

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION DES PLANS
DE RÉHABILITATION DES CONDUITES D'AQUEDUC
PAR L'ALGORITHME GÉNÉTIQUE

NATHALIE OUM
DÉPARTEMENT DES GÉNIES CIVIL, GÉOLOGIQUE ET DES MINES
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CIVIL)
AVRIL 2008



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-41573-3

Our file Notre référence

ISBN: 978-0-494-41573-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

OPTIMISATION DES PLANS
DE RÉHABILITATION DES CONDUITES D'AQUEDUC
PAR L'ALGORITHME GÉNÉTIQUE

présenté par: OUM Nathalie

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. MUSANDJI Fuamba, Ph.D., président

M. LECLERC Guy, Ph.D., membre et directeur de recherche

M. BARBEAU Benoît, Ph.D., membre

DÉDICACE

Je dédie ce mémoire à mon Seigneur Jésus-Christ,

Celui qui m'a enseigné que la fin d'une chose est mieux que son commencement.

Il ne suffit pas de commencer un projet mais de l'achever.

REMERCIEMENTS

Je tiens à remercier M. Guy Leclerc pour sa supervision, sa rigueur, sa patience, sa compréhension et pour l'intérêt particulier qu'il a manifesté à l'endroit de ce projet.

Je tiens également à remercier ma famille qui m'a beaucoup encouragé et conseillé, ainsi que mes amis qui m'ont soutenu et motivé tout au long de ce projet, et plus particulièrement mon ami, M. Cosmas A. Murunga, ma mère spirituelle, Mme Sidonie Aithnard, mon collègue et patron, M. Gaétan Demers.

Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Enfin et surtout je remercie le Seigneur Jésus-Christ qui m'a toujours fortifiée.

RÉSUMÉ

L'algorithme génétique est utilisé pour optimiser les dates d'intervention sur les conduites d'eau d'un petit réseau de distribution. Pour cela, deux modèles sont étudiés : l'un avec contraintes budgétaires et l'autre sans contraintes budgétaires. L'objectif est de minimiser le coût total du système sur un horizon de temps déterminé. Cette étude met en œuvre les étapes de cet algorithme pour les deux modèles analysés. Les avantages et les inconvénients de l'utilisation d'un tel outil pour optimiser les plans de renouvellement d'un réseau d'aqueduc sont notés. Les variables de décision sont les dates d'intervention sur les conduites d'eau et les types d'intervention – la réhabilitation et le remplacement.

Mots clés : aqueducs, réseau de distribution d'eau potable, réhabilitation, remplacement, optimisation, coûts, algorithme génétique.

ABSTRACT

This paper details how genetic algorithm has been used to optimize dates of intervention of the water mains of a small network. Two models have been studied, one with budget constraints and the other without. The goal is to minimize the total cost of the system throughout the planning phase. This study elaborates the algorithm process used in the two different models. The advantages and disadvantages of using such a tool in the optimization plans of water distribution system renewal are also stated. The decision variables used are: the dates of intervention on the water mains, and the types of intervention – rehabilitation and replacement.

Keywords: aqueducts, water distribution system, rehabilitation, replacement, optimization, costs, genetic algorithm.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ.....	vi
ABSTRACT.....	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES ANNEXES	xiii
AVANT-PROPOS	xiv
INTRODUCTION	1
CHAPITRE 1 ALGORITHME GÉNÉTIQUE	5
1.1.Introduction	5
1.2.Mécanismes	7
1.2.1. Adaptation au problème posé	8
1.2.2. Initialisation de la population.....	13
1.2.3. Évaluation.....	14
1.2.4. Sélection	14
1.2.5. Croisement.....	20
1.2.6. Mutation	21
1.2.7. Remplacement.....	22
1.2.8. Arrêt.....	22
CHAPITRE 2 FORMULATION DU PROBLEME	24
2.1. Revue de la littérature	24
2.2. Analyse économique de shamir et howard.....	27
2.2.1. Analyse de l'historique de bris	28
2.2.2. Analyse des coûts	30
2.2.3. Détermination de la date optimale.....	31
2.3. Formulation mathématique du problème.....	33
2.3.1. Hypothèses retenues.....	34

2.3.2. Premier modèle : sans contraintes budgétaires.....	35
2.3.3. Deuxième modèle : avec contraintes budgétaires	35
2.4. Formulation génétique du problème.....	37
2.5. Méthodologie	46
CHAPITRE 3 PREMIER MODÈLE.....	49
3.1. Données d'entrées du modèle.....	49
3.2. Validation de l'algorithme génétique	50
3.2.1. Influence du coefficient A.....	52
3.2.2. Influence du coefficient R.....	52
3.2.3. Influence du taux $N(t_0)$	56
3.2.4. Influence du coût C_b	56
3.2.5. Influence du coefficient C_r	56
3.2.6. Choix d'un taux de bris linéaire.....	60
3.3. Influence des opérateurs et paramètres de l'algorithme.....	62
3.3.1. Opérateur de création du meilleur des meilleurs individus.....	62
3.3.2. Opérateurs de croisement.....	65
3.3.3. Influence du nombre d'individus M	65
3.3.4. Influence du nombre de conduites N	65
3.3.5. Influence du nombre maximal de génération N_{gen}	65
CHAPITRE 4 DEUXIÈME MODÈLE	70
4.1. Scénario sans contraintes	71
4.1.1. Choix de remplacer toutes les conduites du réseau	71
4.1.2. Choix de réhabiliter toutes les conduites du réseau	72
4.2. Premier scénario avec des contraintes budgétaires annuelles.....	75
4.3. Deuxième scénario avec une contrainte sur le coût total du système.....	79
4.4. Discussions sur les résultats obtenus	82
CONCLUSION	85
BIBLIOGRAPHIE	88
ANNEXES	90

LISTE DES TABLEAUX

Tableau 1.1	Correspondance entre le lexique génétique et le lexique mathématique	6
Tableau 1.2	Correspondance entre l'espace de codage et l'espace de solution	9
Tableau 1.3	Calcul de la probabilité de sélection des individus par le tirage de la roulette	15
Tableau 1.4	Calcul de la probabilité pour la sélection stochastique universelle.....	17
Tableau 1.5	Calcul de la probabilité pour la sélection par la roulette biaisée	18
Tableau 2.2	Coûts des interventions sur les conduites d'eau en dollars par mètre linéaire.....	34
Tableau 3.1	Valeurs typiques des paramètres d'entrée.....	50
Tableau 3.2	Valeurs des paramètres d'entrée pour un taux de bris linéaire.....	60
Tableau 4.1	Scénarios sans contraintes budgétaires	73
Tableau 4.2	Proportions des lots de conduites à remplacer sur l'horizon de temps	77
Tableau 4.3	Proportions des lots de conduites à remplacer ou réhabiliter	81

LISTE DES FIGURES

Figure 1.1	Séquences de l'algorithme génétique.....	7
Figure 1.2	Correspondance entre l'espace de codage et l'espace de solutions.....	9
Figure 1.3	Risque lors du décodage du chromosome	10
Figure 1.4	Domaine de faisabilité de la solution	10
Figure 1.5	Sélection par tirage de la roulette	16
Figure 1.6	Sélection stochastique universelle	17
Figure 2.1	Analyse des coûts de remplacement et de réparation actualisés pour déterminer la date optimale de remplacement.....	28
Figure 2.2	Décomposition de la représentation chromosomique	37
Figure 2.3	Déroulement de l'algorithme génétique	39
Figure 3.1	Simulations avec les valeurs d'entrée typiques des paramètres.....	51
Figure 3.2	Influence du coefficient A.....	53
Figure 3.3	Influence du coefficient R.....	55
Figure 3.4	Influence du taux $N(t_0)$	57
Figure 3.5	Influence du coût C_b	58
Figure 3.6	Influence du coefficient C_r	59
Figure 3.7	Simulations pour un modèle de taux de bris linéaire	61
Figure 3.8	Influence de l'opérateur de création du meilleur des meilleurs individus....	63
Figure 3.9	Influence du nombre d'individus M	66
Figure 3.10	Influence du nombre de conduites N	67
Figure 3.11	Influence du nombre maximal de génération N_{gen}	68
Figure 4.1	Scénario sans contraintes budgétaires avec 100% de remplacement.....	74
Figure 4.2	Scénario sans contraintes budgétaires avec 100% de réhabilitation.....	74
Figure 4.3	Budgets annuels en dollars courants	76
Figure 4.4	Date de remplacement avec contraintes budgétaires annuelles	77
Figure 4.5	Coût total du système avec contraintes budgétaires annuelles	78
Figure 4.6	Coût total du système avec contrainte sur le coût total	80

Figure 4.7 Date de remplacement ou de réhabilitation avec contrainte sur le coût total 80

Figure 4.8 Adaptation de la fonction d'évaluation avec les contraintes budgétaires..... 83

LISTE DES ANNEXES

Annexe 1	Développement mathématique pour le calcul de la date optimale	90
Annexe 2	Codes de l'algorithme génétique	92

AVANT-PROPOS

Dans la Ville de Montréal comme dans la plupart des grandes métropoles, la majorité des conduites des réseaux d'eau potable ont atteint, sinon dépassé leur durée de vie utile. La dégradation de ces réseaux se manifeste par des interruptions de service, des bris de conduites, un manque de pression au robinet des consommateurs, etc. Plus le réseau vieillit, plus les coûts de maintenance investis pour assurer un service minimal aux consommateurs augmentent. En faisant une analyse des coûts actualisés de maintenance et ceux du remplacement de la conduite en fonction du temps depuis le moment de sa mise en œuvre, on peut se rendre compte qu'il arrive un moment où il est plus économique de remplacer la conduite que de la maintenir. Ainsi, pour assurer la pérennité des conduites du réseau d'alimentation en eau, il est important de calculer le moment optimal pour procéder à leur remplacement.

Par ailleurs, les fonds alloués à la réfection des conduites d'eau potable sont insuffisants face à l'ampleur des conduites à restaurer. Les contraintes budgétaires ont d'ailleurs poussé les gestionnaires municipaux à se tourner vers des techniques dites de réhabilitation, autres que le remplacement conventionnel des conduites. De ce fait, la réhabilitation, moins coûteuse que le remplacement, devient de plus en plus privilégiée comme solution d'intervention sur la conduite. Sachant que les budgets sont limités, il faut être en mesure de proposer un outil qui permet aux gestionnaires de trouver la date optimale d'intervention (réhabilitation ou remplacement) des conduites en tenant compte des fonds disponibles. Ce projet de recherche s'inscrit dans cette vision en proposant un

modèle d'optimisation des plans d'intervention des conduites d'aqueduc, qui permet une planification des interventions à long terme sur le réseau en tenant compte des restrictions budgétaires.

INTRODUCTION

L'optimisation des plans d'intervention des réseaux d'aqueduc provient d'une volonté des gestionnaires municipaux d'assurer la pérennité des conduites d'eau potable en tenant compte des contraintes budgétaires.

