

Algorithmes d'intelligence en essaim

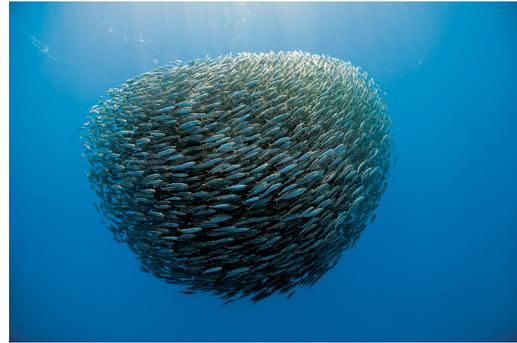
Les algorithmes d'intelligence en essaim sont issus de l'étude du comportement collectif de certaines espèces comme les oiseaux, les poissons, les fourmis ou encore les abeilles. Par exemple, les fourmis laissent des traces de phéromones pour indiquer à leurs congénères le chemin vers les points d'eau et de nourriture, les abeilles s'agglutinent entre elles durant l'hiver pour se protéger du froid et se répartissent les tâches pour organiser la vie de la ruche, les oiseaux adoptent une formation en V pour minimiser les pertes aérodynamiques lors des longues migrations et les poissons adoptent une formation en banc serré notamment pour se protéger des prédateurs. La figure 3.16 présente quelques exemples d'intelligence en essaim pour différentes espèces animales.

Les algorithmes d'intelligence en essaim se caractérisent par l'utilisation d'une population d'agents. Ici, la notion d'agent se différencie de celle de solution, puisqu'ici chaque individu de la population a la possibilité de communiquer avec les autres membres de la population, et dispose de son propre système de décision.

L'intelligence de l'essaim est alors générée par des règles simples qui régissent les interactions entre les agents et leur environnement et aboutit à l'émergence d'un compor-



(a) Formation en V d'un groupe d'oiseaux.



(b) Banc de poissons.



(c) Pont flottant de fourmis.

FIGURE 3.16 – Exemple d'intelligence en essaim dans le règne animal.

tement pour l'essaim entier [85]. Un tel système est *auto organisé*, puisque le comportement de l'essaim n'est pas déterministe et dépendra des règles décisionnelles imposées aux agents.

Les algorithmes les plus populaires de cette famille de métaheuristique sont les méthodes d'optimisation par essaim particulaire (OEP), les colonies de fourmis et les colonies d'abeilles artificielles [96].

Parmi les méthodes citées précédemment, nous avons fait le choix de mettre en œuvre l'optimisation par essaim particulaire ou *particle swarm optimization*. Cette technique est régulièrement utilisée dans la littérature scientifique pour solutionner des problèmes d'optimisation dans une vaste gamme de domaines d'application.

3.4.6.1 Optimisation par essaims particulaires

Initialement, les travaux de Reynolds [97] et Heppner [98] ont permis de mettre en évidence que les animaux au sein d'un groupe en mouvement suivent le mouvement global du groupe grâce aux déplacements de leurs voisins tout en conservant un espacement optimal entre les individus.

Dans la nature, il est possible de remarquer que la dynamique de déplacement d'un groupe d'animaux peut être très complexe alors que chaque individu pris individuellement n'a qu'une connaissance limitée de sa position dans l'essaim (figure 3.16).

Par la suite, en 1995, Kennedy et Eberhart ont repris ces observations pour en déduire une métaheuristique de recherche : l'optimisation par essais particuliers (OEP) [99]. Ainsi, l'OEP exploite ce concept de déplacement coordonné en intégrant une dimension sociale et une mémoire de groupe aux particules composant l'essaim afin de mener une recherche efficace. La stratégie globale de l'essaim s'adapte alors selon les expériences vécues par chaque particule de l'essaim pour atteindre l'optimum global de l'espace des solutions.

La popularité de cet algorithme s'explique, entre autres choses, par sa relative simplicité d'implémentation, le faible nombre de paramètres de réglages pour paramétrer le déroulement de la recherche, sa capacité à explorer efficacement un espace à n -dimensions, ou encore l'utilisation de règles de recherche simple qui n'utilisent pas de gradient.

Dans la pratique, les particules de l'essaim sont considérées comme des solutions possibles au problème. Une particule est définie par une position X_i^n sur l'espace des solutions et une vitesse de déplacement V_i^n ³, elle possède également une mémoire qui stocke la meilleure position visitée $P_{i,best}$ et la valeur de la fonction objectif en ce point. Les particules ont également la capacité de se communiquer entre elles la position de la meilleure solution globale connue de l'essaim G_{best} .

3.4.6.2 Règles de déplacement

La figure 3.17 représente le déplacement d'une particule dans un espace à 2 dimensions. Le déplacement est conditionné par trois composantes :

- Une composante inertielle qui entraîne la particule dans la direction de recherche courante héritée de l'itération précédente (traits bleus).
- Une composante cognitive qui pousse la particule à se diriger vers la meilleure solution qu'elle a visitée $P_{i,best}$ lors des itérations précédentes (traits rouges).
- Une composante sociale qui amène la particule à se diriger vers la meilleure solution G_{best} visitée par les autres particules de son voisinage (traits verts).

L'équation du mouvement est donnée par le système (4.30). ωV_i^n représente l'inertie de la particule, $r_1 \beta_1 (P_{i,best} - X_i^n)$ est la mémoire cognitive de la particule, tandis que le dernier terme $r_2 \beta_2 (G_{best} - X_i^n)$ exprime la mémoire sociale de l'essaim. Le coefficient d'inertie ω sert à contrôler l'importance de la direction de recherche courante sur le déplacement futur et pour régler la capacité d'exploration de l'essaim tandis que β_1 et β_2 , respectivement le coefficient cognitif et le coefficient social, permettent de régler la capacité d'exploitation. r_1 et r_2 sont quant à eux des réels tirés uniformément dans l'intervalle $[0, 1]$. L'ensemble de ces coefficients a donc pour objectif d'orienter la recherche de l'optimum.

$$\begin{cases} V_i^{n+1} = \omega V_i^n + r_1 \beta_1 (P_{i,best} - X_i^n) + r_2 \beta_2 (G_{best} - X_i^n) & (3.14a) \\ X_i^{n+1} = X_i^n + V_i^{n+1} & (3.14b) \end{cases}$$

3. Le terme vitesse est un abus de langage puisque ce vecteur V n'est pas homogène à une vitesse, il s'agit plutôt d'une direction et d'une amplitude de déplacement.

L'utilisation du terme vitesse permet de conserver l'analogie avec le monde animal.

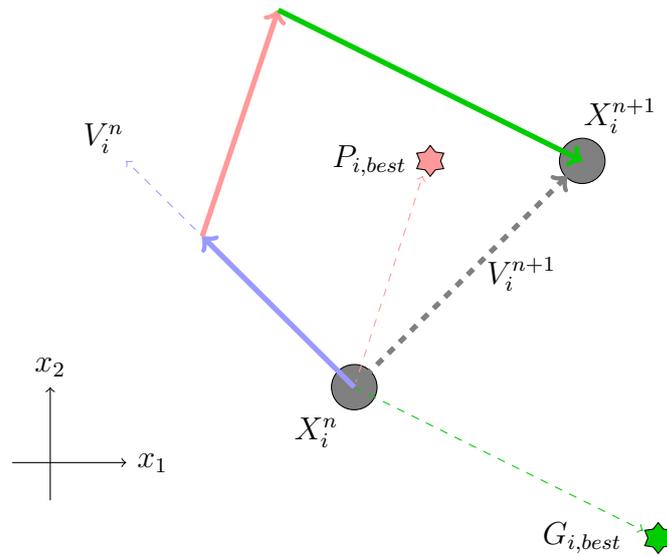


FIGURE 3.17 – Représentation du déplacement d’une particule dans un espace à 2 dimensions.

