RAIRO. RECHERCHE OPÉRATIONNELLE

PHILIPPE BAUDET
CATHERINE AZZARO-PANTEL
LUC PIBOULEAU
SERGE DOMENECH

Un couplage entre un algorithme génétique et un modèle de simulation pour l'ordonnancement à court terme d'un atelier discontinu de chimie fine

RAIRO. Recherche opérationnelle, tome 33, n° 3 (1999), p. 299-338

<http://www.numdam.org/item?id=RO_1999__33_3_299_0>

RAIRO Operations Research
RAIRO Oper. Res. **33** (1999) 299-338

UN COUPLAGE ENTRE UN ALGORITHME GÉNÉTIQUE ET UN MODÈLE DE SIMULATION POUR L'ORDONNANCEMENT À COURT TERME D'UN ATELIER DISCONTINU DE CHIMIE FINE (*)

par Philippe BAUDET, Catherine AZZARO-PANTEL,
Luc PIBOULEAU et Serge DOMENECH ([1])

Communiqué par Jacques A. FERLAND

Résumé. – *Dans cette étude, nous proposons le couplage d'un simulateur à événements discrets avec un algorithme génétique, pour aborder l'ordonnancement d'une campagne de production dans un atelier discontinu de chimie fine, problème au caractère combinatoire très marqué. Le simulateur intègre les principales contraintes et spécificités du domaine chimique et constitue ainsi une fonction d'évaluation réaliste de la procédure d'optimisation stochastique. Après une évaluation de la combinatoire du problème, nous présentons la mise en œuvre de ce couplage et nous montrons, à travers un exemple de taille industriellement réaliste (24 équipements, 140 produits à fabriquer suivant 12 recettes d'élaboration différentes et sous la contrainte de 40 produits à recycler pendant la campagne), comment cette approche peut conduire à des augmentations de performances de production importantes. Trois critères techniques de production sont envisagés : la date d'achèvement de la campagne, le temps de cycle moyen des produits et le respect de dates d'attente imposées pour les produits à élaborer. Deux paramètres d'action fixent le champ d'investigation de la procédure : l'ordre de lancement dans l'atelier et/ou l'attribution des heuristiques pour la gestion des conflits. Nous commentons et analysons l'ensemble des résultats obtenus par la mise en œuvre du couplage, puis nous dégageons quelques perspectives.*

Mots clés : Ordonnancement, job-shop, chimie fine, simulateur à événements discrets, optimisation, algorithme génétique.

Abstract. – *In this paper, a discrete-event simulation model is coupled with a genetic algorithm to treat highly combinatorial scheduling problems encountered in a production campaign of a fine chemistry plant. The main constraints and features of fine chemistry have been taken into account in the development of the model, thus allowing a realistic evaluation of the objective function used in the stochastic optimization procedure. After a presentation of problem combinatorics, the coupling strategy is then proposed and illustrated by an example of industrial size (24 equipment items, 140 products, 12 different production recipes and 40 products to be recycled during the campaign). This example serves as an incentive to show how the approach can improve production performance. Three technical criteria have been studied: campaign completion time, average product cycle time, respect of due-dates. Two kinds of optimization variables have been considered: product input order and/or allocation of heuristics for conflit treatment. The results obtained are then analysed and some perspectives of this work are presented.*

Keywords: Schedulig, job-shop, fine chemistry, discrete-event simulation, optimization, genetic algorithm.

(*) Reçu en mars 1997.

([1]) Laboratoire de Génie Chimique, UMR 5503 du CNRS, ENSIGC INPT-UPS, 18 chemin de la Loge, 31078 Toulouse Cedex, France.

Recherche opérationnelle/Operations Research, 0399-0559/99/03/$ 7.00

1. INTRODUCTION

Dans une récente étude [2–4], un modèle de simulation pour l'ordonnancement à court terme d'un atelier de chimie fine a été développé. Ce type d'atelier, multitâches-multiproduits (ou « jobshop »), met en œuvre des procédés discontinus et présente des contraintes très complexes, telles que la grande diversité des produits et de leurs voies d'élaboration, la présence d'intermédiaires réactionnels, l'utilisation de politiques de stockage particulières (par exemple, limitées en volume et éventuellement dans le temps pour des produits instables), la vérification de bilans matières, la consommation d'utilités (eau, vapeur, électricité...), la limitation du taux de rejet de sous-produits polluants et la présence d'opérateurs (entrée et sortie de produits, interventions diverses en cours de procédés, maintenance...). Nous avons abordé ce problème de nature discrète en développant un modèle de simulation à événements discrets baptisé ADHOC (Ateliers Discontinus - Heuristiques et Ordonnancement à Court-terme) présentant un degré de détail important, pour une prise en compte efficace des contraintes précitées. Des notions peu souvent considérées et pourtant bien réelles sur un site productif, comme le degré de polyvalence des opérateurs, les phases de nettoyage ou de maintenance, ou la nécessité de recyclage de certains sous-produits, ont également été prises en compte lors du développement du modèle.

La mise en œuvre du simulateur a été illustrée par un exemple de taille industriellement réaliste, comportant un parc de 24 équipements, 140 lots à élaborer et 40 lots à recycler selon une recette spécifique, plusieurs niveaux d'utilités, l'affectation des opérateurs dans des zones particulières de l'atelier. Outre la simulation détaillée d'une campagne de production, cet exemple a permis de montrer diverses utilisations possibles d'ADHOC, selon les besoins de la gestion de production (outil prédictif d'aide à la décision, aide directe à la production, réactions aux aléas de production, aide à la conception d'ateliers, optimisation manuelle pour la mise au point de paramètres de fonctionnement, fonction d'évaluation des performances de production).

Dans ce travail, on souhaite accroître encore les capacités de cet outil d'ordonnancement pour proposer automatiquement différentes stratégies pour l'organisation d'une campagne, afin d'optimiser un critère de production donné. L'étude est basée sur la recherche d'un ordre de lancement des lots et/ou sur l'attribution des heuristiques pour la gestion des conflits de l'atelier ; les critères envisagés concernent la minimisation du temps de cycle moyen des lots, des avances et retards par rapport à des dates d'attente spécifiées ou de la date d'achèvement de la campagne.

Le caractère combinatoire extrêmement marqué de ce type de problème, qui est analysé ci-après, nous a conduits à réaliser cette phase d'optimisation à l'aide d'un algorithme stochastique plutôt qu'avec une procédure déterministe de type Branch and Bound. Ainsi, nous avons couplé un algorithme génétique (AG) avec ADHOC, ce dernier étant utilisé comme fonction d'évaluation au sein de la procédure d'optimisation.

Dans la première partie de cette étude, après une analyse bibliographique, nous rappelons rapidement les principes et caractéristiques du simulateur et présentons la combinatoire engendrée par le problème de l'ordonnancement. Dans la deuxième partie, après un bref rappel des principes de base des algorithmes génétiques, nous détaillons celui qui a été spécifiquement développé dans ce travail, en insistant en particulier sur les mécanismes d'évolution dans l'espace de recherche et sur les procédures utilisées pour favoriser la convergence. Enfin, dans la dernière partie, le couplage optimiseur-simulateur est réalisé puis illustré à l'aide de l'exemple de grande taille (24 équipements, 140 lots à produire) déjà évoqué.

2. ANALYSE BIBLIOGRAPHIQUE

L'ordonnancement des ateliers discontinus a fait l'objet de nombreuses études depuis ces dernières années dans la littérature spécialisée en Génie des Procédés. Ainsi, dans la revue bibliographique de Reklaitis [50], on constate que peu d'auteurs ont utilisé des méthodes énumératives de type Branch and Bound ou de programmation dynamique. Par contre, de nombreuses études abordent le problème sous forme de programmation linéaire en variables entières ou mixtes, ou bien sous forme de programmation non linéaire en variables mixtes. Pour des problèmes de taille réelle, ces procédures requièrent des temps d'exécution prohibitifs et sont alors généralement associées à des méthodes heuristiques ou des algorithmes de recherche locale, ce qui leur fait perdre beaucoup de rigueur mathématique. Les critères considérés sont soit la minimisation de la durée de la campagne de production, soit la minimisation du temps de cycle moyen des lots ou encore le respect maximum de dates d'attente imposées pour chacun des lots à produire [7, 20, 22, 35, 36, 38, 39, 41, 42, 49, 55-57, 59-63].

L'analyse effectuée par Rippin [51] est sensiblement identique mais observe de plus l'émergence récente de méthodes d'optimisation stochastiques, en particulier celle du recuit simulé [34]. Cette dernière a en particulier été utilisée par [13, 15, 24, 37, 40, 45, 54]. Une procédure voisine du recuit simulé, la méthode Tabou [26] a été utilisée par Hérault [30]

pour ordonnancer une chaîne de montage. D'autres auteurs ont également recours à des procédures de type réseaux de neurones [30, 31, 43, 53, 58]. Dans une optique un peu similaire, les algorithmes génétiques reposent sur un mimétisme des mécanismes d'évolution des espèces dans leur milieu naturel [16, 27, 28, 33]. Des problèmes d'ordonnancement dans le domaine de la productique ont ainsi été abordés par cette approche [6, 10-12, 16, 19, 23, 44, 48, 57].

L'intérêt évident de l'approche mathématique couramment employée en génie des procédés est de garantir la solution optimale. Malheureusement, pour des problèmes de taille réelle et *a fortiori* pour des ateliers de type job-shop, elle devient inutilisable, dans des temps de calcul raisonnables. Ku et Karimi [36] estiment que la seule prise en compte détaillée des politiques de stockage pour les intermédiaires réactionnels rend le problème suffisamment complexe pour condamner une approche purement mathématique. De même, Hofmeister *et al.* [32] signalent deux limitations principales à l'utilisation des méthodes à approche optimale :

- la première concerne le degré de représentation du système. Ces méthodes ne peuvent représenter finement le système productif, qu'en conduisant très vite à des formalismes non linéaires en variables mixtes ;
- la seconde est liée au problème combinatoire. Plus le degré de détail et la taille du problème sont importants, plus le caractère combinatoire est marqué.

Pour le type de problème que l'on se propose d'étudier (NP-complet, selon Gotha [29]), et pour des cas de taille réelle, la solution optimale ne peut, compte tenu des outils aujourd'hui à disposition, être obtenue en un temps d'exécution raisonnable. L'approche combinant des algorithmes optimaux à des procédures heuristiques permet bien sûr d'accélérer la recherche, mais les méthodes résultantes perdent leur intérêt principal originel, c'est-à-dire l'obtention garantie de la solution optimale.

Parmi les méthodes stochastiques, les procédures de voisinage (recuit simulé, méthodes Tabou, algorithmes génétiques) sont particulièrement intéressantes, compte tenu du peu d'hypothèses requises sur la fonction d'évaluation du problème. Parmi ces trois procédures, aucune n'a véritablement démontré sa supériorité par rapport aux autres [1]. Pourtant, la caractéristique spécifique des algorithmes génétiques, c'est-à-dire l'exploration simultanée de plusieurs points de l'espace des solutions, conduisant finalement à l'obtention de plusieurs bonnes solutions du problème posé, nous semble un argument important pour le choix de

la méthode. En effet, en fin de procédure, on peut laisser la liberté à l'organisateur de production de choisir celle qui lui convient le mieux, eu égard à des critères qui parfois, sont difficiles à modéliser. Notre choix s'est ainsi finalement porté sur un algorithme génétique. Bien sûr, la solution optimale n'est pas garantie, même si les études théoriques sur la convergence de cette méthode continuent à progresser [14, 21].

3. PRINCIPES DE BASE D'ADHOC ET ASPECT COMBINATOIRE DU PROBLÈME

Rappelons que la simulation à événements discrets consiste, pour un système de production donné, à reproduire pas à pas, événement par événement, l'évolution dans le temps de l'état du système. Une étude détaillée est proposée dans [5]. Nous ne présentons ici que les éléments importants pris en compte dans ADHOC. Les objets modélisés sont de natures diverses : les ressources consommables (matières premières), à seuil de disponibilité (utilités), le niveau de rejet maximum toléré, les produits (produits finaux à élaborer, sous-produits recyclés sur site ou non) et les ressources renouvelables (équipements, opérateurs et bacs de stockage pour intermédiaires réactionnels). Ils ont été représentés par le formalisme des machines à états. Nous présentons à titre d'exemple la machine à états « équipement » sur la figure 1. Chaque machine à états est caractérisée par un ensemble d'états envisageables, de transitions possibles entre ces états (conditionnelles ou prédéterminées), ainsi que par un certain nombre d'attributs fixes ou variables supplémentaires de description, pour compléter sa modélisation.

