

Etat de l'art Optimisation en Conception

3. Etat de l’art : Optimisation en Conception

“Le mathématicien ne communique pas ses résultats sous la forme où il les a trouvés ; il les réorganise, il leur donne une forme aussi générale que possible ; il fait de la “didactique pratique” qui consiste à mettre le savoir sous une forme communicable, décontextualisée, dépersonnalisée, détemporalisée.”

Guy Brousseau, 1933

3.1 Introduction

De nos jours, les techniques d’optimisation sont devenues indispensables dans plusieurs domaines de l’ingénierie, de l’économie et de l’industrie. Dans la littérature, on peut noter une croissance spectaculaire des travaux de recherche relatifs aux techniques d’optimisation et leurs applications. Dans le domaine industriel et plus particulièrement en ingénierie de conception, l’optimisation joue un rôle primordial, dans la mesure où l’objectif de son application n’est pas resté focaliser sur la minimisation du coût ou l’amélioration de certaines performances, mais il s’inscrit dans une logique de développement durable en essayant d’apporter des réponses aux questions de préservation des ressources disponibles en agissant sur tout le cycle de vie du produit.

3.2 Optimisation en conception

L’optimisation est un processus qui cherche les meilleures solutions possibles d’un problème donné, dans un domaine de recherche délimité par un ensemble de contraintes. En ingénierie de conception, ce processus est souvent pratiqué manuellement. Il implique une démarche itérative pour identifier les meilleurs paramètres de conception du produit (Roy et al., 2008), en s’appuyant toujours sur les connaissances d’experts (*expert based optimization*) ou l’adoption de plan d’expériences précis, ce qui nécessite en général un temps d’exécution relativement élevé. Devant la complexité croissante de l’activité de conception causée par la globalisation et des besoins des utilisateurs, le manque de connaissances durant les phases amont de conception, ce processus classique éprouvait certaines limites, surtout dans l’exploration complète de l’espace de recherche. Ainsi, l’optimisation algorithmique

contemporaine cherche à combler ces limites par une automatisation de ce processus traditionnel.

En conception, l’optimisation peut être introduite dès que le cahier des charges fonctionnel CdCF est défini. Son utilisation conjointement avec les outils CAX (Computer-Aided Design et Computer-Aided Engineering) donne beaucoup d’avantages aux concepteurs, pour améliorer le processus de conception d’un produit, de la phase conceptuelle jusqu’à la phase de conception détaillée (Rao, 2009).

Dans le cas de conception des produits mécaniques, on peut distinguer deux types de stratégies d’optimisation couramment utilisées (Allaire and Schoenauer, 2007; Laurent, 2013) : l’optimisation de forme et l’optimisation paramétrique.

○ **Optimisation de forme** : ce type d’optimisation consiste à trouver la forme d’une structure soumise à différents types de chargements (mécanique, aérodynamique ou thermique). Dans ce type, les variables d’optimisation sont la forme de la structure elle-même. L’optimisation de forme est essentielle dans de nombreuses applications, mais reste plus compliquée relativement à l’optimisation traditionnelle, où les variables d’optimisation sont les propriétés mécaniques de matériaux. Parmi les problèmes d’optimisation de forme, on peut distinguer trois grandes catégories, du plus “facile” au plus “difficile” (cf. Figure 3.1), optimisation de : (a) forme paramétrique, (b) de forme géométrique. (c) topologique.

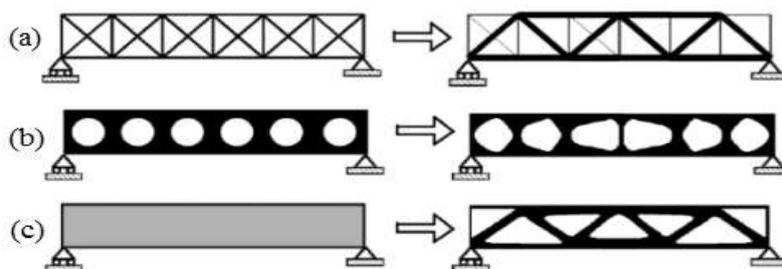


Figure 3.1 : Différents types d’optimisation de forme. (a) Optimisation de forme paramétrique, (b) Optimisation de forme géométrique. (c) Optimisation topologique (Bendsøe and Sigmund, 2013)

- a. **L’optimisation de forme paramétrique** : où les formes sont paramétrées par un nombre réduit de variables (par exemple, une épaisseur, un diamètre, des dimensions), ce qui limite considérablement la variété de formes possibles.
- b. **L’optimisation de forme géométrique** : à partir d’une forme initiale, on s’autorise des variations de la position des frontières de la forme (sans toutefois changer la topologie de la forme).

c. **L’optimisation de forme topologique** : où l’on cherche, sans aucune restriction explicite ou implicite, la meilleure forme possible quitte à changer de topologie.

o **Optimisation paramétrique** : l’optimisation paramétrique s’utilise dans un cadre beaucoup plus général. L’objectif est de chercher les meilleurs paramètres de conception au sens de certains critères (le coût, le poids, etc.). Dans le cas de conception mécanique, ce type d’optimisation ne se limite pas à l’optimisation de forme, mais il peut s’étendre aux autres facteurs (propriétés mécaniques de matériau).

3.3 Formalisation de problème d’optimisation en conception

Dans la conception du produit, dès que le cahier des charges fonctionnel est défini les concepteurs peuvent faire appel aux techniques d’optimisation. Ce cahier des charges est formulé globalement par différents objectifs (économiques, performances, ..., etc.) et contraintes (physiques, industrielles, normatives). En effet, l’optimisation joue un rôle primordial dans le processus de conception, elle sert à explorer d’une façon automatique l’espace de conception, afin d’aboutir à une solution optimale satisfaisant les besoins et les exigences exprimés dans le cahier des charges. Les ingénieurs traduisent généralement le problème de conception en un problème mathématique significatif bien que, dans la plupart des cas, cette formulation n’est pas unique.

Généralement, le processus d’optimisation en conception se déroule en quatre étapes (Figure 3.2), qui sont : 1) l’analyse du cahier des charges, 2) la modélisation du problème d’optimisation, 3) la sélection ou le développement d’une méthode (nouvelle) d’optimisation adéquate, 4) l’exploitation et l’analyse des solutions obtenues (Lei et al., 2016). L’étape de modélisation consiste à déterminer de façon précise les objectifs, les contraintes et les variables de conception appropriées au problème d’optimisation, c’est une étape cruciale et difficile et dépend de la complexité du produit et des connaissances des concepteurs.

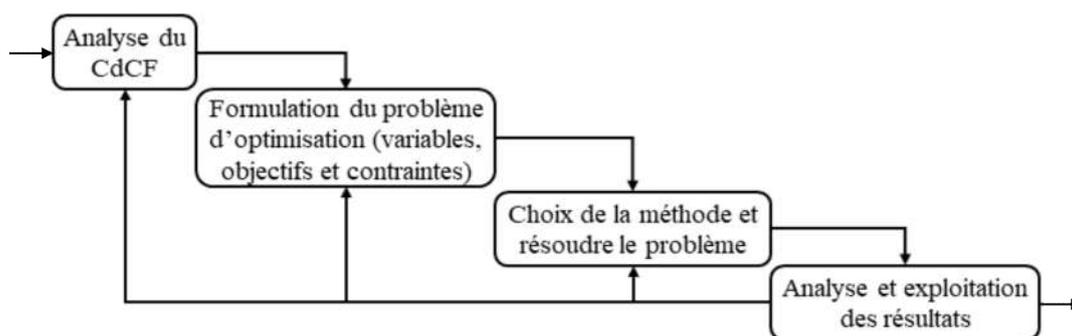


Figure 3.2 : Formulation et résolution d’un problème d’optimisation en conception

Un problème de conception optimale mono-objectif peut être formulé mathématiquement comme suit :

$$\left\{ \begin{array}{l} \min_{x \in \Omega} f(x) \\ \text{s.c.} \\ g_j(x) \leq 0; j = 1, \dots, l \\ h_k(x) = 0; k = 1, \dots, p \\ x_i^{\min} \leq x_i \leq x_i^{\max}; i = 1, \dots, n \end{array} \right. \quad (3.1)$$

où, n est la dimension de l’espace de recherche ; $g_j(x)$ est la $j^{\text{ème}}$ contrainte d’inégalité, $h_k(x)$ est la $k^{\text{ème}}$ contrainte d’égalité et $f(x)$ est la fonction objectif (coût).