3.4.6.3 Codage des solutions

Chaque solution de l’essaim est codée de manière analogue à un chromosome, par une liste des temps d’arrêt qui seront effectués successivement par les trains en exploitation. Une particule est alors un vecteur d’entiers appartenant à un espace n –dimensions et bornés sur chaque dimension du problème. Ces bornes permettent, lors du processus itératif, de ne pas explorer les solutions qui violent les conditions d’exploitations définies en 3.2.3. L’objectif étant que chacune des positions visitées représentent un scénario de stationnement plausible.

3.4.6.4 Notion de voisinage

Au sein de l’essaim, les particules communiquent entre elles via un réseau social appelé voisinage. Il ne s’agit pas ici de voisinage géométrique mais plutôt d’un voisinage topologique. Un voisinage spatial nécessiterait de recalculer à chaque itération le nouveau voisinage géométrique des particules ce qui alourdirait la procédure de recherche. D’après [87], le choix de la topologie influence grandement les propriétés de convergence de l’OEP.

Principalement trois types de topologie de voisinage existent dans la littérature : le voisinage en rayon où toutes les particules ne communiquent qu’avec une particule centrale ; le voisinage en anneau où chaque particule n’est relié qu’avec un nombre limité d’autres particules, les particules tendent alors à se déplacer vers la meilleure particule de son voisinage ; le voisinage en étoile où toutes les particules sont capables de communiquer entre elles [86].

Ici, le choix a été fait d’une communication totale entre tous les individus de l’essaim. Ce choix est notamment motivé par le fait que dans notre cas, l’espace des solutions a une dimension assez élevée $dim_{E_{sol}} \in [10^1; 10^3]$, de sorte qu’il y a peu de probabilités que les particules stagnent dans un optimum local. Cette version de l’algorithme d’OEP où tous les individus de l’essaim communiquent entre eux est appelée *version globale*, puisque toutes les particules ont connaissance d’un même optimal

global.

3.4.6.5 Limitation de la vitesse de déplacement

Lors du processus itératif, il peut arriver que la vitesse d'une particule devienne excessive ce qui pourrait l'amener à passer à côté d'un optimum et à sortir de l'espace des solutions admissibles.

Une solution évoquée par [87], considère de confiner l'espace de recherche en s'assurant soit que l'évaluation de la fonction objectif n'est effectuée que pour les particules se trouvant à l'intérieur de l'espace de recherche, soit que les particules restent dans l'espace de recherche en les arrêtant à la frontière, soit en utilisant un opérateur de rebond pour qu'une fois arrivées à la frontière de l'espace, les particules reviennent dans l'espace des solutions admissibles.

Dans le système (4.30), les termes ω , β_1 et β_2 permettent d'effectuer un compromis entre exploration et exploitation et d'après [100] et [101], l'établissement d'une relation de dépendance entre ces paramètres permettrait d'améliorer les propriétés de convergence de la méthode.

Les auteurs ont donc mis en place un *coefficient de constriction* Ψ . Ce coefficient présente l'avantage de se passer de la définition d'une vitesse maximale de déplacement tout en contrôlant l'amplitude du déplacement des particules. L'équation de mise à jour de la vitesse devient alors (3.15).

Cependant, nos divers essais pour mettre en oeuvre cette variante ne se sont pas avérés probants puisque le taux de convergence était dégradé en utilisant cette modification.

L'une des raisons les plus probables, de cette discordance par rapport aux travaux [100] et [101], serait un trop grand nombre de dimensions dans le problème d'optimisation considéré.

$$V_i^{n+1} = \Psi V_i^n + r_1 \beta_1 (P_{i,best} - X_i^n) + r_2 \beta_2 (G_{i,best} - X_i^n) \quad (3.15)$$

Pour ces raisons, un dimensionnement approprié du coefficient d'inertie ainsi que des paramètres cognitif et social est privilégié pour limiter et orienter le déplacement des particules.

Ainsi, la valeur du coefficient d'inertie détermine si les particules suivent une politique d'exploration globale ($\omega > 1$) ou au contraire une exploration locale ($\omega < 1$) en donnant plus ou moins d'amplitude au déplacement des particules.

Deux approches ont été testées : une approche utilisant un coefficient d'inertie fixe et une autre utilisant une règle de décroissance linéaire de ce coefficient (3.16).

Cette dernière approche a notamment été présentée dans les travaux de [87] et [102]. Dans (3.16), ω_{min} et ω_{max} représentent les bornes de variation du coefficient d'inertie, $n_{iteration}$ est l'indice de l'itération courante et $max_{iteration}$ est le nombre maximal d'itérations prévues pour la résolution du problème.

$$\omega = \omega_{min} + \omega_{max} \frac{n_{iteration}}{max_{iteration}} \quad (3.16)$$

L'utilisation d'un coefficient d'inertie fixe nécessite d'effectuer un grand nombre d'essais pour affiner la valeur de ce paramètre, tandis que l'emploi d'un coefficient

variable requiert de définir un nombre maximal d'itérations, ce qui peut s'avérer délicat en cas de convergence lente.

Néanmoins, en fixant le nombre d'itérations à une valeur élevée et en ajoutant un autre critère d'arrêt pour stopper l'exploration de l'espace des solutions, le coefficient d'inertie suit une décroissance progressive qui permet au fur et à mesure des itérations de passer d'une exploration globale à une exploration locale⁴.

Dans notre étude, le critère d'arrêt supplémentaire vise à stopper le processus d'optimisation lorsque la meilleure solution connue par l'essaim n'évolue plus sur un certains nombre d'itérations, afin de ne pas prolonger le processus de recherche plus qu'il n'est nécessaire.

3.4.6.6 Implémentation de l'algorithme

L'algorithme 7 résume les étapes d'un algorithme classique d'optimisation par essais particuliers, où f représente la fonction coût de l'algorithme.

Algorithme 7 Algorithme d'optimisation par essais particuliers

- 1: Initialiser une population de N_{pop} individus, définis par des vitesses aléatoires et des positions prises dans l'espace des solutions.
 - 2: Evaluer les valeurs de fonction objectif aux positions des N_{pop} individus.
 - 3: Définir les optima initiaux $P_{i,best}$ et G_{best} .
 - 4: **Tant que** Le critère d'arrêt n'est pas vérifié **Faire**
 - 5: **Pour** $i = 1$ à N_{pop} **Faire**
 - 6: Mettre à jour la vitesse de la particule suivant l'équation (3.14a)
 - 7: Mettre à jour la position de la particule suivant l'équation (3.14b)
 - 8: Evaluer la valeur de la fonction objectif à la position de la particule $f(X_i(n))$
 - 9: **Si** $f(X_i(n)) < f(P_{i,best})$ **Alors**
 - 10: $P_{i,best} \leftarrow X_i(n)$
 - 11: **Fin du Si**
 - 12: **Si** $f(X_i(n)) < f(G_{best})$ **Alors**
 - 13: $G_{best} \leftarrow X_i(n)$
 - 14: **Fin du Si**
 - 15: **Fin du Pour**
 - 16: **Fin du Tant que**
-

L'algorithme 7 est volontairement simpliste car il existe dans la littérature un très grand nombre de variantes qui influent sur un ou plusieurs paramètres et démontrent une amélioration des propriétés convergentes de leurs algorithmes.

