

ALGORITHME GENETIQUE ET MODELE DE SIMULATION POUR L'ORDONNANCEMENT D'UN ATELIER DISCONTINU DE CHIMIE

P. Baudet, C. Azzaro-Pantel, S. Domenech et L. Pibouleau

Laboratoire de Génie Chimique - URA 192 du CNRS

ENSIGC INPT

18, Chemin de la Loge - 31078 Toulouse Cedex - FRANCE

RESUME

Nous traitons dans cet article le problème de l'ordonnancement d'un atelier discontinu de chimie de type "Job-Shop". L'algorithme développé est un algorithme génétique que l'on a couplé à un modèle de simulation à événements discrets utilisé comme procédure d'évaluation dans la boucle d'optimisation globale. Le critère à optimiser fait intervenir le temps de cycle moyen des produits élaborés et le nombre de produits en-cours en fin de campagne.

I. POSITION DU PROBLEME

Dans le domaine de la chimie fine, les problèmes d'ordonnancement sont le plus souvent abordés par des méthodes utilisant la programmation en variables mixtes, linéaire ou non (Mixed Integer Linear Programming, Mixed Integer Non Linear Programming [(1),(2)]). L'analyse de la bibliographie montre que ces méthodes s'avèrent difficiles à mettre en oeuvre pour une modélisation assez fine du système de production et pour le traitement de problèmes de taille réelle qui présentent un caractère combinatoire très marqué. Nous avons donc utilisé une approche différente décomposant le problème général de l'ordonnancement en deux parties.

La première est une phase de modélisation utilisant la simulation par événements discrets, qui présente une grande souplesse dans les possibilités de description des caractéristiques et contraintes spécifiques au domaine chimique. Dans la seconde phase, le modèle est couplé à un algorithme génétique développé dans notre laboratoire, particulièrement adapté à la résolution de problèmes combinatoires. Notre étude a porté sur l'optimisation de l'ordre de lancement des produits dans l'atelier afin de satisfaire un critère choisi qui fait intervenir le temps de cycle moyen des produits et le nombre de produits non achevés à la fin de la campagne.

II. LE MODELE DE SIMULATION

Nous présentons ici les caractéristiques et possibilités principales du modèle de simulation développé dans notre laboratoire, qui a déjà été détaillé dans un article précédent (3). Basé sur une approche par événements, le modèle de simulation à événements discrets décrit complètement le fonctionnement dynamique du système productif à travers les événements qui s'y produisent. Une analyse précise du fonctionnement d'un site de production discontinu de chimie nous a permis de répertorier l'ensemble des événements occurrents, les types de ressources disponibles et les contraintes principales liées à l'occurrence de ces événements. Les ressources ont été modélisées comme des machines à états évoluant de façon interdépendante d'un état à un autre, en fonction des événements du système. L'évolution dynamique du système est réalisée par l'occurrence successive de différents événements et de leur logique de changement d'état. La progression du temps a lieu par saut d'événements jusqu'à la fin de la simulation. Le modèle tient compte des principales ressources critiques du système productif (équipements, opérateurs, bacs de stockage, matières premières) et de diverses contraintes de disponibilité sur ces ressources (approvisionnements en matières premières, opérations de maintenance sur les équipements, congés des opérateurs,...). Les recyclages sur site ont également été modélisées ainsi que les différentes contraintes liées au stockage, qu'elles soient spatiales (dues au nombre et à la capacité limités des bacs de stockage

