
Coopération inter-couche dans un algorithme hybride

Illustration avec l'optimisation de constellations de satellites

Version révisée le 14 novembre 2005

Enguerran Grandchamp

Laboratoire GRIMAAG
Université des Antilles et de la Guyane
Campus de Fouillole
B.P. 592 97157 Pointe-à-Pitre
Enguerran.Grandchamp@univ-ag.fr

RÉSUMÉ. Nous nous intéressons ici à l'optimisation hybride. Si le choix des différentes méthodes mises en oeuvre pour résoudre un problème complexe est important, la manière de les combiner s'avère également cruciale pour une recherche efficace. Cet article, focalise sur les échanges entre les couches algorithmiques et décrit la manière de les articuler dans le cadre de l'optimisation de constellations de satellites. Ces échanges permettent une adaptation dynamique des paramètres des méthodes en fonction de leur avancée mutuelle

ABSTRACT. This paper deals with hybrid optimization take different methods to solve a complex problem. As if the choice of concerning methods is important, the way to combine them is determinant for an efficient search. The article deals with cooperation strategy between the different layers of an hybrid optimization algorithm. We describ the way it is articulated for an Satellite Constellation Optimization problem. This cooperation allows a dynamical adaptation of the optimization algorithm parameters according to the mutual advance of each search

MOTS-CLÉS : optimisation hybride, meta-heuristique, recherche taboue, ensemble flou

KEYWORDS: hybrid optimisation, meta-heuristics, tabou search, fuzzy-set

1. Introduction

De nombreux grands problèmes d'optimisation combinatoire ont été traités par des méthodes approchées à caractère global ou local : le problème du voyageur de commerce, le problème du sac à dos, les problèmes de recherche de plus court chemin, ... Ainsi de nombreux algorithmes de résolution ont vu le jour par application d'une méthode d'optimisation unique telle que les algorithmes génétiques [GOL 89] (AG), les colonies de fourmis [RIC 05], la recherche taboue [GLO 97], [GLO 01], le recuit simulé [JOH 89], la recherche dispersée [GLO 94], ou encore les méthodes de satisfaction de contraintes, de simplexe,

Même si les algorithmes de résolution proposés pour ces différents problèmes prouvent leur utilité dans de nombreux domaines : aérospatial, aéronautique [EDW 05], automobile [MUY 03], chimie [YED 02], physique [TRO 98], [DEB 04], télécommunication (positionnement d'antenne), réseaux (optimisation de routage), parallélisme, transport [ZNE 04], macro économie [SCH 04], [DOE 04], théorie des jeux, supervision de procédés, multimédia [CHE 05a] ...). L'optimisation de systèmes industriels réels est devenue plus complexe et l'utilisation de méthodes hybrides nécessaire.

Une première forme d'hybridation a vu le jour avec l'apparition des techniques de clustering ou de redémarrage qui permettent de réaliser une recherche locale à partir d'une méthode à caractère global, ou inversement. Le clustering permet à une recherche globale de focaliser sur des zones et de réaliser une recherche plus poussée uniquement dans celles-ci (caractère local). La technique du redémarrage permet à une recherche locale de sortir des optima locaux en explorant une autre région de la zone de recherche.

On trouve dans la littérature beaucoup d'algorithmes hybrides ([CHE 05b], [LIN 05], [HON 04], [PRA 04], [SHI 04], [PET 04], [KAI], [PEN 04], [HSI 04]).

De manière générale, on associe une méthode d'optimisation globale avec une méthode d'optimisation locale de manière à garantir la convergence vers l'optimum global avec une assurance d'affiner les paramètres au cours de la convergence. Dans ce cadre, Schmidt et Thierauf [H.S 05] ainsi que Botello et al. [BOT 99] présentent une association entre le recuit simulé et les algorithmes génétiques. Les algorithmes génétiques sont très souvent utilisés dans les algorithmes hybrides pour leur propriété de convergence stochastique vers l'optimum global (Burk et al. [BUR 01], [BUR 00], Magoulas et al. [MAG 01]), on parle dans ce cas d'algorithmes évolutionnaires hybrides [P.G 04].

Les approches précédentes diffèrent par la manière de combiner les algorithmes utilisés. Marco et al. [MAR 99] par exemple, utilisent les AGs en début de recherche puis l'algorithme de descente une seule fois en fin de convergence pour améliorer les meilleurs individus. Renders et al. [REN 96], Vicini et al. [VIC 98] et A. Ben Haj Yedder [YED 02] définissent un nouvel opérateur de mutation à l'aide d'une méthode déterministe.

D'autres associations plus spécifiques ont également fait leur apparition. C'est le cas de l'association des problèmes de satisfaction de contraintes (CSP) et des AG ([BAR 04], [P.G 04], [V.B 03]) ou l'association AG et réseaux de neurones [YAN 05], [SUN 04]. On cherche ici à réduire le nombre d'évaluations exactes de la fonction

coût de manière à réduire le temps d'exécution.

L'algorithme hybride présenté ici est composé de deux couches algorithmiques (la recherche taboue et la recherche de plus grande pente) permettant de réaliser deux objectifs distincts (une recherche de caractéristiques de la solution et un affinement des paramètres). La singularité de cette approche ne réside pas dans l'association de ces deux méthodes mais dans leur manière de coopérer et d'échanger des informations pour mutuellement progresser dans la réalisation d'un objectif commun. Cette coopération est rendue possible par l'introduction d'une articulation entre les deux couches algorithmiques : l'ODB (Orbit Data Base).

Au cours de cet article, nous insisterons sur la description de l'ODB ainsi que sur les échanges entre les différentes couches algorithmiques. De plus amples détails sur l'algorithme lui-même sont donnés dans d'autres publications ([GRA 00],[GRA 01b],[GRA 01a]). Les résultats présentés concernent également le comportement du modèle, une étude des résultats de l'algorithme en terme de constellation obtenue est disponible en [GRA 01a] et [GRA 04].

Cet article commence par positionner le problème à traiter (section 2) à travers la définition des variables du problème, de l'espace de recherche et de la fonction d'évaluation. Les sections 3 et 4 présentent le coeur de cet article avec la modélisation du problème et les échanges inter couche. Quelques résultats sur l'évolution des variables du système sont présentés en section 5. Enfin, la section 6 présente les conclusions et perspectives.