De fait, le choix a été fait de développer une méthode généraliste qui a montré des propriétés de convergence satisfaisante dans un grand nombre de classes de problèmes, à condition que les paramètres de l'algorithme soient correctement dimensionnés [86].

4. L'exploration locale s'apparente alors dans ce cas de figure à une exploitation de la recherche effectuée durant les premières itérations.

3.4.7 Hybridation des méthodes d'optimisation

3.4.7.1 L'hybridation dans la littérature

La méthode d'optimisation par essaim particulaire présente l'avantage de posséder une mémoire collective partagée par l'ensemble des particules de l'essaim, cependant, cette mémoire collective peut entraîner des problèmes de convergence prématurée vers un optimum local lorsque le paramétrage du déplacement des particules ne leur permet pas d'explorer suffisamment l'espace des solutions [87].

A l'inverse, dans la méthode d'optimisation par algorithme génétique, les individus n'ont pas de mémoire des meilleures positions visitées, mais les opérateurs évolutionnaires permettent une exploration assez approfondie de l'espace des solutions.

Ainsi, si les probabilités d'occurrence de ces opérateurs ne sont pas maîtrisées, il en résulte une exploration aléatoire de l'espace de recherche sans garantie de convergence vers l'optimum global du problème.

On peut alors estimer grossièrement que l'AG est une méthode qui explore l'espace des solutions, tandis que l'OEP exploite les résultats de la recherche. Intuitivement, une hybridation entre ces deux méthodes permettrait de trouver une alternative au compromis exploration/exploitation. Dans ce cas de figure, l'hybridation permet de capitaliser sur les points forts des méthodes pour accélérer la convergence vers l'optimum global de l'espace de recherche.

L'hybridation de techniques d'optimisation consiste à répartir différentes tâches de recherche entre plusieurs méthodes [103].

L'hybridation de méthodes d'optimisation est régulièrement pratiquée dans la littérature, que ce soit avec des métaheuristiques ou des méthodes exactes. Dans cette section, un état de l'art des hybridations entre un algorithme génétique et un algorithme d'essaim particulaire a été effectué. L'idée sous-jacente de cette étude est qu'en hybridant deux méthodes qui ont fait leur preuve sur des problèmes d'optimisation combinatoire NP-difficiles, une métaheuristique de recherche encore plus efficace sera obtenue.

[104] propose d'hybrider un algorithme d'optimisation par essaim particulaire en lui intégrant des règles évolutionnaires issues d'un algorithme génétique (AG) selon trois variantes :

Hybridation parallèle : l'OEP et l'AG effectuent leurs itérations en parallèle. Lorsque la position de G_{best} n'évolue pas sur plusieurs itérations, une opération de croisement est réalisée entre la particule G_{best} et des individus issus de l'AG.

Hybridation série : l'optimisation par AG sert de point de départ à l'OEP. Une première optimisation est réalisée par algorithme génétique sur un nombre donné d'itérations puis, une optimisation par OEP est effectuée sur un même nombre d'itérations. La population finale issue de l'AG est utilisée comme population initiale de l'OEP.

Hybridation d'insertion : seul un algorithme d'OEP est employé pour réaliser l'optimisation. Lorsque la valeur de $P_{i,best}$ de la particule i n'évolue pas sur un certain nombre d'itérations, un opérateur de mutation est appliqué à la position de $P_{i,best}$.

Dans [104], l'auteur justifie le choix de l'utilisation de règles évolutionnaires dans l'OEP par le fait que le contrôle de la convergence et du déplacement des particules est

beaucoup plus important avec le coefficient d'inertie qu'avec les opérateurs de croisement et de mutation. En outre, chacune de ces variantes a été testé sur un *benchmark* de fonctions continues. Cependant, aucune variante n'a prouvé sa supériorité par rapport aux autres sur chaque problème du *benchmark* (conformément au théorème du *No free lunch*).

Dans [105], l'auteur étudie plusieurs techniques d'hybridation entre différentes métaheuristiques à base de population. L'hybridation entre AG et OEP est réalisée par insertion en remplaçant l'étape de mutation des individus par les mécanismes de mise à jour de la PSO. Cette hybridation a pour objectif de réduire le temps de calcul de l'AG tout en améliorant la convergence de la méthode.

Une approche légèrement différente est explicitée dans [106], où un opérateur de mutation est ajouté à un OEP après l'étape de mise à jour de la position des particules pour augmenter l'exploration de l'espace de recherche.

Dans [107], une hybridation entre OEP et AG est également décrite. A chaque itération, les individus issus de l'optimisation par AG et OEP sont triés par valeur de fitness. Les meilleurs individus de la population sont utilisés par l'AG tandis que le déplacement des moins bons individus est régi par l'OEP. La routine de l'algorithme génétique est exécutée en premier de sorte que l'OEP puisse utiliser la meilleure position trouvée par l'AG pour la mise à jour des positions.

Il est enfin à noter que [87] propose de partitionner l'espace de recherche en assignant à chaque zone de l'espace une population dédiée. Bien que cette idée soit en théorie très judicieuse, en pratique, cela s'avère compliqué à mettre en œuvre quand le nombre de dimensions du problème augmente.

3.4.7.2 Choix d'hybridation retenu

A la lumière des travaux précédents, il a été décidé d'effectuer une hybridation en combinant les caractéristiques de l'hybridation parallèle et de l'hybridation série.

Le principe d'hybridation retenu est qu'à chaque itération, la population issue de l'optimisation par OEP a connaissance de la meilleure position visitée par la population de l'AG, et inversement, l'AG a la capacité d'utiliser les solutions déterminées par OEP pour former de nouveaux individus. De cette manière, chaque population bénéficie des propriétés de recherche et de convergence de l'autre méthode.

Cette hybridation a été privilégiée pour plusieurs raisons. D'une part, l'utilisation de deux populations générées par deux méthodes d'optimisation différentes permet de conserver les propriétés de convergence de chacune des méthodes à chaque itération, d'autre part, le partage des informations entre OEP et AG augmente la vitesse de convergence globale et la diversité des solutions explorées.

Enfin, cette hybridation permet à l'OEP d'exploiter les meilleures solutions explorées par l'AG et ainsi de réaliser un compromis entre exploration et exploitation.

L'algorithme d'optimisation hybride OEP-AG est récapitulé par le diagramme 3.18. Il est à noter qu'à la première itération de l'algorithme, la population AG est la même que la population OEP.