L'optimisation appliquée aux réseaux de distribution d'eau potable porte sur quatre grands axes : 1) les plans d'ensemble à long terme, 2) la conception du réseau, 3) la configuration du réseau et 4) la réhabilitation. Les plans d'ensemble porte sur l'influence bilatérale entre le réseau d'aqueduc et la collectivité à laquelle il est rattaché, par exemple, la vérification de la nécessité d'installer un réservoir. La conception cible les détails de design d'un réseau, par exemple, le choix judicieux des matériaux et de l'emplacement des conduites. La configuration, qui est un cas particulier de la conception, est principalement centrée sur la disposition de tous les éléments du réseau qui sont essentiellement les conduites, les pompes, les vannes, les réservoirs, et les châteaux d'eau. Et enfin, la réhabilitation porte sur les plans d'amélioration du réseau existant au travers d'interventions immédiates ou planifiées sur une période de temps. Dans ce rapport, on va se concentrer sur la réhabilitation des conduites d'eau potable, c'est-à-dire, toute intervention sur la conduite pour améliorer ses capacités hydrauliques et structurales afin de garantir un service minimal ininterrompu.

Au niveau de la réhabilitation, plusieurs critères d'optimisation peuvent être traités. On les rassemble en trois grands critères : 1) économique, 2) de fiabilité et 3) de qualité. Le critère économique vise à réduire les coûts imputés à la réhabilitation d'une partie ou de

l'ensemble du réseau. Le critère de fiabilité vise à fournir un service adéquat et performant en tout temps, c'est-à-dire, un système qui fonctionne sous différentes conditions d'opérations parfois exceptionnelles comme la fermeture brusque d'une vanne. Le critère de qualité vise à diminuer l'effet nocif des réactions chimiques entre l'eau et la conduite qui se détériore, afin de fournir une eau saine aux consommateurs. Dans ce rapport, on a mis de côté les critères de fiabilité et de qualité. Seul le critère économique est abordé et le but de notre étude est de développer un outil d'identification des stratégies d'intervention (sur le réseau de conduites d'eau potable) les moins coûteuses.

L'objectif de ce projet est d'optimiser le coût total des interventions sur le réseau de conduites d'eau potable. Ce coût comprend : les coûts de réparation, les coûts de réhabilitation et les coûts de remplacement. L'outil envisagé cherche à minimiser le coût total actualisé du système sujet à des contraintes budgétaires, soit au niveau du budget total alloué pour une période de temps fixée, soit au niveau des budgets annuelles admissibles.

La méthodologie que l'on a développée pour optimiser les interventions sur le réseau se fait en deux étapes : 1) une programmation sans contraintes budgétaires (premier modèle) et 2) une programmation avec contraintes budgétaires (deuxième modèle). Le premier modèle consiste à faire une programmation des interventions avec un budget illimité. Il sert à valider l'algorithme génétique en comparant les résultats obtenus avec ceux de l'analyse théorique. Une fois l'étape de validation terminée, le modèle va permettre d'une part d'optimiser le coût total du système en déterminant le moment optimal pour intervenir

sur la conduite, et d'autre part d'évaluer le budget total requis pour le renouvellement du réseau sur un horizon de temps choisi. Dans le deuxième modèle, on impose des contraintes budgétaires. Deux scénarios vont être étudiés : le premier porte sur une contrainte au niveau du budget total alloué sur l'horizon de temps, et le second porte sur les budgets annuels admissibles sur l'horizon de temps. On va étudier l'impact de ces restrictions budgétaires sur le réseau. Pour cela, une première programmation des interventions sera faite par le premier modèle sans contraintes budgétaires, et le deuxième modèle proposera une nouvelle programmation en tenant compte des contraintes budgétaires.

La méthode de calcul choisie pour optimiser le système est l'algorithme génétique. Cet algorithme est utilisé environ depuis une dizaine d'année pour traiter certains problèmes d'optimisation dans les réseaux d'eau potable. Cette étude permettra d'une part de maîtriser les étapes de mise en œuvre de l'algorithme génétique et d'autre part, de déterminer les avantages et les inconvénients de l'utilisation d'un tel algorithme pour optimiser le coût total des interventions sur les réseaux d'eau potable. Cet algorithme se distingue des autres par quatre points spécifiques : 1) il utilise un codage des éléments de l'espace de recherche et non les éléments eux-mêmes, 2) il recherche la solution optimale à partir d'une population de points et non à partir d'un seul point, 3) il n'a pas besoin des notions de continuité, de dérivabilité ou de convexité des fonctions, et enfin 4) il utilise des processus stochastique dans sa mise en œuvre. Au début de chaque simulation, une population de solutions potentielles couvrant un large espace de décision (espace de

solutions) est générée. Celle-ci évoluera et convergera vers les meilleures solutions du problème (pas nécessairement optimales).

Ce rapport est organisé comme suit : le premier chapitre porte sur l'algorithme génétique utilisé comme technique de calcul dans le modèle d'optimisation développé. Le deuxième chapitre porte sur la formulation du problème d'optimisation. Le troisième chapitre traite du premier modèle utilisé pour valider les résultats de l'algorithme génétique. Le quatrième chapitre traite du deuxième modèle qui prend en compte les restrictions budgétaires. Une première programmation des interventions sera faite avec le premier modèle sans contraintes budgétaires, et le deuxième modèle proposera une deuxième programmation pour respecter au mieux les restrictions budgétaires. Et enfin, on terminera par une conclusion.

CHAPITRE 1

ALGORITHME GÉNÉTIQUE

1.1. INTRODUCTION

L'algorithme génétique est une méthode stochastique¹ de recherche d'optimum, qui simule les processus biologiques de sélection naturelle darwinienne et de la reproduction génétique, pour optimiser des fonctions non linéaires. Cette méthode a été développée par le professeur John Holland (1975) et a été popularisée par l'un de ses étudiants, David Goldberg (1989).

La sélection naturelle de Charles Darwin (1802–82) repose sur la lutte pour la survie des espèces voulant s'étendre dans un environnement contraint par son espace et ses ressources disponibles. De ce fait, les espèces les mieux adaptées à l'environnement tendent à survivre plus longtemps et à se reproduire plus facilement. Darwin remarqua aussi des variations génétiques entre les espèces, et il conclut que la sélection naturelle se fait par une apparition aléatoire dans les gènes de variations transmissibles à la progéniture. Les lois de variations (croisement et mutation) furent expliquées plus tard par Mendel (1822–84), puis par la génétique moderne.

Ainsi, l'algorithme génétique est caractérisé par deux opérateurs principaux:

- Les opérateurs dits *d'évolution* : la sélection, qui imite les processus d'évolution darwinienne pour créer des populations de génération en génération.

¹ Utilisant des processus aléatoires

- Les opérateurs dits *génétiques* : le croisement et la mutation, qui imitent les processus d'hérédité des gènes expliqués par la génétique moderne, pour créer de nouveaux gènes à chaque génération. Ils permettent de diversifier la population au cours des générations.

Le vocabulaire utilisé est directement inspiré de celui de la théorie de l'évolution et de la reproduction génétique.

Tableau 1.1 Correspondance entre le lexique génétique et le lexique mathématique

<i>Lexique génétique</i>	Lexique mathématique
<i>Population</i>	Ensemble de solutions potentielles
<i>Individus</i>	Solutions potentielles
<i>Gènes</i>	Variables
<i>Chromosome</i>	Partie d'une solution potentielle
<i>Allèle</i>	Bit
<i>Phénotype d'un individu</i>	Valeur de la fonction d'évaluation
<i>Génotype d'un individu</i>	Chaînes de bit qui caractérisent la solution
<i>Locus</i>	Position du bit

1.2. MECANISMES

L'algorithme génétique se fragmente simplement comme suit :

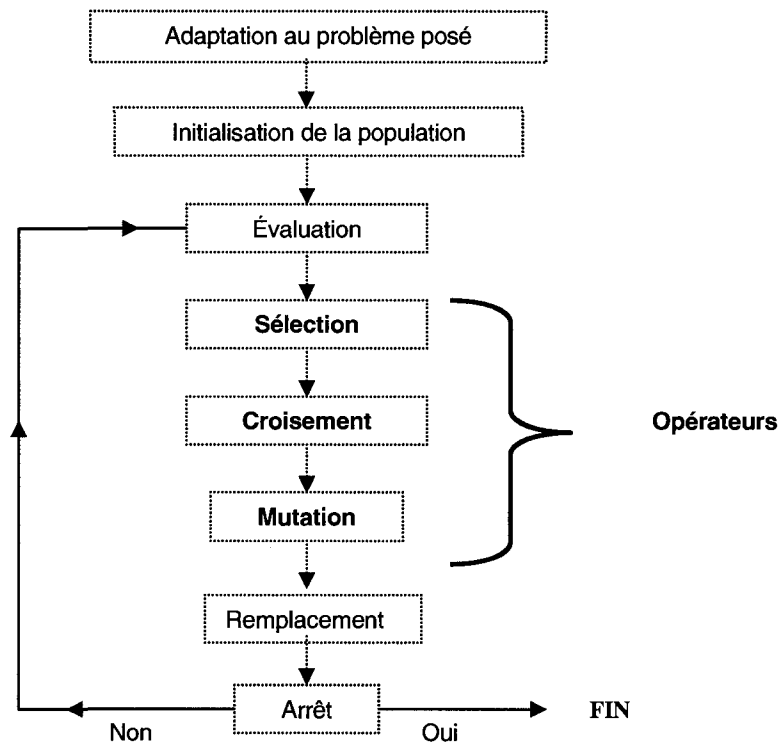


Figure 1.1 Séquences de l'algorithme génétique

Dans les sections suivantes, nous allons exposer chaque étape du déroulement de l'algorithme. Pour illustrer ce dernier, on l'applique au problème suivant :

Minimiser $F(x_1, x_2) = x_1^2 + x_2^2$, sous les trois contraintes suivantes : $x_1 \in [0 ; 5]$, $x_2 \in [0 ; 7]$ et $x_1 + x_2 \leq 10$.

1.2.1. Adaptation au problème posé

Tout d'abord, il faut définir : 1) le codage utilisé, 2) l'espace de recherche, 3) les fonctions économique et d'évaluation, et 4) les paramètres de l'algorithme génétique.

Codage

Une partie clé de l'algorithme génétique est de savoir comment coder une solution du problème par un chromosome ou une série de chromosomes. Pour cela, il faut choisir la représentation chromosomique (génotype) de chaque individu de la population. Les codages les plus populaires sont :

- *Le codage binaire* : le chromosome est une suite de 0 et 1. Il existe le codage binaire standard et le codage de Gray (où deux codes d'entiers successifs ne diffèrent que par un bit contrairement au codage binaire standard).
- *Le codage à caractères multiples* : le chromosome est une chaîne de caractères différents. Le codage binaire est un cas simple de ce type de codage.
- *Le codage flottant* : le chromosome est représenté par un nombre flottant (réel).
- *Le codage sous forme d'arbre* : le chromosome est un nœud d'un arbre. Ce type de codage permet de mieux visualiser les sélections et les mutations. Cependant l'analyse de l'ensemble de ces arbres est complexe et ce codage n'est approprié qu'aux problèmes de taille finie.