2. Position du problème

Le problème de référence traité ici est le problème d'Optimisation de Constellations de Satellites. Ce problème est né avec la conquête spatiale à partir des années 60-70 [WAL 70] et est toujours d'actualité. Les systèmes spatiaux sont de plus en plus complexes tant en nombre de satellites qu'en terme de performances attendues. De la conception du système spatial à son exploitation, les principales étapes [GRA 01a] nécessitent une phase d'optimisation : design de la constellation, mise à poste, maintien à poste.

Ce travail a été réalisé dans un contexte industriel proposé par Alcatel Space Industries à Toulouse. Il a été calibré sur des constellations déjà connues (constellation GPS par exemple) et a donné lieu au développement d'un logiciel complexe permettant le paramétrage, le suivi et la visualisation des résultats.

Le problème général est posé dans un contexte multiobjectif consistant à maximiser les performances et à minimiser les coûts sur l'ensemble des phases. Mais nous abordons ce problème d'optimisation avec une vue d'ingénieur plus que de mathématicien. Nous ne réaliserons donc pas une optimisation multi-objectif. Nous nous intéressons uniquement au design de la constellation et le coût de la constellation est donné par le nombre de satellites avec une valeur maximale fixée.

Cependant, l'optimisation multicritère dans le domaine spatial a été envisagée [LAN 96], [ROM 98] pour des sous problèmes.

2.1. Les variables du problème

L'optimisation du design de la constellation doit résoudre le problème du choix du nombre de satellites et de leur position. La position d'un satellite évolue dans le temps selon des lois horaires définies par l'orbite elliptique qu'elle emprunte. L'orbite est définie par 5 paramètres :

- l'ellipse est définie par son demi grand axe (a) et son excentricité (e)
- la position de l'ellipse par rapport à la terre est définie par trois angles : l'angle que fait le plan de l'orbite avec le plan de l'équateur (c'est l'inclinaison notée i), l'angle que fait le plan de l'orbite avec le plan contenant le méridien de Greenwich (c'est la longitude du noeud ascendant notée Ω), l'angle de rotation de l'orbite dans son plan (c'est l'argument du périhélie noté ω).

La position du satellite sur l'orbite est donnée par un angle (c'est l'anomalie excentrée, notée E). Des détails sur ces paramètres sont donnés en [GRA 01a] et [CNE 89]. Les lois horaires permettant le calcul de la position du satellite sont explicitées dans le paragraphe 2.3.

La complexité du problème à résoudre est en partie due à la nature des paramètres. On trouve des paramètres réels (les six paramètres d'un satellite) et un paramètre entier (le nombre n de satellites). Le nombre de paramètres à déterminer ($6n$) est donc inconnu. Par ailleurs, les six paramètres d'un satellite ont une signification physique différente ce qui implique un traitement particulier de chacun d'eux. Par exemple, les paramètres ne peuvent pas être discrétisés sans compromettre la résolution du problème et ce pour deux raisons majeures : pour obtenir certains types d'orbites (Géostationnaire, héliosynchrone, phasée, ...) les paramètres doivent avoir des valeurs précises calculées en fonction de données extérieures (position de la zone visée, temps de revisite d'un point, ...); l'impact d'une variation d'un paramètre dépend de la valeur de celui-ci (exemple : une variation de quelques kilomètres a un impact plus important à une altitude de 400 km qu'à une altitude de 25000 km).

2.2. L'espace de recherche

L'espace de recherche est constitué de l'ensemble des constellations possibles, c'est-à-dire du produit cartésien des intervalles de variation des $6n$ paramètres précédemment cités. Aucune séparation des variables n'est possible pour réduire la complexité de l'espace de recherche. Le nombre de solutions potentielles est donc infini. Cet espace de recherche est la principale difficulté dans la résolution du problème et nous attacherons un effort particulier à proposer une nouvelle modélisation de celui-ci qui permettra d'organiser la recherche. Les plages de variation des paramètres sont les suivantes

- $e \in [0, 1[$
- $\{\omega, \Omega, E, i\} \in [0, 360]^4$
- $a \in [400, 36000]$

– $n \in [1, 50]$ (pour les tests réalisés ici).

2.3. La fonction d'évaluation

La fonction d'évaluation est le problème central dans la recherche d'une solution optimale. Elle est très complexe et très coûteuse en temps de calculs, empêchant ainsi toute exploration par évaluation d'un grand nombre de solutions potentielles. Différentes fonctions peuvent être utilisées en fonction du domaine d'application. On peut citer les principales :

- en télécommunications, c'est essentiellement un critère de taux de couverture qui sera utilisé.
- en navigation il faudra non seulement quatre satellites visibles pour chacune des zones d'intérêt mais en plus une position particulière des satellites pour une bonne triangulation de la position.
- en observation, des critères tels que le temps de retour au dessus d'un même point, ou l'heure de passage (jour, nuit, ...) seront préférés.

L'évaluation d'une solution est toujours basée sur la simulation de la constellation durant sa période de révolution. En effet, la position des satellites évolue au cours du temps en fonction des lois horaires suivantes

$\frac{d}{dt}$	dérive
a	0
e	0
i	0
ω	$-\frac{3}{4} \left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^2} \sqrt{\frac{\mu}{a^3}} (1 - 5 \cos^2 i)$
Ω	$-\frac{3}{2} \left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^2} \sqrt{\frac{\mu}{a^3}} \cos^2 i$
M	$\left(\frac{\mu}{a^3}\right)^{1/2} - \frac{3}{4} \left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^{3/2}} \sqrt{\frac{\mu}{a^3}} (1 - 3 \cos^2 i)$

Tableau 1. Dérives

Le paramètre M (anomalie moyenne) est lié à l'anomalie excentrée E par la relation suivante

$$M(t) = E(t) - e \sin E(t) \tag{1}$$

Le calcul de la position d'un satellite nécessite la résolution de cette équation afin d'exprimer E . La simulation est donc un processus coûteux qui peut prendre plusieurs minutes sur une machine puissante pour une seule constellation. Par exemple, pour une application en navigation, l'échantillonnage spatial peut être de 10km et temporel de 5mn. Ce qui donne 1475448838 échantillons à tester pour évaluer les performances

de la constellation sur la surface de la terre. Sachant que, dans le cas de la navigation, pour chaque échantillon une minimisation au sens des moindres carrés est nécessaire, le temps de calcul est prohibitif. Des simulations plus coûteuses sont envisageables pour des applications militaires par exemple.

Différents problèmes se posent alors : comment représenter l'espace de recherche afin de progresser efficacement ? Comment minimiser le coût d'évaluation d'une solution ? La modélisation proposée dans la section suivante va tenter d'apporter des éléments de réponse pour chacune de ces questions.