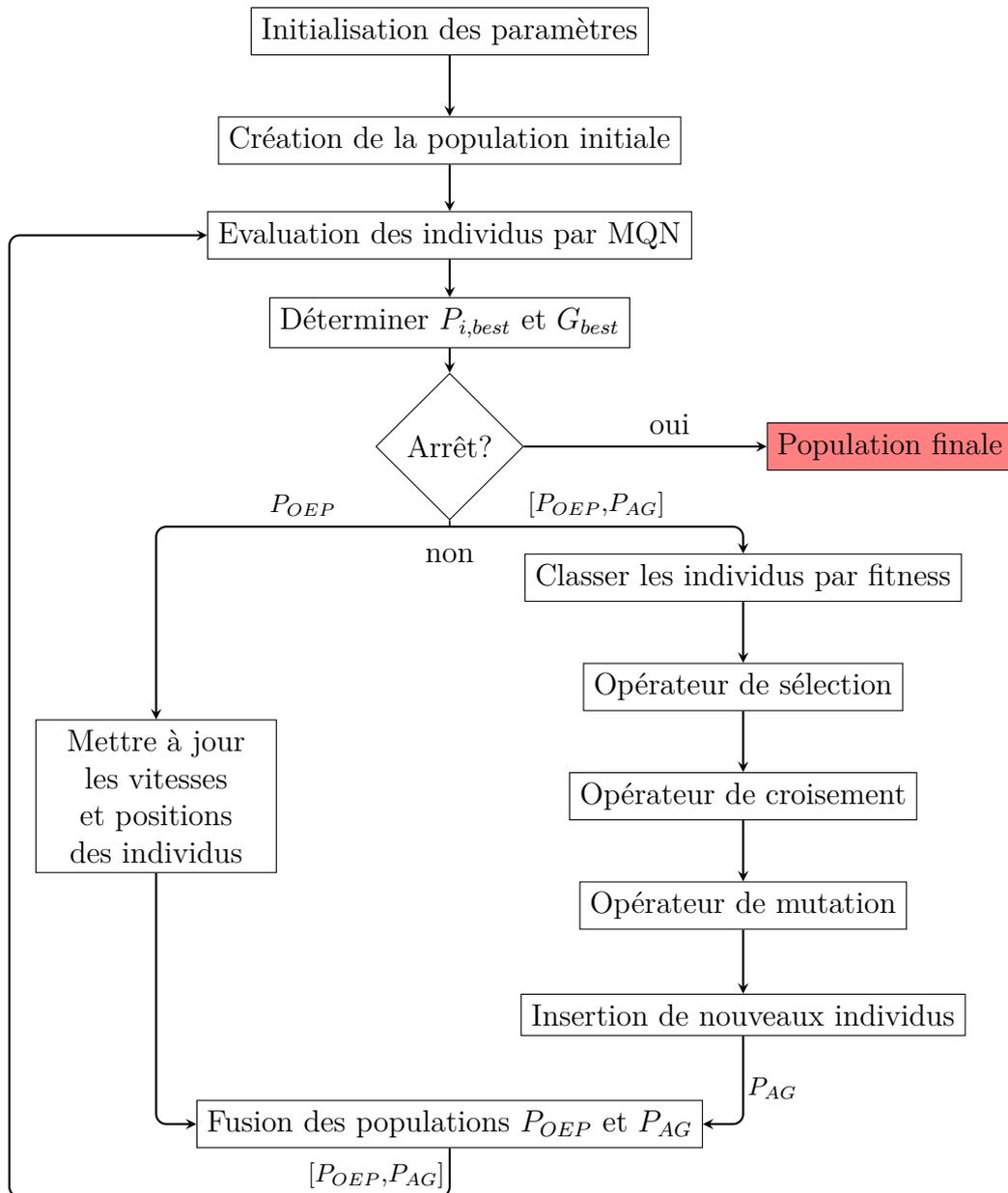


FIGURE 3.18 – Diagramme de l’algorithme hybride OEP-AG.

3.4.8 Comparaison des méthodes d’optimisation

Chacune des métaheuristiques présentées précédemment : OEP, AG et l’hybride OEP-AG a été implémentée en vue de solutionner le problème d’optimisation des temps d’arrêt en station. Ces méthodes ont ensuite été comparées pour déterminer laquelle est la plus efficace pour le type de problème rencontré.

L’indicateur de performance utilisé pour comparer ces techniques est le gain énergétique entre une table horaire utilisant des temps d’arrêt en station nominaux et le meilleur planning de stationnement trouvé par optimisation.

Lors du fonctionnement nominal d’un carrousel établi, sans aléas d’exploitation, les trains sont soumis à un intervalle et suivent une consigne de temps d’arrêt en station. Cette combinaison de temps d’arrêt en station effectuée par l’ensemble des trains du carrousel donne lieu à une consommation énergétique nominale.

Les méthodes d’optimisation permettent de déterminer de nouvelles combinaisons

de temps d'arrêt qui auront une influence sur la consommation énergétique globale de la ligne de métro.

Le gain énergétique est donc défini par rapport à la consommation du carrousel sur un tour de boucle en considérant un fonctionnement nominal.

Le tableau 3.3 présente une comparaison des performances de convergence des populations de l'OEP et de l'AG et la densité de distribution des solutions renvoyées par chaque méthode. La colonne de gauche reprend le cas où chaque population évolue indépendamment l'une de l'autre ; tandis que la colonne du milieu correspond à l'évolution des populations dans le cadre de l'hybridation.

La colonne de droite analyse la distribution des solutions trouvées par chaque méthode afin d'analyser quelle méthode est la plus appropriée pour effectuer une recherche efficace.

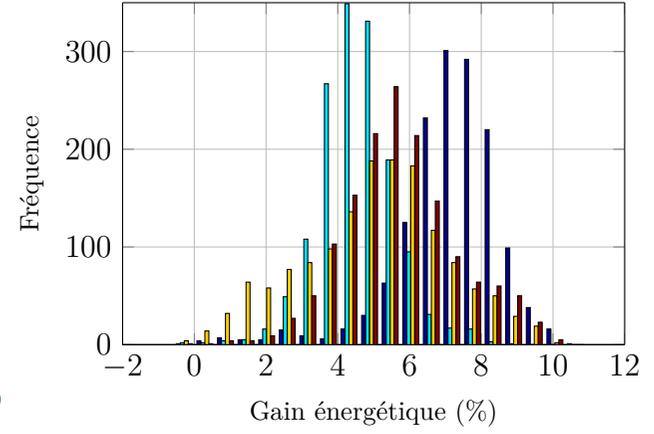
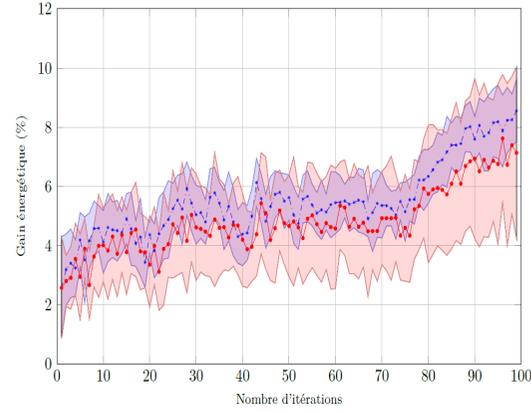
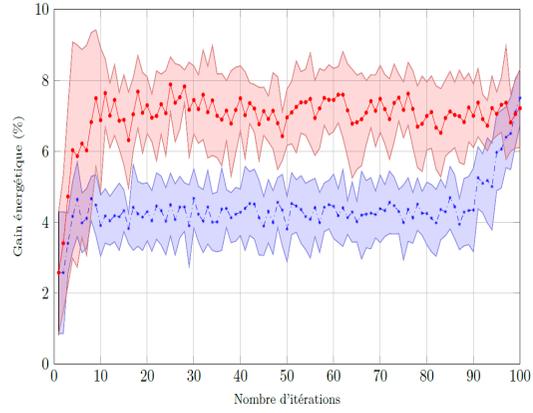
N_{pop}

