous-ensemble des machines. Contrairement à une hypothèse courante dans la littérature traitant des problèmes d'ordonnancement, certaines machines peuvent aussi ne plus être disponibles au bout d'un certain temps d'utilisation. Au cours de leur utilisation, les machines de production sont en effet soumises à une certaine usure, qui implique des arrêts dus à des pannes ou à des opérations de maintenance nécessaires. Garantir la disponibilité d'un système de production devient alors un enjeu important. Les machines étant supposées indépendantes dans le cas des applications visées, la panne de l'une d'entre elles n'entraîne pas nécessairement l'arrêt de toute la plate-forme. La maintenance est toutefois requise sur le long terme et peut générer des coûts non négligeables. Ces coûts peuvent être minimisés par le biais d'une optimisation de la stratégie de maintenance mise en œuvre. Une des solutions proposées dans la littérature consiste à grouper les différentes opérations de maintenance. Cela permet de réduire les coûts engendrés à la fois par les besoins en matériel et en ressources humaines et par les périodes d'arrêt de production. Une seule période de maintenance pour toute la plate-forme est ainsi autorisée.

L'objectif est alors de maximiser la durée de vie de la plate-forme en utilisant au maximum le potentiel fourni par l'ensemble des machines. En d'autres termes, il s'agit d'optimiser l'ordonnancement de la production pour satisfaire la demande le plus longtemps possible. L'originalité principale de la contribution réside dans la proposition de modifier les conditions opératoires des machines. Ceci repose sur l'hypothèse proposée dans la littérature établissant qu'utiliser

une machine avec des performances dégradées par rapport à un fonctionnement nominal permet d'allonger sa durée de vie. Chaque machine est ainsi supposée pouvoir fournir différents niveaux de performances correspondant à différentes conditions opératoires. Afin d'aller encore plus loin dans la maximisation de la durée de vie de chaque machine, nous proposons de prendre en compte une durée de vie basée sur un état de santé réel et non la valeur limite traditionnellement utilisée. Cette valeur, généralement fixée par les constructeurs, est par définition pessimiste, car déterminée avec une marge de sécurité non négligeable. La détermination de l'état de santé réel des machines peut être faite au sein d'un processus PHM (pour Prognostics and Health Management), dans lequel un suivi de l'évolution de l'état de santé des machines est rendu possible par l'étude de certains signaux caractéristiques. Basée sur cet état de santé, une étape de pronostic permet de déterminer la durée de vie résiduelle de chaque machine, ou *RUL* (pour Remaining Useful Life), en fonction de son utilisation passée et future. Au sein du processus PHM, il s'agit alors de définir un ordonnancement de la production, basé sur les données du pronostic et s'adaptant à l'état de santé réel des machines, avec pour objectif la maximisation de l'horizon de production. Ces travaux s'inscrivent dans la partie décisionnelle du processus PHM, dans laquelle les contributions sont jusqu'à présent principalement focalisées sur l'optimisation de l'ordonnancement de la maintenance.

Organisation du manuscrit

Le manuscrit est organisé en six chapitres. Les travaux de recherche sont tout d'abord positionnés dans la littérature traitant du *PHM* et de l'ordonnancement de la production. Suite à une définition du problème d'optimisation traité, plusieurs méthodes de résolution sont ensuite proposées pour le problème d'optimisation considéré.

Le **chapitre 1** pose le cadre général de l'étude. L'émergence du *PHM* est tout d'abord expliquée par le biais d'un survol de l'évolution des politiques de maintenance au cours du temps. Le développement de méthodes de pronostic a permis de passer d'une stratégie purement corrective, déclenchée par les défaillances, à des stratégies plus intelligentes anticipant l'évolution de l'état de santé des machines pour permettre de réagir avant les défaillances et de prévenir la majeure partie des dégradations de performances associées. Les différentes étapes du processus *PHM* sont ensuite décrites, avec une focalisation sur la partie décisionnelle. La décision post-pronostic est présentée suivant deux axes différents traduisant ses applications possibles, à savoir l'optimisation de la maintenance et l'adaptation de la production ou de missions en fonction de l'état de santé des équipements. La présentation des travaux traitant de l'optimisation de la production ou de missions sur la base des données du pronostic permet de mettre en évidence le concept de reconfiguration, utilisé pour adapter l'utilisation des machines à leur état de santé. Ce principe est utilisé dans le cadre de cette thèse pour optimiser l'ordonnancement d'un ensemble de machines.

Le **chapitre 2** fournit une définition de l'ordonnancement de la production, qui est l'objet de l'étude présentée dans ce manuscrit. Les problèmes d'ordonnancement types et les principales méthodes de résolution classiquement utilisées dans ce domaine sont ensuite présentés. La notion de reconfiguration introduite dans le chapitre précédent est ensuite développée dans le contexte de l'ordonnancement de la production. Plusieurs types de reconfiguration ayant été proposés dans la littérature sont détaillés.

Le **chapitre 3** définit le contexte applicatif des travaux, par le biais de la présentation des différentes hypothèses prises en compte pour les développements proposés dans les chapitres suivants. Le problème d'optimisation général traité est ensuite formulé. La seconde partie de ce chapitre détaille les différentes modélisations développées pour définir des profils de fonctionnement pour chaque machine. Chaque modélisation permet de déterminer à la fois les débits disponibles et l'évolution de la durée de vie résiduelle de chacun d'entre eux au cours de l'utilisation des machines. Deux modèles différents correspondant à deux types de machines sont proposés. Le premier permet de modéliser le comportement à l'usure de machines pouvant fournir un nombre discret de niveaux de performances. Ce premier modèle sert de base aux développements proposés dans le chapitre 4. Le second modèle s'applique à des machines dont le débit peut varier de façon continue entre une valeur minimale et une valeur maximale. Il est pris en compte pour les résolutions du problème d'optimisation proposées dans les chapitres 5 et 6.

Le **chapitre 4** présente l'étude du problème d'optimisation dans le cas où les machines considérées peuvent fournir un nombre discret de performances. Des propriétés mathématiques sont tout d'abord proposées, rassemblant la définition d'une borne supérieure pour l'horizon de production, ainsi que l'étude de la complexité de plusieurs variantes du problème d'optimisation. Cette étude permet de cartographier le problème du point de vue de son niveau de difficulté. Dans le cas le plus dur, une formulation basée sur un programme linéaire permet de caractériser la solution optimale. Plusieurs méthodes de résolution sous-optimales sont enfin proposées sous la forme d'heuristiques. L'efficacité des différentes approches est mesurée sur la base de simulations.

Deux approches différentes de résolution sont ensuite proposées pour les machines présentant une variation continue des performances. La première approche est développée dans le **chapitre 5**. Elle définit une construction des ordonnancements par morceaux, sur la base de résolutions successives de sous-problèmes. Pour chacun des sous-problèmes, une approche optimale basée sur une programmation linéaire permet de déterminer un ordonnancement optimal des machines considérant l'objectif de maximisation de l'horizon de production et l'état de santé courant des machines. Le changement de profil de fonctionnement au cours de l'utilisation des machines n'étant pas géré par la programmation linéaire, une stratégie de résolution globale est proposée pour utiliser au maximum le potentiel de la plate-forme considérée.

Les différentes approches de résolution présentées dans les chapitres 4 et 5 permettent de trouver des solutions optimales en temps polynomial uniquement pour des problèmes pour lesquels le nombre de machines et/ou l'horizon de production restent limités. L'utilisation de méthodes de résolution sous-optimales est nécessaire pour traiter des problèmes de grande taille mais limite les performances atteintes. Une autre approche utilisant un paradigme totalement différent est alors finalement proposée dans le **chapitre 6**. L'utilisation de chaque machine est déterminée par le biais d'une optimisation convexe. L'utilisation de ce type d'optimisation mathématique, dans lequel une fonction convexe est à minimiser, apporte certaines propriétés au problème traité, notamment de bonnes propriétés de convergence. La programmation convexe permet de plus de résoudre en temps polynomial des problèmes de grande taille, mettant en jeux un grand nombre de machines et de grands horizons de production. Une formulation mathématique du problème d'optimisation est tout d'abord introduite pour permettre sa résolution avec des méthodes de programmation convexe. Différentes méthodes de résolution basées sur la programmation convexe sont ensuite proposées et leur efficacité est à nouveau mesurées sur la base de résultats de simulation.

Chapitre 1

Décision post-pronostic : définition, contexte et étude bibliographique

Sommaire

Le maintien des performances des équipements dans le temps et la garantie de leur disponibilité sont des enjeux qui ont pris de plus en plus d'importance au cours de l'industrialisation et de la course à la rentabilité. La disponibilité correspond au temps pendant lequel un système de production peut remplir sa mission de production [28]. Elle dépend de la fiabilité du système, mais aussi de l'efficacité de la stratégie de maintenance mise en place [28, 47]. La stratégie de maintenance correspond à l'ensemble des décisions qui conduisent à choisir une politique de maintenance adaptée à la situation. Les politiques de maintenance ont évolué au cours des dernières années, passant d'une maintenance purement corrective à une maintenance de plus en plus intelligente, qui prend en compte l'état de santé des machines pour lancer les opérations de réparation ou de changement de pièces avant l'occurrence d'une panne. Les dernières évolutions ont été permises par différentes techniques de suivi de l'état de santé et de prédiction de la durée de vie résiduelle des machines. Ces procédés sont intégrés dans un processus nommé *PHM* (pour Prognostics and Health Management), dont le but est de maintenir dans le temps les performances opérationnelles des équipements de production et d'améliorer leur utilisation, tout en minimisant leurs coûts de maintenance [19].

Dans ce premier chapitre, l'évolution des politiques de maintenance est tout d'abord montrée pour expliquer l'émergence du *PHM*. Les différentes étapes du *PHM* sont ensuite détaillées. La décision post-pronostic prise en compte dans ce manuscrit est enfin définie et positionnée au sein du *PHM*. Différentes applications sont détaillées et la possibilité de modifier les performances des machines afin d'optimiser leur utilisation est introduite.

1.1 Évolution des politiques de maintenance et émergence du *PHM*

Une présentation de l'évolution des politiques de maintenance utilisées en industrie est proposée pour introduire le domaine du Prognostics and Health Management (*PHM*) dans lequel s'inscrivent les travaux de recherche proposés dans ce manuscrit. Le processus *PHM* est ensuite défini par le biais de la description des différentes étapes qui le composent.

1.1.1 Les différentes politiques de maintenance

Selon la norme *NF EN 13306 X 60-319*, la maintenance est « l'ensemble des activités destinées à rétablir un bien dans un état ou dans des conditions de sûreté de fonctionnement, pour accomplir une fonction requise. Ces activités sont une combinaison d'activités techniques, administratives et de management ». La maintenance englobe donc l'ensemble des activités effectuées dans le but de maintenir un système dans un état dans lequel il peut effectuer sa fonction [23]. En d'autres termes, il s'agit de garantir au système une certaine disponibilité, qui est définie par la norme *NF EN 13306 X 60-319* comme « l'aptitude d'un bien à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné ou durant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs est assurée ».

Les politiques de maintenance ont évolué au cours des dernières décennies pour passer d'une maintenance déclenchée par l'occurrence d'une défaillance à une maintenance mettant en œuvre des stratégies qui anticipent l'évolution de l'état des équipements pour réagir avant les défaillances. Cette évolution est illustrée sur la figure 1.1. Les différentes politiques de maintenance

sont présentées dans la suite dans leur ordre d'apparition.

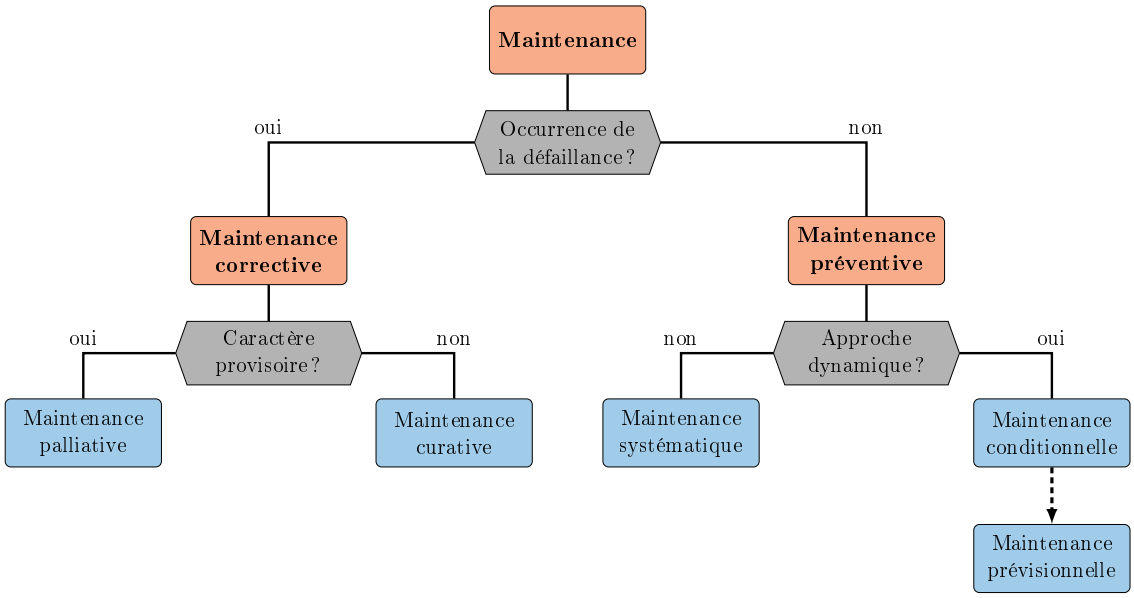


Figure 1.1 – Les différentes politiques de maintenance selon la norme *NF EN 13306 X 60-319*

1.1.1.1 Maintenance corrective

La maintenance corrective est la plus ancienne forme de maintenance [48]. Selon la norme *NF EN 13306 X 60-319*, elle est effectuée « après la défaillance du bien ou la dégradation de sa fonction ». Les opérations de maintenance ne sont ainsi effectuées qu'au moment où les machines de production tombent en panne et ne peuvent plus remplir leur fonction de façon satisfaisante. Deux types de maintenance corrective peuvent être différenciés : la maintenance palliative englobe les opérations revêtant un caractère provisoire, alors que la maintenance curative est basée sur des interventions plus complètes ayant un effet à plus long terme.

Cette politique peut être justifiée dans certains contextes de production pour lesquels la non disponibilité de machines n'affecte pas la performance et la qualité de la production de façon significative ou ne génère pas de coûts trop importants en dehors des coûts de réparation. Les pannes surviennent toutefois généralement à des moments inappropriés, par exemple au cours d'une opération de production, et la maintenance corrective peut alors engendrer des arrêts de production non prévus et donc des coûts liés à la perte de production. La maintenance corrective a ainsi plusieurs inconvénients, tels que des délais de production excessifs et non parfaitement maîtrisés, un manque de visibilité quant aux besoins en pièces de rechange qui peut être la cause d'un stock de ces pièces sur-dimensionné, ou encore des pertes financières dues aux pertes de production associées aux périodes de maintenance. Une stratégie uniquement basée sur du correctif ne permet de plus pas d'organiser au mieux les différentes opérations de maintenance à faire, ce qui peut avoir pour conséquence un accroissement des risques d'accident.

Afin de pallier ces différents inconvénients, d'autres politiques de maintenance ont été développées. Ces dernières tendent à programmer les opérations de maintenance au bon moment pour éviter que les dégradations des machines ne mènent jusqu'à la panne. Les pannes ne pouvant être totalement évitées, la maintenance corrective est toutefois toujours utile, sinon nécessaire.

1.1.1.2 Maintenance préventive

La maintenance préventive est, selon la norme *NF EN 13306 X 60-319*, « exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinés à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien ». Cette politique proactive, qui est la plus répandue en industrie, consiste à prévoir des actions de maintenance à temps pour éviter que les dégradations des machines ne s'aggravent jusqu'à entraîner des défaillances et nécessiter des interventions lourdes [48]. Elle peut aussi être définie comme l'ensemble des tâches permettant d'éviter les causes des défaillances potentielles [51]. Cette politique inclut en effet toutes les opérations de maintenance effectuées avant la panne et se base sur des inspections périodiques et, si nécessaire, sur des réparations mineures afin de réduire le risque de pannes non anticipées.

Elle présente l'avantage de réduire le nombre de pannes et donc les temps de non production et d'étendre la durée de fonctionnement des équipements, tout en diminuant les coûts de maintenance et de réparation et en améliorant la sécurité des opérateurs. Le terme de maintenance préventive rassemble deux politiques de maintenance basées sur des seuils de déclenchement et une troisième politique prévisionnelle.

Maintenance systématique La maintenance systématique est une « maintenance préventive exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage mais sans contrôle préalable de l'état du bien » (norme *NF EN 13306 X 60-319*). Cette politique base le lancement des opérations de maintenance sur un ordonnancement périodique, respectant un cycle défini au préalable. Ce fonctionnement suit l'hypothèse que le taux de défaillance est proportionnel au temps de fonctionnement et qu'un examen périodique des machines suffit pour réduire statistiquement le nombre de pannes. Cela permet de minimiser les coûts de maintenance, tout en évitant les urgences.

Deux dérives peuvent toutefois être observées avec ce type de maintenance. La première survient lorsque la fréquence de maintenance est trop rapide, causant un coût trop important en regard du besoin. A l'inverse, lorsque le temps entre deux maintenances successives est trop long, des pannes ne peuvent être évitées, causant cette fois-ci des coûts de réparation et des coûts dus à un arrêt de la production.

Maintenance conditionnelle La maintenance conditionnelle est une « maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent » (norme *NF EN 13306 X 60-319*). Les machines étant placées sous surveillance, les deux premières phases d'une maintenance conditionnelle efficace sont l'acquisition de données et leur traitement [48]. A l'inverse des politiques de maintenance corrective et de maintenance systématique, une telle maintenance vise à éviter à la fois les défaillances et les opérations de maintenance non nécessaires en tirant parti de la connaissance de l'état de santé des machines [24], déduit de l'observation de paramètres supposés significatifs. Les opérations de maintenance sont alors lancées uniquement lorsqu'un comportement anormal est avéré [54]. Selon Lebold [64], plusieurs facteurs ont accéléré l'utilisation de la maintenance conditionnelle. On peut citer le besoin de réduire les coûts de maintenance et de logistique et la volonté d'améliorer la disponibilité des machines et d'éviter les pannes.

Maintenance prévisionnelle D'après la norme *NF EN 13306 X 60-319*, la maintenance prévisionnelle est une « maintenance conditionnelle exécutée en suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs de la dégradation du bien ». C'est ainsi une politique de maintenance dynamique, qui utilise la connaissance du taux de dégradation et du temps de fonctionnement avant défaillance [50]. L'utilisation de telles informations permet de faire des choix plus éclairés et plus adaptés à la situation réelle en faisant plus que réagir à des passages de seuils ou des alertes de diagnostic [64]. La maintenance prévisionnelle permet ainsi de réduire le nombre de pannes et par extension les périodes d'immobilisation des équipements et les coûts associés, de rendre la production plus fiable et plus efficace et d'améliorer la sécurité des opérateurs.

Une des technologies clés ayant permis le passage de la maintenance systématique à la maintenance prévisionnelle est le pronostic [69], qui est un processus visant à déterminer les états futurs et le temps de fonctionnement avant défaillance restant du système surveillé. Le pronostic est l'étape centrale d'un processus plus large de gestion de l'état de santé des équipements appelé « Prognostics and Health Management » (*PHM*) et présenté ci-après.

1.1.2 Le processus *PHM*

Le *PHM* est un processus composé d'étapes de détection, de diagnostic et de pronostic menant à une étape de décision. Il vise ne vise pas seulement à comprendre les défaillances une fois qu'elles se sont manifestées, mais aussi à anticiper leur apparition pour définir des actions permettant de les éviter [45]. L'utilisation du *PHM* tend de façon générale à maintenir les performances des équipements au cours du temps et à améliorer leur utilisation, tout en minimisant leurs coûts de maintenance [19] et les risques d'indisponibilités non prévues [61].

Asmai et al. [1] ont décrit un outil de pronostic comme une succession de quatre modules : acquisition de données, identification des défaillances, analyse des données de pronostic et décision basée sur les résultats de cette analyse. Ces différents modules peuvent être identifiés dans l'architecture standardisée proposée par Lebold et al. [64] pour la maintenance conditionnelle. Cette architecture, nommée OSA/CBM (pour Open System Architecture for Condition-Based Maintenance), est constituée de sept couches fonctionnelles qui constituent les étapes clés du processus *PHM*. Ces différentes étapes, détaillées dans la suite, peuvent êtres classées dans trois groupes complémentaires : l'observation, l'analyse et l'action. Une illustration de la structure d'un processus *PHM* intégrant ces différentes notations est proposée sur la figure 1.2.

1.1.2.1 Observation

Acquisition des données Cette première étape concerne la mise en place de capteurs sur le système dont l'état de santé doit être surveillé et la récupération de données. Ces données doivent permettre de déduire des informations sur l'état de santé du système et sur les phénomènes de dégradation qu'il subit. Suivant l'application, le contexte de production ou le type de données devant être récupérées, les capteurs sélectionnés peuvent être très différents. On peut citer en exemple des capteurs acoustiques, thermiques, de pression, d'humidité [54], ou encore des capteurs de vibration ou de propriété de fluides tels que l'huile.

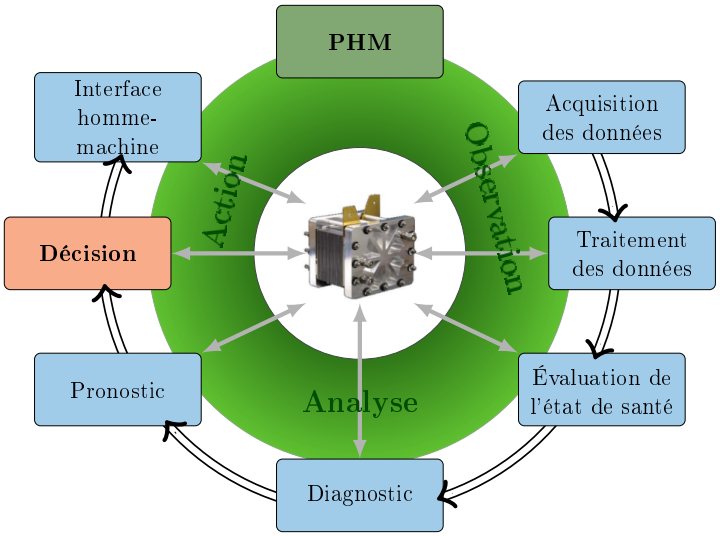


Figure 1.2 – Les différentes étapes du processus *PHM*

Traitement des données Une fois les données récupérées, il est généralement nécessaire de les traiter, par exemple pour supprimer les informations parasites provenant de bruits. Une extraction et une sélection des données représentatives et utiles pour l'étape suivante d'analyse peut aussi être requise.

Toutes les informations récoltées au cours de cette première phase d'observation mènent à des actions de contrôle, généralement embarquées, détaillées dans la partie suivante, et à des actions de planification faisant partie d'un processus logistique plus large (voir partie 1.1.2.3).

1.1.2.2 Analyse

Évaluation de l'état de santé La fonction principale du dispositif de contrôle de l'état de santé est de comparer les données pertinentes provenant de l'étape précédente à des valeurs attendues ou à des seuils définissant des limites de variation normale. La détection d'anomalies déclenche une alerte indiquant une évolution de l'état de santé du système surveillé.

Diagnostic Le diagnostic est un processus d'analyse des modes de défaillance des équipements [19]. Sa fonction principale est de déterminer si le système surveillé est dégradé ou non et de suggérer les causes probables de défaillance. Pour cela, il s'agit de déterminer les relations liant les causes et les effets, afin d'isoler les fautes et d'identifier les causes des défaillances [66].

Pronostic Alors que le diagnostic se concentre sur la détection, l'isolation et l'identification des défaillances au moment où elles surviennent, le pronostic vise à prédire ces défaillances avant leur apparition [54], au moyen d'un indicateur de santé qui reflète la détérioration attendue sous des conditions normales d'utilisation [19]. Ce processus dynamique évolue dans le temps à partir du moment où l'équipement commence à être utilisé, jusqu'à ce qu'il tombe en panne [24]. Le pronostic peut être défini comme un processus d'estimation de la durée de vie résiduelle (*RUL* pour Remaining Useful Life) et de la date de fin de vie (*EOL*, pour End of Life) [92] des

équipements en service, de manière à fournir des informations utiles au processus de décision [90, 64]. Le RUL correspond au temps restant avant l'observation d'une défaillance, étant donné l'état de santé courant du système et ses conditions opératoires présentes et passées [54]. Il peut aussi être défini comme un temps opérationnel [64] ou une durée de fonctionnement avant défaillance.

Différentes approches de pronostic ont été développées au cours des dernières années. Un classement de ces approches sur trois niveaux a été proposé par Lebold et al. [64]. On retrouve les approches fondées sur des modèles physiques, celles guidées par les données provenant de la surveillance des équipements et les approches fiabilistes, fondées sur l'expérience.

1.1.2.3 Action

Décision La fonction principale de l'étape de décision est de fournir des recommandations pour l'ordonnancement des actions de maintenance et pour la modification, soit de la configuration du système considéré, soit du profil de la mission, dans le but de remplir les objectifs de la mission [58]. Il s'agit alors de définir une utilisation des différents résultats découlant de l'interprétation des données faite tout au long des étapes précédentes. Il est montré dans la suite du chapitre que les résultats du pronostic sont utilisés pour optimiser la maintenance dans un grand nombre de domaines différents, mais aussi pour définir une utilisation des machines prenant en compte à la fois les exigences de la production et leur état de santé.

Interface homme-machine Les différents résultats de l'étape d'analyse sont communiqués aux utilisateurs du système par le biais d'une interface permettant par exemple d'afficher les éventuelles alertes devant entraîner une prise de décision.

Les travaux faisant l'objet de ce manuscrit s'inscrivent dans la partie décisionnelle du processus *PHM* décrit ici. Une présentation plus détaillée de la décision post-pronostic et de ses applications est proposée dans la suite.

1.2 La décision post-pronostic

Les bénéfices apportés par le *PHM* sont fortement liés à la partie décisionnelle qui suit la récupération et l'interprétation des données du pronostic [52]. Le but est d'utiliser les résultats du pronostic pour déterminer les meilleures actions à entreprendre étant donné l'état du système.

Une définition de la décision post-pronostic est tout d'abord proposée. Ce type de décision est ensuite positionné sur une échelle temporelle et suivant la dimension du système concerné par la décision. Les applications principales qui sont faites des résultats du pronostic, à savoir le contrôle, l'optimisation de la maintenance et l'adaptation de missions à l'état de santé des équipements sont finalement développées.

1.2.1 Définition

Deux termes spécifiques se retrouvent dans la littérature pour caractériser la décision au sein du processus *PHM*. Iyer et al. [52] ont tout d'abord proposé le terme de « post-prognostic decision », défini comme une utilisation des informations produites par l'étape de pronostic pour

définir des actions incluant la maintenance, la gestion de la chaîne logistique, l'ordonnancement de missions et l'affectation de ressources. Le but de ce type de décision est de tirer profit de toutes les informations disponibles pour optimiser des indicateurs tels que les coûts associés au cycle de vie d'un système, le taux de réussite des missions, les délais de production ou la sécurité [52]. Balaban et al. [2] ont ensuite introduit le terme de « Prognostic Decision Making (PDM) » pour représenter le processus d'utilisation des informations fournies par le pronostic pour déterminer de manière autonome ou semi-autonome des actions effectuées par un système ou sa configuration.

Nous utiliserons dans la suite le terme de « décision post-pronostic » pour nommer la décision prenant en compte les résultats du pronostic sous la forme de durées d'utilisation résiduelles avant maintenance pour définir une utilisation des systèmes basée à la fois sur l'objectif d'optimisation considéré et sur leur état de santé.

1.2.2 Typologie des décisions au sein du *PHM*

Les différentes contributions incluant de la décision au sein du *PHM* peuvent être classifiées suivant différents axes. Nous proposons sur la figure 1.3 une représentation des différents types de décision suivant une échelle temporelle et suivant un axe représentant la dimension du système considéré. Différents types de décision peuvent être mis en évidence à la croisée de ces deux axes, allant du contrôle (au sens de la commande dans le domaine de l'automatique) à des actions logistiques plus étalées dans le temps.

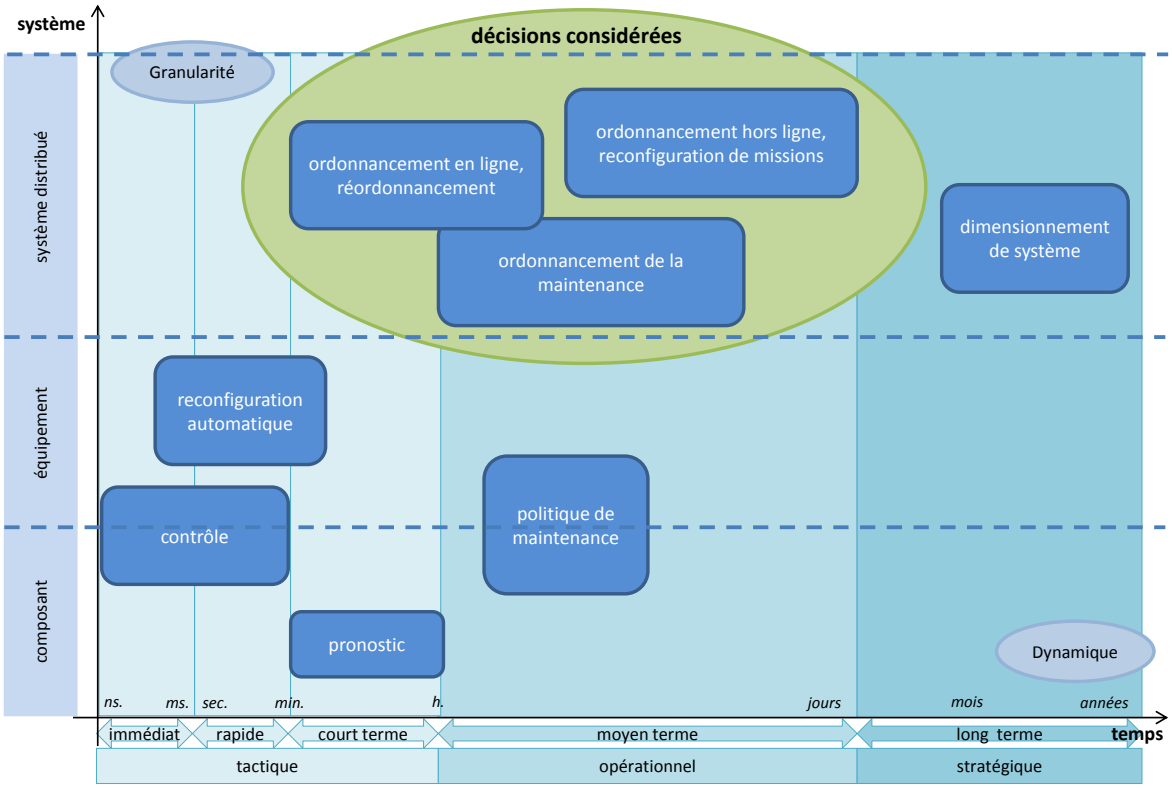


Figure 1.3 – Typologie de la décision post-pronostic

La classification suivant l'échelle temporelle proposée suit la segmentation temporelle de la décision introduite par Bonissone et al. [19] basée sur les horizons de décision. Trois classes ont été identifiées par ces auteurs :

(*i*) les décisions uniques, effectuées une seule fois ;
(*ii*) les décisions multiples et répétées ;
(*iii*) les décisions à plus grande échelle impactant le cycle de vie du système considéré.

Les cas d'applications concernés par nos travaux de recherche s'inscrivent dans la seconde classe, dans laquelle les décisions peuvent être classées suivant trois niveaux en fonction de leur fréquence. Ces niveaux sont appelés de façon classique niveaux tactique (de la nanoseconde à l'heure), opérationnel (jours ou semaines) et stratégique (mois ou années).

Au sein du niveau tactique, on retrouve ainsi un premier niveau immédiat (de la nanoseconde à la milliseconde) regroupant des décisions automatiques de contrôle telles que celles rencontrées dans les domaines de l'électronique, l'électro-mécanique ou l'automatique. L'auto-diagnostic et le contrôle en temps réel entrent dans ce niveau de décision.

On trouve ensuite un second niveau, rapide, pour des fréquences de l'ordre de la seconde, qui comprend des décisions semi-autonomes s'appliquant par exemple dans le domaine de l'automatique pour des tâches de contrôle utilisant le pronostic, de supervision ou de détection d'anomalies. Dans un niveau à court terme, de la minute à plusieurs heures, on peut trouver des décisions de portée plus grande telles que l'ordonnancement en ligne ou le réordonnancement. Le diagnostic et le pronostic entrent aussi dans cette catégorie.

Le niveau opérationnel rassemble quant à lui des décisions à moyen terme, considérant des intervalles de décision plus longs (jours ou semaines), par exemple pour la planification, l'ordonnancement hors ligne ou la reconfiguration de mission. Les travaux présentés dans ce manuscrit entrent dans les deux dernières catégories du processus de décision, à court terme et à moyen terme.

1.2.3 Les différentes utilisations des résultats du pronostic dans la littérature

De nombreuses études entrant dans les catégories temporelles présentées précédemment ont été proposées dans la littérature traitant du *PHM* et ce, pour différents domaines d'application. On peut par exemple citer le domaine aérospatial [2, 25, 81], énergétique avec des applications sur des éoliennes [13, 47, 62, 103, 67] ou des batteries [88, 10, 84], ou encore des études sur des systèmes électroniques [90, 105, 5] ou d'usinage [24]. Un survol des différentes contributions est proposé dans la suite pour les trois types principaux de décision que sont le contrôle, l'optimisation de la maintenance et la gestion de missions de différents types.

1.2.3.1 Le contrôle automatique amélioré par le pronostic

Plusieurs travaux se sont attachés à améliorer le contrôle de systèmes en prenant en compte des résultats du pronostic pour définir des actions automatiques associées à des délais de décision très courts.

Pereira et al. [78] ont par exemple développé une approche de contrôle basée sur un modèle prédictif pour la gestion d'actionneurs qui distribuent les tâches de contrôle sur plusieurs

unités redondantes. Cette allocation des tâches est effectuée en prenant en considération des informations fournies par le pronostic sur la dégradation des unités de contrôle. Bole et al. [18] ont étudié la répartition de tâches sur la base des données du pronostic pour le contrôle d'un véhicule autonome sans pilote soumis à des dégradations dues à des conditions thermiques induisant des problèmes d'isolation du moteur. On peut également citer les travaux de Bogdanov et al. [17] sur des servomoteurs ou ceux de Brown et al. [20, 21] s'appliquant à des contrôleurs électro-mécaniques tolérants aux fautes.

Le contrôle automatique amélioré par le pronostic appliqué dans ces différents travaux est associé à des fréquences de décision élevées et son impact est limité par la nécessaire rapidité des opérations. Ce type de décision n'entre alors pas dans le champ des applications considérées dans ce manuscrit.

1.2.3.2 L'optimisation de la maintenance

La plupart des études proposées dans la littérature traitant du *PHM* se concentrent sur la planification de la maintenance. Le *PHM* permet en effet de baser la maintenance sur l'état de santé actuel des composants susceptibles de générer des pannes et sur l'état de santé global du système [25]. D'un point de vue général, Asmai et al. [1] ont développé un outil intelligent de pronostic pour la maintenance utilisant une approche basée sur les données. Le but est de prévenir les utilisateurs suffisamment tôt pour qu'ils aient le temps de planifier les opérations de maintenance nécessaires. Un grand nombre de contributions ont été proposées sous la forme de règles de maintenance dans des domaines d'application très variés.

Systèmes électroniques Sandborn et al. [90] se sont par exemple attachés à déterminer l'instant propice pour la maintenance de systèmes électroniques. Peu d'études ont été faites sur ce type de systèmes dans le cadre du *PHM*, car la durée de vie des composants électroniques était considérée comme étant bien plus longues que le temps d'utilisation des systèmes [90]. Le domaine électronique est de plus caractérisé d'un point de vue du *PHM* par des conditions de surveillance difficiles rendant les données récupérées imparfaites et incomplètes [90]. Plusieurs modèles d'ordonnancement de la maintenance ont toutefois été développés en ne considérant pour la plupart aucune incertitude dans la surveillance et en faisant l'hypothèse d'une connaissance parfaite de l'état de santé [102, 32]. D'autres travaux ont traité la maintenance des systèmes électroniques en prenant en compte les limites de la surveillance [105, 5].

Domaine aérospatial Un algorithme utilisant des méthodes probabilistes et des informations du pronostic a été développé par Balaban et al. [2] pour générer des règles de maintenance pour des applications aérospatiales. Camci et al. [25] ont quant à eux proposé un outil d'intégration des données de maintenance dans le processus *PHM* pour permettre l'utilisation des méthodes de diagnostic et de pronostic en environnement réel sur des avions de chasse.

Fermes d'éoliennes Beaucoup d'études proposées dans un contexte *PHM* se concentrent sur l'optimisation de la maintenance de fermes d'éoliennes [13, 47, 62, 103, 67]. Des aspects spécifiques liés à la maintenance définissent en effet des contraintes particulièrement sévères pour ce type d'application, telles que le besoin de ressources de maintenances non traditionnelles (équi-

pements spécifiques et main d'œuvre très qualifiée) et de pièces de rechange encombrantes et coûteuses [62], ou encore des contraintes liées aux conditions météorologiques [13]. C'est d'autant plus le cas lorsque les éoliennes considérées sont implantées au large des côtes (éoliennes offshore).

Lei et al. [67] ont proposé une optimisation de la maintenance d'éoliennes offshore utilisant une analyse par les options réelles. Cette étude basée sur un outil financier d'aide à la décision a mené à la conclusion suivante : le temps de déclenchement optimal d'une opération de maintenance prédictive n'est pas le même lorsqu'une ferme d'éoliennes ou une éolienne seule est considérée. Ils ont ainsi montré que les actions les plus appropriées ne sont pas les mêmes si le système monitoré fonctionne de façon individuelle ou s'il fait partie d'un ensemble plus large de systèmes interconnectés et qu'il est important de prendre en compte l'environnement du système pour définir des actions intelligentes basées sur les durées résiduelles d'utilisation (*RUL*).

Différentes contributions ont considéré les éoliennes individuellement. Un nouveau seuil variable pour la détermination de l'état de santé a par exemple été proposé par Vieira et al. [103] pour apporter des informations aidant à réordonnancer et optimiser la maintenance d'un parc d'éoliennes. L'objectif considéré est la maximisation du cycle de vie des composants de chaque éolienne et il a été montré que l'optimisation de la maintenance peut aider à améliorer l'usage qui est fait de chaque éolienne. En plus d'une optimisation de la maintenance, les informations apportées par les étapes d'analyse du *PHM* peuvent ainsi permettre une optimisation de l'utilisation des machines considérées. Besnard et al. [13] ont quant à eux proposé une approche pour optimiser la maintenance conditionnelle de composants d'éoliennes pour lesquels la dégradation peut être classifiée suivant la sévérité des dommages. Ils ont comparé trois stratégies de maintenance des pales d'éoliennes. La première est basée sur des inspections visuelles, la seconde sur des techniques de surveillance de l'état de santé utilisant des ultrasons ou la thermographie. La dernière considère une surveillance continue des pales par fibre optique. Une étude des coûts de maintenance basée sur la méthode de simulation de Monte-Carlo a permis de déterminer la stratégie optimale en fonction des conditions d'usure.

D'autres travaux se sont intéressé à l'optimisation de la maintenance d'une ferme d'éoliennes. Haddad et al. [47] ont par exemple proposé une optimisation consistant à définir un sous-ensemble optimal d'éoliennes offshore à maintenir, étant données des informations sur leur dégradation, les futurs besoins en disponibilité et des contraintes de coût. Kovacs et al. [62] ont quant à eux utilisé une formulation optimale du problème d'optimisation basée sur un programme linéaire mixte en nombres entiers pour ordonnancer les opérations de maintenance.

1.2.3.3 L'adaptation des missions à l'état de santé des équipements

En plus de l'application classique pour l'optimisation de la maintenance, les données du pronostic peuvent être utilisées pour optimiser les missions de production ou des missions d'autres types. La définition d'une utilisation des informations sur l'état de santé pour la prise de décision autonome ou semi-autonome quant à la reconfiguration d'un système ou le réordonnancement de missions n'en est encore qu'à ses débuts [2]. Plusieurs travaux ont toutefois été proposés dans des contextes bien spécifiques, incluant l'ordonnancement de la production [1, 47], la gestion de véhicules autonomes [80, 97, 3], la gestion de batteries [88, 10, 84] et la gestion de réseaux de capteurs [41, 106].

Dans un contexte plus général, Iyer et al. [52] ont proposé un système de support à la décision multiobjectifs permettant d'aider les utilisateurs à prendre de « bonnes » décisions quant à la gestion d'une flotte de systèmes. Ce système est composé de deux modules. Le premier est un module de simulation permettant de modéliser et de tester des scénarios puis d'évaluer les performances associées à chacun d'entre eux par rapport à l'objectif poursuivi. Le second module permet de rechercher les solutions optimales parmi tous les scénarios générés. Si aucune solution optimale n'existe, des changements de paramètres sont proposés pour obtenir des solutions optimales [52]. Une relation de collaboration entre l'utilisateur et le système est défini pendant le processus de décision pour permettre la sélection et l'évaluation de scénarios possibles et des stratégies optimales.

Balaban et al. [2] ont proposé une formalisation de l'approche de décision post-pronostic. Le rôle du système développé est de remettre en cause le plan de mission initial et de trouver un compromis approprié (idéalement optimal) entre l'extension de la durée d'utilisation avant maintenance du système et la maximisation de l'efficacité de la mission, correspondant généralement à la finalisation d'un maximum d'étapes. La même idée se retrouve dans les différentes études développées ci-après.

Ordonnancement de la production Dans [1], Asmai et al. ont montré que la connaissance du *RUL* peut être très utile pour l'ordonnancement de la production. Cette valeur donne en effet une information sur l'état de santé de l'équipement de production, qui peut être prise en compte lors de la décision de lancement de nouvelles tâches de production. Cela peut permettre d'éviter des pertes de production et du gâchis de matières premières qui accompagneraient une panne se produisant au cours d'une opération de production. La décision utilisant l'information fournie par le pronostic peut prendre plusieurs formes, telles qu'un arrêt immédiat de la machine pour éviter des dommages supplémentaires, la continuation de la production normale, une intervention de maintenance préventive, une modification du réglage de la machine pour réduire la charge [47], ou encore un réordonnancement de la production. L'utilisation des résultats de la phase de pronostic peut donc être étendue à la modification des conditions opératoires des machines ou des profils de mission, dans le but de maximiser l'accomplissement des objectifs d'une mission [64, 58]. Cela est valable dans un contexte de production, mais aussi pour des missions de types différents.

Gestion de véhicules autonomes Plusieurs études ont été proposées pour des systèmes autonomes, dont le développement pour un nombre croissant de domaines d'application a fait émerger un nouveau besoin allant au-delà de la gestion de la maintenance seule. Ce type de système doit être capable de prendre des décisions pour remplir une mission donnée tout en s'adaptant à des changements éventuels de conditions dans un délai le plus court possible. La mission est composée d'une séquence de tâches à effectuer suivant un certain ordre et est considérée complétée lorsque toutes les tâches ont été effectuées [80, 82]. Le bon déroulement de ces tâches peut être affecté par des défaillances internes au système, mais aussi par des facteurs extérieurs tels que les conditions météorologiques. Un exemple type de ce genre de système est l'ensemble des véhicules autonomes sans pilotes (AUVs, pour Autonomous Unmanned Vehicles), qui sont de plus en plus utilisés pour des missions d'exploration sur terre, à la surface de la mer, sous l'eau, ou encore pour des missions aériennes (drones) ou dans l'espace (robots mobiles) [97]. Les missions concernées incluent entre autres des opérations de surveillance et d'exploration, de recherche et

de secours, ainsi que des interventions dans des zones contaminées ou d'accès restreint. Dans ce contexte, Prescott et al. [80] ont proposé une stratégie d'analyse de la fiabilité de tels systèmes pour des missions présentant les caractéristiques citées précédemment. Cette stratégie comporte une phase de pronostic en temps réel et est amenée à être intégrée dans un module de décision tel que celui proposé par les mêmes auteurs dans [81] pour des drones militaires.

Tang et al. [97] ont développé un prototype de véhicule test pour évaluer et valider des techniques de gestion des évènements imprévus pour des véhicules autonomes. Ces techniques incluant du diagnostic et du pronostic ont été automatisées, développées pour fonctionner en temps réel et basées sur des données du *PHM* par les mêmes auteurs dans [98, 99, 38, 74, 35] et [107]. Le prototype présenté dans [97] a été soumis à plusieurs types de défaillance, incluant des fuites et des crevaisons au niveau des pneus, des niveaux faibles de charge et des dégradations pour la batterie, la récupération de données erronées sur l'état du véhicule ou des courts-circuits dans le moteur [97]. Des estimations de *RUL* ont été utilisées de deux manières différentes : elles ont soit été prises en compte en tant que contraintes, soit directement intégrées à la fonction de coût utilisée pour la définition des trajectoires du véhicule. Différentes trajectoires ont été testés, pour lesquels différents profils de puissance doivent être fournis pour remplir les missions. Chaque trajectoire à suivre a été définie en fonction de la mission à effectuer et de l'état de santé du véhicule. L'optimisation de la planification de chaque mission a ensuite été faite en temps réel sur la base des prédictions de la durée d'utilisation résiduelle de la batterie avant charge. Une prédiction de la fin de vie de la batterie basée sur son utilisation et sur les dégradations affectant sa capacité de charge a de plus été proposée et utilisée pour la planification à long terme de plusieurs missions. Les auteurs ont montré dans [97] que lorsque la mission est optimisée en considérant la durée d'utilisation de la batterie comme critère, le véhicule parcourt une distance plus grande et met plus de temps à finir la mission, mais use moins la batterie. A l'inverse, lorsque le véhicule est utilisé à une vitesse plus élevée pour optimiser le temps de la mission, il parcourt une distance plus faible et consomme plus d'énergie.

Un type similaire de décision autonome a été proposé par Balaban et al. [3], qui ont étudié l'utilisation des valeurs de *RUL* pour la gestion des missions d'un robot mobile autonome. Une plate-forme de test basée sur un prototype d'un tel robot, décrite dans [4], a été développée pour définir et tester, entre autres, des algorithmes de décision capables de gérer des défaillances dues à des détériorations mécaniques, des défauts électroniques ou une charge de batterie insuffisante. L'objectif n'est pas seulement de déterminer le *RUL* d'un composant, mais aussi de suggérer des actions qui peuvent optimiser la maintenance du robot, assurer la sûreté de la mission, ou bien étendre la durée de la mission. L'idée principale communiquée est la suivante : si les décisions sont faites en tenant compte de l'évolution dans le temps de l'état de santé du système, l'efficacité de la mission peut être maximisée avant l'atteinte des limites d'énergie et de santé. Il s'agit alors de modifier le plan de mission en cours si une défaillance est prévue ou se produit. Une reconfiguration du robot peut aussi être recommandée pour alléger la charge subie par le composant limitant et ainsi allonger sa durée de vie, de manière à assurer, si possible, l'atteinte des objectifs de la mission.

Gestion de batteries Les dernières applications développées pour la gestion de véhicules autonomes considèrent les batteries comme des éléments limitants et se basent sur des prévisions de la durée d'utilisation résiduelle des batteries. Plusieurs techniques ont été proposées pour

effectuer ces prévisions au sein du domaine *PHM* [86, 44, 87].

Le problème de la gestion de l'utilisation des batteries a aussi été traité comme un problème à part entière. Saha et al. [88] ont par exemple développé un modèle de décharge pour des batteries utilisées pour l'alimentation de drones. Ils ont montré que, pour un certain profil de demande, le taux de décharge d'une batterie ne dépend pas seulement de son état initial de charge, mais aussi d'autres facteurs tels que son état de santé et le profil de décharge imposé. Dans le même contexte, une partie des mêmes auteurs a proposé dans [89] un nouveau modèle de pronostic permettant d'étudier le comportement des batteries avec de fortes incertitudes sur le profil de demande. L'objectif considéré dans cette contribution est d'optimiser l'utilisation de l'énergie disponible. Il s'agit alors de définir le profil de vol du drone qui maximise l'utilisation de la capacité de la batterie, tout en limitant la probabilité d'une coupure de l'énergie en cours de vol.

Le développement de batteries constituées de plusieurs blocs ayant fait émerger un nouveau besoin, Benini et al. [10, 11] ont introduit le concept d'« ordonnancement de batteries » pour des systèmes portatifs. La distribution de la capacité totale en plusieurs blocs pénalise en effet la charge totale pouvant être fournie par la batterie. À charge équivalente, une batterie partitionnée a de plus une durée de vie plus faible qu'une batterie monolithique, constituée d'un seul bloc [10]. Le problème considéré par Benini et al. est alors le suivant : étant donné un système de batteries partitionné composé de plusieurs blocs-batteries, il s'agit de maximiser la durée de vie totale du système pour une charge de travail donnée. Chaque bloc-batterie étant caractérisé par sa propre capacité nominale, différentes configurations ont été testées pour la décharge de tous les blocs-batteries (en série, statique ou dynamique). Les résultats ont montré que la définition d'une utilisation adaptée des différents blocs permet d'améliorer la durée de vie d'une batterie partitionnée. Les meilleures stratégies d'ordonnancement peuvent même amener cette durée de vie à une valeur équivalente à celle d'une batterie monolithique classique [10]. Les mêmes résultats ont été obtenus par Rao et al. [84] pour des systèmes électroniques portatifs. Ces auteurs ont de plus défini des bornes pour la durée de vie d'une batterie monolithique ainsi que pour celles de systèmes composés de plusieurs batteries utilisées pour fournir une énergie constante.

Gestion de réseaux de capteurs Quelques travaux ont considéré le problème de gestion de réseaux de capteurs sans fil. Les capteurs sont habituellement utilisés avec une alimentation filaire pour récupérer les informations nécessaires aux différentes étapes d'analyse du processus *PHM* [40]. Une configuration sans fil présente l'avantage de permettre la récupération de données dans des environnements difficiles, tels que des zones surveillées très vastes et difficiles d'accès [41] telles que les profondeurs sous-marines ou encore dangereuses telles que les champs de batailles militaires [94]. Sur des équipements de production ou des systèmes embarqués, l'absence de fils peut aussi permettre de placer les capteurs aux endroits exacts les plus propices pour améliorer la précision des mesures effectuées [40]. Un réseau de capteurs sans fil est typiquement composé de quelques stations récoltant les informations et d'une centaine ou plus de capteurs [40]. Chaque capteur est capable d'effectuer des mesures et de communiquer à la station à laquelle il est connecté les informations récupérées à son emplacement.

Le point limitant dans l'utilisation de tels réseaux est lié à leur alimentation. Les capteurs sont en effet généralement alimentés par des batteries de petite capacité et non rechargeables [30]. Le contrôle de l'utilisation de ces capteurs est donc crucial pour assurer une certaine durée de

vie du réseau. La gestion d'un réseau de capteurs est particulière car la topologie du réseau peut évoluer dans le temps en fonction des potentielles défaillances des capteurs ou de la disponibilité de l'énergie [41]. Elghazel et al. [40, 41] ont proposé une application de l'utilisation d'un tel réseau pour du diagnostic. Le suivi de l'état de santé des capteurs est utilisé pour définir une utilisation intelligente de ces capteurs permettant une surveillance efficace du système à diagnostiquer.

Dans un autre contexte, Yontay et al. [106] ont étudié l'optimisation de l'utilisation d'un réseau de capteurs sans fil dédié à la surveillance des dégradations à l'intérieur de structures composites en fibres de carbone. Le but est de maximiser la durée de vie du réseau de capteurs pour qu'il puisse accomplir sa mission de détection des dégradations et de transfert des informations récoltées. Pour cela, ils proposent de baser la fréquence d'inspection sur un modèle de dégradation de la structure considérée [106]. Si la structure est saine, un maximum de capteurs sont désactivés pour économiser leur énergie et la fréquence d'inspection est gardée à un niveau minimal. Lors de la dégradation de l'état de santé de la structure, prédite par le modèle développé dans la même étude, la fréquence d'inspection est augmentée pour éviter qu'une dégradation ne soit pas détectée. L'ordonnancement d'inspection proposé dans [106] est ainsi dynamique et basé sur un modèle de dégradation. Il maximise la durée de vie du réseau de capteurs, tout en lui permettant d'assurer sa fonction première de surveillance.

1.2.4 Les perspectives pour la décision post-pronostic

Les systèmes utilisés sont de plus en plus complexes et évoluent dans un contexte dynamique. C'est par exemple particulièrement le cas dans le domaine aéronautique, au sein duquel de mauvaises décisions peuvent mener à des dégâts importants, voire à des accidents mortels. Selon Balaban et al. [2], qui ont étudié la prise de décision basée sur les données du pronostic dans ce contexte, il serait avantageux de prendre en compte les besoins en informations nécessaires dans le processus *PHM* dès la conception des systèmes. Cela inclut la mise en place de capteurs aux endroits stratégiques, l'intégration de capacités de calcul suffisantes, la définition de procédures et la conception d'une architecture de communication adaptée [2].

La prise en compte des conditions opératoires futures des systèmes dans le processus de détermination des valeurs de RUL est aussi un enjeu crucial pour le développement de décisions post-pronostic adaptées et performantes. Les différentes définitions et applications du pronostic que l'on peut trouver dans la littérature considèrent en effet pour la plupart uniquement l'état de santé courant des machines et leurs conditions opératoires antérieures. Une grande majorité des techniques de pronostic permettent de plus de définir une durée de vie résiduelle pour des conditions opératoires définies et fixes tout au long de l'utilisation de la machine. Ces conditions opératoires peuvent toutefois changer et entraîner une accélération ou un ralentissement des phénomènes d'usure, ce qui peut faire évoluer la valeur du RUL. Même si la prise en compte de conditions opératoires variables au cours de l'utilisation d'une machine n'est pour l'instant pas totalement maîtrisée pour l'estimation des valeurs de RUL [42], de récents développements permettent d'imaginer des prévisions prenant en compte l'état de santé présent des machines, mais aussi leur état à venir, fonction des conditions opératoires futures. Une méthode de prédiction basée sur une modélisation d'un indicateur de santé variant avec les conditions opératoires a par exemple été récemment proposée par Nguyen et al. [73], permettant de déterminer un RUL en considérant à la fois l'état de santé courant du système et son utilisation future.

1.3 Synthèse

Le processus *PHM* au sein duquel se positionnent les travaux proposés dans ce manuscrit a été introduit par la description des différentes étapes qui le composent. Ces étapes peuvent être regroupées en trois groupes correspondant chacun à une action bien spécifique :

1. *l'observation* du système surveillé pour l'acquisition d'informations relatives à son état de santé ;
2. *l'analyse des données* récoltées pour évaluer l'état de santé du système et pronostiquer une durée de vie résiduelle avant défaillance ;
3. *l'action* basée sur les résultats du pronostic pour optimiser la maintenance du système ou définir une utilisation prenant en compte son état de santé.

La décision post-pronostic a ensuite été définie et développée suivant deux axes principaux correspondant aux échelles de temps prises en compte dans les travaux de recherche proposés dans ce manuscrit, à savoir l'optimisation de la maintenance des équipements et l'adaptation de la production ou de missions à l'état de santé des équipements. La plupart des travaux proposés dans un contexte de décision post-pronostic traitent du premier axe en cherchant à définir des stratégies de maintenance basées sur la connaissance de l'état de santé des équipements. On remarque tout de même un intérêt croissant pour l'utilisation des données du pronostic dans la gestion de l'utilisation des équipements avant maintenance. Une grande partie des travaux suivant ce deuxième axe pour l'optimisation de la production ou des missions se concentre sur des applications mettant en jeu une seule machine ou un système isolé. Les cas d'application considérés dans ce manuscrit vont plus loin dans le sens où ils prennent en compte un ensemble de machines utilisées de façon collaborative pour remplir une mission commune. L'objectif est alors de définir une utilisation de ces machines qui optimise à la fois l'utilisation de chacune d'entre elles au regard de leur état de santé respectif et l'utilisation globale du parc de machines. Ce type de décision se retrouve dans les domaines de la gestion de batteries et de la gestion de réseaux de capteurs. Nos travaux de recherche s'inscrivent dans la même idée mais considèrent des machines de production ayant des caractéristiques différentes.

Le point commun des différents travaux mentionnés dans le cadre de l'optimisation de la production ou de missions présentés dans ce chapitre réside dans la possibilité de modifier les performances des systèmes utilisés afin d'optimiser leur utilisation. Différents types de reconfiguration affectant les performances de machines peuvent être trouvés dans la littérature traitant de l'ordonnancement, amenant à la définition de plusieurs problèmes d'optimisation. Ce sujet est développé dans le chapitre suivant.

Chapitre 2

L'ordonnancement

Sommaire

Ce chapitre s'attache à positionner les travaux de recherche proposés dans cette thèse au sein du domaine de l'ordonnancement. La notion d'ordonnancement se retrouve dans des domaines très différents tels que l'informatique (avec la définition de programmes et de l'utilisation de ressources comme les processeurs ou la mémoire), l'industrie (avec la gestion de la production définissant un ordre pour les tâches à exécuter et les machines à utiliser), ou l'administration (avec la construction d'emplois du temps ou le suivi de projets). Les problèmes d'ordonnancement traités dans la littérature présentent ainsi une diversité très grande, que ce soit du point de vue de l'application, des ressources utilisées, des contraintes à respecter ou de l'objectif à atteindre. Toutes les contributions partagent toutefois un but global commun, qui est l'amélioration de l'utilisation des ressources.

De nombreuses techniques ont été développées pour faire face aux problèmes d'ordonnancement en adaptant l'utilisation de ces ressources de façon à atteindre les objectifs considérés. Le concept de reconfiguration évoqué dans le chapitre précédent et utilisé dans des travaux du domaine *PHM* se retrouve ainsi dans la littérature traitant de l'ordonnancement. Un survol des différentes formes de reconfiguration suit la présentation générale de la thématique de l'ordonnancement, permettant ainsi de positionner notre contribution, à savoir l'utilisation de machines avec différents profils de fonctionnement au sein des moyens existants dans le but d'agir sur la production fournie.

2.1 Ordonnancement

Une définition de l'ordonnancement est tout d'abord donnée dans un cadre très général. Les développements sont ensuite concentrés sur l'ordonnancement de la production, qui correspond au problème traité dans ce manuscrit. Une classification des différents problèmes d'ordonnancement de la production permet de mettre en évidence l'existence d'un grand nombre de variantes. En fonction du type de machines, des contraintes ou des objectifs considérés, les problèmes peuvent en effet être très différents et amenés à être résolus avec des méthodes de résolution faisant appel à plusieurs techniques. Un survol des principales méthodes de résolution utilisées pour l'optimisation des ordonnancements est alors proposé.

2.1.1 Ordonnancement : définition générale

L'ordonnancement consiste de façon générale à programmer l'exécution de différentes tâches en attribuant des ressources à chacune d'entre elles et en fixant leurs dates d'exécution [29]. C'est un processus de décision visant à optimiser un ou plusieurs objectifs [79], tels que la minimisation de la durée totale d'exécution des tâches, le respect des dates de commande ou encore la minimisation d'un coût. De manière générale, une utilisation efficace des ressources, un délai d'exécution des tâches aussi faible que possible et le respect des dates d'achèvement sont les critères principaux pour un ordonnancement efficace de la production. Ces trois objectifs sont essentiels dans la résolution des problèmes d'ordonnancement [29] et se retrouvent dans la grande majorité des contributions dans le domaine.

Des différences notables apparaissent suivant le domaine d'application, l'échelle de temps considérée et la disponibilité des informations sur les tâches à effectuer (telles que leur date de début). Bellanger [8] a proposé une classification des problèmes d'ordonnancement en quatre

familles principales :

- *L'ordonnancement temps réel*, qui s'applique aux systèmes temps réel pour lesquels les tâches à effectuer ne sont connues qu'à l'instant où elles sont disponibles. Deux types de tâches peuvent être différenciés, à savoir les tâches récurrentes, associées à des fenêtres temporelles à planifier, et des tâches accidentelles prioritaires. On retrouve ce type d'ordonnancement dans les systèmes embarqués, les systèmes de surveillance ou encore pour l'allocation de ressources par le système d'exploitation d'un ordinateur ;
- *L'ordonnancement de grille* vise à ordonnancer des tâches dans des grilles de calcul, qui sont des systèmes distribués sur plusieurs machines parallèles. Le nombre de machines disponibles pouvant fluctuer, ainsi que leurs performances, un tel ordonnancement se doit d'être robuste face à l'évolution des ressources ;
- *L'ordonnancement de projet* a pour but la planification de la réalisation de projets de types divers. Ces projets comportent généralement un nombre relativement réduit de tâches, mais un grand nombre de ressources et de contraintes. Ce type d'ordonnancement est utilisé pour des fabrications uniques telles que la construction d'ouvrages ou le développement de nouveaux produits ;
- *L'ordonnancement de la production* est la famille la plus étudiée. Il consiste à gérer les décisions opérationnelles de la gestion de production. À l'inverse de l'ordonnancement de projet, un grand nombre de tâches diverses doivent généralement être affectées à un nombre limité de ressources.

C'est ce dernier type d'ordonnancement qui est considéré dans ce manuscrit. Un rapide survol des différentes typologies des problèmes d'ordonnancement de la production est proposé dans la partie suivante, présenté suivant une classification des problèmes d'ordonnancement largement utilisée dans la communauté.