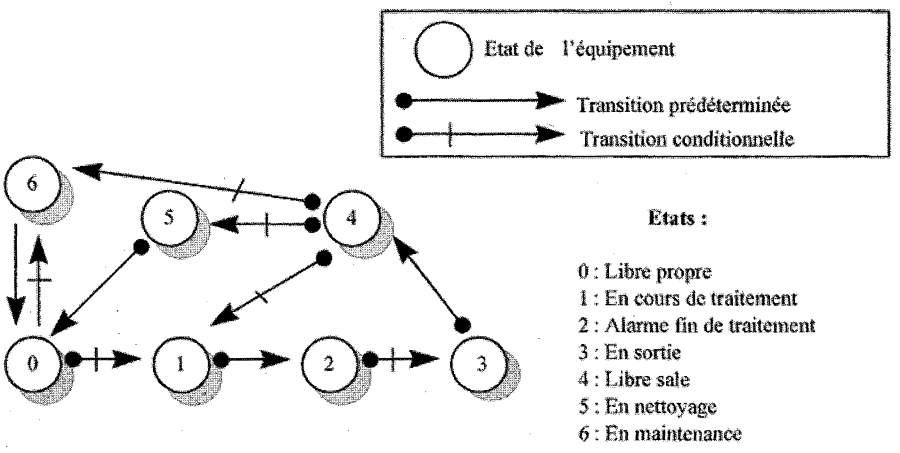

Figure 1. – Principes de la machine à états « équipement ».

Les objets du modèle sont liés entre eux par la notion de recette, de production pour les produits à élaborer, ou de recyclage pour les sous-produits à traiter sur site pendant la campagne. Chaque recette est ainsi une succession d'opérations décrites très précisément pour représenter les processus d'élaboration : équipements envisageables, réactifs, produits, besoins en main d'œuvre et en utilités, effluents, conditions opératoires (température et pression réactionnelles notamment), temps opératoires (temps d'entrée et de traitement des réactifs, temps de sortie des produits, temps de nettoyage de l'équipement après usage).

Les événements considérés au sein du modèle sont de nature différente : conditionnels, prédéterminés ou réservés.

Les événements conditionnels : un événement conditionnel (par exemple, chargement d'un produit dans un équipement) est caractérisé par ses conditions d'occurrence, par la logique de changement d'état liée à son occurrence et par la génération d'événements prédéterminés induits. Les événements conditionnels recensés sont au nombre de 7 (voir Tab. 1).

TABLEAU 1
Présentation des événements conditionnels.

Présentation des événements conditionnels
1. arrêt congé d'un opérateur ;
2. maintenance d'un stockage pour intermédiaire réactionnel ;
3. nettoyage d'un stockage pour intermédiaire réactionnel ;
4. maintenance d'un équipement ;
5. nettoyage d'un équipement ;
6. déchargement des produits d'un équipement ;
7. chargement d'une opération dans un équipement.

Les événements prédéterminés : ces événements ont lieu à des dates préfixées ou prédéterminées (par exemple, le congé d'un opérateur). Un événement prédéterminé ne concerne en général qu'un seul et unique objet du système et son occurrence provoque de la même façon que pour les événements conditionnels, une logique de changement d'état associée.

Les événements réservés : les réservations (par exemple, d'un bac de stockage pour un intermédiaire réactionnel), rendues nécessaires pour le traitement des intermédiaires réactionnels avant des dates au plus tard, compliquent fortement la dynamique de fonctionnement du système, en imposant des événements également déterminés mais plus complexes que

les précédents. Un événement réservé est à la fois prédéterminé (il a lieu sans condition à une date prévue), et conditionnel (avec une logique de changement d'état concernant plusieurs objets et l'induction d'un ou plusieurs événements prédéterminés).

En tenant compte de tous ces éléments, la dynamique du modèle est assez complexe. Certaines des hypothèses retenues pour décrire le fonctionnement sont classiques (temps opératoires fixés *a priori*, tâches non préemptives ...). D'autres sont inhérentes au domaine chimique. L'ensemble des hypothèses est présenté dans [4]. On peut, à titre d'exemple, examiner le cas des réservations, point fondamental dans le modèle.

Ainsi, le chargement d'un produit n'est autorisé que sous des hypothèses strictes, concernant les différents besoins en ressources nécessaires (opérateurs, équipements, utilités, matières premières...) mais également le devenir des produits générés. Ces derniers doivent pouvoir être stockés en sortie d'opération, tout comme l'intermédiaire réactionnel sortant, s'il n'est pas consommé directement. Le fait de ne pouvoir manipuler des intermédiaires instables ou stables que pendant une durée limitée conduit à réserver les ressources nécessaires à leur traitement, pour éviter leur dégradation.

Chaque ressource réservée peut être utilisée si l'on peut garantir sa disponibilité pour la ou les réservations qui lui sont affectées. Ainsi, un équipement réservé pour le traitement d'un intermédiaire non stable pourra indifféremment être utilisé pour la maintenance, le nettoyage ou même le traitement d'une opération, à condition que sa disponibilité pour l'exécution réservée soit assurée.

Au sein du simulateur, nous avons choisi une horloge fonctionnant par « saut » d'événements. Enfin, pour régler les conflits dans l'atelier (la compétition de deux produits en attente du même équipement, par exemple), il faut superposer à l'ensemble des éléments précédents, des règles de décision adaptées. En effet, à un instant donné, plusieurs événements conditionnels peuvent être exécutables. Or, leur ordre d'exécution peut influer directement sur l'évolution de la simulation. Il faut donc aider le modèle à effectuer les choix entre des événements concurrents. Ces problèmes de décision constituent un point crucial du problème de l'ordonnancement. En effet, pour des conditions de fonctionnement imposées, l'efficacité de la gestion de production dépend des choix effectués pour régler chaque conflit rencontré. Sur site, on aborde le problème en utilisant des règles heuristiques. Nous avons ainsi développé au sein du modèle des bases

de règles heuristiques. À titre d'exemple, nous donnons uniquement celles concernant le chargement d'une opération dans un équipement. La base sélectionne en priorité lors de l'occurrence de l'événement « chargement d'un produit dans un équipement » l'équipement disponible selon une des dix règles présentées dans le tableau 2. Notons que pour certaines d'entre elles, sont présentes dans la base une règle et son contraire.

TABLEAU 2
Règles heuristiques implantées dans ADHOC relatives à l'événement « chargement d'une opération ».

Règle heuristique relative à l'événement « chargement d'un produit »
1. Au hasard ;
2. celui disponible depuis le plus longtemps ;
3. celui disponible depuis le moins longtemps ;
4. celui dont la charge de travail déjà effectuée est maximale ;
5. celui dont la charge de travail déjà effectuée est minimale ;
6. celui qui a la charge de travail en attente la plus importante ;
7. celui qui a la charge de travail en attente la moins importante ;
8. celui qui a le nombre d'opérations dans sa file d'attente le plus important ;
9. celui qui a le nombre d'opérations dans sa file d'attente le moins important ;
10. une des neuf règles précédentes choisie au hasard.

3.1. Aspect combinatoire du problème de l'ordonnancement

Nous avons choisi deux variables d'optimisation, utilisées indépendamment ou simultanément, *i.e.*, l'ordre de lancement des lots dans l'atelier et l'affectation des heuristiques pour la gestion des conflits. Commençons la présentation par l'étude de la combinatoire du problème.

Ordre de lancement des lots

Pour Nb lots différents à élaborer, il existe Nb ! possibilités d'effectuer leur lancement dans l'atelier.

Sélection des événements dans le noyau de simulation

L'ordre d'exécution des événements conditionnels est le point déterminant de la procédure d'ordonnancement. Cette gestion des conflits est hiérarchisée en deux niveaux :

– au premier niveau, on choisit un ordre de priorité par type d'événements (choisir, par exemple, d'exécuter les événements de chargement d'opération avant les événements de maintenance). Pour Nev types d'événements conditionnels, on obtient alors Nev ! ordres de priorité

possibles (dans notre cas, 7 types d'événements conditionnels peuvent être concurrents à une même date, voir Tab. 1).

Ensuite, au second niveau, pour les événements de type identique, on doit choisir un ordre d'exécution parmi les événements concurrents suivants : (1) chargement d'une opération (choix de l'équipement; (2) chargement d'une opération (choix de l'opération); (3) déchargement d'un équipement; (4) nettoyage d'un équipement; (5) maintenance d'un équipement; (6) nettoyage d'un stockage; (7) maintenance d'un stockage; (8) choix d'un opérateur. Pour une gestion efficace, on établit alors pour chaque type de conflit, deux règles de décision différentes, dans une base de Nh règles disponibles. La première règle dite principale est le plus souvent suffisante pour régler intégralement le conflit. Dans le cas contraire, on a recours à la seconde (dite règle secondaire). Si cette dernière n'est toujours pas discriminante, la décision est prise finalement au hasard, parmi les événements encore en compétition. Chaque type de conflit peut alors être réglé par un choix ordonné de 2 règles parmi Nh, soit l'arrangement de 2 possibilités parmi Nh (A_2^{Nh} avec Nh égal à 10, dans notre cas).

Attribution globale des heuristiques sur l'atelier

Parmi les 7 événements conditionnels répertoriés dans notre procédure ADHOC, seulement 6 nécessitent une base d'heuristiques (les congés d'opérateurs ne génèrent pas de conflits). Par contre, les événements de type 1 (« chargement d'une opération dans un équipement ») nécessitent deux prises de décision (le choix de l'équipement à charger en priorité d'abord, le choix de l'opération à effectuer ensuite), soit deux bases d'heuristiques. En tout, 8 bases de règles heuristiques doivent être disponibles. D'un point de vue combinatoire, le nombre de possibilités est donc $[A_2^{Nh}]^{Nev+1}$. Nous avons implanté dans chaque cas une base d'heuristiques de 10 règles chacunes (Nh = 10).

Attribution détaillée des heuristiques sur l'atelier

Le dénombrement précédent suppose l'attribution de la même règle heuristique de chargement pour tous les équipements de l'atelier. Si on souhaite utiliser des règles spécifiques pour chacun des Neq équipements de l'atelier, il faut Neq bases de règles, pour déterminer les produits à traiter en priorité. Globalement, le problème combinatoire est donc porté $[A_2^{Nh}]^{Nev+Neq}$ avec Nh égal à 10, Nev égal à 7 et Neq au nombre d'équipements de l'atelier.

Finalement, on peut déterminer les formules littérales relatives au problème combinatoire, en fonction des paramètres d'action. Les trois cas d'études envisagés donnent la combinatoire présentée sur le tableau 3, ce qui correspond aux valeurs fournies dans le tableau 4 pour l'exemple étudié[1]. Cette illustration montre clairement le caractère combinatoire très marqué du problème.

TABLEAU 3
Présentation des paramètres de l'AG.

1. Détermination de l'ordre de lancement des lots (Nb ! combinaisons) ;
2. attribution détaillée des heuristiques ($[A_2^{Nh}]^{Nev+Neq}$ combinaisons) ;
3. détermination simultanée de l'ordre de lancement des lots et des heuristiques, spécifiquement appliquées (Nb ! × $[A_2^{Nh}]^{Nev+Neq}$ combinaisons).

TABLEAU 4
Évaluation de la combinatoire sur l'exemple étudié.

1. 140 !, soit un nombre de l'ordre de 10^{241} ;
2. $[A_2^{10}]^{7+24}$, *soit un nombre de l'ordre de* 10^{64} *combinaisons) ;*
*3. 140 ! ** $[A_2^{10}]^{7+24}$, *soit un nombre de l'ordre de* 10^{305}.

4. MISE EN ŒUVRE DE L'ALGORITHME GÉNÉTIQUE (AG)

Cette technique d'optimisation [29, 33] connaît depuis ces dernières années un succès grandissant pour résoudre les problèmes d'optimisation combinatoire, et particulièrement ceux liés à l'ordonnancement. L'état de l'art sur l'utilisation des AGs en ordonnancement est proposé dans divers articles [12, 46, 47]. Cette approche est basée sur l'analogie entre un individu dans une population et une solution d'un problème, dans un ensemble de solutions. Parmi toutes les solutions d'un problème, une population de taille N est composée de N individus (soit autant de solutions du problème posé), convenablement marqués par un codage spécifique qui les identifie complètement. Pour chacun des individus de la population, on utilise une procédure d'évaluation qui détermine leur « fitness » respectif. Viennent ensuite une phase de sélection et une phase de recombinaison, conduisant à

[1] Dans l'exemple étudié, le nombre de lots à ordonnancer (Nb) vaut 140 (40 des 180 lots à traiter sont des lots de recyclage, non codés dans l'ordre de lancement), le parc (Neq) comprend 24 équipements, 7 événements conditionnels sont répertoriés et chaque base contient 10 règles.

la génération d'une nouvelle population (un nouvel ensemble de N solutions du problème).