AG & OEP indépendants

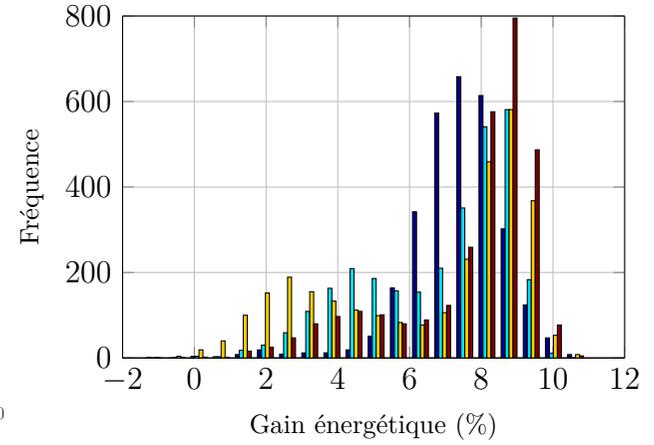
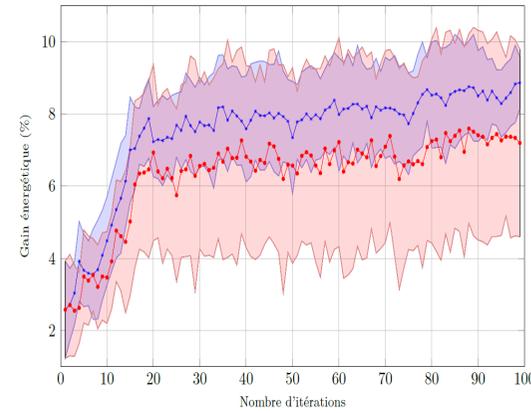
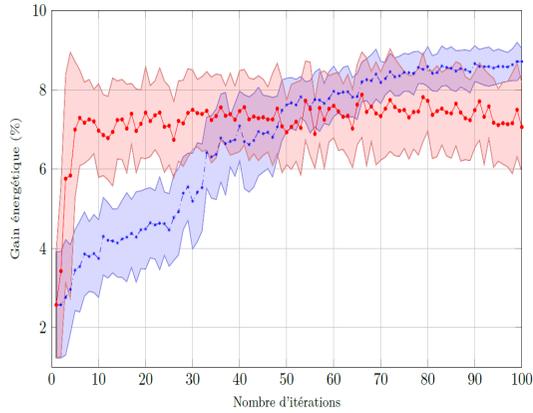
Hybride OEP-AG

Densité de distribution

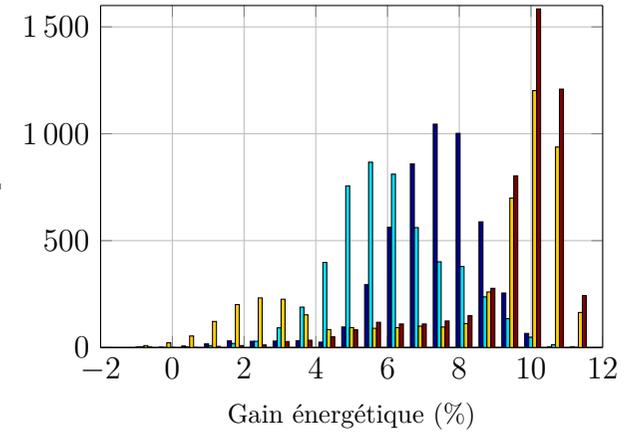
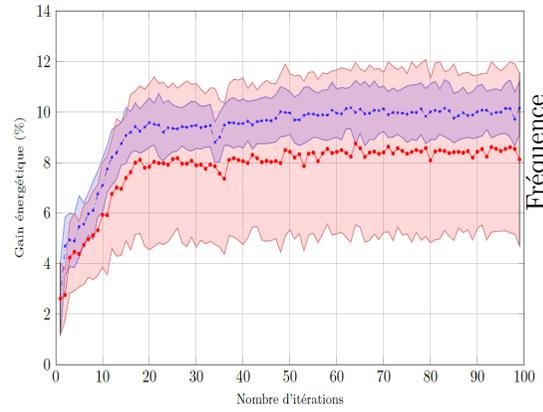
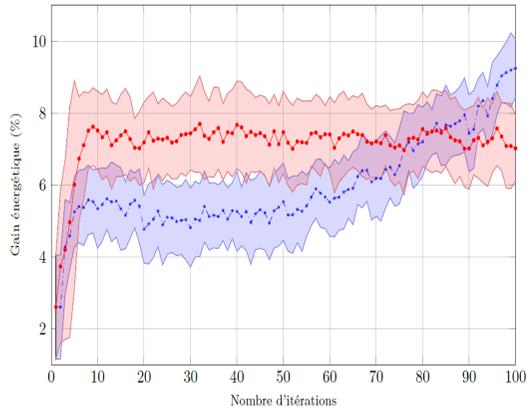
15



30



50



70

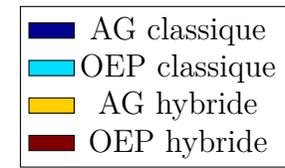
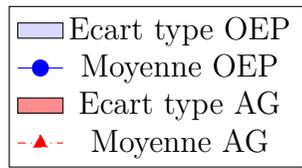
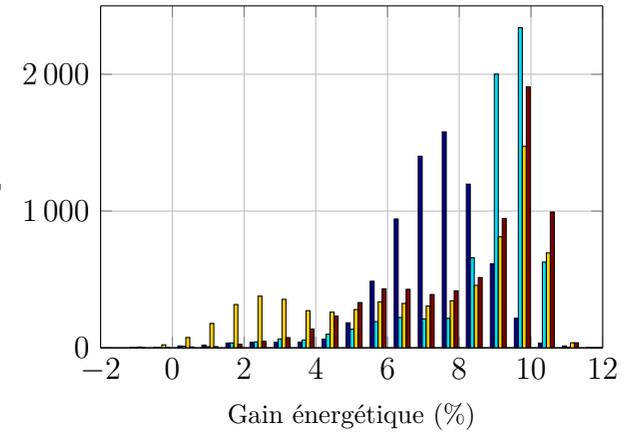
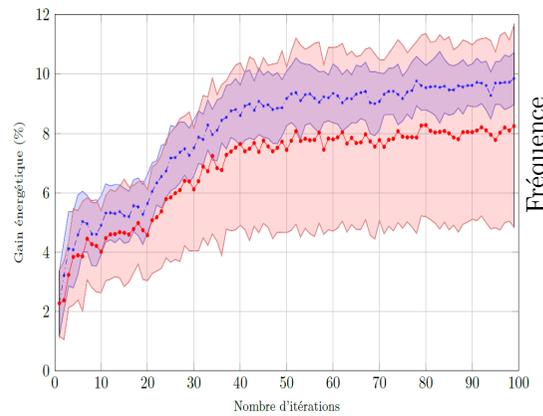
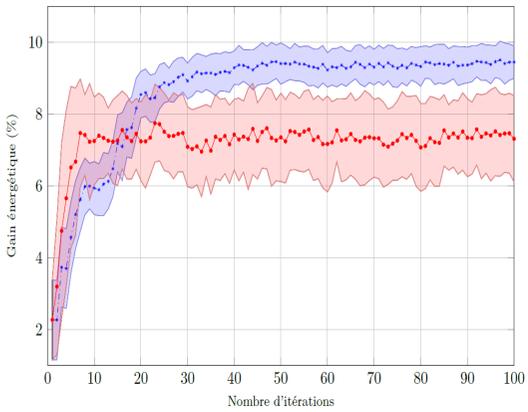


TABLEAU 3.3 – Comparaison des propriétés de convergence des méthodes d'optimisation en fonction de la taille de la population.