2.1.2 Classification des problèmes d'ordonnancement de la production

Une classification des problèmes d'ordonnancement, faite sur la base d'une notation composée de trois champs, a été initiée par Conway et al. [33], puis améliorée successivement par Kan [57], Graham et al. [46] et Blazewicz et al. [16] entre 1967 et 1983. Cette notation, donnée sous la forme $\alpha|\beta|\gamma$, a été largement adoptée par la communauté s'intéressant aux problèmes d'ordonnancement pour identifier et caractériser les différents problèmes traités et a été enrichie par la suite.

Le premier champs α spécifie l'environnement des machines. Il est constitué de deux éléments α_1 et α_2, avec α_1 décrivant le type de machines utilisées et α_2 précisant le nombre de machines disponibles. Une description des différents types de machine couverts par la classification est donnée dans la suite avec la notation correspondante le cas échéant. Pour les différentes descriptions, un ensemble de tâches est nommé un travail. Les différentes configurations possibles sont les suivantes :

- *une seule machine disponible*, permettant de réaliser des travaux composées d'une seule tâche ;
- *plusieurs machines parallèles*, remplissant toutes les mêmes fonctions et permettant à nouveau de réaliser des travaux composés d'une seule tâche. Différentes configurations peuvent être identifiées en fonction de la vitesse d'exécution associée à chaque machine :

machines identiques (P) : toutes les machines ont la même vitesse d'exécution, quelle que soit la tâche ;

machines uniformes (Q) : chaque machine a une vitesse d'exécution propre et identique pour toutes les tâches ;

machines indépendantes (R) : la vitesse d'exécution varie suivant la machine, mais aussi suivant la tâche à exécuter ;

– *plusieurs machines dédiées*, chacune étant spécialisée à l'exécution de certains types de tâche. Plusieurs machines différentes doivent être utilisées pour réaliser les différentes tâches d'un même travail, suivant une gamme de fabrication. Là encore, différentes configurations peuvent être identifiées en fonction du mode de passage des tâches sur les machines :

atelier à cheminement unique ou Flow Shop (F) : toutes les machines sont utilisées dans chaque gamme de fabrication, suivant le même ordre. La ligne de production à la chaîne est un exemple typique de ce genre de configuration ;

atelier à cheminements multiples ou Job Shop (J) : chaque travail est effectué suivant sa propre gamme de fabrication, définissant une utilisation d'un certain sous-ensemble des machines disponibles, dans un certain ordre ;

atelier à cheminement libre ou Open Shop (O) : même définition que pour le Job Shop, mais avec un ordre de passage sur les machines non fixé à priori.

Le second champs β permet d'indiquer un certain nombre de caractéristiques des tâches et des machines. On retrouve par exemple les caractéristiques suivantes :

pmtn : la préemption est autorisée, c'est à dire que toute tâche peut être interrompue au cours de son exécution et redémarrée plus tard au même point, sans perte du travail déjà effectué ;

res : l'utilisation des ressources est limitée à s unités de temps, partagées par les travaux ;

prec : une relation de précédence entre les travaux est spécifiée, contraignant leur exécution suivant un certain ordre. Un travail ne peut être démarré avant la fin du précédent ;

r_j : des dates de début au plus tôt sont définies pour chaque travail, retardant le début de leur exécution si $r_j > 0$;

m : le nombre de tâches pour chaque travail est limité par une borne supérieure ;

p : le temps d'exécution de chaque tâche est fixé ou limité par une borne inférieure et une borne supérieure.

Le dernier champs γ précise enfin le critère d'optimisation choisi. Pour un ordonnancement, de nombreux critères peuvent être envisagés. Il s'agit dans la majorité des cas de minimiser l'un des critères suivants :

C_{max} : temps d'exécution total ;

L_{max} : retard maximum ;

$\sum C_i$: somme des dates de fin d'exécution de toutes les tâches ;

$\sum T_j$: somme des retards de toutes les tâches ;

$\sum U_j$: somme des indicateurs de retard, dont la valeur correspond au nombre de retard.

L'énumération de plusieurs caractéristiques pouvant être représentées par la notation $\alpha|\beta|\gamma$ montre qu'un très grand nombre de problèmes différents peut être considéré dans le domaine de l'ordonnancement de la production. Différentes caractéristiques telles que la quantité des

moyens disponibles, l'organisation de ces moyens, les types et les contraintes de fabrication, les hypothèses sur la demande ou les critères d'optimisation ont en effet un impact important sur la forme du problème et donc sur sa résolution.

2.1.3 Méthodes de résolution

Différentes méthodes de résolution peuvent être utilisées pour résoudre les problèmes d'ordonnancement. Ces méthodes utilisent un grand nombre de techniques d'optimisation différentes et peuvent être séparées en deux groupes distincts. Le premier rassemble les méthodes exactes, qui permettent d'obtenir des solutions optimales. L'utilisation de ces méthodes est restreinte à des problèmes présentant un nombre de variables limité, le plus souvent pour des raisons de complexité mathématique des problèmes à résoudre. D'autres méthodes de résolution approchées sont alors nécessaires pour le traitement de problèmes de grande taille. Le second groupe est ainsi constitué des méthodes heuristiques, qui permettent de trouver des solutions en temps polynomial.

2.1.3.1 Méthodes exactes

Les méthodes exactes ont pour but la construction d'une solution optimale pour un problème de taille finie et dans un temps limité. Quelques méthodes de résolution classiques sont présentées.

L'algorithme de séparation et évaluation (Branch and Bound) a été proposé par Land et al. [63] et est basé sur une méthode arborescente de recherche d'une solution optimale. L'ensemble des solutions est représenté sous la forme d'un arbre d'états composé de nœuds et de feuilles. La méthode consiste à parcourir l'espace des solutions seulement en partie, en s'assurant que la partie non explorée ne contient aucune solution qui puisse être meilleure que la dernière sélectionnée.

La recherche de solutions utilise deux concepts complémentaires : la séparation et l'évaluation. La séparation consiste à diviser l'ensemble des solutions en sous-ensembles. Trouver la meilleure solution dans chaque sous-ensemble permet de déterminer la solution optimale du problème initial. Utilisée seule, la séparation revient à effectuer une énumération exhaustive des solutions. Afin d'éviter cela, l'évaluation est utilisée pour réduire l'espace de recherche en éliminant les sous-ensembles ne contenant pas de solution optimale. Le coût de chaque nœud parcouru est comparé à la solution de plus bas coût rencontrée pendant l'exploration dans le cas d'une minimisation du critère d'optimisation (la solution de plus haut coût est considérée pour une maximisation). Si le coût du nœud considéré est supérieur (ou inférieur dans le cas d'une maximisation) au meilleur coût, l'exploration du sous-ensemble correspondant est arrêtée. Le parcours de l'arbre peut être effectué suivant différentes stratégies, telles que la profondeur d'abord, la largeur d'abord et le meilleur d'abord.

La programmation dynamique repose sur le principe d'optimalité énoncé par Bellman [9], spécifiant que toute solution optimale est composée de solutions intermédiaires optimales de sous-problèmes résolus localement. Appliqué de façon séquentielle, ce principe fournit des formules récursives pour la recherche de solutions optimales. Il s'agit alors de considérer de façon exhaustive l'ensemble des solutions possibles et de choisir la meilleure sur la base d'un critère. Différents sous-problèmes sont résolus de manière ascendante, c'est-à-dire

en démarrant avec le plus petit, pour déduire de façon progressive l'ensemble la solution optimale du problème initial.

Un algorithme de programmation dynamique est composé de quatre étapes successives [34] : (*i*) l'identification de sous-problèmes dont les solutions optimales vont permettre de reconstituer une solution optimale du problème initial ; (*ii*) la définition de la valeur de la solution optimale de manière récursive, à partir des valeurs des solutions optimales des sous-problèmes ; (*iii*) le calcul ascendant de la valeur de la solution optimale ; (*iv*) la reconstruction de la solution optimale, sur la base des valeurs calculées dans l'étape précédente.

La programmation linéaire s'applique aux problèmes d'optimisation sous contraintes, pour lesquels la fonction objectif et les contraintes sont toutes linéaires par rapport aux variables de décision, qui sont les inconnues. Différentes variantes de programmes linéaires peuvent être considérées en fonction de la nature des variables de décision. On peut trouver des programmes linéaires en nombres réels, rationnels ou entiers, en variables binaires ou mixtes. Toute affectation des variables de décision qui respecte les contraintes est une solution réalisable. La solution optimale, s'il en existe au moins une, est celle qui optimise la fonction objectif. Trois situations peuvent être rencontrées. Si l'ensemble des solutions réalisables est vide, le programme linéaire n'a pas de solution. Cela arrive lorsque les contraintes ne sont pas compatibles. Si cet ensemble des solutions est non vide, mais que la fonction objectif n'est pas bornée, il n'est pas possible de trouver une solution optimale au programme linéaire, que ce dernier en admette une ou non. Enfin, si l'ensemble des solutions est non vide et la fonction objectif est bornée, le programme linéaire admet une solution optimale, non nécessairement unique.

Une exploration exhaustive de toutes les solutions requiert un nombre d'opérations qui devient vite grand en fonction du nombre de variables. L'algorithme du simplexe est alors généralement utilisé pour réduire le nombre de solutions à explorer en limitant l'exploration aux sommets qui permettent d'améliorer la valeur de la fonction objectif pour les programmes linéaires à variables réelles [34]. D'autres méthodes de type Branch-and-Cut, basées sur le principe de séparation et évaluation, sont aussi utilisées pour les programmes linéaires plus complexes, par exemple à valeurs entières ou binaires.

2.1.3.2 Méthodes heuristiques

Les méthodes heuristiques permettent de trouver une solution réalisable en un temps polynomial. On cherche généralement à déterminer la meilleure possible, sans forcément avoir une garantie sur son optimalité. Ces méthodes peuvent être classées en trois familles différentes : les heuristiques constructives, les heuristiques d'amélioration et les métaheuristiques. Chacun de ces types d'heuristiques construit les solutions suivant une stratégie bien particulière.

Les heuristiques constructives élaborent des solutions de façon itérative à partir des données initiales. A chaque itération, une solution partielle est complétée, jusqu'à l'obtention de la solution complète. Ces heuristiques sont généralement rapides et relativement simples. La qualité des solutions est en revanche limitée, puisque l'ordonnancement est construit pas à pas, sans optimisation effectuée de façon globale.

Les heuristiques d'amélioration démarrent avec une solution admissible déjà construite et la modifient dans le but d'améliorer la valeur de la fonction objectif [85]. Une solution

initiale, généralement construite avec une heuristique constructive, sert ainsi de base à un processus itératif qui fait évoluer la solution complète en la remplaçant par un de ses voisins ayant une meilleure valeur pour la fonction objectif. Un critère d'arrêt est nécessaire pour ce type de fonctionnement. Il peut prendre la forme d'un nombre fixé d'itérations ou être basé sur une comparaison des solutions successives.

Les métaheuristiques sont à la fois plus complètes et plus complexes que les deux types d'heuristiques précédents. Elles sont basées sur un processus itératif qui combine de façon intelligente différents concepts pour explorer efficacement l'espace de recherche et exploiter les informations disponibles dans le but de guider les solutions obtenues vers une solution optimale [75]. Elles permettent d'obtenir des solutions de bonne qualité dans un laps de temps raisonnable. Alors qu'une heuristique est la plupart du temps définie pour un problème donné, une métaheuristique est une heuristique générique pouvant être décrite de façon abstraite et adaptée à différents types de problèmes avec peu de modifications. Plusieurs métaheuristiques sont basées sur des principes inspirés par des systèmes naturels qui se retrouvent dans divers domaines, tels que la biologie (algorithmes évolutionnaires et génétiques), la physique (recuit simulé) ou l'éthologie (algorithmes de colonies de fourmis). D'autres métaheuristiques, telles que la méthode tabou, ne s'inspirent d'aucun phénomène naturel.

Le choix de l'une ou l'autre de ces méthodes de résolution peut être basé sur le type de problème à résoudre, le temps alloué à la recherche de solution et/ou le niveau de qualité de la solution recherché. Les méthodes exactes et les heuristiques peuvent de plus être combinées pour tirer parti des points forts de chacune d'entre elles afin de fournir des solutions à la fois performantes du point de vue de leur qualité (liée à l'objectif) et du temps d'exécution [83].

2.1.4 Ordonnancement de la production et maintenance

Même si les méthodes de résolution présentées précédemment peuvent être très performantes, leur capacité à atteindre l'objectif du problème d'optimisation considéré est limitée par les différentes hypothèses et contraintes prises en compte. Dans un problème d'ordonnancement de la production, ce sont la plupart du temps les ressources, notamment les machines de production, qui sont limitantes. Ces machines étant soumises à des phénomènes d'usure au cours de leur utilisation, leur disponibilité n'est pas infinie et des opérations de maintenance sont nécessaires [70]. De nombreux travaux ont traité le problème de l'ordonnancement de la maintenance en développant différentes politiques de plus en plus efficaces pour gérer l'état de santé des machines. Ces politiques de maintenance sont développées dans la partie 1.1.1 du chapitre 1. Les opérations de maintenance sont par nature intrinsèquement liées à la production et leur placement dans le programme de production n'est pas trivial. Le lancement des tâches de production a en effet un impact sur l'usure des machines, qui régit le lancement des opérations de maintenance. Ce lien entre les deux activités n'est pas toujours pris en compte dans l'optimisation de l'ordonnancement de la maintenance et cette dernière est la plupart du temps subie par la production, qui reste prioritaire.

On constate alors l'émergence d'un problème qui peut être formulé de la manière suivante : la maintenance est nécessaire pour garantir une certaine disponibilité des machines et donc un bon déroulement de la production, mais va à l'encontre des objectifs d'efficacité et de rentabilité de

la production, qui génère les profits. Différentes approches intégrant à la fois l'ordonnancement de la production et l'ordonnancement de la maintenance ont été développées pour satisfaire les objectifs des deux domaines. Un état de l'art a été proposé par Pandey et al. dans [76], où ils montrent que le problème visant à combiner les deux ordonnancements a été traité de deux manières différentes. Une première stratégie séquentielle consiste à ordonnancer les tâches de production, puis à intégrer les tâches de maintenance. L'ordre d'exécution des tâches de production défini par le premier ordonnancement constitue dans ce cas une contrainte forte pour l'ordonnancement de la maintenance. La seconde stratégie tend à optimiser les deux ordonnancements en intégrant les décisions de maintenance dans l'ordonnancement de la production. Il s'agit de l'ordonnancement conjoint de la production et de la maintenance, qui définit de façon simultanée les tâches de production et de maintenance. Une fonction objectif prenant en compte des critères liés à la fois à la production et à la maintenance est considérée pour optimiser l'ordonnancement des deux activités de manière conjointe.

Afin d'aller plus loin dans l'optimisation de la production et de la maintenance, plusieurs travaux ont proposé d'utiliser les machines de production d'une certaine manière pour agir sur leur disponibilité [91, 49, 101, 68]. Différents concepts pouvant être regroupés sous le terme de reconfiguration ont été proposés dans la littérature pour rendre les ressources plus flexibles et améliorer, sous certaines conditions, la satisfaction de la fonction objectif prise en compte. La notion de reconfiguration est présentée dans la suite dans un contexte d'ordonnancement de la production ou de missions.

2.2 La notion de reconfiguration dans l'ordonnancement

La possibilité de modifier les performances des systèmes afin d'optimiser leur utilisation a été introduite dans le chapitre précédent dans un contexte *PHM*. La même notion de reconfiguration se retrouve dans différents travaux traitant d'ordonnancement de manière générale. L'adaptabilité amenée par cette notion de reconfiguration est utilisée pour optimiser la performance globale d'un ensemble de machines, en prenant en compte des délais de maintenance fixés par la disponibilité des moyens de maintenance ou par des règles d'optimisation de l'ordonnancement de la maintenance. Il s'agit alors de contrôler de façon concertée et coordonnée les performances de chaque composant du système global pour éviter les temps morts dus à des délais de maintenance trop élevés. Cela permet de manière générale d'accroître la performance globale de l'ensemble des machines considérées.

Différents types de reconfiguration affectant le rendement du système considéré sont détaillés ici. Dans le premier cas, la reconfiguration est rendue nécessaire par une dégradation de l'état de santé des machines. Il s'agit alors d'adapter l'utilisation des machines pour assurer la mission au mieux. Ce type de reconfiguration subie est illustré dans la suite par les systèmes tolérants aux fautes. La reconfiguration peut aussi prendre la forme d'une modification délibérée du fonctionnement des machines, qui correspond le plus souvent à une dégradation des performances par rapport aux performances optimales. Ce type de reconfiguration peut avoir deux objectifs : ralentir les effets de l'usure pour allonger le temps de disponibilité des machines, ou bien diminuer la consommation énergétique. Différents exemples d'ordonnancement à vitesse variable sont développés dans la suite pour expliciter cette notion de reconfiguration contrôlée.

2.2.1 Systèmes tolérants aux fautes

Un système tolérant aux fautes est équipé d'un système de contrôle qui permet le bon fonctionnement et l'achèvement des tâches allouées au système, même après l'apparition d'une faute [96]. Le système de contrôle a pour fonctions la détection et l'isolement des fautes, dans le but de minimiser les effets indésirables de ces fautes. Le développement de la théorie autour de ce type de systèmes tire sa source dans le besoin de garantir une certaine fiabilité pour des systèmes pour lesquels la sécurité est critique, tels que par exemple les plates-formes chimiques ou nucléaires, les navettes spatiales ou les aéronefs à commande de vol électrique [77]. Avec la même idée de fiabilité, la principale application motivant l'étude de systèmes tolérant les fautes est la conception de systèmes aéronautiques, rendue complexe par des problèmes de contrôle [77]. Le but principal est dans ce cas de fournir aux avions une certaine capacité d'auto-réparation, pouvant permettre aux pilotes d'atterrir avec plus de sécurité en cas de faute sérieuse. D'autres applications ont bénéficié des avancées dans le contrôle tolérant aux fautes, cette fois-ci non pas pour des raisons de sécurité, mais pour améliorer la qualité du service rendu par le système concerné. On peut citer comme exemples un système électrique de traction ferroviaire [12], un système satellite permettant de mesurer le champ magnétique de la Terre [15], un système de propulsion de bateau [53], un actionneur électro-hydraulique d'une gouverne de direction d'avion [39] ou encore une chaudière à condensation [95].

L'approche de tolérance aux fautes ne devient intéressante que si le système étudié est capable de garder un niveau de performance acceptable pour l'application, ou s'il se dégrade lentement et de façon contrôlable. Ce comportement est évoqué dans la littérature sous le nom de dégradation contrôlée [91]. Ce type de comportement se retrouve dans divers domaines, incluant le traitement d'images, la télécommunication ou ceux utilisant des processeurs à mémoire partagée et des systèmes à mémoire multi-modulaires. La dégradation contrôlée peut ainsi s'appliquer à différents éléments tels que par exemple la qualité d'image ou de son, la puissance de calcul, le débit d'accès à la mémoire, etc [91].

Le but de la dégradation contrôlée est de permettre la poursuite de l'exécution des tâches du système en empêchant les erreurs de causer des dommages au système. En cas de faute avérée, différentes techniques permettent de restaurer le système à un état sans faute. La plupart de ces techniques utilisent une certaine forme de redondance, comme par exemple la réplication de données, de disques, d'unités de mémoire, de liens réseaux, de services, processus ou composants logiciels. Ces différents types de réplicat permettent de réparer les erreurs du système en restaurant ce dernier à un état antérieur exempt de faute. La dégradation contrôlée apporte une autre réponse pour la gestion des erreurs, sans faire usage de réplicats, mais en isolant la ou les parties du système affectées par une ou plusieurs erreurs, permettant ainsi au reste du système de continuer son exécution [91]. Cette technique appartient à la catégorie des techniques de tolérance aux fautes, qui ont pour but la réduction de l'impact des erreurs dans l'exécution d'une application.

Une classification des dégradations contrôlées a été proposée par Saridakis [91]. Trois types différents de dégradation ont été définies en fonction des caractéristiques et de la structure du système impacté par une faute. Chacun de ces trois types entraîne une gestion différente de la faute, impactant plus ou moins l'exécution de la tâche globale allouée au système considéré. La dégradation optimiste correspond tout d'abord à une dégradation entraînant des coûts d'exé-

cution faibles et s'appliquant à une partie du système ayant peu de dépendance avec d'autres parties. Une erreur a dans ce cas peu de chances de se propager et d'affecter le reste du système. La dégradation pessimiste, au contraire, s'applique à des systèmes présentant un grand nombre de dépendances entre leurs constituants et pour lesquels une erreur sur un des composants a de fortes chances de se propager aux autres composants liés par une relation de dépendance. Le troisième type de dégradation contrôlée, la dégradation causale, est un cas particulier de la dégradation pessimiste, dans lequel une erreur peut affecter plusieurs composants de système. La propagation de la dégradation se limite toutefois dans ce cas aux composants liés par une relation de dépendance forte.

Dans le domaine automobile, les besoins croissants en sécurité et en confort ont entraîné une augmentation significative des unités électroniques de contrôle. La plus grande complexité des systèmes qui en découle conduit à une plus grande probabilité d'erreurs. Une gestion de ces erreurs devient alors nécessaire. Dans ce contexte, le concept de dégradation contrôlée peut aider à préserver la plus grande partie des fonctions d'un véhicule en autorisant la dégradation d'un nombre limité de fonctionnalités. Holzknecht et al. [49] ont par exemple étudié la gestion de la dégradation d'un système de détection laser utilisé en aide à la conduite pour la détection des obstacles. Plusieurs niveaux de dégradation ont été pris en compte pour différents paramètres tels que la fréquence des relevés de données, la résolution angulaire, l'angle d'acquisition des données ou encore la fréquence d'actualisation des images transmises. Différentes combinaisons de ces paramétrages ont été testés pour la détection d'un piéton. Ils ont montré que la dégradation des performances du laser rend la détection moins fiable et moins précise, mais permet de réduire la puissance utilisée et donc de minimiser les besoins en énergie, permettant de conserver cette dernière pour des fonctions plus critiques du véhicule.

2.2.2 L'ordonnancement à vitesse variable

L'ordonnancement à vitesse variable est une généralisation du problème classique d'ordonnancement sur machines multiples. Ce type d'ordonnancement gère non seulement l'affectation de ressources aux différentes tâches de production, mais aussi les temps d'exécution des tâches [101]. Le problème consiste à déterminer la meilleure séquence de tâches et les vitesses de fonctionnement des machines de manière à ce qu'une fonction de coût soit minimisée [65]. Ce type d'ordonnancement requiert une certaine flexibilité des machines, qui doivent pouvoir adapter leur vitesse de fonctionnement aux tâches qui leur sont allouées. Cette problématique a été étudiée dans des domaines d'application différents.

Un premier exemple classique a pour cadre d'application la fabrication de pièces par des machines-outils. Dietl et al. [36] ont ainsi proposé une stratégie d'utilisation des machines-outils appartenant à une même ligne de production permettant d'optimiser le remplacement des outils de coupe en regroupant un maximum d'opérations de maintenance. Pour cela, ils ont proposé d'ajuster le temps de fonctionnement avant défaillance de chaque outil afin de faire concorder les instants auxquels chacun d'eux doit être remplacé. Cet ajustement peut être obtenu en diminuant la vitesse d'utilisation des outils de coupe, ce qui permet de diminuer la vitesse d'usure et donc d'augmenter la durée d'utilisation disponible. Une diminution de la vitesse d'exécution des tâches de production est ainsi acceptée pour faire coïncider les instants auxquels l'usure de chaque outil conduirait à la panne, ce qui permet de grouper les opérations de maintenance et d'optimiser

les coûts associés. La réduction de la vitesse de coupe augmente le temps d'usinage des pièces, mais permet généralement d'augmenter le nombre de pièces pouvant être usinées avec le même outil. Il s'agit alors de trouver un compromis entre le débit de production, le nombre de pièces usinées par chaque outil et le lancement des opérations de maintenance. Le problème visant à déterminer des vitesses de coupe pour minimiser les coûts de production tout en assurant un certain débit de production et en respectant des contraintes liées à l'application a par exemple été étudié par Levin et al. [68] dans le cas de la production en ligne en grandes séries. Un programme mathématique permettant de définir une configuration optimale des outils de coupe a été proposé, ainsi qu'une méthode de résolution basée sur une décomposition du problème en sous-problèmes d'optimisation convexes et linéaires plus simples.

L'adaptation des performances de machines utilisées en parallèle se retrouve aussi dans le domaine du calcul scientifique et plus précisément pour l'ordonnancement de tâches sur des systèmes multiprocesseurs. Le lancement de plusieurs tâches en parallèle sur différents processeurs peut mener à un déséquilibre dans les temps d'exécution. En conséquence, des temps morts, correspondant à des périodes d'inactivité, peuvent apparaître sur certains processeurs. L'échelonnage de la fréquence a été proposé pour adapter les temps d'exécution les plus courts dans le but de les faire coïncider avec le temps d'exécution le plus long. Une réduction de la fréquence a en effet pour conséquence l'allongement du temps d'exécution d'une tâche. Des processeurs offrant la possibilité d'un échelonnage dynamique de la fréquence et de la tension ont été développés pour permettre d'adapter les temps d'exécution des différentes tâches dans les cas où la charge de calcul n'est pas parfaitement équilibrée [93, 60]. Cela permet de réduire la consommation énergétique sans induire une pénalisation des performances telle que l'accroissement des temps d'exécution [31].

Dans ce contexte, Wang et al. [104] ont proposé différentes heuristiques définissant un ordonnancement de tâches parallèles qui réduit la consommation énergétique en utilisant le principe de l'échelonnage dynamique de la tension et de la fréquence, sans augmenter le temps d'exécution global. Des réductions de consommation d'énergie importantes sont obtenues en réduisant les performances des processeurs au cours de certaines plages de temps correspondant à des périodes d'inactivité ou à des phases de communication. Semeraro et al. [93] ont aussi utilisé l'échelonnage de la fréquence et de la tension pour limiter les besoins en énergie, mais en adaptant les temps d'exécution des tâches n'étant pas sur le chemin critique de l'application parallèle. Ces tâches sont alors lancées à une fréquence plus faible dans le but d'allonger leur temps d'exécution et minimiser les différences avec le temps d'exécution le plus long.

2.3 Synthèse

Une définition générale de l'ordonnancement a été fournie dans ce second chapitre, amenant des précisions sur les configurations de machines et les objectifs principalement étudiés dans la littérature. Quelque soit le système à ordonnancer, les trois objectifs principaux traités sont l'utilisation efficace des ressources, la minimisation du délai d'exécution de l'ensemble des tâches et le respect des délais de production. Plusieurs méthodes de résolution permettant de définir des ordonnancements respectant à la fois les contraintes et les objectifs des problèmes traités ont été présentées.

La notion de reconfiguration des machines a ensuite été introduite. Nous avons montré que des travaux portent sur la reconfiguration afin de permettre, sous certaines conditions, de préserver une partie des performances d'un système alors même qu'il est soumis à des défaillances. Lorsque des machines sont utilisées simultanément dans un système au sein duquel une machine constitue un goulot d'étranglement, il a ensuite été montré qu'il est intéressant d'adapter les performances de toutes les machines à la performance de la machine limitante, que ce soit pour des raisons énergétiques ou pour limiter l'usure des machines et ainsi optimiser l'ordonnancement de la maintenance. Les modifications faites correspondent le plus souvent à une dégradation des performances des machines par rapport aux performances optimales. Deux objectifs principaux peuvent être distingués dans le cas où la dégradation des performances est maîtrisée. Il s'agit dans le premier cas de ralentir les effets de l'usure et de rallonger le temps de disponibilité des machines. Dans le second cas, l'objectif est de diminuer la consommation énergétique lorsque la reconfiguration de certaines machines n'a pas d'impact sur la productivité du système global.

Dans les différents exemples de reconfiguration des machines développés dans ce chapitre, on ressent le besoin de contrôler la dégradation des performances, ou au moins de connaître le niveau de dégradation et prédire son état futur, par le biais notamment de la vitesse de dégradation. Les informations nécessaires peuvent être apportées par les différentes étapes du *PHM*, qui tendent à étudier les phénomènes de dégradation et à caractériser l'état de santé des machines. Les travaux proposés dans ce manuscrit visent ainsi à définir un ordonnancement original de machines de production, exploitant à la fois la souplesse d'utilisation apportée par la considération de plusieurs profils de fonctionnement et la connaissance de la durée d'utilisation résiduelle avant maintenance apportée par le pronostic.

Chapitre 3

Problème d'optimisation et modèles de profils de fonctionnement

Sommaire

Le problème d'optimisation considéré s'inscrit dans la décision post-pronostic, dans un certain contexte de production au sein duquel des machines indépendantes sont utilisées en parallèle pour fournir une certaine production globale. La quantité de travail fournie à chaque instant par l'ensemble des machines doit permettre de satisfaire une demande connue. Conformément à la définition de l'ordonnancement donnée dans le chapitre précédent, le but est déterminer quelles machines doivent être utilisées à chaque instant et de quelle manière, avec pour objectif la maximisation de l'horizon de production.

Ce chapitre pose les différentes hypothèses sur lesquelles sont basés les développements proposés dans les chapitres suivants. Le problème d'optimisation traité tout au long du manuscrit est tout d'abord détaillé dans la première partie pour le cadre d'application visé. Dans cette même partie est fournie une définition générale des profils de fonctionnement, dont la notion a été introduite dans les chapitres précédents et sur lesquels repose l'originalité des travaux de recherche. Une modélisation plus détaillée est ensuite proposée dans les parties suivantes pour chacun des deux types de profils de fonctionnement considérés. Le premier type est basé sur une variation discrète des performances, tandis que le second définit une évolution continue des performances des machines.

3.1 Énoncé du problème

Cette première partie pose tout d'abord le cadre applicatif de l'étude. Les hypothèses générales du problème sont détaillées, ainsi que les contraintes à respecter. L'objectif associé au problème général d'ordonnancement considéré est finalement présenté.

3.1.1 Cadre d'application

L'application cible consiste en une plate-forme composée d'un ensemble $\mathscr{M}$ de m machines M_j tel que $M_j \in \mathscr{M}$, avec $1 \leqslant j \leqslant m$. Les machines sont supposées effectuer le même type de tâche, de manière indépendante et peuvent être utilisées en parallèle. Elles forment alors un système global qui doit fournir un certain service $\sigma(t)$ tout au long d'une période donnée que nous nommons horizon. Cet horizon, noté $\mathcal{H}$, correspond à la durée la plus grande pendant laquelle le service peut être assuré. Ce service peut être représenté par un flux, c'est-à-dire une quantité produite par unité de temps. Selon l'application, cette quantité peut correspondre à un nombre de pièces fabriquées, une quantité d'énergie générée, une puissance instantanée, ou encore une quantité d'informations fournie.

3.1.2 Hypothèses et contraintes du problème

Plusieurs hypothèses générales sont tout d'abord énoncées pour délimiter l'ensemble des problèmes d'optimisation traité dans ce manuscrit.

H1 À chaque instant t, la production fournie par la plate-forme ($\rho_{tot}(t)$) correspond à la somme des contributions de chaque machine en fonctionnement à l'instant t ;

H2 Cette production totale doit au minimum atteindre la demande : $\rho_{tot}(t) \geqslant \sigma(t)\ \forall\, t \leqslant \mathcal{H}$;

H3 Pour tout instant t de l'horizon de production, la demande $\sigma(t)$ est atteignable. En d'autres termes, considérant chaque machine dans son état de santé initial (au temps $t = 0$), la

plate-forme peut fournir la valeur maximale atteinte par la demande au cours de l'horizon ;

H4 Toutes les machines ne sont pas nécessairement utilisées à chaque instant si un sous-ensemble suffit à atteindre la demande. Certaines machines peuvent aussi ne pas être disponibles à partir d'un certain temps si elles ont atteint leur fin de vie avant maintenance ;

H5 Aucune dégradation n'est considérée lorsqu'une machine n'est pas utilisée. Dans ce cas, son temps résiduel d'utilisation avant maintenance (RUL, pour Remaining Useful Life) reste constant tant qu'elle n'est pas utilisée ;

H6 Le problème de l'approvisionnement des machines n'est pas traité. Toutes les entrées nécessaires à la production (entre autres, matières premières et énergie) sont disponibles durant tout l'horizon de production ;

H7 Aucun moyen de stockage n'étant considéré au sein de la plate-forme, le stockage de la production n'est pas possible et toute surproduction est perdue.

Une utilisation particulière des machines est ensuite prise en compte. Il a été montré dans les chapitres précédents que les performances d'une machine peuvent varier dans le temps. Ces variations peuvent être subies quand elles sont dues à l'usure, mais elles peuvent aussi être sciemment voulues et contrôlées. Nous proposons d'exploiter cette possibilité afin d'optimiser l'utilisation de la plate-forme, sachant que l'objectif est de maximiser l'horizon de production. Chaque machine considérée peut fournir différents niveaux de performance, ou débits. Chacun de ces niveaux de performance est connu et correspond à des conditions opératoires bien spécifiques. Dans un contexte PHM, chaque machine est de plus monitorée et associée à un module de pronostic, capable de fournir une durée résiduelle d'utilisation avant maintenance (RUL) pour chaque débit disponible, en fonction de l'utilisation passée et future de la machine.

Un profil de fonctionnement est alors défini par un niveau de performance contrôlé associé à une durée résiduelle de fonctionnement avant maintenance. Même si la prise en compte de conditions opératoires variables au cours de l'utilisation d'une machine n'est pour l'instant pas totalement maîtrisée pour l'estimation des valeurs de RUL [42], nous faisons ici l'hypothèse que le RUL associé à chaque profil d'utilisation est connu pour chaque machine. La durée d'utilisation résiduelle avant maintenance de chaque machine évolue de plus en fonction des conditions opératoires, c'est-à-dire en fonction des profils de fonctionnement choisis au cours du temps. Différents modèles ont été considérés pour définir le lien entre les différents débits et les valeurs de RUL. Ces modèles sont décrits dans les parties suivantes de ce chapitre.

Quelques contraintes concernant l'utilisation des machines doivent finalement être respectées.

C1 L'utilisation d'une machine pour un temps dépassant le RUL n'est pas autorisée. Cela permet d'éviter des pannes, qui engendreraient des maintenances lourdes, potentiellement plus longues et coûteuses que des opérations de maintenance plus classiques ;

C2 Le stockage de la production n'étant pas géré, la surproduction est évitée autant que possible. Une surproduction entraîne en effet des coûts, liés au stockage ou à des pertes de production si aucun stockage n'est possible. Elle est toutefois autorisée si elle permet d'allonger l'horizon de production de la plate-forme.

3.1.3 Objectif

L'objectif consiste à maximiser la durée d'utilisation d'une plate-forme de machines dans le cadre décrit plus tôt. Pour cela, il s'agit de configurer la plate-forme en tirant parti des

différents profils de fonctionnement disponibles pour chaque machine, de façon à ce que le débit total fourni, $\rho_{tot}(t)$, atteigne la demande $\sigma(t)$ le plus longtemps possible.

Ce problème d'optimisation peut être traité en discrétisant le temps en périodes ΔT. Cette approche est tout à fait cohérente avec une situation réelle de gestion de production. Une période de temps peut en effet représenter une heure, un jour ou une semaine de production. L'horizon de production $\mathcal{H}$ est exprimé sous la forme $\mathcal{H} = K\Delta T$, avec ΔT la longueur d'une période de temps et K le nombre de périodes pour lequel la demande $\sigma(t)$ est satisfaite. Cette demande peut être constante durant tout l'horizon de production et est alors notée σ. Si elle varie au cours du temps, elle est notée σ_k, avec σ_k une valeur constante pour tout t de la période k tel que $1 \leqslant k \leqslant K$ et $(k-1)\Delta T \leqslant t \leqslant k\Delta T$. La longueur des périodes est choisie très petite devant la durée de vie des machines ($\Delta T \ll RUL$).

Par soucis de simplification et sans perte de généralité, la longueur d'une période de temps est fixée à l'unité dans laquelle sont exprimées les valeurs de RUL. Par exemple, si ces dernières sont exprimées en heures, une unité de temps ΔT correspond à une heure. Les différents développements et résultats proposés dans la suite sont facilement adaptables pour des données exprimées en unités différentes.

Considérant cette discrétisation du temps, le problème consiste alors à choisir, pour chaque période de temps k, un sous-ensemble de machines à utiliser pour atteindre la demande σ_k et un profil de fonctionnement pour chaque machine sélectionnée. Dans ce cadre, deux types de variation des performances sont proposées. Le premier considère un nombre discret de débits disponibles pour chaque machine. Pour le second type, les débits disponibles varient de façon continue entre une borne inférieure et une borne supérieure.

3.2 Profils de fonctionnement avec variation discrète des performances

Un premier type de profils de fonctionnement est considéré avec une variation discrète du débit fourni entre une valeur minimale et une valeur maximale. Les éléments développés dans la suite sont définis de manière générale pour des machines de tous types. Le modèle proposé pourrait par exemple s'appliquer à des éoliennes ou des machines-outils.

Une modélisation est proposée pour représenter l'évolution de la disponibilité des profils de fonctionnement au cours de l'utilisation des machines. Le problème à résoudre est ensuite formellement posé. L'approche est finalement motivée par un exemple d'application montrant plusieurs utilisations possibles d'une plate-forme de machines à deux profils de fonctionnement.

3.2.1 Modélisation

Chaque machine M_j $(1 \leqslant j \leqslant m)$ possède un nombre fini de niveaux de performance différents. On pose n le nombre de profils de fonctionnement $N_{i,j}$, définis pour chaque machine de la manière suivante : $N_{i,j} = (\rho_{i,j}, RUL_{i,j})$, avec $0 \leqslant i \leqslant n-1$ et $1 \leqslant j \leqslant m$. Soit $N_{0,j}$ le profil nominal de la machine M_j, fournissant le débit maximal. $N_{0,j}$ est associé au RUL minimum. En comparaison, un profil sous-nominal fournit un débit plus faible, mais pendant un temps plus long (voir figure 3.1). Suivant cette logique, on a les relations suivantes : $\rho_{0,j} > \rho_{1,j} > \ldots > \rho_{n-1,j}$

et $RUL_{0,j}(k) < RUL_{1,j}(k) < \ldots < RUL_{n-1,j}(k)$ pour toute période k $(1 \leqslant k \leqslant K)$.

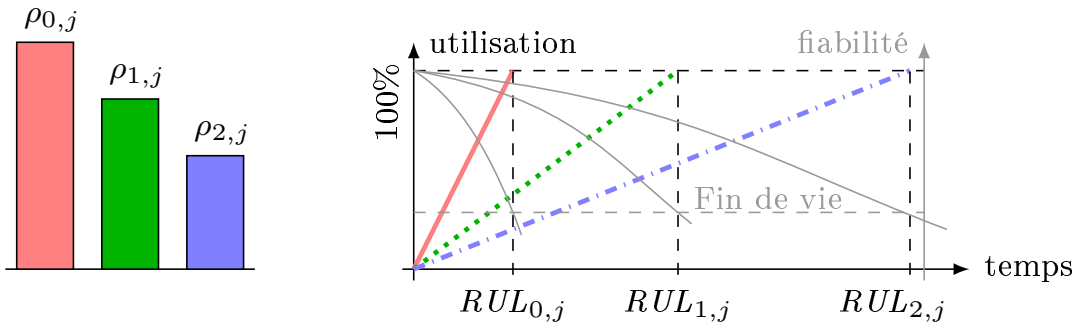


Figure 3.1 – Profils de fonctionnement avec variation discrète des performances pour une machine M_j

Chaque profil de fonctionnement a un impact plus ou moins prononcé sur l'usure d'une machine et agit donc différemment sur son temps de fonctionnement restant. La prise en compte de plusieurs profils de fonctionnement semble ainsi intéressant étant donné que l'utilisation combinée de deux profils ou plus peut permettre d'atteindre un temps de fonctionnement plus grand qu'en utilisant uniquement le profil nominal.

Dans le modèle proposé, l'ordre dans lequel les profils de fonctionnement sont sélectionnés au cours de l'utilisation d'une machine n'a aucun impact sur l'évolution de son état de santé, et donc sur l'évolution des RUL associés à chaque profil. Sans tenir compte des niveaux de performance atteints, les exemples de scénario proposés sur la figure 3.2 montrent qu'il est possible de dépasser le RUL associé au profil nominal, $RUL_{0,j}$, en utilisant une machine avec trois profils de fonctionnement différents successifs, $N_{0,j}$, $N_{1,j}$ et $N_{2,j}$. Le second scénario présenté sur cette figure montre qu'une sélection appropriée des profils $N_{i,j}$ au cours du temps peut permettre d'étendre la durée d'utilisation d'une machine, non seulement au-delà du RUL associé au profil nominal, $RUL_{0,j}$, mais aussi au-delà du RUL associé à un profil sous-nominal ($RUL_{1,j}$ pour l'exemple considéré).

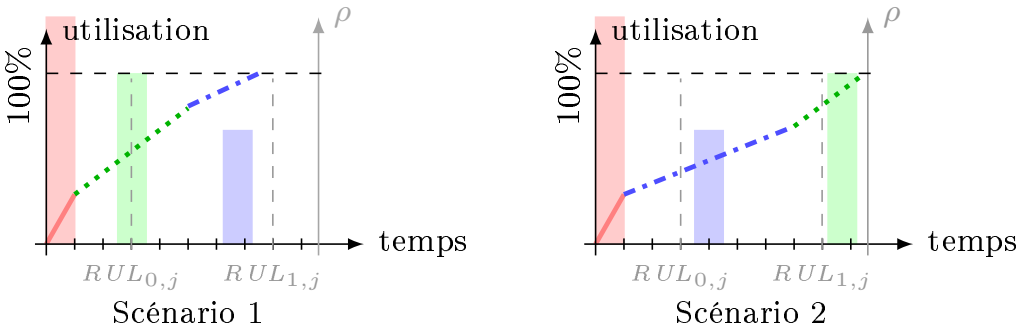


Figure 3.2 – Utilisation d'une machine M_j avec plusieurs profils de fonctionnement considérant une variation discrète du débit fourni

Cet allongement de la durée d'utilisation, rendu possible par l'utilisation de plusieurs profils de fonctionnement, peut être exploité sur plusieurs machines travaillant en parallèle et contribuant toutes à la même demande, à condition que cette demande puisse être satisfaite avec un fonctionnement sous-nominal des machines. La considération de plusieurs profils de fonctionnement permet donc de jouer sur l'ordonnancement d'un ensemble de machines, si l'on considère la

maximisation de l'horizon de production comme objectif. Un exemple mettant en jeu plusieurs machines est proposé dans la suite pour illustrer cette propriété.

3.2.2 Problème d'optimisation maxK

En utilisant les notations définies dans l'énoncé du problème, chaque variante du problème d'optimisation considérant une variation discrète des performances peut être décrite par un cas particulier de la notation suivante : $\text{MAXK}(\sigma_k \,|\, \sigma,\, \rho_{i,j} \,|\, \rho_j \,|\, \rho,\, RUL_{i,j} \,|\, RUL_j)$.

Cette notation générale représente le problème d'optimisation consistant à définir un ordonnancement des machines qui maximise l'horizon de production $\mathcal{H} = K\Delta T$. Le niveau de performances requis, ou demande, est noté σ_k si la demande varie dans le temps. La demande est notée σ si elle reste constante tout au long de l'horizon de production. Les deux derniers paramètres caractérisent les propriétés associées à l'ensemble des machines de $\mathscr{M}$. Chaque machine $M_j \in \mathscr{M}$ $(1 \leqslant j \leqslant m)$ est en effet associée à n profils de fonctionnement $N_{i,j}$ $(0 \leqslant i \leqslant n-1)$, qui fixent les valeurs $\rho_{i,j}$ et les durées d'utilisation associées, $RUL_{i,j}$. Dans le cas où un seul profil de fonctionnement est considéré pour chaque machine $(n = 1)$, l'ensemble de machines peut être soit homogène en termes de performances, soit hétérogène. Dans le premier cas, on a $\rho_{i,j} = \rho$ et dans le second, $\rho_{i,j} = \rho_j$ $(0 \leqslant i \leqslant n-1$ et $1 \leqslant j \leqslant m)$. Les machines pouvant présenter différents états de santé au début de l'ordonnancement, les valeurs de RUL peuvent être différentes quelle que soit la variante du problème d'optimisation considérée. Si un seul profil de fonctionnement est considéré, les valeurs de RUL sont donc toujours notées $RUL_j(k)$ avec k la période couvrant les instants t entre les dates $(k-1)\Delta T$ et $k\Delta T$ et $1 \leqslant k \leqslant K$. Cela permet aussi de représenter les différents états par lesquels passe une machine au cours du processus d'ordonnancement, sachant que celle-ci n'est pas systématiquement choisie.

Par soucis de simplification, on note dans la suite $RUL_{i,j}$ la valeur de RUL correspondant au profil $N_{i,j}$ de la machine M_j au début de l'ordonnancement $(k = 1)$ et RUL_j la valeur de RUL correspondant au profil $N_{0,j}$ de la machine M_j pour la même période, lorsque cette machine n'admet qu'un seul profil.

3.2.3 Exemple d'application

Afin d'illustrer la pertinence de l'approche, un exemple d'application est proposé pour montrer qu'utiliser une machine avec son profil le plus performant n'est pas toujours la meilleure solution quand on cherche à prolonger la durée de production tout en respectant un niveau de service requis, et ce, même si l'efficacité des machines décroît dans des profils de fonctionnement dans lesquels la durée de vie est plus grande mais les performances plus modestes. Pour cet exemple, le rendement $Q_{i,j} = \rho_{i,j} \times RUL_{i,j}$ $(1 \leqslant j \leqslant m$ et $0 \leqslant i \leqslant n-1)$ associé à chaque profil de chaque machine est supposé être maximum pour le profil nominal et décroître lorsque le débit décroît. On a alors les relations suivantes : $Q_{0,j} > Q_{1,j} > \ldots > Q_{i,j} > \ldots > Q_{n-1,j}$, soit $\rho_{i,j} \times RUL_{i,j} > \rho_{i+1,j} \times RUL_{i+1,j}$ $\forall 1 \leqslant j \leqslant m$ et $\forall 0 \leqslant i \leqslant n-2$.

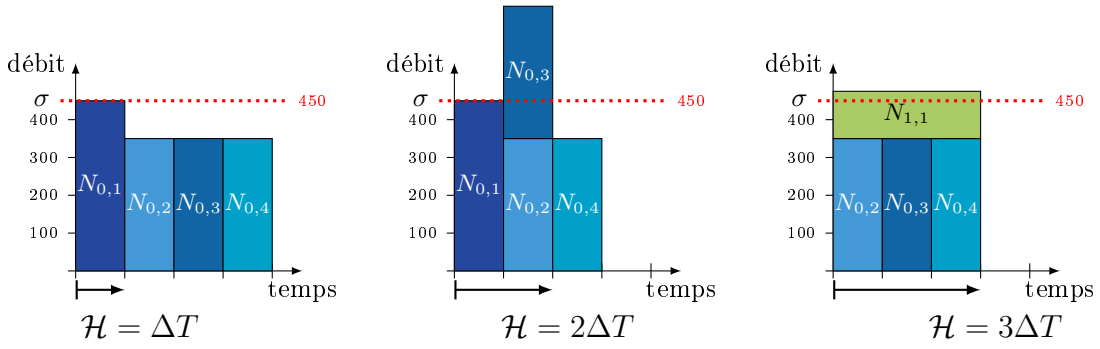
Soit un ensemble $\mathscr{M}$ de quatre machines $(\mathscr{M} = \{M_1, M_2, M_3, M_4\})$ avec lequel on souhaite produire un certain débit. Ce débit total doit au moins atteindre à chaque instant t une demande constante $\sigma = 450$ unités de débit. L'objectif est de maximiser le temps pendant lequel ce niveau de service est atteint. Au temps $t = 0$, M_1 est supposée capable de fournir deux débits

différents, correspondant à deux profils de fonctionnement : $\rho_{0,1} = 450$ pendant une période de temps ($RUL_{0,1}(1) = 1$) ou $\rho_{1,1} = 125$ pendant trois périodes ($RUL_{1,1}(1) = 3$). A $t = 0$, les autres machines M_j ($j = 2, 3, 4$) peuvent produire soit $\rho_{0,j} = 350$ pendant une période de temps ($RUL_{0,j}(1) = 1$), soit $\rho_{1,j} = 75$ pendant trois périodes ($RUL_{1,j}(1) = 3$) – voir table 3.1 –. Les caractéristiques de ces profils de fonctionnement respectent un rendement décroissant lorsque le débit diminue. Pour tout temps $t > 0$, les caractéristiques de chaque profil $N_{i,j}$ ($0 \leqslant i \leqslant n-1$ et $1 \leqslant j \leqslant m$) dépendent de l'usage passé de la machine M_j. Conformément à ces hypothèses, le problème d'optimisation considéré est le suivant : MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$).

Table 3.1 – Évolution des caractéristiques au cours du temps des profils de fonctionnement ($N_{i,j} = (\rho_{i,j}, RUL_{i,j})$) pour chaque machine M_j pour chacun des trois scénarios de la figure 3.3

		$t = 0$	$t = \Delta T$	$t = 2\Delta T$	$t = 3\Delta T$
Scénario *S1* ($\mathcal{H} = \Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(125,3)	(125,0)	(125,0)	(125,0)
	M_2	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,1)
		(75,3)	(75,3)	(75,3)	(75,3)
Scénario *S2* ($\mathcal{H} = 2\Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(125,3)	(125,0)	(125,0)	(125,0)
	M_2	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)
Scénario *S3* ($\mathcal{H} = 3\Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(125,3)	(125,2)	(125,1)	(125,0)
	M_2	(350,1)	(350,0)	(350,0)	(350,0)
		(75,3)	(75,0)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)

Trois scénarios différents d'utilisation des machines sont proposés. Dans le premier scénario (*S1*), les machines sont utilisées avec leur profil le plus efficace, $N_{0,j}$, et aucune surproduction n'est autorisée. On peut voir sur la figure 3.3a que, dans ce cas, la plate-forme peut fonctionner pendant quatre périodes de temps. La demande n'est toutefois satisfaite que pendant la première période ΔT, pendant laquelle la machine M_1 permet à elle seule d'atteindre le débit requis. Pour les trois périodes suivantes ($t > \Delta T$), le débit délivré par les machines ne permet pas de respecter la demande σ. En effet, $\rho_{0,2}, \rho_{0,3}, \rho_{0,4} < \sigma$. L'horizon de production est donc d'une période de temps ($\mathcal{H} = \Delta T$) pour le scénario *S1*, avec $K = 1$.



(a) Scénario *S1* **(b)** Scénario *S2* **(c)** Scénario *S3*

Figure 3.3 – Exemple d'utilisation de machines avec un nombre discret de profils de fonctionnement

Quand une surproduction est autorisée (scénario *S2*), deux machines peuvent être utilisées en parallèle. Cela permet d'augmenter l'horizon à deux périodes utiles (voir la figure 3.3b, avec $\mathcal{H} = 2\Delta T$). La machine M_1 est toujours utilisée pendant la première période, pour $0 \leqslant t \leqslant \Delta T$. Les machines M_2 et M_3 sont ensuite utilisées en parallèle pendant la deuxième période, pour $\Delta T \leqslant t \leqslant 2\Delta T$. Pour $t \geqslant 2\Delta T$, $RUL_{0,1} = RUL_{0,2} = RUL_{0,3} = 0$ et $\rho_{0,4} < \sigma$. Si la production est arrêtée au terme des deux périodes, la plate-forme présente un potentiel résiduel. La machine M_4 n'ayant pas été utilisée, elle ne nécessite aucune maintenance. Si les machines ne peuvent être utilisées qu'avec leur profil de fonctionnement nominal, $N_{0,j}$, l'ordonnancement proposé sur la figure 3.3b est optimal. Aucune autre combinaison des contributions des machines ne permet en effet de satisfaire la demande au delà de deux périodes.

Afin d'étendre encore l'horizon de production, il est donc nécessaire d'utiliser les machines différemment. Le troisième scénario (*S3*) illustré sur la figure 3.3c consiste à utiliser la machine M_1 avec un débit moindre, ce qui permet de l'utiliser pendant un temps plus long qu'avec son profil nominal, et par conséquent d'atteindre la demande pendant trois périodes de temps. En effet, en utilisant cette machine avec le débit $\rho_{1,1} = 125$, sa contribution peut être ajoutée à celle d'une des trois autres machines ($\rho_{0,j}$, avec $j = 2, 3, 4$) à chaque période. Après avoir été utilisées pendant une période, les machines M_2, M_3 et M_4 ont atteint leur fin de vie ($RUL_{0,2} = RUL_{0,3} = RUL_{0,4} = 0$ pour $t \geqslant \Delta T$). C'est aussi le cas pour la machine M_1 à la fin de la troisième période ($RUL_{1,1} = RUL_{0,1} = 0$ pour $t \geqslant 3\Delta T$). Le nombre de combinaisons pour le choix des machines et de leur profil de fonctionnement pour chaque période étant limité, il est facile de voir qu'aucun autre choix ne permet d'atteindre la demande σ pour un plus grand nombre de périodes. $K = 3$ pour $\mathcal{H} = 3\Delta T$ est donc une solution optimale au problème d'optimisation MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) considérant l'ensemble $\mathcal{M}$ de machines détaillé précédemment. Ce dernier scénario *S3* montre que diminuer l'efficacité d'une machine permet d'allonger la durée d'utilisation d'une plate-forme, tout en respectant le niveau de service requis.

3.3 Profils de fonctionnement avec variation continue des performances

Un ensemble discret de profils de fonctionnement ne représentant pas toujours la réalité pour toutes les machines, un second type de profils de fonctionnement est proposé, pour lequel chaque

machine M_j peut fournir des débits ρ_j variant de façon continue dans un intervalle donné. Le modèle proposé dans cette partie définit un intervalle de débits possibles pour chaque machine associe une durée d'utilisation à chacun de ces débit. Une fois le modèle des profils avec variation continue des performances détaillé, quelques propriétés particulières sont illustrées.

Dans le cadre de cette étude, une dimension asymétrique est ajoutée à ce second modèle par la prise en compte d'une décroissance du débit maximal pouvant être fourni pas les machines considérées. Contrairement au cas des profils de fonctionnement avec variation discrète des performances, le séquencement des profils de fonctionnement avec variation continue du débit a alors un impact sur l'évolution de l'état de santé général de chaque machine. Ce choix est dicté par un domaine d'application spécifique, mettant en jeu des plates-formes de piles à combustible. Ces dernières peuvent fournir un nombre infini de puissances dans un intervalle donné et leur comportement à l'usure peut être modélisé par la représentation proposée dans la suite. Cette représentation se base sur des résultats d'études des piles à combustible menées au sein de la Fédération de Recherche CNRS FCLAB (FR CNRS 3539), qui regroupe notamment l'équipe PHM du département AS2M (Automatique et Systèmes Micro-Mécatroniques) de l'institut FEMTO-ST (Besançon) et l'équipe SHARPAC (Systèmes électriques Hybrides, ActionneuRs électriques, systèmes Piles A Combustible) du département Énergie de FEMTO-ST (Belfort). Les piles à combustible ne sont probablement pas les seules machines dont le comportement peut être représenté par cette modélisation. Toute machine présentant des performances variables de façon continue entre deux bornes avec une dégradation de la performance maximale au cours de son utilisation entre plus ou moins directement dans le cadre d'application du modèle proposé ici.

3.3.1 Modélisation

La modélisation retenue des profils de fonctionnement avec variation continue des performances est proposée après une présentation des différentes notations utilisées.

3.3.1.1 Notations

Quelques notations sont tout d'abord définies pour les différentes caractéristiques associées à chaque machine M_j.

$\rho\max_j$: débit maximal pouvant être fourni par la machine M_j, supposé strictement décroissant avec le temps. Cette tendance traduit la dégradation de la performance maximale de la machine au cours de son utilisation. La vitesse de dégradation associée à cette performance maximale est supposée constante et l'évolution du débit maximal disponible au cours du temps est modélisé par une droite d'équation $\rho\max_j(t) = a_j t + b_j$, avec $a_j \leqslant 0$ la vitesse de décroissance de $\rho\max_j$ et $b_j = \rho\max_j(0)$ le débit maximal atteignable par la machine avant toute utilisation. Les deux coefficients a_j et b_j sont des caractéristiques intrinsèques de chaque machine M_j ;

$\rho\min_j$: débit minimal pouvant être fourni par la machine M_j, supposé constant et défini comme un pourcentage $X\min$ du débit maximal initial : $\rho\min_j = X\min \cdot \rho\max_j(0)$, avec $0 \leqslant X\min \leqslant 1$. Les débits inférieurs à $\rho\min_j$ causent une usure trop importante aux piles à combustible qui servent ici de cas d'application. L'utilisation des machines avec ces débits particuliers, notamment le débit nul, est donc évitée autant que possible. Pour

cela, un moyen simple est de respecter une continuité d'utilisation des machines une fois qu'elles ont été démarrées ;

ρ_j : débit fourni par la machine M_j. A chaque instant t, $\rho_j(t) \in [\rho\text{min}_j, \rho\text{max}_j(t)]$;

ρnom_j : débit recommandé pour un fonctionnement nominal de la machine M_j. Cette donnée constructeur, définie comme un pourcentage du débit maximal initial ($\rho\text{nom}_j = X\text{nom} \cdot \rho\text{max}_j(0)$, avec $0 \leqslant X\text{nom} \leqslant 1$), correspond généralement au débit pour lequel la machine a été conçue ;

ρopt_j : débit optimal, pour lequel la durée de vie de la machine est maximale, avec $\rho\text{min}_j < \rho\text{opt}_j < \rho\text{max}_j$. De même que ρmin_j et ρnom_j, le débit optimal est défini comme un pourcentage du débit maximal initial : $\rho\text{opt}_j = X\text{opt} \cdot \rho\text{max}_j(0)$, avec $0 \leqslant X\text{opt} \leqslant X\text{nom} \leqslant 1$;

$RUL_j(\rho_j)$: durée de vie associée au débit $\rho_j(t)$ pour la machine M_j, avec $\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{max}_j(t)$. En particulier, $RUL_j(\rho\text{min}_j) = RUL\text{min}_j = \min_{\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j} RUL_j(\rho_j)$ et $RUL_j(\rho\text{opt}_j) = RUL\text{opt}_j = \max_{\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{max}_j} RUL_j(\rho_j)$.

3.3.1.2 Modèle

Le modèle proposé sur la figure 3.4 représente l'évolution de la disponibilité des différents débits pouvant être fournis par une machine M_j au cours de son utilisation. Cette représentation a été directement inspirée du comportement observé et mesuré sur des piles à combustible à membrane d'échange de protons (aussi nommées piles à combustible à membrane électrolyte polymère) par l'équipe SHARPAC (Systèmes électriques Hybrides, ActionneuRs électriques, systèmes Piles A Combustible) du département Énergie de l'institut FEMTO-ST (Belfort). Ce modèle que nous proposons permet de palier un manque dans la communauté de recherche traitant des problèmes énergétiques, dans laquelle aucune modélisation répondant à nos besoins n'a été proposée pour représenter le comportement d'une pile à combustible en regard de sa durée d'utilisation résiduelle.

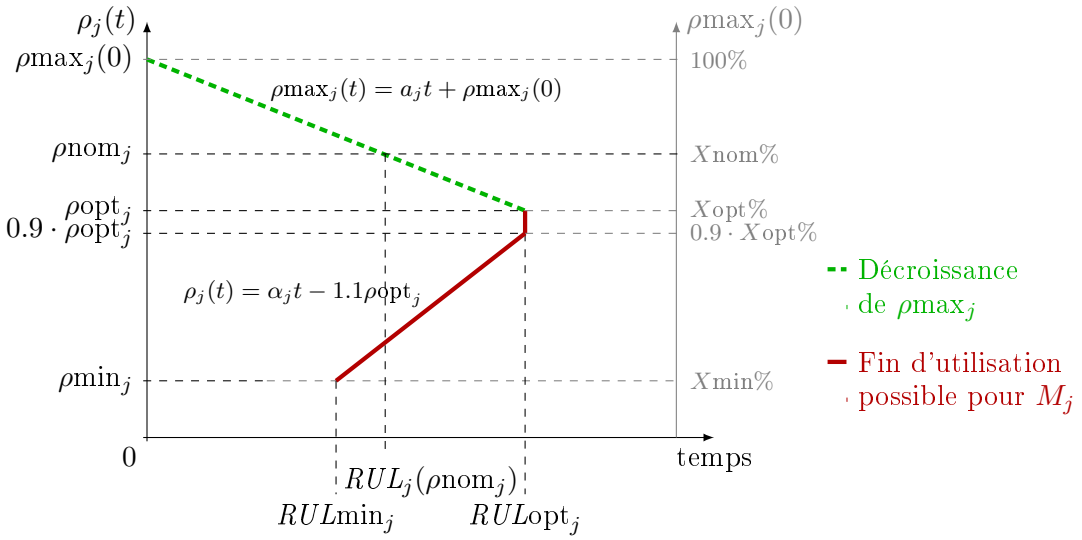


Figure 3.4 – Évolution au cours du temps des caractéristiques d'une machine M_j avec une variation continue des performances

Les pentes des différentes courbes limitant la durée de vie des débits sont fortement exagérées sur la figure 3.4 et les suivantes, mais cette représentation respecte la tendance de l'évolution de l'état de santé des machines. La modélisation proposée est composée de deux parties distinctes, associées à des comportements différents. Ces parties correspondent à des intervalles distincts de débits : $[\rho\text{opt}_j, \rho\text{max}_j(t)]$ pour la partie supérieure et $[\rho\text{min}_j, \rho\text{opt}_j]$ pour la partie inférieure.

P1 ($\rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{max}_j(t)$) : la partie supérieure, limitée sur les figures suivantes par une ligne discontinue verte, correspond à la décroissance du débit maximal au cours de l'utilisation de la machine. Atteindre cette limite signifie que le débit en cours d'utilisation ne peut plus être fourni par la machine. Les débits supérieurs ne sont plus non plus disponibles. Cela ne signifie pas pour autant que la machine a atteint sa fin d'utilisation avant maintenance. Elle peut en effet toujours être utilisée avec un débit inférieur.