3. Modélisation du système

Le principal problème étant le coût de l'évaluation des solutions, nous allons mettre en oeuvre deux idées directrices : la réduction du nombre de solutions évaluées, la réduction du coût d'une évaluation.

Ainsi nous allons guider la recherche au travers d'une décomposition de l'espace de recherche formé par l'ensemble des constellations de satellites. On part donc d'un système composé des constellations et dont le seul critère permettant de rechercher l'optimum est la fonction d'évaluation coûteuse décrite à la section 2.3 pour arriver à une décomposition en trois niveaux (figure 1)

- un niveau numérique agissant au niveau d'une orbite particulière (6 paramètres) dont la fonction d'évaluation, coûteuse, reste la même (section 3.2)
- un niveau heuristique manipulant des classes d'orbites et dont la fonction d'évaluation a une expression simple (section 3.1)
- un niveau métaheuristique manipulant des configurations de classes et dont la fonction d'évaluation est également simple (section 3.3).

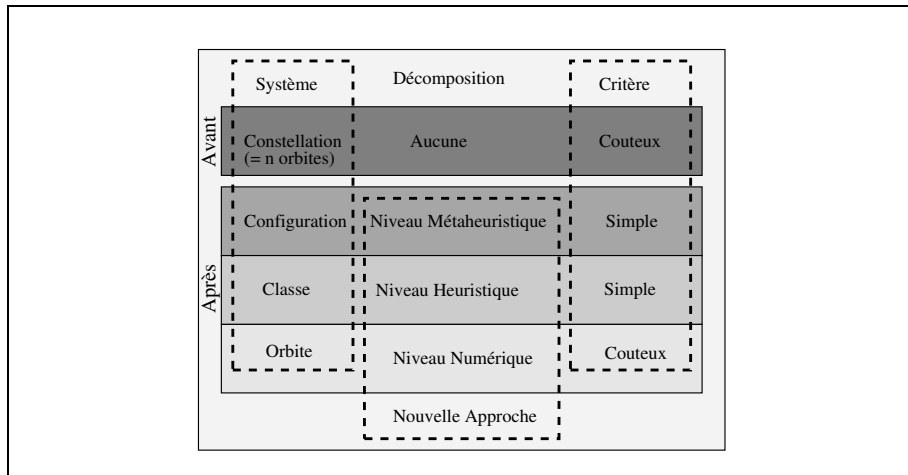


Figure 1. Décomposition du problème

Les fonctions d'évaluation que nous introduisons pour les différents niveaux sont des mesures statistiques sur les classes et des interpolations sur ces mesures. Ces valeurs sont proches de moyennes pondérées permettant de mesurer la qualité de chaque classe et de chaque configuration par rapport aux configurations visitées. Ces valeurs sont donc très représentatives des configurations visitées et fortement liées à celles-ci (valeur a posteriori). Par contre, elles ne sont pas nécessairement représentatives en début de recherche pour l'ensemble des configurations (prédiction avec les valeurs a priori). C'est l'augmentation du nombre de configurations visitées au cours de la recherche qui rendra plus pertinente les valeurs a priori.

Ces nouvelles fonctions d'évaluations ont été définies intuitivement. Elles sont fiables par construction pour l'évaluation des configurations visitées. Pour les configurations non visitées, la fiabilité augmente avec la convergence (section 5).

L'expression de ces fonctions d'évaluation est simple et donc facilement calculable durant les simulations ou a posteriori (section 3.2).

Dans cette section, nous décrivons essentiellement la couche heuristique qui constitue le coeur de l'algorithme de résolution.

3.1. Représentation de l'espace de recherche : la couche heuristique, mémoire de l'algorithme

Cette couche est qualifiée d'heuristique car la représentation de l'espace de recherche va permettre l'introduction d'une expertise. Aucun algorithme particulier n'est associé à cette couche, elle sert seulement de support pour la couche supérieure. La nouvelle représentation de l'espace est basée sur des regroupements d'orbites (6-uplet) ayant des caractéristiques communes dans des *classes orbitales*. Une classe va donc définir un domaine de variation pour chacun des six paramètres orbitaux. Certains paramètres seront indépendant des autres, leur domaine de variation sera alors un intervalle. Pour les paramètres liés à d'autres, on définira soit un intervalle de variation pour l'un des paramètres et les autres seront calculés en fonction de ce dernier soit des ensembles de valeurs possibles pour les paramètres. C'est le cas par exemple pour les orbites héliosynchrones (passage au dessus d'un point à la même heure solaire) qui nécessitent des couples altitude (a) et inclinaison (i) particuliers [GRA 01a].

Par ailleurs, on introduit ici la notion d'intervalle flou [ZAD 65], [LUH 04] afin de traduire l'aspect incertain de certaines frontières. C'est le cas par exemple des orbites à basse et moyenne altitude dont la limite d'altitude n'est pas clairement définie. De plus, si une limite figée existait, classer une orbite en moyenne altitude alors qu'une orbite d'altitude très proche serait classée en basse altitude n'aurait aucun sens. De même sur un autre exemple simple, on peut décider de regrouper les orbites d'altitude basse dans une première classe et les orbites circulaires dans une deuxième classe. L'intersection de ces deux classes est l'ensemble des orbites circulaire à basse altitude.

Une classe est donc un ensemble de 6 domaines de variation (intervalle flou ou rela-

tion entre paramètres). On notera Cl_i la classe numéro i . La *base de donnée orbitale* (*ODB*) regroupe l'ensemble des classes et sera notée : $ODB = \{Cl_1, \dots, Cl_m\}$, m étant le nombre de classes.

Bien que nous ayons défini un ensemble de classes de référence, le choix des classes est laissé libre à l'utilisateur. Il peut ainsi décider de réaliser une partition de l'espace de recherche, ou au contraire d'exclure certaines régions en limitant les classes et/ou en réalisant des chevauchements entre classes. Cette dernière caractéristique peut s'avérer être un atout majeur lors de la mise en oeuvre de l'algorithme, de plus amples détails sont donnés à la section 4.4.

L'espace de recherche du problème initial est l'ensemble des constellations composées de 1 à N_{max} satellites. De la même manière que la classe orbital est un moyen de généraliser une orbite, on introduit la notion de *configuration* qui généralise une constellation. Une configuration est un ensemble de représentants de chacune des classes on la note $Cf_p = \{N_{s1}^p, \dots, N_{sm}^p\}$, N_{si}^p étant le nombre de représentants de la classe Cl_i dans la configuration Cf_p , on note N_p le nombre total de satellites de la configuration Cf_p .