Le tableau 3.3 permet de relever quelques caractéristiques concernant le gain de performance obtenu en réalisant l'hybridation de la méthode AG-OEP :

- Pour un même nombre d'itérations et un même nombre d'individus dans la population, la méthode hybride permet de trouver des combinaisons de temps d'arrêt qui entraînent un plus fort gain énergétique.
- Dans sa version classique, les performances de l'AG semblent être indépendantes de la taille de la population à l'inverse de l'OEP.
- Lorsque les méthodes OEP et AG évoluent indépendamment, l'AG est plus performante que l'OEP durant les premières itérations. Cependant lorsque le nombre d'individus de la population augmente, l'OEP permet d'explorer de meilleures solutions. En outre, l'AG semble converger vers un optimum local dès les premières itérations, tandis que l'OEP voit une amélioration constante des solutions explorées.
- Lorsque les méthodes OEP et AG coopèrent en partageant leurs informations sur les solutions explorées, l'OEP domine l'AG dans tous les cas de figure simulés. En outre, l'écart type des solutions analysées par l'algorithme génétique est beaucoup plus étendu que celui des solutions explorées par essaim particulaire. L'hybridation assure un compromis entre exploration de l'espace et exploitation des résultats. L'AG joue alors le rôle de méthode exploratrice tandis que l'OEP exploite les solutions trouvées aux itérations précédentes pour converger vers un optimum.

D'après le tableau 3.3, une méthode OEP-AG hybride composée de 50 individus semble être la plus appropriée pour obtenir une distribution de solutions de bonne qualité tout en limitant le temps de calcul.

Le tableau 3.3 s'est intéressé à l'étude des performances d'amélioration de la fonction objectif, cependant, un autre aspect important à prendre en compte dans ces travaux est la performance en temps de calcul. Ainsi, le tableau 3.4 présente le temps de calcul moyen d'un individu sur une itération pour les 3 méthodes introduites précédemment, autrement dit le temps de calcul nécessaire à l'exploration d'une solution. Il est à noter que lors des différents essais, le temps de calcul de chacun des cas présentés dans le tableau 3.3 évolue linéairement en fonction du nombre d'individus composant la population.

	Méthode AG	Méthode OEP	Méthode hybride OEP-AG
Temps de calcul unitaire (s)	6.8613	6.7158	14.6305

TABLEAU 3.4 – Temps de calcul moyen d'une solution

D'après le tableau 3.4, une utilisation séquentielle des méthodes AG et OEP est plus rapide que la résolution du problème par l'hybride, avec un gain en temps de calcul d'environ 7.8%. Les temps de calcul correspondent à un PC embarquant un processeur Intel Xeon W3520 cadencé à 2.67GHz, 12 Go de RAM, une carte graphique Quadro FX 3800 avec 1Go de VRAM et le tout sous un environnement Windows 7.

Cette étude des performances des méthodes d'optimisation illustre donc le fait que l'amélioration des propriétés de convergence d'une méthode se fait par dégradation du temps de calcul/exploration.

3.4.9 Répartition des temps de calcul

D'après le tableau 3.4, le temps de calcul moyen d'une solution par la méthode hybride OEP-AG est d'environ 14.6s. Une étude plus approfondie de la répartition du temps de calcul nécessaire à l'exploration d'une solution montre qu'en moyenne 91% du temps de calcul est dédié à la résolution itérative des flux de puissance relatifs à la solution explorée.

Les 9% du temps de calcul restant concernent l'exécution des autres modules algorithmiques impliqués dans la boucle d'optimisation comme la construction du vecteur solution, la mise à jour des solutions générées par OEP, la mise à jour des solutions générées par AG,...

3.4.10 Remarques

Une démarche explorant à la fois l'intervalle d'exploitation et les temps d'arrêt en station aurait pu être adoptée, cependant, diverses expérimentations ont montré que cette approche était plus coûteuse en temps de calcul et fournissait exactement les mêmes résultats qu'une approche décorrélant chacune des étapes de l'optimisation des paramètres d'exploitation.

La taille de l'espace des solutions dans une démarche corrélant les deux paramètres d'exploitation serait donnée par (3.17a) alors qu'en utilisant une approche décorrélant ces deux paramètres, la taille de l'espace des solutions du problème global est donnée par (3.17b), où N_{inter} est le nombre d'intervalles possibles pour chaque carrousel tel que défini par (3.3).

$$N_{solglob} = \begin{cases} N_{inter}^{A_{mod}^{n_{stop}}} & (3.17a) \\ N_{inter} \cdot A_{mod}^{n_{stop}} & (3.17b) \end{cases}$$

3.5 Optimisation d'une table horaire journalière

Dans le domaine ferroviaire, une table horaire journalière définit les horaires de début et de fin d'exploitation, les horaires de départs et d'arrivées de chaque train dans les différentes stations, les horaires d'insertion et de retrait des trains, les temps d'arrêt en station et les intervalles d'exploitation. Certaines de ces informations sont redondantes, mais permettent d'avoir une vision claire des conditions d'exploitation de la ligne.

Les tables horaires sont conçues à long terme pour chaque jour de la semaine et suivent la courbe d'affluence des passagers afin d'assurer la fluidité du transit des usagers.

Une table horaire définit également les durées et l'alternance d'exploitation entre phases établies et phases transitoires. Le terme *phase établie* désigne la période d'exploitation où l'ensemble des trains composant le carrousel sont espacés d'un même intervalle. Le passage d'un carrousel établi à un autre se fait par injection ou retrait de trains et est désigné par le terme *phase transitoire*.

Lors de la conception de la grille horaire, l'exploitant définit pour chaque phase établie, l'intervalle d'exploitation et les temps d'arrêt en station nominaux, cependant,

comme l'ont montré les travaux exposés en section 3.2.2, des gains substantiels peuvent être réalisés en modifiant la valeur de ces temps d'arrêt.

Ainsi, cette section a pour ambition d'utiliser la méthode hybride AG-OEP pour résoudre le problème d'optimisation des temps d'arrêt en station dans le cas d'une table horaire journalière, afin d'étudier les gains énergétiques réalisables à l'issue d'une journée d'exploitation type.

3.5.1 Phases transitoires : notion de train tenant l'horaire

3.5.1.1 Principe de l'injection/retrait

L'injection et le retrait automatique des trains aux terminus de la ligne ont pour objectif d'optimiser à chaque instant le nombre de rames en ligne théoriquement utile pour assurer le programme d'exploitation.

L'augmentation du nombre de rames nécessaires en ligne est obtenue en insérant dans le carrousel une nouvelle rame présente sur le parking d'injection du terminus, dans l'éventualité où aucune des rames présentes en ligne et arrivant à ce terminus ne peut assurer le prochain départ.

La diminution du nombre de rames nécessaire en ligne est, quant à elle, obtenue en retirant une rame lorsque le nombre de rames en ligne arrivant au terminus est supérieur au nombre de rames nécessaires pour assurer les prochains départs.

Selon les configurations rencontrées, un parking d'injection/retrait peut être présent sur un seul ou sur les deux terminus de la ligne de métro.

3.5.1.2 Notion de train tenant l'horaire

La notion de train qui tient l'horaire permet à la régulation de trafic de déterminer à quel moment doit intervenir l'injection/retrait d'un train sur la ligne en comparant la table horaire de départ théorique avec la table horaire d'arrivée théorique au terminus de départ. Trois cas de figures se présentent :

Exploitation en phase établie : Si un seul train en ligne est capable d'assurer le prochain départ avec un retard inférieur au seuil Δr_{inj} .

Condition d'injection : Si aucun train en ligne n'est en mesure d'assurer le prochain départ prévu avec un retard inférieur au seuil Δr_{inj} .

Condition de retrait : Si deux trains sont en mesure d'assurer le prochain départ avec un retard inférieur à un seuil Δr_{ret}

Les seuils Δr_{ret} et Δr_{inj} sont définis de manière à minimiser les mouvements d'injection/retrait en terminus. De plus, le seuil Δr_{inj} doit être supérieur à Δr_{ret} pour ne pas générer une condition d'injection temporaire qui sera suivi par l'occurrence d'une condition de retrait.