Pour les problèmes multi-objectifs plusieurs critères sont optimisés simultanément. Dans ce cas, on cherche à optimiser un vecteur de m fonctions objectifs et la formulation devient comme suit :

$$\left\{ \begin{array}{l} \min_{x \in \Omega} f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.c.} \\ g_j(x) \leq 0; j = 1, \dots, l \\ h_k(x) = 0; k = 1, \dots, p \\ x_i^{\min} \leq x_i \leq x_i^{\max}; i = 1, \dots, n \end{array} \right. \quad (3.2)$$

3.4 Classification des problèmes d’optimisation

Avant d’aborder le traitement d’un problème d’optimisation, il est important d’identifier sa nature afin de choisir la (les) technique(s) de résolution les plus appropriées. Dans cette section, nous allons présenter une classification actualisée de celle développée par Roy et al. (Roy et al., 2008). La classification se fera selon plusieurs éléments expliqués ci-dessous.

Variables de conception : La majorité des problèmes d’optimisation en conception sont multidimensionnels et leur formulation est une activité complexe qui nécessite des connaissances d’experts dans le domaine. Le choix de la méthode de résolution adéquate et la qualité de solution(s) optimale(s) sont généralement très affectés par la nature et le nombre de variables de conception considérées. Ces variables peuvent être indépendantes du temps, c’est le cas de l’optimisation statique et prennent dans ce cas des valeurs entières, discrètes, réelles

ou mixtes. Ces variables peuvent être aussi dépendantes du temps, c’est le cas dans le contexte de l’optimisation dynamique (calcul des variations).

Existence de contraintes : Dans le contexte de la conception, les contraintes définissent les limites fonctionnelles du système ou des lois régissant les phénomènes physiques. Elles peuvent se présenter sous formes linéaires ou non-linéaires, de type égalité ou inégalité. Leur présence rend le problème d’optimisation difficile, affecte la phase de traitement et influence significativement le temps de résolution.

Fonctions objectifs : représentent la forme explicite des performances du système à optimiser (minimiser ou maximiser). Elles sont utilisées pour évaluer les solutions tirées de l’espace de conception. Leur nature et nombre sont des facteurs déterminants de la complexité du problème d’optimisation considéré. En outre, ces fonctions peuvent être classifiées comme des fonctions séparables, non séparables, linéaires, non linéaires, dérivables, non dérivables, convexes, non convexes, monomodales ou multimodales.

Le problème d’optimisation devient très complexe quand le nombre des fonctions objectifs dépasse dix (Corne and Knowles, 2007). Les fonctions objectifs peuvent prendre des formes quantitatives ou qualitatives. Les fonctions quantitatives sont basées généralement sur des modèles analytiques, empiriques (expérimentales) ou simulations numériques (FEM, CFD). Les fonctions qualitatives sont généralement invoquées dans les domaines de la fabrication, l’ergonomie et l’esthétique (des critères non quantifiables).

Domaine d’étude : la majorité des problèmes abordés en conception optimale sont de nature multi-physiques, ce qui nécessite réellement des approches d’optimisation multidisciplinaire (Tomiya et al., 2007). Le couplage entre les domaines physiques et le déroulement du processus d’optimisation est souvent effectué en parallèle, et parfois en mode distribué, comme c’est le cas dans le domaine de la conception aéronautique, le processus d’optimisation devient ainsi plus complexe et nécessite beaucoup d’efforts.

Environnement d’optimisation : l’environnement de conception est caractérisé par la présence d’incertitudes (Yang et al., 2007) qui parviennent généralement de différentes sources (section 2.6), elles traduisent le manque de connaissances sur l’espace de conception et l’optimum recherché. En effet, les variables de conception sont affectées par les imperfections des procédés de fabrication, des moyens de mesure, des modèles de comportement adoptés et des approximations apportées. Par conséquent, il est important d’impliquer les concepteurs et les

experts dans le processus de conception et d’optimisation d’un système afin d’augmenter la fiabilité du processus d’optimisation.

Tableau 3.1 : Classification des problèmes d’optimisation

| Caractéristique | | Propriété | Classification du problème d’optimisation |
|---|-------------|--|---|
| Nombre de variables | | =1 | monovariable, recherche linéaire |
| | | >1 | à plusieurs variables, multidimensionnelle |
| Type des valeurs admissibles pour les variables d’optimisation | | Nombres réels continus | continue |
| | | Nombres réels continus + entiers | mixte |
| | | Entiers en permutation | Combinatoire, discrète ou en nombres entiers |
| Nature des Fonctions du problème | objectif | linéaire/ X | Linéaire |
| | contraintes | linéaire/ X | |
| | objectif | quadratique/X | quadratique |
| | contraintes | linéaire/ X | |
| | objectif | non linéaire/ X | Non linéaire (il suffit qu’une seule condition soit vérifiée) |
| | contraintes | non linéaire/ X | |
| | objectif | continue / x (ou pas) | différentiable / non différentiable |
| | contraintes | crée un ensemble convexe | Programmation convexe |
| objectif | convexe | | |
| Nature physique du problème | | Fonctionnelles (X(t), u(t)), problème à plusieurs phases | Commande optimale, Calcul des variations,... |
| | | Autre | Autre |
| Nature des variables d’optimisation | | Variables stochastiques | Programmation stochastique |
| | | Variables déterministes | Programmation déterministe |
| Nombre de fonctions objectifs | | 1 | Mono-objectif |
| | | >1 | Multi-objectif |
| Formulation du problème | | Avec contraintes | Sous contraintes (pb contraint) |
| | | Sans contraintes | Sans contraintes (pb non contraint) |

3.5 Les méthodes d’optimisation monoobjectif

Les problèmes d’optimisation qui se posent en conception sont de natures très variées et le choix de la méthode d’optimisation appropriée dépend fortement de la formulation du problème considéré. En Effet, la nature des variables de conception (continues, discrètes ou

mixtes) ainsi que celle de la fonction coût (linéaire, quadratique ou non-linéaire), la présence ou non des contraintes, l’existence ou non de plusieurs objectifs, sont tous des facteurs affectant le choix de la technique d’optimisation. La Figure 3.3 présente une classification des principales méthodes d’optimisation mono-objectif. Deux grandes familles de méthodes se partagent la solution de ces problèmes : les méthodes combinatoires et les méthodes continues. Sous ces deux intitulés, plusieurs techniques ont été développées dans la littérature. En fait, les méthodes d’optimisation de problèmes continus ne peuvent pas résoudre efficacement les problèmes combinatoires. De même, les méthodes dédiées aux problèmes sans contraintes ne sont pas directement applicables à ceux avec contraintes. En outre, il n’existe pas d’algorithme efficace pour tous les problèmes ; quand un algorithme progresse sur une classe de problèmes, il régresse sur une autre. Dans la suite, nous allons présenter brièvement les méthodes d’optimisation les plus utilisées pour trouver l’optimum global d’un problème d’optimisation de conception. En particulier on parlera des méthodes déterministes (locales) et des méthodes stochastiques ou métaheuristiques (globales).