L'espace de codage contient l'ensemble des chromosomes tandis que l'espace de solutions contient l'ensemble des solutions du problème. La figure 1.2 illustre le passage de l'espace de codage à l'espace de solutions et vice versa.

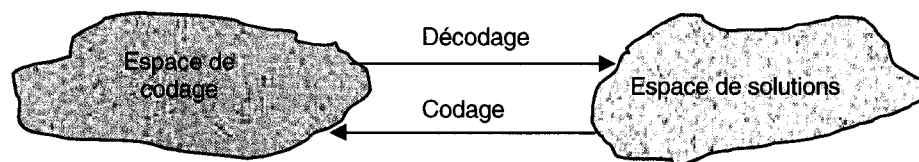


Figure 1.2 Correspondance entre l'espace de codage et l'espace de solutions

Dans le cas de notre exemple avec la fonction $F(x_1, x_2)$, on définit l'espace de solutions par l'ensemble de points (x_1, x_2) tels que $0 \leq x_1 \leq 5$ et $0 \leq x_2 \leq 7$. Le codage binaire standard est choisi pour coder cet ensemble de solutions. L'espace de codage est l'ensemble de chromosomes codés sur 5 bits de la manière expliquée dans le tableau

suivant. Posons $\Delta x_i = \frac{b_i - a_i}{2^5 - 1}$, avec $a_i \leq x_i \leq b_i$ pour $i = 1$ ou 2 .

Tableau 1.2 : Correspondance entre l'espace de codage et l'espace de solution

Chromosome	Valeur entière	Solution
00000	0	a_i
00001	1	$a_i + 1\Delta x_i$
00010	2	$a_i + 2\Delta x_i$
...	...	...
11110	30	$a_i + 30\Delta x_i$
11111	31	$a_i + 31\Delta x_i = b_i$

Une question importante concernant le codage et le décodage entre les chromosomes et les solutions est de savoir si la solution décodée tombe dans une région de solutions faisables,

surtout pour un problème d'optimisation avec contraintes. La figure 1.3 illustre ce risque lors du décodage du chromosome.

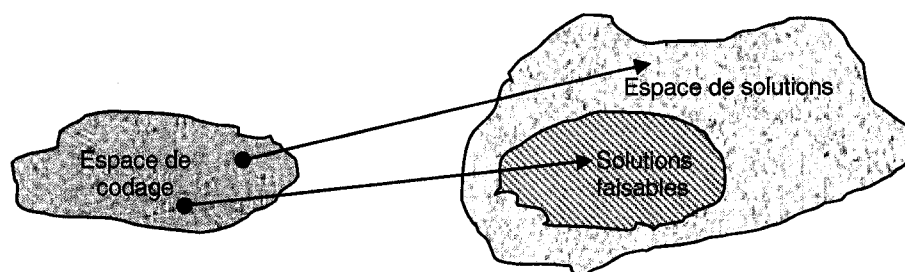


Figure 1.3 Risque lors du décodage du chromosome

Dans le cas de notre exemple, lors du décodage, il y a des risques que la solution obtenue ne soit pas faisable comme on peut l'observer sur la figure 1.4.

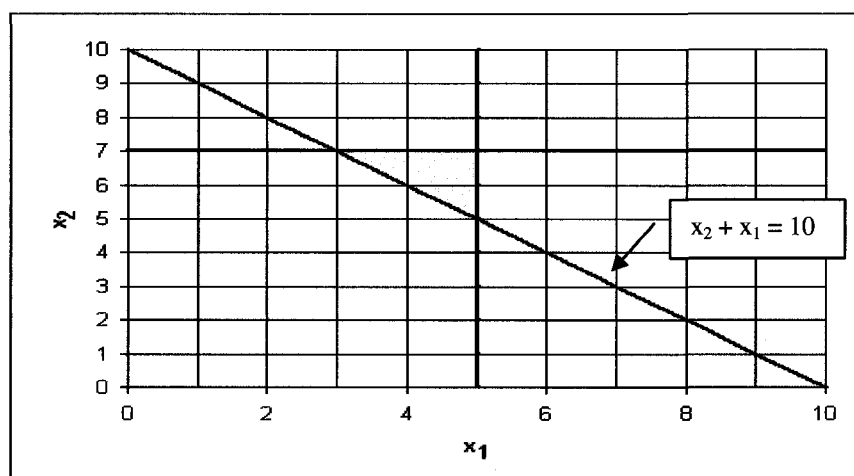


Figure 1.4 : Domaine de faisabilité de la solution

L'espace coloré représente l'espace de solutions. Cependant, les solutions faisables sont contenues uniquement dans l'espace coloré en jaune, où elles respectent la contrainte $x_1 + x_2 \leq 10$.

Espace de recherche

Il correspond à l'espace de solutions faisables. La prise en compte des contraintes permet de délimiter l'espace de recherche (voir Figure 1.4).

Fonctions économique et d'évaluation

- *La fonction économique*

Elle est utilisée pour mesurer la performance de chaque individu pour un problème précis. Dans un problème de minimisation, les meilleurs individus ont les plus petites valeurs de la fonction économique.

Dans le cas de notre exemple, la fonction économique est tout simplement la fonction à minimiser $F(x_1, x_2) = x_1^2 + x_2^2$.

- *La fonction d'évaluation*

Elle est utilisée pour transformer la valeur de la fonction économique en une mesure relative adaptée au problème. Par exemple, si la fonction économique est le coût total du système, pour un problème de minimisation, la fonction d'évaluation peut être choisie comme l'inverse de la fonction économique alors que pour un problème de maximisation, elle peut rester identique à la fonction économique.

La fonction d'évaluation doit aussi pouvoir tenir compte des solutions non faisables si le problème doit satisfaire des contraintes. Dans le cas d'une optimisation avec contraintes, trois stratégies pour traiter ces contraintes sont proposées :

- *Le rejet* : On rejette tous les chromosomes non faisables. Cette stratégie est populaire pour les algorithmes génétiques.
- *Le remplacement* : On prend le chromosome non faisable et on le remplace. Cela implique l'existence d'une procédure systématique qui remplace un chromosome non faisable par un chromosome faisable. Cette procédure est spécifique au problème étudié.
- *La pénalité* : On considère les solutions non faisables en formulant une fonction d'adaptation qui est élaborée par l'addition ou la multiplication d'une fonction de pénalité (statique ou dynamique) à la fonction économique.

Les deux premières stratégies ont l'avantage de ne pas générer des solutions infaisables mais le désavantage de ne pas considérer les solutions à l'extérieur de la zone de faisabilité. Si le problème est hautement contraint, les solutions infaisables peuvent être en grande proportion dans la population.

Paramètres de l'AG

- *La taille de la population N*

Elle affecte la performance et l'efficacité de l'algorithme. Une taille N faible risque de donner un échantillon insignifiant de la population et de faire converger rapidement l'algorithme vers un optimum local. Une taille N élevée demande beaucoup d'évaluations de la fonction d'adaptation par génération et peut entraîner une convergence d'une lenteur inacceptable.

- *Le taux de croisement p_c*

Il détermine la proportion des individus produits à chaque génération. Le nombre attendu d'individus est donc $p_c \times N$. Un taux p_c fort permet une meilleure exploration de l'espace et réduit les chances de stagner sur un optimum local; néanmoins un taux p_c trop élevé résulte en un gaspillage de temps de calcul dû à une exploration inutile des zones non prometteuses dans l'espace de recherche.

- *Le taux de mutation p_m*

Il permet de contrôler la fréquence à laquelle des mutations (voir section 1.2.6) sont effectuées dans la population. Si p_m est trop faible, certains gènes intéressants ne seront jamais essayés ou restitués (après un croisement). Si p_m est trop élevé, la recherche sera fondamentalement aléatoire et la convergence est ainsi freinée.

1.2.2. Initialisation de la population

La première population est générée aléatoirement dans l'espace de codage et celle-ci est appelée à évoluer au cours des générations. Cette population caractérise un ensemble de solutions potentielles au problème. Chaque individu de la population est donc une solution faisable ou non faisable.

Dans le cas de notre exemple, chaque individu est composé de deux chromosomes représentant respectivement les points x_1 et x_2 . Par exemple, un individu est :

01000	10001
x_1	x_2

Si la population comporte cinquante individus, nous allons générer deux séries de cinquante chromosomes à 5 bits pour x_1 et x_2 .

1.2.3. Évaluation

Une fois la fonction d'adaptation définie (étape précédente), on calcule la valeur d'adaptation de chaque individu de la population.

Dans le cas de notre exemple, la valeur de la fonction d'évaluation est donnée tout simplement par $F(x_1, x_2) = x_1^2 + x_2^2$. Selon le tableau 1.2, on a le décodage suivant :

$$x_1 = 01000 \rightarrow x_1 = 0 + 8 \frac{5-0}{2^5-1} \approx 1,29.$$

$$x_2 = 10001 \rightarrow x_2 = 0 + 17 \frac{7-0}{2^5-1} \approx 3,84.$$

D'où la valeur d'adaptation de l'individu égale à $F(x_1, x_2) \approx 1,29^2 + 3,84^2 \approx 16,41$.

1.2.4. Sélection

Compte tenu de la fonction d'adaptation de chaque individu, la sélection s'effectue en choisissant les individus qui vont s'apparier pour créer la génération suivante. Pour cela, les individus sont classés en fonction de leur valeur de la fonction d'adaptation. Habituellement, la probabilité de sélection d'un individu dépend du rang de ce dernier. On distingue trois types principaux de sélection (stochastique, déterministe et mixte).

La sélection stochastique

Elle favorise toujours les meilleurs individus mais de manière stochastique. Ceci laisse une chance aux individus moins performants d'être sélectionnés. Il se peut même que le meilleur individu ne soit pas sélectionné et qu'aucun enfant ne soit trouvé meilleur que le meilleur parent. Les techniques les plus connues sont :

- *La sélection par le tirage de la roulette*

Chaque individu a une chance d'être sélectionné en fonction de sa valeur de la fonction d'adaptation. La probabilité de sélection p_i de l'individu i avec une valeur d'adaptation F_i

sur une population de taille N est $p_i = \frac{F_i}{\sum_{i=1}^N F_i}$.

Dans le cas de notre exemple, prenons quatre individus. Le tableau 1.3 présente le calcul de la probabilité pour le tirage par roulette.

Tableau 1.3 : Calcul de la probabilité de sélection des individus par le tirage de la roulette

Espace de codage		Espace mathématique		Espace de solutions		Espace de solutions	
x_1	x_2	x_1	x_2	x_1	x_2	$F(x_1, x_2)$	p
10001	11101	17	29	2,74	6,55	50,40	0,60
00101	01011	5	11	0,81	2,48	6,82	0,08
01110	00011	14	3	2,26	0,68	5,56	0,07
11011	00110	27	6	4,35	1,35	20,80	0,25

Ensuite, un nombre aléatoire δ est généré entre 0 et 1. Ce nombre représente le marqueur tel qu'illustré à la figure 1.5. Pour utiliser l'image de la roulette, chaque individu a une case dans la roulette dont la longueur est égale à la probabilité de sélection de l'individu.