P2 ($\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$) : les RUL associés aux débits dans la partie inférieure correspondent en revanche bien à une limite d'utilisation avant maintenance et atteindre la limite représentée par une ligne rouge signifie bien la fin de la durée d'utilisation avant maintenance de la machine.

Deux cas peuvent à nouveau être distingués pour la partie inférieure du modèle :

P2-1 ($0.9 \cdot \rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$) : les débits dans cet intervalle sont associés à une durée d'utilisation maximale, notée $RUL\text{opt}_j = \max_{\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{max}_j} RUL_j(\rho_j)$;

P2-2 ($\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$) : la durée résiduelle d'utilisation associée à ces débits est pénalisée par une décroissance représentée par la droite d'équation $\rho_j(t) = \alpha_j t + \beta_j$, avec α_j fonction de la durée d'utilisation maximale de la machine, $RUL\text{opt}_j$, et $\beta_j = -1.1\rho\text{opt}_j$. Cela permet d'exprimer le fait qu'un débit faible est associé à une vitesse d'usure plus grande que le débit optimal ρopt_j. Le débit minimal ρmin_j est ainsi associé à la durée d'utilisation minimale : $RUL_j(\rho\text{min}_j) = RUL\text{min}_j = \min_{\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j} RUL_j(\rho_j)$.

3.3.2 Propriétés particulières

Deux caractéristiques bien particulières découlent de la modélisation développée précédemment. La première concerne la partie supérieure du modèle, dans laquelle il existe une asymétrie dans la disponibilité des débits forts. Une équivalence des temps incluant une proportionnalité entre les débits est ensuite définie dans la partie inférieure du modèle. Cette équivalence est valable pour les débits ρ_j dans l'intervalle $[\rho\text{min}_j, \rho\text{opt}_j$, pour lesquels l'atteinte de la limite fixée par le RUL associé correspond à la fin d'utilisation de la machine M_j.

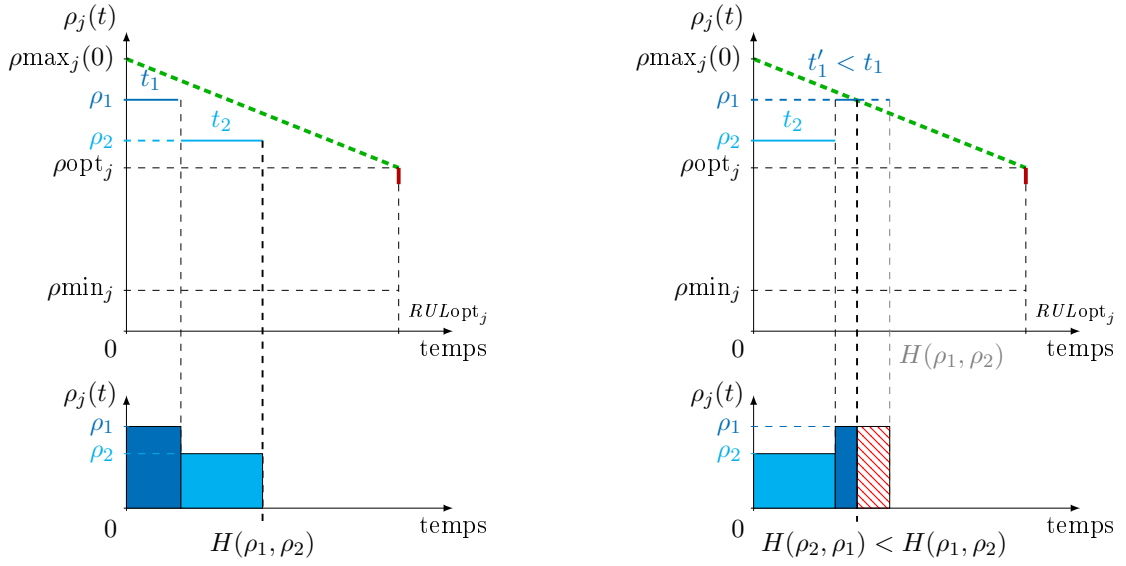
3.3.2.1 Caractère asymétrique pour $\rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{max}_j$

Un comportement asymétrique peut tout d'abord être observé pour les débits de la partie supérieure du modèle proposé : $\rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{max}_j$. Dans un ordonnancement définissant l'évolution de la contribution d'une machine au cours du temps, les périodes associées à des niveaux de performance différents ne peuvent pas toujours être permutées. La disponibilité des débits dans l'intervalle considéré dépend en effet de l'utilisation passée de la machine. Comme illustré sur la figure 3.5, la durée d'utilisation disponible d'une machine change en fonction de l'ordre dans lequel les débits avec lesquels elle est utilisée sont choisis.

Soient les deux scénarios suivants :

1. si une machine M_j est d'abord utilisée avec un débit ρ_1 pendant un temps t_1, puis avec un débit plus faible ρ_2 pendant un temps t_2, le temps d'utilisation de la machine est noté $H(\rho_1, \rho_2)$;
2. si l'ordre d'utilisation des débits est inversé, ρ_2 peut toujours être utilisé pendant le temps t_2, mais après ce temps, la durée d'utilisation résiduelle associée à ρ_1 n'est plus suffisante pour autoriser son utilisation pendant la durée t_1 (voir figure 3.5).

La durée d'utilisation de la machine est ainsi plus faible dans le deuxième cas de figure : $H(\rho_1, \rho_2) < H(\rho_2, \rho_1)$.



(a) Premier scénario : ρ_1 pendant t_1, puis ρ_2 pendant t_2

(b) Second scénario : ρ_2 pendant t_2, puis ρ_1 pendant t'_1

Figure 3.5 – Illustration du caractère asymétrique du modèle pour $\rho\text{opt}_j \leqslant \rho_j \leqslant \rho\max_j$

3.3.2.2 Équivalence des temps pour $\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$

Pour les débits dans la partie basse du modèle ($\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$), atteindre la limite représentée par les lignes rouges sur la figure 3.4 équivaut à une fin d'utilisation avant maintenance de la machine. Une équivalence des temps est définie pour exprimer le fait qu'utiliser un débit ρ_1 pendant un temps t_1 n'induit pas la même usure que lorsqu'un autre débit ρ_2 est utilisé pendant le même temps t_1.

La comparaison de l'état de santé d'une machine M_j après l'utilisation de n'importe quel débit inclus dans la partie inférieure du modèle ($\rho_j \leqslant \rho\text{opt}_j$) peut être faite sur la base d'une quantité U définie de la manière suivante : considérant une valeur initiale $U = 0$ (correspondant à une usure de la machine de 0%), si la machine est utilisée avec un débit ρ_1 durant un temps t_1, l'usure associée est $U = t_1 / RUL_j(\rho_1)$. Après une première utilisation de la machine avec ρ_1, si la même machine doit être utilisée avec un autre débit ρ_2, la quantité U permet de déterminer le temps associé à l'usure qui aurait été atteinte si la machine avait été utilisée avec ρ_2 pendant t_1. Ce temps équivalent, $t_{1,2}$, est défini par l'équation (3.1). Les temps équivalents associés à

tous les débits de l'intervalle considéré peuvent être déterminés en suivant la droite d'équation $\rho_j(t) = \alpha'_j t + \beta'_j$, avec $\alpha'_j = \alpha_j / U$ (voir figure 3.6).

$$t_{1,2} = U \times RUL_j(\rho_2) = t_1 \frac{RUL_j(\rho_2)}{RUL_j(\rho_1)} \tag{3.1}$$

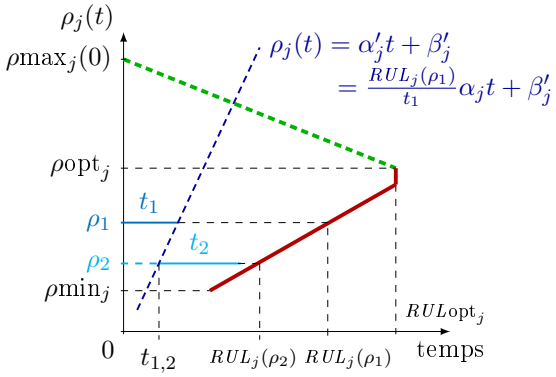


Figure 3.6 – Illustration de l'équivalence des temps pour $\rho\text{min}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$

3.4 Synthèse

Compte tenu des notations et modèles détaillés dans ce chapitre, le problème d'optimisation qui se pose consiste à définir un ordonnancement d'un ensemble de machines pour répondre à une certaine demande, avec pour objectif la maximisation de l'horizon de production. Afin de répondre à ce problème, il est proposé d'utiliser les machines suivant une approche originale. Les machines sont en effet supposées pouvoir fournir un certain nombre de niveaux de performance différents, chacun induisant une vitesse d'usure spécifique. Chaque profil de fonctionnement des machines est ainsi associé à une certaine durée de vie. La relation entre les débits disponibles et leurs durées de vie est modélisée par deux représentations, chacune d'entre elles s'appliquant à un type de machines bien distinct. Le premier modèle proposé définit une variation discrète des débits disponibles, tandis que les performances évoluent de façon continue entre deux extrêmes pour le second.

Différentes méthodes de résolution sont proposées dans les chapitres suivants pour les deux types de machines considérés. Les développements proposés dans le chapitre 4 s'appliquent au premier modèle, tandis que les chapitres 5 et 6 proposent chacun un type de résolution différent pour les machines se comportant suivant le second modèle.

Chapitre 4

Décision avec profils discrets

Sommaire

Le problème d'optimisation détaillé dans le chapitre 3 est tout d'abord étudié pour le cas des machines présentant un nombre discret de profils de fonctionnement, suivant le modèle introduit en partie 3.2. Une étude des propriétés de ce problème et de sa complexité est menée dans la première partie de ce chapitre pour caractériser le problème d'optimisation et déterminer les configurations du problème pour lesquelles une solution optimale peut être trouvée en temps polynomial. Une approche de résolution optimale, basée sur la programmation linéaire en nombres entiers, est ensuite proposée. Cette première approche n'étant efficace que pour des problèmes de petite taille comportant un nombre limité de variables, une résolution sous-optimale du problème est proposée sous la forme de plusieurs heuristiques pour les instances de grande taille. L'efficacité des différentes méthodes de résolution est finalement validée sur plusieurs instances du problème d'optimisation, par le biais de simulations.

4.1 Propriétés et étude de complexité

Une borne supérieure pour le nombre de périodes pouvant être complété K est tout d'abord définie pour le problème le plus général considérant une demande constante dans le temps, MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$). Des résultats de complexités sont ensuite démontrés pour différentes configurations du problème d'optimisation à demande constante. Sous certaines conditions, il est montré qu'une solution optimale peut être trouvée en temps polynomial. La preuve de la NP-complétude du problème le plus général, prenant en compte des machines hétérogènes avec plusieurs profils de fonctionnement, est ensuite apportée.

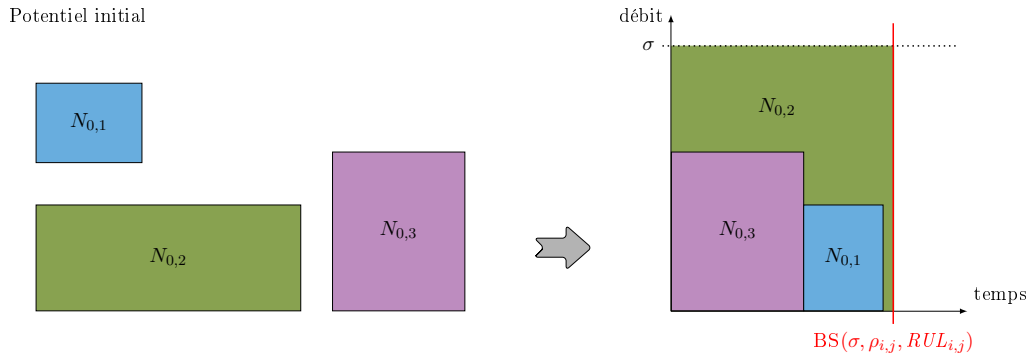
4.1.1 Borne supérieure pour maxK(σ, $\rho_{i,j}$, $RUL_{i,j}$)

Une borne supérieure, notée BS(σ, $\rho_{i,j}$, $RUL_{i,j}$), peut être définie pour le problème général MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) considérant une demande σ constante dans le temps et un ensemble de machines hétérogènes en termes de débits et de RUL. Si toutes les machines sont utilisées avec leur profil de fonctionnement associé au plus grand rendement, $\max_{0\leqslant i<n}(\rho_{i,j} \cdot RUL_{i,j})$, cette borne supérieure, définie par l'équation (4.1), correspond au nombre maximal théorique de périodes K pendant lequel la demande σ peut être atteinte.

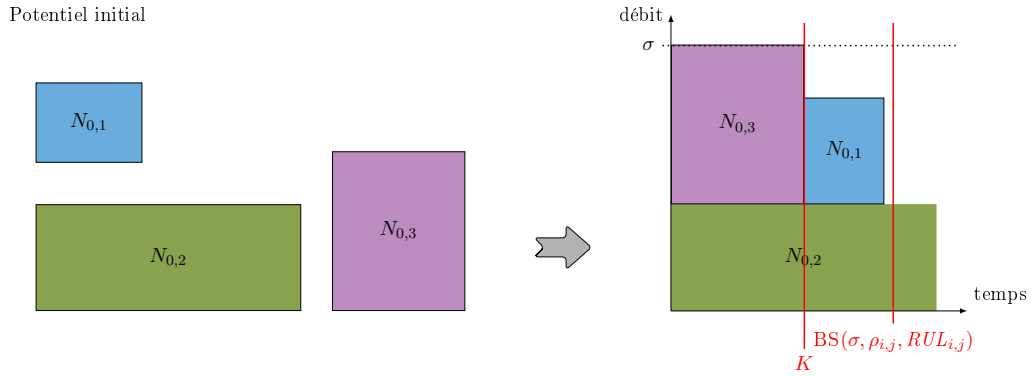
$$\mathrm{BS}(\sigma, \rho_{i,j}, RUL_{i,j}) = \left\lfloor \frac{\sum_{j=1}^{m} \max_{0\leqslant i<n}(\rho_{i,j} \cdot RUL_{i,j})}{\sigma \Delta T} \right\rfloor \tag{4.1}$$

Cette borne n'est atteignable que dans des conditions très restrictives, c'est-à-dire uniquement si aucune surproduction n'est faite durant tout l'horizon de production et si toutes les machines ont atteint leur fin d'utilisation avant maintenance à la fin de l'ordonnancement. En d'autres termes, la borne supérieure est atteignable uniquement si tout le potentiel offert par la plate-forme est utilisé. En pratique, elle n'est que très rarement atteinte. La construction de BS(σ, $\rho_{i,j}$, $RUL_{i,j}$) peut être vue géométriquement comme le remplissage d'une surface rectangulaire, dont la hauteur correspond à la valeur de la demande σ. Cette hauteur étant fixée, le potentiel de toutes les machines est utilisé pour remplir la surface de telle sorte que le plus de rectangles de largeur ΔT soient remplis.

Sauf cas très particulier, l'ordonnancement des machines obtenu avec cette méthode ne respecte pas deux contraintes importantes. La discrétisation du temps n'est tout d'abord pas respectée. Le temps est en effet considéré continu, avec une préemption autorisée à chaque instant pour toutes les machines. Un changement de configuration (machines utilisées et profils de fonctionnement associés) est donc autorisé à chaque instant, même à l'intérieur d'une période de temps ΔT. Le potentiel global de l'ensemble des machines est ensuite utilisé pour remplir le rectangle de hauteur σ sans respecter les caractéristiques des machines. Les valeurs des débits pouvant être fournis par les machines ne sont en effet pas toujours respectées. Une illustration de la construction de la borne supérieure est proposée sur la figure 4.1a pour une demande σ constante dans le temps. On peut voir sur cet exemple que l'utilisation des machines M_1 et M_2 nécessaires pour atteindre la borne supérieure n'est pas conforme aux débits maximaux qu'elles peuvent fournir. Le respect des débits maximaux contraint l'ordonnancement des machines et ne permet d'atteindre qu'une fraction de l'horizon maximal théorique (voir figure 4.1b).



(a) Utilisation du potentiel des machines sans respect de leurs caractéristiques



(b) Ordonnancement avec respect des débits maximaux

Figure 4.1 – Construction de la borne supérieure BS

4.1.2 Complexité du cas avec machines homogènes en débit : maxK(σ, ρ, RUL_j)

Considérons d'abord le problème MAXK(σ, ρ, RUL_j), pour lequel toutes les machines peuvent fournir un seul et même débit ρ. Elles peuvent toutefois être associées à des RUL différents, notés RUL_j. Le débit total requis, σ, est constant dans le temps. Cette demande est par hypothèse atteignable avec q machines telles que $(q-1)\rho \leqslant \sigma \leqslant q \cdot \rho$, avec $q \in \mathbb{N}^*$ le nombre minimum

de machines nécessaires pour atteindre σ pour chaque période de temps ΔT tel que $q \leqslant m$. Le problème peut être exprimé de la manière suivante : comment utiliser q machines, provenant d'un ensemble $\mathscr{M}$ de m machines, de façon à maximiser l'horizon de production ? Le nombre de périodes complétées associé à la solution optimale de ce problème d'optimisation est noté $\mathscr{K}(\mathscr{M}, q)$.

Soit une relaxation de ce problème pour laquelle le temps est continu. Dans ce cas, une machine peut être remplacée par une autre à chaque instant t et l'horizon de production maximal est noté $\mathscr{K}_{cont}(\mathscr{M}, q) \cdot \Delta T$ (voir équation (4.2)).

$$\mathscr{K}_{cont}(\mathscr{M}, q) = \frac{\sum_{j=1}^{m} RUL_j}{q} \tag{4.2}$$

$\mathscr{K}_{cont}(\mathscr{M}, q)$ constitue une borne supérieure pour $\mathscr{K}(\mathscr{M}, q)$, atteignable uniquement sous des conditions très restrictives. De même que pour la borne supérieure définie précédemment, la construction de $\mathscr{K}_{cont}(\mathscr{M}, q)$ peut être vue comme une résolution géométrique, dont la solution ne respecte pas toujours les caractéristiques des profils de fonctionnement des machines.

Par exemple, soient trois machines M_1, M_2 et M_3, avec $RUL_1(1) = 1\Delta T$, $RUL_2(1) = 1\Delta T$ et $RUL_3(1) = 4\Delta T$. Conformément à l'équation (4.2), si deux machines sont utilisées en parallèle ($q = 2$), $\mathscr{K}_{cont}(\mathscr{M}, q) = (1 + 1 + 4)/2 = 3\Delta T$. Les machines M_1 et M_2 peuvent par exemple être utilisées simultanément pendant une période de temps. Dans ce cas, au temps $t = \Delta T$, $RUL_1(2) = RUL_2(2) = 0$ et $RUL_3(2) = 4\Delta T$. Pour atteindre l'horizon de production maximal théorique $\mathscr{K}_{cont}(\mathscr{M}, q) = 3\Delta T$, la machine M_3 devrait être utilisée en parallèle avec elle-même pendant deux périodes, ce qui est impossible.

4.1.2.1 Algorithme glouton optimal pour maxK(σ, ρ, RUL_j)

Le problème MAXK(σ, ρ, RUL_j) est très similaire au problème classique d'ordonnancement de machines parallèles $P_q|prmp|C_{max}$ étudié par Pinedo dans [79], pour lequel m tâches doivent être ordonnancées sur q machines parallèles, avec comme objectif la minimisation du temps total d'exécution des tâches (makespan).

On retrouve tout d'abord deux hypothèses identiques dans chacun de ces problèmes d'ordonnancement, à savoir la discrétisation du temps en périodes et l'autorisation d'une préemption discrète. Pour le problème $P_q|prmp|C_{max}$, cette dernière hypothèse signifie que les tâches peuvent être interrompues avant la fin de leur exécution et redémarrées plus tard, sans conséquence sur leur temps d'exécution. Pour le problème MAXK(σ, ρ, RUL_j), la préemption aurait éventuellement lieu entre deux périodes de temps consécutives et son autorisation signifie qu'un arrêt peut être défini au cours de l'utilisation d'une machine, sans que cela ne modifie sa durée d'utilisation résiduelle.

Un parallèle peut ensuite être défini entre les différents éléments de chacun des problèmes. Les q machines devant fonctionner en parallèle pour atteindre la demande σ dans le problème MAXK(σ, ρ, RUL_j) sont équivalentes aux q ressources parallèles considérées dans $P_q|prmp|C_{max}$. A la place des m tâches devant être exécutées pour $P_q|prmp|C_{max}$, le problème MAXK(σ, ρ, RUL_j) considère m machines à utiliser. Les temps d'exécution des m tâches peuvent être assimilées aux

durées d'utilisation résiduelles (RUL) des m machines. Effectuer une tâche dans sa totalité revient alors à utiliser tout le potentiel d'une machine.

L'algorithme LRPT (Longest Remaining Processing Time first) a été proposé par Pinedo [79] pour trouver un ordonnancement optimal pour le problème $P_q|prmp|C_{max}$ en temps discret. Pour chaque période de temps, cet algorithme consiste à ordonnancer en priorité les tâches ayant les temps d'exécution restants les plus longs. Cela est rendu possible par l'hypothèse de préemption autorisée. L'ordonnancement obtenu est actif, sans délai et sans temps mort. Ces caractéristiques sont compatibles avec le problème d'ordonnancement considéré dans cette section, MAXK(σ, ρ, RUL_j). Ce dernier peut donc être résolu de façon optimale en utilisant l'algorithme glouton $LRUL$ (Longest Remaining Useful Life first), basé sur le principe de l'algorithme LRPT. De la même façon que LRPT, l'algorithme $LRUL$ consiste à utiliser en priorité les machines ayant les durées d'utilisation résiduelles les plus longues. L'arrêt et de le démarrage des machines sont autorisés à chaque début de période de temps. Au début de chaque période k ($1 \leqslant k \leqslant K$), les q machines ayant les RUL les plus longs au temps $k\Delta T$ sont alors engagées pour une période de temps.

4.1.2.2 Complexité du problème maxK(σ, ρ, RUL_j)

Plusieurs lemmes sont utilisés pour définir la complexité du problème d'optimisation traité, MAXK(σ, ρ, RUL_j).

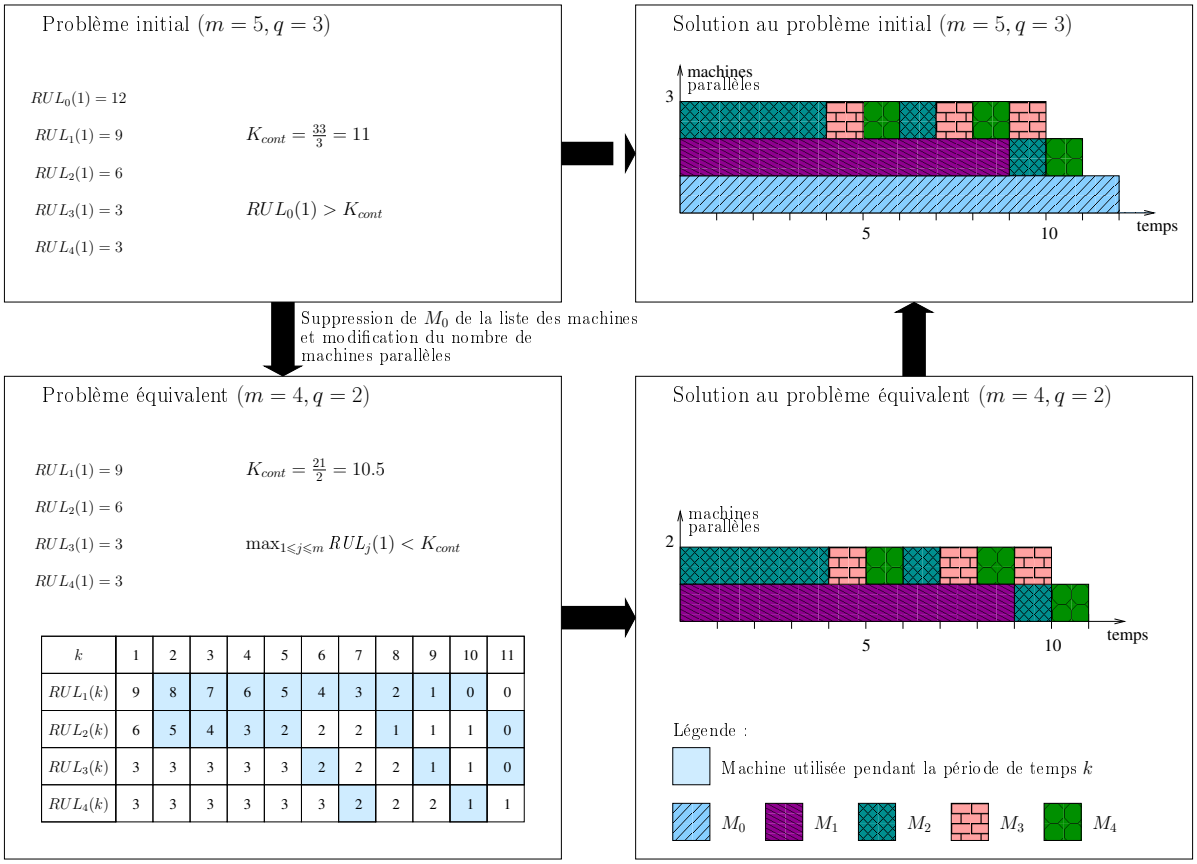
Lemme 1. *Si* $q < m$ *et* $\max_{1\leqslant j\leqslant m}(RUL_j) \geqslant \mathscr{K}_{cont}(\mathscr{M}, q)$, *alors* $\mathscr{K}(\mathscr{M}, q) = \mathscr{K}(\mathscr{M}', q-1)$, *avec* $\mathscr{M}' = \mathscr{M} \setminus \{M_{j'}$ *t.q.* $RUL_{j'} = \max_{1\leqslant j\leqslant m}(RUL_j)\}$.

Selon le lemme 1, résoudre un problème satisfaisant ses propriétés est équivalent à résoudre le même problème pour une demande $\sigma' = \sigma - \rho$, sans prendre en compte la machine ayant le RUL le plus long et dépassant la borne supérieure $\mathscr{K}_{cont}(\mathscr{M}, q)$. Cette simplification est illustrée sur la figure 4.2.

Démonstration. Si $\max_{1\leqslant j\leqslant m}(RUL_j) \geqslant \mathscr{K}_{cont}(\mathscr{M}, q)$, la machine ayant le RUL le plus long est utilisée durant tout l'horizon de l'ordonnancement. Le nombre maximal de périodes pour lequel la demande peut être satisfaite, $\mathscr{K}(\mathscr{M}, q)$, n'est alors pas limité par cette machine. La valeur $\mathscr{K}(\mathscr{M}, q)$ peut donc être déterminée en résolvant le problème sans considérer la machine ayant le RUL le plus long. L'apport de cette machine, ρ, au service global est toutefois conservé. La demande associée au nouveau problème d'optimisation est alors $\sigma' = \sigma - \rho$ et le nombre de machines nécessaires pour atteindre cette demande est $q' = q - 1$. □

Lemme 2. *Si* $q \leqslant m$, $RUL_j(1) \geqslant 1$ $\forall 1 \leqslant j \leqslant m$ *et* $\max_{1\leqslant j\leqslant m}(RUL_j(1)) \leqslant \mathscr{K}_{cont}(\mathscr{M}, q)$, *alors un ordonnancement optimal pour le problème* MAXK(σ, ρ, RUL_j) *peut être déterminé en utilisant l'algorithme glouton LRUL (Largest Remaining Useful Life first). Dans ce cas, l'horizon de production de cet ordonnancement est* $\mathscr{K}(\mathscr{M}, q)\Delta T = \lfloor \mathscr{K}_{cont}(\mathscr{M}, q) \rfloor \Delta T$.

Démonstration. Ce lemme est démontré en découpant l'ordonnancement en deux phases successives. L'algorithme $LRUL$ est utilisé pour distribuer q machines différentes période après période dans l'ordre décroissant de leurs durées d'utilisation restantes. Le suivi des règles de l'algorithme $LRUL$ entraîne une utilisation homogène des machines. Au bout d'un certain temps, cela permet



k	1	2	3	4	5	6	7	8	9	10	11
$RUL_1(k)$	9	8	7	6	5	4	3	2	1	0	0
$RUL_2(k)$	6	5	4	3	2	2	2	1	1	1	0
$RUL_3(k)$	3	3	3	3	3	2	2	2	1	1	0
$RUL_4(k)$	3	3	3	3	3	3	2	2	2	1	1

Figure 4.2 – Simplification d'un problème dans le cadre du Lemme 1

d'éviter que le nombre restant d'unités de RUL à placer dans l'ordonnancement n'appartienne majoritairement qu'à une seule machine.

Les machines sont tout d'abord ordonnancées suivant les règles de l'algorithme $LRUL$ jusqu'à la date $k_1\Delta T$, avec k_1 défini par l'équation (4.3).

$$k_1 = \left\lfloor \frac{\sum_{j=1}^{m} (RUL_j(1) - 1)}{q} \right\rfloor \tag{4.3}$$

D'après la définition de k_1, les durées d'utilisation restantes valent soit 1, soit 2 unités de temps pour chaque machine une fois les k_1 premières périodes effectuées. Le nombre de machines ayant un RUL égal à 2 est de plus strictement inférieur à q. Le nombre d'unités de temps restantes peut être exprimé sous la forme proposée dans l'équation (4.4).

$$\sum_{j=1}^{m} RUL_j(k_1) = m + r, \text{ avec } r \equiv \sum_{j=1}^{m} (RUL_j(1) - 1) \pmod{q} \tag{4.4}$$

A la date $(k_1+1)\Delta T$, toutes les machines avec un RUL égal à 2 ont été utilisées en parallèles d'autres machines et il reste uniquement des RUL valant 1 ou 0. A partir de cet instant, toutes

les machines restantes peuvent être utilisées en parallèle, quelque soit l'ordre dans lequel elles sont placées dans l'ordonnancement. En suivant les règles de l'algorithme $LRUL$, il est alors possible d'ajouter $\lfloor (m+r)/q \rfloor$ périodes aux k_1 périodes déjà complétées. On a alors :

$$\begin{aligned}
\mathscr{K}(\mathscr{M}, q) &= k_1 + \left\lfloor \frac{m+r}{q} \right\rfloor \\
&= \left\lfloor \frac{\sum_{j=1}^{m} (RUL_j(1) - 1)}{q} \right\rfloor + \left\lfloor \frac{m+r}{q} \right\rfloor \\
&= \frac{\sum_{j=1}^{m} (RUL_j(1) - 1)}{q} - \frac{r}{q} + \frac{m+r}{q} - \frac{r'}{q} \qquad \text{avec } r' \equiv (m+r) \pmod{q} \\
&= \frac{\sum_{j=1}^{m} RUL_j(1) - m + m - r + r}{q} - \frac{r'}{q} \\
&= \frac{\sum_{j=1}^{m} RUL_j(1)}{q} - \frac{r'}{q} \\
&= \left\lfloor \frac{\sum_{j=1}^{m} RUL_j(1)}{q} \right\rfloor
\end{aligned}$$

On en déduit finalement que l'horizon de production de l'ordonnancement est $\mathscr{K}(\mathscr{M}, q)\Delta T = \lfloor \mathscr{K}_{cont}(\mathscr{M}, q) \rfloor \Delta T$. □

Théorème 1. *Une solution optimale peut être trouvée en temps polynomial pour le problème* MAXK(σ, ρ, RUL_j)*, en* $O(\mathscr{K}_{cont}(\mathscr{M}, q) \cdot m \cdot \log(m))$.

Démonstration. La complexité du problème d'optimisation est étudiée pour toutes les configurations possibles.

– Cas particuliers :

$$\mathscr{K}(\mathscr{M}, q) = \begin{cases} 0 & \text{si } q > m \ (4.5.1) \\ \sum_{j=1}^{m} RUL_j(1) & \text{si } q = 1 \ (4.5.2) \\ \min_{1 \leqslant j \leqslant m} RUL_j(1) & \text{si } q = m \ (4.5.3) \end{cases} \tag{4.5}$$

Si $q > m$, le potentiel des machines disponibles n'est pas suffisant pour atteindre le niveau de service total requis σ. L'horizon de production est donc égal à zéro (voir l'équation (4.5.1) et la figure 4.3a).

Si $q = 1$, une machine suffit pour atteindre le niveau de service requis. Une seule machine va donc être utilisée à chaque période. Dans ce cas, toutes les machines peuvent être utilisées les unes après les autres, chacune jusqu'à sa fin de vie. L'horizon de production maximal est donc obtenu

en additionnant les valeurs initiales des RUL de toutes les machines (voir l'équation (4.5.2) et la figure 4.3c).

Si $q = m$, toutes les machines disponibles doivent être utilisées en parallèle pour atteindre la demande σ. L'horizon de production est alors limité par la machine dont la durée d'utilisation disponible est la plus petite (voir l'équation (4.5.3) et la figure 4.3b).

Pour ces trois cas particuliers correspondant à des configurations bien précises de l'ensemble de machines considérées, la solution au problème MAXK(σ, ρ, RUL_j) peut être trouvée respectivement en $O(1)$, $O(m)$ et $O(m)$. Pour le cas $q > m$, un seul test suffit en effet à déterminer que la demande n'est pas atteignable. Pour le cas $q = 1$, une somme des RUL des machines doit être faite. Enfin, lorsque $q = m$, il suffit de chercher le RUL le plus petit parmi m valeurs.

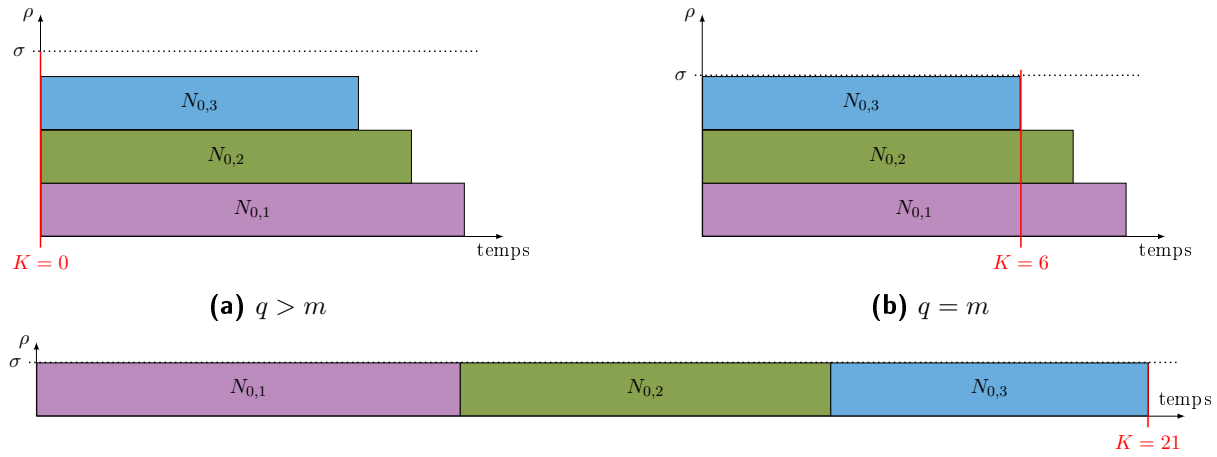


Figure 4.3 – Illustration des cas particuliers pour la résolution du problème MAXK(σ, ρ, RUL_j)

– Autres cas :

$$\mathscr{K}(\mathscr{M}, q) = \left\lfloor \frac{\sum_{j=1}^{m} RUL_j(1)}{q} \right\rfloor \quad \text{si } 1 < q < m \text{ et } RUL_j(1) \geqslant 1 \quad \forall\, 1 \leqslant j \leqslant m \tag{4.6}$$

On suppose que les conditions d'application du lemme 2 sont satisfaites. Dans le cas contraire, le lemme 1 assure qu'il existe un problème équivalent qui satisfait ces conditions. D'après le lemme 2, l'algorithme $LRUL$ définit un ordonnancement optimal, avec un horizon $\mathscr{K}(\mathscr{M}, q)\Delta T$ donné par l'équation (4.6).

Un maximum de $\mathscr{K}_{cont}(\mathscr{M}, q)$ tris des m machines est effectué durant la construction de l'ordonnancement. La complexité de l'algorithme utilisé est donc en $O(\mathscr{K}_{cont}(\mathscr{M}, q) \cdot m \cdot \log(m))$.

□

4.1.3 NP-complétude du cas général

Dans le cas le plus général, le problème MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$), prenant en compte des machines hétérogènes avec plusieurs profils de fonctionnement, s'avère être NP-complet au sens fort.

Théorème 2. *La recherche d'une solution optimale au problème* MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) *est un problème NP-complet au sens fort.*

Démonstration. La démonstration de la NP-complétude du problème considérant le cas le plus général MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) se base sur la démonstration de la NP-complétude du problème particulier MAXK(σ, ρ_j, 1), pour lequel toutes les machines peuvent être utilisées avec un seul profil de fonctionnement pendant une période de temps ($RUL_{i,j} = RUL_j = 1 \ \forall\, 0 \leqslant i \leqslant n-1$, $\forall\, 1 \leqslant j \leqslant m$).

Considérons le problème de décision suivant : étant donné un horizon de K périodes, existe-t-il un ordonnancement qui définit les contributions de chaque machine dans le temps de telle sorte que la demande σ soit atteinte à chaque période k ($1 \leqslant k \leqslant K$) ? En d'autres termes, si $\mathscr{M}_k \subset \mathscr{M}$ est l'ensemble des machines utilisées pendant la période k, $\forall\, k \leqslant K$, a-t-on la propriété $\sum_{j|M_j \in \mathscr{M}_k} \rho_j \geqslant \sigma$? Ce problème est dans NP. En effet, étant donné un ordonnancement de K périodes, il est facile de vérifier en temps polynomial s'il est valide ou non. La NP-complétude est obtenue par le biais d'une réduction à un problème de 3-PARTITION [43], qui est NP-complet au sens fort.

Considérons une instance $\mathcal{I}_1$ de 3-PARTITION. Étant donnés un entier B et $3K$ entiers positifs $a_1, a_2, \ldots, a_{3K}$ tels que, $\forall\, j \in \{1, \ldots, 3K\}$, $B/4 < a_j < B/2$ et avec $\sum_{j=1}^{K} a_j = KB$, existe-t-il une partition $I_1, \ldots, I_K$ de $\{1, \ldots, 3K\}$ tel que $\forall\, k \in \{1, \ldots, K\}$, $|I_k| = 3$ et $\sum_{j \in I_k} a_j = B$? Considérons maintenant de façon similaire l'instance $\mathcal{I}_2$ adaptée au problème de décision $\mathscr{K}(\sigma, \rho_j, 1)$, avec K périodes de temps. Chaque période est supposée être de taille $\Delta T = 1$ unité de temps. La demande est de la forme $\sigma_k = \sigma = B$ pour tout $1 \leqslant k \leqslant K$. L'ensemble $\mathscr{M}$ contient $3K$ machines M_j, avec $RUL_j = 1$ et $\rho_j = a_j$ pour tout $1 \leqslant j \leqslant m = 3K$. $\mathcal{I}_2$ est de taille polynomiale en la taille de l'instance définie précédemment, $\mathcal{I}_1$. Il s'agit maintenant de montrer que $\mathcal{I}_2$ admet une solution si et seulement si $\mathcal{I}_1$ en admet une.

Supposons tout d'abord que $\mathcal{I}_1$ admet une solution. Pour chaque période $1 \leqslant k \leqslant K$, chaque machine M_j est assignée à l'ensemble $\mathscr{M}_k$ pour la période k, avec $j \in I_k$ et $\rho_j = a_j$. On a alors : $\sum_{j|M_j \in \mathscr{M}_k} \rho_j = \sigma = \sum_{j \in I_k} a_j = B$. La demande σ est donc bien atteinte pour les K périodes et la solution de $\mathcal{I}_1$ est aussi une solution pour $\mathcal{I}_2$.

Supposons ensuite que $\mathcal{I}_2$ admet une solution. Soit $\mathscr{M}_k$ l'ensemble des machines utilisées pendant la période k de telle sorte que, pour chaque machine $M_j \in \mathscr{M}_k$ avec $j \in I_k$, $\sum_{j \in I_k} \rho_j = \sigma = B$. Étant données les valeurs des ρ_j, $|\mathscr{M}_k| = |I_k| = 3$. La solution de $\mathcal{I}_2$ est donc aussi solution de $\mathcal{I}_1$. L'équivalence entre $\mathcal{I}_1$ et $\mathcal{I}_2$ étant démontrée et $\mathcal{I}_1$ étant une 3-PARTITION, l'instance $\mathcal{I}_2$ décrivant notre problème en est aussi une.

Cela prouve que le problème MAXK(σ, ρ_j, 1) est NP-complet au sens fort. Le problème MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) en étant une généralisation, il est aussi NP-complet au sens fort. □

4.1.4 Remarques sur le traitement du problème avec demande variable

Les propriétés et résultats de complexité proposés jusqu'à présent s'appliquent aux problèmes considérant une demande σ constante sur tout l'horizon de production. Les différents résultats peuvent toutefois être plus ou moins directement adaptés pour une demande σ_k constante par morceaux.

La borne supérieure BS(σ, $\rho_{i,j}$, $RUL_{i,j}$) définie précédemment peut tout d'abord être modifiée pour respecter la variabilité de la demande. Il s'agit de déterminer la dernière période k pour laquelle la demande σ_k peut être atteinte. Pour cela, le potentiel total de la plate-forme de machines est distribué période par période. Le numéro de la dernière période pouvant être complétée correspond à la valeur maximale de K. La borne supérieure BS(σ_k, $\rho_{i,j}$, $RUL_{i,j}$) pour le nombre de périodes pouvant être complétées pour une demande σ_k variable suit alors la relation définie par l'équation (4.7).

$$\mathrm{BS}(\sigma_k,\, \rho_{i,j},\, RUL_{i,j}) = \underset{K}{\operatorname{argmin}} \left(\sum_{j=1}^{m} \max_{0 \leqslant i < n} (\rho_{i,j} \cdot RUL_{i,j}) - \sum_{k=0}^{K} \sigma_k \Delta T \geqslant 0 \right) \tag{4.7}$$

Si la demande σ_k est de la forme $(q_k - 1)\rho \leqslant \sigma_k \leqslant q_k \cdot \rho$, avec $q_k \in \mathbb{N}^*$ $(1 \leqslant k \leqslant K)$ le nombre minimal de machines nécessaires pour atteindre σ_k à chaque période k, l'algorithme glouton *LRUL* proposé pour définir un ordonnancement optimal pour le problème MAXK(σ, ρ, RUL_j) peut être utilisé pour trouver un ordonnancement optimal pour le problème équivalent MAXK(σ_k, ρ, RUL_j) avec demande variable.

Le problème MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) étant un cas particulier du problème d'optimisation général MAXK(σ_k, $\rho_{i,j}$, $RUL_{i,j}$) avec demandeq variable, cela suffit finalement à prouver que MAXK(σ_k, $\rho_{i,j}$, $RUL_{i,j}$) est aussi NP-complet au sens fort.

4.2 Approche optimale

Une première approche basée sur une méthode de résolution exacte est proposée pour traiter le problème d'optimisation général MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$). Il a été montré dans la section précédente que ce problème est NP-complet au sens fort. Une caractérisation du problème d'optimisation peut toutefois être proposée par le biais d'une formulation optimale. Nous proposons ici d'exprimer le problème sous la forme d'un Programme Linéaire en Nombres Entiers (PLNE).

4.2.1 Programmation Linéaire

Une variante du problème d'optimisation MAXK(σ_k, $\rho_{i,j}$, $RUL_{i,j}$) peut être définie de la manière suivante : considérant l'état de santé des machines à disposition, c'est-à-dire leurs valeurs de $RUL_{i,j}$ $(0 \leqslant i \leqslant n-1$ et $1 \leqslant j \leqslant m)$, existe-t-il un ordonnancement qui permet de satisfaire la demande σ_k durant un nombre donné de périodes K ? Pour un horizon $K\Delta T$ donné, la recherche de solution peut être modélisée sous la forme d'un programme mathématique.

Soit $x_{i,j,k}$ $(0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m$ et $1 \leqslant k \leqslant K)$ une variable binaire telle que $x_{i,j,k} = 1$

si la machine M_j est utilisée avec son profil de fonctionnement $N_{i,j}$ pendant la période k et $x_{i,j,k} = 0$ dans le cas contraire. La définition d'une variable pour chaque période de temps permet d'assurer que les arrêts/démarrages des machines, ainsi que les changements de profil de fonctionnement en cours d'utilisation, ne sont autorisés qu'en début de période.

Les contraintes du programme mathématique expriment à la fois les exigences en termes de service et les limites d'utilisation des machines. Le premier ensemble de contraintes défini par l'équation (4.8) permet de limiter l'utilisation d'une machine durant une période k. Chaque machine ne peut en effet être utilisée qu'une seule fois par période, avec un seul profil de fonctionnement $N_{i,j}$.

$$\sum_{i=0}^{n-1} x_{i,j,k} \leqslant 1 \quad \forall 1 \leqslant j \leqslant m,\ \forall 1 \leqslant k \leqslant K \tag{4.8}$$

Le second ensemble de contraintes, défini par l'équation (4.9), assure que les machines ne sont pas utilisées plus longtemps que leur RUL respectif. On considère que si une machine M_j est utilisée avec le profil de fonctionnement $N_{i,j}$ durant une période k, $\Delta T/RUL_{i,j}$ unités de temps doivent être retranchées à la valeur initiale du RUL associé au profil $N_{i,j}$. La somme de toutes les portions successivement soustraites à la durée de vie initiale au cours de l'utilisation d'une machine ne doit pas dépasser 100%.

$$\sum_{i=0}^{n-1} \frac{\sum_{k=1}^{K} x_{i,j,k} \cdot \Delta T}{RUL_{i,j}} \leqslant 1 \quad \forall 1 \leqslant j \leqslant m \tag{4.9}$$

Le dernier ensemble de contraintes (équation (4.10)) permet d'assurer la satisfaction de la demande σ durant tout l'horizon de production, c'est-à-dire pour toutes les périodes k telles que $1 \leqslant k \leqslant K$.

$$\sum_{j=1}^{m} \sum_{i=0}^{n-1} (x_{i,j,k} \cdot \rho_{i,j}) \geqslant \sigma \quad \forall 1 \leqslant k \leqslant K \tag{4.10}$$

Le programme mathématique défini par les équations (4.8), (4.9) et (4.10) permet de vérifier l'existence d'une solution pour le problème de décision sans aucune fonction objectif. L'ajout d'une fonction objectif permet toutefois, suivant certains critères, de sélectionner la meilleure configuration des machines parmi toutes les solutions du programme mathématique. En cohérence avec les hypothèses détaillées dans l'énoncé du problème dans la partie 3.1, la fonction objectif proposée dans l'équation (4.11) tend à minimiser la surproduction globale sur tout l'horizon de production. La minimisation du potentiel restant à la fin de l'ordonnancement pourrait aussi être considérée. Cette fonction objectif engendrerait potentiellement une certaine surproduction, mais maximiserait le nombre de machines en fin de vie au terme de l'horizon de production. Cela permettrait de grouper un maximum d'opérations de maintenance et donc de minimiser les coûts associés.

$$\min \quad \sum_{k=1}^{K} \left(\sum_{j=1}^{m} \sum_{i=0}^{n-1} x_{i,j,k} \cdot \rho_{i,j,k} - \sigma \right) \tag{4.11}$$

Toutes les contraintes étant linéaires, le programme mathématique peut être résolu par une programmation linéaire. Le Programme Linéaire en Nombres Entiers (PLNE) correspondant est défini par le système d'équations (4.12). Les seules variables sont les variables binaires $x_{i,j,k}$. Les grandeurs $\rho_{i,j,k}$, $RUL_{i,j}$ et σ sont des données du problème.

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{K} \left(\sum_{j=1}^{m} \sum_{i=0}^{n-1} x_{i,j,k} \cdot \rho_{i,j,k} - \sigma \right) & \text{(4.12a)} \\ & \displaystyle\sum_{i=0}^{n-1} x_{i,j,k} \leqslant 1 \qquad \forall 1 \leqslant j \leqslant m,\ \forall 1 \leqslant k \leqslant K & \text{(4.12b)} \\ \text{t.q.} & \displaystyle\sum_{i=0}^{n-1} \frac{\sum_{k=1}^{K} x_{i,j,k} \cdot \Delta T}{RUL_{i,j}} \leqslant 1 \qquad \forall 1 \leqslant j \leqslant m & \text{(4.12c)} \\ & \displaystyle\sum_{j=1}^{m} \sum_{i=0}^{n-1} (x_{i,j,k} \cdot \rho_{i,j}) \geqslant \sigma \qquad \forall 1 \leqslant k \leqslant K & \text{(4.12d)} \\ \text{avec} & x_{i,j,k} \in \{0,1\} \qquad \forall 1 \leqslant i \leqslant n-1,\ \forall 1 \leqslant j \leqslant m,\ \forall 1 \leqslant k \leqslant K & \text{(4.12e)} \end{cases}$$

4.2.2 Résolution optimale

Le programme linéaire présenté dans la partie précédente permet de trouver la meilleure solution au problème de décision considéré, pour un nombre fixe de périodes. Cette approche n'est pas suffisante pour résoudre le problème d'optimisation MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$), qui consiste à déterminer le nombre maximal de périodes K pour lequel une solution existe. Une recherche dichotomique est donc utilisée pour déterminer la plus grande valeur de K possible. L'approche est détaillée dans l'algorithme 1. En première approximation, la recherche dichotomique peut être faite entre 0 et la borne supérieure pour le nombre de périodes complétés définie dans la partie 4.1.1. Afin de restreindre l'espace de recherche, une meilleure borne inférieure peut être déterminée par le biais d'un processus heuristique. L'horizon atteint avec une heuristique étant sous-optimal, le nombre de périodes associé constitue une borne inférieure pour le nombre optimal de périodes.

Le problème d'optimisation MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$) étant NP-complet, des solutions optimales peuvent être trouvées en temps limité uniquement pour des petites instances, considérant à la fois peu de machines, peu de profils de fonctionnement pour chacune d'entre elles et un horizon de production court.

Un moyen de remédier à cette limitation serait de relaxer le programme linéaire en considérant les variables de décision $x_{i,j,k}$ non plus comme des variables binaires, mais comme des variables réelles pouvant prendre toutes les valeurs entre 0 et 1. Dans ce cas, la reconstruction des solutions à partir des valeurs des $x_{i,j,k}$ n'est toutefois pas possible. Les variables $x_{i,j,k}$ contenant à la fois la dimension temporelle (k) et celle associée à l'ensemble des profils de fonctionnement

(i), il est par exemple impossible de déterminer si $x_{i,j,k} = 0.5$ signifie que la machine M_j est utilisée avec son profil de fonctionnement $N_{i,j}$ pendant la moitié de la période k, ou qu'elle est utilisée pendant toute la durée de la période k, mais avec un débit égal à la moitié du $\rho_{i,j}$ correspondant. On voit dans cet exemple qu'un second problème se pose, à savoir que le programme relaxé peut définir une utilisation des machines non conforme aux caractéristiques de leurs profils de fonctionnement disponibles. La relaxation du programme linéaire proposé donnant des solutions qui ne peuvent pas être traduites en un ordonnancement faisable des machines, elle ne permet pas de résoudre le problème d'optimisation considéré.

Algorithme 1: Approche dichotomique permettant de trouver le nombre maximal de périodes K pour lequel le programme linéaire 4.12 admet une solution

Remarque : soit $PLNE(\sigma, \rho_{i,j}, RUL_{i,j})$ le programme linéaire en nombres entiers associé au problème d'optimisation MAXK$(\sigma, \rho_{i,j}, RUL_{i,j})$ et $PL(\sigma, \rho_{i,j}, RUL_{i,j})$ sa relaxation rationnelle

Données :
- Kmin : borne inférieure pour K
- Kmax : borne supérieure pour K
- K : nombre total de périodes pour lequel l'existence d'une solution est testée

Entrées :
- BI$(\sigma, \rho_{i,j}, RUL_{i,j})$: borne inférieure initiale pour K
- BS$(\sigma, \rho_{i,j}, RUL_{i,j})$: borne supérieure initiale pour K

Sorties :
- K : nombre de périodes maximal pour lequel une solution existe

```
Kmin ← BI(σ, ρ_{i,j}, RUL_{i,j})
Kmax ← BS(σ, ρ_{i,j}, RUL_{i,j})
tant que Kmax − Kmin > 0 faire
    K ← (Kmin + Kmax)/2
    si PL(σ, ρ_{i,j}, RUL_{i,j}) admet une solution alors
        si PLNE(σ, ρ_{i,j}, RUL_{i,j}) admet une solution alors
            Kmin ← K
        sinon
            Kmax ← K
    sinon
        Kmax ← K
retourner K
```

4.3 Résolution sous-optimale

L'approche optimale décrite dans la partie précédente ne peut être utilisée que pour des instances de petite taille, avec à la fois un petit nombre de machines, peu de profils de fonctionnement pour chacune d'entre elles et des horizons de production courts. Afin de trouver des ordonnancements pour des problèmes de tailles plus réalistes, nous proposons des heuristiques qui définissent des ordonnancements en temps polynomial avec pour objectif la satisfaction de la demande pendant un temps le plus long possible.

Chaque heuristique suit sa propre stratégie pour sélectionner les machines à utiliser à chaque période de temps et leur profil de fonctionnement, de façon à définir leur contribution au débit global. Une première stratégie consiste à définir l'ordonnancement période par période. Dans

ce cas, une nouvelle sélection de couples machine/profil de fonctionnement est effectuée pour chaque période et appliquée pour une seule période de temps ΔT. Les valeurs de RUL de chaque machine utilisée sont mises à jour à la fin de chaque période complétée. Tout le processus de sélection est réitéré jusqu'à ce que l'ensemble de machines restant ne permette plus d'atteindre la demande. Une seconde stratégie consiste à répéter la même configuration de machines sur le nombre maximum de périodes successives possible. Le nombre de périodes sur lequel une solution peut être appliquée est limité par la machine sélectionnée ayant le RUL le plus petit. A la fin de chaque groupe de périodes ordonnancées, les valeurs de RUL sont mises à jour pour chaque machine de la même façon que pour la première stratégie et le processus est réitéré avec l'ensemble de machines restant. Dans les deux cas, le nombre de périodes complétées K correspond à la durée d'utilisation avant maintenance de la plate-forme.

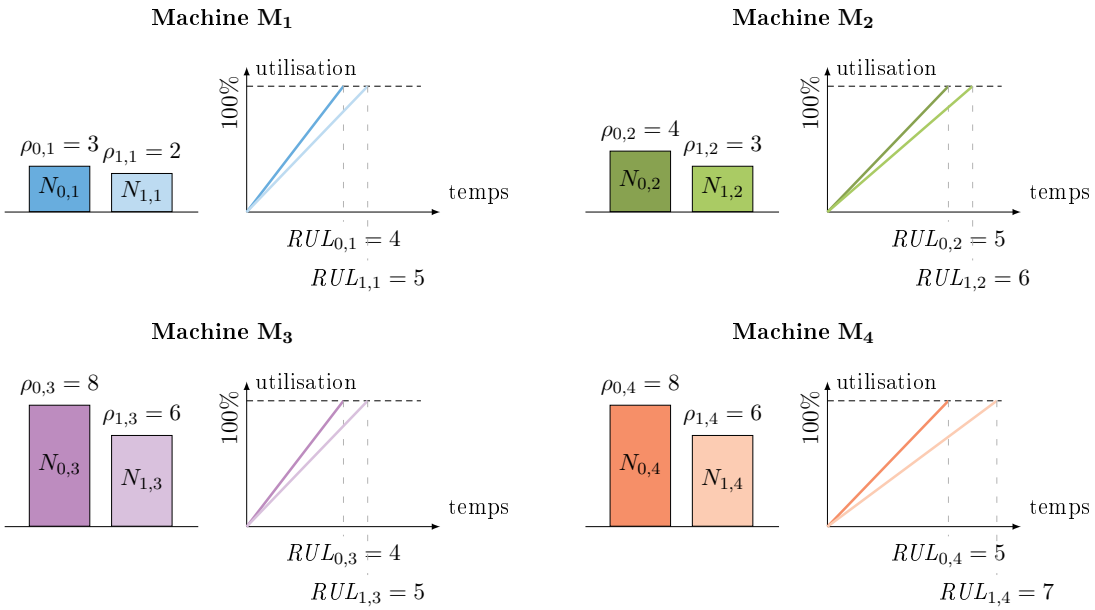


Figure 4.4 – Plate-forme de machines considérée pour les illustrations des solutions obtenues avec les différentes heuristiques

Quelle que soit la stratégie utilisée, trois différents types d'heuristiques peuvent être distingués. Deux heuristiques basiques, H–RAND et H–NAIVE, sont tout d'abord proposées. Elles sont basées sur des processus de sélection très simples et servent principalement de point de comparaison pour les heuristiques présentées dans la suite. Les résultats obtenus avec ces heuristiques basiques montrent l'intérêt de définir des stratégies d'ordonnancement plus élaborées pour étendre la durée d'utilisation d'un système. Le second type d'heuristiques rassemble des heuristiques gloutonnes, H–LRF et H–HTF. Le dernier comprend H–DP, utilisant un algorithme plus élaboré basé sur la programmation dynamique. Le fonctionnement de toutes ces heuristiques est développé ci-après. Pour chacune d'entre elles, une solution basée sur l'ensemble initial de machines, dont les caractéristiques sont représentées sur la figure 4.4, est proposée en illustration. Une amélioration de ces différentes heuristiques est ensuite proposée dans la partie 4.3.2.

4.3.1 Heuristiques

4.3.1.1 H–RAND : heuristique aléatoire

Cette première heuristique H–RAND (pour Random Heuristics) construit un ordonnancement période par période. Pour chaque période de temps k, un nombre juste nécessaire de machines est sélectionné aléatoirement parmi celles qui sont disponibles (avec $RUL_{n-1,j} \geqslant 1$). Pour chaque machine sélectionnée, le profil de fonctionnement associé est également choisi de façon aléatoire en respectant la disponibilité de chaque profil, qui est fonction de la valeur des RUL. La sélection est arrêtée dès que la somme des contributions des machines engagées atteint au moins la demande σ. Le processus de construction de l'ordonnancement, détaillé dans l'algorithme 3, est stoppé à la première période pour laquelle la demande ne peut pas être satisfaite. Cela peut arriver même si le potentiel restant est suffisant pour atteindre σ. A chaque période, les premiers choix contraignent en effet les suivants. Si les premiers choix des couples machine/profil de fonctionnement sont mauvais par rapport à l'objectif, les machines disponibles restantes peuvent ne pas être suffisantes pour atteindre la demande.

Un exemple d'ordonnancement obtenu avec H–RAND pour l'ensemble de machines défini sur la figure 4.4 est proposé sur la figure 4.5.

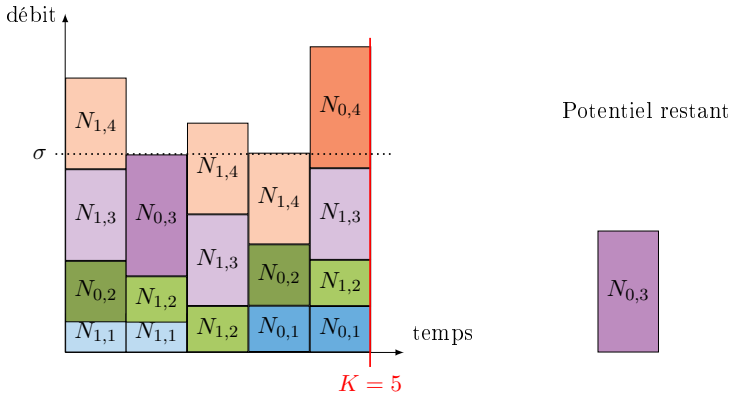


Figure 4.5 – Ordonnancement obtenu avec l'heuristique H–RAND

Algorithme 2: UpdateRUL : mise à jour des valeurs de RUL au cours du processus d'ordonnancement

Données :
u : taux d'utilisation servant à faire le lien entre les valeurs de RUL de tous les profils $N_{i,j}$ pour une même machine M_j
Entrées :
$RUL0[i][j]$: tableau contenant les valeurs initiales $RUL_{i,j}(1)$
$RUL[i][j]$: tableau contenant les valeurs $RUL_{i,j}(k)$ pour la période courante
sol : liste des couples (i, j) sélectionnés pour la période courante
$nbPeriodes$: nombre de périodes pendant lequel la solution sol est appliquée

pour tous les $j \,|\, (*, j) \in sol$ **faire**
  $u \leftarrow nbPeriodes/RUL0[i \,|\, (i, j) \in sol][j]$
  pour $i = 0$ *à* $n - 1$ **faire**
    $RUL[i][j] \leftarrow RUL[i][j] - u \cdot RUL0[i][j]$
retourner RUL

Algorithme 3: H–RAND : heuristique aléatoire

Données :

$list$: liste des couples (i, j) disponibles, tels que $RUL_{i,j}(k) \geqslant 1$ à la période k

sol_k : liste des couples (i, j) sélectionnés pour la période k

ρmax : débit maximal atteignable avec l'ensemble des machines disponibles

ρ_{tot} : débit total fourni par la solution sol_k

$RUL0[i][j]$: tableau contenant les valeurs initiales $RUL_{i,j}(1)$

$RUL[i][j]$: tableau contenant les valeurs $RUL_{i,j}(k)$ pour la période k

$nbPeriodes$: nombre de périodes sur lequel la solution sol_k est appliquée

Entrées :

σ : demande à atteindre à chaque période k

$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j

Sorties :

$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$RUL0[i][j] \leftarrow RUL_{i,j}(1),\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$RUL[i][j] \leftarrow RUL0[i][j],\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$nbPeriodes \leftarrow 1$
$k \leftarrow 1$
répéter
  $list \leftarrow$ liste mise à jour des couples (i, j) disponibles, tels que $RUL[i][j] \geqslant 1$
  $sol_k \leftarrow \emptyset$
  $\rho\text{max} \leftarrow \sum_{j|(*,j)\in list} \max_{i|(i,j)\in list} \rho_{i,j}$
  $\rho_{tot} \leftarrow 0$
  si $\rho max \geqslant \sigma$ **alors**
    tant que $(\rho_{tot} < \sigma)$ *&* $(list \neq \emptyset)$ **faire**
      $(i, j) \leftarrow$ choix aléatoire d'un couple (i, j) dans $list$
      si *la machine sélectionnée M_j n'est pas encore utilisée dans sol_k* **alors**
        $sol_k \leftarrow sol_k \cup \{(i, j)\}$
        $\rho_{tot} \leftarrow \rho_{tot} + \rho_{i,j}$
        $list \leftarrow list \setminus \{(*, j) \in list\}$
    si $\rho_{tot} \geqslant \sigma$ **alors**
      $sol \leftarrow$ ajouter sol_k en fin de sol
      $RUL \leftarrow$ UpdateRUL$(RUL0, RUL, sol_k, nbPeriodes)$ **(voir algorithme 2)**
      $k++$
jusqu'à $\rho_{tot} < \sigma$
retourner $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

4.3.1.2 H–NAIVE : heuristique naïve

La seconde heuristique applique un principe très simple permettant de définir une borne inférieure pour le nombre de périodes complétées. Les machines sont triées par ordre décroissant de leur RUL le plus long, $RUL_{0,j}$, et utilisées avec leur profil de fonctionnement nominal, $N_{0,j}$, fournissant le débit maximal. L'idée est de former autant de groupes de machines que possible de telle sorte que tous les groupes permettent d'atteindre la demande σ. Le nombre de périodes complétées correspond alors à la somme des RUL minimaux dans chaque groupe. La construction de l'ordonnancement suivant le principe de l'heuristique H–NAIVE est détaillée dans l'algorithme 4. Un ordonnancement obtenu avec cette heuristique est proposé sur la figure 4.6.

