

3.1. Introduction

Dans ce chapitre on parle sur l'usage des meta_heuristique qui forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficiles souvent issus des domaines de la recherche et qui manipulent une ou plusieurs solutions, à la recherche de l'optimum.

Les algorithmes génétiques (AGs) sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple. On part avec une population de solutions potentielles (chromosomes) initiales arbitrairement choisies. On évalue leur performance (fitness) relative. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. On recommence ce cycle jusqu'à ce que l'on trouve une solution satisfaisante.

3.2. Optimisation à l'aide de l'algorithme génétique

Nous allons décrire brièvement les travaux d'application des métaheuristiques à population de solution en a choisira Les algorithmes génétiques pour la gestion de l'indisponibilité des sites dans la chaîne logistique.

3.3. Définition sur les algorithmes génétiques

Les algorithmes génétiques appartiennent à la classe des algorithmes évolutionnaires. Ce sont des métaheuristiques inspirées de l'analogie entre un processus d'optimisation et l'évolution des êtres vivants. La simulation des mécanismes de variation et de sélection œuvrant dans les processus évolutifs naturels est exploitée pour résoudre des problèmes artificiels d'optimisation [06].

Dans les algorithmes évolutionnaires, l'analogie avec l'optimisation consiste à considérées les solutions potentielles au problème comme des chromosomes. Ceux-ci sont manipulés par des opérateurs de sélection et mutation. Dans le cas des algorithmes génétiques, un troisième opérateur est utilisé, appelé opérateur de croisement. La qualité de la solution correspondant à chaque chromosome est quantifiée via sa propre fitness. Les procédures de croisement et de mutation on pour but de créer de manière permanente de nouveaux chromosomes.

Les algorithmes génétiques gardent la même taille de la population des solutions potentielles, appelées aussi individus. La population initiale est, généralement, choisie de manière aléatoire.

Le cycle d'évolution est répété jusqu'à ce qu'un critère de terminaison est atteint. Dans le but de faciliter le cycle d'évolution de l'AG, deux opérateurs fondamentaux sont créés : croisement et mutation – aussi la routine de sélection peut être prise comme un autre opérateur. Pour simplifier la procédure on choisit, un mécanisme de croisement à un point aléatoire.