- La phase de sélection favorise la survie des individus (solutions) les plus forts au détriment des individus les plus faibles, en utilisant une procédure stochastique qui tient compte de leur force respective.
- La phase de recombinaison consiste à effectuer d'une part, des croisements sur des couples d'individus aléatoirement appariés, selon une probabilité de croisement déterminée et d'autre part, des mutations sur chaque individu de la population, selon une probabilité de mutation donnée. On utilise des opérateurs de croisement et de mutation artificiels pour réaliser cette phase de recombinaison.

Simulant les lois de survie et de sélection naturelle, cette nouvelle population a donc de « bonnes chances » d'être plus forte, plus adaptée, que la population précédente. De génération en génération, la force des individus de la population augmente régulièrement et après un certain nombre d'itérations (c'est-à-dire après une exploration efficace de l'espace des solutions), la population est constituée d'individus tous très forts, soit de bonnes solutions au problème posé. Le principe général de fonctionnement d'un algorithme génétique est présenté sur la figure 2. Nous détaillons ci-après les divers éléments utilisés pour l'implémentation de l'AG considéré ici.

4.1. Les codages

Trois types de codages en nombres entiers ont été définis :

1. Codage de l'ordre de lancement des lots

Le chromosome (1) comporte Nb gènes qui correspondent à Nb lots différents à produire. Le locus du gène caractérise l'ordre de lancement du lot. Notons que ce type d'approche a déjà été utilisé dans la littérature [8, 17, 52].

Exemple Chromosome 1 : Ordre de lancement des lots

5	4	3	2	1

Production à réaliser dans l'ordre : lots 5, 4, 3, 2 et 1.

2. Codage général des heuristiques

Un chromosome (2), long de sept gènes, caractérise l'ordre de priorité d'exécution entre les 7 types d'événements conditionnels (présentés au

Etape 0 : *Définir un codage du problème*

Etape 1 : *Créer une population initiale d'individus* $pop(0) = \{x_1, x_2, ..., x_N\}$

Etape 2 : *Evaluation*
Calculer la «fonction d'adaptatione» $F(x_i)$ de chaque individu $x_i = 1, ..., N$

Etape 3 : *Sélection*
Sélectionner N individus de pop(t) et les ranger dans un ensemble S(t) (Un même individu de pop(t) peut apparaître plusieurs fois dans S(t))

Etape 4 : *Recombinaison*
Grouper les individus de S(t) par paire, puis pour chaque paire d'individus :

Avec la probabilité p_c appliquer le croisement à la paire et recopier la progéniture dans S(t+1). La paire d'individus parents est éliminée et est remplacée par sa progéniture.

Avec la probabilité 1- p_c recopier la paire d'individus dans S(t+1)

Pour chaque individu de S(t+1)

Avec la probabilité p_m appliquer la mutation à l'individu et le recopier dans pop(t+1)
Avec la probabilité 1- p_m recopier l'individu dans pop(t+1)

Etape 5 : *Incrémenter t et reprendre à l'étape 2 jusqu'à vérification d'un critère d'arrêt.*

Figure 2. – Principe général du fonctionnement de l'algorithme génétique.

Tab. 1). Chaque type d'événement est caractérisé par un gène unique, et l'ordre de priorité est déterminé par le locus du gène.

Exemple Chromosome 2: Ordre de priorité entre types d'événements conditionnels

7	6	1	5	4	2	3

On choisit dans l'ordre les événements suivants: type 7, 6, 1, 5, 4, 2, 3 (*cf.* Tab. 1).

Au chromosome de gestion de priorité entre types d'événements est associé un chromosome (3a) pour la gestion des conflits entre événements de même type. Huit sources de conflits possibles ont été enregistrées. Chaque locus contient ainsi la règle à appliquer pour la gestion des conflits d'un type donné. La règle est issue de la base d'heuristiques implantée (10 au total pour chaque type d'événement). Ainsi, chaque gène peut prendre une valeur comprise entre 1 et 10.

Un gène secondaire, associé au gène principal, contient le numéro de la règle à utiliser si la première règle ne suffit pas pour départager tous les événements concurrents. Il est donc différent du premier gène et est également compris entre 1 et 10.

Exemple Chromosome 3a: Gestion des conflits de même nature (attribution globale des heuristiques)

3	**2**	**2**	**10**	**5**	**6**	**10**	**7**
2	*1*	*1*	*3*	*1*	*4*	*1*	*1*

On applique la règle 3 pour le conflit de type 1 (*sinon* 2) ... (*en gras gène principal, en italique gène secondaire*).

3. Codage spécifique des heuristiques par équipement

Illustrons le principe de ce codage (chromosome (3b)) sur l'exemple du choix de produit à traiter sur un équipement donné. Dans ce cas, le gène du chromosome relatif à ce type de conflit est un vecteur d'entiers (avec une partie principale et une partie secondaire), d'une longueur égale au nombre d'équipements de l'atelier.

Exemple Chromosome 3b: Gestion des conflits de même nature (attribution spécifique des heuristiques)

3	**2**	**7**	**6**	**2**	**2**	**10**	**5**	**6**	**10**	**7**
2	*1*	*2*	*1*	*4*	*1*	*3*	*1*	*4*	*1*	*1*

Dans ce cas, on a une attribution d'une règle spécifique pour 4 équipements de l'atelier (encadrement en gras sur le schéma). On a représenté en gras le gène principal (numéro de la règle à appliquer pour chaque gestion de conflits) et en italique le gène secondaire.

4.2. Les opérateurs de croisement et de mutation [4]

Les opérateurs utilisés pour le croisement des chromosomes « simples » (1 seul gène par locus) sont adaptés au problème considéré. Sept opérateurs de croisement [12, 17] ont été mis en œuvre (MPX, PMX, OX, LOX, ordre uniforme, basé sur les positions, de cycle).

Six opérateurs de croisement ont été développés pour les chromosomes comportant des gènes homologues (croisement naturel, 1 point ou k-points, avec ou sans distinction de dominance).

En ce qui concerne la mutation des chromosomes « simples » [12, 17], sept opérateurs sont disponibles (inversion de deux gènes (consécutifs ou aléatoirement choisis), inversion de segments, procédure de transport, d'inversion, procédure d'inversion des extrêmes ou procédure du chenillard).

Enfin, pour les chromosomes avec deux gènes homologues, quatre opérateurs de mutation sont intégrés (mutation d'un ou plusieurs gènes aléatoirement choisis (avec ou sans distinction de dominance), mutation par inversion de dominance d'un ou plusieurs gènes aléatoirement choisis).

L'ensemble de ces procédures, explicitées dans [4], respectent les contraintes de génotype associées au problème considéré. Ainsi, les procédures opérant sur les chromosomes « simples » ne génèrent pas de doublon dans le code génétique des enfants. Pour des raisons de concision, le détail de ces procédures n'est pas reporté ici ; signalons simplement qu'elles diffèrent entre elles par leur aptitude plus ou moins grande à conserver ou inversement à casser le code génétique initial, au cours de la transformation. Nous présentons à titre d'illustration un exemple de chaque procédure en annexe 1. Dans les quatre cas, de croisement et de mutation, on peut choisir une procédure qui sera utilisée pendant tout le déroulement de l'AG, ou sélectionner aléatoirement à chaque génération, la procédure à appliquer parmi celles disponibles.

4.3. Génération de la population initiale

La population initiale peut être générée de façon totalement aléatoire ou construite à partir d'une ou plusieurs solutions testées en simulation, que l'on va dupliquer, avec pour certaines copies, une faible modification de code par application d'une procédure de mutation peu déstructurante.

4.4. Les procédures d'évaluation

Nous utilisons le modèle de simulation ADHOC comme procédure d'évaluation. Le modèle de simulation est ainsi couplé à l'algorithme génétique, retournant la valeur du critère retenu. Trois critères essentiellement techniques ont été envisagés, portant sur la minimisation :

- de la date d'achèvement des tâches (critère A), soit la date de fin de traitement du dernier lot : combiné au nombre de lots encore en cours

à la fin de la simulation, ce critère peut être une bonne évaluation des performances de l'atelier;

- du temps de cycle moyen d'un lot - différence entre sa date de fin de traitement et sa date d'entrée dans l'atelier - (critère B), sans éliminer la possibilité d'avoir des lots toujours en cours à la fin de la campagne simulée, pour un individu testé. Ceci se traduit par une forte pénalisation du critère en fonction du nombre de produits non achevés à la fin de la campagne;
- enfin, du retard (ou de l'avance) des lots par rapport à une date d'attente fixée *a priori* (critère C): ce critère revient à produire les lots à la date attendue, avec une pénalisation pour les avances (évaluée par la racine carrée de la somme des avances), une pénalisation plus forte pour les retards (donnée par la somme des retards) et une pénalisation encore plus forte pour les lots non achevés en fin de campagne.

Ces trois critères ont été éventuellement adaptés pour permettre une différenciation efficace entre les divers individus (cet élément sera mentionné dans la suite de l'article).

Un algorithme génétique maximise une fonction objectif (sélection des individus les plus « forts »). Pour les trois cas étudiés, il faut donc reformuler le critère pour obtenir un problème de maximisation. La fonction d'adaptation, qui sert uniquement à évaluer la force relative des individus en vue d'une sélection judicieuse, est définie comme suit:

$$f_{\text{adaptation}_{\text{individu } i}} = XC_{\max} - \text{Critère}_{\text{individu } i}. \tag{1.a}$$

Dans cette formule, $f_{\text{adaptation}_{\text{individu } i}}$ désigne la fonction d'adaptation de l'individu i;

$$C_{\max} = \text{Max}_{i=1}^{\text{Nombre d'individus}} (\text{Critère}_{\text{individu } i}) \tag{1.b}$$

et X est un réel strictement positif.

Le coefficient X permet d'affecter une force non nulle à l'individu le plus faible, qui conserve donc une chance d'être sélectionné. D'autre part, l'adaptation des individus, à chaque génération, est calculée par rapport à l'individu le plus faible de la population courante (soit celui qui présente le critère maximum), ce qui permet *a priori* de limiter les écarts entre individus, surtout pour les premières générations de l'algorithme.

Nous avons mis en œuvre une procédure de mise à l'échelle ou « scaling » dans la procédure de sélection pour éviter une discrimination trop importante en début de recherche et inversement pour augmenter artificiellement la

différence entre les individus vers la fin [28]. Nous avons ainsi utilisé une mise à l'échelle exponentielle [1, 25, 28]. Dans ce cas, la fonction d'adaptation transformée f' prend la valeur d'une certaine puissance de l'adaptation brute f :

$$f' = f^{\text{coef}\,(n)} \tag{2.a}$$

n désigne le numéro de la génération courante et N le nombre total de générations.

$$\text{Avec coef}\,(n) = \left\{ \text{tg} \left[\left(\frac{n}{N+1} \right) \frac{\pi}{2} \right] \right\}^{0,1} \tag{2.b}$$

Le facteur exponentiel coef (n) est ainsi très petit pour n petit (pour les toutes premières générations), ce qui réduit de façon considérable les écarts de force entre les différents individus. L'AG se comporte alors comme un algorithme de recherche aléatoire. Ensuite, le facteur exponentiel devient très voisin de 1, sauf pour les dernières générations où il augmente jusqu'à des valeurs supérieures à deux, pour une différenciation efficace entre les individus a priori tous très adaptés, à ce stade de l'algorithme.

4.5. La procédure de sélection

Nous avons utilisé comme procédure de sélection la méthode connue sous le nom d'échantillonnage stochastique de la partie restante avec remplacement, qui permet de réduire les erreurs stochastiques liées à la sélection par roue de loterie classique. Son principe est présenté dans [9, 28].

En ce qui concerne le contrôle du nombre de copies d'un même individu pendant la sélection, nous n'avons pas développé de procédure de partage ou « sharing ». En effet, cet atout supplémentaire pour la convergence de l'AG nécessite une procédure de détermination de similitudes entre les individus de la population. Chaque individu de notre population est représenté par 3 chromosomes, dont 2 peuvent être très longs (ordre de lancement des lots et heuristiques à appliquer si elles sont affectées spécifiquement à chaque équipement). La procédure de sharing peut alors devenir très coûteuse en temps, raison pour laquelle nous contrôlons simplement la sélection, en interdisant la reproduction d'un nombre trop important de copies d'un même individu dans la population survivante. Selon les cas, le nombre de copies peut être limité à un quart ou la moitié de la population survivante.