Dans sa conception, cette décomposition ne réduit ni la complexité du problème à traiter, ni le type des solutions accessibles. Néanmoins, le regroupement d'orbites permet d'apprécier la pertinence d'un type d'orbite sans en explorer tous les représentants. Par ailleurs, la définition de classes d'orbites particulières permet de guider la recherche à un haut niveau en conditionnant le type de solution cherché, on réduit ainsi l'exploration pour accélérer la recherche au détriment de la diversité des solutions envisagées. Le principal avantage de cette représentation est que l'on sépare, au niveau algorithmique, la recherche du nombre de satellites et du type d'orbite de l'affinement des paramètres. On remplace ainsi un problème en nombre de variables inconnu ($6N$) combinant variables discrètes (N) et variables continues (les paramètres orbitaux) en deux problèmes :

- l'un d'optimisation combinatoire : il s'agit de trouver les configurations (classes d'orbite à utiliser et nombre de représentants).
- l'autre d'optimisation continue : à nombre et type de satellite fixé, il s'agit d'optimiser la valeur des paramètres orbitaux.

3.2. La couche numérique

La couche basse de l'algorithme est qualifiée de numérique car elle consiste à affiner la valeur des paramètres orbitaux d'une constellation avec un degré de liberté réduit (pas de changement du nombre de satellites, variation des paramètres orbitaux dans un domaine restreint). Sous ces hypothèses, la fonction d'évaluation est régulière et autorise l'utilisation de méthodes classiques d'optimisation comme la recherche de plus grande pente. Néanmoins, il peut exister plusieurs optima locaux, des techniques de redémarrage sont donc prévus [GRA 01a]. Par ailleurs, l'évaluation d'une constellation est toujours basée sur une simulation nécessitant un grand nombre de calculs pour ne retourner que très peu de valeurs (moyenne des performances, performance

maximale ou minimale). Nous avons donc décidé d'utiliser ce temps de simulation pour extraire un certain nombre de caractéristiques de la constellation évaluée et ainsi diminuer le coût relatif des évaluations. Il faut noter que le surcoût engendré est négligeable (calcul direct des valeurs par opposition aux calculs de moindres carrés par exemple). Nous allons ainsi associer quatre valeurs aux satellites indiquant leur contribution. V_{s1} : reflète l'utilisation du satellite en situation critique, V_{s2} : en situation non critique, V_{s3} : contribution positive à la constellation, V_{s4} : contribution négative à la constellation. La signification précise de ces valeurs ainsi que leur mode de calcul sont donnés en [GRA 01a].

3.3. La couche métaheuristique

C'est le chef d'orchestre de l'algorithme hybride, c'est ici que le choix des configurations à visiter sera effectué. La méthode utilisée pour l'implantation de ce niveau est la recherche taboue car c'est une méthode de recherche locale (donc moins coûteuse en temps de calcul) offrant suffisamment d'aspects avancés pour éviter les optima locaux. L'approche du problème général par les algorithmes génétiques a été écartée pour des raisons de temps de calcul liés à la taille de la population et au coût d'évaluation d'une solution.

Nous ne détaillerons pas ici la recherche taboue, suffisamment décrite dans la littérature [GLO 97]. Nous indiquons uniquement que l'espace de recherche à ce niveau est composé de configurations. Le voisinage d'une configuration Cf_p est l'ensemble des configurations accessibles en ajoutant, supprimant ou remplaçant un satellite. Le nombre de voisins est donc de $m * (N_p + 1)$ [GRA 01a]. Nous avons ici un problème combinatoire à résoudre, pour le problème "trouver la configuration optimale composée de n_{min} à n_{max} satellites choisis parmi m classes, la taille de l'espace de recherche est de $\frac{m^{n_{max}+1} - m^{n_{min}}}{m-1}$ [GRA 01a].

Notons que l'évaluation d'une configuration n'est ni exhaustive ni déterministe puisqu'il s'agit d'évaluer un sous ensemble des constellations possibles au sein d'une même configuration (sous ensemble parcouru par la couche numérique). Deux évaluations d'une même configuration peuvent donc différer. Au cours de la recherche de niveau méta-heuristique, un retour sur une configuration peut donc avoir un intérêt. En effet, si on conserve un point de redémarrage dans chaque configuration visitée, un retour permettra de prolonger l'exploration et ainsi d'affiner la recherche bas niveau.

4. La coopération inter couche

La coopération inter-couche sera réalisée grâce aux différents aspects de la mémoire qui va permettre de réaliser un véritable diagnostic de la recherche à tous les niveaux (numérique, heuristique et métaheuristique) afin de produire des valeurs a priori et a posteriori des classes et configurations. Nous allons successivement décrire ici les échanges d'informations entre les couches afin de comprendre comment les différentes méthodes de recherche peuvent combiner leurs efforts dans la réalisa-

tion d'un objectif commun. Ces échanges sont possibles grâce à l'intégration des différents types de mémoire de la recherche taboue à tous les niveaux de l'algorithme hybride. Nous allons effectivement évaluer chacun des composants de notre nouveau système (satellite, orbite, classe, constellation, configuration) de manière qualitative et quantitative avec une approche court et long terme.

De manière simplifiée, une couche va conditionner la couche inférieure en fixant ses paramètres de recherche. Inversement, après une recherche de bas niveau, un certain nombre de données vont remonter à la couche supérieure et vont lui permettre d'affiner ses choix.

4.1. Descente d'informations de la couche heuristique vers la couche numérique

C'est le transfert le plus simple. Les classes de la couche heuristique permettent de fixer les paramètres suivants pour la couche numérique : la solution initiale (nombre de satellites et valeur des paramètres orbitaux), les domaines de variation des $6n$ paramètres, le critère d'arrêt pour la convergence, la fonction d'évaluation.

La couche heuristique va piloter l'algorithme numérique en fonction des précédentes recherches effectuées sur les mêmes classes ou configurations. La solution initiale proposée peut ainsi être la dernière solution visitée (au début de la recherche par exemple, c'est la mémoire à court terme), ou alors la meilleure constellation rencontrée pour la même configuration lors des précédentes recherches (en phase de convergence par exemple, aspect qualitatif de la mémoire à long terme).

Le critère d'arrêt est soit un seuil sur la progression ou la valeur de la solution atteinte (en début de recherche) soit un nombre d'itérations (en phase de convergence). Sa valeur va varier au cours des itérations pour par exemple limiter l'algorithme de bas niveau en début de recherche et l'exploiter plus longuement en fin de recherche.