Ce dernier est sélectionné si le marqueur pointe sur sa case. Par exemple, si $\delta = 0,40$ alors le premier individu sera sélectionné.

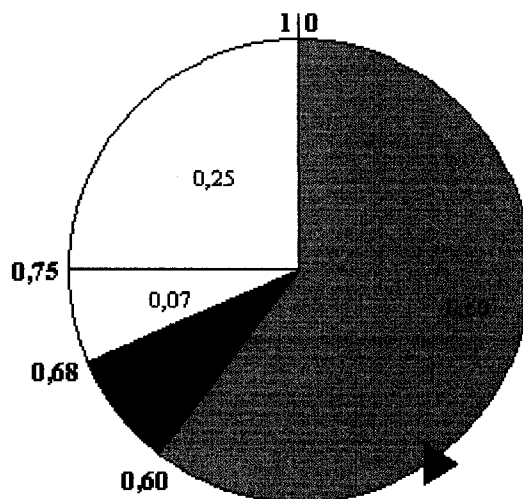


Figure 1.5 : Sélection par tirage de la roulette

Avec cette méthode, un individu faible a des chances d'être choisi autant de fois qu'il y a de place pour la génération suivante et le meilleur individu peut ne pas être sélectionné.

- *La sélection stochastique universelle*

Elle est construite comme la roulette, mais au lieu de posséder un seul marqueur, celle-ci possède N marqueurs espacés de manière équidistante, N étant la taille de la population. La longueur de la case du chromosome est $d_i = N \times p_i$. On sélectionne alors les individus autant de fois qu'un marqueur pointe sur leur case. Un individu est alors sûr d'être sélectionné un minimum de fois égal à la partie entière de p_i et un maximum de fois égal à

la partie entière par excès de p_i . Cette méthode permet de garder le nombre attendu de copies de chaque chromosome.

Dans le cas de notre exemple, $N = 4$.

Tableau 1.4 : Calcul de la probabilité pour la sélection stochastique universelle

p	$d = N \times p$	Partie entière	Partie entière par excès
0,60	2,40	2	3
0,08	0,32	0	1
0,07	0,28	0	1
0,25	1,00	1	1

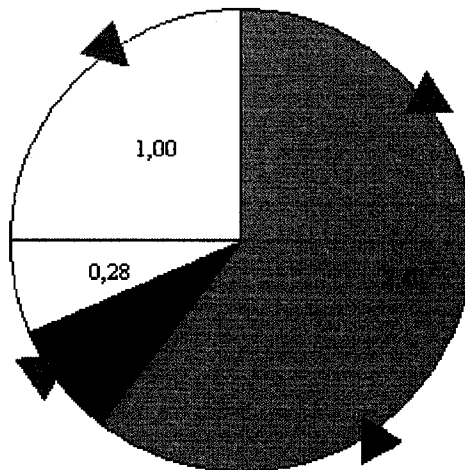


Figure 1.6 : Sélection stochastique universelle

- *La sélection par la roulette biaisée*

Les individus sont classés en fonction de la valeur de la fonction d'évaluation en ordre croissant. Soit r le rang de chaque individu, N la taille de la population et δ un nombre

généralisé aléatoirement entre 0 et 1, on sélectionne l'individu de rang r si δ appartient à

l'intervalle $\Delta = [\sum_{k=1}^{r-1} p_k , \sum_{k=1}^r p_k]$ avec $p_r = \frac{2}{N} \times \frac{r-1}{N-1}$.

Dans le cas de notre exemple, les quatre individus du tableau 1.4 sont classés dans le tableau suivant :

Tableau 1.5 : Calcul de la probabilité pour la sélection par la roulette biaisée

Individu	$F(x_1, x_2)$	r	p_r	Δ
1	50,40	1	0,00	[0 ; 0]
2	6,82	3	0,33	[0,17 ; 0,5]
3	5,56	4	0,50	[0,5 ; 1]
4	20,80	2	0,17	[0 ; 0,17]

L'individu le plus faible a une probabilité quasiment nulle d'être sélectionné.

La sélection déterministe

Elle favorise toujours une sélection des meilleurs individus de manière déterministe, ce qui ne laisse quasiment aucune chance aux individus moins performants. Pour cela, il faut trier l'ensemble de la population selon la valeur de la fonction d'adaptation et choisir les meilleurs. Les techniques les plus connues sont :

- *La sélection par troncature ou par bloc*

Les individus sont triés en fonction de leur valeur de fonction d'évaluation et les meilleurs parents sont sélectionnés pour créer la génération suivante. Dans la sélection par troncature, un seuil T est défini de telle sorte que $T\%$ des meilleurs individus sont sélectionnés et chacun reçoit environ $100/T$ copies. Dans la sélection par bloc, pour une

taille de la population donnée N , on crée k copies des N/k meilleurs individus. Les deux méthodes sont identiques si $k = N/T$.

- *La sélection de Brindel*

La probabilité de sélection de chaque individu i est $p_i = \frac{F_i}{\sum_{i=1}^N F_i}$, F étant la valeur de la

fonction d'évaluation. Le nombre de fois que l'on sélectionne chaque individu est la partie entière de $p_i \times N$, N étant la taille de la population. Le reste des individus à compléter dans la population est pris parmi les meilleurs de la liste triée.

La sélection mixte

Elle permet une sélection des individus à la fois de manière déterministe et stochastique. Les techniques les plus connues sont :

- *La sélection stochastique des restants de Brindel*

Lorsqu'on a fait la sélection de Brindel de façon déterministe, on sélectionne le reste des individus de manière aléatoire pour atteindre la taille de la population.

- *La sélection par tournoi*

La sélection par roulette est utilisée pour choisir des paires successives d'individus et celui qui a la plus grande valeur d'adaptation est sélectionné automatiquement. Le processus continue jusqu'à ce que le nombre d'individus sélectionnés atteigne la taille de la population. Dans le cas de notre exemple, à travers l'opérateur de sélection, les individus vont être sélectionnés deux à deux.

1.2.5. Croisement

Une fois les individus sélectionnés, l'opérateur de croisement favorise l'exploration de l'espace de recherche, par le brassage des individus. Il combine les génotypes de deux individus parents pour obtenir deux individus enfants. Plusieurs méthodes de croisement peuvent être utilisées, entre autres, le croisement multipoint et le croisement uniforme.

Le croisement en k points ou croisement multipoint

On choisit au hasard k points de croisement, avec k compris entre 1 et $L - 1$, où L est la longueur du génotype. Les fragments sont échangés de la manière illustrée ci-dessous.



Le croisement en un point est un cas particulier. Les fragments situés après le point de croisement sont échangés pour donner deux nouveaux génotypes.

Parent 1 :		Enfant 1 :	
Parent 2 :	11000	Enfant 2 :	110

Le croisement uniforme

Il s'agit d'un croisement multipoint dont le nombre de points de croisement n'est pas connu à l'avance et est aléatoire pour chaque couple. Chaque position est un point de croisement potentiel. On crée un masque de croisement de longueur égale à celle du génotype de façon aléatoire et la parité des bits dans le masque indique quel parent va donner quel bit aux enfants.

Parent 1 : 10110

Parent 2 : 00011

Masque : 00110

Enfant 1 : 00111

Enfant 2 : 10010

L'enfant 1 est produit en prenant le bit du parent 1 si le bit correspondant du masque vaut 1 et le bit du parent 2 si le bit correspondant du masque vaut 0. L'enfant 2 est créé en utilisant le masque inverse ou en échangeant les parents 1 et 2.

1.2.6. Mutation

L'opérateur de mutation est utile pour une bonne exploration de l'espace de recherche. En effet, il permet d'éviter le phénomène de *dérive génétique*, où certains gènes favorisés par le hasard peuvent se répandre au détriment des autres et être présents au même endroit sur tous les génotypes. Il permet aussi une meilleure recherche locale, car lorsque les individus ont convergé autour de l'optimum global, le croisement devient inefficace du fait que ces individus sont souvent identiques. La mutation donne alors une chance de s'approcher de l'optimum global. De nombreuses méthodes existent pour réaliser l'opérateur de mutation :

L'inversion

On choisit un bit au hasard dans une chaîne binaire et on l'inverse.

Individu : 01010

Individu muté : 011010

La permutation

On permute deux bits dans une chaîne binaire.

Individu : 0101 Individu muté : 0011



D'autres stratégies de mutations existent. Par exemple, on peut accentuer la mutation vers les individus les plus faibles tout en conservant les gènes des meilleurs individus; ou encore on peut paramétrer l'opérateur de mutation de telle sorte que celle-ci décroisse avec la convergence de la population.

1.2.7. Remplacement

L'opérateur de remplacement sert à maintenir la taille originelle de la population, en sélectionnant les enfants issus du croisement des parents sélectionnés et de la mutation pour créer la nouvelle population. Il existe plusieurs formes de remplacement, entre autres :

- *Le remplacement générationnel* : on remplace l'ensemble des parents par leurs enfants. Une autre version est le remplacement de j mauvais parents par j enfants.
- *L'élitisme* : on applique le remplacement générationnel mais les meilleurs individus de la génération courante sont sélectionnés et passent automatiquement dans la génération suivante. Il permet de toujours conserver la meilleure solution.

1.2.8. Arrêt

Une fois la nouvelle population créée, les séquences d'évaluation, de sélection, de croisement et de mutation, et enfin de remplacement recommencent successivement. Il

faut alors définir un critère d'arrêt. Si ce dernier n'est pas respecté, les séquences se répètent indéfiniment, faisant ainsi évoluer la population. Ce critère peut être un nombre maximal de générations, un temps maximal de calcul, un nombre de fois significatif où le meilleur individu n'a plus changé au cours des générations, ou encore un test sur le meilleur individu trouvé. Si aucune solution acceptable n'est trouvée, on relance la recherche.

En conclusion, l'algorithme génétique commence par une initialisation de la population (ensemble de solutions) de manière aléatoire. Chaque individu de la population représente une solution potentielle du problème. L'individu est représenté par un vecteur de symboles, dans la plupart des cas, il s'agit d'un vecteur binaire. Ces individus évoluent de génération en génération. Durant chaque génération, les individus sont évalués par une fonction d'évaluation (phénotype). Compte tenu de leur valeur de la fonction d'évaluation, les individus parents sont sélectionnés par un processus de sélection (déterministe et/ou stochastique), pour créer de nouveaux individus enfants à l'aide des opérateurs de croisement et de mutation. Une variante de mise en œuvre de l'algorithme génétique est de former la génération suivante en générant une nouvelle population à partir des individus enfants créés et des individus parents choisis, tout en s'assurant de maintenir la taille de la population constante. Le choix de conserver certains individus parents d'une génération courante dans la génération suivante vient du fait qu'il peut arriver qu'aucun des individus enfants créés ne soient trouvés meilleurs que les meilleurs des individus parents. Après plusieurs générations (ou itérations), l'algorithme converge vers le meilleur individu qui est considérée comme la solution optimale du problème.