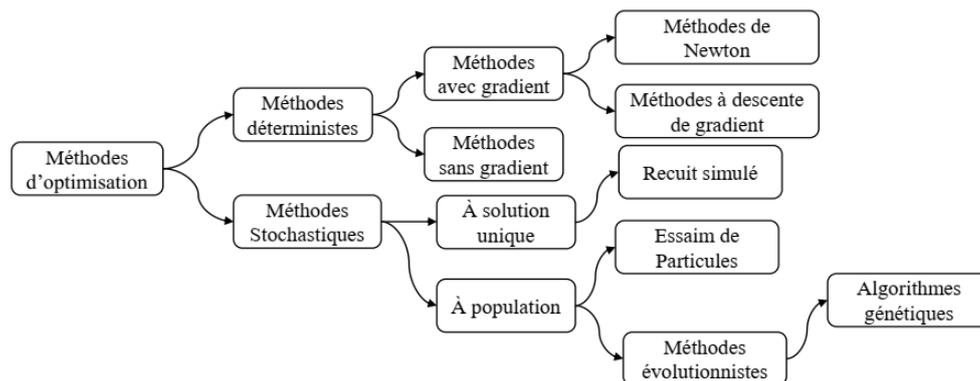


Figure 3.3 : Classification des méthodes d’optimisation

3.5.1 Méthodes déterministes

Les méthodes déterministes (locales) se caractérisent par une exploration méthodique de l’espace de recherche selon un processus déterministe prédéfini. Elles reproduisent la même solution pour le même jeu de paramètres d’entrée. Une première classe qui peut être introduite dans ce cadre-là est celle des méthodes analytiques (appelées également indirectes) qui cherchent l’optimum en résolvant un système d’équations (généralement non-linéaires) issus de conditions dites d’optimalité (nullité du gradient, conditions de KKT, ...). L’utilisation de telles méthodes est généralement réservée aux cas des problèmes relativement simples. Une deuxième classe de méthodes qui appartient à cette catégorie est des méthodes dites du gradient (méthodes numériques directes). Ces méthodes génèrent un ensemble d’itérations sur la base de directions de recherche synthétisées sur la base de gradients et de Hessiens des

fonctions objectif et contraintes. Malheureusement ces méthodes ont le plus souvent un comportement local, c.-à-d. qu'elles convergent vers l'optimum le plus proche de la solution initiale. Les méthodes déterministes peuvent être très efficaces pour des problèmes de grande taille, souvent multi-modaux. Dans, ce cas, la probabilité de trouver l'optimum global dépend essentiellement d'une bonne connaissance du problème.

Par ailleurs, les méthodes déterministes peuvent être classées selon l'ordre qui caractérise les informations utilisées pour mettre en œuvre la stratégie de recherche. Ainsi, on distinguera les méthodes du zéro ordre, premier ordre et du second ordre.

Les algorithmes d'optimisation d'ordre zéro sont ceux qui ne requièrent que l'évaluation de la fonction objectif pour explorer l'espace de recherche. De ce fait, ce genre de techniques est particulièrement intéressant pour l'optimisation des fonctions non dérivables ou nécessitant un effort calculatoire important pour l'obtention des dérivées. A titre d'exemple, nous pouvons citer :

- Les méthodes de Powell et de Nelder-Mead.
- Les méthodes énumératives intelligentes comme celle de séparation et d'évaluation (Branche and bound)

Par ailleurs, les méthodes d'optimisation du premier ordre utilisent le gradient ou une approximation du gradient pour définir les directions de recherche. C'est le cas notamment des méthodes de la plus grande pente et des gradients conjugués. Enfin, les méthodes d'optimisation du second ordre utilisent, en plus du gradient, le Hessien de la fonction objectif (ou une approximation) pour construire les éléments de recherche. Parmi les méthodes du 2^{ème} ordre, nous citons la méthode de Newton et les méthodes quasi-Newtoniennes.

3.5.1.1 Méthodes à descente de gradient

Les méthodes à descente de gradient sont les plus anciennes parmi les méthodes d'optimisation, elles nécessitent que la fonction objectif est définie analytiquement et différentiable dans l'espace de recherche, leur processus de recherche est itératif. Pour chaque itération il détermine la direction du gradient $\nabla f(x)$ et fait un déplacement à la prochaine itération dans la direction opposée en décroissant la fonction objectif (dans le cas de minimisation). Ces méthodes présentent deux inconvénients : 1) converger facilement vers des minima locaux si la fonction objectif est multimodale, donc le choix de la position initiale est important, 2) diverger rapidement si la fonction objectif est bruitée (car le déplacement

dépend du gradient). L'Algorithme 3.1 illustre le principe de base de fonctionnement d'une méthode à descente de gradient.

Algorithme 3.1 : Algorithme de la plus grande descente

Initialize: $x_1 \in \mathbb{R}^n, i \leftarrow 1$

1 While stopping criterion not verified **do**

2 $\alpha_i = \arg \min_{\alpha \in \mathbb{R}^+} f(x_i, \alpha \nabla f(x_i))$

3 $x_{i+1} \leftarrow x_i - \alpha_i \nabla f(x_i)$

4 $i \leftarrow i + 1$

α_i , définit le pas de l'algorithme à chaque itération, selon lequel on peut distinguer plusieurs variantes dans la littérature, comme l'algorithme de gradient à pas constant, l'algorithme de gradient à pas optimal (plus grande pente) (Cauchy, 1847) et l'algorithme de gradients conjugués (Hestenes and Stiefel, 1952).

3.5.1.2 Méthodes de Newton

Les méthodes de Newton consistent à annuler le gradient de la fonction objectif ($\nabla f(x) = 0$), nous présentons brièvement deux méthodes. La première est la méthode de Newton, son principe est que à chaque itération k , une approximation quadratique de la fonction objectif f au voisinage du point x_k est construite et minimisée. Elle requière certaines hypothèses telles que : la fonction objectif soit continue et de classe C^2 , et le Hessien $\nabla^2 f(x)$ soit inversible. Cette méthode cherche à résoudre itérativement la relation suivante :

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k) \right]^{-1} \nabla f(x_k) \quad (3.3)$$

L'inconvénient de cette méthode est que, à chaque itération du calcul, l'inversion de la matrice hessienne est nécessaire, ce qui engendre un temps d'exécution trop élevé. La deuxième est la méthode Quasi-Newton, qui exploite le gradient de la fonction objectif uniquement, l'inversion de la matrice hessienne est calculé par les méthodes DFP (Davidon, 1991; Fletcher and Powell, 1963) ou BFGS (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970).

3.5.2 Les méthodes stochastiques (méta-heuristiques)

Une méthode stochastique est une méthode basée entièrement ou en partie sur un processus aléatoire. Elle utilise uniquement les valeurs de la fonction coût (méthode d'ordre zéro) pour trouver une solution approchée de l'optimum global. Elle est particulièrement recommandée dans le cas de problèmes discontinus ou fortement multimodaux pour éviter le piégeage dans les minima-locaux. La convergence de l'algorithme est assurée soit par une heuristique (méthode de Monte-Carlo) soit par une méta-heuristique (algorithmes évolutionnaires, recuit simulé, recherche tabou, colonies de fourmis, etc.). Les techniques stochastiques d'optimisation peuvent être classées globalement en deux catégories, des méthodes stochastiques à population et des méthodes stochastiques à parcours.

Les méta-heuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération, elle est plutôt axée sur l'exploitation de l'espace de recherche. La notion de voisinage est alors primordiale. Les plus connues dans cette classe sont le recuit simulé, la recherche avec tabous, la recherche à voisinage variable, la méthode GRASP ou encore les méthodes de bruitage. Par contre, dans les méthodes à population, à chaque itération, la méta-heuristique manipule un ensemble de solutions en parallèle. Elle est plutôt exploratoire et permet une meilleure diversification de l'espace de recherche. On peut citer les algorithmes génétiques, l'optimisation par essais particuliers et les algorithmes de colonies de fourmis.

3.5.2.1 Les méta-heuristiques à solution unique

Cette catégorie, commence à partir d'une solution initiale choisie aléatoirement dans l'espace de recherche et converge progressivement vers une solution optimale en formant une trajectoire dans l'espace de recherche (Boussaïd et al., 2013), cette catégorie englobe essentiellement les méthodes suivantes : la méthode de recuit simulé, la méthode de recherche tabou, la méthode de descente et la méthode GRASP.

a. La méthode du recuit simulé

La méthode du recuit simulé (*Simulated annealing*) a été proposée par Kirkpatrick (Kirkpatrick et al., 1983) en 1983 et indépendamment par Černý (Černý, 1985) en 1985. Le principe de cette méthode est inspiré d'un processus utilisé en métallurgie, qui est le recuit physique en s'appuyant essentiellement sur les lois énoncées par Boltzmann en thermodynamiques. Le recuit en métallurgie est un processus qui consiste à réorganiser la structure cristallographique des matériaux par des cycles alternés de refroidissement lent et de réchauffage du matériau, tout en conduisant à minimiser son énergie. A chaque cycle, le

matériau est chauffé et maintenu à haute température jusqu'à l'équilibre thermodynamique s'atteint, puis d'abaisser cette température lentement afin d'obtenir une structure à l'état solide bien ordonnée et d'énergie minimale, tout en évitant les états métastables (caractéristiques des minima locaux de l'énergie).

