



HAL
open science

Optimisation des plans de charge pour un flow-shop dans le cadre d'une production en Juste A Temps : 3- Algorithmes génétiques et approche multi - critères

Emmanuel Vidal, Philippe Duquenne, Hervé Pingaud

► To cite this version:

Emmanuel Vidal, Philippe Duquenne, Hervé Pingaud. Optimisation des plans de charge pour un flow-shop dans le cadre d'une production en Juste A Temps : 3- Algorithmes génétiques et approche multi - critères. 3ème Congrès Franco-Quebequois de Génie Industriel, 1999, Montréal, Canada. pp.1-11. hal-02356246

HAL Id: hal-02356246

<https://hal.science/hal-02356246>

Submitted on 8 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.






Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/20929>

To cite this version:

Vidal, Emmanuel  and Duquenne, Philippe  and Pingaud, Hervé 
Optimisation des plans de charge pour un flow-shop dans le cadre d'une production en Juste A Temps : 3- Algorithmes génétiques et approche multi-critères. (1999) In: 3ème Congrès Franco-Quebequois de Génie Industriel, 1999 - 1999 (Montréal, Canada). (Unpublished)

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Présentation du problème

Description du système :

Le but est de calculer l'ordonnancement d'un atelier de type flow-shop monoproduit, dans lequel chaque station de travail possède les caractéristiques suivantes :

- un ensemble d'opérateurs, correspondant le cas échéant à plusieurs types de compétence, travaille simultanément sur la station de travail ;
- chaque station est caractérisée par la charge de travail (= volume d'heures de travail) par compétence correspondant au passage d'un produit ;
- selon les cas, une station peut effectuer simultanément un ou plusieurs traitements parallèles ;
- la relation entre la charge de travail et l'effectif nécessaire est la suivante :
$$\text{charge} = \text{cycle} * \text{effectif}$$
où le cycle désigne la durée opératoire affectée à la station ;
- le cycle de chaque poste doit être compris dans un intervalle dont la borne inférieure est déterminée par le chemin critique des tâches à effectuer et dont la borne supérieure est établie selon des considérations organisationnelles (taille minimale d'une équipe, par exemple).

Explicitation de la demande :

Le calendrier de livraison est connu *a priori* et peut être caractérisé par :

- l'intervalle de tolérance à l'intérieur duquel peut avoir lieu la livraison ;
- les pénalités de retards encourues en cas de dépassement de la borne supérieure du créneau de livraison ;
- les frais de stockage du produit fini, si celui-ci est terminé avant le début de l'intervalle de livraison.

Objectifs du calcul :

On cherche donc à déterminer les valeurs optimales pour les variables suivantes :

- dates de livraison (à l'intérieur de l'intervalle de tolérance) ;
- cycles de production par poste et par produit ;
- la durée des éventuelles périodes de stockage ;
- horaires de travail (dans le cadre de la modulation d'horaires).

Dans un premier temps, la résolution du problème a suivi un processus séquentiel, issue de la décomposition du problème en une série de sous-problèmes, pour lesquels chaque optimisation porte sur un type homogène de variables de décisions liées à l'organisation (par exemple, valeurs des cycles de production, horaires de travail, etc.).

Afin d'étudier la position des solutions ainsi obtenues par rapport à la solution optimale globale, une formulation s'appuyant sur une représentation discrète et non-uniforme du temps a été établie, permettant d'obtenir la solution optimale (cf. communication : « Optimisation des plans de charge pour un flow shop dans le cadre d'une production en Juste A Temps : Formulation mathématique », Philippe Duquenne, Hervé Pingaud, Emmanuel Vidal).

L'utilisation d'algorithmes génétiques mono - critère permet de résoudre de manière séquentielle les différents sous - problèmes, chaque solution d'un sous problème permettant de figer les valeurs d'un groupe homogène de variables de décision pour la résolution du sous problème suivant. De plus, le recours à des algorithmes génétiques multi - critères rend possible l'obtention de solutions satisfaisantes au regard des différents critères, ainsi que la détermination de la surface de Pareto (dont la définition est donnée ci-après).

Les algorithmes génétiques

Principe de la méthode

Les algorithmes génétiques reposent sur des concepts inspirés de la biologie et visent à reproduire l'évolution d'une population dans un milieu hostile ([2], [3]). Ils constituent un outil intéressant dans le cadre de la résolution de problèmes d'optimisation combinatoire, bien que la théorie mathématique associée (théorie des schémas) soit encore limitée face à la complexité inhérente aux problèmes traités.