CHAPITRE 2

FORMULATION DU PROBLEME

La réduction des coûts totaux du système par une planification efficace des investissements sur le réseau est l'une des problématiques courantes à laquelle font face les ingénieurs, les praticiens et les gestionnaires municipaux dans le but d'apporter une aide à la décision pour des interventions sur les conduites des réseaux d'eau potable. Plusieurs chercheurs se sont penchés depuis les trente dernières années sur ce sujet et ont proposé différents modèles d'optimisation pour minimiser les coûts du système. Les modèles présentés dans la section suivante ont utilisé l'algorithme génétique comme méthode de calcul pour optimiser les coûts totaux du système. Ces modèles sont pertinents dans la mesure où ils permettent la mise en place d'un système solide et fiable d'aide à la décision en matière de renouvellement des conduites des réseaux d'eau.

2.1. REVUE DE LA LITTÉRATURE

L'optimisation des coûts de décision dans les réseaux de distribution d'eau potable a fait l'objet de plusieurs travaux de recherche au cours des trente dernières années. Les méthodes de programmation linéaire, non linéaire, dynamique et les méthodes énumératives ont beaucoup servi comme base de calcul dans la recherche de la solution optimale. Cependant de nouvelles techniques, dites heuristiques, telles que le recuit simulé et l'algorithme génétique, ont été mises au point pour traiter des problèmes

combinatoires d'optimisation mettant en jeu une infinité de choix possibles sous une large variété de contraintes. Dans ce mémoire, on s'intéresse uniquement à l'algorithme génétique, et on a sélectionné ci-après quelques modèles intéressants.

Halhal et *al.* (1997) ont utilisé l'algorithme génétique pour trouver la meilleure façon d'investir une partie ou la totalité du capital disponible pour améliorer la performance du réseau. Cet algorithme est choisi car il est un outil efficace pour résoudre les problèmes mettant en jeu peu de variables de décision pour un grand nombre de possibilités. Le double objectif était de minimiser les coûts des solutions trouvées tout en maximisant les bénéfices résultant du choix de ces solutions. La minimisation des coûts consiste à rechercher les solutions peu coûteuses et satisfaisantes. La maximisation des bénéfices portent sur le manque à gagner en termes de performance hydraulique, d'état structural de la conduite, de conditions d'opération et de qualité de l'eau. Dans ce cas d'optimisation multicritère, il n'existe pas une solution unique comme dans le cas monocritère, mais un ensemble de solutions satisfaisantes connues sous le nom d'ensemble d'optimum de Pareto ; ce sont des solutions de bon compromis entre les coûts et les bénéfices, car on ne peut alors améliorer un critère (coûts) sans détériorer l'autre critère (bénéfices).

Dandy et *al.* (2001) se sont aussi servi de l'algorithme génétique pour minimiser les coûts totaux du système. Cet algorithme est choisi dans le but de montrer son efficacité à trouver la date de remplacement optimale pour les conduites du réseau d'eau. À l'aide d'un modèle de défaillance des conduites et d'une analyse des coûts de maintenance et de remplacement, la date optimale de remplacement est trouvée. Trois cas ont été ensuite

analysés : 1) le cas où les conduites doivent être remplacées ou non à l'instant présent (décision à court terme), 2) le cas où les conduites peuvent être remplacées ou non à des intervalles de périodes fixes sur un horizon de temps (décision à long terme), et enfin 3) le même cas précédent mais avec une possibilité de changer le diamètre des conduites. Dans ce dernier cas, un modèle hydraulique est nécessaire pour vérifier les performances hydrauliques du système. Une contrainte est imposée sur le budget et les variables de décision sont les conduites à remplacer.

Wu et *al.* (2001) ont utilisé l'algorithme génétique pour minimiser les coûts de dimensionnement et de réhabilitation dans le réseau. Cet algorithme est choisi dans le but d'appliquer l'optimisation à un réseau très large. Pour une configuration donnée du réseau, le but est de minimiser les coûts d'interventions à faire tout en s'assurant que les contraintes de performance du réseau sont respectées. Pour cela, Wu et al. (2001) ont développé un logiciel qui intègre un programme de balancement de réseau (version modifiée d'EPANET) et un modèle d'optimisation en C++ (algorithme génétique). La version d'EPANET modifiée vérifie, pour chaque combinaison d'interventions sur les conduites, si les contraintes hydrauliques sont satisfaites. Lorsque ces dernières ne sont pas satisfaites, un coût de pénalité est ajouté au coût du système. Ces travaux demandent une implémentation efficace du logiciel pour accélérer les transferts de données entre EPANET et le modèle d'optimisation. Les interventions de réhabilitation sont le nettoyage, le nettoyage et chemisage combinés, l'augmentation du diamètre et la duplication de la conduite. Toutes ces interventions peuvent entraîner un changement de diamètre. Ainsi, les variables de décision sont les diamètres des conduites en examen.

Wilson et *al.* (2003), de même, ont utilisé l'algorithme génétique pour minimiser les coûts d'intervention dans la rénovation du réseau hydraulique. Cet algorithme est choisi pour évaluer les plans de réhabilitation ou de remplacement des conduites du réseau d'eau de San Diego. Les variables de décision sont les conduites sur lesquelles intervenir, c'est-à-dire, réparer, réhabiliter, remplacer ou dupliquer. Pour chaque conduite, il y a donc une variété d'interventions possibles. L'algorithme génétique a permis de trouver, pour l'ensemble des conduites, la meilleure implémentation. Cet ensemble de solutions trouvées pour chaque conduite satisfait au mieux les conditions hydrauliques à un coût total minimum. Dans cette recherche, il n'y a pas de contraintes sur le budget et lorsque les conditions hydrauliques ne sont pas respectées, un coût de pénalité est affecté au coût total des solutions.

Plusieurs autres travaux intéressants (Afshar et *al.* 2005, Wu et Simpson 2001 et Wu et Simpson 2002) sur le sujet ont aussi fait l'objet d'étude. Dans tous les cas, on peut constater que l'analyse de coûts du système est une étape déterminante dans l'optimisation des interventions. Dans ce rapport, on s'est inspiré de l'analyse des coûts faite par Shamir et Howard (1979).

2.2. ANALYSE ÉCONOMIQUE DE SHAMIR ET HOWARD

Dans ce modèle, Shamir et Howard (1979) ne retiennent que deux façons de procéder pour intervenir sur la conduite : 1) le remplacement et 2) la réparation (ou maintenance).

Le but de cette analyse est de trouver la date optimale de remplacement d'une conduite.

On appellera cette date t^* . Pour cela, il faut procéder en trois étapes :

- l'analyse de l'historique de bris pour déterminer l'évolution du nombre de bris au cours du temps dans la conduite existante et dans la nouvelle conduite;
- l'analyse des coûts de réparation et de remplacement actualisés de la conduite depuis sa mise en œuvre en fonction du temps;
- la détermination de la date optimale de remplacement.

Chacune de ces étapes est détaillée dans les sections suivantes. La figure 2.1 montre la détermination de la date optimale à partir de l'analyse des coûts.

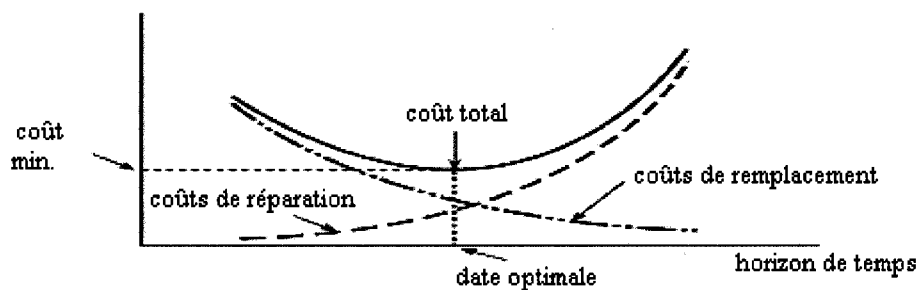


Figure 2.1 Analyse des coûts de remplacement et de réparation actualisés pour déterminer la date optimale de remplacement

2.2.1. Analyse de l'historique de bris

Les causes de bris sur la conduite sont diverses : le matériau et l'âge de la conduite, l'environnement (corrosivité du sol, gel, surcharge externe), la qualité de la main d'œuvre utilisée pour installer la conduite et les conditions de service (pression, coup de bélier). La

connaissance des causes de bris aide à la décision de sélection du matériau, revêtement, protection cathodique et des procédures de construction. Cependant, pour trouver la date optimale de remplacement, les causes des bris ne sont pas nécessaires, il faut juste déterminer l'évolution des bris dans le temps.

▪ **Dans la conduite existante**

Le taux de bris utilisé est le nombre de bris par unité de longueur et par année. L'équation du taux de bris utilisée par Shamir et Howard est d'ordre exponentiel. Soit $N(t)$, le taux de bris à l'année t par unité de longueur, t_0 l'année de pose de la conduite ou la première année où les données sont disponibles et A le taux annuel de croissance :

$$N(t) = N(t_0) \times e^{A(t-t_0)} \quad (\text{E1})$$

▪ **Dans la nouvelle conduite**

La nouvelle conduite va elle aussi développer un historique de bris. Si elle est homogène à la précédente, on s'attend à un même historique de bris que l'ancienne et par conséquent, la même équation de régression est applicable. Si la conduite est différente, le nombre de bris dans le futur va être donné par une autre équation similaire.

L'hypothèse de Shamir et Howard (1979) est de considérer que la nouvelle conduite n'aura pas de bris. En effet, les bris surviendront mais commenceront si loin dans le futur que la valeur présente du coût de réparation de bris dans la nouvelle conduite est négligeable.

2.2.2. Analyse des coûts

Pour effectuer cette analyse, on a besoin d'une prévision du nombre de bris dans le futur pour la conduite existante, du coût de réparation d'un bris C_b , du coût de remplacement de la conduite existante par la nouvelle conduite C_r , et du taux d'actualisation R .

On peut représenter graphiquement les trois courbes suivantes :

- la valeur actualisée des coûts projetés de réparation en fonction de la date de remplacement;
- la valeur actualisée des coûts de remplacement en fonction de la date de remplacement;
- la somme des deux valeurs précédentes.

Le coût total minimum donne la date optimale de remplacement (voir Figure 2.1).

Coûts de réparation

On pose l'hypothèse d'un coût de réparation pour un bris C_b constant au cours du temps.