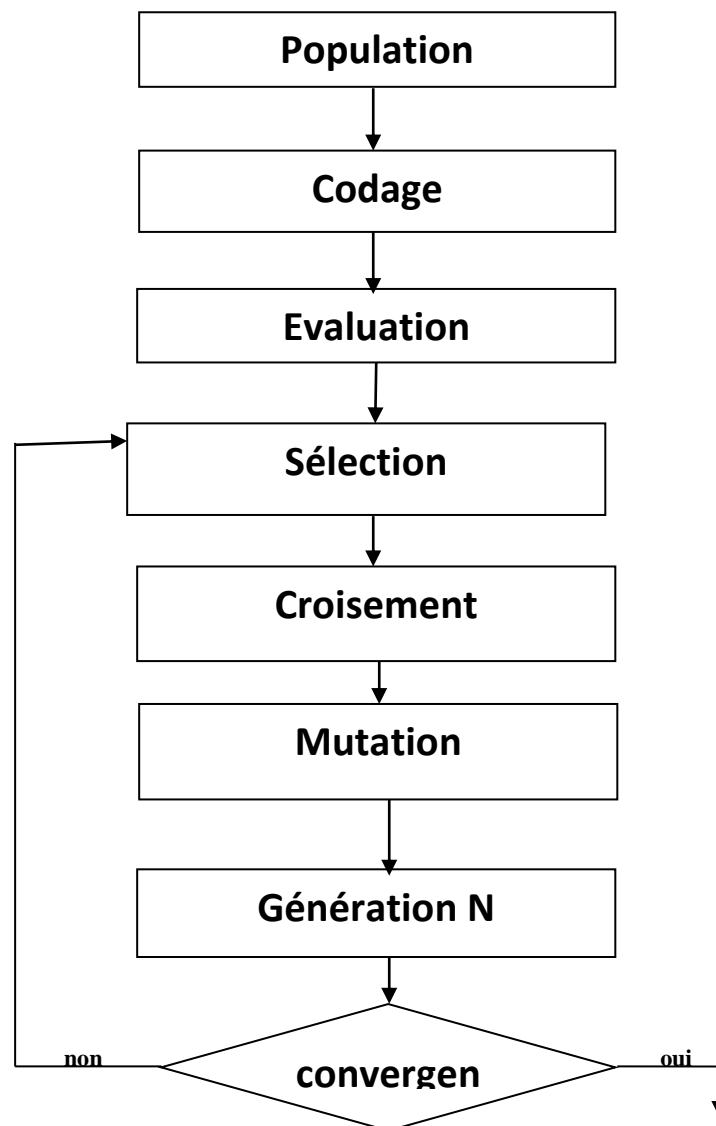
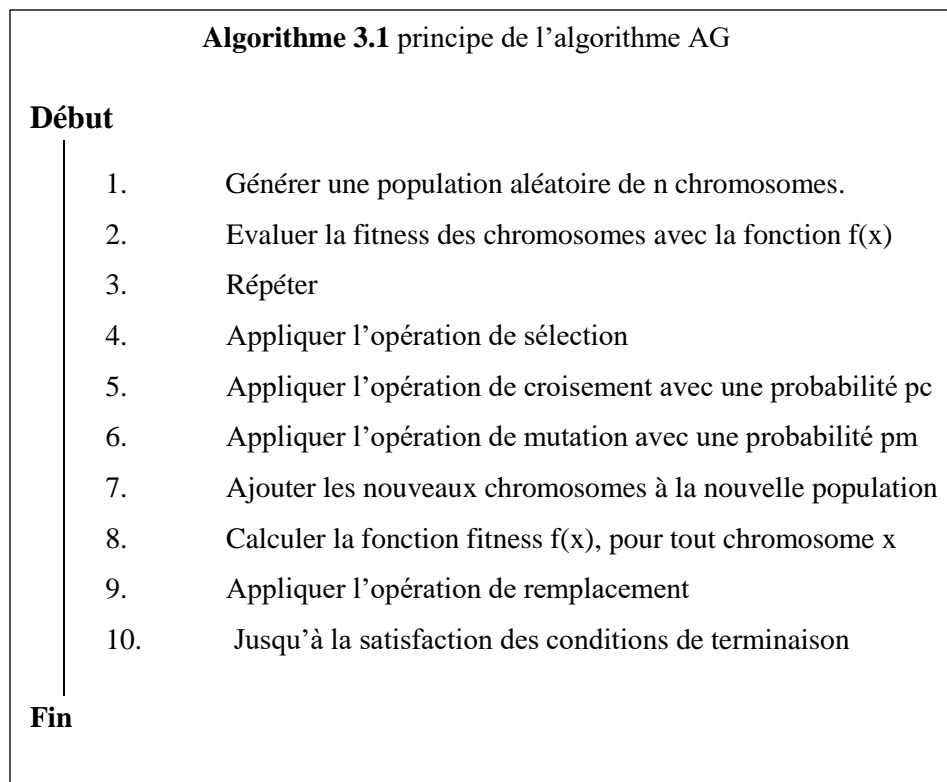


Figure 3.1 organigramme d'un algorithme évolutionnaire

L'algorithme génétique utilise :

- Une population d'individus (où chacune représente une solution potentielle du problème à optimiser).
- Une procédure de sélection, pour déterminer les individus susceptibles de se reproduire sur la base du critère de sélection naturelle.
- Ainsi que quelques opérateurs pour engendrer de nouveaux individus.

Un algorithme génétique générique à la forme suivante :



Les critères d'évaluation pour tester la convergence sont divers :

- Taux minimum désiré pour l'adaptation de la population au problème.
- Temps de calcul à ne pas dépasser.
- Combinaison des points précédents.

3.4. Fonctionnement général des algorithmes génétiques :

Les algorithmes génétiques sont employés afin de répondre aux problèmes n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique.

Les algorithmes génétiques renvoient en résultat des solutions appelées génotypes, répondant plus ou moins au problème, qui sont analysées et éliminées en suivant les principes de l'évolution afin de ne laisser <<survivre>> que les solutions optimales. Ces procédés permettent de tendre rapidement vers une solution optimale.

L'évolution et la sélection sont indépendantes du problème à résoudre. Par contre il y a des fonctions qui se modifient pour pouvoir répondre au problème :

- La fonction qui s'occupe de générer les familles de départ.
- La fonction inverse qui à partir d'un chromosome permet d'obtenir une solution par décodage du génome
- La fonction qui évalue l'adaptation d'une solution à un problème, sa pertinence

L'avantage des algorithmes génétiques, est qu'il n'est pas nécessaire de connaître la façon dont le problème est résolu. Il suffit juste de fournir une fonction ayant des paramètres permettant de coder l'information sous forme de gène et une autre qui permet d'évaluer la pertinence de la solution.