De la même manière la fonction d'évaluation va changer au cours des itérations de l'algorithme de haut niveau. Si un calcul simplifié (échantillonnage spatio-temporel grossier) peut suffire pour avoir une idée, en début de recherche, du type de performance envisageable pour une constellation, un calcul de plus en plus précis devra être effectué au fur et à mesure de la convergence.

4.2. Remontée d'informations de la couche numérique vers la couche heuristique

Après une optimisation numérique, un certain nombre d'informations doit remonter vers la couche heuristique afin de rendre compte de la recherche de bas niveau. L'objectif est de mettre à jour la valeur des classes orbitales, on utilise pour cela la valeur des constellations rencontrées ainsi que l'analyse de leurs satellites. On calcule ainsi la valeur dite *a posteriori* des classes selon la formule suivante.

$$V_{cl}(Cl_i) = \frac{1}{\sum_{j=1}^N N s_i^j} * \sum_{j=1}^N N s_i^j * V_{cf}^1(Cf_j)$$

V_{cf}^1 est la valeur a posteriori d'une configuration, explicitée dans la suite de la section. Cette valeur rend compte de l'utilité d'une classe, elle est d'autant plus fiable que la classe est représentée dans les configurations testées.

4.3. Descente d'informations de la couche méta-heuristique vers la couche heuristique

La couche méta-heuristique va décider de la composition de la configuration à évaluer. Elle ne s'occupe que de fixer le nombre de représentants de chaque classe dans la configuration courante.

Au niveau le plus haut, le voisinage d'une solution est composé de configurations. Le choix du successeur veut donc dire en théorie évaluer et classer les configurations voisines. Mais l'évaluation exacte d'une configuration implique d'explorer tous les représentants de celle-ci (toutes les constellations dont les satellites vérifient les domaines de variation des classes auxquelles ils appartiennent). Cette exploration du voisinage serait trop coûteuse, nous allons donc mettre en place une sélection basée sur d'autres valeurs calculées grâce à l'aspect mémoire de la recherche taboue. On introduit donc la notion de valeur a priori d'une configuration calculée de la manière suivante

$$V_{cf}^2(Cf) = \frac{1}{Card(\Phi)} \sum_{Cf' \in \Phi} \left(V_{cf}^1(Cf') \cdot \frac{M(Cf)}{M(Cf')} \right)$$

$$M(Cf) = \frac{1}{\sum_{i=1}^m Ns_i} \sum_{k=1}^m (Ns_k \cdot V_{cl}(Cl_k))$$

On définit pour cela une distance maximale d_{max} entre la configuration à évaluer (Cf) et les configurations intégrées dans le calcul. Le calcul de distance entre deux configurations est explicité dans [GRA 01a]. On note Φ l'ensemble des configurations déjà visitées situées à une distance inférieure à d_{max} ($\Phi = \{Cf' | Cf' \in \Delta, d(Cf, Cf') \leq d_{max}\}$, avec Δ l'ensemble des configurations visitées).

Cette valeur est moins fiable qu'une exploration mais son calcul est instantané et sa valeur est mise à jour dynamiquement après chaque exploration d'une configuration.

Après avoir fixé la configuration, les 6 paramètres de chacun des représentants sont choisis par la couche heuristique qui dispose de l'information nécessaire (valeur de classe, de satellite, ...). En effet, dans le cas de la suppression (remplacement) d'un satellite par exemple le choix du satellite est à la charge de la couche heuristique. Le choix est effectué en sélectionnant le représentant de la classe à supprimer (remplacer) le moins utile dans la constellation (valeurs V_{s2} et V_{s4} du satellite). De même pour l'ajout d'un satellite, les nouveaux paramètres seront calculés en fonction des paramètres des satellites déjà présents dans la configuration.

Les paramètres des satellites conservés d'une configuration à l'autre sont choisis comme indiqué dans la section 4.1.

4.4. Remontée d'informations de la couche heuristique vers la couche méta-heuristique

Lorsqu'une configuration a été visitée, une nouvelle valeur doit lui être associée, c'est sa valeur *a posteriori*. Elle est calculée de la manière suivante

$$V_{cf}^1(Cf) = \frac{1}{\text{Card}(\Pi_{Cf})} \sum_{Cs \in \Pi_{Cf}} (f(Cs)).$$

Π_{Cf} étant l'ensemble des constellations visitées au sein de la configuration Cf . C'est également à ce niveau que la notion d'intervalle flou va prendre son intérêt et le rôle d'articulation de la couche heuristique son importance. La convergence de certains paramètres dans la couche numérique vers la limite des classes, va entraîner l'évolution de la configuration actuelle vers une nouvelle. C'est une sorte de critère d'aspiration qui va faire ressortir certaines solutions du voisinage de la configuration courante (niveau métaheuristique).

Nous allons illustrer ce transfert dans deux cas très différents. Le premier est directement lié à la notion d'intervalle flou et d'intersection d'ensemble. Si un paramètre tend vers la limite d'un intervalle, on va considérer comme prioritaires les configurations voisines qui vont remplacer cet intervalle par un autre intervalle contenant la valeur courante du paramètre. Ceci permet de guider efficacement la recherche sans explorer le voisinage au niveau métaheuristique. Le second exemple fait intervenir la signification physique des paramètres orbitaux. Si on considère que l'on souhaite obtenir une constellation en orbite basse et que la convergence au niveau numérique fait tendre le paramètre d'altitude des satellites vers la borne supérieure de leurs intervalles respectifs, il n'est alors pas possible de remplacer la classe d'orbite basse par une classe d'orbite moyenne. Néanmoins, cette évolution de l'altitude nous donne une indication cruciale : en prenant de l'altitude les satellites couvrent une surface au sol plus importante, la convergence vers l'altitude maximal tend à montrer que le nombre de satellites de la constellation n'est pas suffisant pour réaliser une bonne couverture à une basse altitude. Les configurations prioritaires dans le voisinage de la configuration courante seront donc les configurations obtenues par ajout d'un satellite.

4.5. Bilan : la couche heuristique, articulation de l'algorithme

La modélisation de l'espace de recherche place l'ODB au coeur de la méthode de résolution : elle centralise la mémoire de l'algorithme à court et long terme, elle est l'articulation permettant les échanges entre les couches algorithmiques.