Alors que la plupart des méthodes d'optimisation combinatoire s'attachent à rechercher la solution optimale à un problème donné en faisant évoluer un seul point à chaque itération, les algorithmes génétiques travaillent sur une

population de chromosomes qui évoluent au cours des itérations. Un chromosome représente en fait un point de l'espace d'état, auquel est associée une valeur (également désignée sous le nom d'adaptation) permettant de situer ce chromosome (ou individu) dans l'espace objectif. Le codage d'un chromosome dépend de la modélisation retenue ; chaque chromosome est composé d'un nombre déterminé de gènes, dont chacun correspond à une variable et auquel on associe une valeur qui évoluera au cours du calcul.

Lorsque le principe de codage des chromosomes est défini, un algorithme génétique se déroule selon le schéma suivant :

1. génération aléatoire de la population initiale, afin d'assurer la plus grande homogénéité des points dans l'espace d'état ;
2. évaluation des chromosomes afin d'associer à chacun d'eux une valeur d'adaptation (ou plusieurs dans le cas de l'optimisation multi-objectifs) ;
3. sélection des individus pour la reproduction ;
4. croisement des individus sélectionnés selon une certaine probabilité (probabilité de croisement) ;
5. mutation des chromosomes (selon la probabilité de mutation) ;
6. retour à l'étape 2 tant que le critère d'arrêt n'est pas atteint.

Chaque séquence constituée des étapes 2 à 5 est appelée une génération. Le nombre de générations à effectuer est un critère d'arrêt fréquemment utilisé.

Par la suite, on supposera que l'on considère un problème de maximisation dans un ensemble de solution à valeurs positives, sachant que l'on peut ramener toute fonction à ce type de caractéristique.

Le croisement a pour objectif d'augmenter la diversité de la population en manipulant la structure des chromosomes. Nous nous cantonnons ici au principe usuel de croisement à découpage de chromosomes (*slicing over*) appliqué à deux parents et permettant d'obtenir deux enfants.

L'opérateur de mutation confère aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace : l'algorithme est ainsi susceptible d'atteindre tous les points de l'espace d'état. De même que pour le croisement, une probabilité est associée à cet opérateur ; les valeurs généralement conseillées sont de l'ordre de 0,5 à 0,8 pour la probabilité de croisement, et de $1/N$ (N = nombre de chromosomes) à 0,1 pour la probabilité de mutation.

Les algorithmes génétiques dans les problèmes d'ordonnement [1]

La complexité élevée des problèmes d'ordonnement a conduit à utiliser les algorithmes génétiques pour leur résolution ; cependant, ces problèmes présentent des caractéristiques et des contraintes particulières en ce qui concerne l'ordre des permutations et le respect des contraintes associées lors d'opérations sur les chromosomes. Pour mémoire, on peut citer les croisements suivants, développés spécifiquement pour les problèmes d'ordonnement :

- croisement PMX (*Partially Mapped Crossover*) : application aux problèmes de job-shop et de voyageur de commerce ;
- croisement MPX (*Maximal Preservative Crossover*) : application au problème du voyageur de commerce ;
- croisement par recombinaison d'arêtes : application au problème du voyageur de commerce symétrique ;
- croisement d'ordre (à deux points, à deux points linéaires, uniforme) ;
- croisement de cycles : application au problème du voyageur de commerce ;
- croisement simple à un 1 point (à k points) pour les permutations.

Les applications pratiques portent sur plusieurs types de problèmes, et notamment :

- problème de voyageur de commerce, qui existent dans le monde industriel sous la forme de minimisation du déplacement d'un outil ou du temps de changement d'un outil ;
- problèmes de job-shop
- ordonnancement de lots dans un flow shop (int J of Prod res C. Jordan).

Cependant, les développements théoriques sur la convergence des algorithmes génétiques restent limités ; ils permettent toutefois d'obtenir des bonnes performances dans un grand nombre de cas, et notamment lorsque le codage et les opérateurs génétiques adaptés aux connaissances du problème sont utilisés.

Codage du problème

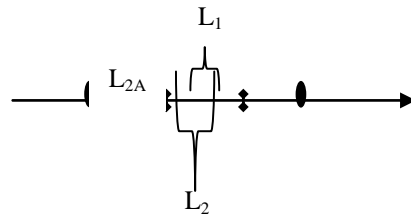
Dans le cadre du problème étudié, les gènes des chromosomes représentent les valeurs :

- des dates de livraison, dans le cas de la détermination de la date optimale ;
- des cycles de chaque poste, dans le cadre de la modulation de cycle ;

Il existe pour chacune de ces valeurs des contraintes sur lesquelles nous reviendrons.