En conséquence les algorithmes génétiques sont efficaces dans divers domaines :

- Optimisation (de fonction, planification, ...)
- Apprentissage (classification, prédiction, robotique, ...)
- Programmation automatique (programme LISP, automates cellulaires, ...)
- Etude du monde réel (marchés économiques, comportements sociaux, ...). [47]

Les principales différences des algorithmes génétiques par rapport aux autres paradigmes sont les suivantes :

On utilise un codage des informations : on représente toutes les caractéristiques d'une solution par un ensemble de gènes, c'est-à-dire un chromosome, sous un certain codage (binaire, réel, code de Gray, etc. ...), valeurs qu'on concatène pour obtenir une chaîne de caractères qui est spécifique à une solution bien particulière (il y a une bijection entre la solution et sa représentation codée)

- On traite une population "d'individus", de solutions : cela introduit donc du parallélisme.
- L'évaluation de l'optimalité du système n'est pas dépendante vis-à-vis du domaine.
- On utilise des règles probabilistes : il n'y a pas d'énumération de l'espace de recherche, on en explore une certaine partie en étant guidé par un semi-hasard : en effet des opérateurs comme la fonction d'évaluation permet de choisir de s'intéresser à une solution qui semble représenter un optimum local, on fait donc un choix délibéré, puis de la croiser avec une autre solution optimale localement, en général la solution obtenue par croisement est meilleure ou du même niveau que ses parents, mais ce n'est pas assuré, cela dépend des aléas du hasard, et cela et d'autant plus vrai pour l'opérateur de mutation qui ne s'applique qu'avec une certaine probabilité et dans le cas où il s'applique choisit aléatoirement sur quel(s) locus(loi) introduire des modifications.[47]

Un algorithme génétique générique à la forme suivante :

- 1) Initialiser la population initiale P.
 - 2) Evaluer P.
 - 3) TantQue (Pas Convergence) faire :
 - a) P ' = Sélection des Parents dans P
 - b) P ' = Appliquer Opérateur de Croisement sur P '
 - c) P ' = Appliquer Opérateur de Mutation sur P '
 - d) P = Remplacer les Anciens de P par leurs Descendants de P '
 - e) Evaluer P
- FinTantQue

3.5. Le principe de l'algorithme génétique :

Un algorithme génétique permet trouver une solution en cherchant un extremum sur un ensemble de solutions donnés. Cet algorithme est construit selon :

- Le codage des éléments de la population. C'est à ce moment que l'on détermine les structures de données utiles. Bien entendu cette étape suit la modélisation du problème posé. La réussite des algorithmes génétiques est issue de la qualité de leur codage. Au début, un codage binaire était très utilisé mais maintenant, on travaille avec des codages réels plus pratiques pour des problèmes à variables réelles comme les problèmes d'optimisation.
- La génération de la population initiale. Ce mécanisme doit produire une population non homogène qui évoluera pour permettre de trouver une solution. La convergence ou la

divergence peuvent être plus ou moins rapide en fonction de la qualité de cette population initiale.

- Une fonction qui retourne le fitness du brin d'ADN pour pouvoir le comparer.
- Des outils qui permettront de créer de l'aléatoire comme la mutation afin de garantir l'exploration des solutions.
- Des paramètres afin de stopper le programme comme un critère de convergence ou divergence ou encore un nombre d'itérations pour effectuer le programme.

Les algorithmes évolutionnaires constituent une approche originale : on ne cherche pas à trouver une solution exacte mais d'obtenir un résultat satisfaisant le plus possible les critères, souvent contradictoires, des problèmes. Les solutions trouvées ne sont peut-être pas optimales mais sont souvent meilleures que les solutions obtenues grâce à des algorithmes plus conventionnels qui donneront, pour un temps identique, un résultat plus éloigné des besoins. Un individu est lié à un environnement par son code génétique contenu dans son ADN, une solution est liée à un problème par son indice de qualité.

3.6. Le codage

Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problème à variables réelles [17]. Chaque variable (information) représente alors un gène et toutes les valeurs que peuvent prendre cette caractéristique représentent les allèles possibles pour ce gène, et en concaténant tous ces gènes pour obtenir un chromosome qui lui représente une solution dans son intégralité.