Au fur et à mesure de la convergence, les paramètres de l'algorithme hybride vont évoluer : les valeurs *a posteriori* vont remplacer les valeurs *a priori*, l'effort réalisé en début de recherche au niveau métaheuristique pour sélectionner les configurations d'intérêt va être réduit au profit d'une optimisation bas niveau plus poussée.

La réduction de l'exploration des configurations est réalisée grâce aux attributs tabous positionnés pour les éléments du système tels que les classes orbitales. En effet, une

classe dont la valeur serait trop faible, serait mise à l'état tabou pour éviter d'inclure dans la constellation courante un de ses représentants. Au cours des itérations, le nombre de classe va donc diminuer (on écarte les classes sans intérêt).

L'évolution des différents paramètres est régulée grâce aux techniques d'oscillation stratégique ([GLO 97], [GLO 01]). C'est-à-dire que l'on va alterner des phases d'intensification consistant à réduire progressivement les degrés de liberté d'une constellation (nombre de satellites, domaine de variation des paramètres, ...) et des phases de diversification consistant à changer la structure de la solution courante (nombre de satellites, classes utilisées, ...). La figure 2 illustre ce principe.

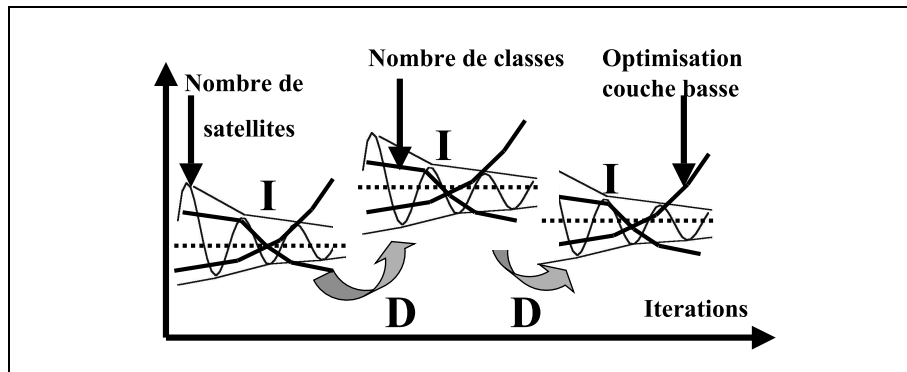


Figure 2. Les oscillations stratégiques

5. Evolution des variables du système : quelques résultats

Nous présentons ici quelques résultats sur l'évolution des paramètres du système afin de vérifier leur cohérence. Des résultats sur les solutions obtenues sont présentées dans [GRA 01a], [GRA 04], [GRA 01b].

Le gain en temps de calcul est réalisé grâce à l'introduction des valeurs a priori, il faut donc vérifier que ces valeurs sont représentatives et les comparer aux valeurs a posteriori. La figure 3 présente l'évolution de la valeur a priori et a posteriori de quatre configurations. On constate qu'après une phase de transition les deux types de valeurs convergent. Les valeurs a priori n'ont de sens que si un minimum de configurations est exploré, ceci explique les changements de la valeur a priori pendant la phase de transition. De même, au début de l'exploration des configurations, la moyenne des constellations rencontrées change rapidement, tant dis qu'au fur et à mesure de la progression de la recherche, la moyenne se stabilise et augmente (convergence vers une bonne solution). Ceci explique les variations de la valeur a posteriori.

La figure 4 illustre la variation du nombre de classes au cours de la recherche. On visualise clairement l'effet des oscillations stratégiques qui alternent phase d'intensification (notée I sur la figure) et phase de diversification (notée D sur la figure). De

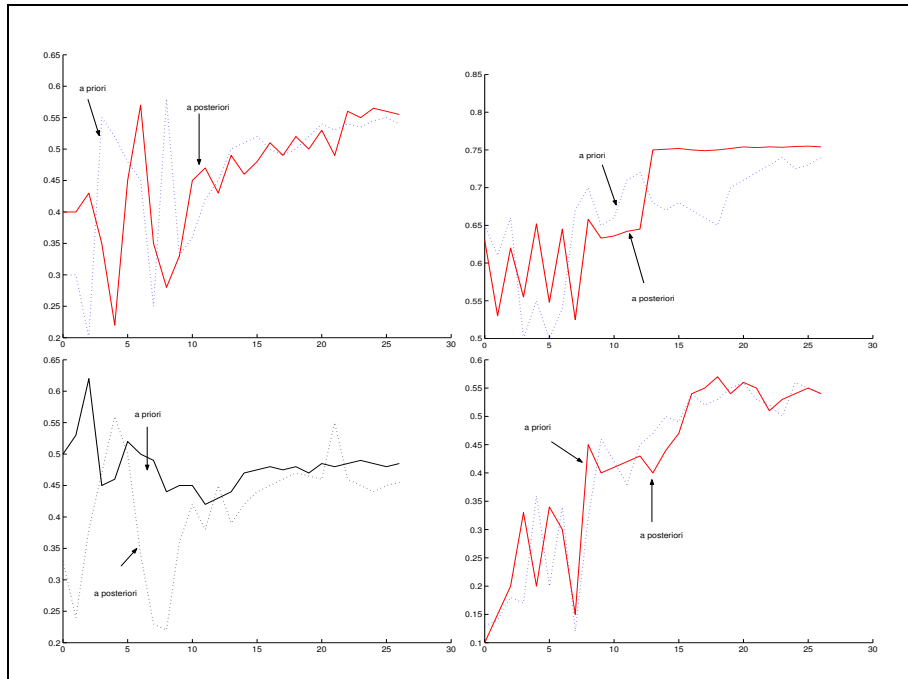


Figure 3. Comparaison des valeurs a priori et a posteriori d'une configuration

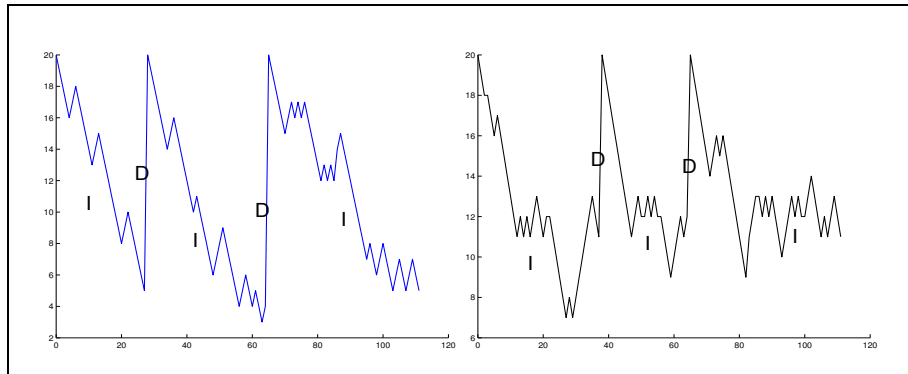


Figure 4. Evolution du nombre de classes

même la figure 5 illustre l'évolution du nombre d'itérations du niveau numérique qui augmente avec la progression de l'intensification.