Ce processus est transposé en optimisation pour résoudre des problèmes d'optimisation, dans lesquels la fonction objectif est assimilée à l'énergie d'un matériau, est alors minimisée à l'aide d'une température effective introduite à l'algorithme, est considérée comme son paramètre contrôlable.

L'algorithme démarre à partir d'une solution initiale générée de manière aléatoire. A chaque itération, une autre solution s' se génère aléatoirement dans le voisinage de la solution actuelle s . La solution s' est acceptée si sa performance est meilleure à celle de la solution courante s (i.e., $f(s') \leq f(s)$). Sinon, la solution s' est acceptée avec une probabilité $\exp\left(\frac{\Delta f}{T}\right)$ qui dépend de deux facteurs : 1) l'importance de la dégradation de la fonction objectif ($\Delta f = f(s') - f(s)$) – des dégradations faibles sont plus favorables à l'acceptation ; 2) la température T – une température élevée signifie que la probabilité d'accepter des dégradations est plus grande.

Dans la méthode SA, la température contrôle les deux mécanismes d'*intensification* et de *diversification*. Elle décroît durant le processus de recherche, de manière qu'à la fin de l'algorithme cette recherche s'intensifie. L'idée est de diminuer progressivement et lentement la chance d'acceptation des solutions dégradant la fonction objectif.

3.5.2.2 Les méta-heuristiques à base de population

Les méta-heuristiques à population de solutions consistent à améliorer une population d'individus, au fur et à mesure des itérations, contrairement aux méta-heuristiques de trajectoire (cités ci-dessus), nous allons présenter dans cette section deux catégories qui sont largement utilisées en l'ingénierie de conception. Il s'agit des algorithmes évolutionnaires, qui sont une catégorie d'algorithmes basée sur la théorie d'évolution par la sélection naturelle de Darwin (Darwin, 1956), et les algorithmes d'intelligence en essaim qui sont inspirés des phénomènes naturels.

a. Les algorithmes évolutionnaires

Les algorithmes évolutionnaires AEs (EC : Evolutionary Computation en anglais), qui regroupent une famille d'algorithmes sont inspirées de la théorie de l'évolution darwinienne

afin de résoudre des problèmes complexes divers (Boussaïd et al., 2013). Selon la théorie du naturaliste Charles Darwin, énoncée en 1859 (Darwin, 1956)), l’évolution des espèces est la conséquence de la conjonction de deux phénomènes qui sont : 1) la sélection naturelle, favorisant les individus les plus adaptés à leur milieu à survivre et à se reproduire, laissant une descendance qui transmettra leurs gènes, 2) la présence de variations non dirigées parmi les traits génétiques des espèces (mutations).

ECs regroupent un ensemble de méthodes méta-heuristiques telles que les algorithmes génétiques (Holland, 1975), les stratégies d’évolution (Rechenberg, 1965), la programmation évolutive (Fogel, 1991), et la programmation génétique (Koza and Koza, 1992). La Figure 3.4 montre le principe général du fonctionnement d’algorithmes évolutionnaires.

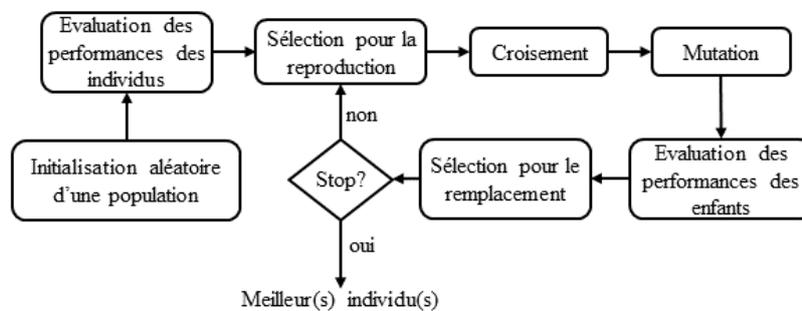


Figure 3.4 : Principe d’algorithmes évolutionnaires (EA) (Dréo et al., 2003)

ECs commencent à partir d’une population initiale générée aléatoirement dans l’espace de recherche. Ensuite, à chaque itération (génération) de l’algorithme une autre population est générée et constituée de plusieurs individus, qui représentent des solutions potentielles du problème d’optimisation, est capable de se reproduire pour les prochaines itérations. Cette population est susceptible à des variations génétiques et à la contrainte de l’environnement étant simulée à l’aide de la fonction d’adaptation, cette contrainte provoque la sélection naturelle des individus, c.à.d. la survie du plus fort individu. Les opérateurs de variation (croisement et mutation) sont appliqués avec une probabilité définie aux individus sélectionnés (parents), ils génèrent de nouveaux descendants (enfants ou offsprings), la mutation et le croisement désignent les opérateurs unaires et binaires (ou n-aires) respectivement. Les individus obtenus par ces opérateurs forment alors une nouvelle population (parents). Le processus d’évolution est itératif, et se termine lorsque les conditions d’arrêt soient vérifiées.

a.1 Les algorithmes génétiques

Les algorithmes génétiques (GA : Genetic Algorithms en anglais) sont la méthode la plus populaire et largement utilisée des EAs. Leurs origines remontent au début des années 1970, avec les travaux de John Holland et ses élèves à l’Université du Michigan en étudiant les systèmes adaptatifs (Holland, 1975). Par la suite, en 1989 D. E. Goldberg (Goldberg and Deb, 1991) a publié un ouvrage de référence sur ces algorithmes qui a fortement participé à leur essor. Ces algorithmes se détachent en grande partie par la représentation des données du génotype, initialement sous forme d’un vecteur binaire et plus généralement sous forme d’une chaîne de caractères.

Pour chaque étape de GA, il y a un opérateur associé qui décrit la façon de manipuler les individus :

- Opérateur de sélection : Pour sélectionner les individus qui sont plus favorables à se reproduire. Il existe plusieurs techniques de sélection, les plus utilisées sont la sélection par tirage à la roulette (roulette-wheel selection), la sélection par tournoi (tournament selection) et la sélection par rang (ranking selection), etc (Blickle and Thiele, 1995; Goldberg and Deb, 1991).
- Opérateur de croisement : consiste à combiner les caractéristiques d’un ensemble d’individus parents (généralement deux), qui sont préalablement sélectionnés en générant de nouveaux individus enfants. Il existe plusieurs types d’opérateur de croisement, par exemple le croisement en un point, le croisement en n-points ($n \geq 2$) et le croisement uniforme (cf. Figure 3.5).
- Opérateur de mutation : consiste à modifier aléatoirement une partie du génotype des individus enfants (descendants).
- Opérateur de remplacement : consiste à sélectionner les survivants. Autrement dit, il remplace certains des parents par certains des enfants. Le cas plus général est de choisir les individus de la population, suivant leurs performances, pour former une nouvelle population de même taille que celle à l’itération précédente.