Nous avons également utilisé le modèle élitiste préconisé par De Jong [8] – qui consiste à réintroduire le meilleur individu de la génération t dans la

génération $t + 1$, s'il ne s'y trouve pas déjà – n'a pas été utilisé dans les simulations présentées ici.

5. EXEMPLE D'ILLUSTRATION DU COUPLAGE ADHOC-AG

Le couplage AG-ADHOC est maintenant illustré par un exemple de taille industriellement réaliste. Le but poursuivi est de montrer que la mise en œuvre du couplage permet d'augmenter de façon significative les performances de production de l'atelier, en un temps d'exécution raisonnable. Les solutions théoriquement optimales étant inconnues, pour les différents cas d'études envisagés, nous ne raisonnons pas en terme de convergence vers la solution optimale, mais plutôt en terme de gains en performance de production, par rapport à ceux observés lors de simulations préalables, à conditions de fonctionnement imposées.

Notons que d'autres exemples de ce couplage ont été étudiés et analysés en détail [4] : si la taille des problèmes traités est nettement plus réduite (le nombre de lots à produire étant de 10), le contexte d'application reste le même. Nous comparerons les résultats obtenus dans les divers cas à la fin de cette présentation.

Pour des raisons de concision, nous ne discutons pas la détermination des paramètres de l'algorithme génétique (taille de la population, nombre de générations, choix des procédures de recombinaison et probabilités des recombinaisons notamment) ; elle a été effectuée, dans chaque cas, par des analyses de sensibilité sur l'efficacité de la procédure. Notons que le critère d'arrêt retenu ici est celui d'un nombre de générations fixé *a priori*. Dans les divers cas étudiés, nous évaluons les performances numériques de l'algorithme d'optimisation par son taux d'exploration de l'espace des solutions, défini comme le pourcentage du nombre de solutions évaluées par rapport au nombre total (le calcul se base sur l'évaluation de la combinatoire effectuée précédemment, pour les différents cas d'études). Nous précisons également que les algorithmes génétiques étant des procédures stochastiques, nous effectuons toujours 10 essais.

5.1. Présentation de l'exemple

L'exemple retenu ne mentionne que les points utilisés par la suite (voir [4]) pour toutes précisions supplémentaires). L'atelier considéré est schématisé sur la figure 3. Cet exemple, inspiré du domaine pharmaceutique, élabore sur un même site, des produits liquides et solides. L'architecture de l'atelier,

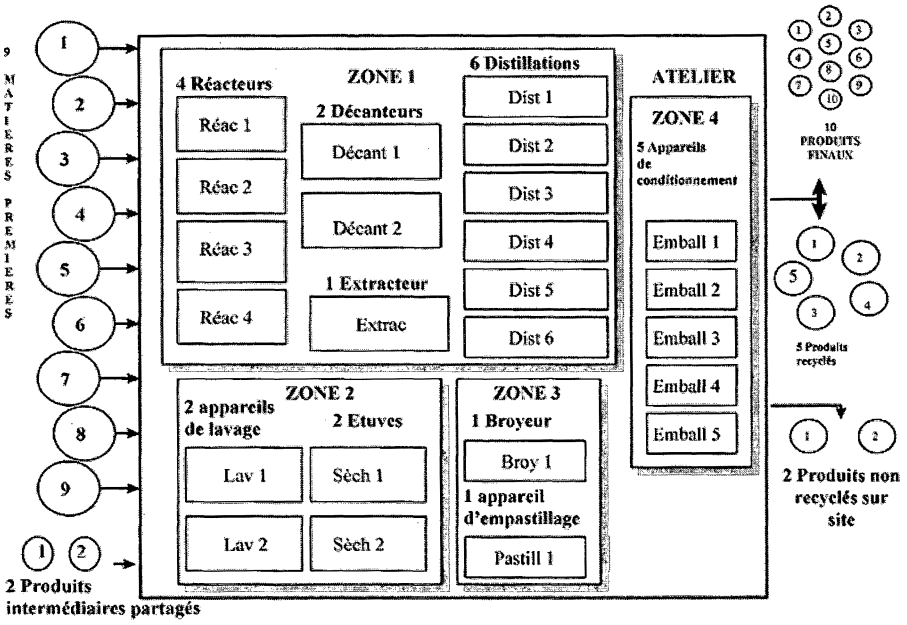

Figure 3. – Schéma de l'atelier.

les ressources, les recettes de production et les conditions opératoires sont fictives. L'atelier est composé de quatre zones distinctes : dans la première, ont lieu les opérations de réaction et de séparation, la deuxième et la troisième sont dédiées au traitement des solides (lavage, séchage, broyage, agglomération) ; enfin, la dernière est destinée à la mise en forme finale des produits. L'atelier compte 24 équipements repartis entre ces quatre zones. Au total, 30 bacs de stockage sont disponibles dans les trois premières zones de l'atelier pour accueillir, si besoin est, des intermédiaires réactionnels entre deux opérations de leur recette d'élaboration. L'atelier fonctionne en continu, 7 jours par semaine, 24 heures par jour. Les équipes d'opérateurs, identiques tant au niveau du nombre (6) que de la polyvalence (2 opérateurs sont polyvalents sur tout l'atelier, les quatre autres étant affectés à une zone spécifique), assurent 3 rotations par jour. Enfin, on suppose que l'atelier est vide en début de simulation, c'est-à-dire sans « en cours » d'une campagne précédente. L'unité de temps choisie est la minute, unité suffisamment petite pour simuler le parallélisme de certaines opérations. Des opérations de maintenance sont prévues pendant la campagne. Le problème de l'ordonnancement consiste à utiliser au mieux ces ressources, sur un horizon de temps de 50 000 minutes (environ 5 semaines), pour réaliser

totalement et dans les meilleures conditions la production demandée. Cette dernière comprend 180 lots à traiter, dont 140 à élaborer et 40 de recyclage, considérés comme des contraintes.

Trois procédures d'optimisation sont testées, chacune pour les trois critères techniques déjà présentés :

- X1 : optimisation de l'ordre de lancement des lots ;
- X2X3 : optimisation de l'attribution détaillée des heuristiques ;
- X1X2X3 : optimisation simultanée de l'ordre de lancement des lots et de l'attribution détaillée des heuristiques.

Les paramètres de l'algorithme génétique sont les suivants : la taille de la population est fixée à 20, le nombre de générations à 200. Au total, le nombre maximum de solutions évaluées sera égal à 4000, ce qui conduit à des taux d'exploration très faibles. Dans le cas des procédures X1 et X1X2X3, il ne peut être calculé précisément (très inférieur à 10^{-99}). Pour la procédure X2X3, il est égal à $2{,}08 \times 10^{-61}$.

Pour cette taille de population et sur une machine IBM 3012, cadencée à 66 MHzs, le temps d'exécution d'une simulation par ADHOC est d'environ 3 secondes CPU. En utilisant le couplage AG-ADHOC, le temps d'exécution, est dans chaque cas, voisin des 200 minutes CPU.

En raison de la complexité du problème envisagé, la population initiale est générée à partir de la solution déterminée par la simulation seule, qui s'avère être une solution relativement bonne du problème, en conduisant à un en cours final nul. Cette solution est l'individu de base et constitue un quart de la population initiale (soit 5 copies de cet individu). La moitié de la population est composée d'individus issus d'une mutation faible de l'individu de base (inversion du locus de 2 paires de gènes), et le dernier quart est généré aléatoirement. Le tableau 5 présente l'ensemble des paramètres de fonctionnement retenus pour l'algorithme génétique utilisé dans les trois cas d'optimisation.

5.2 Simulation de référence et détermination des critères

Plusieurs simulations initiales sont effectuées, pour tester diverses configurations de fonctionnement. La plupart d'entre elles, malgré l'utilisation d'heuristiques *a priori* cohérentes, ne conduisent pas à la réalisation totale de la production. À partir de ces divers essais, un jeu de paramètres qui permet d'élaborer tous les produits a été retenu.

TABLEAU 5
Paramètres de l'AG.

Génération de la population initiale	contrôlée
Taille de la population	20
Nombre de générations	200
Scaling	exponentiel (facteur 0,1)
Limitation du nombre de copies	50
Probabilité de croisement	0,8
Procédure de croisement	– choix aléatoire parmi 7 procédures disponibles pour les chromosomes « ordre de lancement » – choix aléatoire parmi 6 procédures disponibles pour les chromosomes « règles heuristiques »
Probabilité de mutation	0,2
Procédure de mutation	– choix aléatoire parmi 8 procédures disponibles pour les chromosomes « ordre de lancement » – choix aléatoire parmi 4 procédures disponibles pour les chromosomes « règles heuristiques »

La priorité adoptée pour l'ordre d'exécution entre les différents types d'événements est alors le suivant:

1: les événements de type 7 (congés des opérateurs);

2: les événements de type 4 (maintenance des équipements);

3: les événements de type 1 (chargement d'une opération);

4: les événements de type 2 (déchargement des équipements);

5: les événements de type 3 (nettoyage des équipements);

6: les événements de type 5 (nettoyage des stockages);

7: les événements de type 6 (maintenance des stockages).

Les heuristiques utilisées pour les conflits entre événements de même nature sont proposées dans le tableau 6. L'ordre de lancement des lots est libre, fixé par les heuristiques de chargement affectées aux équipements.

Les résultats obtenus par la simulation utilisant ces conditions opératoires ont servi de référence pour l'optimisation des performances de la campagne. Ils sont présentés dans le tableau 7.

Les fonctions d'évaluation retenues (voir Tab. 8) tiennent compte des éventuels produits en cours en fin de campagne, car de nombreuses combinaisons des paramètres de fonctionnement ne permettent pas d'achever la production désirée. On ajoute 1 au nombre de produits en cours pour éviter d'éventuelles multiplications par zéro. Les valeurs numériques des critères pour la simulation de référence sont également présentées dans le tableau 8.

TABLEAU 6
Règles heuristiques appliquées pour la simulation.

Heuristiques « équipement »	1er critère	2e critère	3e critère
Chargement	Celui dont la charge de travail en attente est maximale	Hasard	–
Déchargement	idem	Hasard	–
Nettoyage	Celui qui attend depuis le plus longtemps	Hasard	–
Maintenance	idem	Hasard	–
Heuristiques « produit »	1er critère	2e critère	3e critère
Choix pour chargement	Temps opératoire minimum	Hasard	–
Heuristiques « opérateur »	1er critère	2e critère	3e critère
Choix pour réquisition	Le moins polyvalent habilité à intervenir	Celui dont la charge de travail déjà effectuée est maximale	Hasard
Heuristiques stockages pour intermédiaires		Critère	
Nettoyage Pas de maintenance		Hasard –	

TABLEAU 7
Principaux résultats de la simulation de référence.

Date d'achèvement de la campagne : 44744 minutes.
Temps de cycle moyen des produits (140 lots) : 3568 minutes
Somme de la racine carrée des avances : 7919,4
Somme des retards : 238643 minutes
33 lots de produits sont en retard
Nombre de lots en cours à la fin de la campagne : 0

Les résultats obtenus sont analysés par rapport au meilleur individu obtenu dans le cas d'optimisation le plus efficace, le moins efficace (pour les dix essais) et par rapport à la moyenne des dix meilleures solutions obtenues.

Par ailleurs, nous avons exécuté *nombre d'individus* × *nombre de générations* simulations différentes (avec des configurations choisies de façon aléatoire) et conservé chaque fois le meilleur résultat, pour être comparé ensuite à l'algorithme génétique (procédure $P_{aléat}$).

TABLEAU 8
Fonctions d'évaluation retenues et valeurs des critères pour la simulation de référence.