Soit $N(t)$ le taux de défaillance de la conduite à l'année t , le coût d'une réparation par unité de longueur dans le futur pour l'année t est :

$$C_m(t) = C_b \times N(t) \quad (\text{E2})$$

Soit t_p l'année présente et R le taux d'actualisation, la valeur actualisée du coût de réparation est :

$$\frac{C_m(t)}{(1+R)^{-t_p}} \quad (\text{E3})$$

Soit t_r la date de remplacement de la conduite, le coût de réparation par unité de longueur jusqu'à la date t_r est donc

$$P_m(t_r) = \sum_{t=t_p}^{t_r} \frac{C_m(t)}{(1+R)^{t-t_p}} \quad (\text{E4})$$

Coûts de remplacement

Le coût d'un remplacement en valeur actualisée par unité de longueur est :

$$P_r(t_r) = \frac{C_r}{(1+R)^{t_r-t_p}} \quad (\text{E5})$$

C_r est le coût de remplacement en dollars constant. Lorsque P_r augmente t_r diminue.

Coûts totaux

Les coûts totaux sont la somme des coûts de réparation et de remplacement :

$$P_t(t_r) = P_m(t_r) + P_r(t_r) = \sum_{t=t_p}^{t_r} \frac{C_m(t)}{(1+R)^{t-t_p}} + \frac{C_r}{(1+R)^{t_r-t_p}} \quad (\text{E6})$$

2.2.3. Détermination de la date optimale

La date optimale est t_r qui minimise $P_r(t_r)$. La formulation du problème de Shamir et

Howard est la suivante :

$$\underset{t_r}{Min} \quad [P_r(t_r)] \quad (\text{E7})$$

On peut montrer que la fonction économique de l'équation E7 est équivalente à :

$$P_t(t_r) = \int_{t=tp}^{t=tr} \frac{C_m}{(1+R)^{t-tp}} - \frac{C_r \ln(1+R)}{(1+R)^{t-tp}} dt + K \quad (\text{voir Annexe 1})$$

La dérivée de $P_r(t_r)$ par rapport à t_r est $P'_r(t_r)$, telle que $P_r(t_r) = \int_{t=tp}^{t=tr} P'_r(t) dt + \text{Constante}$.

L'optimum de P_r est trouvé par l'équation $P'_r(t_r) = 0$, ce qui équivaut à :

$$C_m - C_r \ln(1+R) = 0 \quad (\text{E8})$$

La résolution de l'équation (E8) donne la date optimale théorique (voir Annexe 1) :

$$t = t_0 + \frac{1}{A} \ln \left(\frac{C_r \ln(1+R)}{N(t_0)C_b} \right) = t_{opt} \quad (\text{E9})$$

On peut aussi montrer que (voir Annexe 1) :

$$t_{opt} = t_p + \frac{1}{A} \ln \left(\frac{C_r \ln(1+R)}{N(t_p)C_b} \right) \quad (\text{E10})$$

Lorsqu'on égale les équations E9 et E10, nous pouvons conclure que la date optimale théorique de Shamir et Howard est donc indépendante de la date t . Il est important de préciser que la valeur t_{opt} trouvée n'est pas nécessairement supérieure à celle de t_p , on peut avoir tout aussi bien $t_{opt} \leq t_p$ que $t_{opt} > t_p$.

On peut aussi choisir un taux de défaillance linéaire $N(t) = N(t_0) \times A \times (t - t_0)$.

L'équation E3 donne $C_m(t) = C_b \times N(t_0) \times A \times (t - t_0)$.

L'optimum pour un taux de bris linéaire est trouvé par la résolution de l'équation E8 :

$$t_{opt} = t_0 + \frac{1}{A} \frac{\ln(1+R)C_r}{N(t_0)C_b} \quad (\text{E11})$$

En conclusion, la méthode d'analyse développée ci-dessus permet de calculer la date optimale de remplacement d'une conduite.

Plusieurs simulations vont être effectuées pour valider l'algorithme génétique. Pour cela, on va affecter différentes valeurs à chaque paramètre, et on va vérifier les résultats donnés par l'algorithme avec les résultats théoriques de l'analyse de Shamir et Howard (1979).

La section suivante porte sur la formulation du problème d'optimisation.

2.3. FORMULATION MATHÉMATIQUE DU PROBLÈME

La formulation du problème part directement de la démarche de Shamir et Howard (1979). On travaille avec un réseau constitué de N conduites. La date de départ de notre analyse est l'année présente $t_p = 2007$. Pour chaque conduite i du réseau de N conduites, nous allons déterminer la date optimale de remplacement de la conduite t_{ri} par l'algorithme génétique. Pour cela, des conditions initiales sont attribuées à chaque conduite i afin de pouvoir déterminer son taux de défaillance et par conséquent ses coûts de maintenance. Les conditions initiales sont : 1) l'année de pose de la conduite i , t_{0i} , 2) le taux annuel de défaillance A_i , 3) le coût d'une réparation de bris sur la conduite en dollars constant $C_{b,i}$, 4) le coût d'implémentation (remplacement ou réhabilitation) par 1000 pieds en dollars constant $C_{r,i}$, 5) le taux de défaillance de la conduite à l'année de pose t_{0i} , $N_i(t_{0i})$. Les coûts de maintenance de la conduite i sont obtenus par l'équation E3.

2.3.1. Hypothèses retenues

Les hypothèses que nous allons prendre permettent de tenir compte de la réalité des gestionnaires municipaux :

- La date optimale d'implémentation doit être supérieure ou égale à date présente de l'analyse. En effet, la date optimale théorique trouvée par Shamir et Howard (1979) peut être inférieure à la date présente t_p de l'analyse. On impose la condition suivante : quelle que soit la conduite i , $t_p \leq t_{ri}$.
- Le coût d'une réparation de bris C_b est constant sur la période T et ne dépend pas des conditions de la conduite ou du sol.
- Les coûts des interventions de réhabilitation sont inférieurs aux coûts de remplacement. À titre illustratif, le tableau 2.2 permet une comparaison des coûts des interventions (remplacement, réhabilitation structurale, réhabilitation non structurale).

Tableau 2.2 Coûts des interventions sur les conduites d'eau en dollars par mètre linéaire

Diamètre	Remplacement	Réhabilitation structurale	Réhabilitation non structurale
150	1085	420	190
200	1180	600	200
250	1275	-	210
300	1370	-	220

Source : Ville de Montréal (octobre 2007).

- Toutes les conduites du réseau ont une même longueur de 1000 pi (environ 300 m), quelle que soit la configuration du réseau.
- Les coûts de réparation de la nouvelle conduite sont nuls.

2.3.2. Premier modèle : sans contraintes budgétaires

Il s'agit de minimiser les coûts totaux du système sur une période T. On choisit une période T = 32 ans, car la date des interventions sur les conduites sera codée par un vecteur binaire à 5 bits (voir section 2.4). Par conséquent, il y a $2^5 (= 32)$ vecteurs ou dates possibles. Étant donné que chaque date représente une année possible d'intervention sur la conduite, l'horizon de temps qui couvre l'ensemble des possibilités est de 32 ans.

Le problème se formule de la manière suivante :

$$Min_{t_i} \left[\sum_i^N \left(\left(\sum_{t=t_p}^{t_i} \frac{C_{m,i}}{(1+R)^{t-t_p}} \right) + \frac{C_{r,i}}{(1+R)^{t_i-t_p}} \right) \right] \quad (E12)$$

Les variables de décision sont les dates de remplacement des conduites.

2.3.3. Deuxième modèle : avec contraintes budgétaires

Ce modèle est inspiré de l'analyse faite par Karaa et *al.* (1987). Une fois les dates optimales de remplacement des conduites données par le premier modèle, les conduites vont être classées en k groupe de conduites ayant les mêmes caractéristiques (p.e. la même date optimale t). Soit $\alpha_{k,t}$ la proportion de conduites du groupe k à remplacer à la date t et $\beta_{k,t}$ la proportion à réhabiliter à la date t, $C_m^k(t)$ le coût de maintenance à l'année t pour le groupe k, C_{rep}^k et C_{reh}^k les coûts respectifs de remplacement et de réhabilitation en dollars constants pour le groupe k, le problème se formule de la manière suivante :

$$\text{Min} \sum_t \left(\sum_k \frac{\alpha_{k,t} C_{rem}^k + \beta_{k,t} C_{reh}^k + \left(1 - \sum_{t' < t} (\alpha_{k,t'} + \beta_{k,t'}) \right) C_m^k(t)}{(1+R)^{t-t_p}} \right) \quad (\text{E13})$$

Les contraintes associées au problème sont les suivantes :

- Positivité : $\forall k, \forall t, \alpha_{k,t} \geq 0$ et $\beta_{k,t} \geq 0$.
- Sur l'horizon de temps T, on intervient sur toutes les conduites du réseau :

$$\forall k, \sum_t (\beta_{k,t} + \alpha_{k,t}) = 1.$$

Lorsqu'elles sont prises en compte, les contraintes budgétaires sont : soit sur le coût total du système, soit sur les budgets annuels admissibles. Deux scénarios vont être étudiés :

- **Scénario 1** : Contrainte sur le coût total

On fixe un coût total seuil C_{lim} en deçà duquel on ne veut pas descendre.

$$\sum_t \left(\sum_k \frac{\alpha_{k,t} C_{rem}^k + \beta_{k,t} C_{reh}^k + \left(1 - \sum_{t' < t} (\alpha_{k,t'} + \beta_{k,t'}) \right) C_m^k(t)}{(1+R)^{t-t_p}} \right) \geq C_{lim} \quad (\text{E14})$$

- **Scénario 2** : Contrainte sur les budgets annuels admissibles b_t pour l'année t

$$\forall t, \sum_k \alpha_{k,t} C_{rem}^k + \beta_{k,t} C_{reh}^k + \left(1 - \sum_{t' < t} (\alpha_{k,t'} + \beta_{k,t'}) \right) C_m^k(t) \leq b_t \quad (\text{E15})$$

Les variables de décision sont les proportions de conduites à remplacer et à réhabiliter par groupe et date d'intervention.

2.4. FORMULATION GÉNÉTIQUE DU PROBLÈME

La solution est l'ensemble des conduites avec chacune leur date optimale de remplacement. Elle est représentée par un vecteur binaire. Pour cela, on choisit un codage binaire standard. Chaque solution (*individu*) est constituée de N conduites (N *chromosomes*). Une conduite est caractérisée par K variables (K *gènes*). Dans ce cas, seule la date d'intervention est la variable de décision. Par conséquent, chaque *chromosome* a un unique *gène* formé par une série de bits (0 ou 1 en binaire) représentant la date de remplacement et l'espace de codage couvre l'horizon de temps fixé. Cette représentation est illustrée à la figure 2.2.