Enfin, la figure 6 montre l'évolution de la valeur de la meilleure constellation rencontrée qui augmente par palier en fonction de la configuration explorée ainsi que la

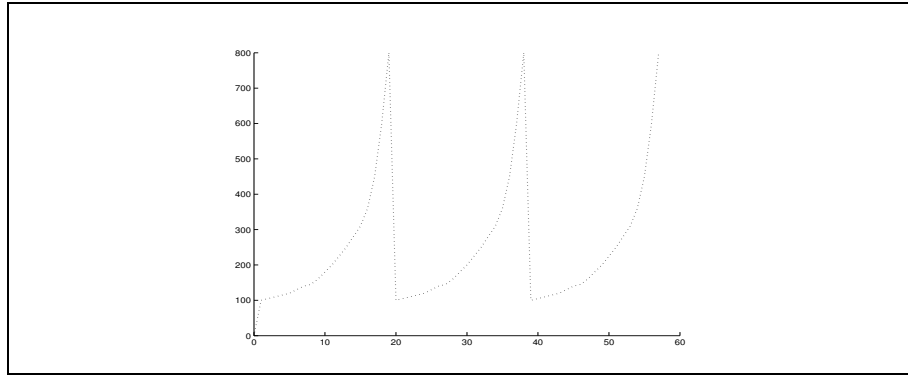


Figure 5. Evolution du nombre d'itérations de la SD

valeur de la meilleure constellation de la configuration courante. L'algorithme numérique permet d'améliorer de manière locale la solution au sein d'une configuration.

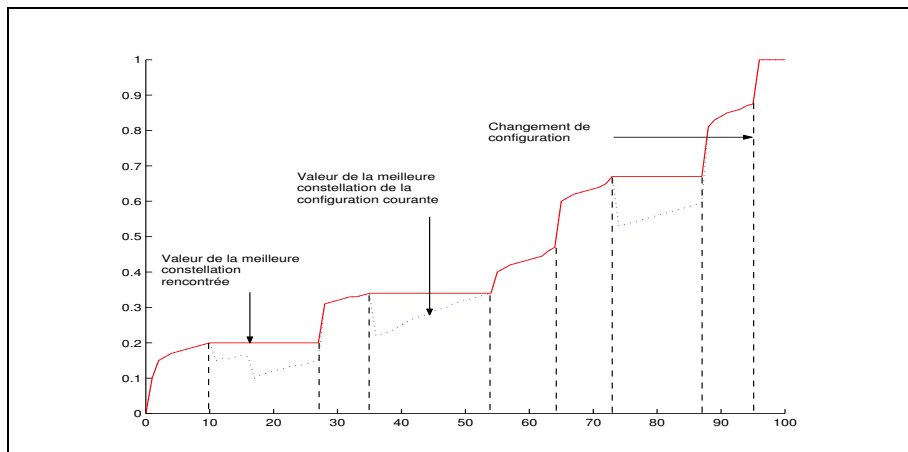


Figure 6. Evolution de la valeur d'une constellation

6. Conclusions et perspectives

L'algorithme hybride présenté ici n'est pas une simple juxtaposition de deux méthodes. Un effort particulier a été fourni pour la modélisation de l'espace de recherche afin de rendre la progression moins aveugle et plus efficace.

Les résultats constatés concernant l'évolution des paramètres de l'algorithme et des différentes valeurs associées aux éléments du système sont relativement fidèles aux attentes.

Néanmoins, la phase de transition en début de recherche peut s'avérer contraignante et

relativement longue en fonction du nombre de satellites et de classes conduisant ainsi à une divergence de l'algorithme (pas de séparation des classes possibles grâce aux valeurs a priori des configurations, ...). Le calcul des valeurs a priori semble être le point critique et on peut espérer améliorer les résultats en ayant des valeurs a priori plus fiables. Ce calcul est un calcul d'interpolation pour lequel une approche par apprentissage (réseaux de neurones par exemple [YAN 05], [SUN 04]) peut être envisagée.

Remerciements

Un remerciement à l'équipe ASP d'Alcatel Space Industries pour avoir fourni un premier cas d'application dans un contexte industriel pour le début de ce travail de recherche.

7. Bibliographie

- [BAR 04] BARICHARD V., AL, « A hybrid evolutionary algorithm for constraint satisfaction problems », *Lecture Notes in Computer science*, , 2004.
- [BOT 99] BOTELLO S., AL, « solving structural optimization problems with genetics algorithms and simulated annealing », *International Journal of Numerical Methods Engineering*, , 1999.
- [BUR 00] BURK E., SMITH, « Hybrid evolutionary techniques for the maintenance scheduling problem », *IEEE Trans. Power System*, , 2000.
- [BUR 01] BURK E., AL, « Hybrid population-based metaheuristic approaches for the space allocation problem », *Congres On Evolutionary Computation*, , 2001.
- [CHE 05a] CHELOUAH R., SIARRY P., « A hybrid method combining continuous tabu search and Nelder'Mead simplex algorithms for the global optimization of multim minima functions », *European Journal of Operational Research*, vol. 161, 2005, p. 636-654.
- [CHE 05b] TAO-MING CHENGA CHUNG-WEI FENGB Y.-L. C., « A hybrid mechanism for optimizing construction simulation models », *Automation in Construction*, vol. 14, 2005, p. 85-98.
- [CNE 89] CNES, *Techniques et Technologies des véhicules spatiaux*, Cépadués, 1989.
- [DEB 04] DEBELS D., AL, « A hybrid scatter search/electromagnetism meta-heuristic for project scheduling », *European Journal of Operational Research*, , 2004.
- [DOE 04] DOERNER K., AL., « Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection », *European Journal of Operational Research*, , 2004.
- [EDW 05] EDWARD KEEDWELL S.-T. K., « A hybrid genetic algorithm for the design of water distribution networks », *Artificial Intelligence*, , 2005.
- [FLE 87] FLETCHER R., « Practical Methods Of Optimization », *Wiley*, , 1987.
- [GLO 94] GLOVER F., « Genetic Algorithms And Scatter Search : Unsuspected Potentials », *Statistics And Computing*, , 1994.
- [GLO 97] GLOVER F., LAGUNA M., « Tabu Search », *Kluwer Academic Publishers*, , 1997.