La date de livraison d'un OF L_i ¹ est comprise entre des bornes que l'on peut définir comme suit (chaque intervalle L_1 et L_2 étant associé à un OF i , l'indice étant omis par souci de clarté) :

- intervalle de livraison L_1 souhaité par le client, à l'intérieur duquel devra avoir lieu la livraison sous peine de pénalités ;
- intervalle L_2 ($L_1 \subseteq L_2$) souhaité par l'industriel.



L'existence de l'intervalle L_2 peut être interprétée comme suit :

- si $L_i \in L_{2A}$, l'OF sera terminé avant que le client n'en prenne livraison, il y aura donc une période de stockage du produit fini, et par conséquent des frais financiers liés à l'immobilisation du capital ;
- si $L_i \in L_1$, l'industriel aura satisfait les souhaits du client ;
- si $L_i \in L_{2B}$, la livraison aura lieu en retard et des pénalités proportionnelles au retard seront dues au client.

Pour toute solution au problème de détermination des dates de livraisons, on devra donc avoir $L_i \in L_2$.

Le cycle d'un OF i sur un poste j est désigné par $c_{i,j}$. Le problème de détermination des cycles peut être envisagé selon deux possibilités :

- choix entre bornes : les cycles autorisés peuvent être choisis de manière continue entre deux bornes minimum et maximum ;
- choix dans liste : les cycles autorisés sont discrétisés et doivent donc être choisis dans une liste prédéterminée.

A partir des valeurs des cycles et des dates de livraison, on calcule :

1. la planification au plus tard des ordres de fabrication, en pénalisant (au niveau de la fonction objectif) les solutions qui ne respectent pas les capacités des postes ou en créant des tampons de stockage pour respecter ces capacités selon les cas (voir ci-dessous : AG mono-objectifs) ;
2. le plan de charge des effectifs, c'est à dire la courbe de besoin des effectifs en fonction du temps ;
3. les différents indicateurs relatifs à la solution, dont certains peuvent tenir le rôle de fonction objectif.

La prise en compte de contraintes, notamment celles liées aux intervalles de tolérance des dates de livraison, aux cycles autorisés ainsi qu'aux liens logiques entre tâches, est assurée par l'intermédiaire de pénalisations associées à la violation de ces contraintes : la fonction objectif est pénalisée (càd majorée dans le cas d'un problème de minimisation, et réciproquement) d'une quantité proportionnelle à la violation de la contrainte. La valeur du coefficient de pénalisation a dans ces conditions des conséquences notables sur la qualité des solutions obtenues (cf. [1]).

On utilise également des algorithmes de réparation (cf. [1]) qui rendent admissibles une solution non admissible générée par un opérateur génétique (par exemple, rétablissement des contraintes de précedence).

Approche séquentielle – Algorithmes génétiques mono – objectifs

Présentation des critères

Les différents critères utilisés pour évaluer les solutions sont les suivants (Cf. : « Optimisation des plans de charge pour un flow shop dans les cadre d'une production en Juste A Temps : Convergence entre besoins et capacité », Philippe Duquenne, Hervé Pingaud, Emmanuel Vidal) :

- perturbation du plan de charge : à minimiser.
- activité des opérateurs : à maximiser.
- coût global : à minimiser.

¹

Les indices i, j et k désignent respectivement les OF, les postes et les jours (unités de temps).

Leur expression mathématique sera présentée ci-après.

La résolution du problème est effectuée dans un premier temps selon une approche séquentielle, en décomposant le problème général en sous-problèmes de tailles réduites qui seront traités les uns après les autres. L'ordre des séquences dans cette démarche a été déterminé expérimentalement, et donne les meilleurs résultats. On procède alors comme suit :

- détermination des dates de livraison optimales ;
- détermination des cycles de production adaptés ;
- application de la modulation d'horaires ;
- application des compétences multiples.

Détermination des dates de livraison

La première étape de la résolution est la détermination de la date de livraison optimale pour chaque ordre de fabrication, à l'intérieur de l'intervalle de tolérance défini avec le client. Les valeurs des cycles sont alors des données du problème (égales pour chaque poste à la valeur minimale du cycle autorisé), elles ne seront donc pas modifiées au cours de cette étape. Le nombre de variables est alors égal au nombre d'ordre de fabrications à traiter.