- ✓ **Un gène:** est un ensemble de symboles représentant la valeur d'une variable. Dans la plupart des cas, un gène est représenté par un seul symbole (un bit, un entier, un réel..)
- ✓ **Un chromosome :** est une suite de gène, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position : son locus sur le chromosome en question. [47]
- ✓ **Un individu:** est composé d'un ou de plusieurs chromosomes. Il représente une solution possible au problème traité.
- ✓ **Une population:** est représentée par un ensemble d'individus (i.e. l'ensemble des solutions du problème).

✓ **Une génération:** est une succession d'itérations composées d'un ensemble d'opérations permettant le passage d'une population à une autre.

Donc chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble d'individus :

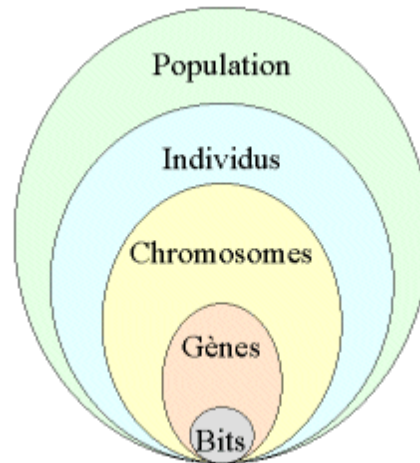


Figure 3.2 les cinq niveaux d'organisation d'un algorithme génétique

Il ya a trois principaux type de codage utilisables, et on peut passer de l'un à l'autre relativement facilement :

- Le codage binaire : c'est le plus utilisé. Chaque gène dispose du même alphabet binaire $\{0, 1\}$, un gène est alors représenté par un entier long (32 bits), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes.

0	1	1	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---

Figure 3.3 Codage binaire d'un chromosome

- Le codage réel : cela peut-être utile notamment dans le cas où l'on recherche le maximum d'une fonction réelle. Exemple : si le chromosome contient trois variable donc trois gènes et si le premier a la valeur $X1=3.256$ donc son codage sera 10010011.

0.23	1.25	10.5	0.48	3.00	-20.87
------	------	------	------	------	--------

Figure 3.4 Codage réel d'un chromosome

- Le codage de Gray : le codage de Gray est un codage qui a comme propriété qu'entre un élément n et un élément $n+1$, donc voisin dans l'espace de recherche, un seul bit diffère.

3.7. Les opérateurs de l'algorithme génétique

Les opérateurs utilisés dans l'AG emploient les paramètres suivant :

3.7.1. L'opérateur de sélection :

L'opérateur de sélection est chargé de définir quels seront les individus de p qui vont être dupliqués dans la nouvelle population p' et vont servir de parents (application de l'opérateur de croisement) [17] Soit n le nombre d'individus de p , on doit en sélection $n/2$ (l'opérateur de croisement nous permet de repasser à n individus).

On trouve essentiellement quatre types de méthodes de sélection différentes :

- La méthode de la "loterie biaisée" (roulette wheel)

La sélection des individus par le système de la roulette s'inspire des roues de loterie. A chacun des individus de la population est associé un secteur d'une roue. L'angle du secteur étant proportionnel à la qualité de l'individu qu'il représente. Vous tournez la roue et vous obtenez un individu. Les tirages des individus sont ainsi pondérés par leur qualité. Et presque logiquement, les meilleurs individus ont plus de chance d'être croisés et de participer à l'amélioration de notre population. [06]

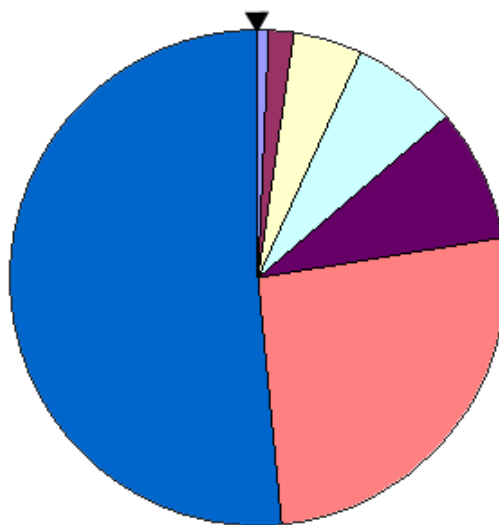


Figure 3.5 Le schéma de roulette [06]

- La sélection par range :

La sélection par rang est une variante du système de roulette. Il s'agit également d'implémenter une roulette, mais cette fois ci les secteurs de la roue ne sont plus proportionnels à la qualité des individus, mais à leur rang dans la population triée en fonction de la qualité des individus.