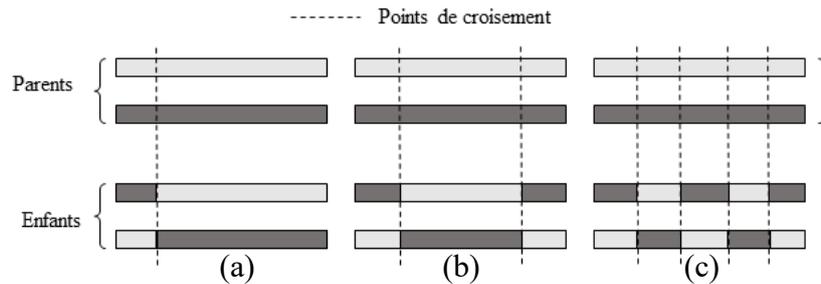


Figure 3.5 : Exemples de croisement : (a) croisement simple en un point, (b) croisement en deux points, (c) croisement uniforme

b. L’intelligence en essaim

L’intelligence en essaim (SI : Swarm Intelligence en anglais) a été développée dans le cadre de la modélisation mathématique et informatique des phénomènes biologiques rencontrés en éthologie (Bonabeau et al., 1999). Elle englobe un ensemble de méthodes à base de population, chaque individu dans la population représente un agent simple (entité capable d’exécuter certaines opérations), qui peut interagir localement avec les autres agents et avec son environnement. Ces agents, dont la capacité individuelle est très restreinte, qui peuvent conjointement effectuer de nombreuses tâches complexes pour assurer leur survie. Bien qu’il n’existe pas de structure de contrôle centralisée qui indique la manière dont les agents individuels devraient se comporter, les interactions locales entre ces agents amènent souvent à l’émergence d’un comportement collectif global et auto-organisé.

b.1 L’optimisation par Essaim particulière

L’optimisation par essaim particulière (PSO : Particle Swarm Optimization en anglais) a été développée en 1995 par Russel Eberhart et James Kennedy (Kennedy and Eberhart, 1995), mais ses origines remontent aux travaux de C. Reynolds (Reynolds, 1987) et de Heppner et Grenander (Heppner and Grenander, 1990) qui ont développé des modèles mathématiques permettent de simuler un vol de groupe d’oiseaux et bancs de poissons. Cette méthode est inspirée des déplacements collectifs remarquables chez certains animaux sociaux comme les poissons et les oiseaux migrateurs ayant aptitude à imiter les comportements réussis qu’ils observent dans leur entourage, tout en y apportant leurs modifications personnelles.

L’essaim particulière représente une population d’agents appelés « particules ». Chaque particule est modélisée comme une solution potentielle du problème d’optimisation, elle parcourt l’espace de recherche, en cherchant de l’optimum global du problème.

Son déplacement dans l’espace de recherche, est influencé généralement par trois composantes (cf. Figure 3.6) : 1) une composante physique : la particule tend à suivre sa

direction actuelle de déplacement ; 2) une composante cognitive : la particule tend à s’orienter vers le meilleur site par lequel elle est déjà passée ; 3) une composante sociale : la particule tend à se fier à l’expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins.

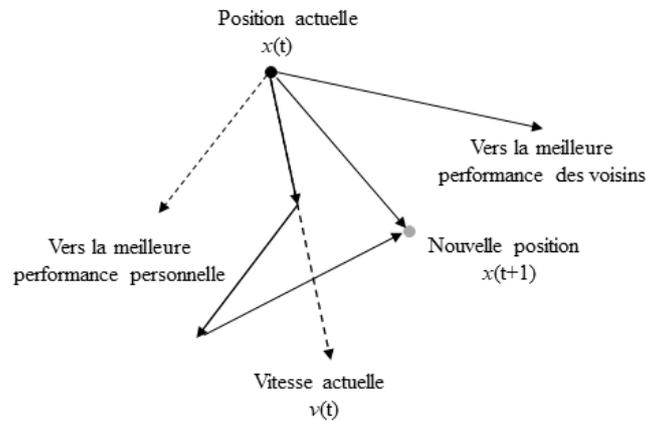


Figure 3.6 : PSO-Déplacement d’une particule dans l’espace de recherche

Le voisinage est défini spatialement par la détermination de la distance euclidienne entre les positions de deux particules, et qui permet de déterminer la position de l’individu dans l’essaim (Kennedy and Martins, 2014). Chaque particule i de l’essaim est caractérisée dans l’espace par vecteur de position x_i et par un vecteur de changement de position v_i (vitesse). Chaque particule dispose d’une mémoire lui permettant de se souvenir de sa meilleure solution découverte par le passé, noté p_i (personal best) et de la meilleure position connue de son voisinage, notée p_g (global best). À chaque itération, chaque particule se déplace dans l’espace de recherche en suivant un vecteur, calculé comme somme pondérée des vecteurs représentant sa vitesse courante $v_i(t)$, ainsi que sa p_i et sa p_g (cf. Equation (3.4) et Figure 3.6). Sa nouvelle vitesse $v_i(t+1)$ est déterminée de la façon suivante :

$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + c_1 \varphi_1 (\vec{p}_i(t) - \vec{x}_i(t)) + c_2 \varphi_2 (\vec{p}_g(t) - \vec{x}_i(t)) \quad (3.4)$$

où $i = 1, \dots, n$, et n est la taille de la population (taille de l’essaim) ; ω est le coefficient d’inertie permet de contrôler l’influence de la vitesse obtenue au pas précédent (Shi and Eberhart, 1998). Un grand coefficient d’inertie provoque une grande amplitude de mouvement tandis qu’un petit coefficient d’inertie concentre la recherche sur un petit espace ; φ_1 et φ_2 sont deux paramètres aléatoires formés uniformément dans l’intervalle $[0,1]$, c_1 et c_2 sont deux constantes représentant une accélération positive. Elles correspondent respectivement à

la composante cognitive et sociale du déplacement. La position de chaque particule dans l’espace est mise à jour à chaque itération par l’équation suivante :

$$\bar{x}_i(t+1) = \bar{x}_i(t) + \bar{v}_i(t) \quad (3.5)$$

Les particules peuvent passer éventuellement à côté de l’optimum sans le détecter, à cause de leur déplacement rapide dans l’espace de recherche. Afin de remédier ce problème, une vitesse maximale doit être fixer, noté v_{\max} , de telle sorte que chaque composante de \bar{v}_i appartient toujours dans l’intervalle $[-v_{\max}, +v_{\max}]$ (Eberhart et al., 1996). Le choix de v_{\max} est une tâche délicate car ce paramètre influence énormément sur la balance *exploration/exploitation*. Un coefficient de constriction χ peut être utilisé en permettant de s’affranchir de la définition de la vitesse maximale v_{\max} (Clerc and Kennedy, 2002).

3.6 L’optimisation multi-objectif

Les problèmes d’optimisation mono-objectif ne cherchent à optimiser qu’une seule fonction objectif à la fois. Cependant, la plupart des cas réels en conception des produits, sont en fait des problèmes multi-objectifs, dans lequel plusieurs critères sont optimisés simultanément. Ces critères, dans la majorité des cas sont antagonistes (par exemple, on cherche à maximiser la fiabilité d’un produit tout en minimisant le coût de développement). Ainsi, il n’existe pas une solution unique au problème, mais un ensemble de solutions, qui représentent finalement des compromis entre les différentes fonctions objectifs, L’optimisation multi-objectif (dites aussi multicritères ou vectorielle) trouve ses racines au cours du XIXème siècle dans les travaux en économie d’Edgeworth (Edgeworth, 1885) et de Pareto (Pareto, 1896).

Dans la Figure 3.7, Y est l’espace des objectifs (ensemble des objectifs réalisables), qui représente l’image de l’espace de décision X (ensemble des points réalisables) par la fonction objectif.