On peut ainsi calculer la planification au plus tard en fonction des dates obtenues, tout en respectant les capacités de production, et en déduire le plan de charge des effectifs. A ce stade-là, on cherche à minimiser les perturbations du plan de charge, sans se préoccuper des valeurs de la fonction coût et du taux d'activité.

La démarche de calcul de la planification et du plan de charge des effectifs suit alors le pseudo-code suivant :

```
m = nombre d'OF
n = nombre de postes
Eff(i,j) = effectif sur le poste j pour l'OF i
E(k) = effectif total pour le jour k
T = durée de l'horizon de temps (en jours)

de i = m à 1 par pas de -1
  f(i,n) = L(i)
  de j = n à 2 par pas de -1 faire
    d(i,j) = f(i,j) - c(i,j)
    f(i,j-1) = d(i,j)
  fin faire
  d(i,1) = f(i,1) - c(i,1)
fin faire

de i = 1 à m faire
  de j = 1 à n faire
    de k = d(i,j) à f(i,j) faire
      E(k) = E(k) + Eff(i,j)
    fin faire
  fin faire
fin faire
```

On peut alors déterminer la valeur de la fonction d'évaluation F correspondant aux perturbations du plan de charge, puis la pénaliser si la solution proposée entraîne la violation de contraintes d'occupation des postes (càd si deux livraisons L_i et L_{i+1} sont prévues à un intervalle inférieur au plus long des cycles de l'OF i , M_i) :

```
de k=1, T-1 faire
  F = F + |E(k+1) - E(k)|
fin faire

de i = 1 à m-1
  si  $L_{i+1} - L_i < M_i$  alors
    F = F + K * ( $L_i - L_{i+1} + M_i$ )
  fin si
```

fin de

Le coefficient K mentionné ci-dessus est le coefficient de pénalisation ; il permet de dégrader les solutions violant cette contrainte, sans toutefois les éliminer totalement, car il est essentiel lors de l'utilisation d'algorithmes génétiques de préserver, au moins durant les premières générations, des solutions « décevantes » afin de préserver la diversité de la population.

La solution obtenue est alors utilisée pour le calcul des cycles associés à chaque poste et à chaque ordre de fabrication.

Modulation de cycles

On considère maintenant que les dates de livraison obtenues lors de l'étape précédente sont figées, et l'on fait varier la durée opératoire sur chaque poste et pour chaque ordre de fabrication. Le nombre de variables est alors égal au produit du nombre d'OF par le nombre de postes de travail.

On calcule tout d'abord la planification, en respectant cette fois les contraintes de capacité :

```
f(m, n) = L(n)
de j = n à 2 par pas de -1 faire
    d(m, j) = f(m, j) - c(m, j)
    f(m, j-1) = d(m, j)
fin faire
d(m, 1) = f(m, 1) - c(m, 1)
de i = m-1 à 1 par pas de -1
    f(i, n) = min(L(i), Li+1 - Mi)
    de j = n à 2 par pas de -1 faire
        d(i, j) = f(i, j) - c(i, j)
        f(i, j-1) = d(i, j)
    fin faire
    d(i, 1) = f(i, 1) - c(i, 1)
fin faire

de i = 1 à m faire
    de j = 1 à n faire
        de k = d(i, j) à f(i, j) faire
            E(k) = E(k) + Eff(i, j)
        fin faire
    fin faire
fin faire
```

On cherche alors à minimiser la fonction coût F, critère économique prenant en compte les frais liés aux stockages et aux en-cours, ainsi que le coût du travail, de l'inactivité et des perturbations liées à la modulation de cycles.

Ces deux étapes peuvent éventuellement être répétées plusieurs fois de suite afin d'obtenir une solution satisfaisante, voire stable, c'est à dire ne variant pas par l'application de l'algorithme génétique.

Résultats expérimentaux

Les graphiques ci-dessous présentent les résultats obtenus dans le cadre de la détermination des dates de livraison optimales, en prenant comme critère les perturbations du plan de charge. Ce critère doit bien sûr être minimiser ; on utilise alors la formule suivante afin de se situer dans le cadre d'un problème de maximisation :

$$F(x) = C_{\text{Max}} - C(x)$$

où est la valeur de la perturbation et C_{Max} un majorant de $C(x)$.