Algorithme 4: H–NAIVE : heuristique naïve

Données :
$list$: liste des couples (i,j) disponibles, tels que $RUL_{i,j}(k) \geqslant 1$ à la période k
sol_k : liste des couples (i,j) sélectionnés pour la période k
ρmax : débit maximal atteignable avec l'ensemble des machines disponibles
ρ_{tot} : débit total fourni par la solution sol_k
$RUL0[j]$: tableau contenant les valeurs initiales $RUL_{0,j}(1)$
$nbPeriodes$: nombre de périodes sur lequel la solution sol_k est appliquée
Entrées :
σ : demande à atteindre à chaque période k
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$RUL0[j] \leftarrow RUL_{0,j}(1),\ \forall\ 1 \leqslant j \leqslant m$
$k \leftarrow 1$
répéter
  $sol_k \leftarrow \emptyset$
  $\rho\text{max} \leftarrow \sum_{j|(*,j)\in list} \max_{i|(i,j)\in list} \rho_{i,j}$
  $\rho_{tot} \leftarrow 0$
  si $\rho max \geqslant \sigma$ **alors**
    tant que $(\rho_{tot} < \sigma)$ *&* $(list \neq \emptyset)$ **faire**
      $sol_k \leftarrow sol_k \cup \{(0,j') \,|\, RUL_{0,j'} = \max_{j|(0,j)\in list} RUL_{0,j}\}$
      $\rho_{tot} \leftarrow \rho_{tot} + \rho_{0,j'}$
      $list \leftarrow list \setminus \{(i,j') \in list\ \forall\ 0 \leqslant i \leqslant n-1\}$
    si $\rho_{tot} \geqslant \sigma$ **alors**
      $nbPeriodes \leftarrow \min_{j|(0,j)\in sol_k} RUL_{0,j}$
      pour $k' = k$ *à* $k + nbPeriodes$ **faire**
        $sol \leftarrow$ ajouter sol_k en fin de sol
      $k \leftarrow k + nbPeriodes$
jusqu'à $\rho_{tot} < \sigma$
retourner $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

Ces deux premières heuristiques basiques servent principalement de point de comparaison avec les heuristiques présentées dans la suite. Les résultats obtenus avec H–RAND et H–NAIVE, détaillés dans la partie 4.4.3.1 de ce chapitre, montrent l'intérêt de définir des stratégies d'ordonnancement plus élaborées pour étendre la durée d'utilisation avant maintenance d'un système.

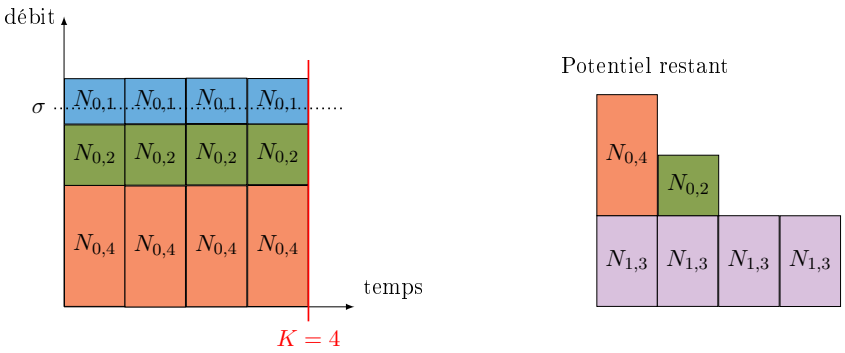


Figure 4.6 – Ordonnancement obtenu avec l'heuristique H–NAIVE

4.3.1.3 H–LRF : heuristique favorisant les *RUL* les plus longs

L'heuristique H–LRF (pour Largest *RUL* First Heuristics) fonctionne par groupes de périodes et favorise l'utilisation des machines dans leur profil de fonctionnement $N_{n-1,j}$ fournissant le débit le plus faible et associé au *RUL* le plus long. L'idée est de maximiser la durée d'utilisation de chaque machines pour maximiser celle de l'ensemble. Un sous-ensemble de machines ayant les $RUL_{n-1,j}$ les plus grands est sélectionné de telle sorte que le débit global produit, ρ_{tot}, atteigne au moins la demande σ. Si les débits des profils $N_{n-1,j}$ des machines disponibles ne permettent pas d'atteindre σ ($\rho_{tot} < \sigma$), le débit total est ensuite augmenté en utilisant si possible la machine M_j ayant le $RUL_{n-1,j}$ le plus élevé avec le profil $N_{i-1,j}$ au lieu de $N_{i,j}$. Ce processus est réitéré jusqu'à ce que le débit total fourni atteigne au moins la demande ($\rho_{tot} \geqslant \sigma$). La solution est appliquée sur le nombre maximal de périodes, correspondant au *RUL* minimal des machines sélectionnées. Le processus de sélection des machines et des profils de fonctionnement associés, détaillé dans l'algorithme 5 et illustré sur la figure 4.7, est arrêté dès que le potentiel restant ne permet plus d'atteindre la demande.

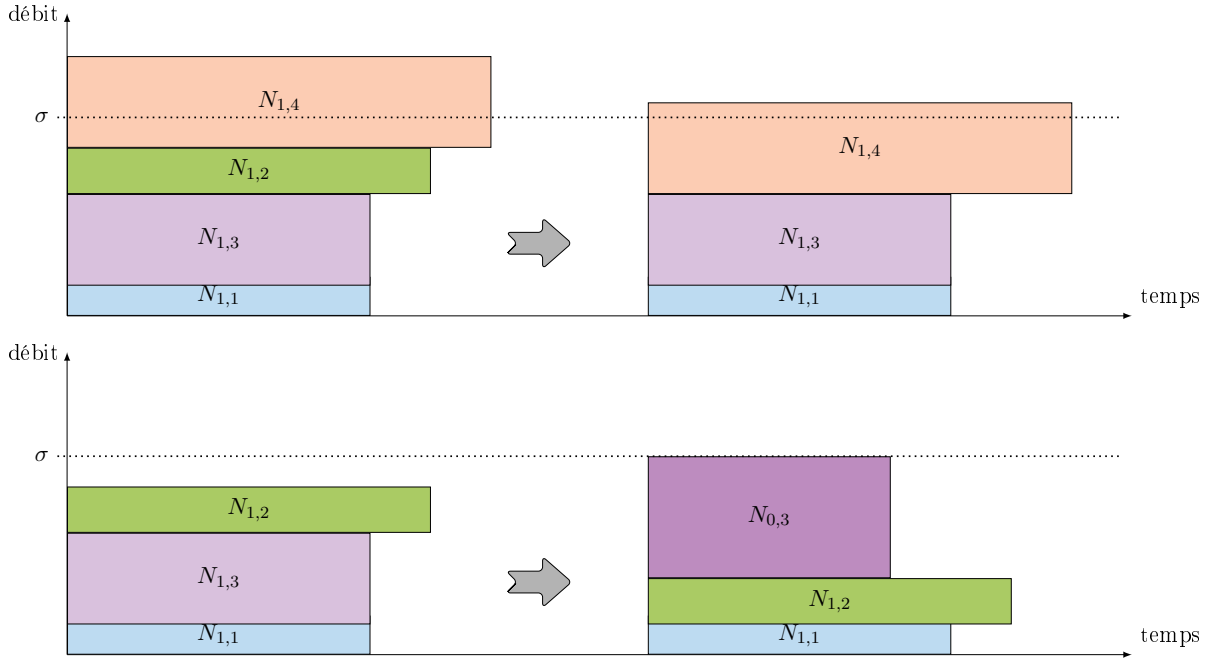


Figure 4.7 – Illustration du principe de fonctionnement de l'heuristique H–LRF

Algorithme 5: H–LRF : heuristique favorisant les RUL les plus longs

Données :
$list$: liste des couples (i,j) disponibles, tels que $RUL_{i,j}(k) \geqslant 1$ à la période k
sol_k : liste des couples (i,j) sélectionnés pour la période k
ρmax : débit maximal atteignable avec l'ensemble des machines disponibles
ρ_{tot} : débit total fourni par la solution sol_k
$RUL0[i][j]$: tableau contenant les valeurs initiales $RUL_{i,j}(1)$
$nbPeriodes$: nombre de périodes sur lequel la solution sol_k est appliquée
Entrées :
σ : demande à atteindre à chaque période k
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$RUL0[i][j] \leftarrow RUL_{i,j}(1),\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$RUL[i][j] \leftarrow RUL0[i][j],\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$k \leftarrow 1$
répéter
  $list \leftarrow$ liste mise à jour des couples (i,j) tels que $RUL[i][j] \geqslant 1$
  $sol_k \leftarrow \emptyset$
  $\rho\text{max} \leftarrow \sum_{j|(*,j)\in list} \max_{i|(i,j)\in list} \rho_{i,j}$
  $\rho_{tot} \leftarrow 0$
  si $\rho max \geqslant \sigma$ **alors**
    $sol_k \leftarrow$ liste des couples $(n-1,j)$ disponibles, tels que $RUL[n-1][j] \geqslant 1\ \forall\ 1 \leqslant j \leqslant m$
    $\rho_{tot} \leftarrow \sum_{j|(n-1,j)\in sol_k} \rho_{n-1,j}$
    tant que $\rho_{tot} < \sigma$ **faire**
      $j' \leftarrow j \,|\, RUL_{*,j} = \max_{j|(*,j)\in sol_k} RUL[*][j]$
      $i' \leftarrow i \,|\, (i,j') \in sol_k$
      $sol_k \leftarrow sol_k \setminus \{(i',j')\}$
      $sol_k \leftarrow sol_k \cup \{(i'-1,j')\}$
      $\rho_{tot} \leftarrow \sum_{j|(i,j)\in sol_k} \rho_{i,j}$
    tant que $(\rho_{tot} > \sigma)$ *&* $(list \neq \emptyset)$ **faire**
      $j' \leftarrow j' \,|\, \rho_{*,j'} = \max_{j|(i,j)\in sol_k} \rho_{i,j}$
      $i' \leftarrow i \,|\, (i,j') \in sol_k$
      si $\rho_{i,j} < \rho_{tot} - \sigma$ **alors**
        $sol_k \leftarrow sol_k \setminus \{(i,j)\}$
        $\rho_{tot} \leftarrow \sum_{j|(i,j)\in sol_k} \rho_{i,j}$
      $list \leftarrow list \setminus \{(i',j')\}$
    si $\rho_{tot} \geqslant \sigma$ **alors**
      $nbPeriodes \leftarrow \min_{j|(*,j)\in sol_k} RUL_{*,j}$
      pour $k' = k$ *à* $k + nbPeriodes$ **faire**
        $sol \leftarrow$ ajouter sol_k en fin de sol
      $RUL \leftarrow \text{UpdateRUL}(RUL0, RUL, sol_k, nbPeriodes)$ **(voir algorithme 2)**
      $k \leftarrow k + nbPeriodes$
jusqu'à $\rho_{tot} < \sigma$
retourner $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

La figure 4.8 représente une solution obtenue avec l'heuristique H–LRF en considérant l'ensemble des machines décrit sur la figure 4.4.

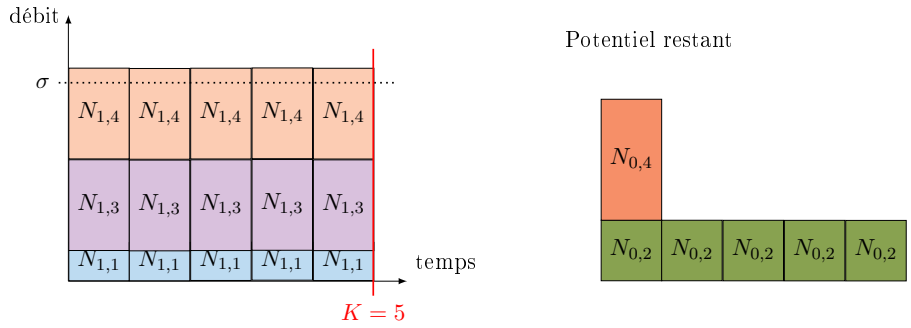
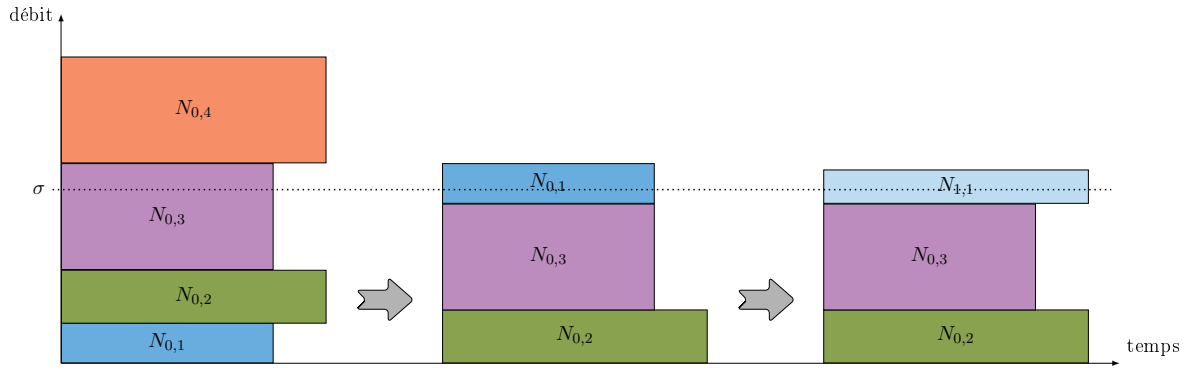


Figure 4.8 – Ordonnancement obtenu avec l'heuristique H–LRF

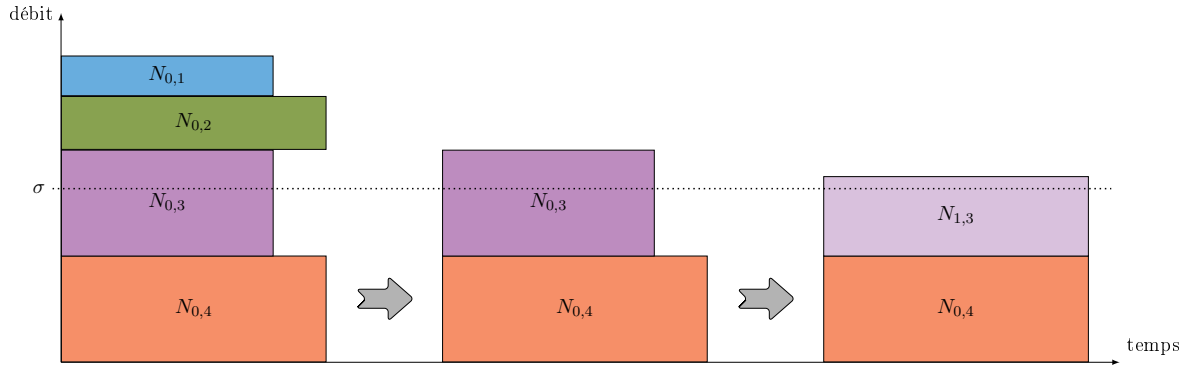
4.3.1.4 H–HTF : heuristique favorisant les débits les plus forts

L'heuristique H–HTF (pour Highest Throughput First Heuristics) est basée sur le même principe de fonctionnement que H–LRF. La différence réside dans le fait que les profils de fonctionnement fournissant les débits les plus forts, $N_{0,j} = (\rho_{0,j}, RUL_{0,j})$, sont cette fois favorisés. Pour rappel, $\rho_{0,j} = \rho\max_j = \max_{0 \leqslant i < n} \rho_{i,j}$ et $RUL_{0,j} = RUL\min_j = \min_{0 \leqslant i < n} RUL_{i,j}$ pour toute machine M_j. Deux variantes de H–HTF peuvent être considérées. La première, H–HTFlt (pour Highest Throughput First, low throughput), sélectionne en premier les machines ayant les débits nominaux $\rho_{0,j}$ les plus petits et la seconde, H–HTFht (pour Highest Throughput First, high throughput), celles ayant les débits nominaux les plus forts. Pour les deux variantes, le plus petit sous-ensemble de machines permettant d'atteindre la demande σ est sélectionné. Si le débit atteint est supérieur à la demande ($\rho_{tot} > \sigma$), la contribution de la machine M_l dont le RUL associé au profil sélectionné (noté $N_{i,l}$) est le plus petit est diminuée, mais seulement si le débit total reste supérieure ou égal à la demande. Si possible, la machine M_l est alors utilisée avec le profil $N_{i+1,l}$ au lieu du profil $N_{i,l}$ précédemment sélectionné. Cela permet d'allonger la durée de vie de la machine M_l, et donc l'horizon de la solution. De même que pour l'heuristique H–LRF, le nombre de périodes pendant lequel la solution peut être appliquée est limité par la machine sélectionnée ayant le RUL le plus petit. Ce processus est répété tant que les machines restantes permettent d'atteindre la demande σ. Les différentes étapes sont décrites dans l'algorithme 6 et illustrées dans la figure 4.9 pour les deux variantes de l'heuristique, H–HTFlt et H–HTFlt.

Les solutions obtenue avec chacune des deux variantes de l'heuristique H–HTF sont détaillées sur la figure 4.10. Pour l'ensemble de machines considéré (voir figure 4.4), l'horizon atteint est le même ($\mathcal{H} = 5\Delta T$), mais l'engagement des machines est différent.

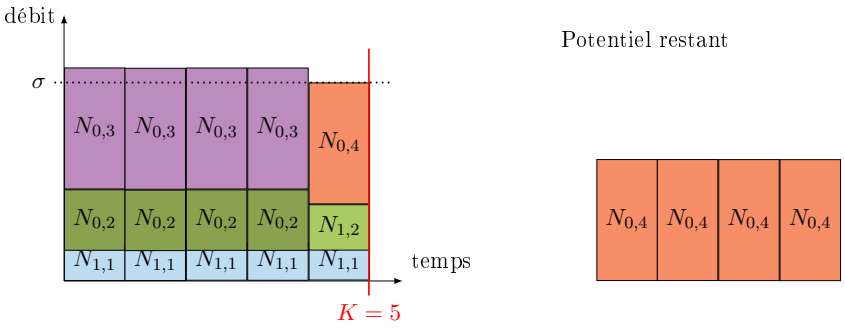


(a) Variante 1 : H–HTFlt

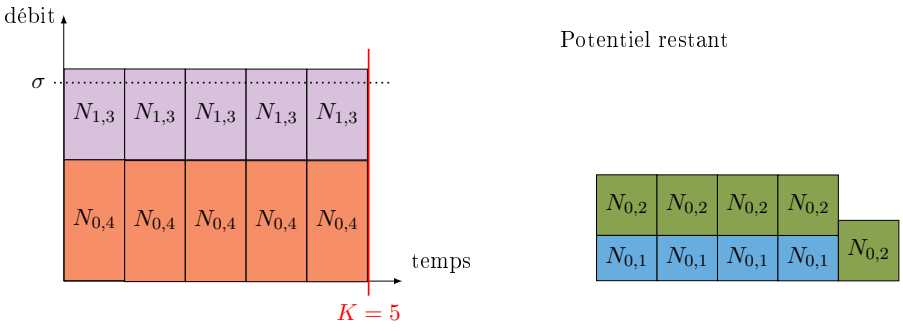


(b) Variante 2 : H–HTFht

Figure 4.9 – Illustration du principe de fonctionnement des deux variantes de l'heuristique H–HTF



(a) Variante 1 : H–HTFlt



(b) Variante 2 : H–HTFht

Figure 4.10 – Ordonnancements obtenus avec les deux variantes de l'heuristique H–HTF

Algorithme 6: H–HTF : heuristique favorisant les débits les plus forts

Données :
$list$: liste des couples (i,j) disponibles, tels que $RUL_{i,j}(k) \geqslant 1$ à la période k
sol_k : liste des couples (i,j) sélectionnés pour la période k
ρmax : débit maximal atteignable avec l'ensemble des machines disponibles
ρ_{tot} : débit total fourni par la solution sol_k
$RUL0[i][j]$: tableau contenant les valeurs initiales $RUL_{i,j}(1)$
$nbPeriodes$: nombre de périodes sur lequel la solution sol_k est appliquée
Entrées :
σ : demande à atteindre à chaque période k
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$RUL0[i][j] \leftarrow RUL_{i,j}(1),\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$RUL[i][j] \leftarrow RUL0[i][j],\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$k \leftarrow 1$
répéter
  $list \leftarrow$ liste mise à jour des couples (i,j) tels que $RUL[i][j] \geqslant 1$
  $sol_k \leftarrow \emptyset$
  $\rho\text{max} \leftarrow \sum_{j|(*,j)\in list} \max_{i|(i,j)\in list} \rho_{i,j}$
  $\rho_{tot} \leftarrow 0$
  si $\rho max \geqslant \sigma$ **alors**
    tant que $\rho_{tot} < \sigma$ **faire**
      `// Première variante H-HTFlt :`
      $sol_k \leftarrow sol_k \cup \{(i',j') \,|\, \rho_{i',j'} = \min_{j|(i,j)\in list} (\max_{0\leqslant i<n} \rho_{i,j})\}$
      `// Seconde variante H-HTFht :`
      $sol_k \leftarrow sol_k \cup \{(i',j') \,|\, \rho_{i',j'} = \max_{j|(i,j)\in list} (\max_{0\leqslant i<n} \rho_{i,j})\}$
      $\rho_{tot} \leftarrow \sum_{j|(i,j)\in sol_k} \rho_{i,j}$
    tant que $\rho_{tot} > \sigma$ *&* $list \neq \emptyset$ **faire**
      $j' \leftarrow j \,|\, RUL_{*,j} = \min_{j|(*,j)\in sol_k} RUL[*][j]$
      $i' \leftarrow i \,|\, (i,j') \in sol_k$
      si $\rho_{tot} - \rho_{i',j'} + \rho_{i'+1,j'} \geqslant \sigma$ **alors**
        $sol_k \leftarrow sol_k \setminus \{(i',j')\}$
        $sol_k \leftarrow sol_k \cup \{(i'+1,j')\}$
        $\rho_{tot} \leftarrow \sum_{j|(i,j)\in sol_k} \rho_{i,j}$
      $list \leftarrow list \setminus \{(i',j')\}$
    si $\rho_{tot} \geqslant \sigma$ **alors**
      $nbPeriodes \leftarrow \min_{j|(*,j)\in sol_k} RUL_{*,j}$
      pour $k' = k$ *à* $k + nbPeriodes$ **faire**
        $sol \leftarrow$ ajouter sol_k en fin de sol
      $RUL \leftarrow \text{UpdateRUL}(RUL0, RUL, sol_k, nbPeriodes)$ **(voir algorithme 2)**
      $k \leftarrow k + nbPeriodes$
jusqu'à $\rho_{tot} < \sigma$
retourner $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

4.3.1.5 H–DP : heuristique basée sur la programmation dynamique

L'heuristique H–DP (pour Dynamic Programming Heuristics) est plus élaborée que les précédentes. Pour chaque période de temps k et compte tenu des machines disponibles au début de la période, l'objectif est de trouver le meilleur ordonnancement des machines permettant d'atteindre au moins la demande σ, en minimisant la surproduction. Il s'agit alors de déterminer le meilleur sous-ensemble de machines à utiliser et, pour chacune d'entre elles, de sélectionner le profil de fonctionnement le plus adapté. L'algorithme permettant de faire ces choix est basé sur le principe de résolution d'un problème type "sac à dos". La principale différence avec ce problème classique d'optimisation combinatoire réside dans le fait que la somme des valeurs $(\rho_{i,j})$ des objets sélectionnés (M_j) doit être supérieure ou égale à la capacité du sac à dos (σ). Chaque objet M_j peut de plus être associé à différentes valeurs $\rho_{i,j}$ $(0 \leqslant i < n)$, une seule d'entre elles pouvant être sélectionnée dans la solution. L'objectif considéré est la minimisation de la somme des valeurs associées aux machines, dans le cas où cette somme dépasse le poids total maximum du sac à dos, σ.

L'algorithme 7 développé pour H–DP utilise l'approche de la programmation dynamique à deux dimensions. La recherche de solution peut être illustrée par le remplissage d'un tableau $2D$ (voir figure 4.11). Chaque machine disponible est successivement considérée, suivant un ordre croissant des débits nominaux $\rho_{0,j}$ des machines disponibles. Ce tri permet de limiter le nombre de solutions intermédiaires enregistrées et donc de minimiser à la fois la mémoire nécessaire et le temps d'exécution. Les machines dont les débits nominaux sont égaux sont de plus triées dans l'ordre décroissant de leur $RUL_{n-1,j}$. Chaque profil de fonctionnement de chaque machine est alors successivement considéré, du plus sous-nominal $(N_{n-1,j})$ au nominal $(N_{0,j})$. Le respect de cet ordre pour chaque recherche de solution permet d'user le parc de machines de façon homogène. Un certain turnover est en effet respecté, ce qui permet de conserver le maximum de machines disponibles différentes jusqu'à la fin de l'ordonnancement. Cela permet d'allonger l'horizon de production.

Pour chaque machine M_j, le débit total visé, σ', est incrémenté de 1 à σ. Pour chaque valeur successive de σ', chaque profil de fonctionnement disponible de M_j est considéré pour déterminer si la machine doit être sélectionnée ou non. Le test de toutes les possibilités permet de définir la meilleure configuration au regard de l'objectif. Une formulation mathématique récursive de construction de la solution optimale est définie par les équations (4.13), (4.14) et (4.15).

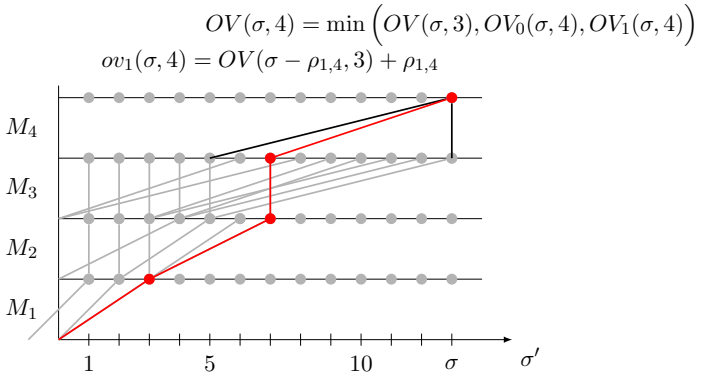


Figure 4.11 – Illustration du principe de fonctionnement de la programmation dynamique à deux dimensions

Soit $ov_i(\sigma', j)$ le débit global obtenu avec les j premières machines, en utilisant à la fois la j^e machine dans son i^e profil de fonctionnement et la configuration optimale obtenue avec les $j-1$ premières machines pour la demande $\sigma' - \rho_{i,j}$ (voir équation (4.13)). Soit $OV_i(\sigma', j)$ la variable contenant la valeur de $ov_i(\sigma', j)$ si elle atteint au moins la demande σ. Dans le cas contraire, $OV_i(\sigma', j) = +\infty$ (voir équation (4.14)). Cela permet de ne considérer que les solutions permettant d'atteindre au moins σ' à chaque étape, et donc σ pour la solution globale. Soit finalement $OV(\sigma', j)$ égal à la valeur du débit total optimal, c'est-à-dire un débit supérieur ou égal à la demande σ', obtenu avec les j premières machines (voir équation (4.15)). Le débit total optimal pour la période courante se retrouve à la position $OV(\sigma, m)$ de la matrice $2D$ OV utilisée par l'algorithme. Chaque choix effectué pour chaque couple (σ', j) est stocké au cours du processus de recherche de solution. Cela permet de reconstruire la solution pour la demande $\sigma' = \sigma$. En cas de solutions équivalentes, l'algorithme choisit celle qui implique le nombre minimal de machines différentes.

$$ov_i(\sigma', j) = OV(\sigma' - \rho_{i,j}, j-1) + \rho_{i,j} \quad \text{avec} \quad 1 \leqslant i \leqslant n-1 \tag{4.13}$$

$$OV_i(\sigma', j) = \begin{cases} ov_i(\sigma', j) \ \text{ si } \ ov_i(\sigma', j) \geqslant \sigma' \\ +\infty \ \text{ sinon} \end{cases} \tag{4.14}$$

$$OV(\sigma', j) = \min\left(OV(\sigma', j-1), \min_{0 \leqslant i \leqslant n-1} OV_i(\sigma', j)\right) \tag{4.15}$$

Un exemple de solution est donné sur la figure 4.12. On peut voir que le processus de sélection mis en œuvre par l'heuristique H–DP minimise la surproduction le plus longtemps possible. La solution appliquée à chaque période de temps est optimale considérant l'état de santé de l'ensemble des machines au début de la période. L'ordonnancement global n'est cependant pas nécessairement optimal. Cela peut être vu dans la partie 4.4.4, dans laquelle les résultats obtenus avec les meilleures heuristiques sont comparés aux solutions optimales.

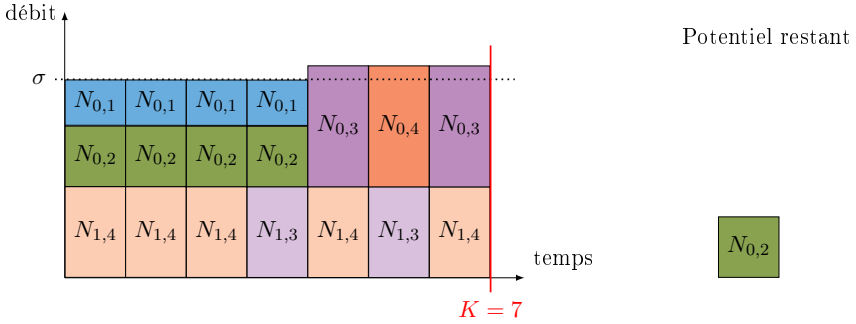


Figure 4.12 – Ordonnancement obtenu avec l'heuristique H–DP

Algorithme 7: H–DP : heuristique basée sur la programmation dynamique

Données :
$list$ (resp. $listeMachTriee$) : couples (i,j) disponibles (resp. triés)
m_k : nombre de machines disponibles à la période k (avec $RUL_{i,j}(k) \geqslant 1$)
sol_k : liste des couples (i,j) sélectionnés pour la demande σ pour la période k
$sol_{\sigma',j}$: couples (i,j) sélectionnés parmi les j premières machines de $listeMachtriee$ pour σ'
ρmax : débit maximal atteignable avec l'ensemble des machines disponibles
ρ_{tot} : débit total fourni par la solution sol_k
$RUL[i][j]$ (resp. $RUL0[i][j]$) (resp. initiales) : valeurs $RUL_{i,j}$ pour la période k
$nbPeriodes$: nombre de périodes sur lequel la solution $sol_{k,\sigma}$ est appliquée
σ_{init} : première demande considérée pour chaque machine
σ_{cumul} : dernière demande considérée pour chaque machine
Entrées :
σ : demande à atteindre à chaque période k
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$RUL0[i][j] \leftarrow RUL_{i,j}(1)$; $RUL[i][j] \leftarrow RUL0[i][j]\ \forall\ 0 \leqslant i \leqslant n-1,\ \forall\ 1 \leqslant j \leqslant m$
$nbPeriodes \leftarrow 1$; $k \leftarrow 1$
répéter
  $list \leftarrow$ liste mise à jour des couples (i,j) disponibles, tels que $RUL[i][j] \geqslant 1$
  $sol_k \leftarrow \emptyset$; $\rho_{tot} \leftarrow 0$
  $\rho\text{max} \leftarrow \sum_{1 \leqslant j \leqslant m} \max_{0 \leqslant i \leqslant n-1} \rho_{i,j}$
  si $\rho max \geqslant \sigma$ **alors**
    $listeMachTriee \leftarrow$ couples $(*,j)$ disponibles, triés dans l'ordre croissant de leur débit le plus nominal $\rho_{*,j}$ et dans l'ordre décroissant de leur $RUL_{n-1,j}$ en cas de débits égaux
    `// Les éléments de` $listeMachTriee$ `sont renumérotés de 1 à` m_k`. Une trace des permutations effectuées au cours du tri est conservée pour permettre la reconstruction de la solution à la fin du processus d'ordonnancement`
    $\sigma_{init} \leftarrow 1$
    pour chaque $j = 1$ *à* m_k **faire**
      si $j = m_k$ **alors** $\sigma_{init} \leftarrow \sigma$; $\sigma_{cumul} \leftarrow \sigma$**sinon** $\sigma_{cumul} \leftarrow \sum_{1 \leqslant l \leqslant j} \max_{0 \leqslant i < n} \rho_{i,l}$
      pour $\sigma' = \sigma_{init}$ *à* $\min(\sigma_{cumul}, \sigma)$ **faire**
        `//` Hypothèse : $OV(s,*) = 0$ `et` $sol_{s,*} = \emptyset\ \forall\ s \leqslant 0$
        $OV(\sigma', j) \leftarrow OV(\sigma', j-1)$ pour $j > 1$, 0 pour $j = 1$
        $sol_{\sigma',j} \leftarrow sol_{\sigma',j-1}$ pour $j > 1$, $\emptyset$ pour $j = 1$
        pour $i = n-1$ *à* 0 **faire**
          si $(OV(\sigma' - \rho_{i,j}, j-1) < OV(\sigma', j))$ // $(OV(\sigma' - \rho_{i,j}, j-1) + \rho_{i,j} = OV(\sigma', j)$ & $|sol_{\sigma'-\rho_{i,j},j-1}| + 1 < |sol_{\sigma',j}|)$ **alors**
            $OV(\sigma', j) \leftarrow OV(\sigma' - \rho_{i,j}, j-1) + \rho_{i,j}$
            $sol_{\sigma',j} \leftarrow sol_{\sigma'-\rho_{i,j},j-1} \cup \{(i,j)\}$
    $\rho_{tot} \leftarrow \sum_{j|(i,j) \in sol_{\sigma,m_k}} \rho_{i,j}$
    si $\rho_{tot} \geqslant \sigma$ **alors**
      $sol_k \leftarrow sol_{\sigma,m_k}$
      $RUL \leftarrow$ UpdateRUL$(RUL0, RUL, sol_k, nbPeriodes)$ **(voir algorithme 2)**
      $sol \leftarrow$ ajouter sol_k en fin de sol
      $k{+}{+}$
jusqu'à $\rho_{tot} < \sigma$
retourner $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

4.3.2 Amélioration des heuristiques

Les résultats développés dans la partie 4.4 montrent que les heuristiques proposées jusqu'ici ne permettent pas d'obtenir des solutions optimales. Les horizons de production atteints sont en effet en moyenne à 82% de l'optimal pour l'heuristique donnant les meilleurs résultats. Une amélioration des solutions obtenues est alors encore possible. On peut de plus observer un potentiel résiduel à la fin de la plupart des ordonnancements : toutes les machines ne sont pas utilisées jusqu'à leur fin de vie. Certaines d'entre elles peuvent donc toujours fournir un certain débit, sans que cela soit suffisant pour atteindre la demande pour une période supplémentaire. Le principe de la réparation proposée ici repose sur l'utilisation de ce potentiel restant pour tenter d'étendre les horizons de production des solutions données par les heuristiques proposées précédemment.

Nous proposons d'améliorer les heuristiques proposées dans la partie 4.3.1 en modifiant les ordonnancements obtenus avec ces stratégies d'engagement des machines. Les ordonnancements étant construits hors-ligne, cette étape de modification peut intervenir après la construction d'un premier ordonnancement. Le principe de réparation est tout d'abord motivé avec un exemple très simple. La stratégie mise en œuvre est ensuite détaillée.

4.3.2.1 Illustration du principe de réparation

Le principe de réparation peut être expliqué sur un exemple mettant en jeu trois machines pouvant être utilisées avec un seul profil de fonctionnement. L'ordonnancement obtenu avec l'heuristique basée sur la programmation dynamique, H–DP, est montré sur la figure 4.13a. On peut voir que la machine M_3 n'est jamais utilisée au cours de cet ordonnancement. Il y a donc un potentiel restant, mais aucune période ne peut être ajoutée dans l'état. La machine M_3 n'est en effet pas assez puissante pour atteindre seule la demande σ. Il faudrait utiliser la même machine en parallèle avec elle-même, ce qui n'est pas possible.

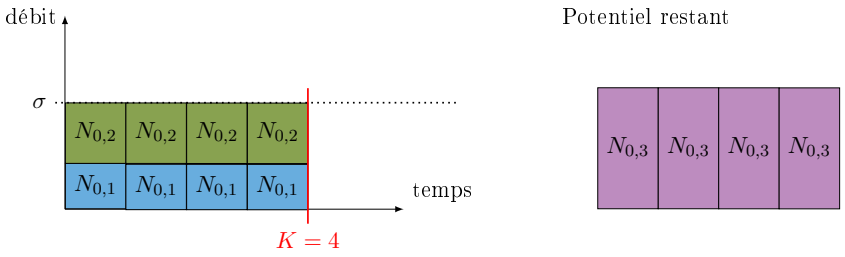
Il y a malgré tout une autre possibilité pour utiliser cette machine M_3. Cette dernière n'étant pas utilisée dans la première période de l'ordonnancement ($0 \leqslant t \leqslant \Delta T$), elle peut être échangée avec la machine M_2 pour une période. Cela entraîne une surproduction au cours de cette première période, mais permet d'avoir deux machines différentes disponibles à la fin de l'ordonnancement. La demande peut ainsi être satisfaite pour une période supplémentaire en utilisant les machines M_2 et M_3 en parallèle (voir figure 4.13b). Le même échange peut être effectué dans la seconde période de l'ordonnancement. Cela permet de récupérer la machine M_2 pour une période et d'accroître à nouveau le nombre de périodes complétées de 1 (voir la figure 4.13c). Sur cet exemple, appliquer le principe de la réparation permet ainsi d'utiliser entièrement le potentiel offert par l'ensemble des machines et d'étendre l'horizon de production de $\mathcal{H} = 4\Delta T$ à $\mathcal{H} = 6\Delta T$.

4.3.2.2 Stratégie de réparation

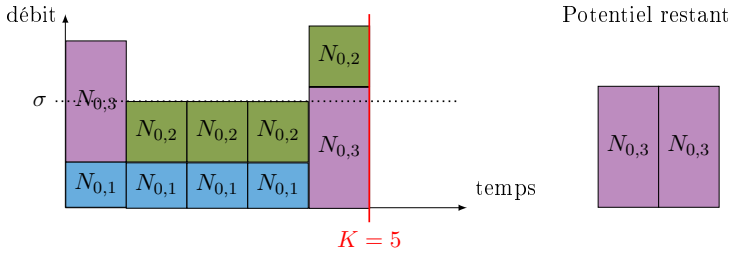
La réparation est basée sur un algorithme glouton et peut être appliquée à la solution de n'importe laquelle des heuristiques précédemment définies, les seules restrictions étant les suivantes :

(i) l'ordonnancement doit être constitué d'au moins une période de temps ($K \geqslant 1$) ;
(ii) au moins une machine doit rester avec un $RUL_{n-1,j} \geqslant 1$;

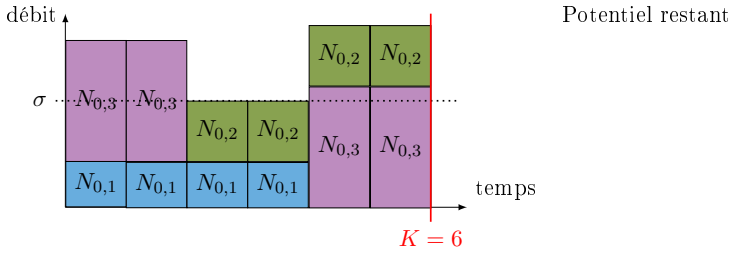
(*iii*) la somme des RUL restantes doit être au moins égale à 2 : $\sum_{1 \leqslant j \leqslant m} RUL_{n-1,j} \geqslant 2$ à la fin de l'horizon courant $K\Delta T$.



(a) Ordonnancement initial, obtenu avec l'heuristique H–DP



(b) Première étape de réparation



(c) Seconde étape de réparation, ordonnancement obtenu avec l'heuristique H–DP-R

Figure 4.13 – Stratégie de réparation

Le principe de base est d'échanger les machines restantes, dont le RUL est supérieur à 1 à la fin de l'ordonnancement, contre une ou plusieurs autres machines utilisées dans une période de l'ordonnancement initial et dont le RUL est nul pour $k > K$, avec $K\Delta T$ l'horizon de production atteint par l'ordonnancement initial. La récupération de ces machines permet d'accroître le nombre de machines différentes restant à la fin de l'ordonnancement. Ces machines étant différentes, elles peuvent être utilisées en parallèles et ainsi permettre d'atteindre potentiellement la demande σ pour une ou plusieurs période(s) supplémentaire(s).

Soient les ensembles suivants :

$\mathscr{M}\mathbf{Rest} = \{M_j \,|\, RUL_{n-1,j}(K+1) > 0,\ 1 \leqslant j \leqslant m\}$ l'ensemble des machines M_j pouvant encore être utilisées à la fin de l'ordonnancement initial ;

$\mathscr{M}\overline{\mathbf{Rest}} = \{M_j \,|\, RUL_{n-1,j}(K+1) = 0,\ 1 \leqslant j \leqslant m\}$ l'ensemble des machines M_j ayant atteint leur fin d'utilisation avant maintenance ;

$\mathscr{M}\mathbf{Rest}_{k\mathbf{swap}} = \{M_j \in \mathscr{M}\mathrm{Rest} \,|\, (*, j) \notin sol_{k\mathrm{swap}} \text{ et } RUL_{n-1,j}(K+1) \geqslant RUL_{n-1,j+1}(K+1)\ \forall\ 1 \leqslant j \leqslant m\}$ l'ensemble des machines restantes non utilisées à la période $k$swap, triées par ordre décroissant de leur RUL ;

$\mathscr{M}\overline{\mathbf{Rest}}_{k\mathbf{swap}} = \{M_j \in \mathscr{M}\overline{\mathrm{Rest}} \,|\, (*, j) \in sol_{k\mathrm{swap}}\}$ l'ensemble des machines utilisées à la pé-

riode $k$swap ayant atteint leur fin d'utilisation avant maintenance.

La réparation consiste à trouver les périodes $k$swap de l'ordonnancement à réparer (avec $1 \leqslant k\text{swap} \leqslant K$) dans lesquelles un échange peut être fait, c'est-à-dire les périodes répondant au critère suivant :

$$\sum_{j|M_j \in \mathscr{M}\text{Rest}_{k\text{swap}}} \max_{i|RUL_{i,j}>0} \rho_{i,j} \geqslant \left(\sum_{j|M_j \in \mathscr{M}\overline{\text{Rest}}_{k\text{swap}}} \rho_{*,j} \right) - surplus \tag{4.16}$$

$$\text{avec} \quad surplus = \left(\sum_{j|(i,j) \in sol_{k\text{swap}}} \rho_{i,j} \right) - \sigma$$

et avec $\rho_{*,j}$ le débit avec lequel la machine M_j est utilisée à la période $k$swap et $\max_{i|RUL_{i,j}>0} \rho_{i,j}$ le débit maximal encore disponible pour la machine M_j de $\mathscr{M}\text{Rest}_{k\text{swap}}$.

Les périodes sont testées dans l'ordre défini par l'ordonnancement initial devant être réparé. Pour chaque période $k$swap, il s'agit d'échanger le nombre de machines de $\mathscr{M}\text{Rest}_{k\text{swap}}$ juste suffisant pour récupérer les machines de $\mathscr{M}\overline{\text{Rest}}_{k\text{swap}}$. L'ensemble de machines $\mathscr{M}\overline{\text{Rest}}_{k\text{swap}}$ est alors remplacé dans l'ordonnancement à la période $k$swap par l'ensemble de machines $\mathscr{M}$RestSwap défini de la manière suivante :

$$\mathscr{M}\textbf{RestSwap} = \{M_j \in \mathscr{M}\text{Rest}_{k\text{swap}} \mid \sum_{j|M_j \in \mathscr{M}\text{Rest}_{k\text{swap}}} \max_{i|RUL_{i,j}>0} \rho_{i,j} \geqslant \left(\sum_{j|M_j \in \mathscr{M}\overline{\text{Rest}}_{k\text{swap}}} \rho_{*,j} \right) - surplus\}.$$

Dès que le potentiel total des machines restantes est suffisant pour atteindre la demande σ, l'heuristique avec laquelle la solution initiale a été construite est réutilisée pour ajouter une période à l'ordonnancement. Tout le processus est reconduit tant qu'une période additionnelle peut être ajoutée à l'ordonnancement (voir les algorithmes 9 et 10). La réparation est stoppée lorsque plus aucun échange utile de machines ne peut être effectué.

Algorithme 8: PossReparation : retourne *VRAI* si une réparation peut être effectuée

Entrées :
K : nombre de périodes de l'ordonnancement à réparer
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}), 0 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant m\}$: caractéristiques des profils de fonctionnement pour chaque machine M_j
Sorties :
VRAI si une réparation est possible, *FAUX* sinon

si *($K \geqslant 1$) et ($\sum_{j|M_j \in \mathscr{M}} RUL_{n-1,j} \geqslant 2$)* **alors**
  retourner *VRAI*
sinon
  retourner *FAUX*

La réparation est appliquée sur les trois heuristiques les plus performantes, à savoir H–LRF, H–HTF et H–DP. En utilisant successivement le principe de chacune de ces heuristiques et l'étape de réparation, on obtient trois nouvelles heuristiques : H–LRF-R, H–HTF-R et H–DP-R. Les performances de toutes les heuristiques proposées dans cette partie sont comparées dans la partie suivante. La qualité des solutions est appréciée en comparant les horizons de production atteints à la borne supérieure définie précédemment pour le problème MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$).

Algorithme 9: H–*-R : heuristiques avec réparation – Implémentation globale

Données :
sol_k : liste des couples (i, j) sélectionnés pour la période k
Kinit : horizon de l'ordonnancement à réparer
$\mathscr{M}$Rest : liste des machines restantes
$\mathscr{M}\overline{\text{Rest}}$: liste des machines ayant atteint leur fin de vie
rep : booléen permettant de ne lancer le processus de réparation qu'après un premier lancement de l'heuristique associée
$first$: booléen permettant de n'initialiser qu'une seule fois l'ensemble des machines restantes, après le premier lancement de l'heuristique associée
$poss$: booléen indiquant si une réparation peut être effectuée
$swap$: booléen indiquant si au moins un échange a été effectué durant la réparation
Entrées :
σ : demande à atteindre à chaque période k
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

$rep \leftarrow FAUX$
$first \leftarrow VRAI$
$poss \leftarrow VRAI$
$swap \leftarrow VRAI$
répéter
  si *rep est VRAI* **alors**
    si *first est VRAI* **alors**
      Kinit $\leftarrow$ taille de sol
      $\mathscr{M}$Rest $\leftarrow \{M_j | RUL_{n-1,j}(K\text{init}+1) \geqslant RUL_{n-1,j+1}(K\text{init}+1) \geqslant 1 \quad \forall\ 1 \leqslant j \leqslant m-1\}$
      $\mathscr{M}\overline{\text{Rest}} \leftarrow \{M_j | RUL_{n-1,j}(K\text{init}+1) = 0 \quad \forall\ 1 \leqslant j \leqslant m-1\}$
      $first \leftarrow FAUX$
    sinon
      $\mathscr{M}$Rest $\leftarrow \{M_j | RUL_{n-1,j}(K\text{init}+1) \geqslant RUL_{n-1,j+1}(K\text{init}+1) \geqslant 1 \quad \forall\ 1 \leqslant j \leqslant m-1\}$
    $K \leftarrow$ taille de sol
    si *(PossReparation*$(K, \mathcal{N})$ *est VRAI) & (poss est VRAI)* **alors**
      $(swap, sol) \leftarrow Reparation(\sigma, sol, \mathcal{N}, \mathscr{M}\text{Rest}, \mathscr{M}\overline{\text{Rest}}, K\text{init})$**(voir algorithme 10)**
      $poss \leftarrow PossReparation(K, \mathcal{N})$**(voir algorithme 8)**
    sinon
      $poss \leftarrow FAUX$
  $sol \leftarrow$ H–*$(\sigma, \mathcal{N})$ **(voir algorithme 3, 4, 5, 6 ou 7)**
  $rep \leftarrow VRAI$
jusqu'à *(poss est FAUX) ou (swap est FAUX)*
retourner $sol = (sol_1, sol_2, \ldots, sol_K)$

Algorithme 10: *Reparation*$(\sigma, sol, \mathcal{N}, \mathscr{M}\text{Rest}, \mathscr{M}\overline{\text{Rest}}, K\text{init})$ – Échange de machines

Données :
$swap$ booléen indiquant si un échange de machines a été effectué
$k$swap : période testée
sol_k : liste des couples (i, j) sélectionnés pour la période k
$\mathscr{M}\text{Rest}_{k\text{swap}}$: liste des machines restantes non utilisées dans la période $k$swap
$\mathscr{M}\overline{\text{Rest}}_{k\text{swap}}$: liste des machines ayant atteint leur fin de vie et utilisée dans la période $k$swap
$\mathscr{M}\text{RestSwap}$: liste des machines restantes à échanger
$\rho\text{maxRest}_{\text{swap}}$: débit maximal atteignable avec les machines de $\mathscr{M}\text{Rest}_{k\text{swap}}$
$\rho\text{max}_{\text{swap}}$: débit maximal atteignable avec les machines de $\mathscr{M}\overline{\text{Rest}}_{k\text{swap}}$
$surplus$: surplus de production dans la période courante
Entrées :
σ : demande à atteindre à chaque période k
$sol = (sol_1, sol_2, \ldots, sol_K)$: ordonnancement initial
$\mathcal{N} = \{N_{i,j} = (\rho_{i,j}, RUL_{i,j}),\ 0 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m\}$: caractéristiques initiales des profils de fonctionnement pour chaque machine M_j
Kinit : horizon de l'ordonnancement initial
$\mathscr{M}\text{Rest}$: liste des machines restantes
$\mathscr{M}\overline{\text{Rest}}$: liste des machines restantes ayant atteint leur fin de vie
Kinit : horizon de l'ordonnancement à réparer
Sorties :
$swap$: $VRAI$ si au moins un échange de machines a été effectué
$sol = (sol_1, sol_2, \ldots, sol_K)$: ordonnancement réparé

$swap \leftarrow FAUX$
$k$swap $\leftarrow 1$
tant que *(kswap < Kinit) & (swap est FAUX)* **faire**
  $\mathscr{M}\text{Rest}_{k\text{swap}} \leftarrow \{M_j \in \mathscr{M}\text{Rest} \,|\, (*, j) \notin sol_{k\text{swap}} \text{ et } RUL_{n-1,j}(K+1) \geqslant RUL_{n-1,j+1}(K+1)\ \forall\ 1 \leqslant j \leqslant m\}$
  si $\mathscr{M}Rest_{kswap} \neq \emptyset$ **alors**
    $\mathscr{M}\overline{\text{Rest}}_{k\text{swap}} \leftarrow \{M_j \in \mathscr{M}\overline{\text{Rest}} \,|\, (*, j) \in sol_{k\text{swap}}\}$
    pour chaque $M_j \in \mathscr{M}Rest_{kswap}$ **faire**
      $\rho\text{maxRest}_{\text{swap}} \leftarrow \sum_{j|M_j \in \mathscr{M}\text{Rest}_{k\text{swap}}} \max_{i|RUL_{i,j}>0} \rho_{i,j}$
      $\rho\text{max}_{\text{swap}} \leftarrow \sum_{j|M_j \in \mathscr{M}\overline{\text{Rest}}_{k\text{swap}}} \rho_{i,j}$ avec $i|(i,j) \in sol_{k\text{swap}}$
      $surplus \leftarrow \left(\sum_{j|(i,j) \in sol_{k\text{swap}}} \rho_{i,j}\right) - \sigma$
      si $\rho maxRest_{swap} \geqslant \rho max_{swap} - surplus$ **alors**
        $\mathscr{M}\text{RestSwap} \leftarrow \{M_j \in \mathscr{M}\text{Rest}_{k\text{swap}} \,|\, \sum_{j|M_j \in \mathscr{M}\text{Rest}_{k\text{swap}}} \max_{i|RUL_{i,j}>0} \rho_{i,j} \geqslant \left(\sum_{j|M_j \in \mathscr{M}\overline{\text{Rest}}_{k\text{swap}}} \rho_{*,j}\right) - surplus\}$
        $sol_k \leftarrow sol_k \setminus \{(*, j) \,|\, M_j \in \mathscr{M}\overline{\text{Rest}}_{k\text{swap}}\}$
        $sol_k \leftarrow sol_k \cup \{(*, j) \,|\, M_j \in \mathscr{M}\text{RestSwap}\}$
        $swap \leftarrow VRAI$
  $k$swap $\leftarrow k$swap $+ 1$
retourner *(swap, sol)*

4.4 Résultats de simulation

Les approches de résolution proposées dans ce chapitre sont testées sur plusieurs instances du problème d'optimisation MAXK(σ, $\rho_{i,j}$, $RUL_{i,j}$). Chaque instance est générée à l'aide d'un simulateur et configurée avec différents paramètres. La génération et le paramétrage des problèmes traités sont tout d'abord détaillés. L'efficacité des différentes heuristiques proposées dans la partie 4.3.1 est ensuite comparée pour des demandes σ constantes dans le temps, sur la base des horizons de production $\mathcal{H} = K\Delta T$ atteints. Les résultats obtenus avec les heuristiques les plus performantes sont ensuite comparés à l'optimal pour les problèmes pour lesquels le programme linéaire défini dans la partie 4.2 peut fournir une solution en temps raisonnable. Une comparaison à une borne supérieure est ensuite proposée pour des problèmes plus grands. La robustesse de l'approche quant à la variabilité de la demande est finalement montrée sur des instances de problème considérant une demande σ_k constante par morceaux.

4.4.1 Génération des problèmes

Pour chaque problème d'optimisation, plusieurs configurations sont générées pour servir de base à l'application des stratégies mises en œuvre par les différentes heuristiques. Chaque configuration correspond à une plate-forme de machines et est paramétrée avec différents éléments.

4.4.1.1 Paramétrage des plates-formes de machines

Les paramètres principaux sont le nombre m de machines constituant la plate-forme et le nombre n de profils de fonctionnement avec lequel chaque machine peut être utilisée. Les machines d'une même plate-forme étant supposées être du même type, elles ont toutes le même nombre de profils de fonctionnement. Conformément au modèle suivi dans ce chapitre, ce nombre est supposé fini. Pour les résultats proposés dans la suite, le rendement $Q_{i,j}$ de chaque machine M_j, défini pour chaque profil de fonctionnement comme le produit de débit et du RUL associé ($Q_{i,j} = \rho_{i,j} \cdot RUL_{i,j}$), évolue dans le même sens que le débit. Dans ce cas, $Q_{0,j} > Q_{1,j} > \ldots > Q_{n-1,j}$ $\forall\ 1 \leqslant j \leqslant m$, avec $\rho_{0,j} > \rho_{1,j} > \ldots > \rho_{n-1,j}$ et $RUL_{0,j} < RUL_{1,j} < \ldots < RUL_{n-1,j}$. Pour chaque machine, le profil de fonctionnement nominal, fournissant le débit maximal et associé au RUL le plus faible, est donc le plus performant en termes de rendement.

Les m machines prises en compte pour chaque simulation sont sélectionnées aléatoirement parmi un ensemble de machines types, composé de machines à faible débit nominal et de machines à fort débit nominal. La sélection est régie par un pourcentage P de machines à faible débit nominal de la plate-forme. Plusieurs machines d'une même plate-forme peuvent fournir les mêmes niveaux de débit. Elles sont toutefois différenciées par leurs valeurs de RUL. Les machines peuvent en effet présenter un état de santé différent au début de l'ordonnancement (pour $t = 0$). Les valeurs de paramètres prises en compte pour l'obtention des résultats présentés dans la suite sont les suivantes : $m \in \{10, 25, 50\}$, $n \in \{1, 2, 5, 10\}$ et $P \in \{0\%, 20\%, 50\%, 80\%, 100\%\}$.

4.4.1.2 Paramétrage de la demande

La comparaison des heuristiques est tout d'abord faite en considérant une demande σ constante durant tout l'horizon de production. Une seule valeur σ a donc été associée à chaque

configuration du problème d'optimisation. Différentes demandes correspondant à différentes instances de problème ont toutefois été testées. Leurs valeurs ont été fixées suivant la relation définie par l'équation (4.17), avec $\rho\max_{tot}$ le débit total maximal pouvant être généré avec l'ensemble de machines considéré et χ une charge variant entre 30% et 90%.

$$\sigma = \chi \cdot \rho\max_{tot} \tag{4.17}$$
$$\text{avec} \quad \rho\max_{tot} = \sum_{j=1}^{m} \max_{0 \leqslant i < n} \rho_{i,j}$$

Sur les graphes suivants, les résultats sont représentés en fonction de la charge $\chi = \sigma / \rho\max_{tot}$. Chaque point représenté correspond à la moyenne des résultats de 20 instances différentes de la configuration du problème d'optimisation considérée. Afin de faciliter la lecture des résultats, les différents points représentés pour une même charge χ ont été dispersés autour de l'abscisse correspondante.

4.4.2 Résultats préliminaires

Sur la base de différents tests menés sur plusieurs ensembles de machines, il apparaît que les résultats obtenus ne varient pas de façon significative avec le nombre de machines m. Il en est de même avec le paramètre P, qui détermine le pourcentage de machines de faible débit dans la plate-forme considérée. Les résultats présentés dans la suite ont été obtenus avec $m = 25$ machines et $P = 50\%$. Si aucune précision n'est apportée, les remarques et conclusions données restent valables pour toutes les valeurs de m et de P.

Les deux variantes de l'heuristique H–HTF sont aussi performantes l'une que l'autre. On peut en effet voir sur la figure 4.14 que la distance des résultats obtenus avec H–HTFlt à ceux obtenus avec H–HTFht est inférieure à 2.5% pour $m = 25$ machines et différents nombres de profils de fonctionnement ($n = \{1, 2, 5, 10\}$). Cette distance reste inférieure à 16% pour tous les cas de figure testés. Seule la version sélectionnant en priorité les machines fournissant les débits les plus forts, H–HTFht, est alors considérée pour la suite de l'étude et est nommée H–HTF.

4.4.3 Comparaison des heuristiques

Les performances en termes d'horizon de production atteint, K, des heuristiques sans réparation sont tout d'abord comparées. L'amélioration obtenue avec la stratégie de réparation des ordonnancements obtenus avec ces premières heuristiques est ensuite étudiée pour les meilleures d'entre elles.