Critères	Valeurs des critères pour la simulation de référence
A *Minimisation de la date d'achèvement de la campagne.* f_{derdate} = (date d'achèvement/1000). (nombre de produits en cours + 1)^2	44,74
B *Minimisation du temps de cycle moyen des lots.* f_{tcmoy} = (temps de cycle moyen/100). (nombre de produits en cours + 1)^2	35,68
C *Minimisation de l'avance et du retard des lots.* On pénalise ici aussi les retards plus que les avances. $f_{\text{av/ret}} = \left\{ \left[\sum_{j=1}^{n1} \sqrt{\text{avance}\,(j)} + \sum_{j=1}^{n2} \text{retard}\,(j) \right] /1000 \right\}^2$ {nombre de produits en cours + 1^2} avec $n1$, nombre de lots en avanche et $n2$, nombre de lots en retard, par rapport à leur date d'attente spécifiée.	246,56 (dont 96,7 % liés au retard)

5.3 Cas d'optimisation X1

Les résultats d'optimisation, pour les trois critères envisagés, sont proposés sur les figures 4 à 6. Ces figures présentent sur le même graphe, l'évolution du meilleur individu mémorisé et de l'adaptation moyenne de la population, en fonction du nombre de générations. Quel que soit le critère, les différences de valeurs entre le meilleur individu et l'adaptation moyenne nous ont conduit à utiliser 2 échelles (l'échelle de gauche est relative à l'adaptation du meilleur

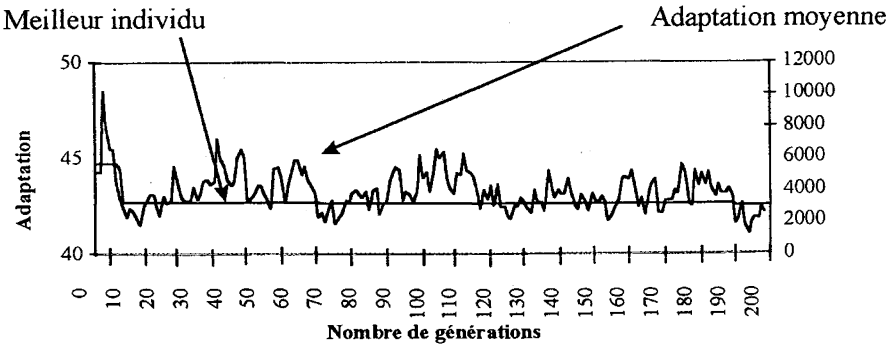

Figure 4. – Courbes d'évolution du cas X1. Critère de la date d'achèvement.

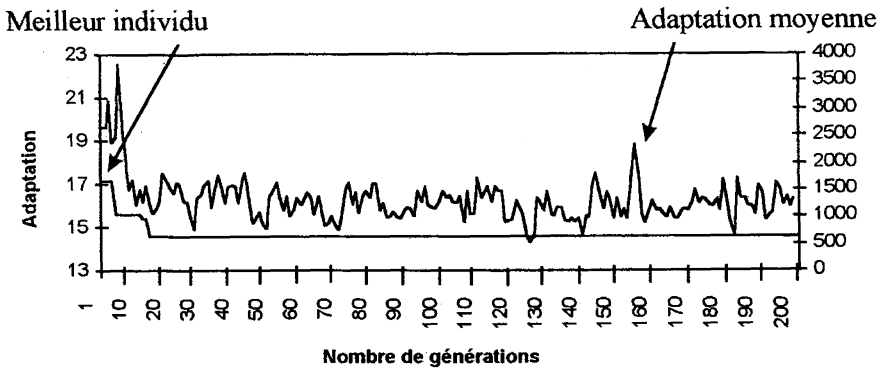

Figure 5. – Courbes d'évolution du cas X1. Critère du temps de cycle moyen.

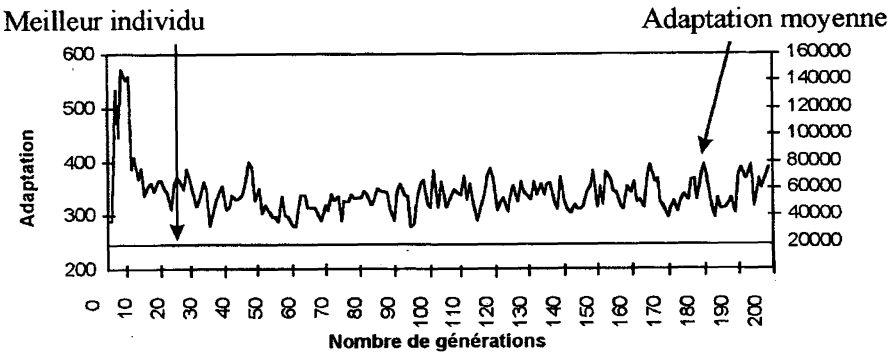

Figure 6. – Courbes d'évolution du cas X1. Critère des avances/retards.

individu). Le tableau 9 résume les résultats obtenus pour les trois critères étudiés.

Les résultats obtenus ne sont probants en terme d'augmentation de performance de production que dans le cas du critère relatif au temps de cycle moyen, avec un gain de l'ordre de 60 % par rapport au résultat obtenu par la simulation. Pour les deux autres critères, les gains sont faibles (4,8 %) ou nuls. On remarque que la procédure $P_{aléat}$ trouve des solutions très éloignées de l'AG, alors que la simulation permet d'avoir une solution acceptable (pour les critères A et B). En ce qui concerne le critère C, les résultats montrent que le meilleur individu obtenu dans le cas d'optimisation le moins efficace est d'une force quasi-équivalente à celui de la procédure aléatoire.

Pour les trois procédures, on peut observer sur les figures 4 à 6, la difficulté de convergence, avec une adaptation moyenne toujours très fluctuante, peu

TABLEAU 9
Résultats de l'optimisation X1.

Valeur de la fonction d'adaptation	Date d'achèvement	Temps de cycle moyen	Avances/retards
Meilleur individu obtenu, dans le cas d'optimisation X1 le plus efficace (AG)	42,58 nenc = 0	15,45 nenc = 0	246,56 nenc = 0
Meilleur individu obtenu dans le cas d'optimisation X1 le moins efficace (AG)	45,48 nenc = 0	22,48 nenc = 0	73822 nenc = 5
Moyenne des dix meilleures solutions obtenues (AG)	43,99 nenc = 0	20,06 nenc = 0	1107 nenc = 0,5
Résultats de simulation (ADHOC)	44,74 nenc = 0	35,68 nenc = 0	246,6 nenc = 0
Meilleur individu obtenu pour 4000 générations aléatoires de X1	1791 nenc = 5	612 nenc = 5	64620 nenc = 5
Moyenne des 4000 individus aléatoires	7056 nenc = 11	2181 nenc = 11	247999 nenc = 11

nenc représente le nombre de lots en cours en fin de campagne ;
10 essais ont été réalisés pour le couplage AG-ADHOC ;
10 essais ont été réalisés pour la génération aléatoire de 4000 individus.

adaptée, et qui reste, en fin de procédure, beaucoup plus élevée que celle du meilleur individu obtenu.

Les résultats font état d'une variance relativement élevée pour le critère des avances/retards (ce phénomène se retrouve pour les cas d'optimisation X2X3 et X1X2X3) ; les essais expérimentaux que nous avons menés montrent qu'il s'agit d'un critère assez difficile à satisfaire ; les *optima* sont disséminés à travers l'espace des solutions et sont en pratique assez difficiles à atteindre. Notre procédure d'optimisation a montré cependant qu'elle donnait de bons résultats par rapport à une recherche purement aléatoire.

Par ailleurs, la génération contrôlée de la population initiale permet de créer une population initiale assez adaptée, dans le cas des avances/retards notamment, où la population initiale est la meilleure obtenue pendant toute la procédure. On remarque d'ailleurs que les meilleurs individus fournis par les procédures apparaissent dans les toutes premières générations (1, 9 et 11), et que la suite ne permet, ni d'améliorer le meilleur individu, ni d'augmenter

de façon significative, l'adaptation moyenne de la population (sauf dans le cas du temps de cycle moyen).

Globalement, le cas d'optimisation X1 est confronté à la génération importante de solutions conduisant à un en-cours final non nul. La probabilité importante d'obtenir ces solutions « non satisfaisantes » par les procédures de recombinaison (croisement et mutation), pénalise très fortement la convergence de ce cas d'optimisation. Seul le critère relatif au temps de cycle moyen apporte une augmentation de performance très importante, avec 56,7 % de gain, en dépit de l'évolution peu convaincante de l'adaptation moyenne. C'est pourquoi nous avons pénalisé de façon importante les solutions à en-cours final non nul.

5.4. Cas d'optimisation X2X3

Le tableau 10 résume les résultats obtenus pour les trois critères étudiés. Le cas d'optimisation X2X3, pour lequel l'ordre de lancement des lots est déterminé par les heuristiques appliquées pour le chargement

TABLEAU 10
Résultats de l'optimisation X2X3.

Valeur de la fonction d'adaptation	Date d'achèvement	Temps de cycle moyen	Avances/retards
Meilleur individu obtenu, dans le cas d'optimisation X2X3 le plus efficace (AG)	37,09 nenc = 0	21,13 nenc = 0	70,11 nenc = 0
Meilleur individu obtenu dans le cas d'optimisation X2X3 le moins efficace (AG)	42,99 nenc = 0	24,19 nenc = 0	384,12 nenc = 5
Moyenne des dix meilleures solutions obtenues (AG)	38,14 nenc = 0	21,84 nenc = 0	192,17 nenc = 0
Résultats de simulation (ADHOC)	44,74 nenc = 0	35,68 nenc = 0	246,6 nenc = 0
Meilleur individu obtenu pour 4000 générations aléatoires de X2X3	192,28 nenc = 1	62,24 nenc = 1	2643 nenc = 1
Moyenne des 4000 individus aléatoires	1789 nenc = 9	587 nenc = 5	25263 nenc = 5

nenc représente le nombre de lots en cours en fin de campagne ;
10 essais ont été réalisés pour le couplage AG-ADHOC ;
10 essais ont été réalisés pour la génération aléatoire de 4000 individus.

des équipements, présente l'avantage considérable de ne générer quasi-exclusivement que des ordonnancements conduisant à des niveaux d'en cours nuls.

La procédure d'optimisation donne des résultats supérieurs à la simulation (dans tous les cas pour les critères A et B, 8 cas sur 10 pour le critère C): on observe des gains de performances allant de 17,1 % à 71,6 % pour le critère des avances/retards. On note également que la procédure $P_{aléat}$, ne permet pas d'obtenir une solution acceptable, d'où l'intérêt de la procédure d'optimisation.

Sur les figures 7 à 9, on observe l'évolution du meilleur individu et de l'adaptation moyenne de la population, en fonction du nombre de générations. La génération contrôlée de la population initiale fournit une première génération déjà assez adaptée, qui évolue peu par la suite. L'adaptation moyenne finale obtenue est très satisfaisante, notamment dans le cas du critère relatif à la date d'achèvement de la campagne, égal à 40,64, soit mieux que le résultat obtenu par la procédure X1 dans le cas le plus favorable.

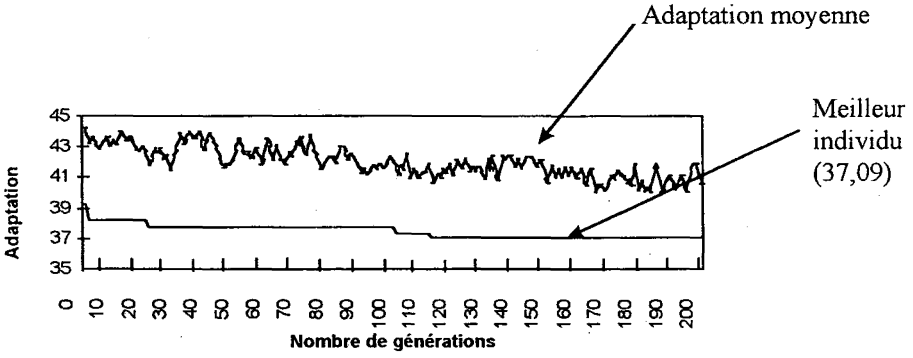

Figure 7. – Courbes d'évolution du cas X2X3. Critère de la date d'achèvement.

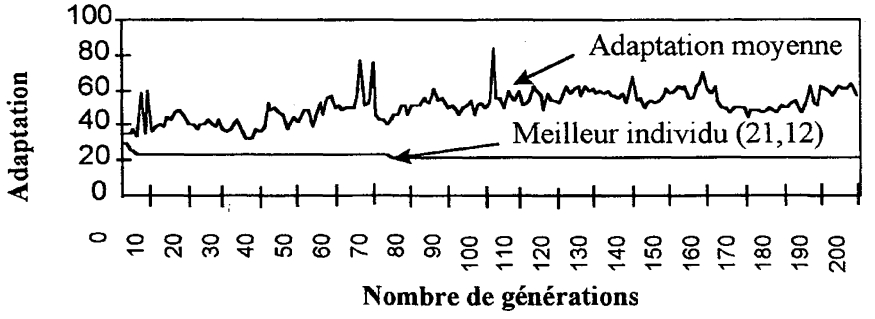

Figure 8. – Courbes d'évolution du cas X2X3. Critère du temps de cycle moyen.