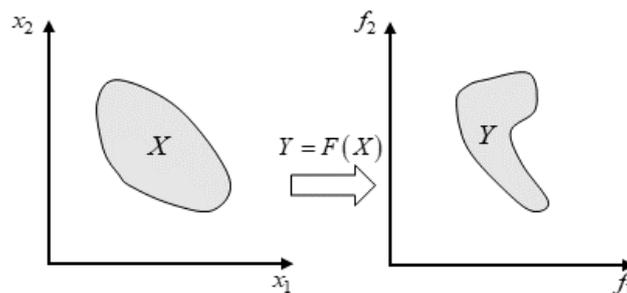


Figure 3.7 : Correspondance entre l’espace de décision et l’espace des objectifs

3.6.1 Dominance de Pareto et optimalité

L’optimisation multi-objectifs a pour objectif principal de trouver une solution optimale au problème décrit en Equation (3.2). Pour résoudre ce problème, la plupart des algorithmes d’optimisation multi-objectifs modernes utilisent le concept de dominance de Pareto. Le principe de dominance consiste à comparer deux solutions afin de déterminer si l’une d’elles domine l’autre ou non. La dominance de Pareto est définie comme suit :

Définition 3.1 (*Pareto dominance*)

Soit x_a et x_b sont deux solutions réalisables différentes du problème d’optimisation multi-objectif. Alors x_a domine x_b (notée $x_a \leq x_b$) si et seulement si:

$$f_i(x_a) \leq f_i(x_b) \quad \forall i \in \{1, \dots, m\} \quad (3.6)$$

et

$$\exists k \in \{1, \dots, m\} \quad f_k(x_a) < f_k(x_b) \quad (3.7)$$

Définition 3.2 (*solution Pareto-optimal*)

Une solution x^* est dite Pareto-optimal s’il n’existe pas une solution qui la domine:

$$\nexists x \in \Omega : x \leq x^* \quad (3.8)$$

Définition 3.3 (*ensemble Pareto-optimal*)

Pour un problème d’optimisation multi-objectif donné, l’ensemble X^* de toutes les solutions de Pareto-optimal, est dites l’ensemble de Pareto-optimal et définies comme suit :

$$X^* = \left\{ \exists x^* \in \Omega / \nexists x \in \Omega, f(x) \prec f(x^*) \right\}$$

Définition 3.4 (*front de Pareto-optimal*)

Le front de Pareto-optimal est l’ensemble F^* des solutions réalisables qui correspondent aux solutions :

$$F^* = \left\{ f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \text{ tels que: } x^* \in X^* \right\} \quad (3.9)$$

3.6.2 Classification des méthodes d’optimisation multi-objectif

Dans la littérature, nous pouvons rencontrer deux types de classification des méthodes de résolution des problèmes d’optimisation multi-objectif. La première classification adopte un point de vue décideur. La deuxième classification adopte un point de vue du concepteur, que nous détaillons dans la suite de cette section.

Décideur : les méthodes d’optimisation sont classées en trois familles suivant la coopération entre la méthode et le décideur :

- Les méthodes *a priori* : Le décideur intervient en amont de la méthode d’optimisation. Il définit ses préférences en pondérant par exemple les objectifs selon leurs importances.

- Les méthodes *progressives* ou *interactives* : Le décideur affine son choix, en écartant ou en favorisant des solutions, au fur et à mesure du déroulement du processus de résolution.
- Les méthodes *a posteriori* : Le décideur intervient en aval de l’optimisation. Il choisit la solution qui lui semble la plus intéressante parmi l’ensemble des solutions fournies par la méthode d’optimisation.

Concepteur : Les méthodes d’optimisation sont classées en trois familles en fonctions de la manière dont elles traitent la préférence de plusieurs objectifs :

- Les méthodes agrégées,
- Les méthodes basées sur l’équilibre de Pareto
- Les méthodes non agrégées et non Pareto.

3.6.2.1 Méthodes agrégées

En général, les méthodes agrégées consistent à rendre un problème d’optimisation multi-objectif en un problème mono-objectif, en regroupant les fonctions objectifs à optimiser dans une seule fonction objectif. Dans ce cadre, une fonction de mise à l’échelle de chaque fonction objectif peut être utilisée afin de permettre les additionner (modèle additif) ou bien les multiplier (modèle multiplicatif), Dans cette catégorie nous présentons la méthode la plus populaire, qui est la méthode de moyenne pondérée (Weighting Method).

La méthode de la moyenne pondérée est la plus connue dans le cadre de l’optimisation multi-objectif (Gass and Saaty, 1955; Zadeh, 1963), et la plus utilisée pour générer les solutions Pareto optimales. Elle consiste à transformer un problème d’optimisation multi-objectif en un problème mono-objectif en additionnant toutes les critères après avoir affecter un poids à chacun. Le problème d’optimisation obtenu conserve sa linéarité si toutes les fonctions objectifs et les contraintes sont linéaires, et peut être résolu facilement en sélectionnant une méthode d’optimisation adéquate :

$$\min_x \sum_{i=1}^n w_i f_i(x) \quad (3.10)$$

où les poids vérifient : $w_i > 0$ et $\sum_{i=1}^n w_i = 1$.

Cette méthode peut être utilisée comme *a posteriori* ou *a priori* (Sengupta et al., 2016). Bien qu’elle soit largement utilisée, elle présente toutefois quelques problèmes. D’une part, le poids de chaque critère doit être déterminé a priori par le décideur, ce que requière certaines

connaissances sur le problème. D'une autre part, l'interaction entre les différents critères doit être exprimée, pour certain problème cette tâche parfois est devenue plus difficile (Marichal, 2002). En outre, cette méthode ne peut pas accéder à certaines solutions (cf. Figure 3.8b) dans le cas où le front de Pareto est non convexe (Marler and Arora, 2004). Enfin, cette méthode pourrait trouver des solutions qui ne reflètent pas vraiment les préférences du décideur.

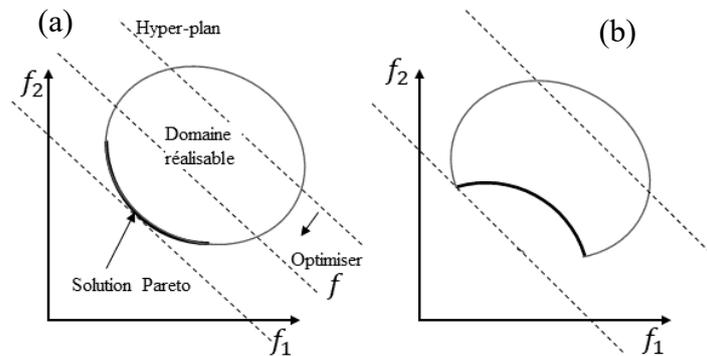


Figure 3.8 : Méthode d'agrégation : (a) Front de Pareto convexe, (b) Front de Pareto non convexe

3.6.2.2 Méthodes basées sur l'équilibre de Pareto

Le principe de ces méthodes s'appuie sur le postulat de V. Pareto (Pareto, 1896) : « *Il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critères* ». Les méthodes d'optimisation qui transforment un problème d'optimisation multi-objectif en mono-objectif sont dépassées au profit de la notion d'ensemble de solutions Pareto optimales, suivant le critère de dominance au sens de Pareto. En effet, l'ensemble des meilleurs compromis réalisables entre les objectifs contradictoires forme ce qu'on appelle le front de Pareto. Dans lequel, il n'existe aucune solution étant strictement meilleur que les autres. L'objectif de méthodes basées sur l'équilibre de Pareto, est d'identifier cet ensemble de compromis optimaux entre les critères. La Figure 3.9 présente un exemple de front de Pareto. Les solutions appartenant à ce front ne peuvent pas être comparées, car aucune solution n'est systématiquement meilleure que les autres sur tous les critères. Donc, le rôle du décideur est de choisir la ou les solutions à retenir. Notez que dans certains cas de conception, le front de Pareto ne prend pas toujours une forme régulière, peut être concave, discontinu, etc.

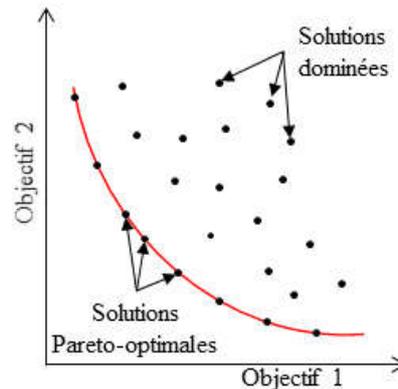


Figure 3.9 : Front de Pareto pour un problème d'optimisation bi-objectif

Dans ce cadre, on trouve plusieurs algorithmes génétiques qui ont été développées afin de résoudre les problèmes d'optimisation multi-objectif. Parmi ces algorithmes les plus significatifs sont : l'algorithme Multi Objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993), l'algorithme Niched Pareto Genetic Algorithm (NPGA) (reynolds Horn et al., 1994), qui utilise une sélection par tournoi, basée principalement sur la dominance de Pareto, l'algorithme Non Dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1994) et enfin l'algorithme NSGA-II (Deb, 2000) qui est l'algorithme le plus populaire et le plus utilisé.

a. Nondominated Sorting Genetic Algorithm II (NSGA-II)

L'algorithme NSGA-II « Nondominated Sorting Genetic Algorithm II », (Deb et al., 2002), est l'un des algorithmes génétiques les plus populaires et les plus utilisés pour résoudre une large palette de problèmes d'optimisation multi-objectif dans différents domaines d'ingénierie.