3.5.2 Méthodologie d'implémentation

Initialisation : Une fois que les intervalles d'exploitation nominaux ont été spécifiés pour les différentes phases établies composant la table horaire journalière, il convient d'initialiser le processus d'optimisation des temps d'arrêt en station. Pour cela, une population initiale de solution est définie.

Chaque solution est une liste des temps d'arrêt qui seront effectuées par les trains en station ordonnée selon les horaires nominaux d'arrivée en station des trains.

Comme illustré par l'équation (3.18), les temps d'arrêt composant cette population initiale sont pris aléatoirement dans une plage de variation définie en accord avec l'exploitant pour respecter au maximum les contraintes d'exploitation.

$$s_{i,nom} - \Delta s_{1i} \leq s_{i,j} \leq s_{i,nom} + \Delta s_{2i} \quad (3.18)$$

L'étape d'initialisation nécessite également de paramétrer correctement l'algorithme d'optimisation pour permettre une amélioration continue des solutions optimales à chaque itération, mais également de définir le nombre d'individus composant la population initiale.

Processus d'optimisation : L'étape initiale de la méthode d'optimisation consiste à évaluer la population initiale pour déterminer l'impact de chaque solution sur la consommation énergétique de la ligne.

Dans le cas d'une phase établie, cette évaluation s'effectue en simulant un carrousel établi effectuant un tour de boucle à vitesse nominale et effectuant les arrêts spécifiés par la solution, puis en déterminant les flux de puissance instantanée qui se produisent au sein du réseau électrique à chaque pas de temps grâce à la méthode de résolution itérative présentée en section 2.6.4. Ces flux de puissance permettent ensuite de calculer l'énergie électrique consommée par l'ensemble des trains circulant sur la ligne et de pouvoir ainsi comparer les solutions initiales.

Dans le cas d'une phase transitoire, le mode opératoire diffère sensiblement par la manière de simuler le carrousel transitoire, en effet, il est nécessaire de recréer la régulation de trafic qui s'effectue pour passer d'un carrousel établi à un autre. De fait, selon les cas d'étude, il est nécessaire d'insérer ou de retirer un certain nombre de trains, en suivant les règles de régulation utilisées par l'exploitant de la ligne de métro.

Ensuite, les algorithmes d'optimisation effectuent leurs routines jusqu'à ce que la condition d'arrêt basée sur la stagnation de l'optimum soit atteinte.

3.5.3 Résultats d'optimisation

La figure 3.19 présente une comparaison des puissances électriques moyennes consommées par le carrousel dans le cas de l'utilisation d'une table horaire nominale et dans le cas d'une table horaire optimisée.

La figure 3.20 illustre, quant à elle, l'évolution du gain énergétique global de la ligne de métro au cours de la journée d'exploitation.

Cette optimisation a été réalisée en utilisant la méthode hybride OEP-AG composée d'une cinquantaine d'individus. Le critère d'arrêt retenu porte sur la stagnation du coût de la fonction objectif : si sur un certain nombre d'itérations, l'optimum trouvé n'évolue pas, l'algorithme est déclaré convergent et la procédure d'optimisation est stoppée.

Pour le cas d'étude présenté ici, un gain énergétique journalier de l'ordre de 8% a été enregistré.

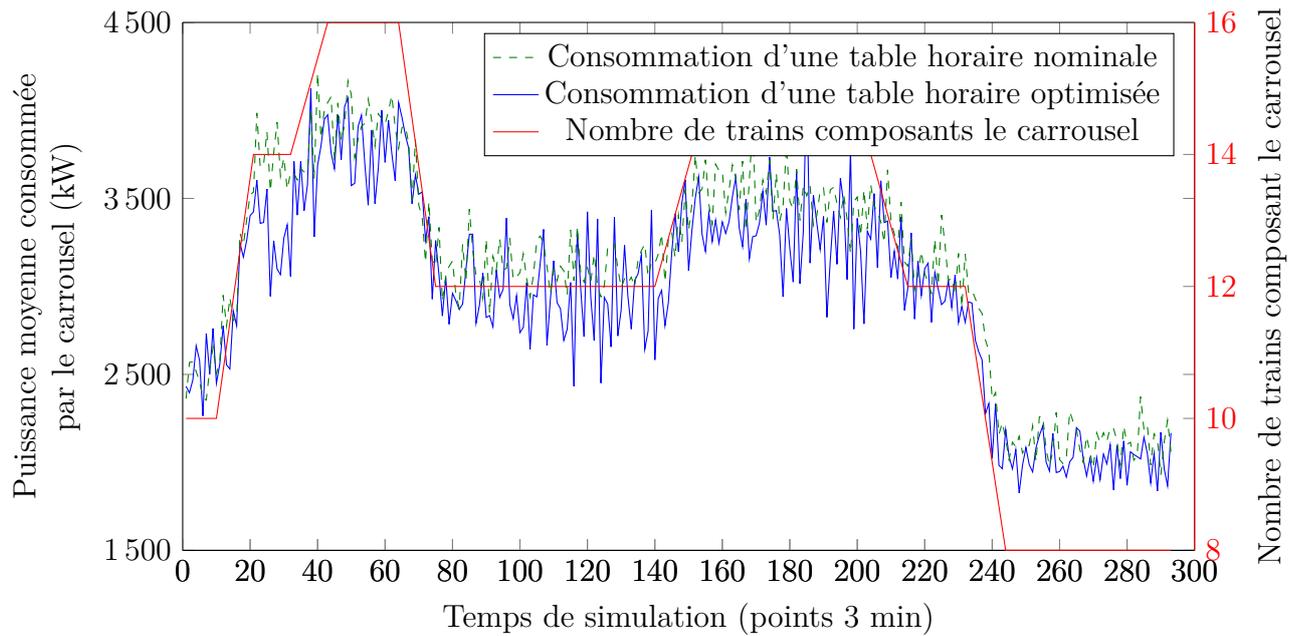


FIGURE 3.19 – Comparaison de puissances électriques consommées par une ligne de métro sur une journée d'exploitation pour deux tables horaires différentes.

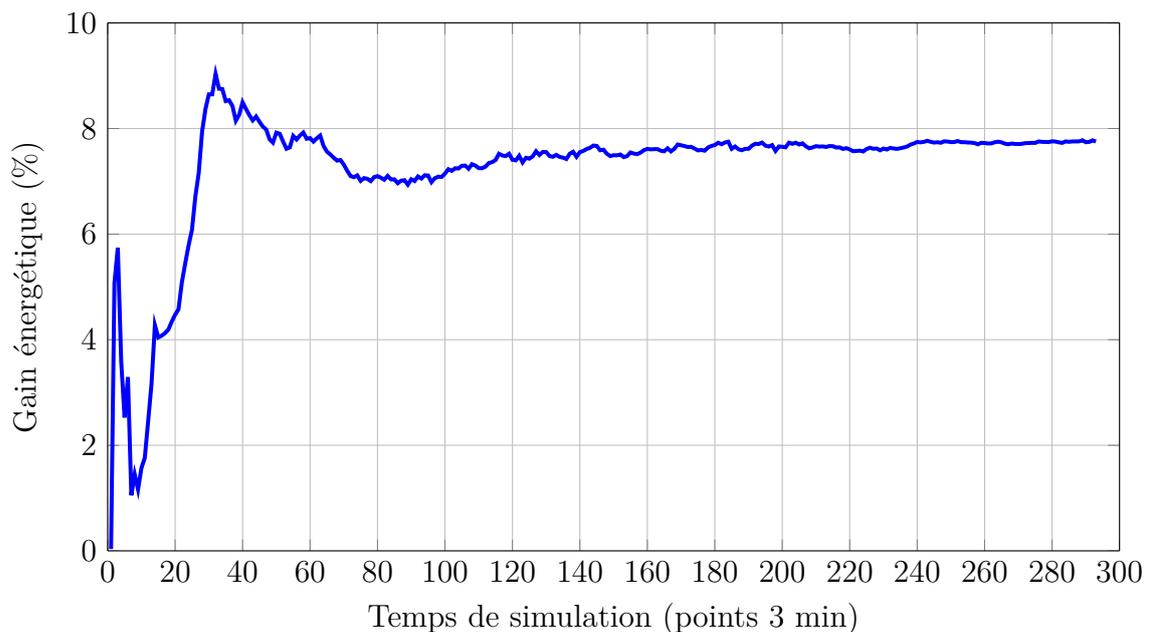


FIGURE 3.20 – Gain énergétique réalisé en exploitant la table horaire optimisée.