D'une manière plus parlante, il faut trier la population en fonction de la qualité des individus puis leur attribuer à chacun un rang. Les individus de moins bonne qualité obtiennent un rang faible (à partir de 1). Et ainsi en itérant sur chaque individu on finit par attribuer le rang N au meilleur individu (où N est la taille de la population). La suite de la méthode consiste uniquement en l'implémentation d'une roulette basée sur les rangs des individus. L'angle de chaque secteur de la roue sera proportionnel au rang de l'individu qu'il représente.

- La méthode "élitiste" :

Cette méthode de sélection permet de mettre en avant les meilleurs individus de la population. Ce sont donc les individus les plus prometteurs qui vont participer à l'amélioration de notre population. Cette méthode a l'avantage de permettre une convergence (plus) rapide des solutions, mais au détriment de la diversité des individus. On prend en effet le risque d'écarter des individus de piètre qualité, mais qui aurait pu apporter de quoi créer de très bonnes solutions dans les générations suivantes.

Une autre possibilité relevant aussi du domaine de l'élitisme, pour s'assurer que les meilleurs individus feront effectivement partie de la prochaine génération, est tout simplement de les sauvegarder pour pouvoir les rajouter à coup sûr dans la population suivante.

Les méthodes de sélection permettent de déterminer quels individus nous allons croiser.

- La sélection par tournois :

Le principe de la sélection par tournoi augmente les chances pour les individus de piètre qualité de participer à l'amélioration de la population. Le principe est très rapide à implémenter. Un tournoi consiste en une rencontre entre plusieurs individus pris au hasard dans la population. Le vainqueur du tournoi est l'individu de meilleure qualité. Vous pouvez

choisir de ne conserver que le vainqueur comme vous pouvez choisir de conserver les 2 meilleurs individus ou les 3 meilleurs. A vous de voir, selon que vous souhaitez créer beaucoup de tournois, ou bien créer des tournois avec beaucoup de participants ou bien mettre en avant ceux qui gagnent les tournois haut la main. Vous pouvez faire participer un même individu à plusieurs tournois. Une fois de plus, vous êtes totalement libre quant à la manière d'implémenter cette technique de sélection. [50]

- **La sélection universelle stochastique**

Cette méthode semble être très peu utilisée et qui plus est possède une variance faible, donc introduit peu de diversité, nous n'entrerons donc pas dans les détails, on se contentera d'exposer sa mise en œuvre. On prend l'image d'un segment découpé en autant de sous segments qu'il y a d'individus. Les individus sélectionnés sont désignés par un ensemble de points équidistants [47].

3.7.2. L'opérateur de croisement :

Une fois certains individus sélectionnés, on les fait se reproduire entre eux, pour cela on utilise l'opérateur croisement. C'est l'opérateur essentiel de recherche d'un algorithme génétique. Il combine les génotypes de deux individus pour en obtenir deux nouveaux. Avec cet opérateur, les génotypes sont vus comme une chaîne de nombres binaires. Plusieurs méthodes de croisement sont utilisées : [33]

- **Le croisement n-point**

Ce type de croisement consiste à choisir n-points de coupures ($n=1, 2, \dots$), puis échanger les fragments de gènes délimités par les points de coupure choisis. On distingue dans ce type deux autres types de croisement qui sont très utilisés: le croisement 1-point et le croisement 2-points.

- **Croisement en un point** : on choisit au hasard un point de coupure identique sur les deux génotypes et on échange les fragments situés après le point de coupure pour donner les deux nouveaux génotypes. Comme le montre la figure 3.6.

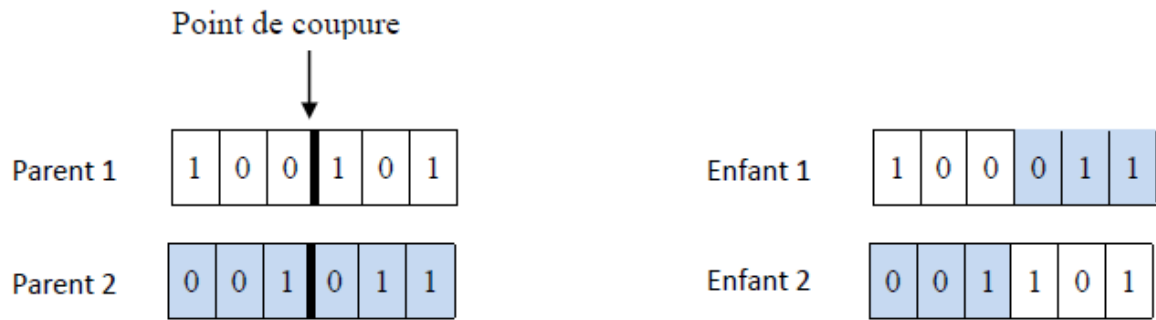


Figure 3.6 Opérateur Croisement a un point

- **Croisement en deux points** : on choisit au hasard deux points de croisement et on échange les fragments situés entre ces deux points. Comme le montre la figure 3.7