Le principe de fonctionnement de cet algorithme est présenté ci-dessous.

Le propre de NSGA-II, réside dans l'étape de classement des individus. Ce dernier est effectué en deux temps : La première étape du classement repose sur le concept de dominance de Pareto. Dans la population courante R_g , tous les individus non dominés formant un front de Pareto prennent le rang 1. Ensuite, ces individus sont retirés et on répète la même procédure pour déterminer les individus de rang 2 et ainsi de suite jusqu'au rang k (exemple de la Figure 3.10, $k = 5$). Le processus se termine lorsque tous les individus de la population courante prennent un rang unique.

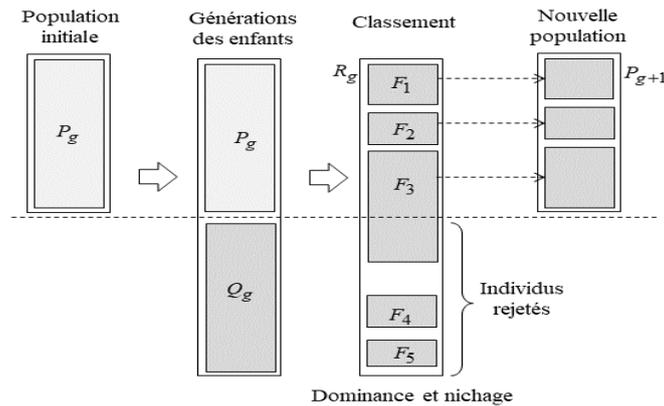


Figure 3.10 : Obtention de la nouvelle population par croisement et mutation, puis classement par dominance et nichage (Deb et al., 2002).

La deuxième étape du classement s'appuie sur une technique de nichage. La nouvelle population P_{g+1} est formée par les individus des premiers rangs (cf. Figure 3.10). Cependant, la taille de cette population doit être égale à celle de la population initiale P_g , et il est possible que les individus du dernier rang n'aient pas une place dans la formulation de la nouvelle population (le rang F_3 dans cet exemple). Dans ce cas, une technique de nichage est utilisée afin de sélectionner les individus de même rang qui peuvent participer à la création de la nouvelle population. Cette technique permet de sélectionner des individus qui ne sont pas trop proches les uns des autres après avoir déterminé leurs distances d'encombrement (crowding) afin que les solutions optimales de front de Pareto soient suffisamment diversifiées et bien réparties. Le pseudo-code qui décrit le déroulement de l'algorithme NSGA-II est présenté en Annexe.

3.6.3 Traitement des contraintes

En raison de la présence des contraintes, l'espace de recherche est divisé en régions réalisables et non réalisables. De nombreuses méthodes de traitement des contraintes ont été proposées pour résoudre le problème d'optimisation multi-objectif sous contraintes (Coello, 2002; Michalewicz and Schoenauer, 1996). Dans cette thèse, deux méthodes différentes de gestion de contraintes, à savoir la méthode de pénalité et la méthode de dominance de contrainte, ont été utilisées.

a. Méthode de pénalité

L'idée de méthode de pénalité a été introduite pour la première fois dans (Aguirre et al., 2004). Cette méthode consiste à remplacer les problèmes d'optimisation sous contraintes par

un problème d'optimisation sans contraintes, en introduisant de nouvelles fonctions objectifs à optimiser :

$$\phi_k(x) = f_k(x) + r\varphi_q(x) \quad (3.11)$$

où la fonction de pénalité choisie ici est :

$$\varphi_q(x) = \sum_{j=1}^l \max\{0, g_j(x)\}^q + \sum_{k=1}^p |h_k(x)|^q \quad (3.12)$$

où q est l'exposant de la pénalité et r est un paramètre positif de pénalité. Le problème est alors résolu directement pour une valeur de r suffisamment grande pour que les contraintes soient satisfaites.

b. La méthode de domination sous contraintes

Cette méthode est proposée par Deb (Deb et al., 2002), pour la résolution des problèmes d'optimisation multi-objectif sous contraintes, elle est basée sur le concept de domination sous contraintes, et également connue sous le nom de supériorité de la solution réalisable. En présence de contraintes, chaque solution peut être réalisable ou non réalisable. Une solution x_a est dite domine sous contrainte une solution x_b si l'une des conditions suivantes est vraie:

1. x_a est réalisable et x_b est non réalisables.
2. x_a et x_b sont irréalisables et x_a a une valeur de violation de contrainte plus petite.
3. x_a et x_b sont réalisables et x_a domine x_b avec le principe de domination usuel.

3.7 L'optimisation en conception multidisciplinaire

L'optimisation en conception multidisciplinaire (OMD ou MDO : Multidisciplinary Design Optimisation en anglais) est un domaine d'ingénierie relativement nouveau, elle a été lancée à la fin des années 1980 dans le domaine de l'aéronautique. Elle avait pour objectif de réduire le coût et le délai d'une nouvelle étude de conception dans la phase de conception préliminaire (conceptuelle). Cette méthodologie a été adaptée à la conception des systèmes complexes, en impliquant deux ou plusieurs disciplines couplées. Son utilisation a enregistré un grand succès dans différents secteurs comme l'automobile, la marine, etc., mais son succès majeur a été enregistré dans les applications aérospatiales (Anselmi et al., 2019).

Un système est dit couplé, lorsque plusieurs disciplines (plus de deux) partagent les mêmes variables de conception entre elles, ou lorsque la sortie d'une discipline correspond à l'entrée d'une autre discipline.

Pour la conception de ce type du système, les approches d’optimisation traditionnelles ne sont pas efficaces, car elles cherchent à répondre aux exigences de cahier des charges plutôt qu’à trouver des solutions optimales du problème d’optimisation, cela se fait par une série des simulations disciplinaires et plusieurs optimisations individuelles. En revanche, la méthodologie d’OMD est bien adaptée, toutes les disciplines ayant le plus grand impact sur la conception sont identifiées et considérées simultanément. Pour chaque discipline, on détermine un ensemble de variables de conception représentatives et on identifie des variables de couplage afin de bien gérer les interactions entre les sous-systèmes (disciplines) (Alexandrov, 2000).

3.7.1 Formulation mathématique d’un problème OMD

Dans un problème OMD, une discipline i peut être modélisée par une fonction $c_i(\cdot)$, des variables de conception et des variables de couplage. Les deux types de variables sont considérées comme des entrées et en calculant à la sortie des variables de couplage disciplinaire. Une discipline i est présentée en Figure 3.11 :

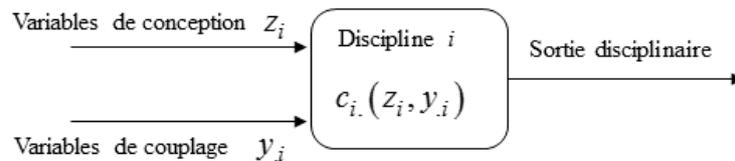


Figure 3.11 : Modélisation d’une discipline

Un problème ODM peut être formulé mathématiquement comme suit (Balesdent et al., 2012):

$$\left\{ \begin{array}{l} \min_{z,y,x} f(z,y,x) \\ \text{s.c.} \\ \mathbf{g}(z,y,x) \leq 0 \\ \mathbf{h}(z,y,x) = 0 \\ \forall (i,j) \in \{1,\dots,n\}^2, i \neq j, y_{ij} = c_{ij}(z_i, y_i, x_i) \\ \forall i \in \{1,\dots,n\}, r_i(z_i, y_i, x_i) = 0 \\ z_{\min} \leq z \leq z_{\max} \end{array} \right. \quad (3.13)$$

Toutes les variables et les fonctions sont expliquées ci-après. Dans un problème OMD, on distingue trois types de variables qui sont :

- \mathbf{z} est le vecteur de variables de conception. Ces variables peuvent être partagées entre plusieurs disciplines (nommées \mathbf{z}_{sh}) ou spécifiques à une discipline i (nommées $\bar{\mathbf{z}}_i$).