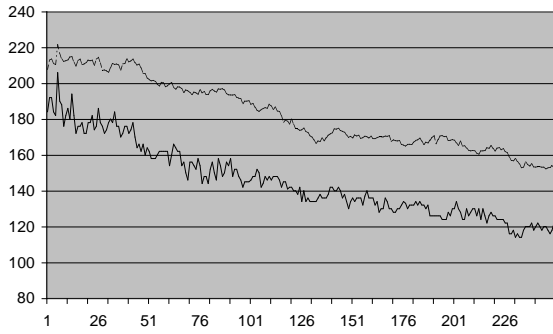
Les courbes ci-dessous représentent, en fonction du nombre de générations et du mode de sélection retenu :

- l'adaptation du meilleur chromosome (1A, 2A, 3A) ;
- la moyenne de l'adaptation de tous les chromosomes de la génération courante (1A, 2A, 3A) ;
- le nombre de chromosomes satisfaisant les contraintes (1B, 2B, 3B : échelle de gauche du graphique) ;

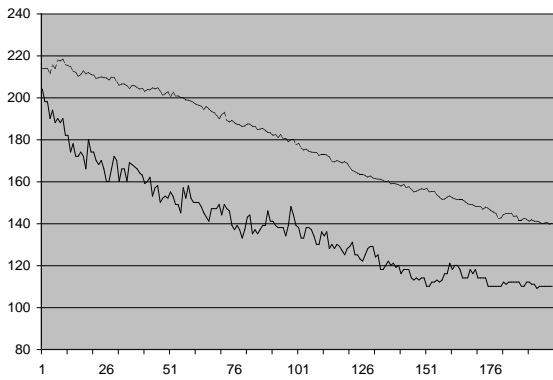
- la valeur moyenne de violation des contraintes (1B, 2B, 3B : échelle de droite du graphique).

Evolution de l'adaptation du meilleur chromosome et de la moyenne des adaptations en fonction de la génération

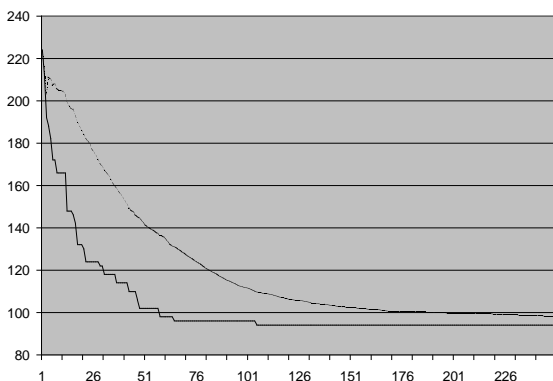
Meilleur chromosome : ———
 Moyenne des chromosomes : - - - - -



1A: sélection par roulette de casino



2A : sélection par division stochastique

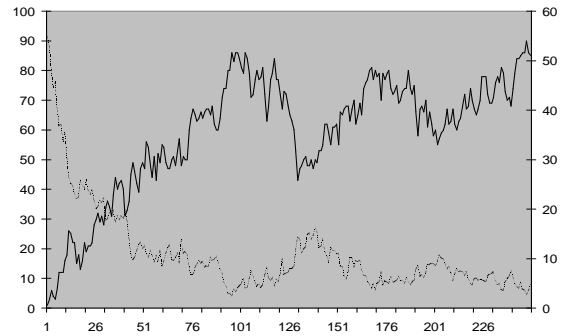


3A : sélection par crowding

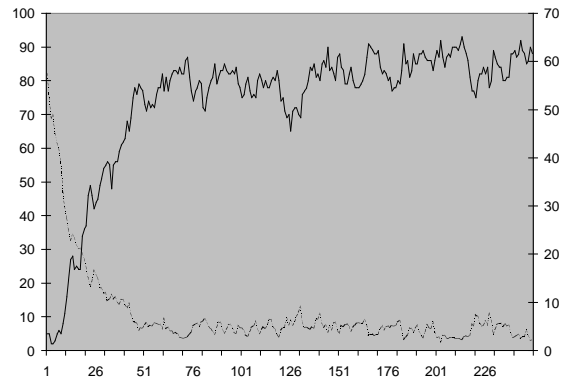
Nombre de chromosomes satisfaisant les contraintes et valeur moyenne de la violation de contrainte

Nombre de chromosomes satisfaisant les contraintes et valeur moyenne de la violation de contrainte