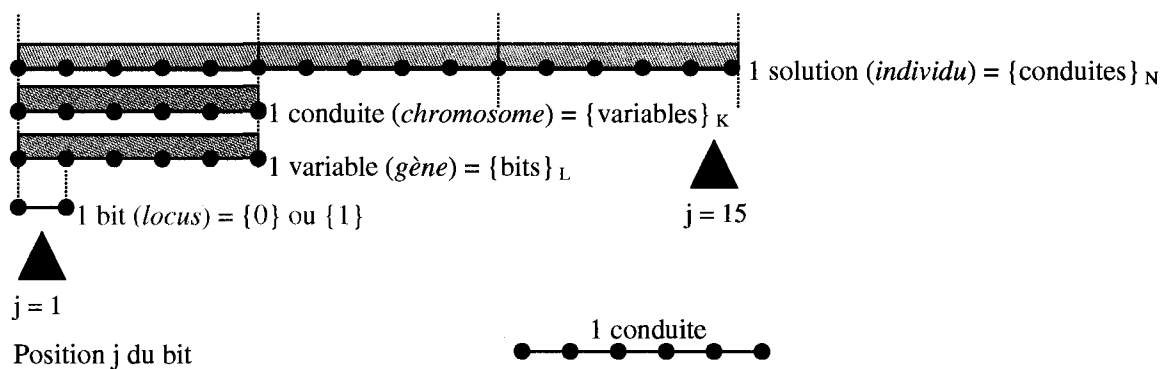


Figure 2.2 Décomposition de la représentation chromosomique

Dans la figure 2.2, la solution est un réseau de 3 conduites dont chacune a une variable de décision. Ainsi, la conduite et sa variable sont codées sur la même longueur de bits. Si la conduite avait deux variables de décision (par exemple, la date et le diamètre), la somme des longueurs de bits sur lesquelles les deux variables seront codées sera égale à la longueur de bits de la conduite.

Le nombre de bits N_{bit} de la variable de décision est la partie entière de $\frac{\log(N_{\text{val}})}{\log(b)}$, avec

N_{val} étant le nombre de valeurs possibles de la variable et b la base (en binaire $b = 2$).

Connaissant N_{bit} et N le nombre de conduites, il est alors possible de déterminer la taille

d'individu qui est de $N \times N_{\text{bit}}$. Soit un nombre binaire $a_N a_{N-1} \dots a_i \dots a_1 a_0$, la conversion en

nombre décimal vaut $\sum_{i=0}^N a_i \times 2^i$. Prenons un exemple de 3 conduites à remplacer sur une

période de 20 ans. Le nombre de bits pour coder la variable temporelle est la partie entière

de $\log(32)/\log(2)$, soit 5. Chaque conduite sera codée sur 5 bits. Une solution est donc

générée aléatoirement par un vecteur de taille $15 = 3$ (conduites) $\times$ 5 (bits). Soit le vecteur

d'une solution suivant :

01110	10101	10001	}	1 solution
conduite 1	conduite 2	conduite 3		

Lorsqu'on convertit les nombres binaires en nombres décimaux, on obtient les dates de

remplacement $t_1(\mathbf{01110}) = 14$, $t_2(\mathbf{10101}) = 21$ et $t_3(\mathbf{10001}) = 17$. Si la date de simulation

($t = 0$) est 2007, la conduite 1 doit être remplacée en 2021, la conduite 2 en 2028 et la

conduite 3 en 2024.

Le tableau de correspondance du lexique utilisé est le suivant :

Lexique génétique	Lexique mathématique	Lexique physique
Individu	Vecteur ligne de la matrice ou Solution	Réseau ou ensemble de conduites
Chromosome	Fraction du vecteur	Conduite
Gène	Fraction du vecteur	Date d'implémentation

La figure 2.3 montre le déroulement de l'algorithme.

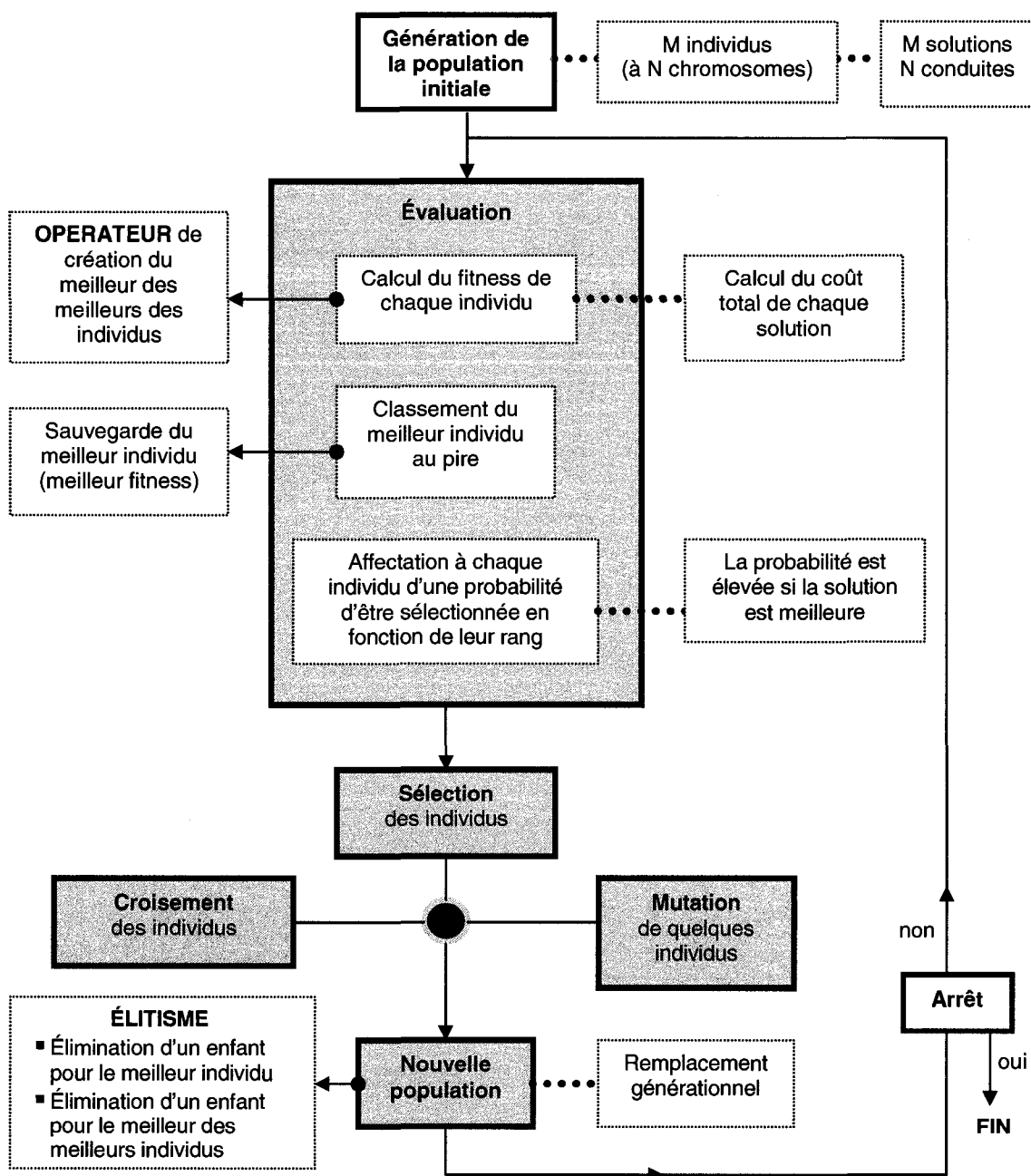


Figure 2.3 Déroulement de l'algorithme génétique

Tel qu'illustré à la figure 2.3, l'algorithme se déroule comme suit :

Génération de la population

On génère une matrice constituée de 0 et de 1, avec M lignes et $N \times N_{\text{bit}}$ colonnes. Cette matrice représente la population à M individus et N conduites. Chaque individu est une solution possible du problème qui symbolise le réseau à N conduites, chacune ayant leur date de remplacement.

Par exemple, pour un réseau à 3 conduites, on va générer une matrice 3×15 à 3 lignes (solutions/individus) et 15 colonnes (3 conduites).

conduite 1	conduite 2	conduite 3	
0 1 1 1 0	1 0 1 0 1	1 0 0 0 1	solution 1
0 0 1 1 0	0 0 1 0 1	0 0 0 0 0	solution 2
1 1 1 1 1	1 0 0 0 0	0 0 0 0 1	solution 3

Évaluation

Pour chaque ligne de la matrice, c'est-à-dire, chaque individu, on décode le temps optimal de chaque conduite.

14	21	17	solution 1
6	5	0	solution 2
31	16	1	solution 3

En ramenant à la date présente 2007 qui correspond à 0, on obtient :

2021	2028	2024	solution 1
2013	2012	2007	solution 2
2038	2023	2008	solution 3

Ensuite, on évalue le coût total du système pour chaque individu (c'est-à-dire solution) par l'équation E12, avec t_{ri} étant les dates générées aléatoirement pour chaque conduite i . On a donc :

$C_G^1 = 69\,623\$$	solution 1
$C_G^2 = 122\,390\$$	solution 2
$C_G^3 = 91\,822 \$$	solution 3

Une fois le travail précédent effectué pour chaque ligne (individu/solution), on classe les individus par rapport à leur coût total. Le rang de chaque individu (solution) sera utilisé pour calculer la probabilité de sélection de l'individu.

Opérateurs

On définit les opérateurs de sélection, de croisement, de mutation et de création du meilleur des meilleurs individus.

▪ Opérateur de sélection

La sélection se fait de façon aléatoire. On choisit la sélection par la roulette biaisée (voir section 1.2.4).

Dans notre exemple : $C_G^1 < C_G^3 < C_G^2$. On calcule les intervalles de probabilité Δ .

Individu	C_G	r	p_r	Δ
1	69 623	3	0,67	[0,33 ; 1]
2	122 390	1	0,00	[0 ; 0]
3	91 822	2	0,33	[0 ; 0,33]

Avec 3 solutions, on va appliquer l'opérateur de sélection 2 x 3 fois pour sélectionner 3 couples de parents qui donneront un enfant unique.