pour intermédiaires réactionnels ou sous-produits) ou temporelles (dues à la présence possible d'intermédiaires instables ou stables en temps limité). Les produits à élaborer ou à recycler sont définis par une recette qui définit précisément chacune des opérations du procédé. On fixe en début de simulation une quantité totale à produire (ou à recycler) pendant la campagne, qui sera divisée en différents lots, de volumes non nécessairement identiques, appelés communément des "batches". La gestion des conflits est assurée par l'implantation d'une base de règles heuristiques, au choix de l'utilisateur, qui détermine l'ordre d'exécution des tâches concurrentes à une date donnée. Le modèle développé permet donc de simuler la campagne de réalisation de la totalité des batches dans l'atelier discontinu de chimie, soumis à la politique de gestion des conflits et à ses contraintes spécifiques. Les résultats du modèle sont nombreux pour un suivi de production convenable (statistiques sur l'activité des divers éléments du système, suivi graphique des différents stocks de produits, diagramme de Gantt...) et contribuent à une gestion efficace de la production. Des résultats techniques supplémentaires, tels le temps de cycle, le temps total d'attente, la date de production des produits, ou le nombre de produits en-cours peuvent être utilisés comme mesures de performance du système.

III. OPTIMISATION DE L'ORDRE DE LANCEMENT DES PRODUITS

Dans cette étude, nous avons choisi d'optimiser l'ordre de lancement des produits dans l'atelier de façon à minimiser une fonction objectif faisant intervenir le temps de cycle moyen des produits dans l'atelier et le nombre de produits en-cours en fin de campagne. La procédure globale d'optimisation est un algorithme génétique qui utilise le modèle de simulation comme fonction d'évaluation pour chaque solution testée.

IV. LES ALGORITHMES GENETIQUES

Les algorithmes génétiques, développés par John Holland puis Goldberg (4), présentent des qualités intéressantes pour la résolution de problèmes combinatoires complexes. Ils sont basés sur la théorie de l'évolution des espèces dans leur milieu naturel, soit une transposition artificielle des concepts basiques de la génétique et des lois de survie énoncées par Darwin.

Rappelons que la génétique représente un individu par un code, c'est-à-dire un ensemble de données (appelées chromosomes), identifiant complètement l'individu. La reproduction est dans ce domaine, un mixage aléatoire des chromosomes de deux individus, donnant naissance à des individus enfants ayant une empreinte génétique nouvelle, héritée des parents. L'évolution "normale" d'une espèce, d'une génération à la suivante est alors une simple recombinaison obtenue par croisement aléatoire des chromosomes d'individus appariés. La mutation génétique est caractérisée dans le code génétique d'un individu enfant par l'apparition d'un chromosome nouveau, inexistant chez les individus parents. Ce phénomène génétique d'apparition de "mutants" est rare mais permet d'expliquer les changements dans la morphologie des espèces, toujours dans le sens d'une meilleure adaptation au milieu naturel. La disparition de certaines espèces est expliquée par les "lois de survie", selon lesquelles seuls les individus les mieux adaptés à leur milieu naturel auront une longévité suffisante pour générer une descendance. Les individus peu adaptés auront tendance à disparaître. C'est une "sélection naturelle", qui conduit de génération en génération à une population composée d'individus de plus en plus adaptés, de plus en plus "forts", en somme.

Un algorithme génétique est construit de manière tout à fait analogue. Dans l'ensemble des solutions d'un problème combinatoire, une population de taille N est constituée de N solutions (les individus de la population) convenablement marquées par un codage qui les identifie complètement. Une procédure d'évaluation est nécessaire à la détermination de la "force" de chacun des individus de la population. Viennent ensuite une phase de sélection (simulant artificiellement les lois de survie de Darwin en sélectionnant les individus au prorata de leur force) et une phase de recombinaison (appariement aléatoire des individus, opérateurs artificiels de croisement et de mutation) qui génèrent une nouvelle population d'individus, qui ont de

"bonnes chances" d'être plus "forts" que ceux de la génération précédente. De génération en génération, la force des individus de la population augmente et après un certain nombre d'itérations, la population est constituée d'individus tous très forts, soit de solutions quasi-optimales du problème combinatoire posé.

V. LA MISE EN OEUVRE DE L'ALGORITHME GENETIQUE

La réalisation de l'algorithme génétique consiste à développer ces différentes phases:

V.1 Le codage

Le codage que nous avons retenu est très simple, de longueur égale au nombre total de batches (NB) à réaliser pendant la campagne. Nous dirons alors que le code génétique d'une solution envisagée est constituée de NB chromosomes. La figure 1 présente le codage d'une solution quelconque. Le batch numéro 4 sera lancé en production le premier, suivi du batch numéro 7...et enfin le batch numéro 2, placé dans le dernier chromosome du code, sera lancé le dernier. Chacun des NB! ordres de lancement possibles peut ainsi être spécifiquement codé et identifié.