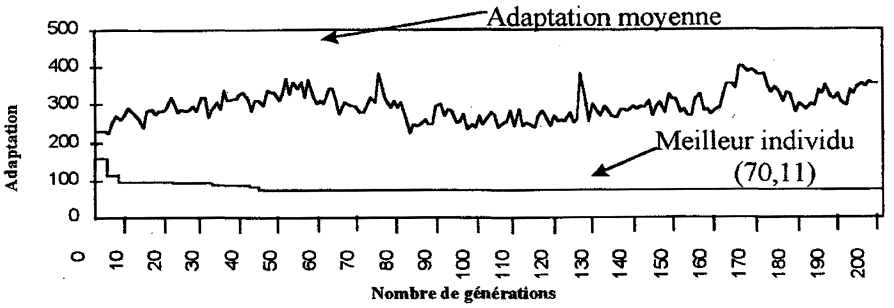

Figure 9. – Courbes d'évolution du cas X2X3. Critère des avances/retards.

5.5 Cas d'optimisation X1X2X3

Le tableau 11 résume les résultats obtenus pour les trois critères étudiés. Les résultats sont sensiblement identiques à ceux fournis par le cas X1, pour les mêmes raisons. L'évolution de la fonction d'adaptation au cours des générations, non reportée ici, présente les mêmes caractéristiques que les figures 4 à 6, obtenues dans le cas X1.

TABLEAU 11
Résultats de l'optimisation X1X2X3.

Valeur de la fonction d'adaptation	Date d'achèvement	Temps de cycle moyen	Avances/retards
Meilleur individu obtenu, dans le cas d'optimisation X1X2X3 le plus efficace (AG)	42,32 nenc = 0	15,71 nenc = 0	243,74 nenc = 0
Meilleur individu obtenu dans le cas d'optimisation X2X3 le moins efficace (AG)	199,28 nenc = 1	18,92 nenc = 0	6685 nenc = 1
Moyenne des dix meilleures solutions obtenues (AG)	91,21 nenc = 0	17,04 nenc = 0	891,17 nenc = 0,1
Résultats de simulation (ADHOC)	44,74 nenc = 0	35,68 nenc = 0	246,6 nenc = 0
Meilleur individu obtenu pour 4000 générations aléatoires de X1X2X3	47,8 nenc = 0	18,44 nenc = 0	351,53 nenc = 0
Moyenne des 4000 individus aléatoires	7407 nenc = 13	3181 nenc = 13	320855 nenc = 13

nenc représente le nombre de lots en cours en fin de campagne ;
10 essais ont été réalisés pour le couplage AG-ADHOC ;
10 essais ont été réalisés pour la génération aléatoire de 4000 individus.

Pour le critère A, dans 3 cas sur 10, la procédure X1X2X3 ne trouve pas de solution à en-cours final nul. Dans le meilleur des cas, le gain obtenu est faible par rapport à la simulation alors que la procédure aléatoire a trouvé une solution à en-cours final nul. Même la forte pénalisation pour l'en-cours final non nul ne garantit pas l'obtention d'une solution acceptable en fin de procédure. Pour le critère C, les gains sont également faibles même si 9 fois sur 10, on est conduit à des solutions à en-cours nul (en raison de la forte pénalisation introduite dans le critère).

Pour le critère B, nous obtenons dans tous les cas des résultats très satisfaisants par la procédure d'optimisation (la procédure aléatoire a trouvé une seule fois une solution à en-cours final nul pour le critère B, reportée Tab. 11).

Globalement, le cas X1X2X3 ne permet pas d'augmenter les performances par rapport au cas X1. Même si les moyennes obtenus ne sont pas très bonnes, cette procédure a néanmoins trouvé des résultats intéressants avec le meilleur individu sur deux des critères.

5.6. Cas X2X3 avec fonction d'évaluation adaptée

Le cas X2X3, qui a fourni les meilleurs résultats, fait cependant apparaître une faible évolution de l'adaptation moyenne, « assez bonne » dès la première génération. Comme cette procédure génère une grande majorité de solutions conduisant à un niveau d'en cours nul, nous avons modifié la fonction d'évaluation pour augmenter la différenciation entre les individus à en cours final nul. Les nouvelles fonctions d'évaluation retenues sont présentées dans le tableau 12.

TABLEAU 12

Nouvelles fonctions d'évaluation retenues et valeurs des critères pour la simulation de référence.

Fonction d'évaluation adaptée
A *Minimisation de la date d'achèvement de la campagne.* f_{derdate} = (date d'achèvement/1000). (nombre de produits en cours + 1)^2
B *Minimisation du temps de cycle moyen des lots.* f_{tcmoy} = (temps de cycle moyen/100). (nombre de produits en cours + 1)^2
C *Minimisation de l'avance et du retard des lots.* On pénalise ici aussi les retards plus que les avances. $f_{\text{av/ret}} = \left\{ \left[\sum_{j=1}^{n1} \sqrt{\text{avance}\,(j)} + \sum_{j=1}^{n2} \text{retard}\,(j) \right] /1000 \right\}^2 \{\text{nombre de produits en cours} + 1\}^2$ avec $n1$, nombre de lots en avanche et $n2$, nombre de lots en retard, par rapport à leur date d'attente spécifiée.

Nous présentons les nouveaux résultats obtenus dans le tableau 13. Les figures 10 à 12 montrent les évolutions du meilleur individu et de l'adaptation moyenne. Dans les trois cas, la modification de la fonction d'adaptation s'est avérée très bénéfique pour le comportement de la méthode. Les figures 10 à 12 présentent une évolution très satisfaisante, tant pour le meilleur individu que pour l'adaptation de la population. Par cette procédure, le gain minimum est de 19,9 % par rapport à la simulation, soit une diminution de la durée de la campagne égale à 8920 minutes. Pour le temps de cycle moyen, les gains s'élèvent à 48,6 % même si nous n'atteignons toujours pas les résultats obtenus par les deux cas d'optimisation X1 et X1X2X3. Enfin, la procédure développée semble tout à fait efficace pour aborder le problème de respect des dates d'attente, avec un gain porté à plus de 85 %, par rapport à la simulation. À titre de comparaison, la somme des temps de retards des lots atteint 238 643 minutes pour la simulation, contre seulement 28 097 minutes, pour le meilleur individu obtenu par la procédure X2X3 modifiée. Au total, 33 lots étaient en retard par simulation, les retards représentant 96,7 % de la pénalisation totale, contre seulement 16 lots en retard après la procédure d'optimisation.

TABLEAU 13
Résultats de l'optimisation X2X3 avec fonction d'évaluatin adaptée.

Valeur de la fonction d'adaptation	Date d'achèvement	Temps de cycle moyen	Avances/retards
Meilleur individu obtenu, dans le cas d'optimisation X2X3 le plus efficace (AG)	12,83 nenc = 0	3,37 nenc = 0	13,19 nenc = 0
Meilleur individu obtenu dans le cas d'optimisation X2X3 le moins efficace (AG)	15,01 nenc = 0	5,01 nenc = 0	22,51 nenc = 0
Moyenne des dix meilleures solutions obtenues (AG)	13,24 nenc = 0	4,46 nenc = 0	19,14 nenc = 0
Résultats de simulation (ADHOC)	44,74 nenc = 0	35,68 nenc = 0	246,6 nenc = 0
Meilleur individu obtenu pour 4000 générations aléatoires de X2X3	192,28 nenc = 1	62,24 nenc = 1	2643 nenc = 1
Moyenne des 4000 individus aléatoires	1789 nenc = 9	587 nenc = 5	25263 nenc = 5

nenc représente le nombre de lots en cours en fin de campagne ;
10 essais ont été réalisés pour le couplage AG-ADHOC ;
10 essais ont été réalisés pour la génération aléatoire de 4000 individus.

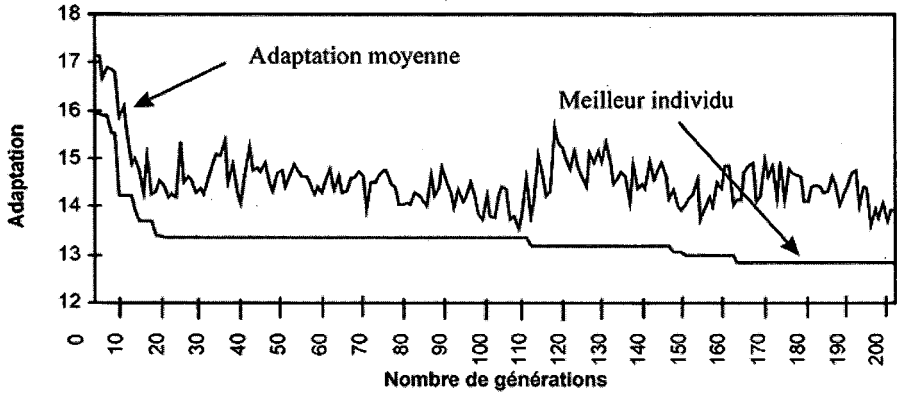

Figure 10. – Cas X2X3 avec fonction d'évaluation adaptée. Critère de la date d'achèvement.

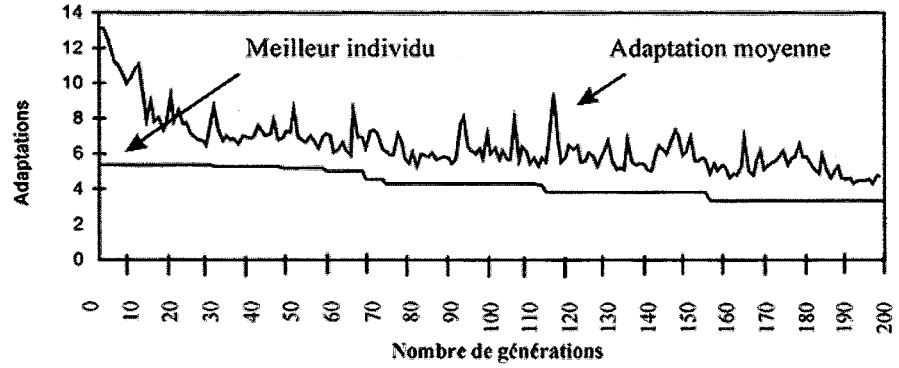

Figure 11. – Cas X2X3 avec fonction d'évaluation adaptée. Critère du temps de cycle moyen.

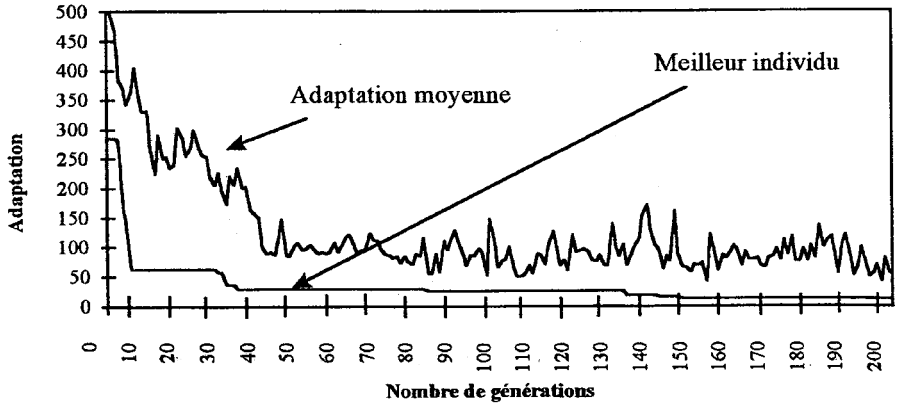

Figure 12. – Cas X2X3 avec fonction d'évaluation adaptée. Critère des avances/retards.

Pour l'adaptation moyenne, on peut considérer approximativement qu'elle est, en fin de procédure, égale au meilleur individu obtenu par la fonction d'adaptation précédente. En d'autres termes, la population finale obtenue dans les trois cas est composé d'individus tous très adaptés, d'une force voisine de celle du meilleur individu créé. On dispose ainsi, dans la population finale, de vingt organisations différentes, très performantes, qui pourront éventuellement être envisagées.

Cette version « modifiée » du cas X2X3 donne des résultats très satisfaisants, tant pour l'obtention de l'individu le plus fort, que pour la convergence de l'adaptation moyenne vers le meilleur critère obtenu.

5.7 Analyse globale des résultats

Le tableau 14 présente les meilleurs résultats obtenus par toutes les procédures d'optimisation en compétition. Compte tenu des très nombreuses configurations générées, dans les cas mettant en jeu le chromosome X1, conduisant à un niveau d'en cours final non nul, on ne peut pas attribuer ces résultats à « une mauvaise simulation » de référence.

TABLEAU 14
Synthèse des meilleurs résultats obtenus.