4.4.3.1 Heuristiques sans réparation

Il s'avère tout d'abord que la sélection aléatoire des machines et des profils de fonctionnement effectuée par l'heuristique H–RAND permet de définir des ordonnancements assez performants en termes d'horizon de production atteint lorsqu'un seul profil de fonctionnement est considéré pour chaque machine ($n = 1$). Ceci peut être expliqué par le fait que cette heuristique ne sélectionne

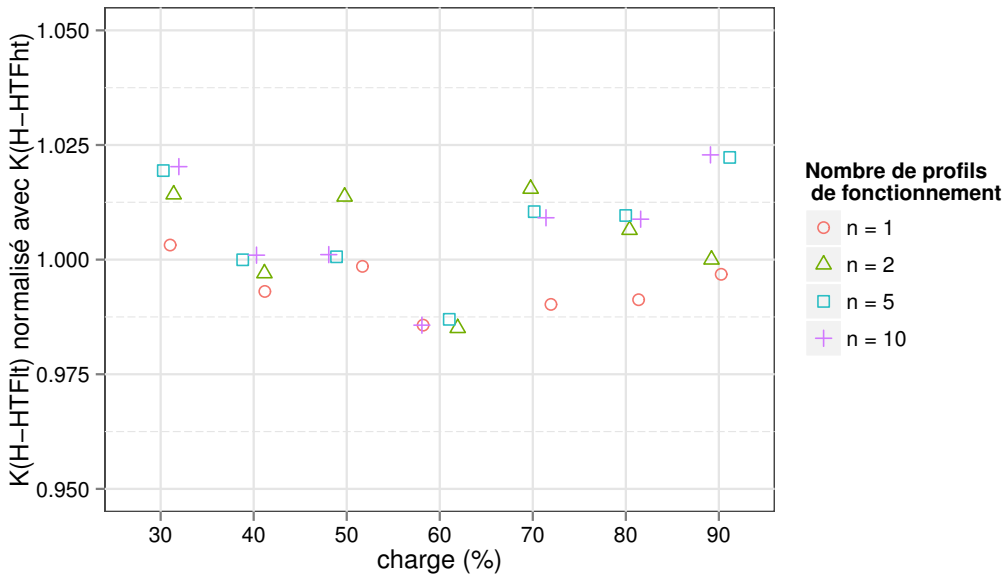


Figure 4.14 – Comparaison des résultats obtenus avec les deux variantes de l'heuristique H–HTF pour $m = 25$ machines et différents nombres de profils de fonctionnement ($n = \{1, 2, 5, 10\}$)

que le nombre de machines strictement nécessaire pour atteindre la demande. La surproduction maximale est alors égale à $\rho\max - 1$, avec $\rho\max = \max_{1 \leqslant j \leqslant m} \rho_{0,j}$ le débit maximal pouvant être fourni par une machine de la plate-forme. H–RAND n'est toutefois pas fiable lorsque plus d'un profil de fonctionnement est considéré ($n > 1$). Le nombre de possibilités pour le choix des couples machine/profil de fonctionnement croît en effet avec n. Si plusieurs machines sont sélectionnées avec un profil de fonctionnement fournissant un débit trop faible, un grand nombre de machines est nécessaire pour atteindre la demande σ. A un instant du processus de sélection, les machines restantes ne permettent plus d'atteindre σ, même si elles sont sélectionnées dans leur profil nominal fournissant le plus fort débit. Le nombre de périodes atteint avec H–RAND décroît fortement pour les charges χ supérieures à 50% dès que deux profils de fonctionnement sont considérés (voir figure 4.15).

La seconde heuristique basique, H–NAIVE, met en œuvre un processus de sélection un peu plus élaboré que H–RAND. Le nombre de périodes complétées est alors plus élevé. Pour les charges χ inférieurs à 50%, plusieurs groupes de machines peuvent être formés et les horizons atteints sont meilleurs que ceux obtenus avec H–LRF pour les charges χ inférieures ou égales à 40% (voir figure 4.16). H–NAIVE ne sélectionne en effet que les profils de fonctionnement nominaux, qui fournissent les débits les plus forts, alors que H–LRF favorise les profils sous-nominaux. Pour les charges faibles, les débits faibles associés aux profils sous-nominaux sont suffisants pour atteindre la demande et sont donc sélectionnés par H–LRF. Cela entraîne une sélection d'un nombre plus important de machines pour chaque période de temps. Pour des charges χ supérieures ou égales à 50%, l'heuristique H–NAIVE devient moins performante. En effet, pour ces charges plus importantes, au moins la moitié des machines disponible ($m/2$) doivent être utilisées pour atteindre la demande. Dans la plupart des cas, $m/2+1$ machines sont nécessaires. Cela est dû au fait que les machines peuvent fournir des débits différents, qui peuvent être plus faibles que la valeur σ/m. La réutilisation des machines n'étant pas gérée par l'heuristique H–NAIVE

même si leur RUL restant est supérieur à 1 après leur première utilisation, le nombre de périodes complétées K n'est pas très élevé pour des charges χ supérieures ou égales à 50%.

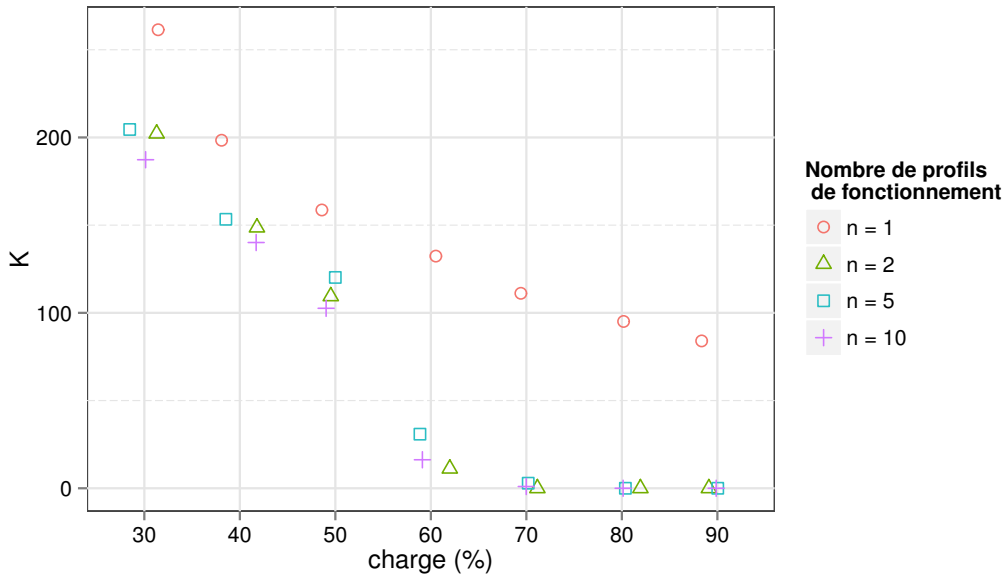


Figure 4.15 – Résultats obtenus avec l'heuristique basique H–RAND pour $m = 25$ machines et différents nombres de profils de fonctionnement ($n = \{1, 2, 5, 10\}$)

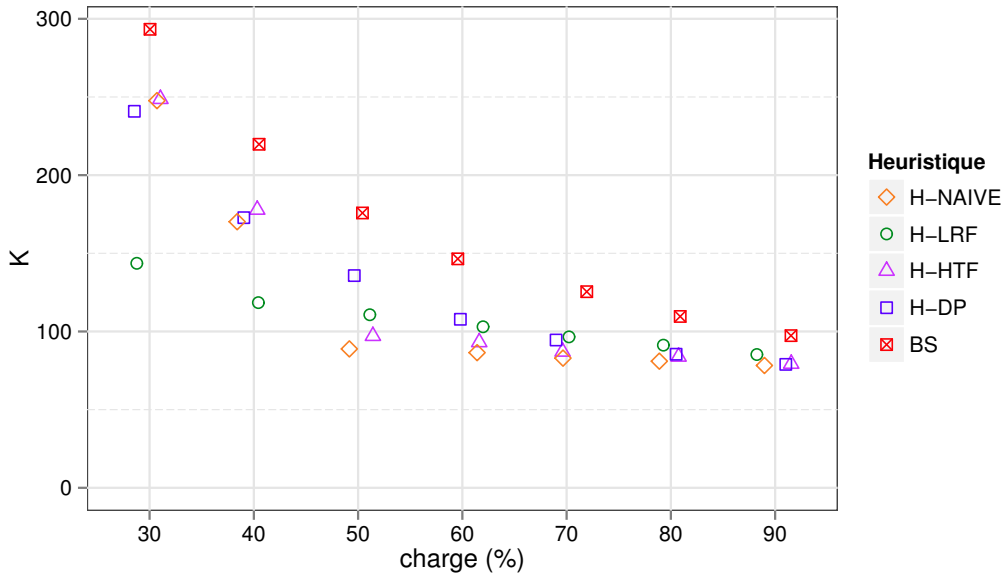


Figure 4.16 – Nombre moyen de périodes complétées (K) en fonction de la charge χ – $m = 25$ machines, $n = 5$ profils de fonctionnement

Les résultats obtenus avec les trois autres heuristiques, H–LRF, H–HTF et H–DP, sont ensuite comparés sur la figure 4.16, sur laquelle est aussi représentée la borne supérieure définie dans la partie 4.1.1, BS(σ, $\rho_{i,j}$, $RUL_{i,j}$). Cette borne décroît lorsque la charge augmente. Une charge élevée correspond à une demande σ élevée. Ainsi, plus la charge est élevée, plus le nombre de machines nécessaires pour atteindre la demande est important et le nombre de périodes

pouvant être complétées est faible.

La variation du nombre m de machines, ainsi que celle du nombre n de profils de fonctionnement n'ont pas d'effet significatif sur les résultats obtenus avec l'heuristique H–HTF. Cette heuristique favorise en effet les profils de fonctionnement nominaux. Pour une même machine M_j, le profil nominal $N_{0,j}$ est toujours associé au même débit $\rho_{0,j}$ et à la même valeur de RUL, $RUL_{0,j}$, quelque soit le nombre de profils de fonctionnement considéré. Utiliser les machines avec différents profils de fonctionnement durant leur durée de vie peut toutefois être intéressant. La figure 4.17 montre en effet que l'horizon de production atteint augmente avec le nombre de profils de fonctionnement lorsque l'heuristique H–LRF, qui favorise les profils sous-nominaux, est utilisée.

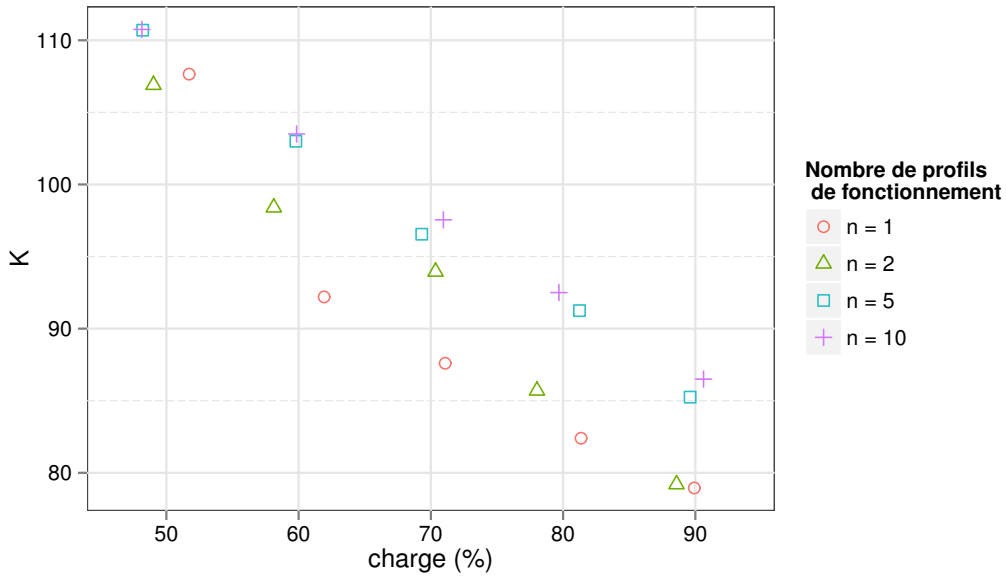


Figure 4.17 – Évolution de l'efficacité de l'heuristique H–LRF en fonction du nombre de profils de fonctionnement – $m = 25$ machines, $n = \{1, 2, 5, 10\}$ profils de fonctionnement

L'heuristique H–LRF reste moins performante que les autres heuristiques pour les charges faibles, mais son efficacité se rapproche de celle de H–HTF à partir de $\chi = 50\%$ (voir figure 4.16). Pour les charges élevées, les débits fournis par les profils sous-nominaux ne sont potentiellement pas suffisants pour atteindre la demande. H–LRF tend alors à sélectionner certaines machines avec leur profil nominal, d'autant plus que la charge augmente. La différence de stratégie entre H–LRF et H–HTF réside alors principalement dans la sélection des machines à utiliser pour chaque période de l'ordonnancement : H–HTF favorise les machines fournissant les débits nominaux les plus forts, alors que H–LRF sélectionne celles fournissant les débits les plus faibles afin de minimiser la surproduction. Cette dernière stratégie permet de conserver plus de potentiel pour la fin de l'ordonnancement et donc de compléter plus de périodes. Pour χ supérieur ou égal à 70%, H–LRF surpasse ainsi H–HTF. H–LRF permet aussi d'atteindre des horizons de production plus grands que H–DP pour ces charges élevées. H–DP permet toutefois d'obtenir les meilleurs résultats pour des charges moyennes variant entre 50% et 60%.

Même si l'heuristique H–DP, basée sur la programmation dynamique, ne permet pas d'obtenir les meilleurs résultats pour toutes les charges, ses performances ne sont jamais loin de celles des

heuristiques qui la surpassent. H–DP présente de plus l'avantage d'être efficace et fiable dans tous les cas considérés, quelque soit la charge, le nombre de machines ou le nombre de profils de fonctionnement.

On peut voir sur les graphes précédents que les résultats obtenus avec toutes les heuristiques tendent à se rapprocher les uns des autres lorsque la charge augmente. Quelque soit la stratégie mise en œuvre, le nombre de possibilités pour la sélection des machines et des profils de fonctionnement associés décroît en effet lorsque la demande σ se rapproche du débit maximal atteignable par la plate-forme.

4.4.3.2 Amélioration obtenue avec la réparation

La figure 4.18 montre l'amélioration apportée par le module de réparation, quelque soit l'efficacité initiale de l'heuristique sans réparation. Les résultats de chaque heuristique avec réparation (H–*-R) sont normalisés avec les résultats de l'heuristique correspondante (H–*).

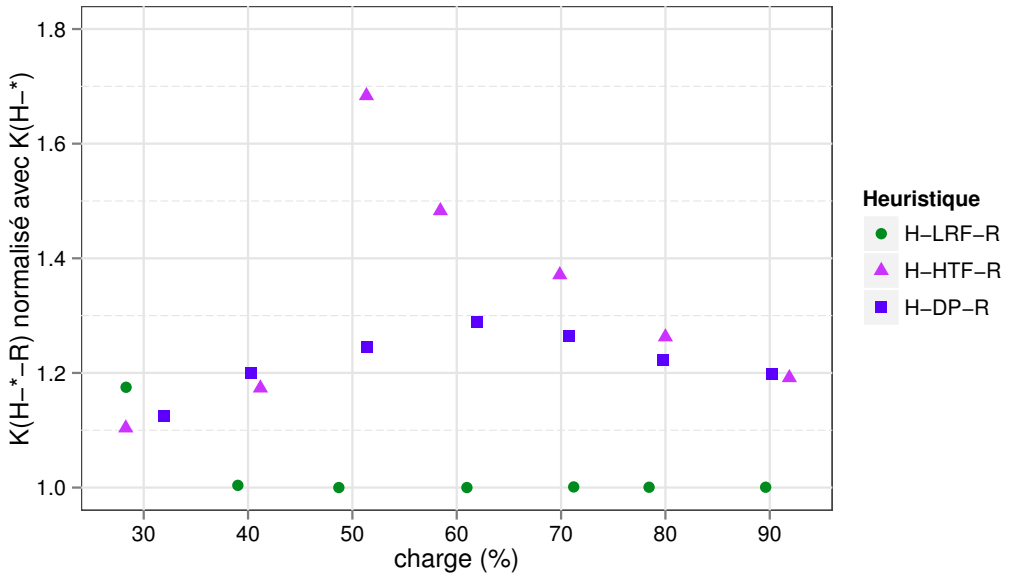


Figure 4.18 – Amélioration obtenue avec la réparation - $m = 25$ machines, $n = 5$ profils de fonctionnement

Une première observation révèle que presque aucune réparation ne peut être effectuée sur les ordonnancements obtenus avec l'heuristique H–LRF. Cette heuristique favorise en effet les débits les plus faibles. Un grand nombre de machines doit alors être utilisé en parallèle pour atteindre la demande à chaque période. Même s'il reste des machines avec un potentiel non nul à la fin des ordonnancements initiaux, très peu d'échanges sont possibles puisque les machines restantes ont beaucoup de chance d'avoir déjà été utilisées dans la plupart des périodes complétées. Ce phénomène peut aussi expliquer la décroissance de l'efficacité de la réparation pour toutes les heuristiques pour de fortes charges.

L'efficacité de la réparation est maximale pour H–HTF, pour des charges χ supérieures ou égales à 50%. On peut remarquer sur la figure 4.16 que, sans réparation, cette heuristique donne les résultats les moins bons pour ces charges. Moins de périodes étant complétées dans l'ordonnancement initial, le potentiel total restant à la fin de l'ordonnancement est plus grand que pour

les autres heuristiques. Un plus grand nombre d'échanges peut donc être fait durant le processus de réparation des ordonnancements obtenus avec H–HTF. Même si l'efficacité relative de la réparation est moindre sur les ordonnancements obtenus avec l'heuristique H–DP, l'heuristique H–DP-R donne d'aussi bons résultats que H–HTF-R. Cela est détaillé dans la suite.

4.4.4 Comparaison à l'optimal

Des solutions optimales peuvent être trouvées en temps raisonnable uniquement pour des instances de problème de petite taille, avec $n \leqslant 2$ profils de fonctionnement, $m \leqslant 5$ machines et $K \leqslant 20$ périodes de temps. Les résultats obtenus avec les heuristiques peuvent donc être comparés aux résultats optimaux uniquement pour ces cas là. Sur la figure 4.19 (respectivement la figure 4.20), la distance à la borne supérieure $\mathrm{BS}(\sigma, \rho_{i,j}, RUL_{i,j})$ (respectivement à l'optimal) du nombre de périodes complétées K est représentée en fonction de la charge χ. Les instances de problèmes considérés ont été limités à 5 machines et 2 profils de fonctionnement pour chacune d'entre elles. Les valeurs de RUL de chaque machine ont de plus été limités pour que l'horizon de production atteignable ne dépasse pas $\mathcal{H} = 20\Delta T$.

On peut voir sur la figure 4.19 que les horizons de production des solutions optimales sont en moyenne à 17% de la borne supérieure $\mathrm{BS}(\sigma, \rho_{i,j}, RUL_{i,j})$, non atteignable. Les trois meilleures heuristiques permettent d'obtenir des résultats proches de l'optimal pour de fortes charges ($\chi \geqslant 70\%$). Pour les cas considérés (peu de machines et peu de profils de fonctionnement), les heuristiques avec réparation sont aussi très efficace, et ce pour toutes les charges. Les horizons atteints avec H–DP-R et H–HTF-R sont en effet en moyenne respectivement à 16% et 8% de la valeur optimale (voir figure 4.20).

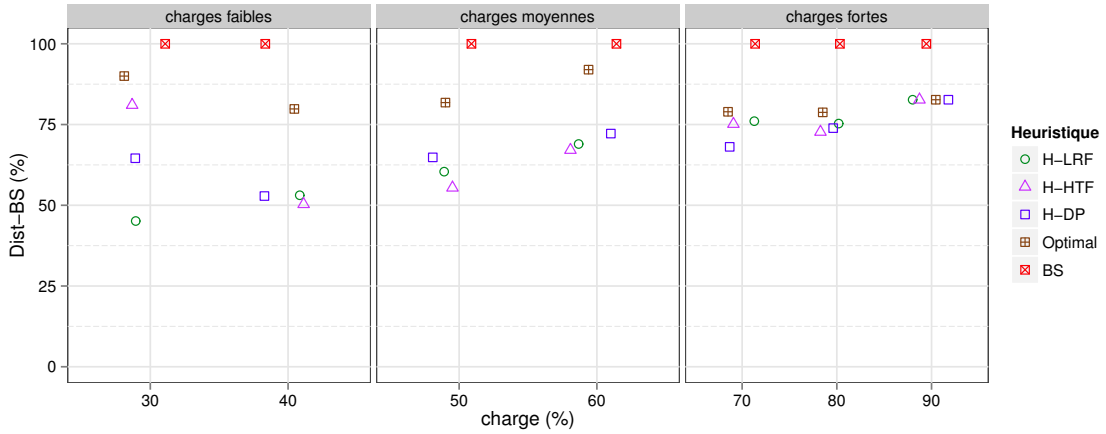


Figure 4.19 – Comparaison à la borne supérieure – $m = 5$ machines, $n = 2$ profils de fonctionnement

4.4.5 Comparaison à la borne supérieure

Dans les figures suivantes, la distance à la borne supérieure $\mathrm{BS}(\sigma, \rho_{i,j}, RUL_{i,j})$ du nombre de périodes complétées K est représentée en fonction de la charge χ pour des problèmes de plus grande taille. L'horizon de la solution optimale étant inférieure à la borne supérieure pour chaque cas considéré, les résultats sont en réalité meilleurs que les pourcentages représentés. Seuls les résultats des trois heuristiques les plus efficaces, H–LRF, H–HTF et H–DP, sont représentés.

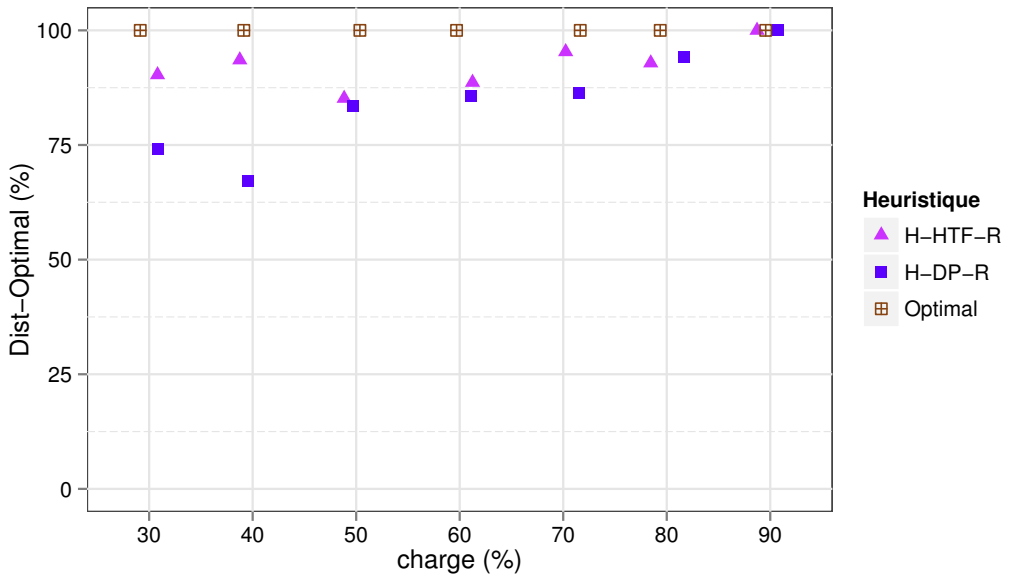


Figure 4.20 – Comparaison à l'optimal pour H–DP-R et H–HTF-R – $m = 5$ machines, $n = 2$ profils de fonctionnement

4.4.5.1 Heuristiques sans réparation

D'après les résultats proposés sur la figure 4.21, le nombre de périodes complétées avec toutes les heuristiques atteint au moins 50% de la borne supérieure pour toutes les charges. La distance maximale à cette borne supérieure est réduite à 30% pour l'heuristique H–DP. Dans les meilleurs cas, les résultats obtenus avec H–DP atteignent 90% du nombre de périodes théorique maximal. En considérant un grand nombre de profils de fonctionnement, les résultats obtenus avec l'heuristique H–LRF atteignent aussi cette performance pour des charges élevées.

Le nombre de machines nécessaires pour atteindre la demande étant d'autant plus élevé que la charge est grande, le nombre de combinaisons pour les choix des couples machine/profil diminue lorsque la charge augmente. Cela explique pourquoi les résultats obtenus avec toutes les heuristiques se rapprochent à la fois les uns des autres et de la borne supérieure pour les charges élevées.

4.4.5.2 Amélioration obtenue avec la réparation

Comme discuté précédemment, l'étape de réparation ne permet pas d'améliorer les horizons des ordonnancements obtenus avec l'heuristique H–LRF pour des charges χ supérieures à 30% lorsque les machines considérées peuvent être utilisées avec plus d'un profil de fonctionnement ($n > 1$). La réparation permet toutefois d'améliorer les résultats obtenus avec les heuristiques H–HTF et H–DP de manière significative. Le nombre de périodes complétées avec H–HTF-R (respectivement avec H–DP-R) atteint en moyenne 94% (respectivement 93%) de la borne supérieure BS(σ, $\rho_{i,j}$, $RUL_{i,j}$) pour toutes les charges (voir figure 4.22). Lorsque des échanges de machines sont possibles, la réparation améliore ainsi les résultats à des valeurs très proches de l'optimal et ce, quelle que soit l'efficacité de l'heuristique initiale (sans réparation).

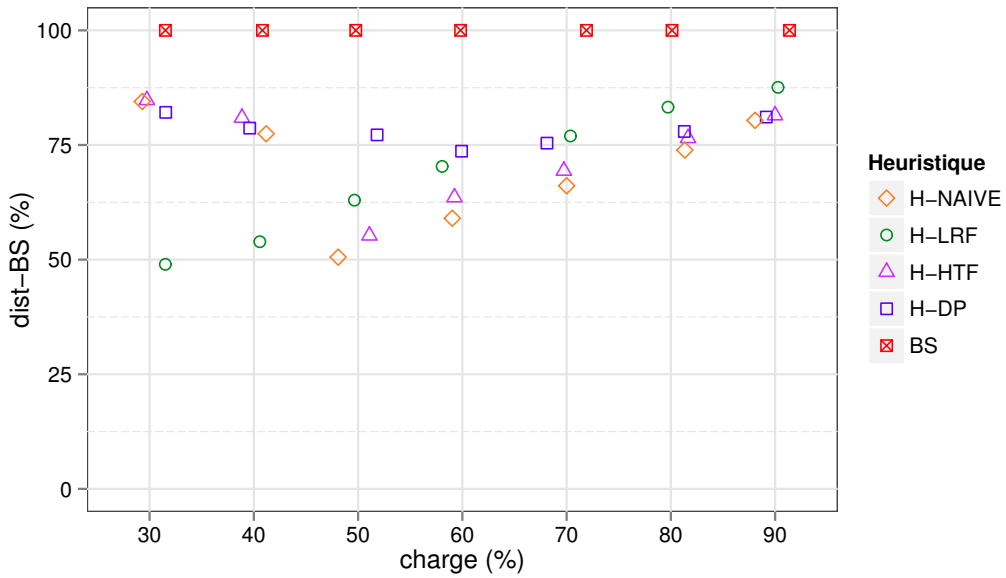


Figure 4.21 – Comparaison des résultats à la borne supérieure – $m = 5$ machines, $n = 2$ profils de fonctionnement

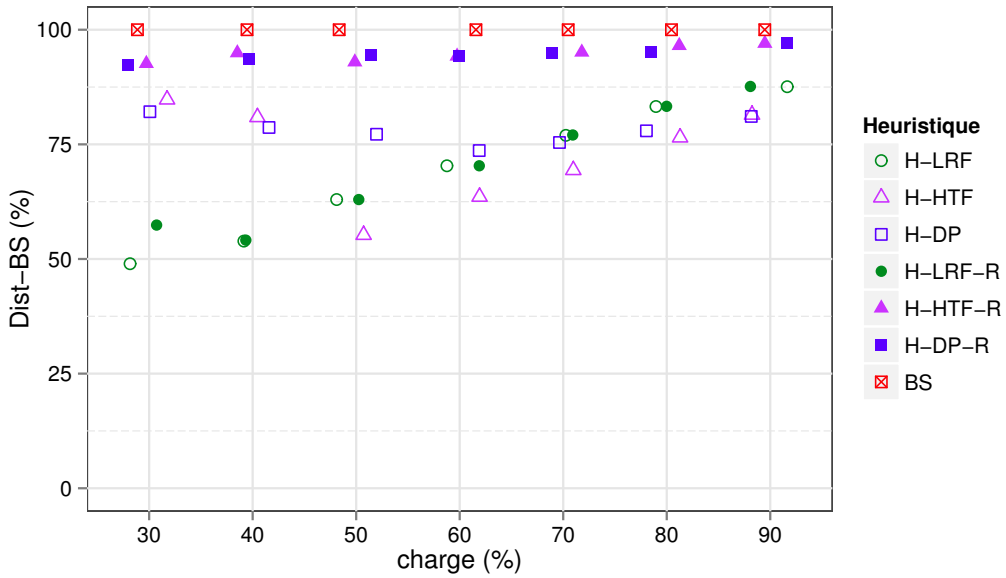


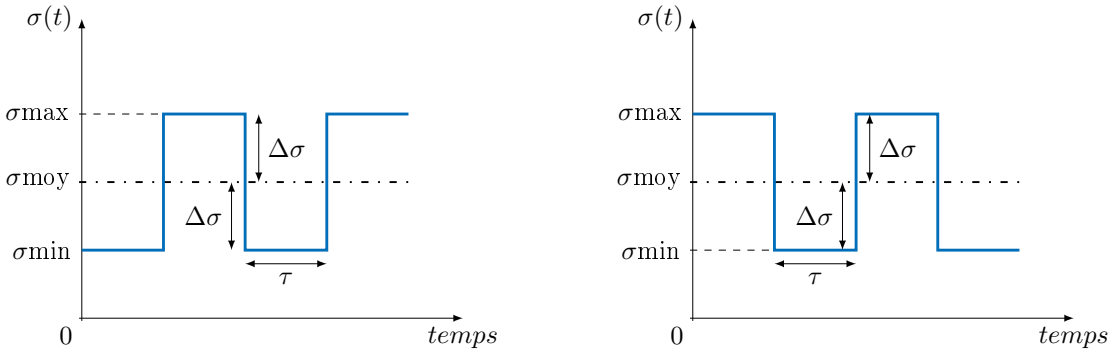
Figure 4.22 – Amélioration obtenue avec la réparation - $m = 25$ machines, $n = 5$ profils de fonctionnement

4.4.6 Robustesse de l'approche pour une demande variable

Les heuristiques proposées dans ce chapitre pour construire des ordonnancements ont été définies pour une demande σ constante durant tout l'horizon de production. Les différentes stratégies mises en œuvre peuvent toutefois être facilement adaptées pour une demande σ_k

constante par morceaux. Les heuristiques construisant les solutions période par période peuvent être utilisées sans modification. Celles fonctionnant par groupes de périodes nécessitent une étape d'adaptation très simple. Il s'agit d'appliquer chaque solution successive (correspondant à une sélection d'un groupe de machines à utiliser et des profils de fonctionnement associés) sur le minimum des deux valeurs suivantes : (i) le RUL minimum des machines sélectionnées ; (ii) le temps pendant lequel la demande σ_k reste constante.

La robustesse des heuristiques sans réparation donnant les meilleurs résultats en termes d'horizon de production atteint est validée sur la base de simulations prenant en compte une demande σ_k constante par morceaux. Cette dernière est définie par une demande moyenne σmoy, à laquelle une variation de $\pm\Delta\sigma$ est appliquée suivant deux scénarios décrits sur la figure 4.23. Le premier scénario débute avec une demande faible alors que le second débute avec une demande forte. Chaque palier est appliqué sur un temps τ. Pour les résultats présentés dans la suite, tous les paliers sont de la même longueur $\tau = \mathrm{BS}/\xi$, avec BS l'horizon de production théorique maximal pour le cas considéré et ξ le nombre de paliers. De la même façon que pour une demande constante, plusieurs valeurs moyennes σmoy sont testées, avec des valeurs fixées suivant la relation définie précédemment par l'équation (4.17).

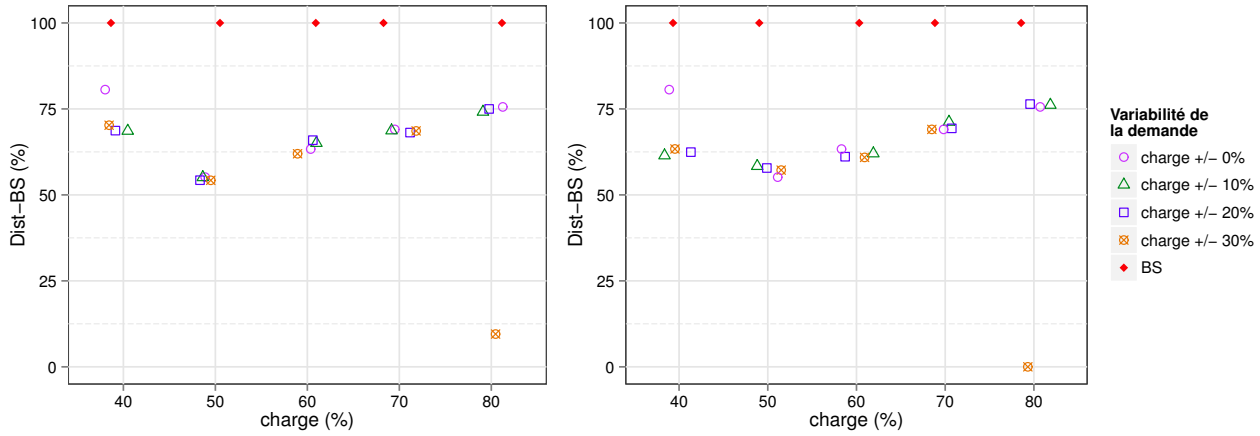


(a) Profil de demande variable type 1 **(b)** Profil de demande variable type 2

Figure 4.23 – Évolutions types d'une demande variable, constante par morceaux σ_k

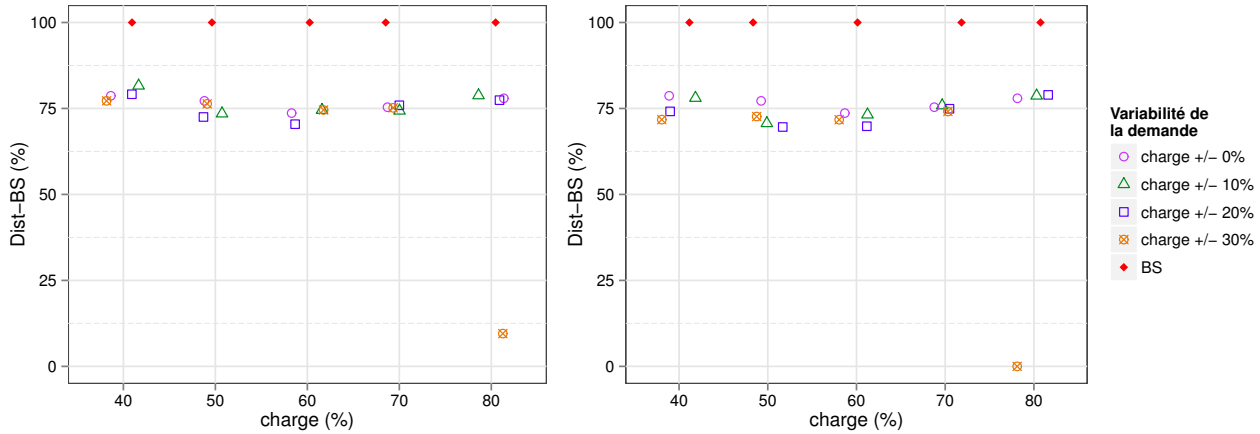
Pour les résultats présentés sur les figures suivantes, les profils de demande ont été divisés en 10 périodes de temps. L'efficacité de chaque stratégie est testée sur trois amplitudes $\Delta\sigma$ de variation de la demande. La première correspond à une variation de $\pm 10\%$ de la charge χ, telle que $\sigma\mathrm{min} = 0.9 \cdot \sigma\mathrm{moy}$ et $\sigma\mathrm{max} = 1.1 \cdot \sigma\mathrm{moy}$, avec $\sigma\mathrm{moy} = \chi \cdot \rho\mathrm{max}_{tot}$. La seconde amplitude testée considère une variation de $\pm 20\%$ et la troisième de $\pm 30\%$. La demande minimale σmin valant zéro avec $\chi - 30\%$ pour une charge $\chi = 30\%$ et la demande maximale σmax étant dans la plupart des cas inatteignable pour une charge $\chi = 90\%$, seuls les résultats pour les charges variant de 40% à 80% sont représentés sur les figures 4.24 et 4.25, représentant respectivement les résultats obtenus avec les heuristiques H–HTF et H–DP.

Les figures 4.24 et 4.25 permettent de comparer les résultats obtenus avec chacun des profils types de demande et pour différentes variations de la demande. Une légère différence peut être notée entre les résultats obtenus avec les deux heuristiques pour chaque profil de demande. Les stratégies sont légèrement moins efficaces lorsque la demande commence par une valeur élevée (profil type 2). Les deux heuristiques restent toutefois performantes pour remplir l'objectif de maximisation de l'horizon de production lorsque la demande varie par morceaux.



(a) Profil de demande variable type 1 **(b)** Profil de demande variable type 2

Figure 4.24 – Robustesse de l'heuristique H–HTF pour une demande σ_k constante par morceaux - $m = 25$ machines, $n = 5$ profils de fonctionnement



(a) Profil de demande variable type 1 **(b)** Profil de demande variable type 2

Figure 4.25 – Robustesse de l'heuristique H–DP pour une demande σ_k constante par morceaux - $m = 25$ machines, $n = 5$ profils de fonctionnement

4.4.7 Synthèse des résultats

Les heuristiques proposées pour la résolution du problème d'optimisation traité dans ce chapitre sont de deux types. Les premières, détaillées dans la partie 4.3.1, sont des heuristiques dites constructives. Elles débutent avec une solution vide, qu'elles améliorent de façon itérative jusqu'à ce qu'une solution complète soit trouvée. Les résultats obtenus avec ces heuristiques (H–RAND, H–NAIVE, H–LRF, H–HTF, H–DP) sont dans la plupart des cas relativement proches de la borne supérieure, mais peuvent encore être améliorés. Une amélioration a donc été proposée via l'étape de réparation, qui correspond à une heuristique d'amélioration. Cette dernière débute avec une solution initiale complète, qu'elle tente d'améliorer de façon itérative et de manière

déterministe [85]. L'étape de réparation permet d'obtenir de solutions plus proches de l'optimal, avec toutefois un temps de calcul potentiellement plus grand que pour les heuristiques sans réparation.

Les résultats développés dans les parties précédentes montrent qu'il n'est pas nécessaire de définir des heuristiques complexes pour obtenir des ordonnancement efficaces pour des charges faibles. En comparaison avec les autres heuristiques, les heuristiques les plus simples, H–RAND et H–NAIVE, permettent en effet d'obtenir de bons résultats pour des charges χ inférieures ou égales à 40%. Des heuristiques plus sophistiquées sont toutefois nécessaires pour des charges plus fortes, pour lesquelles une sélection trop basique des machines à utiliser n'est pas assez fiable compte tenu de l'objectif. Pour ces heuristiques appliquant des stratégies d'ordonnancement plus complexes, il a été montré que H–HTF est plus performante pour des charges faibles, alors que H–LRF donne de meilleurs résultats pour des charges élevées, lorsque plusieurs profils de fonctionnement sont considérés. H–DP est globalement efficace pour toutes les configurations.

L'heuristique H–DP a été définie de telle sorte que la surproduction soit minimisée à chaque période de l'ordonnancement. Cet objectif est respecté, puisque quasiment aucune surproduction n'est effectuée lorsque cette heuristique est utilisée, notamment pour les charges faibles. Cet objectif ne semble toutefois pas être le meilleur à suivre, puisque les ordonnancements générés avec l'heuristique H–NAIVE atteignent dans certains cas des horizons de production plus grand alors qu'ils génèrent beaucoup plus de surproduction.

Même si H–HTF est moins efficace que H–DP dans la plupart des cas, l'"étape de réparation permet de mettre les deux heuristiques à égalité en termes d'horizon de production atteint. Avec la réparation, H–HTF-R est en effet aussi efficace que H–DP-R. L'heuristique H–HTF-R présente de plus l'avantage d'être moins gourmande en temps de calcul. Les temps de calcul augmentent avec la charge χ pour toutes les heuristiques, mais de grandes différences peuvent être constatées suivant la stratégie appliquée. Avec $m = 25$ machines et $n = 5$ profils de fonctionnement, les heuristiques gloutonnes (H–LRF et H–HTF) définissent des ordonnancements en moins de $20\,ms$ [1], alors que la programmation dynamique (H–DP) a besoin de $4\,min$ en moyenne. Avec l'étape de réparation, les temps de calcul de H–LRF-R et H–HTF-R passent respectivement à $40\,ms$ et $80\,ms$. Les temps de calcul requis par H–DP-R atteignent $7\,min$ en moyenne.

Lors du processus de réparation, les machines ajoutées à la place des machines récupérées dans l'ordonnancement initial sont utilisées avec leur profil de fonctionnement le plus sous-nominal possible. Cela permet de maximiser leur durée d'utilisation résiduelle avant maintenance et donc le nombre de fois pour lequel elles peuvent être échangées et/ou utilisées pour des périodes additionnelles. Dans les ordonnancements obtenus avec toutes les heuristiques avec réparation, même avec H–HTF-R, les machines sont donc la plupart du temps utilisées avec un profil de fonctionnement sous-nominal. Ces ordonnancements ayant des horizons de production plus longs que ceux obtenus en ne considérant que les profils nominaux, cela montre l'intérêt d'utiliser les machines avec différents profils de fonctionnement.

1. Les simulations ont été effectuées en Java sur le Mésocentre de calcul de Franche-Comté (Paramètres : Processeur Intel® Xeon® CPU X5550 @ 2.67GHz×4, 24 à 96 Go, 64 bits)

4.5 Synthèse

Plusieurs méthodes de résolution ont été proposées dans ce chapitre pour le problème de maximisation de la durée de vie d'un ensemble de machines lorsque chacune d'entre elles peut fournir un nombre discret de niveaux de performances. Il a été démontré que le problème d'optimisation est NP-complet au sens fort dans le cas général. Une formulation optimale utilisant un programme linéaire en nombres entiers permet toutefois de trouver des solutions optimales en temps raisonnable pour des problèmes de petite taille, mettant en jeu un petit nombre de machines avec peu de profils de fonctionnement et sur des horizons de production limités. Pour des problèmes de taille plus importante et donc plus réalistes, plusieurs heuristiques ont été proposées pour définir des ordonnancements définissant quelles machines utiliser à chaque instant et avec quel profil de fonctionnement. Plusieurs de ces heuristiques sont très performantes puisqu'elles permettent de définir des ordonnancements dont les horizons de production se rapprochent de la valeur optimale. Les horizons de production des solutions fournies par ces heuristiques sont en effet proches de la borne supérieure que nous avons définie.

Les temps de résolution associés aux heuristiques sont de plus cohérents avec l'ordonnancement hors-ligne considéré, et ce, même pour les plus performantes d'entre elles et pour un grand nombre de machines. L'heuristique H–HTF-R étant à la fois efficace du point de vue de l'objectif de maximisation de l'horizon de production et rapide pour fournir une solution (quelques millisecondes), son utilisation pourrait aussi être envisagée pour un ordonnancement en ligne.

Afin d'optimiser encore l'utilisation du potentiel des machines, il s'agit de répondre à la demande tout en maximisant le potentiel restant à chaque instant. Pour cela, une plus grande finesse dans l'adaptation des débits fournis par chaque machine pourrait être utile pour ne fournir que le strict minimum permettant d'atteindre la demande. L'idée sous-jacente est alors d'augmenter le nombre de profils de fonctionnement avec lequel les machines peuvent être utilisées. En poussant ce concept au maximum, on obtient un nombre infini de niveaux de performance, entre le débit minimum correspondant au profil de fonctionnement le plus sous-nominal, $N_{n-1,j}$, et le débit maximum qui correspond au profil nominal $N_{0,j}$. Ce cas de figure correspond au modèle de profils de fonctionnement avec variation continue des performances proposé dans le chapitre précédent. Plusieurs méthodes de résolution du problème d'optimisation considéré dans ce chapitre sont proposées dans les chapitres suivants pour des machines fonctionnant suivant ce modèle continu.

Chapitre 5

Décision avec profils continus - Approche discrète

Sommaire

Ce chapitre présente une première approche de résolution du problème d'ordonnancement considérant des machines dont les performances évoluent de façon continue entre une borne inférieure et une borne supérieure, suivant le modèle introduit en partie 3.3.

Une première idée serait de reprendre la formulation optimale basée sur un programme linéaire proposée dans le chapitre précédent pour des machines présentant un nombre discret de profils de fonctionnement. Il s'agirait de relaxer la contrainte discrète sur les débits en supprimant la dimension i associée aux profils de fonctionnement pour obtenir une variable binaire de décision $x_{j,k}$. Le nombre de variables du programme linéaire serait divisé par n, avec n le nombre de profils de fonctionnement précédemment considéré. Une variable réelle $\rho_{j,k}$ représentant les valeurs de débit pour chaque machine M_j à chaque période k devrait cependant être ajoutée, transformant le programme linéaire en nombres entiers en un programme linéaire mixte. Ce dernier faisant intervenir un nombre de variables plus faible, il permettrait la résolution de problèmes de plus grande taille, sans lever tout à fait la limitation liée au nombre de variables. Le passage en continu sur la dimension des débits n'est toutefois pas trivial. Chaque valeur de RUL étant exprimée de façon continue en fonction du débit, les RUL des machines ne sont plus des données du problème, mais deviennent des inconnues. L'expression des contraintes liées au respect du RUL devient alors fortement non linéaire, même si la relation liant les débits aux RUL est linéaire. Un programme mixte basé sur la formulation proposée dans le chapitre précédent et intégrant à la fois une variation continue des débits et la discrétisation du temps n'est alors pas adapté pour le problème considéré ici.

La méthode de résolution proposée dans ce chapitre utilise tout de même la programmation linéaire, mais limite le nombre de variables de chaque programme utilisé à un nombre raisonnable en découpant l'horizon de production en plusieurs morceaux. Chaque programme linéaire ne prend pas en compte la dimension temporelle autrement que par la limite d'utilisation fixée par le RUL associé à chaque débit. L'ordonnancement est ainsi effectué par phases successives et l'évolution de l'état de santé des machines en fonction de leur utilisation est traitée en dehors de la programmation linéaire par un processus heuristique.

5.1 Résolution

La méthode de résolution proposée fonctionne par groupes de périodes. La résolution du problème d'optimisation est en effet faite par le biais de la résolution de sous-problèmes pour lesquels le débit fourni par chaque machine reste constant tout au long de son utilisation. Les ordonnancements obtenus sont alors constitués d'une succession de phases au sein desquelles les machines utilisées gardent la même configuration.

Une méthode optimale est utilisée pour définir la configuration des machines au sein de chaque phase d'ordonnancement. Cette méthode, basée sur une programmation linéaire, est tout d'abord présentée pour la résolution de chaque sous-problème. Le principe général de l'algorithme de résolution définissant une utilisation des machines sur tout l'horizon de production atteignable est détaillé dans un second temps.

5.1.1 Approche optimale par morceaux

Soit un ensemble de m machines caractérisées par des valeurs de débit ($\rho\min_j$, $\rho\max_j$ et ρopt_j avec $1 \leqslant j \leqslant m$) et des valeurs de RUL ($RUL\text{opt}_j$). Soit de plus une demande constante σ à satisfaire. Le problème est de déterminer un sous-ensemble de machines à utiliser et un débit ρ_j constant pour chaque machine sélectionnée de façon à atteindre la demande σ le plus longtemps possible.

Un programme mathématique exprimant l'objectif de maximisation de la durée d'utilisation des machines sous différentes contraintes liées aux caractéristiques des machines considérées et à la demande à satisfaire est tout d'abord détaillé. Une formulation linéaire de ce programme mathématique est ensuite proposée pour permettre la résolution du problème par une programmation linéaire.

5.1.1.1 Programme mathématique

Une formulation mathématique du problème d'optimisation est détaillée dans le système d'équations (5.1). Le programme mathématique proposé contient des variables de décision binaires x_j telles que $x_j \in \{0,1\}$ pour $1 \leqslant j \leqslant m$. On pose $x_j = 1$ si la machine M_j est utilisée dans la solution et $x_j = 0$ dans le cas contraire. Des variables réelles ρ_j ($1 \leqslant j \leqslant m$) définissent la contribution de chaque machine au débit total de la solution. Pour chaque machine M_j, la valeur prise par le débit fourni ρ_j est contrainte entre un débit minimum, $\rho\min_j$, et une valeur maximale, $\rho\max_j$. Ces bornes correspondent aux valeurs extrêmes de l'intervalle de définition des débits pour le modèle considéré, défini dans la partie 3.3.1.2 du chapitre 3.

$$
\begin{cases}
\max \quad \min\limits_{j|x_j=1} RUL_j(\rho_j) & \text{(5.1a)} \\
\quad \sum\limits_{j=1}^{m} x_j \cdot \rho_j \geqslant \sigma & \text{(5.1b)} \\
\text{t.q.} \quad 0 \leqslant RUL_j(\rho_j) \leqslant \dfrac{x_j \cdot \rho_j}{a_j} - \dfrac{b_j}{a_j} \cdot x_j \qquad \forall 1 \leqslant j \leqslant m & \text{(5.1c)} \\
\quad 0 \leqslant RUL_j(\rho_j) \leqslant RUL\text{opt}_j \cdot x_j \qquad \forall 1 \leqslant j \leqslant m & \text{(5.1d)} \\
\quad 0 \leqslant RUL_j(\rho_j) \leqslant \dfrac{x_j \cdot \rho_j}{\alpha_j} - \dfrac{\beta_j}{\alpha_j} \cdot x_j \qquad \forall 1 \leqslant j \leqslant m & \text{(5.1e)} \\
\text{avec} \quad \rho\min_j \leqslant \rho_j \leqslant \rho\max_j \qquad \forall 1 \leqslant j \leqslant m & \text{(5.1f)} \\
\quad x_j \in \{0,1\} \qquad \forall 1 \leqslant j \leqslant m & \text{(5.1g)}
\end{cases}
$$

Les contraintes du programme mathématique permettent de respecter à la fois les caractéristiques des machines et les besoins liés au problème d'optimisation considéré. La disponibilité limitée de chaque débit est tout d'abord exprimée dans les équations (5.1c), (5.1d) et (5.1e). Chacun de ces ensembles de contraintes correspond à une partie de la limitation de la durée de vie des machines définie par le modèle de comportement à l'usure pris en compte. L'équation (5.1c) exprime la décroissance du débit maximal $\rho\max_j$ représentée par une ligne verte en traits discontinus sur la figure 3.4 et s'applique pour les débits $\rho\text{opt}_j \leqslant \rho_j \leqslant \rho\max_j$. L'équation (5.1d) s'applique aux débits $0.9 \cdot \rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$ et l'équation (5.1e) aux débits $\rho\min_j \leqslant \rho_j \leqslant 0.9 \cdot \rho\text{opt}_j$.

Ces deux dernières équations correspondent à la fin de vie des machines, représentée par des lignes rouges sur la figure 3.4. La forme de l'évolution des valeurs de RUL en fonction des débits étant concave et les trois équations utilisées définissant des valeurs supérieures pour les valeurs de RUL, la prise en compte simultanée de ces équations permet de relier le débit choisi par le programme ρ_j à la bonne valeur de RUL, $RUL_j(\rho_j)$.

L'atteinte de la demande σ est ensuite assurée par la contrainte exprimée par l'équation (5.1b). Très simplement, la somme des débits fournis par les machines utilisées (pour lesquelles $x_j = 1$) doit au minimum atteindre la valeur σ.

La fonction objectif prise en compte, définie par l'équation (5.1a), correspond à la maximisation sous les contraintes définies précédemment de l'horizon de production $\widetilde{K}\Delta T$ atteint par la configuration définie par le programme linéaire. Le nombre d'unités de temps $\widetilde{K}$ complétées est limité par le RUL minimal des machines sélectionnées avec le niveau de performance ρ_j sélectionné : $\widetilde{K} = \min_{1 \leqslant j \leqslant m | x_j = 1} RUL_j(\rho_j)$. Cela correspond à un problème Max-Min. Une contrainte particulière est alors ajoutée au programme mathématique pour limiter le nombre de périodes $\widetilde{K}$ optimisé en fonction de la solution (voir équation (5.2b)). Un terme de la forme $C(1 - x_j)$ est de plus ajouté aux équations (5.1c), (5.1d) et (5.1e) pour surestimer arbitrairement les RUL des machines M_j ($1 \leqslant j \leqslant m$) pour lesquelles $x_j = 0$, avec $C \in \mathbb{R}$, tel que $C > \max_{1 \leqslant j \leqslant m} (RULopt_j)$, de façon à ce que le nombre de périodes complétées $\widetilde{K}$ de la solution ne soit pas limité par le RUL d'une machine non utilisée.

Une minimisation de la surproduction est finalement ajoutée dans la fonction objectif du programme mathématique (voir équation (5.2a)). Aucun stockage n'étant considéré, la restriction de la surproduction permet de minimiser les pertes, qui peuvent être dues à une production non utilisée et donc perdue, ou à une consommation excessive de matières premières. Par construction, la minimisation de la surproduction entraîne la minimisation du nombre de machines utilisées simultanément pour atteindre la demande σ. Une des conséquences est la maximisation du potentiel global restant, qui peut permettre l'augmentation de l'horizon de production global. Cet objectif secondaire est pondéré avec un facteur multiplicateur λ, permettant de lui accorder plus ou moins d'importance dans la détermination de la solution.

Le programme mathématique modifié est détaillé dans le système d'équations (5.2).

$$
\begin{cases}
\max \quad \widetilde{K} - \lambda \left(\sum_{j=1}^{m} x_j \cdot \rho_j - \sigma \right) & (5.2a) \\
\widetilde{K} \leqslant RUL_j(\rho_j) \qquad \forall 1 \leqslant j \leqslant m & (5.2b) \\
\sum_{j=1}^{m} x_j \cdot \rho_j \geqslant \sigma & (5.2c) \\
\text{t.q.} \quad 0 \leqslant RUL_j(\rho_j) \leqslant \frac{x_j \cdot \rho_j}{a_j} - \frac{b_j}{a_j} \cdot x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.2d) \\
0 \leqslant RUL_j(\rho_j) \leqslant RULopt_j \cdot x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.2e) \\
0 \leqslant RUL_j(\rho_j) \leqslant \frac{x_j \cdot \rho_j}{\alpha_j} - \frac{\beta_j}{\alpha_j} \cdot x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.2f) \\
\text{avec} \quad \rho\min_j \leqslant \rho_j \leqslant \rho\max_j \qquad \forall 1 \leqslant j \leqslant m & (5.2g) \\
x_j \in \{0, 1\} \qquad \forall 1 \leqslant j \leqslant m & (5.2h)
\end{cases}
$$

5.1.1.2 Programme linéaire

Une optimisation linéaire est proposée pour résoudre le problème de maximisation. Chaque terme $x_j \cdot \rho_j$ utilisé dans la formulation mathématique du problème étant le produit d'une variable binaire et d'une variable réelle, le programme mathématique défini précédemment n'est pas linéaire. Une linéarisation est alors nécessaire. Cette dernière est faite suivant les propriétés définies dans l'ensemble d'équations (5.3) (voir [14]).

$$\forall x \in \{0,1\},\ \forall y \in [0, U(y)] \text{ et } \forall e \in \mathbb{R},\ e = xy \text{ si et seulement si :}$$

$$\begin{cases} e \leqslant xU(y) & (5.3a) \\ e \leqslant y & (5.3b) \\ e \geqslant y - (1-x)U(y) & (5.3c) \\ e \geqslant 0 & (5.3d) \end{cases}$$

La formulation linéarisée du programme mathématique est détaillée dans l'ensemble d'équations (5.4), avec $e_j = x_j \cdot \rho_j$ pour $1 \leqslant j \leqslant m$. Ce programme linéaire permet de définir un ordonnancement optimal des machines à débits constants, c'est-à-dire de déterminer le sous-ensemble des machines à utiliser et des valeurs de débit constantes pour chacune d'entre elles, qui maximisent l'horizon $\widetilde{K}$ de la solution.

$$\begin{cases} \max & \widetilde{K} - \lambda\left(\sum_{j=1}^{m} e_j - \sigma\right) & (5.4a) \\ & \widetilde{K} \leqslant RUL_j \qquad \forall 1 \leqslant j \leqslant m & (5.4b) \\ & \sum_{j=1}^{m} e_j \geqslant \sigma & (5.4c) \\ & e_j \leqslant x_j \cdot \rho\max_j \qquad \forall 1 \leqslant j \leqslant m & (5.4d) \\ \text{t.q.} & e_j \leqslant \rho_j \qquad 1 \leqslant j \leqslant m & (5.4e) \\ & e_j \geqslant \rho_j - (1-x_j)\rho\max_j \qquad \forall 1 \leqslant j \leqslant m & (5.4f) \\ & e_j \geqslant 0 \qquad \forall 1 \leqslant j \leqslant m & (5.4g) \\ & 0 \leqslant RUL_j \leqslant \frac{e_j}{a_j} - \frac{b_j}{a_j} \cdot x_j + C(1-x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.4h) \\ & 0 \leqslant RUL_j \leqslant RUL\text{opt}_j \cdot x_j + C(1-x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.4i) \\ & 0 \leqslant RUL_j \leqslant \frac{e_j}{\alpha_j} - \frac{\beta_j}{\alpha_j} \cdot x_j + C(1-x_j) \qquad \forall 1 \leqslant j \leqslant m & (5.4j) \\ \text{avec} & \rho\min_j \leqslant \rho_j \leqslant \rho\max_j \qquad \forall 1 \leqslant j \leqslant m & (5.4k) \\ & x_j \in \{0,1\} \qquad \forall 1 \leqslant j \leqslant m & (5.4l) \end{cases}$$

5.1.2 Algorithme global de résolution

En supposant que toutes les machines ne sont pas nécessaires pour atteindre le niveau de service requis, ou que la machine limitante peut encore être utilisée avec un débit plus faible à la fin d'une phase d'ordonnancement, plusieurs sous-problèmes successifs peuvent être résolus

les uns après les autres jusqu'à ce que le potentiel global restant de la plate-forme passe sous la valeur de la demande ($\rho\max_{tot} = \sum_{1 \leqslant j \leqslant m} \rho\max_j < \sigma$). Un principe de résolution globale est alors nécessaire pour construire un ordonnancement maximisant la durée d'utilisation de la plate-forme de machines considérée.

La succession des différentes phases d'ordonnancement utilisant le programme linéaire décrit précédemment est définie par l'algorithme 11. Cet algorithme gère le calcul de l'horizon global de production, ainsi que la mise à jour des caractéristiques des machines (états de santé et valeurs de débits disponibles en fonction du temps) entre chaque lancement du programme linéaire.

Une continuité d'utilisation des machines est respectée entre chaque phase successive d'ordonnancement. Dès qu'une machine est démarrée, elle est en effet utilisée en continu tant que sa fin d'utilisation avant maintenance n'est pas atteinte. Les valeurs des variables binaires x_j sont ainsi fixées pour chaque programme linéaire suivant les règles suivantes : $x_j = 0$ si la machine M_j a atteint sa fin de vie ; $x_j = 1$ si la machine M_j a été utilisée dans la solution précédente et peut toujours être utilisée. Si la machine M_j est toujours disponible mais n'a pas été utilisée au cours de la période de temps précédente, la détermination de la valeur de x_j est faite librement par le programme linéaire. Le respect d'une continuité d'utilisation des machines est cohérent dans un contexte de production de biens, où l'arrêt/démarrage des machines peut avoir un coût, qu'il soit temporel ou lié à un besoin accru de matières premières pour relancer la machine. Une étude de l'impact du respect de la continuité d'utilisation sur l'horizon de production atteint est proposée dans la suite.

La durée d'utilisation avant maintenance globale $\mathcal{H}$ de l'ensemble des machines considérées correspond à la somme des horizons $\widetilde{K}$ atteints par chaque solution des programmes linéaires successifs. Chaque solution fournie par chaque programme linéaire est optimale considérant l'état de santé des machines au début de chaque recherche de solution et la contrainte d'utilisation à débit constant. Les ordonnancements obtenus avec une succession de résolutions optimales ne sont toutefois pas nécessairement optimaux.

5.2 Résultats de simulation

Des simulations ont été conduites pour évaluer l'approche de résolution proposée dans ce chapitre. Après une description de la génération des problèmes utilisés, les paramètres de la méthode sont fixés sur la base de résultats préliminaires. L'approche est ensuite validée pour une demande constante dans le temps. Il est enfin montré que l'approche proposée reste valable pour une demande variable, constante par morceaux.

5.2.1 Génération des problèmes

Différentes instances du problème d'optimisation considéré ont été générées à l'aide d'un simulateur et configurées avec différents paramètres, tels que le nombre m de machines, les caractéristiques intrinsèques des machines (débit maximum, débit minimum, etc.), ainsi que les caractéristiques de la demande à satisfaire. Chaque instance correspond ainsi à une plate-forme donnée de machines.