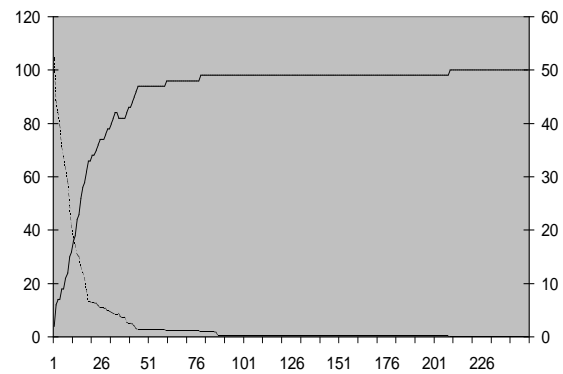
Nombre de chromosomes satisfaisant la contrainte : ———
 Valeur moyenne de la violation de contrainte : - - - - -



1B: sélection par roulette de casino



2B : sélection par division stochastique



3B : sélection par crowding

Les méthodes de sélection suivantes ont été testées (cf. [2], [3], [5]) :

- roulette de casino ;
- division stochastique ;

- *crowding* (comparaison entre parents et enfants à chaque génération, afin de préserver les meilleurs individus entre deux générations successives).

Les meilleurs résultats sont obtenus avec la division stochastique et le *crowding*, qui permettent de préserver les bons individus à chaque génération : ces deux principes de sélection sont ceux qui permettent d'obtenir le plus souvent la meilleure solution, en gardant un nombre raisonnable de chromosomes et de générations.

Cependant, en examinant les résultats obtenus pour un grand nombre d'exécution d'une même occurrence du problème, on constate qu'un nombre important de solution conduisent à des valeurs d'adaptation peu différentes : si l'on ne s'intéresse qu'aux chromosomes dont l'adaptation est au moins égale à 95% de la solution optimale, on obtient onze chromosomes différents.

Dans le cas de la modulation de cycles, les courbes ci-dessous montrent que la convergence est lente, aussi bien pour la courbe de la meilleure valeur que pour la courbe de la valeur moyenne ; en examinant en détail les résultats, on remarque qu'un nombre important de solutions correspondent à des valeurs de la fonction coût peu différentes, mais qu'elles présentent une grande diversité en ce qui concerne les valeurs de taux d'occupation et de taux de variation des effectifs. Cette remarque nous conduit alors à envisager une approche multi-critères qui permettrait de déterminer les solutions pour lesquelles les trois critères présentent simultanément des valeurs satisfaisantes. Le développement de cette démarche est décrit dans le paragraphe suivant.

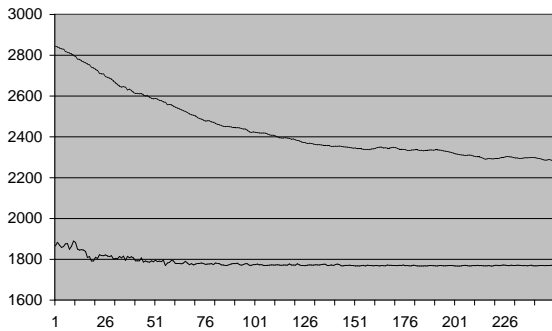


Fig 4 : Evolution du meilleur individu et de la moyenne dans la modulation de cycles

Approche globale – Algorithmes génétiques multi – critères

Les algorithmes génétiques multi – objectifs ([3], [4]) permettent d'aborder des problèmes pour lesquels on cherche à améliorer simultanément plusieurs critères, qui peuvent parfois être contradictoires ; on peut ainsi chercher à maximiser un gain tout en minimisant un coût. Une méthode consiste à pondérer les différents critères, mais le résultat est alors très sensible aux valeurs des coefficients. Pour une approche globale, indépendante de ce type de pondération, on utilise alors la notion d'optimum de Pareto.