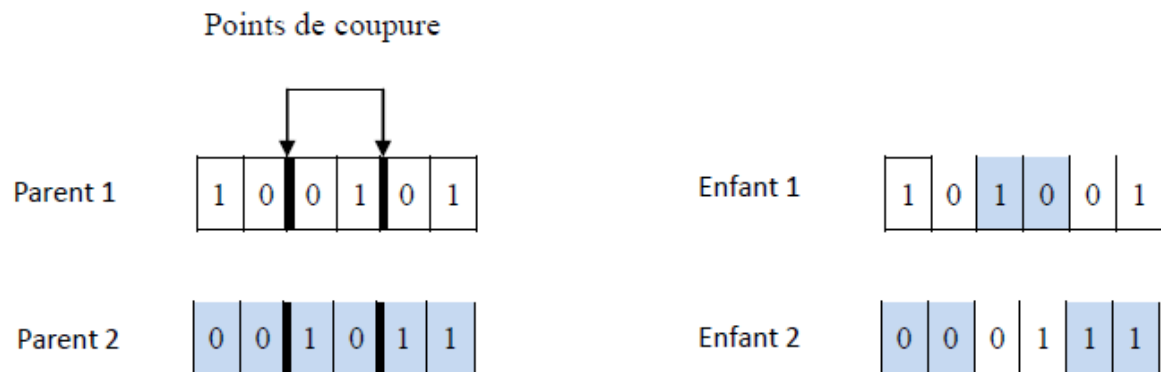


Figure 3.7 Opérateur Croisement a trois pointes

- **croisement uniforme** : La mise en œuvre de ce procédé est fort simple, elle consiste à définir de manière aléatoire un "masque", c'est-à-dire une chaîne de bits de même longueur que les chromosomes des parents sur lesquels il sera appliqué. Ce masque est destiné à savoir, pour chaque locus, de quel parent le premier fils devra hériter du gène s'y trouvant; si face à un locus le masque présente un 0, le fils héritera le gène s'y trouvant du parent n° 1, si il présente un 1 il en héritera du parent n° 2. La création du fils n° 2 se fait de manière symétrique : si pour un gène donné le masque indique que le fils n° 1 devra recevoir celui-ci du parent n° 1 alors le fils n° 2 le recevra du parent n°2, et si le fils n° 1 le reçoit du parent n° 2 alors le fils 2 le recevra du parent n° 1.

Donc Ce type de croisement est fondé sur la probabilité. En fait, il permet la génération d'un enfant en échangeant chaque gène des deux parents avec une probabilité égale à 0.5, comme le montre la figure 3.8

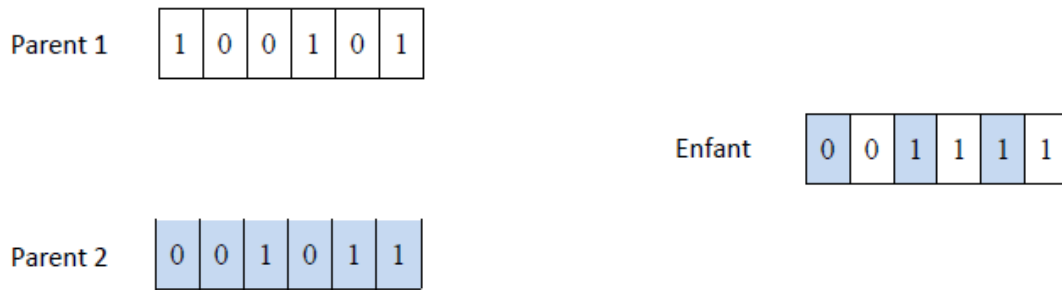


Figure 3.8 croisement uniforme

3.7.3. L'opérateur de mutation :

L'opérateur de mutation permet d'apporter des modifications partielles (légères) aux chromosomes avec une certaine probabilité très faible. L'objectif de l'opérateur de mutation est d'aider l'algorithme à échapper au problème de stagnation de la recherche causé par les optimaux locaux. Il permet d'explorer l'espace de recherche en apportant à l'algorithme la possibilité de couvrir la totalité de l'espace de recherche [44]. L'opérateur de mutation dépend du type de codage utilisé.