Algorithme 11: Algorithme de résolution globale

Données :
sol_k : liste des couples (i, j) sélectionnés pour la période k
$\rho\text{max}[j]$: tableau contenant les valeurs des débits maximaux atteignables ρmax_j à la période k
$RUL\text{opt}[j]$: tableau contenant les valeurs des $RUL\text{opt}_j$ à la période k
$x[j]$: tableau contenant les valeurs des x_j forçant ou non l'utilisation des machines
$\widetilde{K}$: nombre de périodes complétées par un programme linéaire
Entrées :
σ : demande à atteindre à chaque période k
$\rho\text{max0}[j]$: tableau contenant les valeurs initiales des débits maximaux atteignables $\rho\text{max}_j(1)$
$\rho\text{min}[j]$: tableau contenant les valeurs initiales des débits minimaux ρmin_j
$\rho\text{opt}[j]$: tableau contenant les valeurs initiales des débits optimaux ρopt_j
$RUL\text{opt0}[j]$: tableau contenant les valeurs des $RUL\text{opt}_j(0)$
$a[j]$: tableau contenant les pentes de décroissance de chaque débit maximum ρmax_j
Sorties :
$sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$: liste des solutions pour chaque période k complétée

`// Remarque : soit` $PL(\rho\text{min}, \rho\text{max}, RUL\text{opt}, a, \sigma, x)$ `le programme linéaire utilisé`

$\rho\text{max}[j] \leftarrow \rho\text{max0}[j]\ \forall\ 1 \leqslant j \leqslant m$
$RUL\text{opt}[j] \leftarrow RUL\text{opt0}[j]\ \forall\ 1 \leqslant j \leqslant m$
$x[j] \leftarrow$ indéterminé $\forall\ 1 \leqslant j \leqslant m$
$k \leftarrow 1$
$(sol_k, \widetilde{K}) \leftarrow$ résolution du $PL(\rho\text{min}, \rho\text{max}, RUL\text{opt}, a, \sigma, x)$
répéter
  pour $k' = k$ *à* $k + \widetilde{K}$ **faire**
    $sol \leftarrow$ ajouter sol_k en fin de sol
  $\rho\text{max}[j] \leftarrow$ mise à jour des valeurs de $\rho\text{max}_j\ \forall\ 1 \leqslant j \leqslant m$
  $RUL\text{opt}[j] \leftarrow$ mise à jour des valeurs de $RUL\text{opt}_j\ \forall\ 1 \leqslant j \leqslant m$

$$x[j] \leftarrow \begin{cases} 0 & \text{si } M_j \text{ a atteint sa fin de vie } (RUL\text{opt}_j < \Delta T); \\ 1 & \text{si } M_j \text{ est utilisée dans la solution } sol_k \text{ et n'a pas atteint sa fin de vie ;} \\ & \text{est laissé indéterminé sinon.} \end{cases}$$

  $k \leftarrow k + \widetilde{K}$
  $(sol_k, \widetilde{K}) \leftarrow$ résolution du $PL(\rho\text{min}, \rho\text{max}, RUL\text{opt}, a, \sigma, x)$
jusqu'à $PL(\rho min, \rho max, RULopt, a, \sigma)$ *n'admet plus de solution avec un horizon* $\widetilde{K}.\Delta T \geqslant \Delta T$
return $sol = (sol_1, \ldots, sol_k, \ldots, sol_K)$

5.2.1.1 Paramétrage des plates-formes de machines

L'application cible de la problématique développée dans ce chapitre est l'optimisation de l'utilisation d'un ensemble de piles à combustible. Les valeurs des paramètres de chaque machine considérée pour les simulations ont donc été basées sur les caractéristiques de piles à combustible types. Ces caractéristiques ont été définies sur la base de spécifications de fabricants. Les débits considérés sont alors exprimés en puissance, pour lesquelles les valeurs prises en compte sont les suivantes pour chaque machine M_j : $\rho\text{max}_j(0) = 500\,W\ \pm 5\%$; $\rho\text{min}_j = 0.15 \cdot \rho\text{max}_j(0)$; $\rho\text{opt}_j = 0.6 \cdot \rho\text{max}_j(0)$.

Une plate-forme peut comporter des machines avec les mêmes caractéristiques de débit. Elles peuvent toutefois être différenciées par leur état de santé au début de l'ordonnancement, exprimé par leur $RUL\text{opt}_j$. Les valeurs des paramètres a_j, exprimant la vitesse de décroissance de ρmax_j

en fonction du temps, ont ainsi été fixées en considérant une durée de vie maximale $RULopt_j = 1500$ heures $\pm 20\%$ pour chaque machine M_j. On a alors $a_j \in [-0.08, -0.13]\, W.h^{-1}\ \forall\ 1 \leqslant j \leqslant m$.

Les résultats proposés dans la suite ont été obtenus avec $m = 25$ machines. Des simulations ont toutefois été lancées pour différentes valeurs de m. Si aucune précision n'est apportée, les conclusions faites pour $m = 25$ machines restent valables pour des plates-formes constituées d'un nombre différent de machines.

5.2.1.2 Paramétrage de la demande

De même que pour les machines présentant un nombre discret de profils de fonctionnement, la demande σ est fixée en fonction du potentiel nominal de la plate-forme et d'une charge χ variant entre 30% et 90% (voir partie 4.4.1.2). Dans ce chapitre, le potentiel nominal total est défini de la façon suivante :

$$\rho\text{nom}_{tot} = \sum_{j=1}^{m} \rho\text{nom}_j \tag{5.5}$$

$$\text{avec} \quad \rho\text{nom}_j = 0.75 \cdot \rho\text{max}_j(0) \quad \forall\ 1 \leqslant j \leqslant m$$

De même que précédemment, les résultats sont représentés en fonction de la charge $\chi = \sigma / \rho\text{max}_{tot}$. Chaque point représenté correspond à la moyenne des résultats obtenus après la simulation de 20 instances différentes de la configuration du problème considérée. Afin de faciliter la lecture des résultats, les différents points représentés pour une même charge χ ont été dispersés autour de l'abscisse correspondante.

5.2.2 Résultats préliminaires

Pour les résultats développés dans la suite, les horizons des phases d'ordonnancement successives, déterminés avec la programmation linéaire, ont été tronqués par valeur inférieure à un nombre entier de périodes. Cela permet de faciliter l'application des solutions, en limitant le changement de configuration des machines à des instants correspondant à des unités entières (par exemple, modification des réglages machines à la minute, à l'heure ou à la journée suivant l'application).

5.2.2.1 Paramétrage de la fonction objectif du programme linéaire

Une première étude du comportement des solutions obtenues avec l'approche de résolution proposée dans ce chapitre permet de déterminer une valeur adéquate pour le paramètre λ utilisé dans chaque programme linéaire. Ce paramètre définit une pondération pour la seconde partie de la fonction objectif, qui tend à minimiser la surproduction. Les résultats présentés sur les deux figures suivantes ont été obtenus avec une succession de plusieurs programmes linéaires et correspondent à un seul test sur une plate-forme de machines. On peut tout d'abord observer sur la figure 5.1 que la prise en compte de la seconde partie de la fonction objectif permet effectivement de minimiser la surproduction. Dans les cas où $\lambda = 0$, ce qui revient à ne maximiser que l'horizon de production, une surproduction non négligeable est faite. Sur les cas testés ($\lambda = \{0, 0.5, 1, 1.5, 2, 5, 10\}$), lorsque la minimisation de la surproduction est prise en compte,

le surplus de production tombe à 0 dès $\lambda \neq 0$. La figure 5.2 montre ensuite que l'influence du paramètre λ est légèrement plus grande pour les charges faibles ($\chi = 30$ à 50%), quelque soit le nombre de machines considérées. Quelque soit la charge, on peut aussi observer que les cas avec une surproduction non nulle (pour $\lambda = 0$) sont associés à des horizons de production plus faibles.

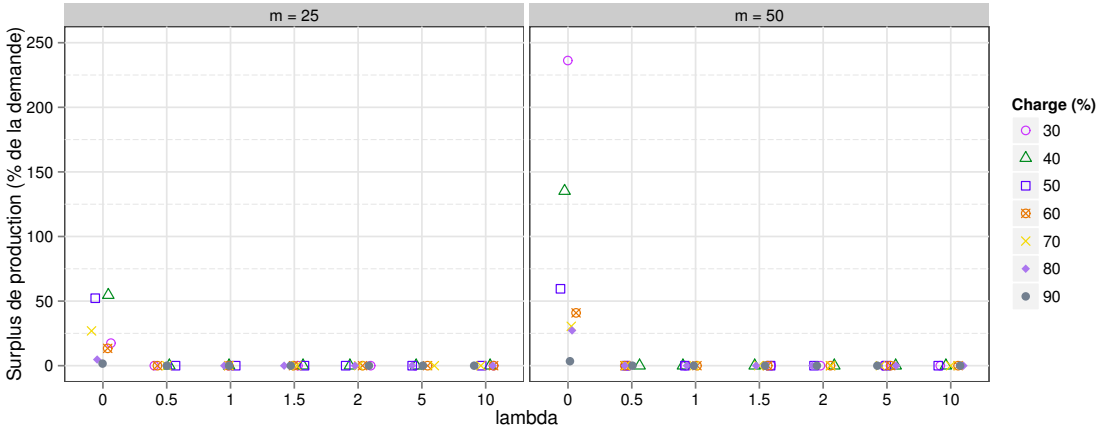


Figure 5.1 – Surplus de production effectué pour chaque charge, en fonction de la valeur du paramètre λ – $m = 25$ et 50 machines

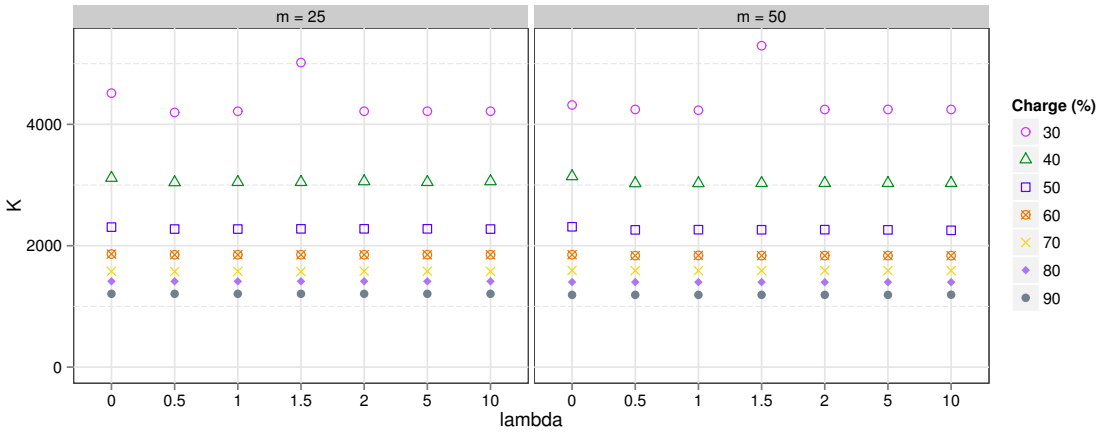


Figure 5.2 – Nombre de périodes complétées (K) sur un test, pour chaque charge, en fonction de la valeur du paramètre λ – $m = 25$ et 50 machines

La minimisation de la surproduction permet donc d'allonger les horizons de production obtenus avec la stratégie de résolution globale. D'après les résultats présentés sur les figures précédentes, qui donnent une tendance représentative de tous les cas testés, le poids donné à la minimisation de la surproduction a été fixé arbitrairement à $\lambda = 1.5$ pour les développements suivants.

5.2.2.2 Continuité d'utilisation des machines

Sur la figure 5.3, les résultats obtenus sans utilisation continue ont été normalisés avec ceux obtenus avec utilisation continue forcée des machines. On peut voir que forcer une utilisation

continue des machines une fois qu'elles ont été démarrées ne pénalise pas significativement l'horizon de production atteint. Quelque soit le nombre de machines considérées, ne pas contraindre la continuité d'utilisation des machines ne permet d'améliorer les résultats que de 8% en moyenne (24% au maximum avec les paramètres considérés). Pour les résultats présentés dans la suite, une continuité d'utilisation des machines démarrées est respectée.

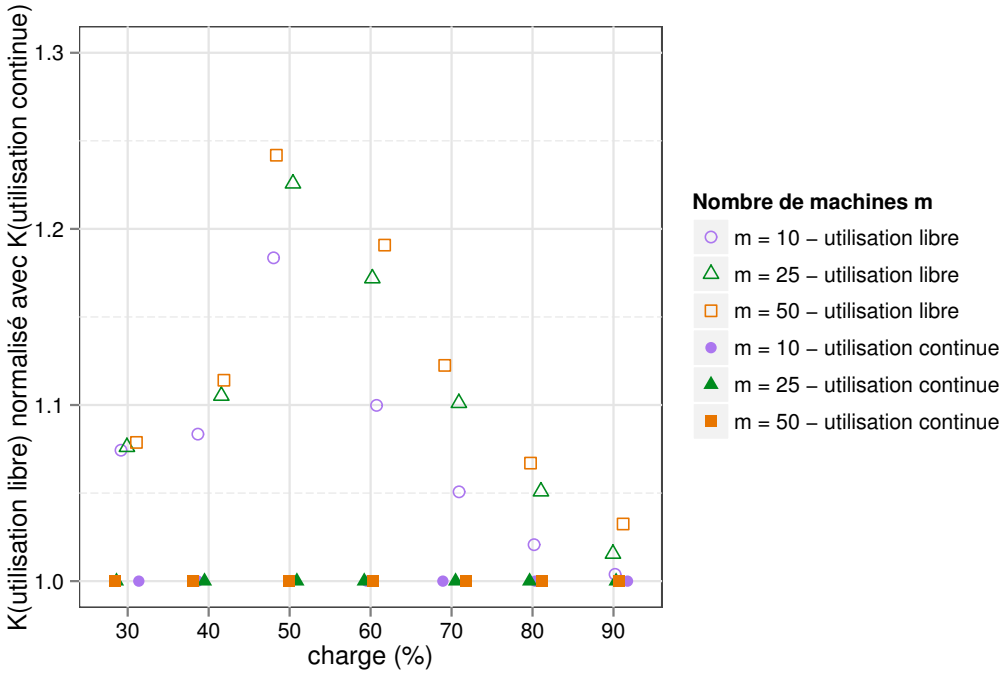


Figure 5.3 – Comparaison des résultats obtenus avec et sans continuité d'utilisation des machines – $m = 10$, 25 et 50 machines

5.2.3 Validation de l'approche pour une demande constante

L'efficacité de l'approche proposée dans ce chapitre est validée pour une demande σ constante tout au long de l'horizon de production. L'originalité de la stratégie appliquée réside dans la possibilité de choisir le débit fourni par chaque machine dans un ensemble continu de valeurs disponibles. Les résultats qui suivent tendent à montrer que cette stratégie, notée *S1*, permet d'obtenir de meilleurs résultats que d'autres stratégies plus basiques. La première stratégie de comparaison considérée, *S2*, suit une utilisation classique des machines, à des conditions opératoires constantes au cours du temps. Chaque machine M_j est ainsi supposée fournir un débit constant fixé à la valeur nominale ρnom_j tout au long de sa durée de vie. Pour les résultats présentés dans la suite, la valeur de ρnom_j est basée sur le cas des piles à combustible et fixée à la valeur préconisée par les constructeurs, soit $\rho\text{nom}_j = 0.75 \cdot \rho\text{max}_j(0)$. La troisième stratégie *S3* limite quant à elle la sélection des débits dans l'intervalle $[0.9 \cdot \rho\text{opt}_j, \rho\text{opt}_j]$, contenant les débits associés à la durée de vie maximale de chaque machine.

Les horizons de production K atteints avec chacune des stratégies définies précédemment sont comparés sur la figure 5.4 et représentés en fonction de la charge χ. L'évolution d'une borne supérieure pour K, notée BS, est aussi représentée. Cette borne supérieure, définie par l'équation (5.6), correspond au temps maximal pendant lequel la demande constante peut être

atteinte avec l'ensemble des machines considérées. De façon simplifiée, elle peut être définie géométriquement par la somme pour chaque machine de la surface entre les courbes $\rho_j(t) = \rho\min_j$ et $\rho_j(t) = \rho\max_j(t)$, divisée par la valeur de la demande σ. Cette expression ne prenant pas en compte la pénalisation des durées de vie des débits faibles $\rho\min_j \leqslant \rho_j \leqslant 0.9 \cdot \rho\mathrm{opt}_j$, la borne supérieure BS n'est jamais atteignable.

$$\mathrm{BS} = \left\lfloor \frac{\sum_{j=1}^{m} \frac{1}{2}(1.6\rho\max_j(0) - 2\rho\min_j) \cdot RUL\mathrm{opt}_j}{\sigma \cdot \Delta T} \right\rfloor \tag{5.6}$$

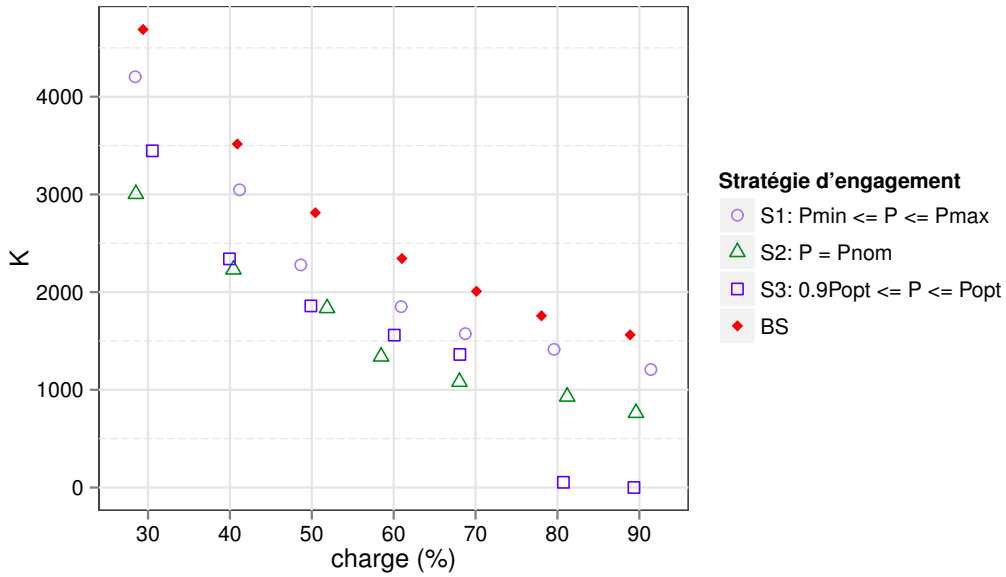


Figure 5.4 – Comparaison des horizons de production atteints avec les stratégies *S1*, *S2* et *S3* – $m = 25$ machines

On peut voir sur la figure 5.4 que les horizons de production atteints avec la stratégie *S2*, fixant la contribution de chaque machine à son débit nominal $\rho\mathrm{nom}_j$, décroissent avec la charge χ. Aucun choix de débit n'est effectué avec cette stratégie et la durée de vie de chaque machine est limité par la durée de vie associée à $\rho\mathrm{nom}_j$. Seul le nombre de machines utilisées en parallèle pour atteindre la demande σ varie en fonction de la charge. L'horizon de production atteint avec la stratégie *S2* est alors déterminé par la somme des durées de vie des machines limitantes dans chacun des groupes de machines formés. Lorsque la charge augmente, le nombre de machines parallèles nécessaires pour atteindre la demande augmente, ce qui implique une diminution du nombre de groupe de machines pouvant être formés et donc la diminution de l'horizon K. La stratégie *S3*, limitant le choix des débits dans l'intervalle $[0.9 \cdot \rho\mathrm{opt}_j, \rho\mathrm{opt}_j]$ pour chaque machine, est légèrement plus efficace, puisque les débits choisis sont associés à des durées de vie plus longues. Cette troisième stratégie n'est toutefois pas fiable pour les charges fortes ($\chi \geqslant 80\%$). Les valeurs des demandes testées étant définies en fonction du débit nominal total $\rho\mathrm{nom}_{tot}$ et sachant que $\rho\mathrm{opt}_j = 0.8 \cdot \rho\mathrm{nom}_j$ pour chaque machine M_j, une charge $\chi = 90\%$ n'est pas atteignable lorsque les débits que peuvent fournir chaque machine sont limités à $\rho\mathrm{opt}_j$. Une

charge $\chi = 80\%$ n'est généralement pas atteignable non plus à cause d'arrondis effectués au cours de la détermination de la valeur ρnom_{tot}. Les horizons de production les plus grands sont atteints avec la stratégie *S1*. Les résultats obtenus avec cette stratégie étant proches de la borne supérieure BS, ils sont aussi proches de l'optimal.

Cela peut être vu sur la figure 5.5, qui montre la distance à la borne supérieure BS des horizons de production atteints avec chaque stratégie. La stratégie *S2* permet d'atteindre un horizon moyen égal à environ 58% de la borne supérieure. Avec la stratégie *S3*, ce taux monte à 67% en moyenne pour les charges variant de 30% à 70%, mais est proche de 0% pour les charges fortes. Comme montré précédemment, les meilleurs résultats sont obtenus avec *S1*, qui permet d'atteindre des horizons de production entre 77% et 90% des horizons maximum atteignables et 80% en moyenne. Cette stratégie est donc au pire entre 77% et 90% de l'optimal pour les expériences réalisées.

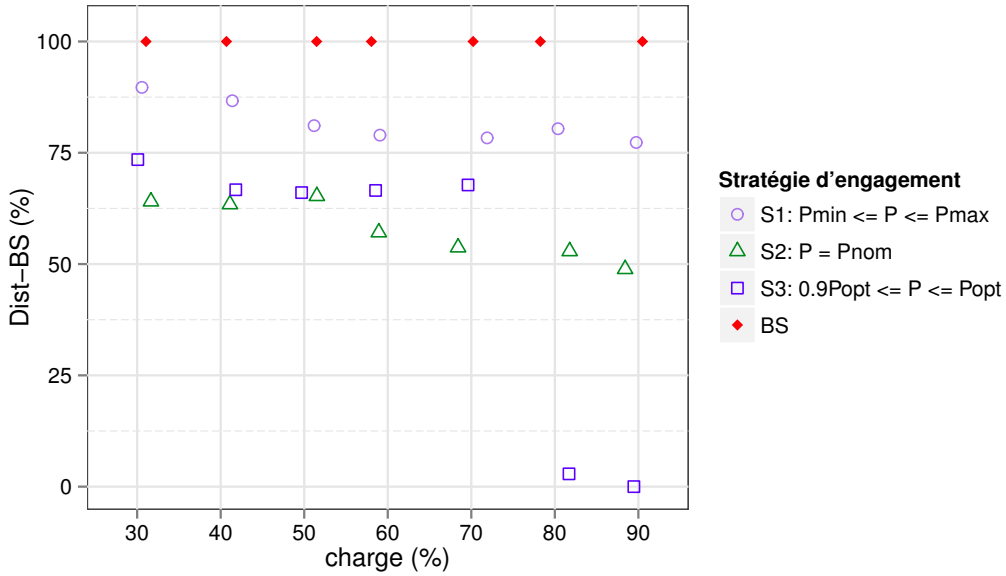


Figure 5.5 – Distance à la borne supérieure BS des horizons de production atteints avec les stratégies *S1*, *S2* et *S3* – $m = 25$ machines

Lorsqu'une demande constante est considérée, la stratégie d'engagement des machines proposée dans ce chapitre permet donc d'étendre la durée d'utilisation avant maintenance d'une plate-forme par rapport à des stratégies limitant l'amplitude de l'intervalle de débits disponibles.

5.2.4 Limite du modèle

Une limite du modèle considéré dans ce chapitre peut être mise en évidence en comparant les résultats obtenus avec la stratégie proposée *S1* et ceux obtenus avec une stratégie *S4*, qui limite le choix des débits à chaque instant t dans l'intervalle $[0.9 \cdot \rho\text{opt}_j, \rho\text{max}_j(t)]$ pour chaque machine M_j. On peut en effet voir sur la figure 5.6 que les performances de ces deux stratégies sont similaires. La principale différence réside dans la surproduction effectuée lorsque le débit ρopt_j est sélectionné à la place d'un débit plus faible pour une machine M_j, sans que cela n'ait un impact significatif sur la durée d'utilisation avant maintenance de cette machine.

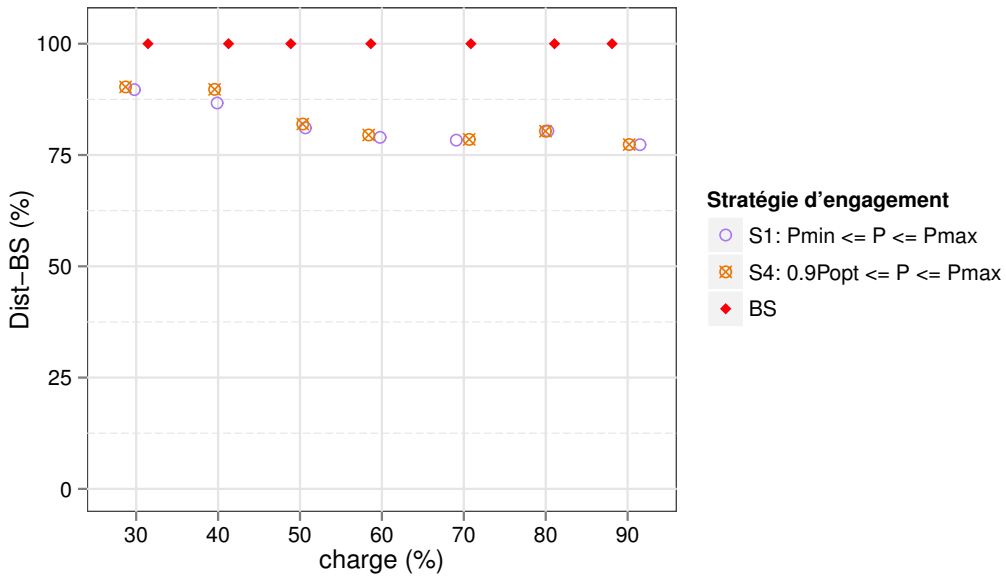


Figure 5.6 – Distance à la borne supérieure BS des horizons de production atteints avec les stratégies *S1* et *S4* – $m = 25$ machines

Ce comportement est lié à la limitation des performances dans le modèle utilisé pour des débits faibles. Il est alors plus intéressant du point de vue de la durée d'utilisation de choisir un débit associé à la durée d'utilisation avant maintenance optimale ($0.9 \cdot \rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{opt}_j$), même si la valeur du débit dépasse les besoins. Cette propriété est illustrée sur la figure 5.7. De manière générale, si un système de stockage permet de stocker une éventuelle surproduction, il paraît donc plus approprié de n'utiliser que les débits de la partie supérieure du modèle ($0.9 \cdot \rho\text{opt}_j \leqslant \rho_j \leqslant \rho\text{max}_j$) pour chaque machine.

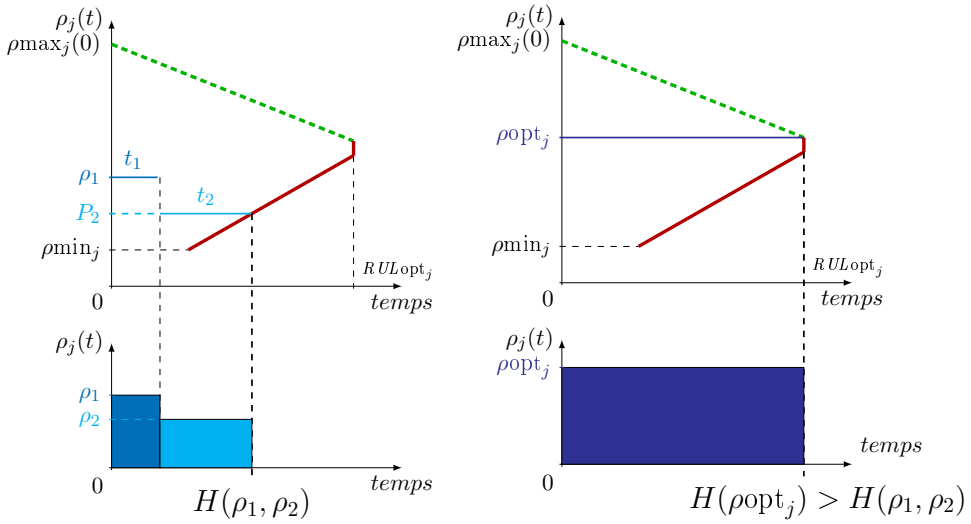


Figure 5.7 – Restriction à la partie supérieure du modèle

5.2.5 Temps de résolution

Les temps de résolution sont du même ordre de grandeur pour toutes les stratégies proposées dans ce chapitre. La figure 5.8 compare les temps de résolution nécessaires pour chacune des stratégies, considérant une demande σ constante sur tout l'horizon de production. Dans chaque cas, le temps de résolution décroît légèrement lorsque la charge χ augmente. Un grand nombre de machines devant être utilisées en parallèle pour atteindre les demandes associées aux grandes charges, le nombre de programmes linéaires nécessaires pour définir un ordonnancement utilisant le maximum du potentiel disponible est en effet plus petit pour les charges élevées que pour les charges faibles. Les stratégies pour lesquelles les intervalles de choix pour les débits sont les plus restreints (*S2* et *S3*) nécessitent de plus moins de temps pour fournir une solution que les stratégies plus générales, *S1* et *S4*.

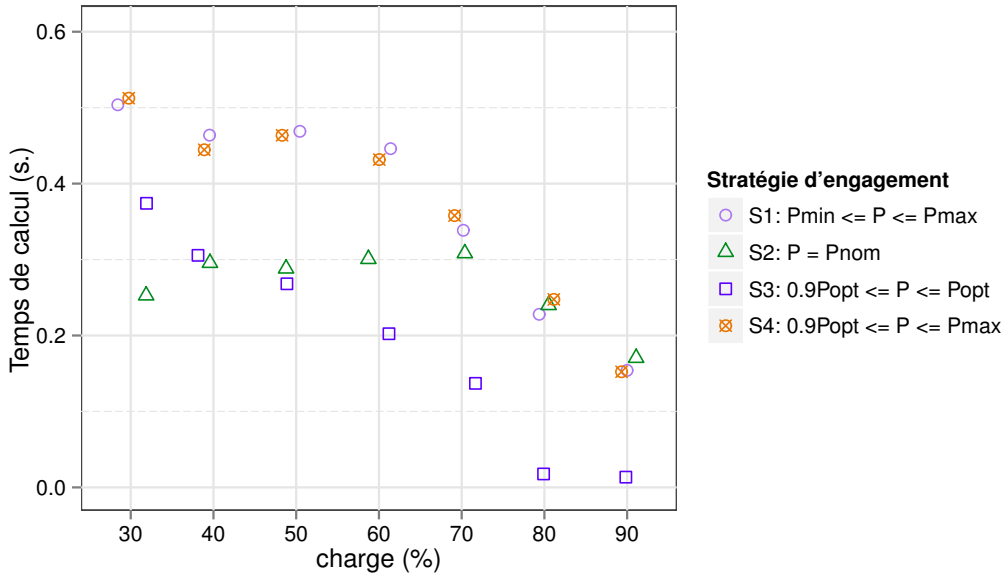


Figure 5.8 – Temps de résolution [5] pour les différentes stratégies *S1*, *S2*, *S3* et *S4* – $m = 25$ machines

Pour la stratégie la plus générale, *S1*, on peut voir sur la figure 5.9 que le temps de résolution augmente avec le nombre m de machines. Le nombre de machines considérées intervient en effet dans la dimension du problème et donc dans le nombre de variables de chaque programme linéaire. Des solutions sont toutefois trouvées en moins d'une minute pour toutes les configurations testées considérant un nombre de machines $m \leqslant 300$ et en moins de cinq minutes pour $m \leqslant 400$. La stratégie d'engagement des machines proposée dans ce chapitre est donc très rapide pour des plates-formes composées d'un nombre de machines raisonnable.

5.2.6 Robustesse de l'approche pour une demande variable

L'approche de résolution développée dans ce chapitre pour une demande constante σ peut être utilisé pour une demande constante par morceaux σ_k moyennant quelques modifications mineures. Les horizons des solutions fournies par chaque programme linéaire utilisé peuvent tout

5. Les simulations ont été effectuées sur Matlab (Paramètres : Processeur Intel® Core™ i5-3550 CPU @ 3.30GHz×4, 15.6 Gio, 64 bits)

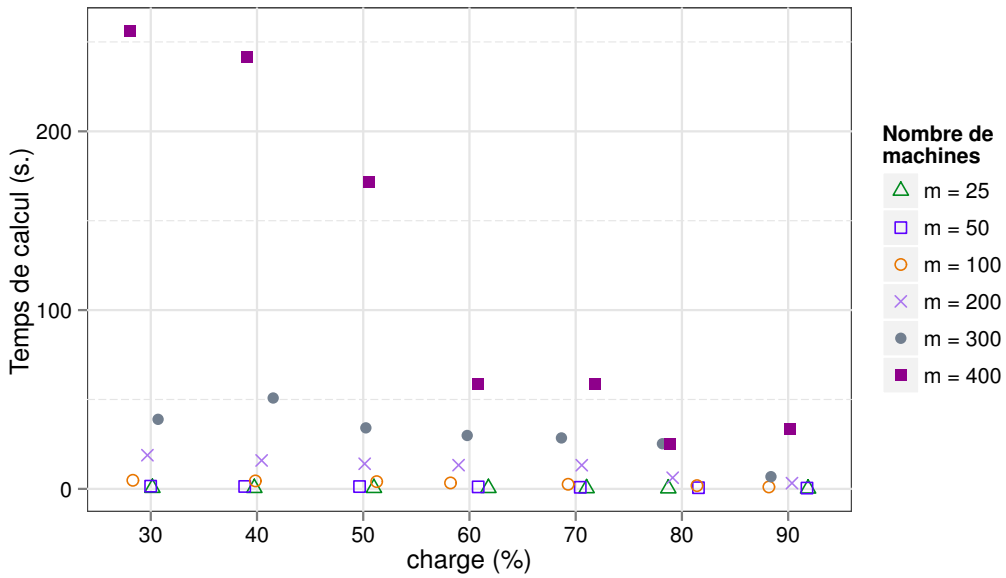


Figure 5.9 – Temps de résolution [5] pour différentes valeurs de m – Stratégie *S1*

simplement être tronqués au temps pendant lequel la demande reste constante. De meilleures solutions peuvent toutefois être trouvées pour un horizon plus court. Afin de trouver l'ordonnancement optimal des machines pour une valeur de demande σ_k restant constante durant le temps associé t_k, une contrainte additionnelle est définie pour limiter l'horizon de la solution déterminée par le programme linéaire (voir l'équation (5.7c)). La fonction objectif reste la même, ainsi que l'ensemble des autres contraintes. Que l'horizon de la solution soit limité par le RUL de la machine limitante ou par le temps durant lequel la demande reste constante, la solution trouvée par le biais du programme linéaire adapté, détaillé dans le système d'équations (5.7), est alors optimale pour une utilisation des machines à débits constants.

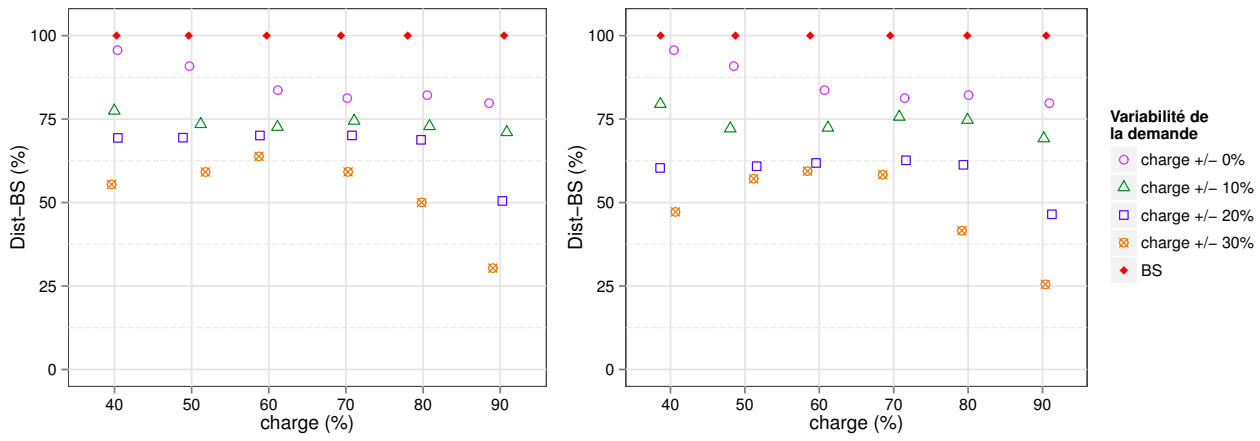
En plus de cette modification dans le programme linéaire, une étape supplémentaire doit être effectuée dans l'algorithme de résolution globale pour mettre à jour la demande σ_k à atteindre en fonction du profil de demande et de l'horizon de production global $\mathcal{H}$ atteint avant le lancement de chaque programme linéaire.

La robustesse de la stratégie *S1* face à un changement de demande au cours de l'horizon de production est validée en considérant une demande $\sigma(t) = \sigma_k$ constante par morceaux paramétrée suivant les profils types définis dans la partie 4.4.6. Les profils de demande ont été divisés en 10 périodes de temps. L'efficacité de la stratégie est testée sur trois amplitudes $\Delta\sigma$ de variation de la demande. La première correspond à une variation de $\pm 10\%$ de la charge χ, tel que $\sigma\text{min} = 0.9 \cdot \sigma\text{moy}$ et $\sigma\text{max} = 1.1 \cdot \sigma\text{moy}$, avec $\sigma\text{moy} = \chi \cdot \rho\text{nom}_{tot}$. La seconde amplitude testée considère une variation de $\pm 20\%$ et la troisième de $\pm 30\%$. La demande minimale σmin valant zéro avec $\chi - 30\%$ pour une charge $\chi = 30\%$, seuls les résultats pour les charges variant de 40% à 90% sont représentés sur les figures suivantes.

On peut tout d'abord remarquer sur la figure 5.10 que l'efficacité de la stratégie d'engagement des machines décroît lorsque le taux de variation de la charge augmente, quelque soit le type de profil de demande variable considéré. Plus la variabilité de la demande est forte, moins la stratégie proposée est donc efficace. Les résultats restent toutefois prometteurs puisque les

horizons de production atteignent au moins 71% (respectivement 64%) de l'horizon théorique maximal $\mathrm{BS} \cdot \Delta T$ pour une charge moyenne $\chi = 50\%$ avec le profil type 1 (respectivement le profil type 2).

$$
\begin{cases}
\max & \widetilde{K} - \lambda\left(\sum_{j=1}^{m} e_j - \sigma_k\right) & & (5.7a)\\
& \widetilde{K} \leqslant RUL_j \qquad \forall 1 \leqslant j \leqslant m & & (5.7b)\\
& \widetilde{K} \leqslant t_k \qquad \forall 1 \leqslant j \leqslant m & & (5.7c)\\
& \sum_{j=1}^{m} e_j \geqslant \sigma_k & & (5.7d)\\
& e_j \leqslant x_j.\rho\mathrm{max}_j \qquad \forall 1 \leqslant j \leqslant m & & (5.7e)\\
\text{t.q.} & e_j \leqslant \rho_j \qquad \forall j = 1, \ldots, m & & (5.7f)\\
& e_j \geqslant \rho_j - (1 - x_j)\rho\mathrm{max}_j \qquad \forall 1 \leqslant j \leqslant m & & (5.7g)\\
& e_j \geqslant 0 \qquad \forall 1 \leqslant j \leqslant m & & (5.7h)\\
& 0 \leqslant RUL_j \leqslant \frac{e_j}{a_j} - \frac{b_j}{a_j}.x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & & (5.7i)\\
& 0 \leqslant RUL_j \leqslant RUL\mathrm{opt}_j.x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & & (5.7j)\\
& 0 \leqslant RUL_j \leqslant \frac{e_j}{\alpha_j} - \frac{\beta_j}{\alpha_j}.x_j + C(1 - x_j) \qquad \forall 1 \leqslant j \leqslant m & & (5.7k)\\
\text{avec} & \rho\mathrm{min}_j \leqslant \rho_j \leqslant \rho\mathrm{max}_j \qquad \forall 1 \leqslant j \leqslant m & & (5.7l)\\
& x_j \in \{0, 1\} \qquad \forall 1 \leqslant j \leqslant m & & (5.7m)
\end{cases}
$$



(a) Profil de demande variable type 1 **(b)** Profil de demande variable type 2

Figure 5.10 – Distance à la borne supérieure BS des horizons de production atteints avec la stratégie *S1* pour les deux profils de demande variable – $m = 25$ machines

Une petite différence peut être notée entre les résultats obtenus avec les deux profils de demande variable type. La stratégie est moins efficace pour le profil type 2, commençant avec une demande forte. La demande associée au premier palier n'étant pas la même pour les deux profils, le nombre de machines utilisées peut être différent. Ce nombre peut être plus élevé pour

le profil type 2. Une continuité d'utilisation des machines engagées étant imposée par la stratégie d'ordonnancement, les choix faits pour le premier palier de demande impactent les suivants et, par extension, l'horizon de production atteint. Un nombre trop important de machines est ainsi potentiellement utilisé lorsque la demande diminue, ce qui entraîne de la surproduction et de la perte de potentiel. Aucun stockage n'étant considéré, le potentiel perdu ne peut être réutilisé dans la suite de l'ordonnancement.

De même que pour une demande constante sur tout l'horizon de production, le temps de calcul des solutions augmente avec la charge et avec le nombre de machines considérées. Les ordonnancements sont toutefois obtenus en moins d'une minute pour toutes les configurations de problème testées.

5.2.7 Synthèse des résultats

Les différents résultats présentés montrent que suivre la stratégie proposée dans ce chapitre permet d'allonger la durée d'utilisation avant maintenance d'une plate-forme de machines pouvant fournir des performances variant de façon continue entre deux bornes. En comparaison avec la stratégie *S2*, qui fixe l'utilisation des machines à leur débit nominal et qui est actuellement la stratégie généralement suivie pour l'utilisation des piles à combustibles, la stratégie *S1* proposée permet en effet d'augmenter l'horizon de production de 38% en moyenne.

D'un point de vue mathématique, le nombre de variables de chaque programme linéaire, fonction du nombre m de machines considérées, constitue une limite pour la méthode de résolution optimale par morceaux proposée. Cette dernière permet tout de même de définir très rapidement des solutions de bonne qualité pour des plates-formes composées d'un nombre de machines inférieur à quelques centaines, ce qui est cohérent avec un contexte de production classique. Que ce soit pour une demande σ constante sur tout l'horizon de production, ou une demande variable σ_k, les temps de résolution restent de plus cohérents avec les échelles de temps associées aux décisions à moyen terme considérées (fréquences de décision en heures, en jours ou en semaines).

5.3 Synthèse

Une première méthode de résolution a été proposée dans ce chapitre pour le problème d'optimisation considérant des machines dont les performances peuvent varier de façon continue entre deux valeurs extrêmes. Cette méthode est constituée de résolutions successives de sous-problèmes. Une approche optimale basée sur une programmation linéaire en nombres réels permet de déterminer un ordonnancement optimal des machines considérant l'objectif de maximisation de l'horizon de production lorsque l'utilisation des machines est limitée à un débit fixe. Les horizons des solutions obtenus pour chaque sous-problème sont limités soit par le RUL minimal des machines sélectionnées, soit par le temps pendant lequel la demande reste constante. En supposant que toutes les machines ne sont pas nécessaires pour atteindre la demande, ou que la machine limitante peut encore être utilisée avec un débit plus faible, plusieurs programmes linéaires peuvent être lancés les uns après les autres jusqu'à ce que le potentiel global de la plate-forme passe au-dessous de la valeurs de la demande. Le changement de profil de fonctionnement au cours de l'utilisation des machines n'étant pas géré par l'approche optimale, une stratégie de

résolution globale a donc été proposée pour utiliser au maximum le potentiel de la plate-forme considérée.

Même si le résultat visé est la définition d'un ordonnancement hors ligne, cette construction des solutions pas à pas est compatible avec un ordonnancement en ligne, qui définirait la configuration future des machines au fur et à mesure de leur utilisation. Le temps de résolution de chaque programme linéaire est cohérent avec les exigences temporelles d'un tel ordonnancement.

L'approche développée dans ce chapitre est inspirée de l'approche optimale proposée dans le chapitre 4 pour le problème considérant des machines avec une variation discrète des performances. Le passage en continu dans la dimension des débits permet de simplifier les programmes linéaires et donc de résoudre des problèmes de plus grande taille, avec un nombre de machines plus important et des horizons de production plus longs. La résolution par morceaux rendue nécessaire par la relation existant entre les débits ρ_j et les valeurs de RUL associées ne permet toutefois pas d'obtenir des solutions optimales globalement sur tout l'horizon de production.

Un type de méthode de résolution totalement différent de ceux utilisés jusqu'ici est proposé dans le chapitre suivant pour résoudre le problème d'optimisation considérant des machines dont les performances varient de façon continue entre deux extrêmes. Ce changement est motivé par la volonté de s'affranchir du côté sous-optimal des solutions associé au raisonnement par morceaux. L'idée est alors de permettre une évolution continue dans le temps des débits fournis par les machines. La résolution développée dans le chapitre 6 utilise ainsi une formulation du problème et une approche de résolution qui permettent de définir un ordonnancement des machines de façon globale, en considérant l'horizon de production comme un tout.

Chapitre 6

Décision avec profils continus - Approche globale

Sommaire

Les différentes approches de résolution proposées dans les chapitres précédents permettent de trouver des solutions optimales en temps polynomial uniquement pour des problèmes pour lesquels le nombre de machines et/ou l'horizon de production restent petits. Nous avons montré que l'utilisation de méthodes de résolution sous-optimales est nécessaire pour traiter des problèmes de grande taille. Bien que les résultats obtenus avec les méthodes proposées soient, dans la plupart des cas, proches de la borne supérieure, une amélioration est encore possible. Le point limitant principal des méthodes sous-optimales proposées jusqu'ici est lié à la discrétisation du temps. Les solutions sont en effet construites de façon itérative sur l'horizon de production et la définition à chaque instant d'une configuration optimale (considérant l'état courant du système) ne suffit pas à garantir l'optimalité de la solution globale.

Une autre approche utilisant un paradigme totalement différent est proposée dans ce chapitre pour traiter des problèmes d'optimisation de grande taille. La recherche d'une solution est faite de façon globale sur tout l'horizon de production. La contribution de chaque machine tout au long de sa durée de vie est dans ce cas considérée dans son ensemble et optimisée globalement sur tout l'horizon. Chaque contribution est déterminée par le biais d'une optimisation convexe, qui présente l'intérêt d'aller de pair avec des phénomènes globaux. Étant donnée la forme des éléments mis en jeu, les caractéristiques locales sont en effet aussi des caractéristiques globales, valables sur tout l'espace de définition. Un extremum local est ainsi par exemple aussi un extremum global. L'utilisation de l'optimisation convexe permet de plus de résoudre en temps polynomial les problèmes de grande taille, mettant en jeux dans notre cas un grand nombre de machines sur des horizons longs.

Les machines considérées dans ce chapitre présentent des profils de fonctionnement avec une variation continue des performances sur un intervalle donné. Le modèle considéré est celui défini dans la partie 3.3.1. Une formulation mathématique du problème d'optimisation est tout d'abord introduite pour permettre sa résolution avec des méthodes de programmation convexe. Différentes méthodes de résolution sont ensuite proposées et comparées par le biais de résultats de simulations.

6.1 Formulation mathématique du problème d'optimisation

Une nouvelle formulation mathématique du problème d'optimisation mettant en jeu des éléments convexes est nécessaire pour une résolution basée sur la programmation convexe. Une définition vectorielle des contributions des machines au cours du temps ainsi que des différentes contraintes inhérentes à leur utilisation est proposée. L'idée générale de résolution, qui se retrouve dans toutes les méthodes proposées dans la suite, est la suivante : il s'agit de minimiser une fonction convexe soumise à un ensemble de contraintes. Le programme mathématique décrivant l'objectif et les contraintes à respecter est détaillé dans le système d'équations (6.1).

On note $f_j(t)$ $(t \in [\![0, T]\!])$ le vecteur définissant l'évolution du débit fourni par la machine M_j au cours du temps, avec T la longueur de l'horizon de décision considéré. Le lien entre l'horizon de décision et l'horizon de la solution est explicité dans la partie 6.2.4. Les contributions de toutes les machines M_j $(1 \leqslant j \leqslant m)$ sont regroupées dans un vecteur $F \in \mathbb{R}^{m(T+1)}$ tel que :

$$F = [f_1(0), \ldots, f_1(T), \ldots, f_j(t), \ldots f_m(0), \ldots, f_m(T)]$$

$$\begin{cases} \min \quad \sigma(t) - \sum_{j=1}^{m} f_j(t) \qquad \forall\ 0 \leqslant t \leqslant T & \text{(6.1a)} \\ \text{t.q.} \quad \mathit{f}\max_j(t) = \mathit{f}\max_j(t-1) + a_j \text{ si } f_j(t) > 0 \qquad \forall\ 1 \leqslant j \leqslant m,\ \forall 1 \leqslant t \leqslant T & \text{(6.1b)} \\ \qquad \mathit{f}\max_j(t) = \mathit{f}\max_j(t-1) \text{ si } f_j(t) = 0 \qquad \forall\ 1 \leqslant j \leqslant m,\ \forall 1 \leqslant t \leqslant T & \text{(6.1c)} \\ \text{avec} \quad f_j(t) = 0 \text{ ou } \mathit{f}\min_j \leqslant f_j(t) \leqslant \mathit{f}\max_j(t) \qquad \forall 1 \leqslant j \leqslant m,\ \forall 0 \leqslant t \leqslant T & \text{(6.1d)} \end{cases}$$

La fonction que l'on cherche à minimiser est liée à l'atteinte de la demande et est exprimée sous la forme définie dans l'équation (6.1a). A chaque instant t, il s'agit de minimiser la différence entre le débit total fourni par l'ensemble des machines et la demande $\sigma(t)$. Cet objectif est cohérent avec l'objectif de maximisation de l'horizon. Éviter la surproduction permet en effet de conserver du potentiel pouvant être utilisé pour atteindre la demande pendant un temps plus long.

Chaque contribution $f_j(t)$ est soit nulle, soit définie entre une valeur minimale $\mathit{f}\min_j$ et une valeur maximale $\mathit{f}\max_j(t)$ (voir équation (6.1d)). $f_j(t) = 0$ signifie que la machine M_j n'est pas utilisée à l'instant t.

Les deux contraintes du programme mathématique expriment la variabilité du débit maximal atteignable $\mathit{f}\max_j$ pour chaque machine M_j. Conformément au modèle utilisé dans ce chapitre, chaque débit maximal évolue au cours de l'utilisation de la machine correspondante en suivant une droite d'équation $\mathit{f}\max_j(t) = a_j t + b_j$, avec a_j la vitesse de décroissance de $\mathit{f}\max_j$ et $b_j = \mathit{f}\max_j(0)$. La prise en compte de la décroissance de $\mathit{f}\max_j$ en fonction de l'utilisation de M_j, exprimée par f_j, permet d'assurer le respect de la durée de vie maximale de chaque machine M_j. A chaque instant t, chaque $\mathit{f}\max_j$ n'évolue que si la machine M_j est utilisée, c'est-à-dire si $f_j(t) > 0$. Si la machine n'est pas utilisée, son débit maximal atteignable ne change pas. Ces deux cas de figure sont exprimés par les équations (6.1b) et (6.1c).

Ce programme mathématique n'étant pas convexe, une modification des différentes relations est proposée pour permettre sa résolution par des méthodes de résolution convexes. Associée à la possibilité pour les $f_j(t)$ d'être nuls à certains instants, la contrainte (6.1d) limitant les variations de chaque contribution $f_j(t)$ n'est tout d'abord pas convexe et ne peut donc pas être prise en compte en l'état. Les bornes de définition de chaque $f_j(t)$ sont alors modifiées de manière à inclure la valeur 0 dans leurs intervalles de définition. On a alors la relation définie par l'équation (6.2c). Une étape d'adaptation des solutions obtenues est intégrée à chaque méthode de résolution proposée dans la suite pour garantir le respect du débit minimal $\mathit{f}\min_j$ au cours de l'utilisation des machines.

Une seconde formulation de l'évolution des $\mathit{f}\max_j$ est ensuite proposée dans l'équation (6.2b). Cette expression ne respecte pas exactement l'évolution réelle des $\mathit{f}\max_j$, mais présente l'avantage d'être convexe. Un réglage des paramètres υ et μ permet d'adapter la forme prise par chaque $\mathit{f}\max_j$ pour qu'elle se rapproche au mieux de la forme réelle. Le programme convexe obtenu est détaillé dans le système d'équations (6.2).

$$\begin{cases} \min \quad \sigma(t) - \sum_{j=1}^{m} f_j(t) \qquad \forall\, 0 \leqslant t \leqslant T & (6.2a) \\ \text{t.q.} \quad f\max_j(t) \leqslant f\max_j(t-1) + \mu \cdot a_j \cdot (f_j(t-1))^{\upsilon} \quad \forall\, 1 \leqslant j \leqslant m,\ \forall 1 \leqslant t \leqslant T & (6.2b) \\ \qquad (\text{avec } \mu \text{ et } \upsilon \in \mathbb{R}^{*+} \text{ et } \upsilon > 1) & \\ \text{avec} \quad 0 \leqslant f_j(t) \leqslant f\max_j(t) \qquad \forall 1 \leqslant j \leqslant m,\ \forall 0 \leqslant t \leqslant T & (6.2c) \end{cases}$$

De manière à pouvoir utiliser des méthodes de résolution convexe, différentes modifications ont ainsi été apportées au modèle initial proposé dans le chapitre 3 pour des machines avec variation continue des performances (voir figure 3.4). Dans la suite des développements de ce chapitre, le comportement de chaque machine suit alors l'évolution simplifiée des caractéristiques représentée sur la figure 6.1.

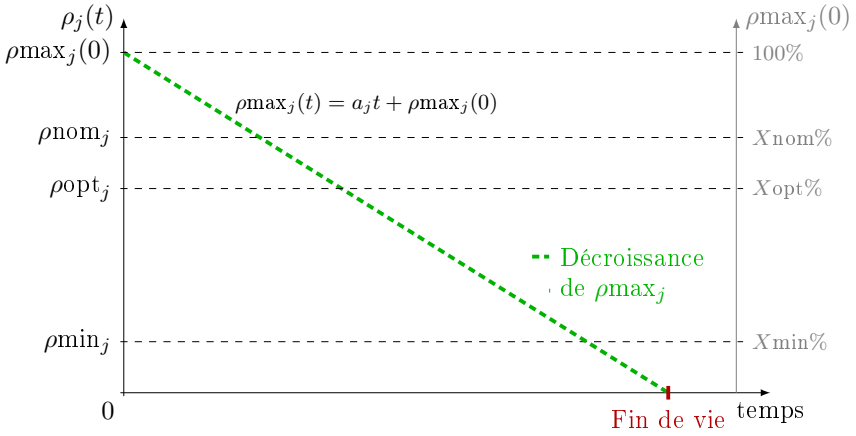


Figure 6.1 – Évolution simplifiée des caractéristiques d'une machine M_j avec une variation continue des performances

6.2 Méthodes de résolution

Différentes méthodes de résolution convexe sont proposées pour la résolution du programme mathématique défini précédemment. Une méthode mettant en œuvre des projections successives sur les contraintes est tout d'abord définie dans la partie 6.2.1. Cette première méthode ne respecte pas la vision globale visée dans ce chapitre pour la détermination de la solution, mais constitue un point de comparaison pour les autres méthodes proposées. Les projections définies sont de plus utilisées dans les deux méthodes suivantes pour accélérer leur convergence. Ces autres méthodes respectent une approche globale et suivent un raisonnement prenant en compte la totalité de l'horizon de production pour la détermination de chaque solution. La seconde méthode proposée dans la partie 6.2.2 est basée sur la résolution d'un problème primal-dual. Le principe de l'algorithme Mirror Prox for Saddle Points (MP-SP), proposé par Nemirovski [71], est utilisé pour optimiser une fonction objectif visant à garantir le respect des différentes contraintes du problème. La dernière méthode proposée dans la partie 6.2.3.1 utilise finalement la technique du Lasso (pour Least Absolute Shrinkage and Selection Operator) développée par Tibshirani [100] pour optimiser l'engagement des machines, associée à des projections sur certaines contraintes.

6.2.1 Projections

La première méthode proposée utilise des projections sur les contraintes pour construire la solution. L'idée générale est d'effectuer des projections convexes successives sur chacun des ensembles de contraintes définis par le programme convexe (6.2) pour générer une séquence de points supposée converger vers une solution du problème d'optimisation considéré [6]. Le processus global de projection est tout d'abord présenté, puis les différentes projections mises en œuvre tout au long de ce processus sont détaillées.

6.2.1.1 Algorithme de résolution

Le processus de résolution proposé, détaillé dans l'algorithme 12, met en oeuvre des projections successives sur les trois contraintes principales, à savoir : le respect de la demande, l'évolution du débit maximal atteignable et le respect de ce débit maximal. Le respect de la positivité des éléments de F est assuré par une initialisation positive de ces éléments ($f_j^{(0)}(t) \geqslant 0 \ \forall\ 1 \leqslant j \leqslant m,\ \forall\ 0 \leqslant t \leqslant T$).

Algorithme 12: Algorithme de résolution utilisant les projections

Données :
T : horizon de décision
F^{prec} : valeur de la solution F au début de chaque itération
Entrées :
$f\max_j(t)$: débit maximal atteignable à chaque temps t, pour chaque machine M_j
a_j : vitesse de décroissance du débit maximal $f\max_j(t)$
$\sigma(t)$: profil de demande
BS : borne supérieure pour l'horizon de production
ε : valeur de seuil pour la condition d'arrêt du processus de résolution
$erreur\_demande$: valeur de sous-production acceptée
ΔT : longueur de l'intervalle de temps sur lequel est appliquée projection sur la demande
$(meth, tri)$: méthode de projection sur la demande et tri des machines à appliquer
μ, υ : coefficients d'approximation convexe de l'évolution de $f\max$
Sorties :
$F = [f_1(t), f_2(t), \ldots, f_j(t), \ldots, f_m(t)]$: ordonnancement définissant les contributions de chaque machine M_j pour chaque temps t avec $0 \leqslant t \leqslant T$
K : horizon de production atteint par l'ordonnancement F

$T \leftarrow BS$
$f_j(t) \leftarrow 0\ \forall\ 1 \leqslant j \leqslant m,\ \forall\ 0 \leqslant t \leqslant T$
répéter
  $F^{prec} \leftarrow F$
  $F \leftarrow \text{projDemande}(\sigma, T, F, erreur\_demande, \Delta T, meth, tri)$ **(voir algorithme 13)**
  $f\max \leftarrow \text{MAJfmax}(F, f\max, a, \mu, \upsilon)$ **(voir algorithme 14)**
  $F \leftarrow \text{projfmax}(F, f\max)$ **(voir algorithme 15)**
jusqu'à $|F - F^{prec}| < \varepsilon$
$K \leftarrow$ horizon de production de la solution F
retourner F, K

L'évolution des vecteurs $f\max_j$ étant fonction des valeurs prises par chaque f_j, une détermination combinée de ces deux entités est requise. Plusieurs lancements successifs des projections liées à l'atteinte de la demande et à la mise à jour des débits maximum atteignables sont alors nécessaires pour optimiser la solution. Plusieurs itérations de l'enchaînement des trois projec-

tions sont ainsi effectuées pour définir un ordonnancement maximisant l'horizon de production, tout en respectant toutes les contraintes. Afin d'éviter une étape de détermination du nombre d'itérations nécessaires pour obtenir une solution satisfaisante, une condition d'arrêt est utilisée. Le nombre d'itérations nécessaires dépend en effet fortement de la configuration du problème considéré, à savoir le nombre de machines, la forme des contributions maximales atteignables $f\max_j$ et la valeur de la demande $\sigma(t)$. Le processus d'optimisation de la solution F est alors stoppé lorsque ses variations ne sont plus significatives, c'est-à-dire dès que la différence entre deux valeurs successives de F ($|F - F^{prec}|$) passe en dessous d'un certain seuil ε.

6.2.1.2 Détail des projections

La première contrainte considérée correspond au respect de la demande $\sigma(t)$. L'horizon de production est séparé en intervalles de temps de longueur ΔT, avec $1 \leqslant \Delta T \leqslant T$. Pour chaque intervalle, délimité par les temps t_{start} et t_{end} tel que $t_{start} \leqslant t \leqslant t_{end}$, la projection sur la demande est faite suivant une stratégie qui détermine à la fois les machines dont les contributions sont modifiées en premier lieu et la méthode globale de projection sur les dimensions temporelle et de débit. Les machines dont les contributions sont augmentées sont déterminées par un tri, dont les variantes sont *tri=proche* et *tri=loin* et la méthode de projection, dont les variantes sont $meth{=}\Delta Tsuff$, $meth{=}tsuff$, $meth{=}\Delta Tmin$ et $meth{=}\Delta Tmax$. Chaque variante des stratégies est détaillée dans les listes suivantes :

- Tri des machines :

  ***tri=proche* :** les machines M_j sont triées par ordre croissant de la différence $(f\max_j(t) - f_j(t))$, pour augmenter en priorité les contributions des machines dont les contributions f_j avant la projection sont les plus proches de leur contribution maximale atteignable $f\max_j$. Ce tri favorise la modification des contributions des machines déjà utilisées dans la solution. Avec $ecartfmax(j) = \max\left(f\max_j(t_{test}) - f_j(t_{test}), 0\right)$ l'écart de la contribution courante à la contribution maximale pour chaque machine M_j, l'indice de la machine $M_{j'}$ dont la contribution doit être modifiée est déterminé par :

  $$j' = \underset{1 \leqslant j \leqslant m | ecartfmax(j) > 0}{\operatorname{argmin}} ecartfmax(j) \tag{6.3}$$

  La définition de chaque valeur $ecartfmax(j)$ comme un maximum entre l'écart considéré et zéro permet d'éviter la prise en compte des machines dont les contributions dépassent les valeurs maximales avant la projection $(M_j \mid f_j(t) > f\max_j(t))$;

  ***tri=loin* :** les machines M_j sont triées par ordre décroissant de la différence $(f\max_j(t) - f_j(t))$, pour augmenter en priorité les contributions des machines dont les contributions f_j avant la projection sont les plus éloignées de leur contribution maximale atteignable $f\max_j$. De même que pour le tri précédent, l'indice de la machine $M_{j'}$ dont la contribution doit être modifiée est déterminé par :

  $$j' = \underset{1 \leqslant j \leqslant m | ecartfmax(j) > 0}{\operatorname{argmax}} ecartfmax(j) \tag{6.4}$$

Soit $M_{j'}$ la machine déterminée par le tri choisi.