3.6 Limites de l'approche hors-ligne

Dans ce chapitre, l'objectif était de présenter des méthodes permettant d'optimiser des phases d'exploitations idéales sans aléas d'exploitation. Cependant, aux vues des performances des méthodes d'optimisation en terme de temps de calcul, l'utilisation d'une telle approche hors-ligne, n'est pas envisageable dans une approche dynamique.

En effet, le temps de calcul nécessaire pour effectuer l'optimisation d'une phase d'exploitation est trop élevée pour une utilisation de ces travaux en temps réel.

3.6.1 Temps de calcul

La contrainte sur le temps de calcul de la boucle d'optimisation est primordiale pour respecter la définition du temps réel.

Une application est ainsi qualifiée de temps réel lorsqu'elle a la capacité d'analyser un certain nombre d'informations (ou événements) issus d'un processus sans perdre un seul de ces événements, ou lorsque son temps de réponse est négligeable par rapport aux phénomènes physiques observés.

Dans le cadre de l'optimisation des temps d'arrêt en station, une mise en œuvre en temps réel signifierait que l'on serait capable de déterminer le temps optimal de stationnement que doit effectuer un train une fois que celui-ci est arrivé en station. La notion de temps réel doit alors s'appliquer avec une granularité de l'ordre de la dizaine de seconde, soit la durée moyenne d'un temps de stationnement nominal.

De fait, la durée de la boucle d'optimisation ne doit pas dépasser cette valeur afin de respecter le critère de temps réel.

Cependant, d'après le tableau 3.4, dans cet intervalle de temps, seule une solution peut être explorée par une méthode AG ou OEP, ce qui n'est ni suffisant pour assurer l'optimalité de la solution explorée ni pour atteindre une convergence.

3.6.2 Optimalité des solutions trouvées

Une autre limite de la méthodologie d'optimisation, développée dans ce chapitre, concerne l'optimalité des solutions trouvées. La méthode hybride OEP-AG permet de concilier les deux aspects fondamentaux d'une technique d'optimisation : la diversification et l'intensification.

Néanmoins, bien que cette méthode semble converger en un nombre fini d'itération, aucune garantie d'optimalité ne peut être apportée pour justifier l'atteinte de l'optimum global de l'espace des solutions. Il serait alors nécessaire de multiplier les essais pour tenter d'améliorer l'optimum trouvé, ce qui n'est pas réalisable avec la puissance de calcul à disposition.

3.7 Conclusion

3.7.1 Résumé des travaux effectués

Après avoir déterminé les paramètres sur lesquels il était possible d'influer afin de réaliser l'objectif de réduction de la consommation énergétique d'une ligne de métro, le problème d'optimisation a été formulé de manière à expliciter les contraintes à respecter et à les intégrer dans la définition de la fonction objectif.

Une première méthodologie d'optimisation de l'intervalle d'exploitation a ensuite été proposée pour déterminer le cadencement optimal d'une ligne de métro dans le cas d'une exploitation sans aléa. Cette première étape d'optimisation révèle que la consommation énergétique d'une ligne de métro peut être fortement réduite en modifiant le cadencement de la ligne de quelques secondes.

La deuxième étape d'optimisation présentée dans ce chapitre concerne la détermination des temps d'arrêt en station permettant de resynchroniser les phases d'accélération et de freinage des trains. De par l'importance de l'espace des solutions à explorer pour résoudre ce problème, l'utilisation d'une optimisation par métaheuristique a été privilégiée pour ses bonnes propriétés de convergence dans des espaces de grandes dimensions.

Les techniques d'optimisation par AG et OEP ont alors été développées pour assurer la résolution du problème, cependant, la nécessité d'assurer le compromis exploration-exploitation en un temps de calcul le plus faible possible a amené à définir une méthode hybride permettant de synthétiser les points forts de l'AG et de l'OEP dans un seul algorithme de recherche.

Moyennant une détérioration des performances en temps de calcul, cette méthode hybride permet alors d'augmenter grandement la densité de solutions jugées acceptables tout en assurant une convergence vers un optimum plus élevé que ceux trouvés par les méthodes AG et OEP.

Puis, l'ensemble de la méthodologie d'optimisation a alors été appliquée à une table horaire journalière et un gain énergétique d'environ 8% a été obtenu en considérant des conditions idéales d'exploitation ne comportant pas d'aléa.

Enfin, dans ce chapitre, l'approche hors-ligne qui a été proposée n'a permis de répondre que partiellement à la problématique initiale de la thèse. En effet, avec cette approche, il est possible d'effectuer une optimisation de la consommation énergétique d'une ligne de métro dans le cas d'une exploitation maîtrisée.

Cependant, il est encore nécessaire d'apporter une solution au problème de re-définition en temps réel des temps d'arrêt en station pour réduire la consommation énergétique tout en intégrant les perturbations de trafic qui se produisent dans les conditions réelles d'exploitation.

3.7.2 Perspectives

Ainsi, des études supplémentaires doivent encore être réalisées afin de répondre aux deux sous-problèmes suivants :

Réduction du temps de calcul de la boucle d'optimisation Il serait nécessaire soit d'augmenter la capacité de l'unité de calcul servant à effectuer l'optimisation, soit de diminuer le nombre d'itérations nécessaires pour effectuer l'optimisation ou alors de synthétiser la méthode d'optimisation par un approximateur universel. L'augmentation de la capacité de l'unité de calcul n'est pas envisageable pour des raisons économiques et logistiques tandis que la diminution du nombre d'itération n'est pas concevable car cela amènerait à détériorer la qualité des solutions trouvées. En revanche, l'implémentation d'un approximateur universel est parfaitement adaptée pour fournir une aide à la décision en un temps très faible [108].

Optimalité de l'aide à la décision Une première mesure consisterait à procéder à de multiples optimisations partant de points initiaux distincts et une seconde serait d'effectuer un apprentissage de l'ensemble des solutions explorées.

La première solution est en contradiction directe avec l'objectif de réduction du temps de calcul pour effectuer la procédure d'optimisation en temps réel, tandis que la deuxième solution présente l'avantage de pouvoir choisir l'action la plus adaptée au cas d'exploitation étudié.

Il est également à noter qu'une méthode mathématique exacte serait également une option envisageable pour déterminer l'optimum global de l'espace des solutions du problème. Cependant, cette piste n'a pas été étudiée.

Le chapitre suivant s'efforcera donc de concilier les deux contraintes précédentes afin de mettre en œuvre une méthode capable de fournir une solution optimale dans l'intervalle de temps alloué par le temps de stationnement des trains.