Cet opérateur consiste à changer la valeur allélique d'un gène avec une probabilité P_m très faible, généralement comprise entre 0.01 et 0.001. On peut aussi prendre $P_m = 1/\lg$ où \lg est la longueur de la chaîne de bits codant notre chromosome. Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire ; on peut donc résumer la mutation de la façon suivante : [14]

On utilise une fonction censée nous retourner true avec une probabilité P_m .

Algorithme 3.2 Principe de l'algorithme mutation d'AG

Pour chaque locus **faire**

Faire appel à la fonction

/ **Si** cette fonction nous renvoie true alors

/ / on inverse le bit se trouvant à ce locus

/ **Fin si**

/ **Fin pour**

L'opérateur de mutation modifie donc de manière complètement aléatoire les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population.

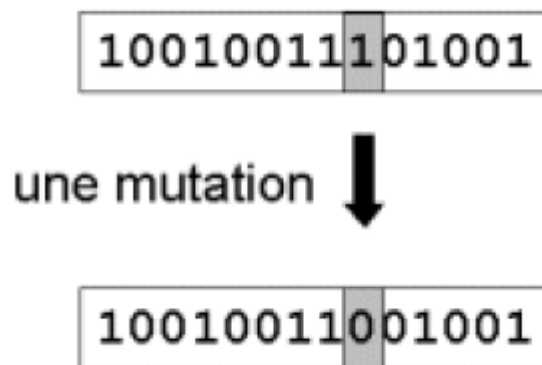


Figure 3.9 Opérateur mutation d'AG

La mutation est traditionnellement considérée comme un opérateur marginal bien qu'elle confère en quelque sorte aux algorithmes génétiques la propriété d'ergodicité (i.e. tous les points de l'espace de recherche peuvent être atteints). Cet opérateur est donc d'une grande

importance. Il a de fait un double rôle : celui d'effectuer une recherche locale et/ou de sortir d'une trappe (recherche éloignée). [41]

3.7.4. Evaluation de la fonction fitness

On appelle adaptation d'une séquence A une valeur positive que nous noterons $f(A)$. f est la fonction objectif ou fitness du problème à résoudre.

Un critère permettant de déterminer l'adaptation d'un individu par rapport à l'environnement afin de classer les individus entre eux. Dans le vocabulaire des algorithmes génétiques, on dénomme par "fitness" (traduction anglaise du mot adaptation) le critère à optimiser.

L'évaluation de la Fitness est généralement l'étape dans laquelle on mesure la performance de chaque individu. Pour pouvoir juger la qualité d'un individu et ainsi le comparer aux autres, il faut établir une mesure commune d'évaluation. Aucune règle n'existe pour définir cette fonction, son calcul peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation. [39]

3.7.5. Opérateur de remplacement :

Cet opérateur est le plus simple, son travail consiste à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation (la population P') dans la population de leurs parents (la population P).

Ce faisant il vont remplacer une certaine proportion de ceux-ci, proportion pouvant bien sûr être choisie. Le rapport entre le nombre d'individus nouveaux allant être introduits dans la population P et le nombre d'individus de cette population est connu sous le nom de génération gap.

On trouve essentiellement deux méthodes de remplacement différentes :

- Le remplacement stationnaire : consiste à remplacer automatiquement les parents par les enfants. Ceci induit une grande diversité de population.
- Le remplacement élitiste : on ne garde que les individus performants en regardant sur la génération complète et non que sur les parents.

3.8. Avantages et inconvénients des Algorithmes génétiques :

3.8.1. Avantages :

- Elimination de solution non valide.
- Permet de traiter des espaces de recherche important (beaucoup de solution, pas de parcourt exhaustif envisagé).
- Nombre de solution important.
- Relativité de la qualité de la solution selon le degré de précision demandé.

3.8.2. Inconvénients :

- Nécessitent plus de calculs que les autres algorithmes méta heuristiques (notamment la fonction évaluation).
- Paramètres difficiles à fixer (taille population, % mutation).
- Choix de la fonction d'évaluation délicat.
- Pas assuré que la solution trouvée est la meilleure, mais juste une approximation de la solution optimale.
- Problème des optimums locaux si paramètres mal évalués.

3.9 .Problème de conception des chaînes logistiques

La conception d'une chaîne logistique concerne généralement plusieurs phases allant de l'approvisionnement, la production à la distribution.