– Méthode de projection :

***meth*=ΔT*suff* :** la contribution de la machine $M_{j'}$ est augmentée sur chaque intervalle de longueur ΔT de la quantité juste nécessaire pour atteindre la demande sur tout l'intervalle. Cette quantité vaut $ecart = \sigma(t_{end}) - \sum_{j=1}^{m} f_j(t_{end})$ et la contribution après projection de $M_{j'}$ est définie par :

$$f_{j'}(t) = \min\left(f_{j'}(t_{end}) + ecart,\, \mathit{f}\max_{j'}(t_{end})\right) \quad \forall\, t_{start} \leqslant t \leqslant t_{end} \tag{6.5}$$

***meth*=*tsuff* :** la contribution de la machine $M_{j'}$ est augmentée à chaque temps $0 \leqslant t \leqslant T$ de la valeur juste nécessaire pour atteindre la demande $\sigma(t)$. De manière similaire que pour *meth*=ΔT*suff*, la contribution après projection de $M_{j'}$ est définie par :

$$f_{j'}(t) = \min\left(f_{j'}(t) + ecart,\, \mathit{f}\max_{j'}(t)\right) \quad \forall\, 0 \leqslant t \leqslant T \tag{6.6}$$

***meth*=ΔT*min* :** la contribution de la machine $M_{j'}$ est augmentée à la valeur minimale prise par $\mathit{f}\max_j$ sur chaque intervalle de longueur ΔT. La contribution après projection de $M_{j'}$ est alors :

$$f_{j'}(t) = \min_{t_{start} \leqslant t \leqslant t_{end}} \mathit{f}\max_{j'}(t) = \mathit{f}\max_{j'}(t_{end}) \quad \forall\, t_{start} \leqslant t \leqslant t_{end} \tag{6.7}$$

***meth*=ΔT*max* :** la contribution de la machine $M_{j'}$ est augmentée à la valeur maximale prise par $\mathit{f}\max_j$ sur chaque intervalle de longueur ΔT. De même que pour *meth*=ΔT*min*, la contribution après projection de $M_{j'}$ est alors :

$$f_{j'}(t) = \max_{t_{start} \leqslant t \leqslant t_{end}} \mathit{f}\max_{j'}(t) = \mathit{f}\max_{j'}(t_{start}) \quad \forall\, t_{start} \leqslant t \leqslant t_{end} \tag{6.8}$$

Le processus de projection sur la demande est détaillé dans l'algorithme 13 pour toutes les stratégies décrites précédemment. Une fois les contributions des machines modifiées pour atteindre au mieux la demande $\sigma(t)$, les contributions maximales atteignables $\mathit{f}\max_j(t)$ sont mises à jour en fonction des valeurs prises au cours du temps par chaque $f_j(t)$ pour chaque machine M_j (voir algorithme 14). Cette mise à jour suit la relation définie par l'équation (6.2b). Une projection des contributions sur les contributions maximales mises à jour peut finalement être nécessaire pour rattraper d'éventuels dépassements et garantir le respect de la contrainte associée. Cette projection limite simplement chaque contribution $f_j(t)$ au $\mathit{f}\max_j(t)$ correspondant. Le processus est détaillé dans l'algorithme 15.

Algorithme 13: projDemande : projection sur la demande $\sigma(t)$

Données :
t_{start} : début de l'intervalle de projection
t_{end} : fin de l'intervalle de projection
t_{test} : temps de l'intervalle de projection pour lequel la comparaison de la contribution des machines f_j à la valeur maximale $f\max_j$ est effectuée
ecart : valeur de la production globale manquante pour atteindre la demande au temps t_{test}
ecartfmax(j) : écart entre $f\max_j(t_{test}$ et $f_j(t_{test})$ pour chaque machine M_j
Entrées :
$\sigma(t)$: profil de demande
T : horizon de décision
F : contributions initiales de chaque machine M_j, définies avant la projection
erreur_demande : valeur de sous-production acceptée
ΔT : longueur de l'intervalle de temps sur lequel est appliquée la projection
$(meth, tri)$: méthode de projection et tri des machines à appliquer
Sorties :
F : contributions des machines ajustées pour atteindre si possible la demande

$t_{start} \leftarrow 1$
tant que $t_{start} < T$ **faire**

$$t_{end} \leftarrow \begin{cases} t_{start} & \text{si } meth{=}tsuff\ ; \\ \min\left(t_{start} + \Delta T - 1, T\right) & \text{sinon} \end{cases}$$

$$t_{test} \leftarrow \begin{cases} t_{start} & \text{si } meth{=}\Delta Tmax\ ; \\ t_{end} & \text{sinon} \end{cases}$$

$$ecart \leftarrow \sigma(t_{test}) - \sum_{j=1}^{m} f_j(t_{test})$$
$$ecartfmax(j) \leftarrow \max\left(f\max_j(t_{test}) - f_j(t_{test}), 0\right)\ \forall\ j$$
si $ecart \leqslant erreur\_demande$ **alors**
$t_{start} \leftarrow t_{start} + 1$
sinon
tant que $ecart > erreur\_demande$ & $\sum_{j=1}^{m} ecartfmax(j) > erreur\_demande$ **faire**

$$j' \leftarrow \begin{cases} \operatorname{argmin}_{1 \leqslant j \leqslant m | ecartfmax(j) > 0} ecartfmax(j) & \text{si } tri{=}proche\ ; \\ \operatorname{argmax}_{1 \leqslant j \leqslant m | ecartfmax(j) > 0} ecartfmax(j) & \text{si } tri{=}loin \end{cases}$$

$$f_{j'}(t_{start} \leqslant t \leqslant t_{end}) \leftarrow \begin{cases} \min\left(f_{j'}(t_{start}) + ecart, f\max_{j'}(t_{end})\right) & \text{si } meth{=}\Delta Tsuff \\ & \text{ou } "tsuff"\ ; \\ \max_t f\max_{j'}(t) = f\max_{j'}(t_{start}) & \text{si } meth{=}\Delta Tmax\ ; \\ \min_t f\max_{j'}(t) = f\max_{j'}(t_{end}) & \text{si } meth{=}\Delta Tmin \end{cases}$$

$$ecart \leftarrow \sigma(test) - \sum_{j=1}^{m} f_j(t_{test})$$
$$ecartfmax(j) \leftarrow \max\left(f\max_j(t_{test}) - f_j(t_{test}), 0\right)\ \forall\ 1 \leqslant j \leqslant m$$

$$t_{start} \leftarrow \begin{cases} t_{start} + 1 & \text{si } meth{=}tsuff\ ; \\ t_{start} + \Delta T & \text{sinon} \end{cases}$$

retourner F

Algorithme 14: MAJfmax : mise à jour des contributions maximales *f*max

Entrées :
F : contributions des machines M_j avant la projection
$\mathit{f}\max = [\mathit{f}\max_1, \ldots, \mathit{f}\max_j, \ldots, \mathit{f}\max_m]$: vecteur des contributions maximales atteignables pour toutes les machines M_j avant la projection
a_j : vitesse de décroissance du débit maximal $\mathit{f}\max_j(t)$
μ, υ : coefficients d'approximation convexe de l'évolution de *f*max
Sorties :
*f*max : contributions maximales atteignables pour toutes les machines M_j après leur mise à jour en fonction des valeurs de F

$\mathit{f}\max_j(t) = \mathit{f}\max_j(t-1) + \mu a_j(f_j(t-1))^{\upsilon} \quad \forall\, 1 \leqslant j \leqslant m,\ \forall\, 1 \leqslant t \leqslant T$
retourner *fmax*

Algorithme 15: projfmax : projection sur les contributions maximales *f*max

Entrées :
F : contributions des machines M_j avant la projection
$\mathit{f}\max = [\mathit{f}\max_1, \ldots, \mathit{f}\max_j, \ldots, \mathit{f}\max_m]$: vecteur des contributions maximales atteignables pour toutes les machines M_j avant la projection
Sorties :
F : contributions des machines M_j limitées aux contributions maximales atteignables *f*max

$f_j(t) = \mathit{f}\max_j(t) \quad$ si $f_j(t) > \mathit{f}\max_j(t) \quad \forall\, 1 \leqslant j \leqslant m,\ \forall\, 0 \leqslant t \leqslant T$
retourner F

Les solutions obtenues avec cette méthode de projections successives sont définies par morceaux et les horizons de production atteints, qui correspondent au temps pendant lequel toutes les contraintes sont satisfaites (voir partie 6.2.4), dépendent fortement à la fois de l'initialisation du vecteur solution F et de la stratégie de projection appliquée. Cette première méthode ne respecte ainsi pas la vision globale visée dans ce chapitre pour la détermination de la solution et son efficacité est donc limitée. Elle permet toutefois de définir des solutions constituant des points de comparaison pour les méthodes proposées dans la suite.

6.2.2 Mirror Prox for Saddle Points (MP-SP)

La seconde méthode de résolution proposée est basée sur le principe de l'algorithme Mirror Prox, qui est une variante de l'algorithme Mirror Descent (algorithme de descente en miroir), développé par Nemirovsky et Yudin [72] pour la minimisation d'une fonction convexe sujette à des contraintes convexes. L'estimation de la variable à déterminer est efficace dans le sens où elle ne dépend que très peu de sa dimension. Cette méthode peut donc être utilisée pour résoudre des problèmes d'optimisation de très grande taille [7], ce qui est le but recherché dans ce chapitre.

6.2.2.1 Principe de l'algorithme Mirror Descent

La méthode du Mirror Descent est basée sur une résolution du problème de minimisation dans les espaces primal et dual, permettant l'intégration de contraintes. Une description très pédagogique de la méthode Mirror Descent a été proposée par Bubeck dans [22].

Soit ϕ la fonction objectif prise en compte, supposée différentiable sur son domaine de dé-

finition. Le but est de minimiser cette fonction, tout en respectant les différentes contraintes du problème. Soit $\mathcal{C}$ l'ensemble de ces contraintes. L'algorithme du Mirror Descent utilise une descente de gradient pour trouver un minimum de la fonction ϕ considérée. Cette descente de gradient étant effectuée dans l'espace dual, la stratégie du Mirror Descent est basée sur une fonction miroir (mirror map) θ. Cette fonction est définie sur un ensemble $\mathcal{D} \in \mathbb{R}^n$ $(n \geqslant 1)$, ouvert et convexe et tel que l'ensemble des contraintes $\mathcal{C}$ est inclus dans la frontière de $\mathcal{D}$ ($\mathcal{C} \in \overline{\mathcal{D}}$ et $\mathcal{C} \cap \mathcal{D} = \emptyset$). Elle respecte de plus les propriétés suivantes :

(i) θ est strictement convexe et dérivable ;
(ii) le gradient de θ peut prendre toutes les valeurs possibles : $\nabla\theta(\mathcal{D}) = \mathbb{R}^n$ $(n \geqslant 1)$;
(iii) le gradient de θ diverge aux frontières de l'ensemble $\mathcal{D}$: $\lim_{x \to \partial\mathcal{D}} \|\nabla\theta(x)\| = +\infty$.

Cette fonction miroir permet de faire la transition entre l'espace primal, dans lequel sont définies les contraintes du problème, et l'espace dual. Comme illustré sur la figure 6.2, un point $x_t \in \mathcal{C} \cap \mathcal{D}$ du primal est ainsi transformé en $\nabla\theta(x_t)$ dans l'espace dual. Un pas de gradient est ensuite appliqué à ce premier point dual, qui prend la forme définie par l'équation (6.9).

$$\nabla\theta(y_{t+1}) = \nabla\theta(x_t) - \lambda\nabla\phi(x_t) \qquad \lambda \geqslant 0 \tag{6.9}$$

Le point y_{t+1} n'étant pas forcément à l'intérieur de l'ensemble des contraintes $\mathcal{C}$, une étape de projection est finalement nécessaire. Cette dernière est effectuée par le biais d'une divergence de Bregman associée à la fonction miroir θ et définie par l'équation (6.10). La projection de $y_{t+1} \in \mathcal{D}$ sur $\mathcal{C}$ prend alors la forme définie par l'équation (6.11).

$$D_\theta(x, y) = \theta(x) - \theta(y) - \nabla\theta(y)^T(x - y) \tag{6.10}$$

$$x_{t+1} = P_{\mathcal{C}}(\nabla\theta(y_{t+1})) = \underset{x \in \mathcal{C} \cap \mathcal{D}}{\operatorname{argmin}} D_\theta(x, y_{t+1}) \tag{6.11}$$

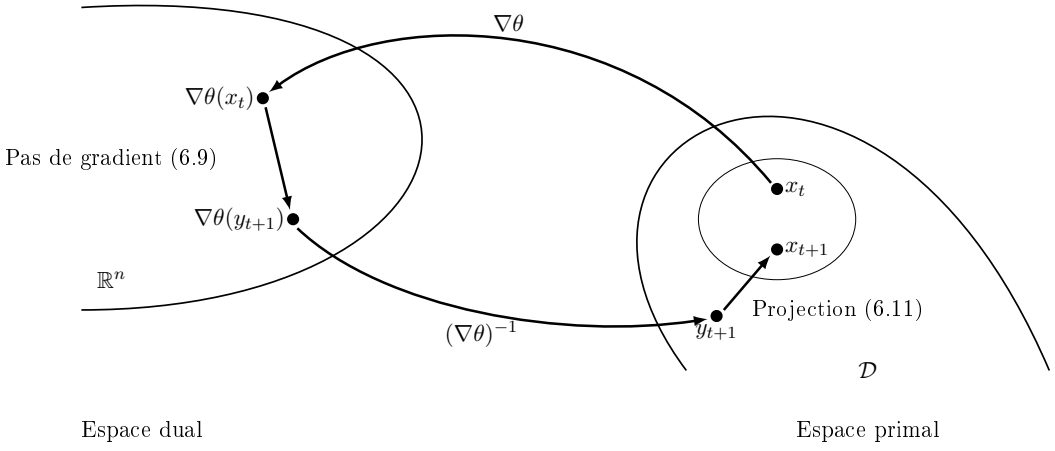


Figure 6.2 – Principe de fonctionnement de la méthode du Mirror Descent

6.2.2.2 Principe de l'algorithme du Mirror Prox

Une variante du Mirror Descent a été proposée par Nemirovski [71] sous la forme de l'algorithme du Mirror Prox, utilisable avec des fonctions convexes et différentiables à n'importe quel ordre dans leur domaine de définition. Comme illustré sur la figure 6.3, l'algorithme du Mirror Prox effectue une première étape de Mirror Descent pour aller de x_t à y_{t+1}, puis une seconde étape similaire pour obtenir x_{t+1}. Cette seconde étape part à nouveau de x_t, mais utilise cette fois ci le gradient de ϕ évalué en y_{t+1} (au lieu de x_t). Ces différentes étapes sont définies par les équations (6.12) à (6.15).

$$\nabla\theta(y'_{t+1}) = \nabla\theta(x_t) - \lambda\nabla\phi(x_t) \qquad \text{avec } \lambda \geqslant 0 \tag{6.12}$$

$$y_{t+1} = P_{\mathcal{C}}(\nabla\theta(y'_{t+1})) = \underset{x\in\mathcal{C}\cap\mathcal{D}}{\operatorname{argmin}} D_\theta(x, y'_{t+1}) \tag{6.13}$$

$$\nabla\theta(x'_{t+1}) = \nabla\theta(x_t) - \lambda\nabla\phi(y_{t+1}) \qquad \text{avec } \lambda \geqslant 0 \tag{6.14}$$

$$x_{t+1} = P_{\mathcal{C}}(\nabla\theta(x'_{t+1})) = \underset{x\in\mathcal{C}\cap\mathcal{D}}{\operatorname{argmin}} D_\theta(x, x'_{t+1}) \tag{6.15}$$

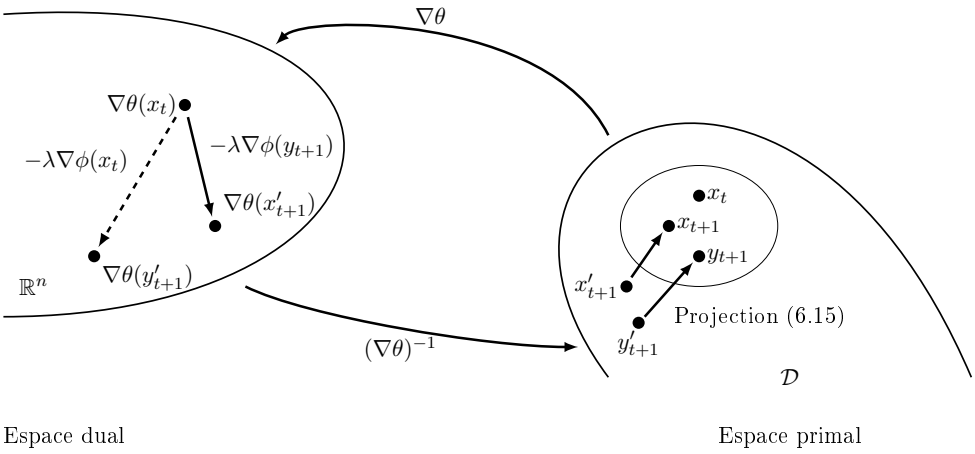


Figure 6.3 – Principe de fonctionnement de la méthode du Mirror Prox

6.2.2.3 Algorithme de résolution

L'algorithme de résolution proposé utilise le principe de l'algorithme du Mirror Prox. La résolution du problème primal-dual présentant une forte sensibilité aux variations des différents paramètres de la méthode, le respect des contraintes du problème d'optimisation est intégré dans l'objectif. La fonction objectif proposée, $\phi(F, f\max, \sigma)$, définie par l'équation (6.16), vise ainsi à satisfaire directement ces contraintes. Elle est composée de deux parties. La première, $\lambda_{dem}h_{dem}(F, \sigma)$, tend à satisfaire la demande $\sigma(t)$. La fonction $h_{dem}(F, \sigma)$, détaillée dans l'équation (6.17), permet de définir la variable F de manière à ce que la somme des contributions

des machines tende vers $\sigma(t)$. L'utilisation de la fonction exponentielle proposée permet de favoriser l'atteinte de la demande au début de l'ordonnancement, pour les premiers temps t, avec $0 \leqslant t \leqslant T$. La seconde partie de la fonction objectif, $\lambda_{slope} h_{slope}(F, f\max)$, avec $h_{slope}(F, f\max)$ détaillé dans l'équation (6.18), se rapporte à l'évolution des débits maximaux disponibles, $f\max_j$, en fonction des contributions des machines M_j, définies par chaque composante f_j de F.

$$\phi(F, f\max, \sigma) = \lambda_{dem} h_{dem}(F, \sigma) + \lambda_{slope} h_{slope}(F, f\max) \qquad \text{avec } \lambda_{dem},\ \lambda_{slope} > 0 \tag{6.16}$$

$$h_{dem}(F, \sigma) = \sum_{t=0}^{T} \frac{1}{T+1} \exp\left(-\gamma \left(\sum_{j=1}^{m} f_j(t) - \sigma(t) \right) \right) \qquad \text{avec } \gamma > 0 \tag{6.17}$$

$$h_{slope}(F, f\max) = \sum_{t=1}^{T} \sum_{j=1}^{m} \exp\left(\delta \left(f\max_j(t) - f\max_j(t-1) - \mu' a_j (f_j(t-1))^{\upsilon'} \right) \right) \tag{6.18}$$

$$\text{avec } \delta > 0,\ \mu' > 0,\ \upsilon' > 1$$

La fonction miroir θ prise en compte est définie sur $\mathbb{R}^{m(T+1)}$ par l'équation (6.19).

$$\theta(F) = \sum_{t=0}^{T} \sum_{j=1}^{m} F \ln(F) \tag{6.19}$$

Le gradient de cette fonction miroir θ, utilisé dans les étapes de Mirror Descent, est alors l'expression proposée dans l'équation (6.20).

$$\nabla\theta(F) = \ln(F) + 1 \tag{6.20}$$

Le Mirror Prox utilisé dans l'algorithme de résolution proposé est donc de la forme suivante (équations (6.21) à (6.24)) :

$$\nabla\theta(F^{(l+1)}) = \nabla\theta(F^{(l)}) - \lambda \nabla_F \phi(F^{(l)}, f\max, \sigma) \qquad \text{avec } \lambda \geqslant 0 \tag{6.21}$$

$$F_{int} = \exp\left(\nabla\theta(F^{(l+1)}) - 1 \right) \tag{6.22}$$

$$\nabla\theta(F^{(l+1)}) = \nabla\theta(F^{(l)}) - \lambda \nabla_{F_{int}} \phi(F^{(l)}, f\max, \sigma) \qquad \text{avec } \lambda \geqslant 0 \tag{6.23}$$

$$F^{(l+1)} = \exp\left(\nabla\theta(F^{(l+1)}) - 1 \right) \tag{6.24}$$

avec le gradient de ϕ défini par les Équations (6.25) à (6.27).

$$\nabla_F \phi(F^{(l)}, \mathit{fmax}, \sigma) = \lambda_{dem} \nabla_F h_{dem}(F, \sigma) + \lambda_{slope} \nabla_F h_{slope}(F, \mathit{fmax}) \quad (6.25)$$
$$\text{avec } \lambda_{dem},\ \lambda_{slope} > 0$$

$$\nabla_F h_{dem}(F, \sigma) = -\gamma \sum_{t=0}^{T} \frac{1}{T+1} \exp\Big(-\gamma \Big(\sum_{j=1}^{m} f_j(t) - \sigma(t) \Big) \Big) \qquad \text{avec } \gamma > 0 \quad (6.26)$$

$$\nabla_F h_{slope}(F, \mathit{fmax}) = -\delta \mu' \upsilon' \sum_{t=0}^{T} \sum_{j=1}^{m} a_j f_j(t-1)^{\upsilon'-1} \exp\Big(\delta \Big(\mathit{fmax}_j(t) - \mathit{fmax}_j(t-1) - \mu' a_j (f_j(t-1))^{\upsilon'} \Big) \Big) \quad (6.27)$$
$$\text{avec } \delta > 0,\ \mu' > 0,\ \upsilon' > 1$$

Afin d'améliorer la vitesse de convergence de la méthode, la distance de la solution à sa projection fictive sur $\sigma(t)$, $(F - F^{proj})$, est ajoutée à chaque pas de gradient effectué par chaque étape de Mirror Descent. Cela permet d'accélérer l'évolution de la solution vers une forme qui respecte au mieux les différentes contraintes. La nouvelle formulation du Mirror Prox est alors détaillée dans les équations (6.28) à (6.31) avec le vecteur F^{proj} déterminé par une projection de la solution sur la demande $\sigma(t)$ telle que définie dans l'algorithme 13 et $gF^{proj} = w_{grad}(F^{(l)} - F^{proj})$ avec le coefficient $w_{grad} > 0$:

$$\nabla\theta(F^{(l+1)}) = \nabla\theta(F^{(l)}) - \lambda \Big(\nabla_F \phi(F^{(l)}, \mathit{fmax}, \sigma) + gF^{proj} \Big) \qquad \text{avec } \lambda \geqslant 0 \quad (6.28)$$

$$F_{int} = \exp\Big(\nabla\theta(F^{(l+1)}) - 1 \Big) \quad (6.29)$$

$$\nabla\theta(F^{(l+1)}) = \nabla\theta(F^{(l)}) - \lambda \Big(\nabla_{F_{int}} \phi(F^{(l)}, \mathit{fmax}, \sigma) + gF^{proj} \Big) \qquad \text{avec } \lambda \geqslant 0 \quad (6.30)$$

$$F^{(l+1)} = \exp\Big(\nabla\theta(F^{(l+1)}) - 1 \Big) \quad (6.31)$$

L'utilisation du pas de gradient supplémentaire permet de plus de définir une condition d'arrêt pour le processus global détaillé dans l'algorithme 16, ce qui permet de s'affranchir de la détermination d'un nombre d'itérations adapté à la méthode. La modification de la solution F est stoppée lorsque les différences entre chaque composante de F et la composante correspondante de F^{proj} sont toutes inférieures à une certaine valeur. En d'autres termes, la méthode est arrêtée lorsque l'atteinte de la demande ne peut plus être améliorée. Après chaque étape de Mirror Prox, différentes projections sur les contraintes sont effectuées afin de garantir le respect des

contraintes et d'améliorer la vitesse de convergence de la solution. Une mise à jour des débits maximum atteignables $f\max$ en fonction des valeurs de f_j permet ainsi d'accélérer leur évolution, déjà gérée par le Mirror Prox. Une projection des contributions f_j de chaque machine M_j sur le débit maximal atteignable correspondant, $f\max_j$, permet ensuite de respecter l'évolution de la valeur du débit maximal.

Algorithme 16: Algorithme de résolution utilisant le Mirror Prox for Saddle Points

Données :
T : horizon de décision
l : numéro d'itération pour l'optimisation de la solution F
F^{proj} : projection de la solution F sur la demande $\sigma(t)$
Entrées :
$f\max_j(t)$: débit maximal atteignable à chaque instant t, pour chaque machine M_j
a_j : vitesse de décroissance du débit maximal $f\max_j(t)$
$\sigma(t)$: profil de demande
λ : poids pour la détermination de F dans l'étape de Mirror Descent
w_{grad} : poids pour la détermination du pas de gradient pour le respect de la demande
ε : valeur de seuil pour la condition d'arrêt du processus de résolution
$erreur\_demande$: valeur de sous-production acceptée
ΔT : longueur de l'intervalle de temps sur lequel est appliquée projection sur la demande
$(meth, tri)$: méthode de projection sur la demande et tri des machines à appliquer
μ, υ : coefficients d'approximation convexe de l'évolution de $f\max$
Sorties :
$F = [f_1(t), f_2(t), \ldots, f_j(t), \ldots, f_m(t)]$: ordonnancement définissant les contributions de chaque machine M_j pour chaque temps t avec $0 \leqslant t \leqslant T$
K : horizon de production atteint par l'ordonnancement F

$l \leftarrow 0$
$f_j^{(l)}(t) \leftarrow 0 \; \forall \; 1 \leqslant j \leqslant m, \; \forall \; 0 \leqslant t \leqslant T$
$gF_j^{proj}(t) \leftarrow 0 \; \forall \; 1 \leqslant j \leqslant m, \; \forall \; 0 \leqslant t \leqslant T$
répéter
  // 1ère étape : Mirror Descent
  $mF^{(l)} \leftarrow \ln(F) + 1$
  $gF \leftarrow \nabla_F \phi(F, f\max)$
  $mF^{(l+1)} \leftarrow mF^{(l)} - \lambda(gF + gF^{proj})$
  $F_{int} \leftarrow \exp(mF^{(l+1)} - 1)$
  // 2nde étape : Mirror Prox
  $l \leftarrow l + 1$
  $gF \leftarrow \nabla_{F_{int}} \phi(F, f\max)$
  $mF^{(l+1)} \leftarrow mF^{(l)} - \lambda(gF + gF^{proj})$
  $F^{(l+1)} \leftarrow \exp(mF^{(l+1)} - 1)$
  $l \leftarrow l + 1$
  // Projection sur les contraintes :
  $f\max \leftarrow$ MAJfmax$(F^{(l)}, f\max, a, \mu, \upsilon)$**(voir algorithme 14)**
  $F^{(l+1)} \leftarrow$ projfmax$(F^{(l)}, f\max)$**(voir algorithme 15)**
  // Pas de gradient avec projection sur la demande :
  $F^{proj} \leftarrow$ projDemande$(\sigma, T, F^{proj}, erreur_demande, \Delta T, meth, tri)$**(voir algorithme 13)**
  $l \leftarrow l + 1$
  $gF^{proj} \leftarrow w_{grad}(F^{(l)} - F^{proj})$
jusqu'à $|F^{(l)} - F^{proj}| < \varepsilon$
$K \leftarrow$ horizon de production de la solution $F^{(l)}$
retourner $F^{(l)}$, K

Avec cette seconde méthode de résolution, les solutions sont définies de façon globale sur tout l'horizon de décision. La détermination de l'horizon de production atteint par chaque solution est détaillée dans la partie 6.2.4.

6.2.3 Méthode de régression linéaire : Lasso

Pour la dernière méthode de résolution du problème d'optimisation, l'utilisation d'une pénalisation ℓ_1 est proposée, sur la base de l'intuition suivante : pour maximiser l'horizon de production, il vaudrait mieux n'utiliser que les machines strictement nécessaires à chaque instant pour atteindre la demande. Cela revient à minimiser la surproduction tout en minimisant le nombre de machines utilisées en même temps, afin de conserver du potentiel pour la fin de l'ordonnancement et éventuellement atteindre la demande pendant un temps plus long. Cela peut être fait par le biais du contrôle de la sparsité [1] du vecteur solution F, qui a alors la même structure que si le nombre de composantes non nulles étaient contraint.

La pénalisation par une norme ℓ_1 a fait l'objet de nombreux travaux dans les domaines de l'intelligence artificielle pour l'apprentissage automatique (machine learning), des statistiques et du traitement du signal [37, 27]. Elle a en particulier été proposée pour la recherche d'une solution creuse pour un système d'équations linéaires sous déterminé (qui admet une infinité de solutions) [37] et pour l'approximation affine par morceaux [59]. Ces différents travaux ont conduit à d'importantes découvertes dans les domaines des mathématiques et de l'informatique en lien avec la frontière entre P et NP [26]. Dans le domaine des statistiques, l'utilisation d'une norme ℓ_1 pour une régression linéaire se retrouve dans l'algorithme du Lasso (pour Least Absolute Shrinkage and Selection Operator), proposé par Tibshirani [100]. Le cas du Lasso est une instance très particulière d'approche par relaxation convexe d'un problème NP dur. Elle a en effet conduit à soulever la question de la probabilité avec laquelle l'approche convexe permet de retrouver le support de la solution, à une époque où la majeure partie de la communauté de la théorie de la complexité se préoccupait essentiellement de donner des schémas d'approximation pire-cas.

C'est ce type d'approche que nous nous proposons d'utiliser pour résoudre le problème d'optimisation considéré. le principe du Lasso est tout d'abord expliqué dans la partie 6.2.3.1, puis l'algorithme de résolution basé sur le Lasso est détaillé dans la partie 6.2.3.2.

6.2.3.1 Principe du Lasso

Le Lasso est une technique de régression linéaire basée sur la fonctionnelle des moindres carrés, à laquelle est ajoutée une pénalisation des coefficients de régression prenant la forme d'une norme ℓ_1. Cette norme ℓ_1 est un substitut pour la quasi-norme ℓ_0, qui, par définition, mesure le nombre de composantes non nulles d'un vecteur. La solution du problème d'optimisation est appelée un estimateur Lasso. Pour le problème considéré ici et suivant les notations introduites précédemment, l'estimateur Lasso $\widehat{F}$ est défini par l'équation (6.32) :

1. La notion de « sparsité » peut être rapprochée de celle de densité. On dit qu'un vecteur est « sparse » ou « creux » lorsqu'une très grande proportion de ses composantes est nulle (voir [26]).

$$\widehat{F} = \underset{F\in\mathbb{R}^{m(T+1)}}{\operatorname{argmin}} \left(\frac{1}{2} \left\| \sigma(t) - \sum_{j=1}^{m} f_j(t) \right\|_2^2 + \lambda \left\| F \right\|_1 \right) \quad \forall\ 0 \leqslant t \leqslant T, \text{ avec } \lambda \in \mathbb{R}^+ \tag{6.32}$$

Chaque composante de F étant par hypothèse positif, cet estimateur Lasso peut être exprimé de la façon suivante :

$$\widehat{F} = \underset{F\in\mathbb{R}^{m(T+1)}}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{t=0}^{T} \left(\sigma(t) - \sum_{j=1}^{m} f_j(t) \right)^2 + \lambda \sum_{t=0}^{T} \sum_{j=1}^{m} f_j(t) \right) \quad \text{avec } \lambda \in \mathbb{R}^+ \tag{6.33}$$

L'objectif est d'estimer les composantes de F, tout en en éliminant certaines en ramenant leurs coefficients à zéro. Le Lasso est ainsi une méthode effectuant simultanément une estimation et une sélection de variables [100]. La sélection est effectuée par la pénalisation ℓ_1, définie par le second terme de l'équation (6.32). Cette partie de l'estimateur Lasso fait ainsi tendre de façon continue des composantes du vecteur F vers 0 lorsque λ augmente. Lorsque λ est assez grand, certaines composantes $f_j(t)$ sont fixées à 0. Cela impose une certaine sparsité à la solution F. La définition d'un coefficient λ approprié permet ainsi de faire une sélection des machines et de ne pas toutes les utiliser à chaque instant. Cela est cohérent avec l'objectif de maximisation de l'horizon. Cette sélection des machines à utiliser à chaque instant est faite conjointement avec la détermination de leur contribution au débit total fourni. Cette étape d'estimation est effectuée par le premier terme de l'estimateur Lasso défini dans l'équation (6.32). Ce premier terme permet d'assurer que le débit total fourni par la plate-forme de machines considérée atteint au moins la demande $\sigma(t)$ à chaque instant t, sachant que la détermination de chaque composante $f_j(t)$ est faite sous la contrainte $0 \leqslant f_j(t) \leqslant f\max_j(t)\ \forall\ 0 \leqslant t \leqslant T,\ \forall\ 1 \leqslant j \leqslant m$.

6.2.3.2 Algorithme de résolution

La technique du Lasso présentée dans la partie précédente est utilisée pour la résolution du problème d'optimisation avec une de ses variantes, l'Adaptive Lasso. Cette technique, proposée par Zou dans [108], est une version adaptée du Lasso classique présenté précédemment, dans laquelle la pénalisation ℓ_1 est pondérée. Toutes les composantes de F ne sont pas pénalisées de manière égale, mais suivant un vecteur $w = [w_1(t), \ldots, w_j(t), \ldots, w_m(t)]$. Cette pondération est utilisée pour adapter la pénalisation du vecteur à optimiser à sa valeur à l'itération précédente. Chaque $w_j(t)$ est alors défini en fonction du $f_j(t)$ correspondant. Cela permet d'accélérer la vitesse de convergence de la méthode globale. Il a été démontré par Zou que l'Adaptive Lasso est aussi efficace que d'autres techniques de modélisation sparse [108].

$$\widehat{F} = \underset{F}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{t=0}^{T} \left(\sigma(t) - \sum_{j=1}^{m} f_j(t) \right)^2 + \lambda \sum_{t=0}^{T} \sum_{j=1}^{m} w_j(t) f_j(t) \right) \quad \text{avec } \lambda \in \mathbb{R}^+ \tag{6.34}$$

La méthode globale de résolution, détaillée dans l'algorithme 17, comporte une première

application du Lasso classique, utilisée pour initialiser le vecteur F. Ce même vecteur est ensuite optimisé sur plusieurs itérations. Au cours de chacune de ces itérations, un ou plusieurs Adaptive Lasso sont appliqués, suivis d'une mise à jour des débits maximum atteignables en fonction des valeurs des f_j (voir algorithme 14) et d'une projection des contributions f_j de chaque machine M_j sur le débit maximal atteignable correspondant $\mathit{f}\max_j$ (voir algorithme 15). De même que pour la méthode des projections successives, l'optimisation du vecteur solution F est stoppée lorsque ses variations ne sont plus significatives. La méthode utilisant le principe du Lasso est alors répétée tant que la différence entre deux valeurs successives de F reste supérieure à un certain seuil ε : $|F - F^{prec}| > \varepsilon$.

Algorithme 17: Algorithme de résolution utilisant la technique du Lasso

Données :
T : horizon de décision
l : numéro d'itération pour l'optimisation de la solution F
w : vecteur de pondération de la pénalisation ℓ_1
Entrées :
$\mathit{f}\max_j(t)$: débit maximal atteignable à chaque temps t, pour chaque machine M_j
a_j : vitesse de décroissance du débit maximal $\mathit{f}\max_j(t)$
$\sigma(t)$: profil de demande
$ITlasso$: nombre d'itérations de la méthode de l'Adaptive Lasso
λ : coefficient de la pénalisation ℓ_1 de la méthode du Lasso
ε : valeur de seuil pour la condition d'arrêt du processus de résolution
μ, υ : coefficients d'approximation convexe de l'évolution de $\mathit{f}\max$
Sorties :
$F = [f_1(t), f_2(t), \ldots, f_j(t), \ldots, f_m(t)]$: ordonnancement définissant les contributions de chaque machine M_j pour chaque temps t avec $0 \leqslant t \leqslant T$
K : horizon de production atteint par l'ordonnancement F

$l \leftarrow 0$
$f_j^{(l)}(t) \leftarrow 0 \quad \forall\, 1 \leqslant j \leqslant m,\ \forall\, 0 \leqslant t \leqslant T$
`// Première application du Lasso classique :`
$F^{(l+1)} \leftarrow \text{argmin}_{F^{(l)}} \left(\frac{1}{2} \sum_{t=0}^{T} \left(\sigma(t) - \sum_{j=1}^{m} f_j^{(l)}(t) \right)^2 + \lambda \sum_{j=1}^{m} f_j^{(l)}(t) \right) \quad \forall\, 0 \leqslant t \leqslant T$
$l \leftarrow l + 1$
répéter
  $F^{prec} \leftarrow F^{(l)}$
  `// Application de l'Adaptive Lasso :`
  pour $itlasso = 1$ *à* $ITlasso$ **faire**
    $w \leftarrow$ mise à jour des pondérations de la pénalisation ℓ_1
    $F^{(l+1)} \leftarrow \text{argmin}_{F^{(l)}} \left(\frac{1}{2} \sum_{t=0}^{T} \left(\sigma(t) - \sum_{j=1}^{m} f_j^{(l)}(t) \right)^2 + \lambda \sum_{t=0}^{T} \sum_{j=1}^{m} w_j(t) f_j^{(l)}(t) \right)$
    $l \leftarrow l + 1$
  `// Projections sur les contraintes :`
  $\mathit{f}\max \leftarrow \text{MAJfmax}(F^{(l)}, \mathit{f}\max, a, \mu, \upsilon)$**(voir algorithme 14)**
  $F^{(l+1)} \leftarrow \text{projfmax}(F^{(l)}, \mathit{f}\max)$**(voir algorithme 15)**
  $l \leftarrow l + 1$
jusqu'à $|F^{(l)} - F^{prec}| < \varepsilon$
$K \leftarrow$ horizon de production de la solution $F^{(l)}$
retourner $F^{(l)}$, K

6.2.4 Étape de post-processing

Avec chacune des méthodes de résolution proposées dans ce chapitre, la construction des solutions est faite de telle sorte que les instants pour lesquels la demande est atteinte sont rassemblés en début d'ordonnancement. La forme des fonctions modélisant l'évolution des contributions maximales atteignables $f\max_j$ joue aussi un rôle dans cette propriété des solutions. Chacune de ces fonctions étant strictement décroissante avec l'utilisation de la machine associée, la somme des contributions maximales atteignables passe sous la demande après un certain temps d'ordonnancement t, sans pouvoir repasser au-dessus de la valeur $\sigma(t)$. Ce fonctionnement permet d'appliquer les méthodes de résolution sur des horizons de décision T sur-dimensionnés tout en restant compatible avec l'objectif de maximisation de l'horizon de production. L'horizon $K\Delta T$ associé à chaque solution de longueur T correspond ainsi au temps maximal pendant lequel toutes les contraintes sont satisfaites. En pratique, la satisfaction stricte de la contrainte liée à $F\max$ étant forcée lors de chaque résolution, K correspond au nombre maximum de périodes consécutives pendant lequel la demande $\sigma(t)$ est atteinte.

La contrainte associée au respect du débit minimal $\rho\min_j$ pour chaque machine M_j ayant été convexifiée, une modification des solutions peut être nécessaire pour satisfaire toutes les contraintes initiales du problème. Les contributions des machines M_j qui sont strictement supérieures à zéro mais inférieures au débit minimal $\rho\min_j$ à un instant t ($0 < \rho_j(t) < \rho\min_j$) sont fixées à $\rho\min_j$. Pour le modèle considéré, tous les débits ρ_j tels que $0 < \rho_j \leqslant \rho\min_j$ ($1 \leqslant j \leqslant m$) étant associés à la même durée de vie, chaque modification à un instant t' n'a aucun impact sur les décisions faites pour les temps $t > t'$. Les solutions après modification étant basées sur les solutions convexes obtenues avec une des méthodes proposées, elles restent proches de l'optimal convexe.

6.3 Résultats de simulation

6.3.1 Génération des problèmes

De même que pour les résultats présentés dans les chapitres précédents, différentes instances du problème général d'optimisation considéré ont été générées à l'aide d'un simulateur et configurées avec différents paramètres. Chaque instance correspond à une plate-forme donnée de machines, caractérisée par un nombre m de machines dotées de propriétés intrinsèques (débit maximal et minimal, durée de vie optimale, vitesse de dégradation, etc.). Le paramétrage des plates-formes de machines prises en compte pour les simulations suit le même principe que celui détaillé dans le chapitre 5 pour l'approche optimale par morceaux (voir partie 5.2.1.1). Il en est de même pour le paramétrage de la demande $\sigma(t)$ (voir partie 5.2.1.2).

6.3.2 Résultats préliminaires

6.3.2.1 Paramètres communs

Quelque soit la méthode de résolution utilisée, la valeur initiale de chaque contribution est tout d'abord fixée à zéro : $f_j(t) = 0 \; \forall \; 1 \leqslant j \leqslant m, \; \forall \; 0 \leqslant t \leqslant T$. Les valeurs des différents paramètres intrinsèques aux méthodes de résolution ont ensuite été déterminés sur la base de

simulations. Les valeurs retenues sont détaillées dans la suite pour chaque méthode.

Afin d'éviter la détermination d'un nombre d'itérations en fonction des données du problème, des conditions d'arrêt basées sur l'évolution de la solution F au cours de sa construction ont été proposées pour chaque méthode de résolution. Toutes ces conditions sont associées à un paramètre ε, qui correspond à une valeur de seuil. Afin d'adapter la condition d'arrêt à la dimension du problème considéré, cette valeur a été définie pour chaque simulation en fonction de la valeur de la demande moyenne σ. Pour les résultats développés dans la suite, ce paramètre a été fixé à $\varepsilon = 0.1 \cdot \sigma(t)$.

Une certaine marge d'erreur a ensuite été acceptée pour l'atteinte de la demande. Cela se traduit par la mise en place de deux valeurs de seuil supplémentaires. La première, fixée à $0.01 \cdot \sigma(t)$, correspond à l'erreur acceptée lors de la construction de la solution F. La seconde est utilisée pour la détermination de l'horizon atteint et a été fixée à $0.1 \cdot \sigma(t)$. Uniquement les temps t pour lesquels la contribution totale ρ_{tot} des machines atteint $0.9 \cdot \sigma(t)$ sont ainsi pris en compte pour le calcul de l'horizon de production $K\Delta T$.

6.3.2.2 Paramétrage de la méthode de projection

Les différentes stratégies de projection proposées pour la première méthode de résolution mettant en œuvre des projections successives sur les contraintes sont tout d'abord comparées entre elles. Cela permet de déterminer la stratégie donnant les meilleurs résultats en termes d'horizon de production atteint, qui pourra être comparée aux autres méthodes de résolution convexe développées pour le problème d'optimisation considéré.

Les deux types de projection, convexifiée ou basée sur l'évolution réelle des contributions maximales en fonction de l'utilisation des machines, sont tout d'abord comparées pour une stratégie de projection. Les valeurs des paramètres utilisées pour l'expression convexifiée de l'évolution des $f\max_j$ (voir équation (6.2b)) sont les suivantes : $\mu = 0.2$, $\upsilon = 0.3$. On peut voir sur la figure 6.4 que la convexification de la projection sur les contributions maximales disponibles pénalise les horizons de production atteints. La convexification de l'évolution des $f\max_j$ est en effet pessimiste en termes de potentiel restant, ce qui limite les horizons atteignables. On peut toutefois voir sur la même figure que la pénalisation reste limitée à 40% en moyenne et à 58% au maximum, pour les grandes charges.

On peut ensuite voir sur la figure 6.5 que les stratégies utilisant le tri *tri=loin* donnent de meilleurs résultats en termes d'horizon de production atteint que celles utilisant le tri *tri=proche*. Le premier tri favorise en effet une homogénéité dans l'utilisation des machines permettant la plupart du temps d'étendre l'horizon de production. Le second tri donne des résultats un peu moins performants, mais présente l'avantage de favoriser une continuité d'utilisation des machines engagées, ce qui est important au regard de l'application aux piles à combustible envisagée.

Les différentes stratégies de projection comparées se séparent en deux groupes distincts. Le premier est constitué des méthodes $meth{=}\Delta Tsuff$ et $meth{=}\Delta Tmin$, qui respectent une construction de chaque contribution par paliers. Cela permet de respecter une préférence de comportement liée à l'application des piles à combustible donnée en exemple. Le second groupe est constitué des méthodes $meth{=}\Delta Tmax$ et $meth{=}tsuff$. Ces dernières donnent de meilleurs résultats en termes d'horizon de production atteint, mais ne permettent pas d'obtenir des so-

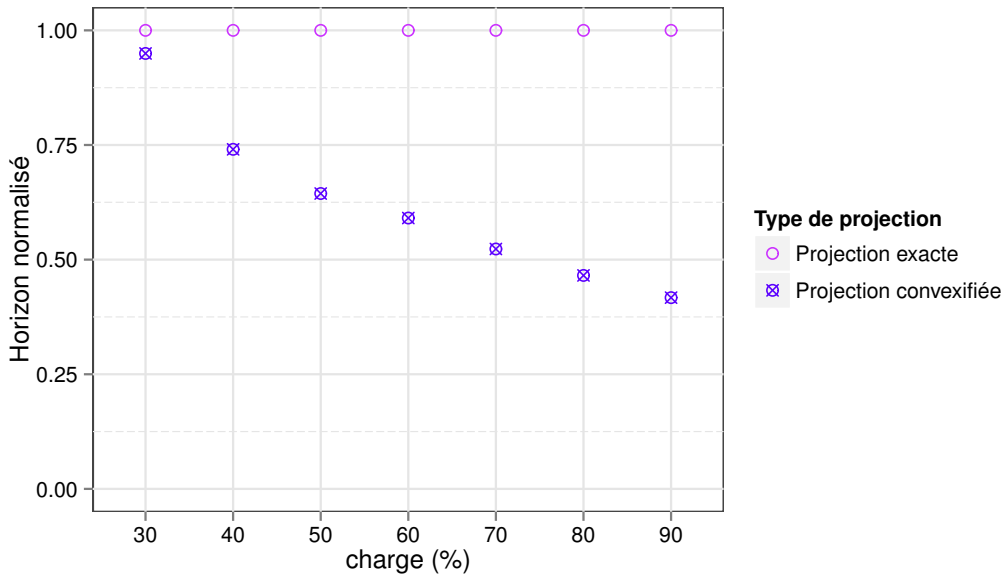


Figure 6.4 – Effet de la convexification de la projection sur *F*max – m = 25, *meth=tsuff, tri=proche*

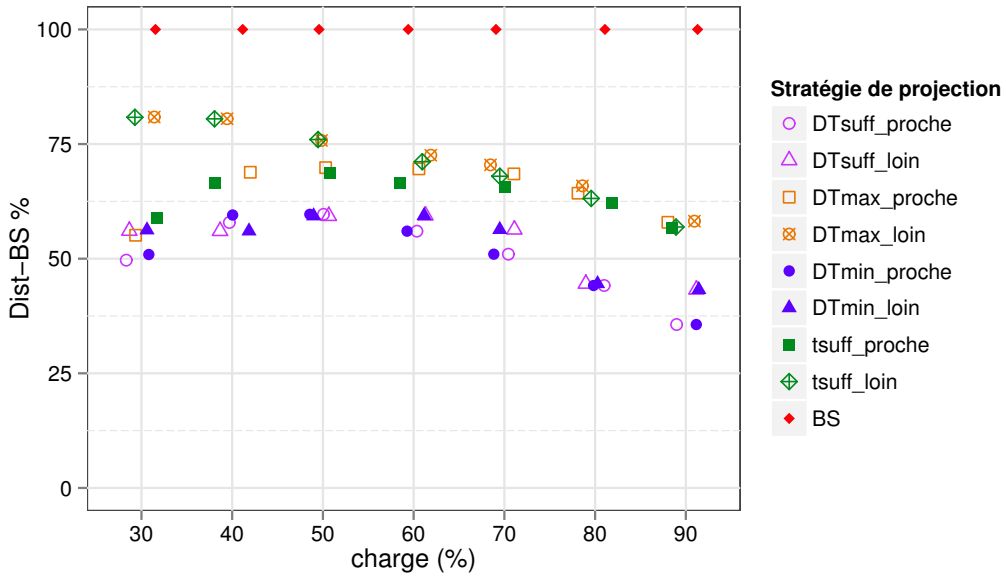


Figure 6.5 – Distance à la borne supérieure BS des horizons de production atteints avec les différentes stratégies de projection – m = 25 machines

lutions constantes par morceaux pour chaque machine. La méthode *meth=tsuff* construit en effet les contributions temps par temps. La méthode *meth=*$\Delta Tmax$ les construit quant à elle tout d'abord par paliers, mais en surestimant les capacités des machines, modélisées par les $f\max_j$. Cette surestimation est ensuite corrigée par des projections sur chaque $f\max_j$, qui modifient les contributions temps par temps. Dans la suite, les résultats obtenus avec les autres méthodes de résolution convexe sont comparés à ceux obtenus avec la stratégie de projection la plus simple permettant d'atteindre les meilleurs horizons de production, utilisant le tri *tri=loin* et la méthode *meth=tsuff*.

6.3.2.3 Paramétrage de l'algorithme MP-SP

Les différents paramètres de la méthode de résolution basée sur l'algorithme du Mirror Prox ont été fixés de la manière suivante : $\lambda = 5 \times 10^{-5}$, $\lambda_{dem} = 100$, $\lambda_{slope} = 100$, $\mu' = 1$, $\upsilon' = 1$, $\delta = 100$, $\gamma = 100$, $w_{grad} = 10$. Les valeurs des paramètres prises en compte pour la projection sur les $f\max_j$ sont les suivantes : $\mu = 0.2$, $\upsilon = 0.3$.

6.3.2.4 Paramétrage de la méthode du Lasso

Le paramètre utilisé dans l'estimateur Lasso a tout d'abord été fixé en fonction du nombre de machines considérées : $\lambda = 10 \cdot m$. Cela permet d'adapter le contrôle du caractère creux de la solution à la taille du problème d'optimisation considéré. Un seul lancement de l'Adaptive Lasso par itération suffit à guider l'évolution de la solution F ($ITlasso = 1$). Pour les projections sur les contributions maximales atteignables $f\max_j$, les paramètres ont été fixés de la même manière que pour la méthode utilisant le principe du Mirror Prox : $\mu = 0.2$, $\upsilon = 0.3$. La mise à jour des pondérations de la pénalisation ℓ_1 est enfin faite suivant le système d'équations (6.35).

$$w_j(t) \leftarrow \begin{cases} \min\left(\frac{1}{f_j^{(l)}(t)}, 10\right) & \text{si } f_j(t) > 0,\ \forall\ 1 \leqslant j \leqslant m,\ \forall\ 0 \leqslant t \leqslant T\,; \\ 1 & \text{si } f_j(t) = 0,\ \forall\ 1 \leqslant j \leqslant m,\ \forall\ 0 \leqslant t \leqslant T \end{cases} \tag{6.35}$$

6.3.3 Comparaison des méthodes de résolution

L'efficacité des trois méthodes de résolution proposées dans ce chapitre est validée pour une demande constante $\sigma(t) = \sigma\ \ \forall 0 \leqslant t \leqslant T$. Les résultats obtenus sont comparés par le biais d'une distance à une borne supérieure, définie par l'équation (6.36). Cette borne correspond à la somme pour chaque machine M_j du potentiel disponible, représenté par la surface sous la courbe $f_j(t) = f\max_j(t)$ divisée par la valeur de la demande σ.

$$\mathrm{BS} = \left\lfloor \frac{\sum_{j=1}^{m} 0.6 \cdot f\max_j(0) \cdot RUL(f\min_j)}{\sigma} \right\rfloor \tag{6.36}$$

La figure 6.6 montre la distance à la borne supérieure BS des horizons de production atteints avec les différentes méthodes convexes proposées en fonction de la charge. On peut voir sur cette figure que la méthode des projections successives donne les moins bons résultats. Comme expliqué dans la partie 6.2.1, cette méthode ne détermine pas l'ordonnancement de manière globale sur tout l'horizon de production, mais sur la base d'une discrétisation du temps. Les deux autres méthodes, qui respectent une vision globale, permettent d'atteindre de meilleurs horizons de production. Cela valide le choix d'une approche globale pour la construction des solutions, définissant ces dernières en prenant en compte tout l'horizon de production. Alors que la méthode des projections atteint 39% de la borne supérieure en moyenne, les méthodes basées sur les algorithmes du Mirror Prox et du Lasso atteignent en moyenne respectivement 64.3% et 65.7% de la borne supérieure BS. Ces deux dernières méthodes présentent des performances

similaires en termes d'optimisation de l'horizon de production, mais fournissent des solutions de formes bien différentes et ne nécessitent pas le même temps de résolution. Ces deux aspects sont développés dans la suite.

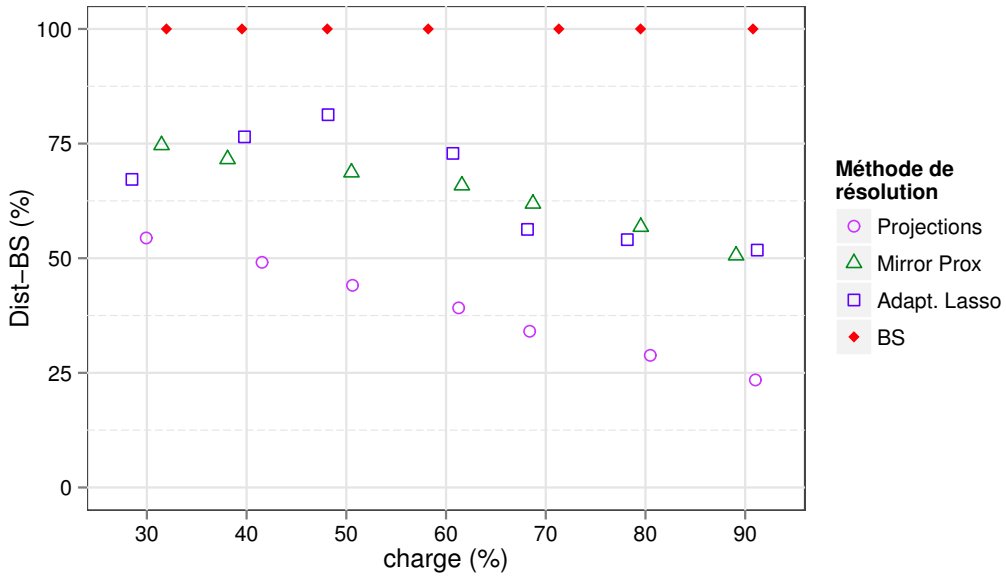
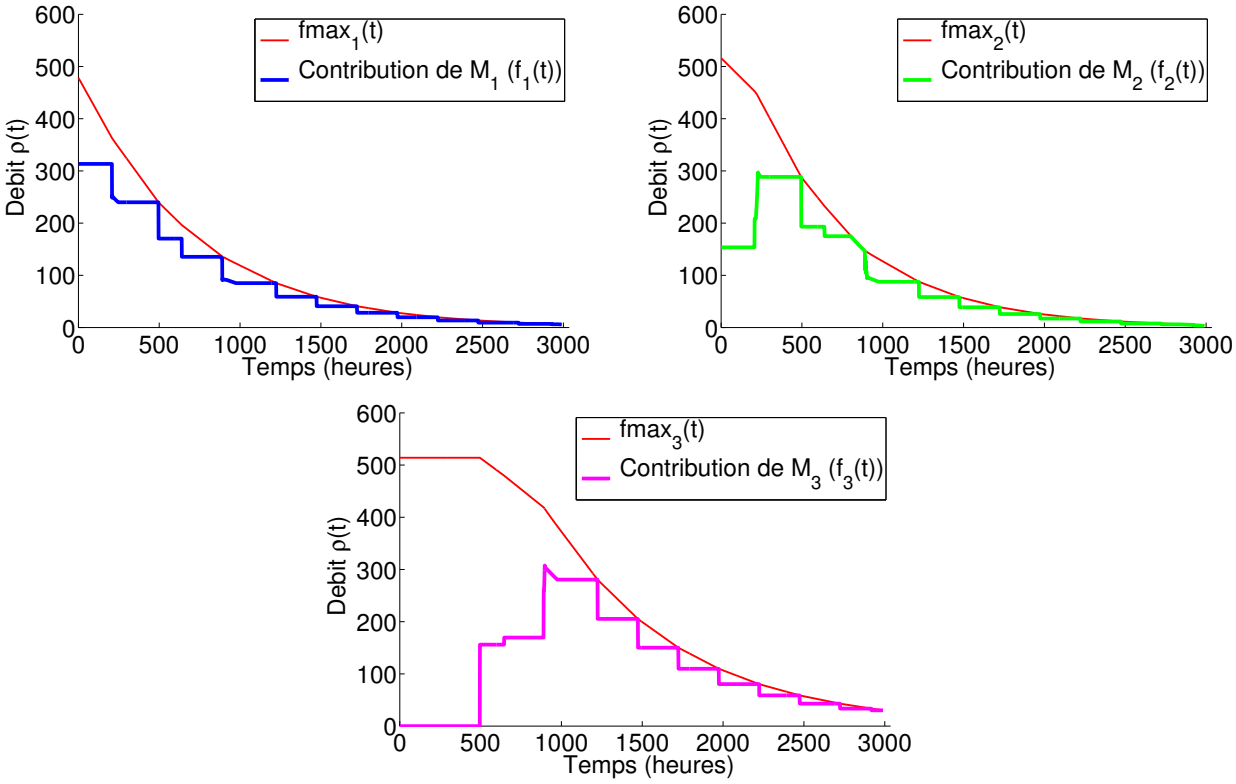


Figure 6.6 – Distance à la borne supérieure BS des horizons de production atteints avec les différentes méthodes de résolution convexe – m = 25 machines

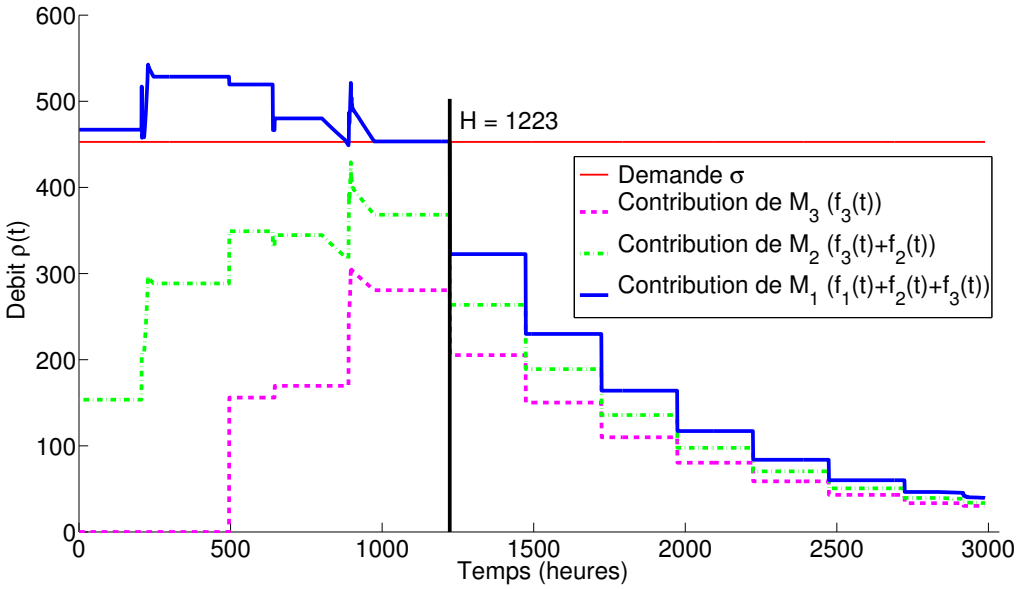
Les figures 6.7, 6.8 et 6.9 montrent des ordonnancements obtenus respectivement avec la méthode des projections successives, l'algorithme du Mirror Prox et la méthode du Lasso pour un même ensemble de trois machines et une demande σ correspondant à 40% du débit total maximal atteignable. L'évolution des contributions de chaque machine à la production totale est aussi montrée sur ces mêmes figures. On peut tout d'abord voir sur la figure 6.7 que la méthode mettant en œuvre des projections successives sur les contraintes permet de contrôler de façon précise la forme de chaque contribution. Pour la solution proposée ici, un intervalle ΔT de 50 heures a été défini pour chaque projection. Les contributions de chacune des machines évoluent alors en respectant des paliers dont la longueur est fonction du paramètre ΔT et de la forme des $f\max_j$. La méthode des projections successives présente de plus l'avantage de ne nécessiter que très peu de temps pour fournir une solution. On peut en effet voir sur la figure 6.10 que les temps de résolution associés à chaque charge sont très proches de 0.

On peut ensuite voir sur la figure 6.8 que la méthode basée sur l'algorithme du Mirror Prox donne des solutions de formes totalement différentes. Les contributions de chaque machine évoluent à chaque instant, sans respecter de palier constant. L'utilisation plus fluide des machines qui en découle permet de rendre le décalage des démarrages des machines plus performant et d'allonger les horizons de production. De même que pour la première méthode de résolution, l'approche basée sur l'algorithme du Mirror Prox permet de respecter une continuité d'utilisation stricte des machines. Les solutions n'atteignent par contre pas exactement la demande σ. Ce phénomène étant constant sur tout l'horizon de décision, cela peut toutefois être facilement rattrapé en modifiant la valeur de la demande en entrée de la méthode. Pour des horizons de production de même ordre, cette seconde méthode basée sur l'algorithme du Mirror Prox est

moins rapide que la méthode utilisant l'algorithme du Lasso lorsqu'un nombre limité de machines est considéré ($m \leqslant 25$, voir figure 6.10). Elle est par contre plus rapide pour des problèmes de plus grande taille, considérant plus de machines et/ou des horizons de décision plus longs.