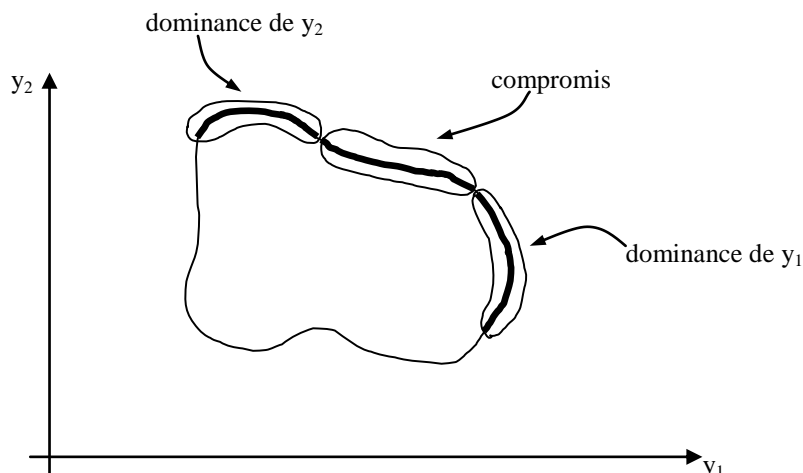
Notion d'optimum de Pareto

Soient y_i , les différents critères de la fonction multi – objectifs à optimiser ; on appelle surface de Pareto la limite du domaine telle qu'aucune solution ne permette d'améliorer simultanément tous les critères de la fonction.

Dans le cas d'une fonction à deux objectifs, on obtient alors un arc séparant le plan objectif en deux sous-domaines :

- l'ensemble des points qui ne correspondent à aucune solution ;
- l'ensemble des points correspondants à une solution du problème.

Les points de cet arc sont donc des solutions au regard de cette notion d'optimum de Pareto. On remarquera que cet arc correspond à la trajectoire décrite par $\alpha*y_1+(1-\alpha)*y_2$ lorsque α varie entre 0 et 1.



Dans le cas de deux objectifs, on retient généralement trois zones (cf graphique ci-dessous) :

- dominance de y_1 (α proche de 1) ;
- compromis (aucun critère n'émerge par rapport à l'autre) ;
- dominance de y_2 (α proche de 0).

Principe de sélection dans les algorithmes génétiques multi-objectifs

Les principes de sélection présentés ci-dessus ne peuvent plus s'appliquer puisque l'on prend maintenant en compte plusieurs valeurs correspondant à un seul chromosome. On applique donc un nouveau principe de sélection, mettant en avant la notion de dominance.

On tire tout d'abord au hasard deux individus dans la population I_1 et I_2 ; puis on tire au hasard une sous-population SP appelée à servir de référence, dont la taille est déterminée par un paramètre nommée t_{dom} . trois cas sont alors possibles :

- I_1 est dominant par rapport à SP, c'est à dire qu'il domine tous les éléments de SP, et I_2 n'est pas dominant par rapport à SP : I_1 est sélectionné ;
- I_2 est dominant par rapport à SP, c'est à dire qu'il domine tous les éléments de SP, et I_1 n'est pas dominant par rapport à SP : I_2 est sélectionné ;
- I_1 et I_2 sont tous les deux dominants ou tous les deux non-dominants ; on pourrait alors choisir aléatoirement l'un des deux individus, mais on risque alors de limiter la répartition des individus à une portion réduite de la surface de Paréto. On cherche donc à sélectionner l'individu situé dans la région la moins peuplée de l'espace objectif (partage, ou *sharing* dans l'espace objectif).

Ce partage ne modifie pas les critères, mais sélectionne les individus en fonction de la répartition dans l'espace objectif. Le but est de favoriser la reproduction d'individus appartenant à des régions où la densité de population est faible. On peut également envisager de réaliser ce partage dans l'espace d'état, mais on observe généralement de moins résultats.

Le pseudo-code ci-dessous explique schématiquement le processus suivi (en prenant comme critère d'arrêt le nombre de générations, qui est un paramètre du problème) :

```
Génération de la population initiale
Evaluation de la population initiale

Tant que (critère d'arrêt) est Vrai Faire
  Sélection des chromosomes (selon principe ci-dessus)
  Croisement et mutations des individus sélectionnés
  Evaluation des individus
  Comparaison des parents et de leurs enfants :
    Détermination des paires parents enfants les plus proches
    Dans chaque paire, sélection de l'individu dominant
    A défaut d'individu dominant, sélection de l'individu situé dans la région la moins dense.
Fin tant que
```

Les graphiques (Figures 5A à 5C) ci-dessous représentent l'évolution des individus entre la population initiale et la génération finale (sur 50 générations), évalués selon les trois objectifs :

- Minimisation de la fonction coût (F1) ;
- Maximisation du taux d'activité (F2) ;
- Minimisation du taux de variation (F3).

Chaque graphique représente les points dans un projection de l'espace objectif, où chaque axe représente un des objectifs. Les résultats intéressants sont donc situés dans les zones suivantes :

- F2 en fonction de F1 (Fig 5A) : portion supérieure gauche ;
- F3 en fonction de F1 (Fig 5B) : portion inférieure gauche ;
- FF3 en fonction de F2 (Fig 5C) : portion inférieure droite.