La population initiale sera composée de N individus générés aléatoirement (N représente la taille de la population et a été fixée à 30).

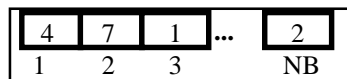


Figure 1: Codage d'une solution

V.2 La procédure d'évaluation

Pour l'évaluation de chaque individu, le modèle utilise son codage, exécute la simulation avec l'ordre de lancement des produits correspondant et retourne les valeurs du temps de cycle moyen des produits (TCmoy) et du nombre de produits en-cours en fin de campagne (NENC). Un algorithme génétique ne peut que maximiser une performance (population de plus en plus "forte") alors que nous cherchons ici à minimiser simultanément deux critères. La force des individus doit donc être adaptée aux algorithmes génétiques. Nous avons choisi pour le calcul de la force f_i de l'individu i , la fonction objectif suivante:

$$f_i = 1 / [(TCmoy)^{NENC+1}]$$

V.3 La procédure de sélection

La procédure de sélection que nous avons utilisée est la "roulette biaisée de Goldberg"(4) et fonctionne sur le principe suivant :

Chaque individu i est représenté par sa force relative f_i telle que:

$$f_i^{rel} = f_i / \sum_{i=1,N} f_i \quad ; \quad \sum_{i=1,N} f_i^{rel} = 1$$

On procède alors à la sélection de N individus survivants en effectuant N tirages aléatoires dans le domaine [0,1]. Soit R le nombre aléatoire tiré, l'individu j sélectionné sera tel que:

$$\sum_{i=1,J} f_i^{rel} \geq R$$

(Selon cette procédure de sélection, plus un individu est fort, plus il aura des chances de survivre).

V.4 Le croisement

Les individus survivants à la phase de sélection sont appariés aléatoirement et chaque paire formée va subir le croisement avec une probabilité P_{cross} (les paires croisées sont remplacées

dans la population par leur descendance). De nombreux types de croisement différents existent dans la littérature (5), préservant plus ou moins l'identité génétique des parents et permettent de se déplacer dans tout l'espace des solutions. Plus la probabilité de croisement est grande, plus la convergence de l'algorithme est rapide (mais plus on risque de converger vers un optimum local). Nous avons utilisé le croisement MPX par lequel le codage des fils présente une forte analogie avec celui des parents. Ce croisement conserve dans fils i la zone interne de parent i (zone comprise entre $N1$ et $N2$ tirés aléatoirement). Ensuite, pour les cases vides restantes, le k -ième chromosome de fils i sera rempli avec le k -ième de parent j s'il ne provoque pas de doublon. A défaut, on essaiera le k -ième chromosome de parent i ou enfin le premier chromosome de la zone interne de parent j qui ne provoque pas de doublon. Ce croisement est illustré sur la figure 2 (NB=12).

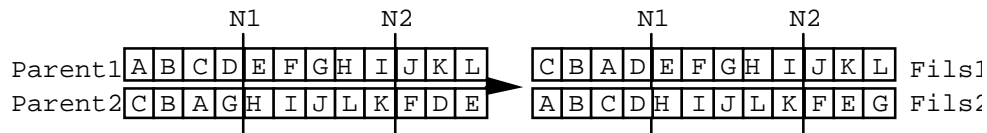


Figure 2 : Croisement MPX

V.5 La mutation

Les individus de la population issue du croisement vont ensuite subir un processus de mutation avec une probabilité P_{mut} . Comme pour les croisements, de nombreuses méthodes de mutation ont été développées dans la littérature du domaine. Nous présentons ici le plus simple qui consiste à inverser le locus de deux chromosomes choisis aléatoirement. Cette méthode est illustrée sur la figure 3.