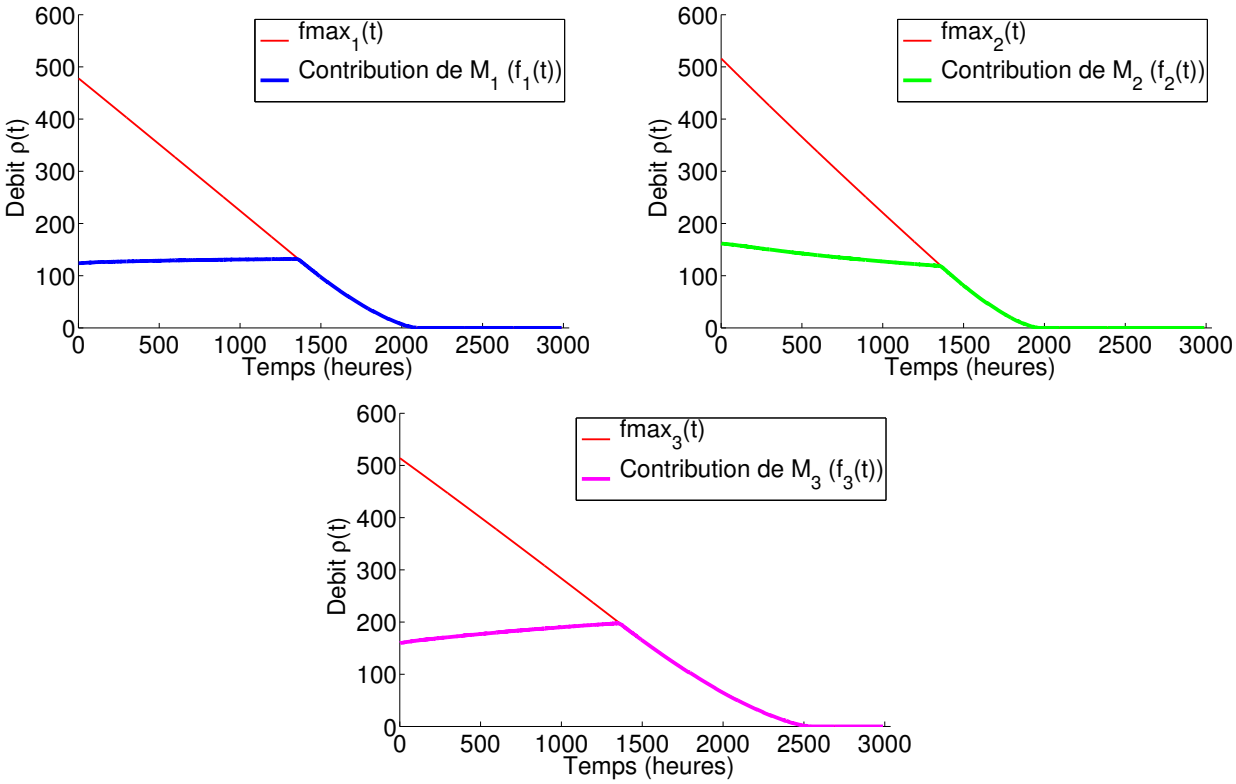


(a) Contribution de chaque machine

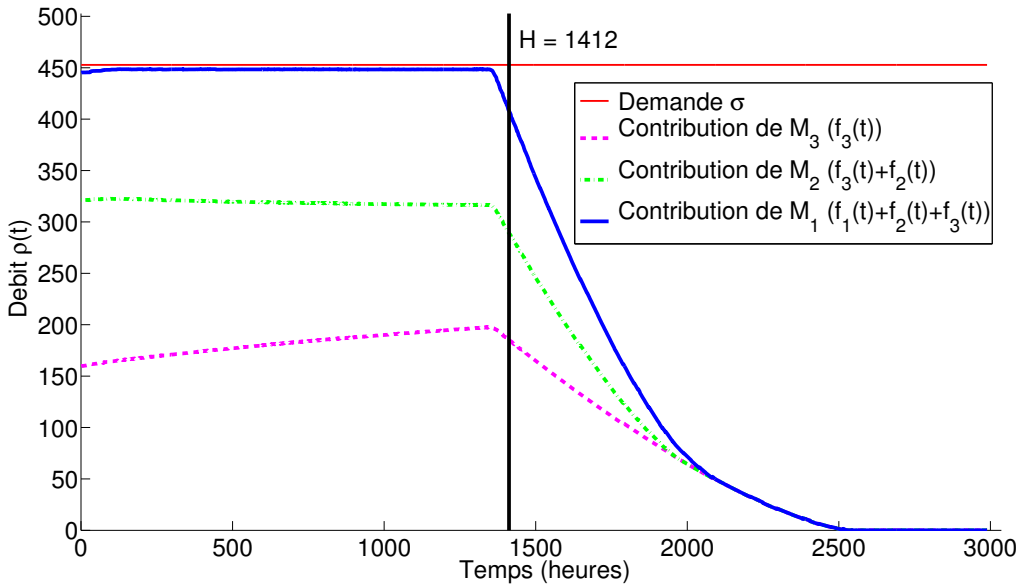


(b) Ordonnancement

Figure 6.7 – Solution obtenue avec la méthode des projections successives (stratégie = *"ΔTsuff_proche"*) – $m = 3$ machines, $\sigma = 0.4 \cdot \rho\text{max}_{tot}$

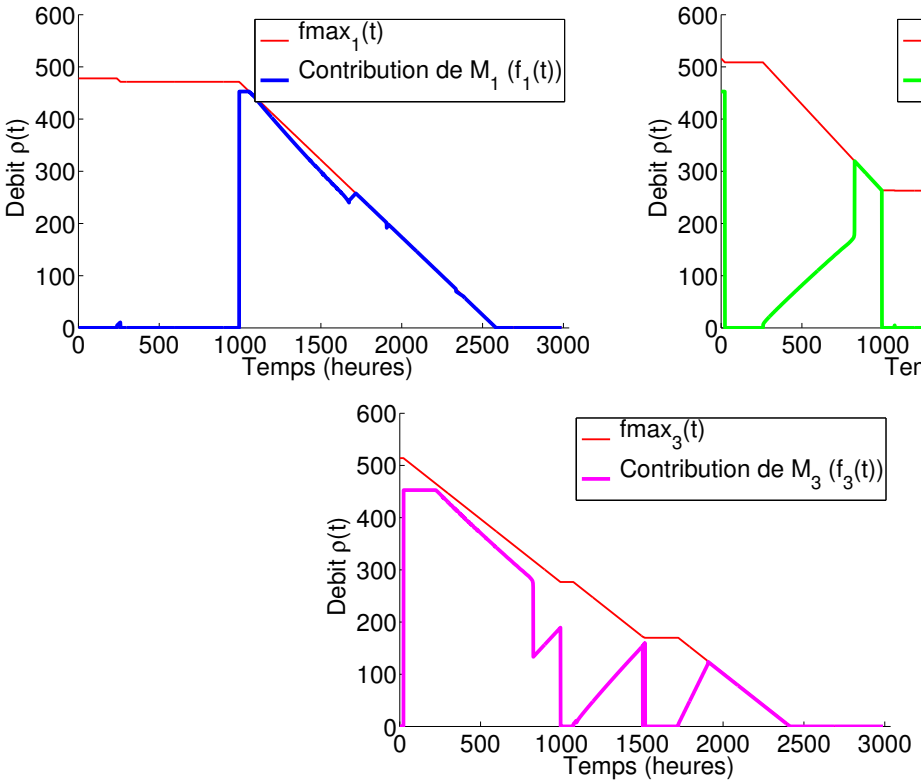


(a) Contribution de chaque machine

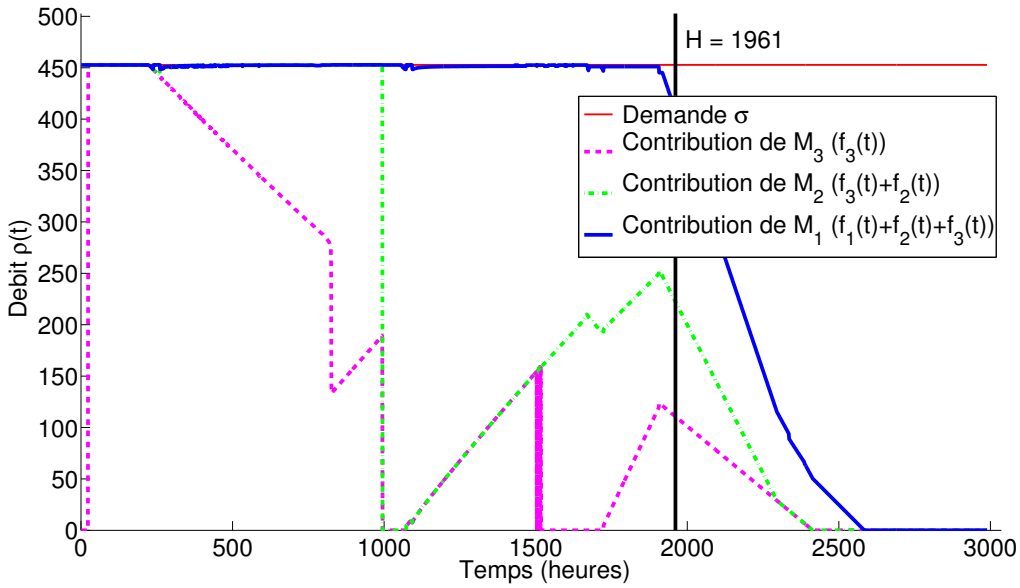


(b) Ordonnancement

Figure 6.8 – Solution obtenue avec l'algorithme du Mirror Prox – $m = 3$ machines, $\sigma = 0.4 \cdot \rho\max_{tot}$



(a) Contribution de chaque machine



(b) Ordonnancement

Figure 6.9 – Solution obtenue avec la méthode du Lasso – $m = 3$ machines, $\sigma = 0.4 \cdot \rho\max_{tot}$

La méthode utilisant la technique du Lasso, pour laquelle un exemple de solution est donné sur la figure 6.9, permet de minimiser à chaque instant le nombre de machines utilisées en parallèle pour atteindre la demande. Cela permet d'allonger les horizons de production atteints, mais présente l'inconvénient de fournir des solutions qui ne respectent pas une utilisation continue des machines une fois que celles-ci ont été démarrées. Les contributions maximales atteignables étant décroissantes, les solutions obtenues ne peuvent pas être modifiées de façon triviale pour rassembler les périodes de non-utilisation de chaque machine en début ou en fin d'ordonnancement. Cet inconvénient de forme mis à part, cette dernière méthode basée sur l'algorithme du Lasso fournit les meilleurs résultats en termes d'horizons de production atteints, et ce, en un temps limité. On peut en effet voir sur la figure 6.10 qu'elle est plus rapide que la méthode basée sur l'algorithme du Mirror Prox.

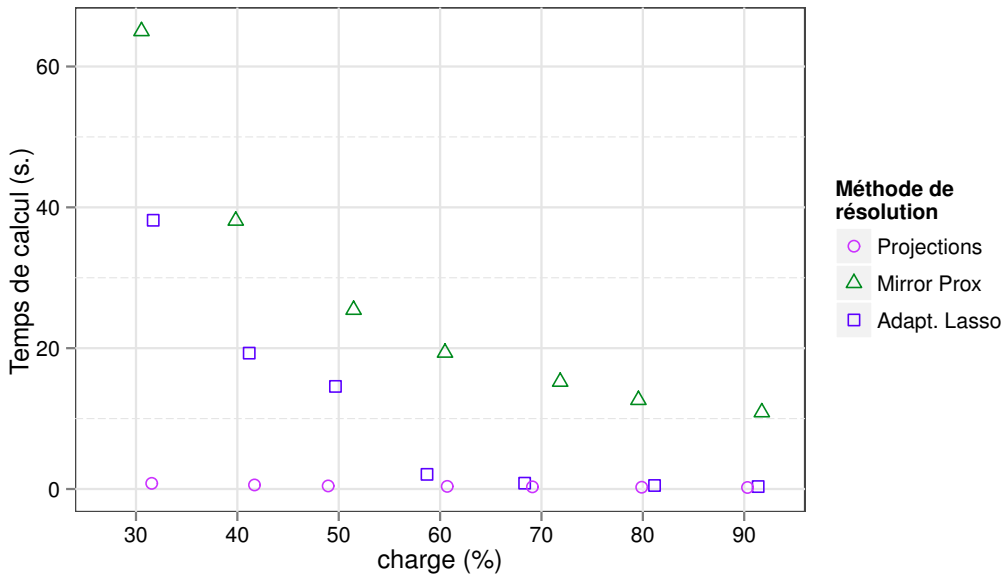


Figure 6.10 – Temps de résolution [6] pour les différentes méthodes convexes – m = 25 machines

6.3.4 Synthèse des résultats

Chaque méthode de résolution proposée traite le respect des contraintes du problème de façon différente. Avec la première méthode, le respect de toutes les contraintes est assuré uniquement par des projections. Avec la seconde méthode, basée sur la résolution d'un Mirror Prox, le respect de l'atteinte de la demande et de l'évolution des contributions maximales atteignables, $f\max_j$, est assuré par l'optimisation de la fonction objectif. Des projections supplémentaires sur les contraintes permettent d'accélérer la convergence de la solution. Dans la troisième méthode proposée, le principe du Lasso est utilisé pour optimiser à la fois la distance de la contribution totale de la plate-forme à la demande et le nombre de machines utilisées à chaque instant en parallèle. La mise à jour des contributions maximales en fonction des contributions des machines est faite par le biais d'une projection. Dans toutes les méthodes de résolution proposées dans ce chapitre, des projections sur les contraintes sont donc utilisées pour assurer le respect de toutes

6. Les simulations ont été effectuées sur Matlab (Paramètres : Processeur Intel® Core™ i5-3550 CPU @ 3.30GHz×4, 15.6 Gio, 64 bits)

les contraintes du problème. Ces projections altèrent d'une certaine façon le côté optimal des résolutions, mais permettent d'améliorer la vitesse de convergence. On retrouve aussi dans toutes les méthodes un certain contrôle du caractère creux des solutions. Seul le nombre strictement nécessaire de machines est en effet généralement utilisé dans chaque solution et on peut voir apparaître des démarrages décalés des machines dans le temps. Cela est principalement dû à la formulation de la projection sur la demande, qui respecte un choix des machines à utiliser à chaque instant qui est cohérent avec la formation de solutions creuses. La méthode basée sur l'algorithme du Lasso, qui intègre par définition une norme ℓ_1 pour répondre à ce genre de problématique, reste la plus performante pour ce qui est de la minimisation du nombre de machines utilisées en parallèle.

Les solutions fournies par la méthode des projections ne sont pas efficaces en termes d'horizon de production atteints. Elles sont toutefois obtenues très rapidement et pourraient constituer un point de départ pour la recherche de solutions. Cela permettrait éventuellement de diminuer le temps de convergence des deux autres méthodes proposées. Ces méthodes sont toutes deux efficaces pour optimiser l'horizon de production, mais ne construisent pas les solutions de la même manière, ce qui donne à ces dernières des allures très différentes. Une préférence pour l'une ou l'autre méthode peut ainsi être définie en fonction de l'application considérée et de l'importance donnée à chacune des contraintes initiales prises en compte pour la définition du modèle.

Tous les résultats présentés dans ce chapitre ont été générés en considérant une demande constante tout au long de l'horizon de décision. La formulation mathématique du problème d'optimisation proposée dans la partie 6.1 permet toutefois d'appliquer les différentes méthodes de résolution à une demande variable dans le temps, quelque soit sa forme.

6.4 Synthèse

Les différentes approches proposées dans ce chapitre sont basées sur des méthodes d'optimisation convexe. L'utilisation de ce type d'optimisation mathématique, dans lequel une fonction convexe est à minimiser, apporte de bonnes propriétés de convergence à la résolution. Une expression convexe du problème permet en effet de restreindre l'espace des solutions au cours de l'optimisation.

La formulation convexe du problème, nécessaire pour le type de résolution utilisé, a amené à simplifier le modèle de comportement des machines pris en compte (voir figure 6.1). Toutes les caractéristiques traduites dans le modèle initial présenté dans le chapitre 3 (partie 3.3.1) ne sont ainsi pas respectées par le modèle mathématique détaillé dans la partie 6.1. Cette première étude permet toutefois de montrer que le problème d'optimisation considéré peut être traité de façon globale sur tout l'horizon de production.

Les temps de résolution des méthodes proposées dans ce chapitre sont enfin plus longs que ceux associés à la méthode optimale par morceaux proposée dans le chapitre 5, mais restent tout à fait acceptables pour le niveau de décision à moyen terme considéré. Toutes les solutions sont en effet obtenues en moyenne en moins de quelques minutes, alors que les fréquences de décision associées au problème d'optimisation traité se comptent en heures, voire en semaines. Les méthodes convexes permettent de plus de proposer des solutions à des problèmes de plus

grande taille. En effet, alors que l'utilisation de la programmation linéaire limite la taille des problèmes d'optimisation pouvant être traités à environ 2000 variables, ce qui correspond à 500 machines, la résolution basée sur la programmation convexe permet de résoudre des problèmes mettant en jeu plus de 1000 machines en temps raisonnable. Pour une demande correspondant à une charge χ de 50% pour laquelle l'horizon de production est d'environ deux fois le RUL moyen des m machines considérées (1500 heures), cela correspond à environ $6000 \cdot m$ variables.

Conclusion et perspectives

Synthèse générale

Les travaux de recherche proposés dans ce manuscrit s'intéressent à l'ordonnancement de plates-formes constituées de machines hétérogènes et distribuées, utilisées en parallèle pour fournir un service commun. Pour des raisons de minimisation des coûts, toutes les opérations de maintenance sont regroupées à la fin de l'ordonnancement pour toutes les machines. Il s'agit alors d'optimiser l'utilisation du potentiel fourni par chaque machine pour rentabiliser au mieux la mobilisation des moyens de maintenance. L'objectif pris en compte est alors la maximisation de la durée d'utilisation de cette plate-forme avant maintenance. Une certaine demande devant être satisfaite, le problème revient à maximiser l'horizon de production de la plate-forme.

L'originalité de l'approche consiste à définir une utilisation des machines suivant des conditions opératoires variables et à lier chaque niveau de performance pouvant être fourni à une durée d'utilisation disponible différente. Chaque machine est ainsi supposée pouvoir fournir un certain nombre de niveaux de performance différents. Cette étude est proposée dans un contexte *PHM*, dans lequel il est possible de connaître l'état de santé des machines et d'en déduire des durées résiduelles d'utilisation avant défaillance. Il s'agit alors de prendre des décisions éclairées s'appuyant sur les résultats du pronostic. Ce problème s'inscrit dans la décision post-pronostic, au sein de laquelle les contributions que l'on peut trouver dans la littérature s'intéressent jusqu'à présent en grande majorité à l'optimisation de la maintenance.

Nous avons montré que considérer la possibilité de modifier les conditions opératoires des machines au cours de leur utilisation peut permettre d'allonger la durée d'utilisation avant maintenance d'une plate-forme. Cela est généralement associé à une dégradation des performances des machines, qui se traduit par une diminution du débit fourni. Le but n'est toutefois pas d'optimiser l'utilisation de chaque machine prise individuellement, mais celle de l'ensemble des machines considérées.

Ces travaux de recherche ont d'abord été positionnés dans la littérature traitant du *PHM* et de l'ordonnancement de la production. Les différentes étapes du processus *PHM* ont été décrites avant une présentation plus précise de la décision post-pronostic. L'ordonnancement de la production a ensuite été introduit par la présentation des problèmes types traités dans la littérature et les principales méthodes de résolution utilisées. Nous avons montré que la notion de reconfiguration, utilisée dans les travaux de recherche présentés ici pour adapter l'utilisation des machines à leur état de santé, se retrouve à la fois dans le domaine du *PHM* et dans celui de l'ordonnancement.

Une fois les travaux de recherche positionnés dans la littérature, le contexte applicatif des travaux et le problème d'optimisation traité ont été définis par le biais de la présentation des différentes hypothèses prises en compte. Deux modélisations différentes ont ensuite été proposées pour définir des profils de fonctionnement pour chaque machine, définissant les débits disponibles à chaque instant et les durées d'utilisation résiduelle avant maintenance associées à chacun d'eux. Le premier modèle s'applique à un premier type de machines pouvant fournir un nombre discret de niveaux de performance. Le second modèle développé permet ensuite de modéliser le comportement à l'usure de machines pour lesquelles les performances évoluent de façon continue entre deux bornes.

Des premiers résultats ont été proposés pour le problème d'ordonnancement considérant des machines pouvant fournir un nombre discret de performances. Une borne supérieure pour l'horizon de production atteignable a tout d'abord été définie. Une étude de la complexité de plusieurs variantes du problème d'optimisation a ensuite permis de montrer qu'une solution optimale peut être trouvée en temps polynomial dans un cas bien particulier, mais que le problème est NP-complet au sens fort dans le cas général. Une formulation optimale du problème a été proposée sous la forme d'un programme linéaire en nombres entiers, permettant de résoudre des problèmes présentant un nombre de variables limité. Plusieurs heuristiques ont finalement été développées pour traiter des problèmes de tailles plus réalistes. Nous avons montré que les meilleures heuristiques permettent d'obtenir des solutions avec des horizons de production très proches de l'optimal.

Deux approches différentes ont ensuite été proposées pour ordonnancer des machines dont les performances varient de façon continue dans un certain intervalle. La première approche construit les ordonnancements par morceaux en utilisant des résolutions optimales successives de sous-problèmes. Une approche optimale basée sur une programmation linéaire mixte permet de déterminer des configurations optimales des machines lorsque leur utilisation est limitée à un débit fixe. Un processus heuristique a été défini pour gérer le changement de profil de fonctionnement au cours de l'utilisation des machines et construire des solutions utilisant tout le potentiel des plates-formes. La durée de vie globale de chaque plate-forme considérée correspond alors à la somme des horizons atteints par chaque solution des programmes linéaires successifs. Les solutions fournies par chaque programme linéaire sont optimales considérant l'état de santé des machines au début de chaque recherche de solution. Les ordonnancements obtenus avec une succession de résolutions optimales ne sont toutefois pas nécessairement optimaux.

Les différentes approches de résolution présentées jusqu'à ce stade ont permis de trouver des solutions optimales en temps polynomial uniquement pour des problèmes pour lesquels le nombre de machines et/ou l'horizon de production restent limités. L'utilisation de méthodes de résolution sous-optimales est nécessaire pour traiter des problèmes de grande taille. Bien que les résultats obtenus avec les méthodes proposées soient, dans la plupart des cas, proches de la borne supérieure, une amélioration est encore possible. Le point limitant principal des méthodes sous-optimales proposées est lié à une construction des solutions par morceaux. Les solutions sont en effet construites de façon itérative sur l'horizon de production et la définition à chaque instant d'une configuration optimale (considérant l'état courant du système) ne suffit pas à garantir l'optimalité de la solution globale. Une autre approche utilisant un paradigme totalement différent a alors été proposée pour traiter des problèmes d'optimisation de grande taille en effectuant la recherche de solutions de façon globale sur tout l'horizon de production. La

contribution de chaque machine tout au long de sa durée de vie a dans ce cas été considérée dans son ensemble et optimisée globalement sur tout l'horizon. Chaque contribution a été déterminée par le biais d'une optimisation convexe. L'utilisation de ce type d'optimisation mathématique, dans lequel une fonction convexe est à minimiser, apporte certaines propriétés au problème traité, notamment de bonnes propriétés de convergence. La programmation convexe permet de plus de résoudre en temps polynomial des problèmes de grande taille, mettant en jeux un grand nombre de machines sur de grands horizons de production. Une formulation mathématique du problème d'optimisation a été introduite pour permettre sa résolution avec des méthodes de programmation convexe. Différentes méthodes de résolution ont été proposées. La première méthode, basée sur des projections successives sur les contraintes, est très rapide, mais a des performances limitées. La seconde utilise le principe de la méthode Mirror Descent, développée pour minimiser une fonction convexe sujette à des contraintes convexes. L'utilisation de sa variante Mirror Prox a été proposée pour optimiser une fonction objectif visant à satisfaire les différentes contraintes du problème. Une troisième méthode de résolution convexe, basée sur l'algorithme du Lasso (Least Absolute Shrinkage and Selection Operator), qui associe la méthode des moindres carrés à une pénalisation de type ℓ_1, a également été étudiée. La première partie a été utilisée pour déterminer la contribution de chaque machine à la production totale au cours du temps, tandis que la seconde a permis de contrôler la sparsité (ou caractère creux) de la solution. Ces trois approches convexes ont à nouveau été évaluées sur la base de résultats de simulation.

Un large panel de méthodes de résolution différentes a ainsi été utilisé pour traiter le problème d'optimisation posé dans cette thèse. Des approches exactes ont tout d'abord été proposées pour la construction de solutions optimales dans les cas où la recherche de solution n'est pas limitée par la taille du problème. Un programme linéaire en nombres entiers a ainsi été proposé pour trouver une configuration optimale des machines lorsque ces dernières peuvent être utilisées avec un nombre discret de profils de fonctionnement différents. La programmation dynamique a ensuite été utilisée au sein d'une heuristique pour définir une utilisation optimale par morceaux des machines. Le même principe de concaténation de sous-solutions optimales a été utilisé pour l'ordonnancement de machines présentant des performances variant de façon continue entre deux bornes. Dans ce cas, les sous-solutions optimales sont obtenues avec un programme linéaire mixte. Les solutions obtenues avec ces deux dernières méthodes ne sont pas nécessairement optimales. Chaque sous-solution composant la solution globale est en effet optimale considérant l'état de santé courant des machines, mais les décisions prises à un instant donné sont dépendantes des décisions prises aux instants précédents.

Différentes méthodes heuristiques ont ensuite été utilisées dans les derniers chapitres pour l'élaboration des ordonnancements. Plusieurs heuristiques constructives suivant différentes règles de sélection des machines et des profils de fonctionnement ont été développées pour le cas des machines avec variation discrète des performances. Une heuristique d'amélioration a été proposée pour modifier les ordonnancements obtenus avec ces premières heuristiques constructives dans le but d'allonger leurs horizons de production. Pour le cas des machines dont les performances évoluent de façon continue, un premier processus heuristique a permis la mise à jour des caractéristiques des machines entre deux résolutions optimales lorsque la résolution du problème est faite par morceaux. Plusieurs heuristiques basées sur des méthodes de résolution convexe ont finalement été développées pour une résolution du problème considérant l'horizon de production

comme un tout.

On voit apparaître au fil des chapitres une évolution dans la manière de construire les solutions d'ordonnancement. Dans le cas des machines à variation discrète des performances, les ordonnancements sont tout d'abord élaborés de façon discrète, c'est-à-dire période par période. Cette discrétisation évolue dans les deux contributions proposées pour les machines dont les performance peuvent être modifiées de manière continue. La première résolution suit encore une certaine discrétisation du temps dans la construction des ordonnancements. Ces derniers sont toutefois découpés en différentes phases correspondant chacune à une certaine configuration de la plate-forme et de longueurs variables. Le dernier type de résolution proposé suit finalement une vision globale de la dimension temporelle. L'horizon de production est en effet considéré dans son ensemble et les contributions des machines sont définies en une fois sur toute la durée de l'ordonnancement.

Une approche originale d'ordonnancement de la production d'une plate-forme, qui base les décisions sur l'état de santé réel des machines, a donc été proposée. Dans les différents cas traités dans ce manuscrit, nous avons montré que la considération de plusieurs profils de fonctionnement a un intérêt lorsqu'une dégradation des performances de certaines machines permet d'allonger leur durée de fonctionnement avant maintenance et lorsque l'objectif est de maximiser l'horizon de production d'un ensemble de machines parallèles indépendantes. Ces travaux, qui s'inscrivent dans la partie décisionnelle du processus *PHM*, permettent de déterminer la durée maximale d'utilisation avant défaillance d'un système à partir des durées d'utilisation résiduelles (RUL) des équipements qui le composent. Il s'agit en quelque sorte de définir un RUL global du système à partir du RUL de ses composants. Différentes perspectives, détaillées dans la suite, peuvent être considérées.

Perspectives de recherche

Différents axes de perspectives peuvent être envisagés à la suite de ces travaux. Des perspectives directes englobent tout d'abord des compléments permettant d'améliorer les résultats obtenus et de compléter l'étude théorique. Des perspectives plus générales s'inscrivant dans un contexte plus large sont ensuite proposées.

Seules la manière d'utiliser les méthodes d'optimisation convexe pour la résolution du problème considéré et la comparaison de l'efficacité de chacune d'entre elles ont été détaillées dans le chapitre 6 basant l'ordonnancement de machines avec variation continue des performances sur la programmation convexe. Une des principales perspectives de ces travaux comprend l'étude théorique de la convergence de chacune des trois méthodes de résolution développées. Plusieurs résultats ont été proposés dans la littérature quant à la vitesse de convergence des différentes méthodes utilisées, en fonction des caractéristiques des fonctions mises en jeu. Une étude plus poussée de l'évolution des solutions en fonction du nombre d'itérations pourrait de plus permettre d'améliorer les temps de résolution des méthodes utilisant les techniques du Mirror Prox et du Lasso en définissant des critères d'arrêt plus performants.

La formulation convexe du problème, nécessaire pour le type de résolution utilisé, a amené à simplifier le modèle de comportement des machines pris en compte. Toutes les caractéristiques

traduites dans le modèle initial ne sont pas respectées par le modèle mathématique pris en compte dans le chapitre 6. Une seconde perspective consiste alors à améliorer la formulation convexe des propriétés du modèle afin de respecter au mieux les caractéristiques réelles des machines considérées. Une adaptation de la forme des contributions maximales disponibles, $f\max_j$, permettrait par exemple de modéliser l'accélération de la dégradation pour les débits inférieurs au débit optimal ($\rho_j < \rho\text{opt}_j$), associé dans le modèle initial à la durée de vie la plus longue. L'ajout d'une minimisation de la norme ℓ_1 des différences successives des composantes du vecteur solution F permettrait de plus de pallier le problème de non-continuité d'utilisation des machines obtenu avec la méthode basée sur l'algorithme du Lasso. La considération d'une telle norme permettrait en effet de contrôler la vitesse de variation des contributions de chaque machine et donc de faire tendre les solutions obtenues vers des profils plus lisses. Cet ajout nécessiterait une nouvelle détermination des valeurs des paramètres de la méthode du Lasso en fonction de l'importance donnée aux deux caractéristiques recherchées, à savoir la continuité d'utilisation et le côté lisse des contributions, qui peuvent être contradictoires avec l'objectif de maximisation de l'horizon considéré.

Nous avons montré que les différentes stratégies d'ordonnancement proposées, développées pour des demandes restant constantes tout au long de l'horizon de production, peuvent être utilisées pour construire des ordonnancements satisfaisant des demandes variables, constantes par morceaux. Les résultats obtenus pour des profils de demande avec une variation limitée sont encourageants. Il semble toutefois plus raisonnable de définir de nouvelles stratégies d'ordonnancement pour les cas où la demande est fortement variable. Un problème d'optimisation totalement différent pourrait en effet être défini pour prendre en compte les possibilités d'amélioration offertes par l'ajout d'un stockage de la production. Une gestion spécifique du stock et de son utilisation en parallèle de la production des machines au cours de l'horizon de production serait alors à définir. La prise en compte du stockage offrirait de plus la possibilité de gérer des incertitudes sur le profil de demande.

Suivant la même idée, les valeurs de RUL prises en compte dans le processus de décision peuvent être considérées comme étant entachées d'une certaine erreur. Les méthodes de pronostic sont de plus en plus performantes et permettent dans beaucoup de cas d'application d'atteindre des précisions de prévision relativement élevées. Des erreurs de prévision sont toutefois toujours possibles. Iyer et al. [52] ont proposé d'associer à la prévision des valeurs de RUL la détermination de deux seuils encadrant le RUL brut. Ces seuils peuvent prendre la forme d'une borne inférieure et d'une borne supérieure. Deux cas peuvent alors se présenter. Les prévisions peuvent être pessimistes et définir des RUL plus courts que les durées réelles d'utilisation résiduelles des machines. Cela n'affecte que très peu, voire pas du tout les ordonnancements obtenus avec les méthodes proposées dans ce manuscrit. Le seul risque est d'avoir d'avantage de potentiel restant qu'espéré à la fin des ordonnancements. En revanche, des prévisions trop optimistes des valeurs de RUL entraînent un risque de pannes non prévues et donc une éventuelle indisponibilité des machines. Cela peut remettre en question les ordonnancements définis et les horizons de production associés.

Une première solution envisageable pour diminuer le risque de pannes est d'effectuer des mises à jour des valeurs de RUL au cours de l'utilisation des machines. La récupération de nouvelles données provenant de la surveillance des machines et traitées par l'étape de pronostic

entraîne alors la redéfinition du potentiel restant pour chaque machine. Cela permet d'adapter l'ordonnancement défini préalablement au nouvel état de santé connu de la plate-forme. Ce fonctionnement s'apparente à un ordonnancement en ligne. Les différentes méthodes de résolution proposées dans ce manuscrit ont été développées pour fonctionner hors ligne, mais les temps de résolution associés à chacune d'entre elles sont assez courts pour permettre leur utilisation pour des calculs en ligne dès lors que la fréquence de décision reste raisonnable (de l'ordre de la minute). La plupart des heuristiques proposées fonctionnant par période, un changement des données en cours d'ordonnancement n'aurait que très peu d'impact sur la construction des solutions. Pour ces heuristiques, les décisions prises à un instant ne présagent en effet en rien des choix futurs d'ordonnancement. Des ajustements relativement simples pourraient être faits pour les heuristiques fonctionnant par groupe de périodes. La robustesse face au changement des valeurs initiales de RUL des méthodes de résolution basées sur la programmation convexe est cependant plus faible. Ces méthodes définissent en effet l'utilisation des machines en considérant tout l'horizon de production et l'évolution des solutions est limitée à chaque itération à un voisinage proche. Un changement des données en cours de recherche de solution aurait alors des conséquences potentiellement importantes sur le temps de résolution des méthodes ou sur la qualité des solutions obtenues.

La validation des différentes méthodes de résolution proposées dans ce manuscrit a enfin été faite par des simulations basées sur des données fictives, générées de manière à représenter au mieux la réalité. Cela a été nécessaire compte-tenu de l'absence de données réelles de pronostic prenant en compte des variations de profils au cours de l'utilisation des machines. Nous avons en effet montré dans le chapitre 1 que les différentes définitions et applications du pronostic que l'on peut trouver dans la littérature ne permettent pas de fournir des valeurs de RUL pour des conditions opératoires variables. Les travaux récents de Nguyen et al. [73] proposant une méthode de prédiction basée sur une modélisation d'un indicateur de santé variant avec les conditions opératoires sont toutefois prometteurs pour l'évolution du pronostic.

D'autres travaux récents participent à cette évolution pour le cas d'applications particulier des piles à combustible (PEMFC). Jouin et al. [55, 56] ont proposé un modèle permettant de prédire le vieillissement de piles à combustible à membrane d'échange de protons. Ce modèle est utilisé pour prédire le comportement futur des piles et pour déterminer leurs valeur de RUL pour différents profils de mission. Des profils constants ont d'abord été utilisés [56], puis la méthode a été validée sur des profils de mission variables dans le temps [55]. Ces résultats pourraient être utilisés pour valider les ordonnancements construits avec les différentes méthodes de résolution proposées dans ce manuscrit. Il s'agirait par exemple de définir un ordonnancement des machines sur un certain horizon, puis de vérifier avec l'outil de pronostic développé par Jouin et al [55] si l'utilisation des machines définie par le module de décision que nous avons développé est viable compte-tenu de l'état de santé initial des machines et de l'évolution prévue des différents indicateurs de santé.

De manière générale, au vu des dernières perspectives proposées ici, il serait intéressant de développer un processus itératif au sein du PHM définissant un échange d'informations entre l'étape de pronostic et celle de décision. Les informations apportées par la décision permettraient de guider le pronostic pour la génération des données utiles. Ces données seraient ensuite à nouveau récupérées par le module de décision et utilisées pour améliorer les stratégies définies.

Liste des publications

L'ensemble des travaux et résultats présentés dans cette thèse a fait l'objet de publications dans des conférences internationales avec comité de lecture. Des versions étendues de certains articles ont proposées dans des rapports de recherche.

Des résultats préliminaires ont de plus été présentés lors de workshops sur le thème de l'ordonnancement. Le premier, intitulé « New Challenges in Scheduling Theory », s'est tenu à Aussois en Avril 2014. Le second, « Scheduling for Large Scale Systems », a eu lieu à Lyon en Juillet 2014.

Articles parus en conférences internationales avec comité de lecture

[PHM2015] Stéphane Chrétien, Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *A Post-Prognostics Decision Approach to Optimize the Commitment of Fuel Cell Systems in Stationary Applications*, Proceedings of IEEE International Conference on Prognostics and Health management (PHM), Austin, TX, June 22-25, 2015.

[FDFC2015] Nathalie Herr, Jean-Marc Nicod, Christophe Varnier, Louise Jardin, Antonella Sorrentino, Rafaël Gouriveau, Daniel Hissel, Marie-Cécile Péra. *Decision Process to Manage Useful Life of Multi-Stacks Fuel Cell Systems uner Service Constraint*, Proceedings of 6th International Conference on Fundamentals & Development of Fuel Cells (FDFC), Toulouse, France, Februar 3-5, 2015.

[PGMO2014] Stéphane Chrétien, Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *Scheduling independent parallel machines with convex programming*, Proceedings of Gaspard Monge Program for Optimization - Conference on Optimization & Practices in Industry (PGMO-COPI'14), Paris, France, October 28-31, 2014.

[CASE2014] Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *Prognostics-based Scheduling in a Distributed Platform : Model, Complexity and Resolution*, Proceedings of IEEE International Conference on Automation Science and Engeneering (CASE), Taipei, Taiwan, August 18-22, 2014.

[PHM2014] Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *Prognostic Decision Making to Extend a Platform Useful Life under Service Constraint*, Proceedings of IEEE International Conference on Prognostics and Health management (PHM), Spokane, Washington, June 22-25, 2014. – *Best Paper Award* –

Rapports de recherche

[RR2015] Stéphane Chrétien, Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *Scheduling independent parallel machines with convex programming*, Research Report number RR-FEMTO-ST-8264, FEMTO-ST Institute, France, 2015.

[RR2014] Nathalie Herr, Jean-Marc Nicod and Christophe Varnier. *Prognostics-based Scheduling to Extend a Distributed Platform Production Horizon under Service Constraint : Model, Complexity and Resolution*, Research Report number RR-FEMTO-ST-2577, hal-01005443, FEMTO-ST Institute, France, 2014.

Bibliographie

[1] S. A. Asmai, B. Hussin, and M. M. Yusof. A framework of an intelligent maintenance prognosis tool. In *Proceedings of the 2nd International Conference on Computer Research and Development (ICCRD), Huala Lumpur, Malaysia*, number 5489525, pages 241 – 245, May 2010. 9, 14, 15, 16

[2] E. Balaban and J. J. Alonso. An approach to prognostic decision making in the aerospace domain. In *Proceedings of The Annual Conference of the Prognostics and Health Management Society, Minneapolis, Minnesota*, pages 396–415, September 2012. 12, 13, 14, 15, 16, 19

[3] E. Balaban, S. Narasimhan, M. Daigle, J. Celaya, I. Roychoudhury, B. Saha, S. Saha, and K. Goebel. A mobile robot testbed for prognostic-enabled autonomous decision making. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Montreal, Quebec, Canada*, September 2011. 15, 17

[4] E. Balaban, S. Narasimhan, M. J. Daigle, I. Roychoudhury, A. Sweet, C. Bond, J. R. Celaya, and G. Gorospe. Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *International Journal of Pronostics and Health Management*, 4 (1), 2013. 17

[5] A. Barros, C. Berenguer, and A. Grall. Optimization of replacement times using imperfect monitoring information. *IEEE Transactions on Reliability*, 52(4) :523–533, December 2003. 13, 14

[6] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *Society for Industrial and Applied Mathematics (SIAM) Review*, 38 (3) :367–426, 1996. 113

[7] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31 :267–175, 2003. 117

[8] A. Bellanger. *Ordonnancement sur les machines à traitement par batches et contraintes de compatibilité*. Thèse en informatique, Institut National Polytechnique de Lorraine - INPL, 2009. 22

[9] R. Bellman. The threory of dynamic programming. Technical Report No. RAND-P-550, Rand Corp. Santa Monica, CA, 1954. 25

[10] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi. Extending lifetime of portable systems by battery scheduling. In *Proceedings of the Conference on Design, Automation and Test in Europe, Munich, Germany*, pages 191–203, March 13-16 2001. 13, 15, 18

[11] L. Benini, G. Castelli, A. Macii, and R. Scarsi. Battery-driven dynamic power management. *IEEE Design & Test of Computers*, (2) :53–60, 2001. 18

[12] S. M. Bennett, R. J. P. J. Patton, and S. Daley. Using bilinear motor model for a sensor falut tolerant rail traction drive. In *Proceedings of IFAC SAFEPROCESS'97, Hull, UK*, 1997. 29

[13] F. Besnard and L. Bertling. An approach for condition-based maintenance optimization applied to wind turbine blades. *IEEE Transactions on Sustainable Energy*, 1(2) :77–83, July 2010. 13, 14, 15

[14] A. Billionnet. *Optimisation discrète - De la modélisation à la résolution par des logiciels de programmation mathématique*. Applications & Métiers. Dunod, 2007. 95

[15] M. Blanke, R. Izadi-Zamanabadi, S. A. Bogh, and C. P. Lunau. Fault-tolerant control systems - a holistic view. *Control Engineering Practice*, 5(5) :693–702, 1997. 29

[16] J. Blazewicz, J. Lenstra, and A. H. G. R. Kan. Scheduling subject to resource constraints : classification and complexity. *Discrete Applied Mathematics*, 5 (1) :11–24, 1983. 23

[17] A. Bogdanov, S. Chiu, L. U. Gökdere, and J. Vian. Stochastic optimal control of a servo motor with a lifetime constraint. In *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pages 4182–4187, December 13-15 2006. 14

[18] B. Bole, L. Tang, K. Goebel, and G. Vachtsevanos. Adaptive load-allocation for prognosis-based risk management. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Montreal QC, Canada*, pages 1–10, 2011. 14

[19] P. P. Bonissone and N. Iyer. Soft computing applications to prognostics and health management (PHM) : Leveraging field data and domain knowledge. In *Proceedings of 9th International Work-Conference on Artificial Neural Networks, (IWANN), San Sebastian, Spain*, pages 928–939, June 2007. 6, 9, 10, 13

[20] D. W. Brown, G. Georgoulas, B. Bole, H.-L. Pei, M. Orchard, L. Tang, B. Saha, A. Saxena, and K. G. G. Vachtsevanos. Prognostics enhanced reconfigurable control of electro-mechanical actuators. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, San Diego, CA*, pages 1–17, September 27 - October 1 2009. 14

[21] D. W. Brown and J. Vachtsevanos. A prognostic health management based framework for fault-tolerant control. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Montreal, Canada*, 2011. 14

[22] S. Bubeck. *Theory of Convex Optimization for Machine Learning*. arXiv preprint arXiv :1405.4980, 2014. 117

[23] G. Budai, R. Dekker, and R. Nicolai. A review of planning models for maintenance and production. Technical report, Econometric Institute, Erasmus University Rotterdam, 2006. 6

[24] F. Camci and R. B. Chinnam. Health-state estimation and prognostics in machining processes. *IEEE Transactions on Automation Science and Engineering*, 7 (3) :581 – 597, July 2010. 8, 10, 13

[25] F. Camci, G. S. Valentine, and K. Navarra. Methodologies for integration of PHM systems with maintenance data. In *Proceedings of the IEEE Aerospace Conference, Big Sky, Montana*, pages 4110–4118, March 2007. 13, 14

[26] E. J. Candès. Mathematics of sparsity (and a few other things). In *Proceedings of the International Congress of Mathematicians, Seoul, South Korea*, 2014. 123

[27] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6) :877–905, 2008. 123

[28] F. Carazas and G. F. M. de Souza. Risk-based decision making method for maintenance policy selection of thermal power plant equipment. *Energy*, 35 (2) :964–975, 2010. 6

[29] J. Carlier and P. Chrétienne. *Problèmes d'ordonnancement : modélisation, complexité et algorithmes*. Masson, 1988. 22

[30] D. W. Carman, P. S. Kruus, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical report, DARPA Project Report, Cryptographic Technologies Group, Trusted Information System, NAI Labs, 2000. 18

[31] G. Chen, K. Malkowski, M. Kandemir, and P. Raghavan. Reducing power with performance constraints for parallel sparse applications. In *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, Boston, USA*, 2005. 31

[32] D. Cho and M. Parlar. A survey of preventative maintenance model for multi-unit systems. *European Journal of Operational Research*, 51 :1–23, 1991. 14

[33] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Addison-Welsey, Massachussets, 1967. 23

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Algorithmique*. Dunod, 2010. 26

[35] J. A. DeCastro, L. Tang, B. Zhang, and G. Vachtsevanos. Systems verification approach to fault-tolerant aircraft supervisory control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Portlan, Oregon*, 2011. 17

[36] C. Dietl and U. K. Rakowsky. An operating strategy for high-availability multi-station transfer lines. *International Journal of Automation and Computing*, 2 :1125–130, 2006. 30

[37] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of National Academy of Science*, 100 (5) :2197–2202, 2003. 123

[38] D. Edwards, M. Orchard, L. Tang, K. Goebel, and G. Vachtsevanos. Impact of input uncertainty on failure prognostic algorithms : Extextend the remaining useful life on nonlinear systems. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Portland, OR*, 2010. 17

[39] J. Eich and B. Sattler. Fault tolerant control system design using robust control techniques. In *Proceedings of IFAC SAFEPROCESS'97, Hull, UK*, pages 1246–1251, 1997. 29

[40] W. Elghazel, J. M. Bahi, C. Guyeux, M. Hakem, K. Medjaher, and N. Zerhouni. Dependable wireless sensor networks for prognostics and health management : A survey. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, FortWorth, TX*, September 29 - October 02 2014. 18, 19

[41] W. Elghazel, K. Medjaher, N. Zerhouni, J. M. Bahi, A. Farhat, C. Guyeux, and M. Hakem. Random forest for industrial device functionning diagnostics using wireless sensor networks. In *Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA*, pages 1–9, March 07-14 2015. 15, 18, 19

[42] W. Elghazel, N. Zerhouni, J. Bahi, K. Medjaher, M. Hakem, and C. Guyeux. Dependability of sensor networks for prognostics and health management. Technical report, FEMTO-ST, dpt AS2M, 25000 Besançon, FRANCE, 2013. 19, 35

[43] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W.H. Freeman & Co, 1979. 56

[44] K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen. Prognostics in battery health management. *IEEE Instrumentation & Measurement Magazine*, 11(4) :33, 2008. 17

[45] R. Gouriveau, K. Medjaher, E. Ramasso, and N. Zerhouni. PHM - Prognostics and health management - de la surveillance au pronostic de défaillances de systèmes complexes. *Techniques de l'Ingénieur*, 2013. 9

[46] R. Graham, E. Lawler, J. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling : a survey. *Annals of Discrete Mathematics*, 5 :287–326, 1979. 23

[47] G. Haddad, P. Sandborn, and M. Pecht. A real options optimization model to meet availability requirements for offshore wind turbines. In *Proceedings of MFPT : The Applied Systems Health Management Conference, Virginia Beach, Virginia*, 2011. 6, 13, 14, 15, 16

[48] A. Heng, S. Zhang, A. C. Tan, and J. Mathew. Rotating machinery prognostics : State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23 (3) :724 – 739, 2009. 7, 8

[49] S. Holzknecht, E. Biebl, and H. U. Michel. Graceful degradation for driver assistance systems. *Advanced Microsystems for Automotive Applications, Spinger Berlin Heidelberg*, pages 255–265, 2009. 28, 30

[50] A. V. Horenbeek and L. Pintelon. A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, 120 :39–50, 2013. 9

[51] Z. Imam, B. Conrard, and M. Bayart. Optimization of maintenance actions for a multi-component control system and for planned mission duration. In *Proceedings of 2nd IFAC Workshop on Advanced Maintenance Engineering, Service and Technology (A-Mest'12), Seville, Spain*, volume 2, November 2012. 8

[52] N. Iyer, K. Goebel, and P. Bonissone. Framework for post-prognostic decision support. In *Proceedings of 2005 IEEE Aerospace Conference, Big Sky, MT*, pages 1–10, March 4-11 2006. 11, 12, 15, 16, 141

[53] R. Izadi-Zamanabadi and M. Blanke. A ship propulsion system as a benchmark for fault-tolerant control. *Control Engineering Practice*, 7(2) :227–239, 1999. 29

[54] A. K. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20 (7) :1483 – 1510, 2006. 8, 9, 10, 11

[55] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni. Aging modeling of PEMFC for prognostics and health assessment. In *Proceedings of the 9th International Symposium on Fault Dectection, Supervision and Safety for Technical Processes (SAFEPROCESS'15), Paris, France*, September 2-4 2015. 142

[56] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni. Joint particle filters prognostics for PEMFC power prediction at constant current solicitation. *IEEE Transactions on Reliability*, pages 1–14, February 2015. 142

[57] A. H. G. R. Kan. *Machine scheduling problems : classification, complexity and computations.* Nijhoff, The Hague, 1976. 23

[58] M.-H. Karray, B. Chebel-Morello, and N. Zerhouni. A trace based system for decision activities in CBM process. In *Proceedings of IEEE International Conference on Prognostics and Health Management, Gaithersburg, Maryland*, June 2013. 11, 16

[59] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *Society for Industrial and Applied Mathematics (SIAM) Review*, 51 (2)(2) :339–360, 2009. 123

[60] H. Kimura, M. Sato, Y. Hotta, T. Boku, and D. Takahashi. Empirical study on reducing energy of parallel programs using slack reclamation by dvfs in a power-scalable high performance cluster. In *Proceedings of IEEE International Conference on Cluster Computing, Barcelona, Spain*, 2006. 31

[61] R. Kothamasu. *Intelligent Condition Based Maintenance - A Soft Computing Approach to System Diagnosis and Prognosis.* Phd thesis in industrial engineering, University of Cincinnati, 2005. 9

[62] A. Kovacs, G. Erdos, L. Monostori, and Z. J. Viharos. Scheduling the maintenance of wind farms for minimizing production loss. In *Proceedings of the 18th IFAC World Congress, Milano, Italy*, 2011. 13, 14, 15

[63] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica : Journal of the Econometric Society*, 28(3) :497–520, July 1960. 25

[64] M. Lebold and M. Thurston. Open standards for condition-based maintenance and prognostic systems. In *Proceedings of the 5th Annual Maintenance and Reliability Conferance (MARCON), Gatlinburg, USA*, 2001. 8, 9, 11, 16

[65] C.-Y. Lee and V. Leon. Machine scheduling with a rate-modifying activity. *European Journal of Operational Research*, 128 :119–128, 2001. 30

[66] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel. Prognostics and health management design for rotary machinery systems - review, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1) :314–334, 2014. 10

[67] X. Lei, P. Sandborn, R. Bakhshi, A. Kashani-Pour, and N. Goudarzi. PHM based predictive maintenance optimization for offshore wind farms. In *Proceedings of IEEE International Conference on Prognostics and Health Management, Austin, Texas*, June 22-25 2015. 13, 14, 15

[68] G. Levin, B. Rozin, and A. Dolgui. Optimization of multi-tool cutting modes in multi-item batch manufacturing system. In *Proceedings of the IFAC Conference on Manufacturing Modelling, Management, and Control (MIM), Saint Petersburg, Russia*, June 2013. 28, 31

[69] Y. Li and P. Nilkitsaranont. Gas turbine performance prognostic for condition-based maintenance. *Applied Energy*, 86 (10) :2152–2161, 2009. 9

[70] Y. Ma, C. Chu, and C. Zuo. A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58 :199–211, 2010. 27

[71] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15 (1) :229–251, 2004. 112, 119

[72] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley - Interscience Series in Discrete Mathematics, 1983. 117

[73] T.-B.-L. Nguyen, M. Djeziri, B. Ananou, M. Ouladsine, and J. Pinaton. Degradation modelling with operating mode changes. In *Proceedings of IEEE International Conference on Prognostics and Health Management (PHM), Austin, TX*, June 2015. 19, 142

[74] M. Orchard, L. Tang, B. Saha, K. Goebel, and G. Vachtsevanos. Risk-sensitive particle-filtering-based prognosis framework for estimation of remaining useful life in energy storage devices. *Studies in Informatics and Control*, 19(3) :209–218, 2010. 17

[75] I. H. Osman and G. Laporte. Metaheuristics : A bibliography. *Annals of Operations research*, 63(5) :511–623, 1996. 27

[76] D. Pandey, M. S. Kulkarni, and P. Vrat. Joint consideration of production scheduling, maintenance and quality policies : a review and conceptual framework. *International Journale of Advanced Operations Management*, 2 (1-2) :1–24, 2010. 28

[77] R. J. Patton. Fault-tolerant control systems : the 1997 situation. In *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, pages 1033–1054, 1997. 29

[78] E. B. Pereira, R. K. H. Galvão, and T. Yoneyama. Model predictive control using prognosis and health monitoring of actuators. In *Proceedings of IEEE International Symposium on Industrial Electronics (ISIE), Bari, Italy*, pages 237–243, 2010. 13

[79] M. Pinedo. *Scheduling - Theory, Algorithms, and Systems.* Prentice Hall Series in Industrial and Systems Engineering. Prentice Hall, 1995. 22, 51, 52

[80] D. R. Prescott, J. D. Andrews, and C. G. Downes. Multiplatform phased mission reliability modelling for mission planning. *Journal of Risk and Reliability*, 223 (1) :27–39, 2009. 15, 16, 17

[81] D. R. Prescott, R. Remenyte-Prescott, and J. D. Andrews. A systems reliability approach to decision making in autonomous multi-platform missions operating a phased mission. In *Proceedings of the Annual Reliability and Maintainabiilty Symposium*, pages 8–14, 2008. 13, 17

[82] D. R. Prescott, R. Remenyte-Prescott, S. Reed, J. D. Andrews, and C. G. Downes. A reliability analysis method using binary decision diagrams in phased mission planning. *Proceedings of the Institution of Mechanical Engineers, Part O : Journal of Risk and Reliability*, 223(2) :133–143, 2009. 16

[83] J. Puchinger and G. R. Raidl. *Combining metaheuristics and exact algorithms in combinatorial optimization : A survey and classification.* Springer Berlin Heidelberg, 2005. 27

[84] R. Rao, S. Vrudhula, and D. Rakhmatov. Analysis of discharge techniques for multiple battery systems. In *Proceedings of the International Sympoium on Low Power Electronics and Design (ISLPED'03), Seoul, Korea*, 2003. 13, 15, 18

[85] F. Rothlauf. *Design of Modern Heuristics : Principles and Application.* Natural Computing Series. Springer, 2011. 26, 89

[86] B. Saha and K. Goebel. Uncertainty management for diagnostics and prognostics of batteries using bayesian techniques. In *Proceedings of IEEE Aerospace Conference, Big Sky, MT, USA*, pages 1–8, 2008. 17

[87] B. Saha, K. Goebel, S. Poll, and J. Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, 58(2) :291–296, February 2009. 17

[88] B. Saha, E. Koshimoto, C. C. Quach, E. F. Hogge, T. H. Strom, B. L. Hill, S. L. Vazquez, and K. Goebel. Battery health management system for electric uavs. In *Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA*, March 2011. 13, 15, 18

[89] B. Saha, C. C. Quach, and K. Goebel. Optimizing battery life for electric uavs using a bayesian framework. In *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, March 2012. 18

[90] P. Sandborn. A decision support model for determining the applicability of prognostic health management (PHM) approaches to electronic systems. In *Proceedings of Reliability and Maintainability Symposium (RAMS), Arlington, VA*, 2005. 11, 13, 14

[91] T. Saridakis. Design patterns for graceful degradation. *Transactions on Pattern Languages of Programming, Springer Berlin Heidelberg*, 1 :67–93, 2009. 28, 29

[92] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher. Metrics for evaluating performance of prognostic techniques. In *Proceedings of IEEE International Conference on Prognostics and Health Management, Denver, CO*, pages 1–17, 2008. 10

[93] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott. Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In *Proceedings of the 8th International Symposium on High-Performance Computer Architecture (HPCA), Cambridge, MA*, February 2002. 31

[94] A. Singh and A. Rossi. A genetic algorithm based exact approach for lifetime maximization of directional sensor networks. *Ad Hoc Networks*, 11 :1006–1021, 2013. 18

[95] W.-K. Son, O.-K. Kwon, and M. E. Lee. Fault tolerant model-based predictive control with application to boiler systems. In *Proceedings of IFAC SAFEPROCESS'97, Hull, UK*, 1997. 29

[96] R. F. Stengel. Intelligent failure-tolerant control. *IEEE Control Systems Magazine*, 11(4) :14–23, June 1991. 29

[97] L. Tang, E. Hettler, B. Zhang, and J. DeCastro. A testbed for real-time autonomous vehicle PHM and contingency management applications. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Montreal, Quebec, Canada*, pages 1–11, September 2011. 15, 16, 17

[98] L. Tang, G. J. Kacprzynski, K. Goebel, A. Saxena, and G. Vachtsevanos. Prognostics-enhanced automated contingency management for advanced autonomous systems. In *Proceedings of the IEEE International Conference on Prognostics and Health Management, Denver, CO*, 2008. 17

[99] L. Tang, G. J. Kacprzynski, K. Goebel, and G. Vachtsevanos. Case systems for prognostics-enhanced automated contingency management for aircraft systems. In *Proceedings of IEEE Aerospace Conference, Big Sky, MT, USA*, March 06-13 2010. 17

[100] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58 (1) :267–288, 1996. 112, 123, 124

[101] M. A. Trick. Scheduling multiple variable-speed machines. *Operations Research*, 42 :234–248, 1994. 28, 30

[102] C. Valdez-Flores and R. Feldman. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics (NRL)*, 36(4) :419–446, 1989. 14

[103] R. J. Vieira, M. Lietti, and M. A. Sanz-Bobi. New variable health threshold based on the life observed for improving the scheduled maintenance of a wind turbine. In *Proceedings of the 2nd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology, Seville, Spain*, November 2012. 13, 14, 15

[104] L. Wang, G. von Laszewski, J. Dayal, and F. Wang. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In *Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), Melbourne, Australia*, 2010. 31

[105] W. Wang. A model to determine the optimal critical level and the monitoring intervals in condition-based maintenance. *International Journal of Production Research*, 38(6) :1425–1436, 2000. 13, 14

[106] P. Yontay, R. Pan, and O. A. Vanli. Sampling schedule optimization of embedded wireless sensors for degradation monitoring. In *Proceedings of the IEEE International Conference on Prognostics and Health Management (PHM), Gaithersburg, MD, USA*, pages 1–6, June 24-27 2013. 15, 19

[107] B. Zhang, L. Tang, J. A. DeCastro, and K. Goebel. Prognostics-enhanced receding horizon mission planning for field unmanned vehicles. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Portlan, Oregon*, 2011. 17

[108] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101 (476) :1418–1429, 2006. 124

Résumé :

Cette thèse propose une approche originale d'ordonnancement de la production de plates-formes de machines hétérogènes et distribuées, utilisées en parallèle pour fournir un service global commun. L'originalité de la contribution réside dans la proposition de modifier les conditions opératoires des machines au cours de leur utilisation. Il est supposé qu'utiliser une machine avec des performances dégradées par rapport à un fonctionnement nominal permet d'allonger sa durée de vie avant maintenance. L'étude s'inscrit dans la partie décisionnelle du PHM (Prognostics and Health Management), au sein duquel une étape de pronostic permet de déterminer les durées de vie résiduelles des machines. Le problème d'optimisation consiste à déterminer à chaque instant l'ensemble des machines à utiliser et un profil de fonctionnement pour chacune d'entre elles de manière à maximiser l'horizon de production de la plate-forme avant maintenance. Deux modèles sont proposés pour la définition des profils de fonctionnement. Le premier traduit le comportement à l'usure de machines pouvant fournir un nombre discret de performances. Pour ce cas, la complexité de plusieurs variantes du problème d'optimisation est étudiée et plusieurs méthodes de résolution optimales et sous-optimales sont proposées pour traiter le problème d'ordonnancement. Plusieurs méthodes de résolution sous-optimales sont ensuite proposées pour le second modèle, qui s'applique à des machines dont le débit peut varier de manière continue entre deux bornes. Ces travaux permettent de déterminer la durée maximale d'utilisation avant défaillance d'un système à partir des durées de vie résiduelles des équipements qui le composent.

Mots-clés : Décision, Ordonnancement, Décision post-pronostic, Résolution optimale, Heuristiques

Abstract:

This thesis addresses the problem of maximizing the production horizon of a heterogeneous distributed platform composed of parallel machines and which has to provide a global production service. Each machine is supposed to be able to provide several throughputs corresponding to different operating conditions. It is assumed that using a machine with degraded performances compared to nominal ones allows to extend its useful life before maintenance. The study falls within the decisional step of PHM (Prognostics and Health Management), in which a prognostics phase allows to determine remaining useful lives of machines. The optimization problem consists in determining the set of machines to use at each time and a running profile for each of them so as to maximize the production horizon before maintenance. Machines running profiles are defined on the basis of two models. First one depicts the behavior of machines used with a discrete number of performances. For this case, the problem complexity is first studied considering many variants of the optimization problem. Several optimal and sub-optimal resolution methods are proposed to deal with the scheduling problem. Several sub-optimal resolution methods are then proposed for the second model, which applies to machines whose throughput rate can vary continuously between two bounds. These research works allow to determine the time before failure of a system on the basis of its components remaining useful lives.

Keywords: Decision Making, Scheduling, Post-prognostics decision, Optimal resolution, Heuristics

SPIM

■ École doctorale SPIM 16 route de Gray F - 25030 Besançon cedex
■ tél. +33 (0)3 81 66 66 02 ■ **ed-spim@univ-fcomte.fr** ■ www.ed-spim.univ-fcomte.fr

UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