Les résultats obtenus confirment les capacités de convergence de l'algorithme vers les zones où les trois critères atteignent simultanément des valeurs satisfaisantes. De plus, on observe donc une convergence rapide des chromosomes : alors que la population initiale occupe l'espace objectif de manière homogène, la population se concentre au fur à mesure vers une zone limite connue sous le nom de surface de Pareto, et séparant les points correspondant à des solutions des points auxquels aucune solution ne peut être associée.

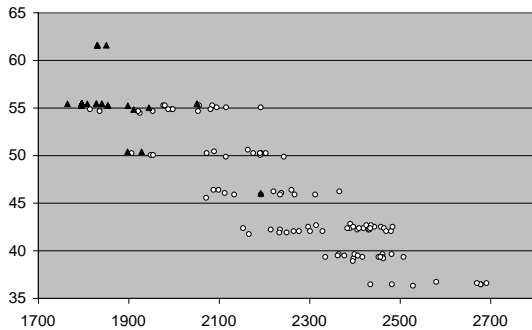


Fig 5A : F2 en fonction de F1

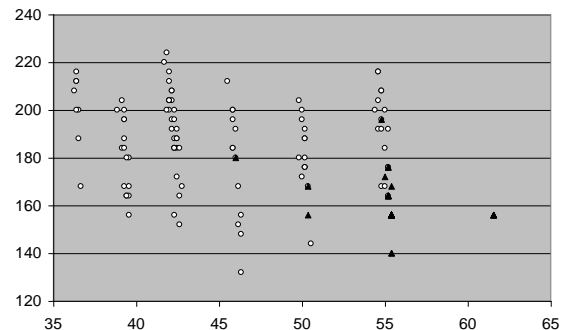


Fig 5B : F3 en fonction de F2

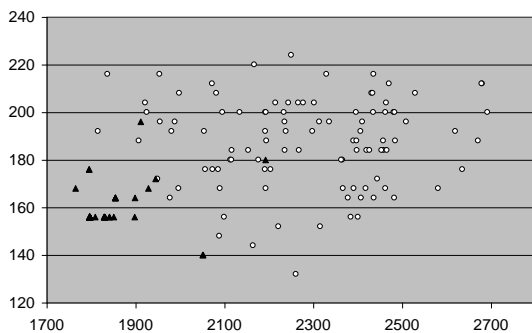


Fig 5C : F3 en fonction de F1

Conclusion:

Bien que les algorithmes génétiques ne présentent aucune garantie quant à l'obtention d'une solution optimale, ils permettent de trouver des solutions proche de l'optimum dans des temps de calcul réduits ; de plus, ils se montrent particulièrement adaptés pour des problèmes caractérisés par leur aspect hautement combinatoire ainsi que le manque de propriétés mathématiques de leur fonction objectif, nécessaire pour appliquer les outils classiques de programmation mathématique.

Dans le présent exemple, leur aptitude à résoudre des problèmes d'optimisation multi-critères sont mis en évidence. En outre, la résolution d'un tel problème par ces méthodes ne requiert ni l'augmentation du nombre de chromosomes traités à chaque génération, ni l'accroissement du nombre de génération à effectuer ; l'obtention d'un ensemble de solutions, proches de la surface de Pareto, n'a donc pas d'influence sensible sur le temps de calcul et permet de proposer aux décideurs un plus large éventail de solutions, desquelles pourront être éliminées celles qui ne respectent pas d'éventuelles contraintes supplémentaires.

Bibliographie :

- [1] Caux C., Pierreval H., Portmann M. C., « Les algorithmes génétiques et leur application aux problèmes d'ordonnancement », *Journées d'étude « Ordonnancement et entreprise : Applications Concrètes et Outils pour le Futur »* Preselction Revisited, Toulouse, Juin 1994
- [2] Davis, L., « Handbook of Genetic Algorithms », Van Nostrand Reinhold, 1991
- [3] Delahaye D., « Optimisation de la sectorisation de l'espace aérien par algorithmes génétiques », Thèse 1995

- [4] Horn, J., Nafpliotis, N., Goldberg, D.E., « A niched Pareto genetic algorithm for multiobjective optimization », In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Volume 1, 1994*
- [5] Mahfoud, S.W., « Crowding and Preselection Revisited », In *Parallel Problem Solving From Nature, 2 (PPSN 2)*, 27-36, 1992