	Date d'achèvement	Temps de cycle moyen	Avances/retards
Adaptation du meilleur individu	12,83	3,37	13,19
Gain/simulation	19,93 %	48,6 %	85,3 %
Procédure	X2X3 modifié	X2X3 modifié	X2X3 modifié

Le gain par rapport à la simulation a été calculé sur la base du critère précédent à des fins de comparaison.

Nous avons obtenu des gains très intéressants par rapport aux résultats donnés par la simulation, allant de 20 % à plus de 85 % pour le critère des avances/retards.

Plusieurs constatations peuvent être tirées de ces résultats :

- le paramètre de l'ordre de lancement X1 qui induit une combinatoire très importante (de l'ordre de 10^{241} combinaisons possibles), génère de très nombreuses solutions qui conduisent à un niveau d'en cours final non nul, ce qui perturbe très fortement les performances de l'optimiseur. En dépit de l'évolution peu convaincante de l'adaptation moyenne, les

procédures mises en jeu permettent d'obtenir des gains de performance très importants dans le cas du temps de cycle moyen. Pour les deux autres critères, on ne peut pas garantir l'obtention d'une solution acceptable, même si la probabilité d'en obtenir une est plus grande qu'avec la procédure aléatoire (grâce notamment à la sélection des individus).

- la remarque suivante découle directement de la précédente: les deux cas X1 et X1X2X3 deviennent équivalents (d'où une augmentation du degré de sophistication de la procédure qui devient inutile dans ce cas), limitée par le paramètre X1.
- le cas X2X3 s'est avéré très efficace, quel que soit le critère technique envisagé, en particulier pour le critère des avances/retards. Il garantit l'obtention d'une solution acceptable dans tous les cas. Il donne des résultats supérieurs à la simulation dans tous les cas pour 2 critères sur 3, légèrement inférieurs à la simulation pour seulement 20 % des essais réalisés avec le critère des avances/retard.

5.8. Conclusions générales sur l'ensemble des résultats expérimentaux

Cette même méthodologie d'analyse des résultats a été menée sur des problèmes similaires mais de taille plus réduite [4]. Sans présenter de façon exhaustive, l'ensemble des résultats obtenus, on peut néanmoins dire que les mêmes tendances se dégagent, avec des gains relativement plus faibles que pour l'exemple de grande taille. Nous avons également constaté que plus la procédure d'optimisation est sophistiquée (*i.e.*, plus l'espace de recherche exploré est important), plus les performances sont satisfisantes).

6. CONCLUSIONS ET PERSPECTIVES

Les résultats obtenus montrent bien l'efficacité de l'algorithme génétique pour augmenter de façon significative les performances de production en un temps d'exécution raisonnable (aucune procédure n'a excédé 200 minutes CPU et les taux d'exploration de l'espace des solutions sont remarquablement faibles).

La procédure X2X3, après modification de la fonction d'adaptation, a permis d'obtenir des gains de production importants, en particulier pour le critère des avances/retards avec un gain excédant 85 %. Pour une architecture donnée, cette procédure offre l'avantage de conserver un aspect combinatoire constant, indépendant du nombre de lots à traiter.

Le cas X1X2X3 est le plus satisfaisant des cas étudiés, du point de vue de l'espace de recherche exploré, mais la combinatoire due au paramètre d'action X1 ajoutée à la complexité engendrée par la génération fréquente de solutions « monstrueuses » rend la convergence extrêmement difficile. La population dans son ensemble reste ainsi très faiblement adaptée, en dépit de quoi, de bons résultats ont été obtenus dans le cas du temps de cycle moyen (meilleur que dans le cas d'optimisation X2X3).

Le cas X1, le plus fréquemment abordé dans la littérature, présente les deux inconvénients suivants : d'une part, il ne traîte le problème combinatoire qu'en entrée de l'atelier (certes, l'ordre de lancement a un impact sur tout le déroulement de la campagne), en réglant les conflits sur tout le reste de l'atelier par des heuristiques figées. D'autre part, le très grand domaine à explorer pour le chromosome X1, dans le cas de problèmes de grande taille, provoque les mêmes conséquences que dans le cas X1X2X3.

Nous avons montré la difficulté d'aborder le problème par la voie de l'ordre de lancement des lots. Dans cette étude, avec 140 lots à ordonnancer en entrée d'atelier, cette procédure éprouve certaines difficultés de convergence. Il semble donc que le paramètre d'action « ordre de lancement des produits dans l'atelier » ne soit pas très efficace pour aborder les problèmes de grande taille, en raison de l'aspect combinatoire engendré (soit Nb !, où Nb désigne le nombre de lots à ordonnancer, ce qui conduit rapidement à un nombre de combinaisons très important). Par contre, le cas d'optimisation abordant le problème par l'attribution détaillée des heuristiques, qui présente l'avantage de garder une combinatoire indépendante du nombre de lots à traiter, a montré une très bonne efficacité, quel que soit le critère technique envisagé, en particulier vis-à-vis du respect des dates d'attente.

Diverses évolutions possibles peuvent être suggérées, pour l'approche proposée :

- nous avons affecté des heuristiques spécifiques pour chaque équipement. Étant entendu que la « meilleure règle » à attribuer à un équipement n'a aucune raison d'être la même du début à la fin de la campagne, on peut, par la suite, envisager une attribution dynamique de ces heuristiques. Le chromosome représentatif devra être adapté, pour représenter l'ordre complet de la séquence d'heuristiques à imposer ;
- l'ordre de lancement des lots a été codé dans un chromosome X1, ce qui s'est avéré peu efficace sur l'exemple de taille importante proposé, excepté, pour le critère du temps de cycle moyen. On peut envisager d'aborder l'ordre de lancement des lots, non par un chromosome

particulier mais par des règles heuristiques, comme les autres conflits de l'atelier.

ANNEXE 1. Exemples de procédures de croisement et de mutation mises en œuvre [12, 17]

Il est nécessaire de définir des opérateurs de croisement et de mutation adaptés au problème. Les chromosomes représentant l'ordre de lancement des produits comportent un gène par locus, mais tous les gènes entiers le composant doivent être différents, variables de 1 au nombre total de gènes. Les chromosomes représentatifs des règles heuristiques à appliquer au simulateur comportent deux gènes entiers « homologues » (un gène principal et un gène secondaire) différents, compris entre 1 et 10. Les opérateurs génétiques doivent générer des individus enfants qui respectent les contraintes de génotype des individus parents.

Les opérateurs mis en place pour le croisement des chromosomes représentatifs de l'ordre de lancement ne doivent donc pas générer de doublon dans le code génétique des enfants. Nous présentons successivement un exemple de croisement et de mutation utilisé.

Croisement MPX (Maximal Preservative X)

Dans la description de cet opérateur, on considère deux enfants i et j produits à partir de deux parents i et j. Le croisement MPX génère deux enfants dont le code génétique diffère faiblement de celui des deux parents. Deux points de coupure sont générés aléatoirement pour séparer le code génétique des parents en trois zones, le fils i héritant alors de la zone interne du parent i. Le code génétique manquant des fils est alors constitué comme suit :

1. le $k^{\text{ième}}$ gène du fils i contiendra le $k^{\text{ième}}$ gène du parent j, à condition que ce gène ne provoque pas de doublon dans son génotype.
2. Si le gène du parent j ne convient pas, on essaie alors le $k^{\text{ième}}$ gène du parent i.
3. Si aucun de ces deux gènes ne convient, on explore la zone interne du parent j, pour sélectionner le premier des gènes qui ne provoque pas de doublon.

Cette procédure de croisement est illustrée sur la figure A1. Les points de coupure générés aléatoirement sont les points 5 et 9, qui déterminent la zone interne du code génétique, et deux zones externes. L'application de l'opérateur MPX conduit ici à un fils i très voisin du parent i (9 gènes sur

12 identiques et en même position) et un fils j moins similaire au parent j (6 gènes hérités de parent j seulement).

Mutation par inversion de 2 gènes

Cette procédure de mutation est très simple. Elle consiste à inverser la place de 2 gènes choisis aléatoirement. On sélectionne ainsi aléatoirement deux locus du chromosome et on inverse la place des deux gènes concernés. Une variante de cette procédure consiste à ne sélectionner qu'un seul locus et à inverser sa place avec celui du gène suivant. Les deux procédures d'inversion de deux gènes consécutifs ou aléatoirement sélectionnés sont présentées sur la figure A2.

Le croisement des chromosomes avec gènes homologues qui contrôlent les différentes heuristiques à appliquer au modèle de simulation pose moins de problème que celui des chromosomes précédents, puisque la génération de doublon n'est pas interdite. Nous présentons successivement un exemple de croisement et de mutation utilisés pour ce type de chromosomes.

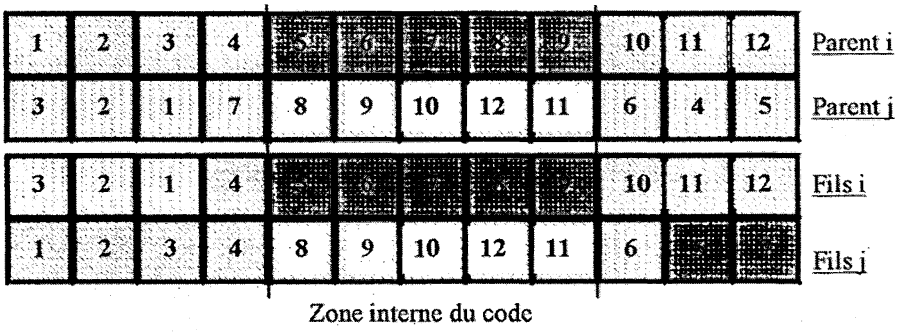

Figure A1. – Exemple de croisement MPX.

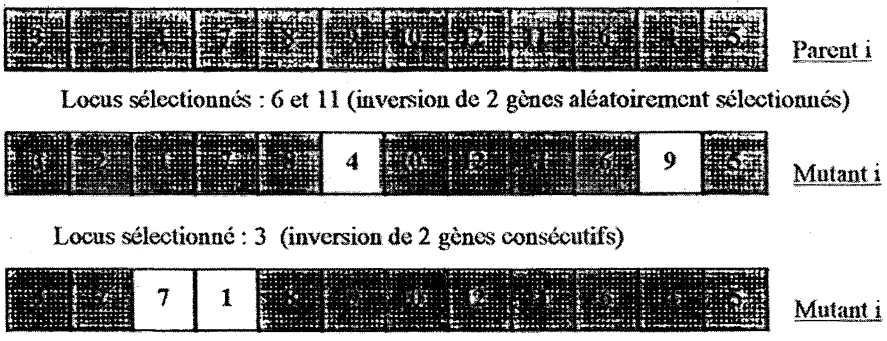

Figure A2. – Procédures d'inversion de 2 gènes (consécutifs ou aléatoirement sélectionnés).

1	2	7	6	2	2	10	5	6	10	7
2	1	2	1	4	1	1	1	1	1	1

Parent i

4	2	4	1	3	2	8	7	4	10	6
10	1	2	4	4	4	1	4	1	4	1

Parent j

a) Croisement 1-point concernant les gènes principaux et secondaires (point de coupure 8).

1	2	7	6	2	2	10	5	4	10	6
2	1	2	1	4	1	1	1	1	1	1

Fils i

4	2	4	1	3	2	8	7	6	10	7
10	1	2	4	4	4	1	4	1	1	1

Fils j

b) Croisement 1-point concernant seulement les gènes principaux (point de coupure 8).

1	2	7	6	2	2	10	5	4	10	6
2	1	2	1	4	1	1	1	1	1	1

Fils i

4	2	4	1	3	2	8	7	6	10	7
10	1	2	4	4	4	1	4	1	4	1

Fils j

Figure A3. – Croisement 1-point pour des chromosomes avec gènes homologues.

Croisement 1 point classique

Cette procédure de croisement est la plus classique. Un point de coupure est généré aléatoirement et sépare les deux chromosomes en deux parties. Ensuite, le fils i est généré avec la première partie du parent i et la deuxième partie du parent j, et inversement pour la génération du fils j. Notons que le point de coupure peut ou non concerner les gènes secondaires. Dans le cas où le croisement ne concerne que les gènes principaux, on doit éviter la génération de doublons entre les gènes principaux et secondaires des fils produits. La figure A3 propose un exemple de croisement de ce type, dans le cas où les gènes secondaires sont concernés par le croisement, et dans le cas où ils ne le sont pas.