Donc, $\mathbf{z}_i = \{\bar{\mathbf{z}}_{sh}, \bar{\mathbf{z}}_i\}$ est le vecteur de variables de conception d'entrée de la discipline $i \in \{1, \dots, n\}$ avec n est le nombre de disciplines.

- \mathbf{x} est le vecteur de variables d'état. Contrairement à $\mathbf{z} = \{\mathbf{z}_{sh}, \bar{\mathbf{z}}\}$, les variables d'état dépendent de variables de conception \mathbf{z} , de variables de couplage \mathbf{y} et d'équations d'état caractérisées par les résidus \mathbf{r} . Ces variables sont souvent définies par des relations implicites nécessitant à leur résolution des méthodes numériques spécifiques.
- Dans un environnement multidisciplinaire, les disciplines s'échangent des variables de couplage \mathbf{y} (Figure 3.12). Ces dernières relient les différentes disciplines pour modéliser leurs interactions. $\mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{ji}, \mathbf{x}_i)$ est une fonction de couplage utilisée pour calculer le vecteur de variables de couplage de sortie de la discipline i et d'entrée de la discipline j . \mathbf{y}_{ji} réfère au vecteur de toutes les variables de couplage d'entrée de la discipline i et \mathbf{y}_{ij} est le vecteur de variables de couplage d'entrée de la discipline j et de sortie de la discipline i .

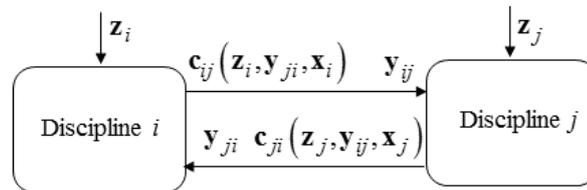


Figure 3.12 : Couplage entre la discipline i et la discipline j

La Figure 3.13 illustre un exemple d'analyse classique de l'aéro-structure d'un lanceur (Filomeno Coelho et al., 2009; Kennedy and Martins, 2014). Dans cet exemple, l'analyse aéro-structure implique une analyse couplée entre la discipline aérodynamique (la géométrie du lanceur et la déformation sont nécessaires) et la discipline structure (efforts aérodynamiques sur la structure du lanceur sont nécessaires). Il est clair que dans la conception des systèmes couplés, les objectifs sont souvent antagonistes, par exemple, la réduction du poids peut entraîner des contraintes structurelles élevées.

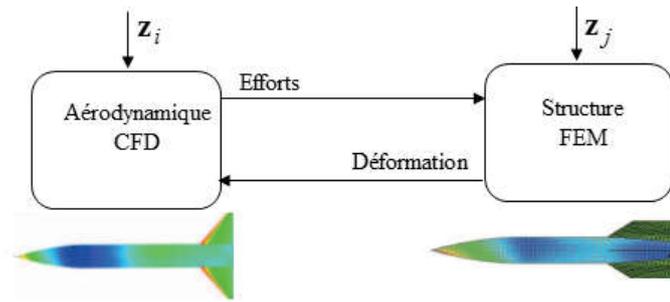


Figure 3.13 : Couplage entre deux disciplines : Structure et Aérodynamique

- Contraintes d'égalité et d'inégalité : la solution d'un problème OMD doit satisfaire les contraintes d'inégalités $\mathbf{g}(\cdot)$ et les contraintes d'égalités $\mathbf{h}(\cdot)$. Ces contraintes traduisent les exigences d'un système en termes de performances cibles.
- Réalisation disciplinaire (Individual Disciplinary Feasible) : La solution de problème OMD doit assurer la satisfaction disciplinaire à travers les contraintes d'égalité sur le résidu $\mathbf{r}_i(\cdot)$. le résidu quantifie la satisfaction de l'équation d'état dans la discipline i . Le vecteur de variable d'état \mathbf{x} est la solution des équations d'état de la discipline i .
- Réalisation multidisciplinaire (Multidisciplinary feasibility) : La solution de problème OMD doit satisfaire la contrainte d'égalité interdisciplinaire entre le vecteur des variables de couplage d'entrée \mathbf{y} et le vecteur des variables de couplage de sortie $\mathbf{c}_i(\cdot)$. Le couplage entre les disciplines i et j est cohérent lorsque le système d'équations interdisciplinaire suivant est vérifié :

$$\begin{cases} \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_i) \\ \mathbf{y}_{ji} = \mathbf{c}_{ji}(\mathbf{z}_j, \mathbf{y}_j) \end{cases} \quad (3.14)$$

D'après le système (Equation (3.14)), lorsque tous les couplages sont cohérents (satisfaits), le système est dit *Multidisciplinary feasible*. La satisfaction des couplages interdisciplinaires est essentielle car c'est une condition nécessaire pourvu que le système étudié soit physiquement réalisable, sera détaillée dans la section 3.7.2.

- Solution optimale de problème OMD : f est la fonction objectif (performance) à optimiser. Le problème OMD peut contenir plusieurs fonctions objectifs (performances).

3.7.2 Satisfaction de couplage multidisciplinaire

Dans l'optimisation multidisciplinaire, on peut distinguer deux catégories de méthodes afin de satisfaire les couplages interdisciplinaires (Balling and Sobieszczanski-Sobieski, 1996): les approches couplées et les approches découplées.

3.7.2.1 Approches couplées (Coupled approaches)

Les approches couplées (Figure 3.14) effectuent une analyse multidisciplinaire (MDA) afin d'assurer les couplages interdisciplinaires à chaque itération de l'optimisation au niveau système global. La MDA est une analyse auxiliaire qui cherche à trouver un équilibre entre les disciplines en résolvant le système d'équations interdisciplinaires (équations d'état) (Filomeno Coelho et al., 2009). Autrement dit, MDA consiste à trouver la valeur des variables de couplage d'entrée y qui satisfont le système d'équations interdisciplinaires. Un processus itératif est nécessaire pour résoudre le système d'équations en raison du couplage des disciplines. On peut distinguer deux méthodes classiques de MDA : 1) Méthode du point fixe (Fixed Point Iteration), 2) Méthode basée sur la minimisation des résidus d'équations interdisciplinaires (Breitkopf and Coelho, 2013; Filomeno Coelho et al., 2009).

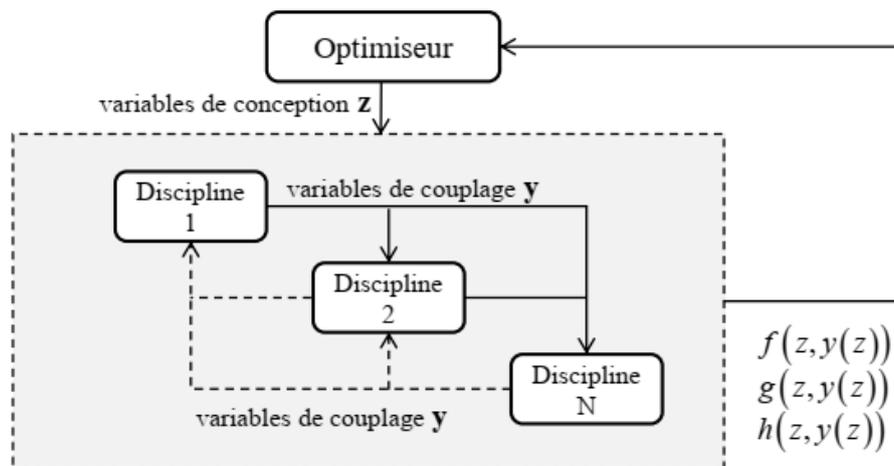


Figure 3.14 : Approche couplée

- Méthode de point fixe :** est une procédure itérative impliquant des boucles entre les disciplines sans aucun contrôle sur les variables de couplage (exception de l'initialisation), ces dernières sont des résultats des simulations disciplinaires. Cette méthode peut considérer comme une extension de l'algorithme de Gauss-Seidel (la résolution d'équations algébriques). Dans cette méthode, un seul vecteur de couplage est initialisé (y_j dans la Figure 3.15). Généralement, elle n'assure pas toujours la propriété de convergence, il y a des conditions théoriques (Ortega, 1973) montrent que