- [GLO 01] GLOVER F., AL., « Finite Convergence Of Tabu Search », *MIC*, , 2001.
- [GOL 89] GOLDBERG D., « Genetic Algorithms In Search, Optimization And Machine Learning », *Addison Wesley*, , 1989.
- [GRA 00] GRANDCHAMP E., CHARVILLAT V., « Satellite constellation optimization with metaheuristics », *ICAI*, , 2000.
- [GRA 01a] GRANDCHAMP E., « Quelques Contribution Pour l'Optimisation de Constellations de Satellites », *Thèse de doctorat, Institut National Polytechnique de Toulouse*, , 2001.
- [GRA 01b] GRANDCHAMP E., CHARVILLAT V., « Metaheuristics to design satellite constellation », *MIC*, , 2001.
- [GRA 04] GRANDCHAMP E., *A hybrid approach to real complex system optimization*, kluwer HPCSE, 2004.
- [HON 04] HONG ZHANG M. I., « A solution to combinatorial optimization with time-varying parameters by a hybrid genetic algorithm », *International Congress Series*, vol. 1269, 2004, p. 149-152.
- [H.S 05] H. SCHMIDT G. T., « A combined heuristic optimization technique », *Advances in Engineering Software*, vol. 36, 2005, p. 11-19.
- [HSI 04] HSIAO-FAN WANG K.-Y. W., « Hybrid genetic algorithm for optimization problems with permutation property », *Computer and Operations Research*, vol. 31, 2004, p. 2453-2471.
- [JOH 89] JOHNSON D. S., AL., « Optimizzation By Simulated Annealing : An Experimental Evaluation ; Part I », 1989.
- [KAI] KAISA MIETTINEN MARKO M. MÄKELÄ H. M., « Efficient hybrid methods for global continuous optimization based on simulated annealing », *Computer and Operations Research*.
- [LAN 96] LANSARD E., AL., « Global design of satellite constellations : a multi-criteria performance comparison of classical walker patterns and nex design patterns », *International Astronautical Federation Congress*, vol. 47, 1996.
- [LIN 05] LINGYUN WEI M. Z., « A niche hybrid genetic algorithm for global optimization of continuous multimodal functions », *Applied Mathematics and Computation*, vol. 160, 2005, p. 649-661.
- [LUH 04] LUHANDJULA M., « Optimisation under hybrid uncertainty », *Fuzzy sets and systems*, vol. 146, 2004, p. 187-203.
- [MAG 01] MAGOULAS G., AL., « Hybrid methods using evolutionary algorithms for on-line training », *International Joint Conference on Neural Networks*, , 2001.
- [MAR 99] MARCO-BLASZKA N., DÉSIDÉRI J., « Numérical solution of optimization test-cases by genetic algorithms », *Rapport de recherche INRIA Sophia Antipolis*, , n° 3622, 1999.
- [MUY 03] MUYL F., AL., « Réduction des émissions de CO2 : optimisation de la traînée aérodynamique automobile par algorithmes hybrides », *CANUM*, , 2003, p. 1-6.
- [PEN 04] PENDHARKAR P. C., « Hybrid approaches for classification under information acquisition cost constraint », *Decision Support Systems*, , 2004.
- [PET 04] PETER WINKER M. G., « Applications of optimization heuristics to estimation and modelling problems », *Computational Statistics and Data Analysis*, vol. 47, 2004, p. 211-223.

- [P.G 04] P. GALINIER J.K. HAO F. S., « Hybrid evolutionary algorithm for graph coloring », *Journal of Combinatorial Optimization*, , 2004.
- [PRA 04] PRADEEP KUMAR GUDLA R. G., « An automated hybrid genetic-conjugate gradient algorithm for multimodal optimization problems », *Applied Mathematics and Computation*, , 2004.
- [REN 96] RENDERS J., FLASSE S., « Hybrid methods using genetic algorithms for global optimization », *IEEE Transactions on systems, man and cybernetics*, vol. 26, 1996, p. 243-258.
- [RIC 05] RICHARD JENSEN Q. S., « Fuzzy-rough data reduction with ant colony optimization », *Fuzzy Sets and Systems*, vol. 149, 2005, p. 5-20.
- [ROM 98] ROMAY-MERINO M., AL., « Design of high performance and cost efficient constellations for gnss-2 », *GNSS*, , 1998.
- [SCH 04] SCHLOTTMANN F., SEESE D., « A hybrid heuristic approach to discrete multi-objective optimization of credit portfolios », *Computational Statistics and Data Analysis*, vol. 47, 2004, p. 373-399.
- [SHI 04] SHI X., AL., « An improved GA and a novel PSO-GA-based hybrid algorithm », *Information Processing Letters*, , 2004.
- [SUN 04] SUNG-KWUN OHA WITOLD PEDRYCZ H.-S. P., « Multi-layer hybrid fuzzy polynomial neural networks : a design in the framework of computational intelligence », *Neurocomputing*, , 2004.
- [TRO 98] TROMPETTE P., « Optimisation hybride de la position de raidisseurs sur une plaque : AG et recherche unidirectionnelle classique », *JET*, , 1998.
- [V.B 03] V. BARICHARD J. H., AL., « A population and interval constraint propagation algorithm », *Lecture Notes in Computer science*, , 2003.
- [VIC 98] VICINI A., QUAGLIARELLA D., « Airfoilm and wing design through hybrid optimization strategies », *AIAA*, , 1998, p. 27-29.
- [WAL 70] WALKER J., « Circular Orbit Patterns Providing Whole Earth Coverage », *Royale Aircraft Establishment Technical Report*, , 1970.
- [YAN 05] YANG X., AL., « A gray-encoded, hybrid-accelerated, genetic algorithm for global optimizations in dynamical systems », *Nonlinear Science and Numerical Simulation*, vol. 10, 2005, p. 355-363.
- [YED 02] YEDDER A. B. H., « Optimisation numérique et contrôle optimal : application en chimie moléculaire », *Phd Thesis, Ecole Nationale des Ponts et Chaussées*, , 2002.
- [ZAD 65] ZADEH, AL., « Fuzzy Sets », , 1965.
- [ZNE 04] ZNE-JUNG LEE A C.-Y. L., « A hybrid search algorithm with heuristics for resource allocation problem », *Information Sciences*, , 2004.