Mutation d'un gène choisi aléatoirement

Cette procédure, très simple, consiste à choisir aléatoirement un gène du chromosome et d'en modifier le code. Les gènes principal et secondaire sont ainsi remplacés par des gènes générés aléatoirement (entre 1 et 10 dans notre cas, différents des 2 gènes antérieurs). On peut ainsi réinjecter dans le code génétique de l'individu, des gènes ayant disparu suite à une convergence prématurée de l'algorithme. Cette procédure nécessite la génération de 3 nombres aléatoires, l'un pour sélectionner le gène mutant, le deuxième et le troisième pour générer les gènes principal et secondaire, qui doivent être différents.

BIBLIOGRAPHIE

1. J. M. Alliot, Techniques d'optimisation stochastique appliquées aux problèmes du trafic aérien, Habilitation à Diriger les Recherches, INPT, 28 Mai 1996.
2. P. Baudet, C. Azzaro-Pantel, S. Domenech et L. Pibouleau, *A discrete-event simulation model for batch chemical plant scheduling*, ADEDOPS Workshop (workshop on analysis and design of event-driven operations in process systems), Imperial College, Londres, 10-11 avril 1995.
3. P. Baudet, C. Azzaro-Pantel, S. Domenech, L. Pibouleau, *Un modèle de simulation à événements discrets pour la gestion de production d'un atelier de chimie fine*, Colloque modélisation des systèmes réactifs, AFCET, Brest, 28-29 mars, publié dans les actes du congrès, 1996, p. 211-221.
4. P. Baudet, Ordonnancement à court terme d'un atelier discontinu de chimie fine ; cas du fonctionnement job-shop, thèse de Doctorat, INP Toulouse, 1997.
5. G. Bel et D. Dubois, Modélisation et simulation des systèmes automatisés, *RAIRO APII*, 1985, *19*, p. 42-52.
6. J. E. Biegel et J. J. Davern, Genetic algorithms and job-shop scheduling, *Comp. Ind. Engng.*, 1990, *19*, p. 81-91.

7. D. B. Birewar et I. E. Grossman, Simultaneous production planning and scheduling in multiproduct batch plants, *Ind. Eng. Chem. Res.*, 1990, *29*, p. 570-580.
8. J. L. Blanton et R. L. wainwright, *Multiple vehicle routing with time and capacity constraints using genetic algorithms*, Proc. Of the Fifth Int. Conf. On Genetic Algorithms, 1993, p. 452-459.
9. A. Brindle, Genetic Algorithms for function optimization, Unpublished doctoral dissertation, University of Alberta, Edmonton, 1981.
10. M. Cartwright et A. Long, Simultaneous optimization of chemical flowshop sequencing and topology using Genetic algorithms, *Ind. Eng. Chem. Res.*, 1993, *32*, p. 2706-2713.
11. M. Cartwright et A. Tuson, *Genetic Algorithms and Flowshop Scheduling: Towards the Development of a Real-time Process Control System*, AISB workshop on Evolutionary Computing, Leeds (GB), Avril 1994.
12. C. Caux, H. Pierreval, M. C. Portmann, *Les algorithmes génétiques et leur application aux problèmes d'ordonnancement*, Journées d'études « ordonnancement et entreprise », Toulouse, Juin 1994.
13. C. Caux, G. fleury, M. Gourgand et P. Kellert, Couplage méthodes d'ordonnancement-simulation pour l'ordonnancement de systèmes industriels de traitement de surfaces, *RAIRO Oper. Res.*, 1995, *29*, p. 391-413.
14. R. Cerf, Une théorie asymptotique pour les algorithmes génétiques, Thèse, Montpellier 1994.
15. H. Das, P. T. Cummings et M. D. Le Van, Scheduling of serial multiproduct batch processes *via* simulated annealing, *Comp. Chem. Engng.*, 1990, *14*, p. 1351-1362.
16. L. Davis, *Job-shop scheduling with Genetic algorithms*, Proc. 1th Conf. on Gas and their applications, Lawrence Erlbaum, Hillsdale, 1985, p. 136-140.
17. L. Davis, Handbook of Genetic algorithms, Van Nostrand Rienhold, Ed., New York, 1991.
18. K. A. De Jong, An analsis of the behavior of a class of genetic adaptative systems, Doctoral Dissertation, University of Michigan), Dissertation Abstract International, *36*, p. 5140B.
19. L. Djerid et M. C. Portmann, *Comment entrecroiser des procédures par séparation et évaluation et des algorithmes génétiques: Application à des problèmes d'ordonnancement à contraintes disjonctives,* Francoro, Mons, Juin 1995.
20. U. M. Egli et D. W. T. Rippin, Short term scheduling for multiproduct batch chemical plants, *Comput. Chem. Engng.*, 1986, *10*, p. 303-325.
21. A. E. Eiben, E. H. L. Aarts et K. M. Van Lee, *Global convergence of genetics algorithms: a markov chain analysis*, Proc. of the 1st workshop Ppsni, Dortmund, 1991.
22. A. Espuna et L. Puigjaner, On the solution of the retrofitting problem for multi-product batch semi-continuous chemical plants, *Comput. Chem. Engng.*, 1989, *13*, p. 483-490.
23. E. Falkenauer et S. Bouffouix, *A genetic algorithm for job-shop*, Proceedings of the IEEE Int. Conf. on Robotics and Automation, Sacramento, 1991, p. 824-829.
24. G. Fleury, *Application du recuit simulé et de ses variantes à des problèmes d'ordonnancement*, Journées d'études « ordonnancement et entreprise », Toulouse, Juin 1994.
25. A. M. Gillies, Machine learning procedures for generating image domain feature detectors, Unpublished doctoral dissertation, University of Michigan, Ann Arbor, 1985.
26. F. Glover, Future paths for integer programming and links to artificial intelligence, *Compu. and Oper. Res.*, 1986, *13*, p. 533-549.

27. D. E. GOLDBERG, Genetics algorithms in search, optimization and machine learning, Addison Wesley, Ed., 1989.
28. D. E. GOLDBERG, Les algorithmes génétiques, Addison Wesley, Ed., 1994.
29. Gotha, *Les problèmes d'ordonnancement*, RAIRO Recherche opérationnelle, 1993, *27*, p. 77-150.
30. L. HERAULT, *Réseaux neuromimétiques pour les problèmes d'ordonnancement - application à l'allocation de ressources,* Journées d'études « ordonnancement et entreprise », Toulouse, Juin 1994.
31. J. HERTZ, A. KROGH et R. PALMER, Introduction to the theory of neural computation, Addison Wesley, 1991.
32. M. HOFMEISTER, L. HALASZ et D. W. T. RIPPIN, Knowledge-based tools for batch processing systems, Third Conference on Process Systems Engineering, PSE'88, *Comput. Chem. Engng.*, 1989, *13*, p.1255-1261.
33. J. HOLLAND, Adaptation in natural and artificial systems, Mit. Press, Cambridge, Mass., 1975.
34. S. KIRKPATRICK, C. D. GELATT et M. P. VECCHI, *Optimization by simulated annealing*, Resp. Rep. R. C. 9335, IBM TJW, Center Yorktown, NY, 1982.
35. H. KU et I. A. KARIMI, Completion Time Algorithms for Serial Multiproduct Batch Processes with Shared Storage, *Comput. Chem. Engng.*, 1990, *14*, p. 49-69.
36. H. KU et I. A. KARIMI, Scheduling in serial multiproduct batch processes with due-date penalties, *Ind. Eng. Chem. Res.*, 1990, *29*, p. 580-590.
37. H. KU et I. A. KARIMI, An evaluation of simulated annealing for batch process scheduling, *Ind. Eng. Chem. Res.*, 1991, *30*, p. 163-169.
38. K. KURIYAN, G. V. REKLAITIS et G. S. JOGLEKAR, Multiproduct plant scheduling studies using BOSS, *Ind. Eng. Chem. Res.*, 1987, *26*, p. 1551-1558.
39. K. KURIYAN et G. V. REKLAITIS, Scheduling network flowshops so as to minimize makespan, *Comput. Chem. Engng.*, 1989, *13*, p. 187-200.
40. P. J. M. LAARHOVEN, E. H. L. AARTS et J. K. LENSTRA, Job-shop scheduling by silulated annealing, *Oper. Res.*, 1992, *40*, p. 113.
41. G. LAZAROS et C. PANTELIDES, Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 1. Mathematical Formulation, *Ind. Eng. Chem Res.*, 1996, *35*, p. 488-509.
42. R. MUSIER et L. EVANS, An approximative method for the production scheduling of industrial batch processes with parallel units, *Comput. Chem. Engng.*, 1989, *13*, p. 229-238.
43. J. P. NADAL, Réseaux de neurones, de la physique à la psychologie, Éditions Armand Colin, 1993.
44. R. NAKANO et T. YAMADA, *Conventional Genetic algorithm for job-shop problems*, Proc. 4th Int. Conf. on Gas, Kaufmann Ed., San Mateo, California, 1991, p. 474-479.
45. E. PEYROL, P. FLOQUET, L. PIBOULEAU et S. DOMENECH, Scheduling and simulated annealing. Application to a semiconductor circuit fabrication plant, *Comput. Chem. Engng.*, 1993, *17*, p. S39-44.
46. M. C. PORTMANN, Scheduling methodology: Optimization and compu-search approaches I, Production and scheduling of manufacturing systems, Artiba A., Elmaghraby S.E., Eds., Chapman & Hall, 1996.
47. M. C. PORTMANN, *Various genetic algorithm approaches for solving scheduling problem families*, JETAI 96, Pavia, 21-23 mars 1996.
48. M. C. PORTMANN et F. GHEDJATI, *Méthodes approchées pour le problème d'ordonnancement avec machines non identiques en parallèles et contraintes de précédence*, Proceedings of AGI'94, Poitiers, Juin 1994.

49. D. RAJAGOLAPAN et I. A. KARIMI, Completion times in a serial mixed-storage multiproduct processes with transfer and set-up times, *Comput. Chem. Engng.*, 1989, *13*, p. 175-186.

50. G. V. REKLAITIS, *Overview on scheduling and planning of process operations*, NATO Advanced Study Institute on Batch Processing Systems Engineering, Antalya, Turkey, 1992.

51. D. W. T. RIPPIN, Batch process systems engineering: A retrospective and prospective review, *Comput. Chem. Engng.*, 1993, *17*, p. S1-S536.

52. D. SMITH, *Bin packing with adaptative search*, Proc. of the First Int. Conf. on Genetic Algorithms, 1985, p. 202-207.

53. G. TAGLIARINI, J. CHRIST et E. PAGE, Optimization using neural network, *IEEE Trans. Comp.*, 1991, *40*, p. 1347-1358.

54. M. TANDON, P. T. CUMMINGS et M. D. LE VAN, Scheduling of Multiple Products on Parallel Units with Tardiness Penalties using Simulated Annealing, *Comput. Chem Engng.*, 1995, *19*, p. 1069-1076.

55. A. G. TSIRUKIS et G. REKLAITIS, Feature extraction algorithms for constrained global optimisation. 1. Mathematical foundation, *Ann. Oper. Res.*, 1993, *42*, p. 229-273.

56. A. G. TSIRUKIS et G. REKLAITIS, Feature extraction algorithms for constrained global optimisation. 2. Batch process scheduling application, *Ann. Oper. Res.*, 1993, *42*, p. 225-312.

57. L. TUSON, *The implementation of a Genetic algorithm for the scheduling and Topology optimisation of chemical flowshops,* TRGA94-01, Oxford (UK), Juin 1994.

58. M. TSUYOSHI, M. KATSUMI et N. NOBUTO, Neural Network Approach for Minimizing the Makespan of the General Job-shop, *Internat. J. Produc. Econom.*, 1994, *33*, p. 67-74.

59. M. C. WELLONS et G. V. REKLAITIS, Optimal Schedule for a Single-Product Production Line-I. Problem, *Comput. Chem. Engng.*, 1989, *13*, p. 201- 212.

60. W. WIEDE, K. KURIYAN et G. V. REKLAITIS, Determination of completion times for serial multiproduct processes. 1. A two unit finite intermediate storage system, *Comput. Chem. Engng.*, 1987, *11*, p. 337-344.

61. W. WIEDE, K. KURIYAN et G. V. REKLAITIS, Determination of completion times for serial multiproduct processes. 2. A multi-unit finite intermediate storage system, *Comput. Chem. Engng.*, 1987, *11*, p. 345-356.

62. W. WIEDE, K. KURIYAN et G. V. REKLAITIS, Determination of completion times for serial multiproduct processes. 3. Mixed intermediate storage system, *Comp. Chem. Engng.*, 1987, *11*, p. 357-368.

63. A. YOUNG et G. V. REKLAITIS, Capacity expansion study of a batch production line, *Ind. Eng. Chem. Res.*, 1989, *28*, p. 772-777.