Figure 3: Mutation par permutation de deux locus

La mutation évite la dégénérescence de la population (en d'autres termes, elle permet de quitter les extrémas locaux). La probabilité de mutation doit cependant rester faible pour que l'on puisse assurer la convergence de l'algorithme génétique. Elle est généralement fixée autour de 5%, valeur adoptée dans notre procédure d'optimisation.

L'algorithme génétique évolue ainsi de phase en phase et les individus de la population deviennent de génération en génération de meilleures solutions du problème de l'ordonnancement. Notons que si la mise en oeuvre de l'algorithme génétique est très simple et ne nécessite aucune hypothèse ou information sur le système optimisé (pas de calcul de gradient par exemple), le réglage des paramètres (taille de la population, mode et probabilité de croisement et de mutation, nombre total de générations) influe fortement sur la convergence de l'algorithme.

VI. EXEMPLE

Cette procédure d'optimisation a été utilisée sur un exemple de campagne de production comprenant 30 batches de produits finaux. L'atelier est soumis à de nombreuses contraintes (approvisionnements en matières premières, plusieurs arrêts maintenance pour équipements, e.t.c). La taille de la population a été fixée à 30, et le nombre de générations à 500 (soit 1500 évaluations du critère parmi $30! = 2,6 \cdot 10^{32}$ solutions possibles). Nous avons utilisé le croisement MPX, avec une probabilité de croisement égale à 0,8. La mutation utilisée est la permutation aléatoire du locus de deux chromosomes, avec une probabilité de 0,1. Une "niche" a été mise en place pour sauvegarder la meilleure solution obtenue pendant l'exécution.

La meilleure solution obtenue, après 427 générations conduit à un temps de cycle moyen TCM égal à 1787 unités de temps et à un nombre d'en-cours final NENC nul.

Une simulation effectuée avec le modèle seul, où l'ordre de lancement dépend des heuristiques appliquées (les mêmes que dans l'optimisation précédente) conduit à un temps de cycle moyen TCM égal à 2145 unités de temps et à un en-cours final égal à 2.

On remarque que pour un nombre d'appels au modèle de simulation relativement restreint (500), on obtient une augmentation significative des performances du système productif.

VII. CONCLUSIONS- PERSPECTIVES

La procédure d'optimisation que nous avons développée est simple à mettre à oeuvre et très efficace pour la génération de plans de production en adéquation avec un critère technique choisi. Le nombre d'appels à la procédure d'évaluation est relativement faible par rapport à l'espace total du domaine des solutions, ce qui rend cette méthode intéressante au niveau du temps de calcul.

Un algorithme génétique peut permettre de coder plusieurs paramètres en même temps et nous envisageons dans la suite de nos travaux d'optimiser simultanément l'ordre de lancement des produits et les heuristiques appliquées sur site. Nous allons par ailleurs développer une procédure à apprentissage pour une détermination automatique des paramètres nécessaires à l'optimisation (type et probabilité de croisement et de mutation notamment).

REFERENCES

- [1] Birewar D.B. and I.E Grossman, Efficient Optimization Algorithms for Zero-Wait of Multiproduct Batch Plants, *Ind. Eng. Chem. Res.*, 1989, 28, 1333-1345.
- [2] Sahinidis N.V. and Grossman I.E., MINLP model for cyclic multiproduct scheduling on continuous parallel lines, *Computers chem. Eng.*, Vol.15, No 2, pp. 85-103, 1991.
- [3] - Baudet P., Azzaro-Pantel C., Domenech S., Pibouleau L., A discrete-event simulation model for batch chemical plant scheduling, ADEDOPS Workshop, Imperial College, Londres, 10-11 Avril 1995
- [4] Goldberg D., Algorithmes génétiques, *Adisson Wesley*, France, 1994.
- [5] Caux C., Pierreval H., Portmann MC., les algorithmes génétiques et leur application aux problèmes d'ordonnancement, *Congrès d'Ordonnancement* ,Toulouse, France, Juin 1994.