Pour l'activité d'approvisionnement, le décideur a besoin d'identifier les fournisseurs potentiels à choisir pour alimenter les différentes usines en matières premières, en composants et en produits semi-finis. Pour l'activité de production, il a à déterminer les meilleures localisations de ses usines pour assurer les performances et la rentabilité des activités de production. Pour obtenir une chaîne de la distribution efficace, le décideur doit déterminer le nombre et la localisation de ses différents centres de distribution. Une fois le choix des différents fournisseurs, usines et centres de distribution établi, il est indispensable de trouver la meilleure structure de connexion reliant ces sites. Ainsi, Cette conception sera faite dans le respect des contraintes économiques, sociales et environnementales tout en minimisant les coûts, maximisant la satisfaction des clients, réduisant au minimum l'impact sur l'environnement, etc.

3.10 .Problématique considérée

Ce travail aborde deux problèmes liés au domaine des chaînes logistiques : la conception des réseaux/ chaînes logistiques et la gestion des indisponibilités des sites logistiques. Notre démarche est divisée en deux étapes distinctes. Le but de la première étape E1 consiste à trouver la structure optimale du réseau logistique à travers la prise en compte de trois types de décisions qui sont : localisation des centres de distribution (DCs), affectation des détaillants aux DCs et choix des fournisseurs, une fois la structure du réseau définie, nous considérons dans la deuxième étape E2 que certains DCs peuvent devenir indisponibles. Ainsi, l'objectif de cette étape est l'optimisation de la gestion de ces indisponibilités

Notre problème consiste à trouver les meilleures localisations des DCs (chaque DC est identifié par la zone de sa localisation) ainsi que la meilleure affectation des zones de demandes/détaillants aux DCs et des DCs aux fournisseurs ainsi que la répartition des ordres d'approvisionnement sur les différentes connexions de transport reliant les fournisseurs et les DCs.

3.11 .Approche de résolution de notre problème

➤ Optimisation multicritères par l'algorithme génétique

Avant d'aborder les algorithmes génétiques multicritères, nous rappelons brièvement les principes de base de l'optimisation multicritères pour permettre au lecteur une meilleure compréhension de notre approche.

L'optimisation multicritères s'adresse aux problèmes de prise de décision en présence de critères multiples, éventuellement contradictoires. Ces problèmes de prise de décisions sont rencontrés à toutes les échelles de gestion, dans tous les domaines économiques, sociaux et environnementaux, la plupart des décisions faisant l'objet d'étude multicritères sont de natures complexes avec des conséquences importantes, l'aspect conflictuel des critères, le manque et parfois l'absence d'informations liées au problème sont souvent avancés comme arguments justifiant sa complexité.

Donc dans un premier temps, la résolution du notre problème consiste pendant l'étape E₁ en la prise de trois types décisions qui sont : localisation des DCs, affectation des détaillants aux DCs et choix des fournisseurs. Ces trois premières décisions sont obtenues en considérant l'existence d'une seule et unique liaison de transport entre chaque couple fournisseur- DC, ce

problème est résolu en utilisant un algorithme génétique cet algorithme permet d'obtenir la structure optimale de notre réseau de distribution.

Partant de la structure de la chaîne logistique obtenue durant la première étape, nous simulons le comportement de cette chaîne pendant la seconde étape E_2 , et ceci face à la présence d'éventuels défaillances d'un ou plusieurs DCs.

Nous supposons, que les DCs ne peuvent pas être indisponibles simultanément. Le but de cette étape est d'optimiser la gestion des indisponibilités des DCs tout en minimisant le coût total généré.

3.12. Conclusion :

Donc dans ce chapitre comprendre qu'un algorithme génétique vous donne une grande liberté dans le paramétrage et dans l'implémentation des différents traitements. Libre à vous ensuite de modifier tel ou tel paramètre si les solutions obtenues ne vous conviennent pas, et ces algorithmes, peuvent permettre de trouver une solution optimale et approchée très rapidement.

Les algorithmes génétiques ont l'énorme avantage de pouvoir être appliqués dans un grand nombre de domaines de recherche de solution, lorsqu'il n'est pas nécessaire d'avoir la solution optimale, qui prendrait par exemple trop de temps et de ressources pour être calculée.

Et dans le chapitre suivant nous verrons l'implémentation des étapes des algorithmes génétiques appliquées sur notre problème : conception et gestion des indisponibilités des sites dans la chaîne logistique.