1 ^{ère} fois	2 ^{ème} fois	3 ^{ème} fois	4 ^{ème} fois	5 ^{ème} fois	6 ^{ème} fois
$\delta = 0.95$	$\delta = 0.23$	$\delta = 0.61$	$\delta = 0.48$	$\delta = 0.89$	$\delta = 0.76$
1	3	1	1	1	1
Couple 1		Couple 2		Couple 3	

▪ Opérateur de croisement

La probabilité de croisement p_c qui définit le pourcentage d'enfants à générer est égale à 1. Il y a donc autant de parents que d'enfants et la même taille de la population est ainsi conservée. Pour notre exemple, on a choisi un croisement en 2 points. Les points de coupure k_1 et k_2 sont des entiers générés aléatoirement entre 1 et 14 :

Couple 1	{	$k_1 = 1$ $k_2 = 12$ 0 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1	solution 1 solution 3
		Enfant 1	
		Enfant 2	
Couple 2	{	$k_1 = 7$ $k_2 = 10$ 0 1 1 1 0 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 1	solution 1 solution 1
		Enfant 1	
		Enfant 2	

▪ Opérateur de création du meilleur des meilleurs individus

Cet opérateur a été conçu pour accélérer la convergence de la solution. Il est parfaitement adapté au problème que l'on étudie. En effet, on travaille sur un problème de minimisation d'une somme de coûts. Pour minimiser la somme des coûts du réseau (ensemble des conduites), il faut en réalité minimiser les coûts de chaque conduite :

Min ($\sum_i \sum \text{coûts}_i$) est équivalent à $\sum_i \text{min} (\sum \text{coûts}_i)$

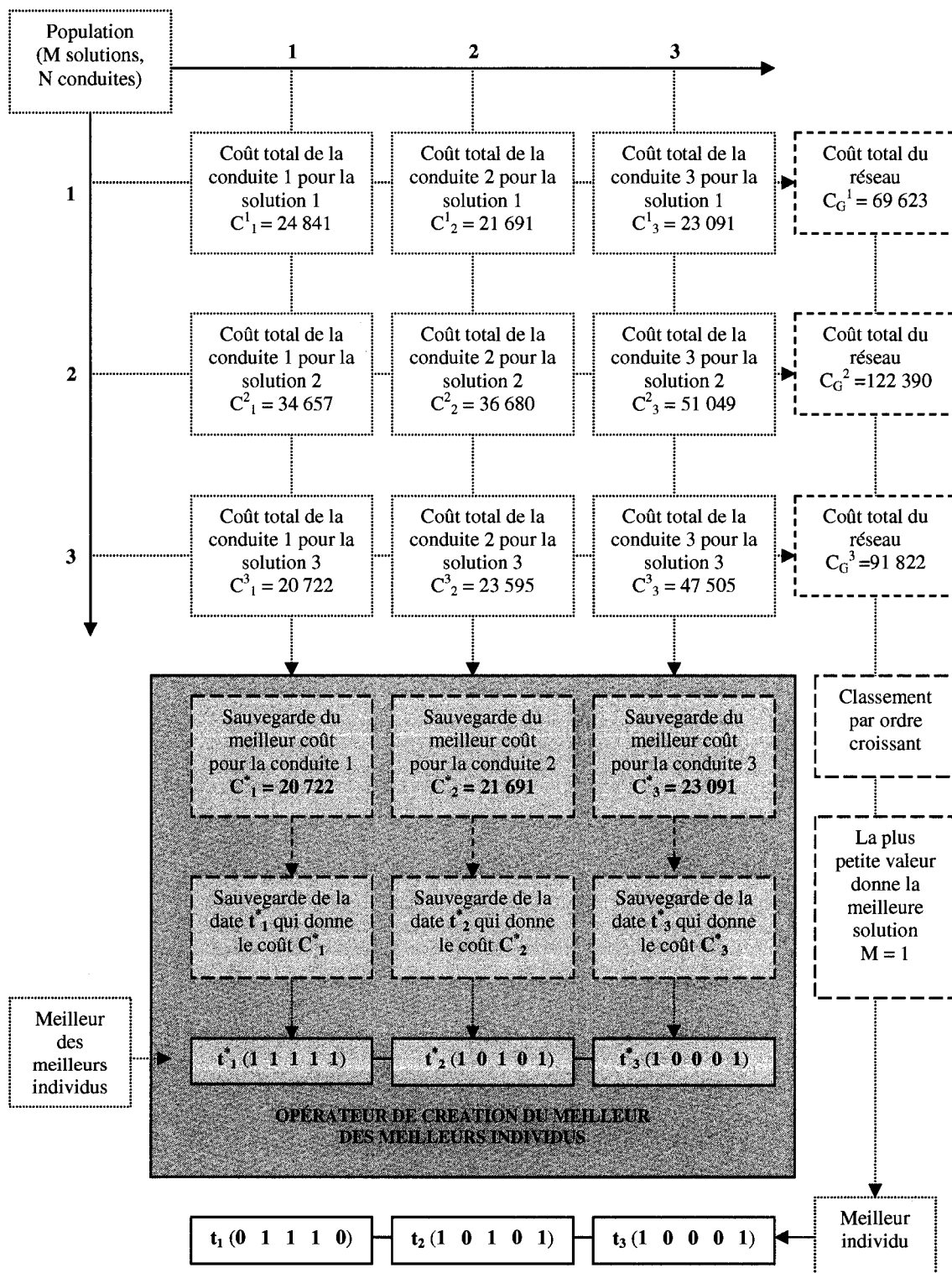
Pour chaque population, cet opérateur trouve le meilleur coût_i de chaque conduite i parmi les M individus. Ce meilleur coût_i correspond à une date de remplacement t_{ri}. Cet ensemble {t_{ri}}_{i=1, N} correspond au coût total minimum. De génération en génération, cet ensemble évolue en conservant toujours pour chaque conduite les dates des plus petits coûts trouvés depuis la première génération.

Prenons notre cas d'exemple du départ, toutes les 3 conduites ont les mêmes conditions initiales résumées dans le tableau 3.1.

0 1 1 1 0	1 0 1 0 1	1 0 0 0 1	solution 1
0 0 1 1 0	0 0 1 0 1	0 0 0 0 0	solution 2
1 1 1 1 1	1 0 0 0 0	0 0 0 0 1	solution 3

L'idée de concevoir cet opérateur provient de l'observation des premiers résultats où, pour des conditions initiales identiques de conduites, l'algorithme génétique convergait vers la solution optimale pour un très petit nombre de conduites (environ 5), et ne convergait plus vers la solution optimale pour un nombre plus élevé de conduites (environ 50).

Le schéma suivant explique le fonctionnement de cet opérateur.



Nouvelle population

La nouvelle population est créée en remplaçant les individus parents par les individus enfants créés. Ensuite, la sélection élitiste est appliquée en sauvegardant le meilleur individu parent de la génération courante pour le placer automatiquement dans la génération suivante afin de toujours conserver le meilleur individu. Pour cela, on élimine un des enfants créés, choisi au hasard, et on le remplace par le meilleur parent. La même procédure est faite pour remplacer le meilleur des meilleurs individus par la suppression d'un autre enfant créé.

Arrêt

L'algorithme s'arrête lorsqu'un nombre maximal de générations est atteint. La meilleure solution est celle qui a le plus petit coût total du système. En général, on choisit le nombre maximal de générations en fonction de la convergence vers la solution optimale trouvée. En effet, si celle-ci n'a plus évolué depuis un nombre considérable de générations, on peut considérer que le nombre maximal de générations choisi est acceptable.

2.5. METHODOLOGIE

Deux modèles vont être étudiés : le premier modèle sans contraintes budgétaires est inspiré de la démarche de Shamir et Howard (1979) et le deuxième modèle avec contraintes budgétaires est inspiré de l'approche de Karaa et *al.* (1987).

Le premier modèle sans contraintes budgétaires va servir à valider l'algorithme génétique. Dans ce cas, toutes les conduites auront les mêmes caractéristiques, à savoir les mêmes paramètres A , $N(t_0)$, t_0 , C_b , C_r et R . En faisant varier ces paramètres à tour de rôle, les dates optimales obtenues par l'algorithme génétique vont être comparées aux dates obtenues avec l'équation E11. Une fois l'algorithme validé, l'influence de l'opérateur de création du meilleur des meilleurs individus sur la solution obtenue va être examinée. De plus, une comparaison entre deux stratégies de croisement (uniforme et en 2-points) va être faite. Et enfin, l'influence des paramètres de l'algorithme génétique, tels que le nombre d'individus N , la taille de la population M et le nombre maximal de générations sur la convergence vers la solution optimale va être observée.

Dans le deuxième modèle, on impose des contraintes budgétaires. Dans ce cas, les conduites vont être réparties en cinq groupes de conduites ayant les mêmes caractéristiques. Deux scénarios de contraintes budgétaires vont être analysés. Dans le premier scénario, les contraintes sont appliquées au niveau des budgets annuels admissibles, et le remplacement est la seule intervention possible. Dans le deuxième scénario, les contraintes sont appliquées sur le coût total du système, et deux options d'interventions qui sont la réhabilitation et le remplacement vont être acceptées. Dans le premier tout comme dans le deuxième scénario, le premier modèle va être utilisé pour obtenir les dates optimales de remplacement sans contraintes budgétaires. Ainsi l'influence des contraintes budgétaires dans le deuxième modèle pourra être analysée à partir des résultats du premier modèle. Les dates optimales issues du deuxième modèle peuvent être inférieures, égales ou supérieures aux dates du premier modèle.

Dans le deuxième modèle, la conduite va également être codée sur 5 bits. On aurait pu choisir de coder la conduite par un vecteur de 6 bits à 2 gènes : le premier gène à 1 bit pour la méthode d'intervention (0 pour la réhabilitation et 1 pour le remplacement), et le deuxième gène à 5 bits pour la date d'intervention. Par exemple, 0 00110 veut dire que la conduite est réhabilitée en 2013. Cependant, afin d'éviter de changer la longueur de bits de la conduite et donc du vecteur solution, à chaque génération, pour évaluer la valeur d'adaptation de chaque solution, on génère aléatoirement un vecteur binaire de taille 2 fois N (nombre de conduites) qui constitue les choix d'interventions sur la conduite selon le décodage suivant :

00 : RÉHABILITATION	11 : RÉHABILITATION
01 : REMPLACEMENT	10 : REMPLACEMENT

Le choix de coder les interventions sur 2 bits plutôt que 1 provient du fait qu'on pourrait essayer différents coûts de réhabilitation et de remplacement. Toutefois, pour le premier modèle, on n'utilise pas cette nouvelle fonctionnalité.

Enfin, dans le deuxième modèle, la fonction d'optimisation reste le coût total du système; néanmoins, la sélection du meilleur individu dans la génération courante est faite à partir de la valeur d'adaptation telle que illustrée à la figure 4.8.

En conclusion, la formulation mathématique est une description analytique du problème à résoudre, à savoir, la minimisation des coûts totaux d'un réseau de conduites d'eau sur un horizon de temps fixé. La formulation génétique quant à elle montre la procédure de codage de la solution et la résolution du problème par l'algorithme génétique. Les deux prochains chapitres portent sur le premier et le deuxième modèle.

CHAPITRE 3

PREMIER MODÈLE

Il s'agit de valider la performance de l'algorithme génétique en comparant les dates optimales trouvées par l'algorithme génétique et les dates trouvées de manière analytique par l'analyse économique de Shamir et Howard (1979).

3.1. DONNÉES D'ENTRÉES DU MODÈLE

Le réseau est constitué de 50 conduites ayant toute la même longueur 1000 pi. On travaille sur un horizon de temps de 32 ans. Selon notre formule, le nombre de bits est la partie entière de $\log(32)/\log(2)$, soit 5 bits. Chaque conduite est codée par un vecteur à 5 bits, allant de 0 à $2^5 = 31$, avec 0 correspondant à 2007 et 31 correspondant à 2038. La valeur de la date de remplacement la plus petite est la date d'analyse 2007 et la valeur la plus tardive est 2038. Autrement dit, si la date optimale théorique est inférieure à 2007, l'algorithme donnera automatiquement 2007 et si la date optimale est supérieure à 2038, l'algorithme donnera 2038. Entre 2007 et 2038, la valeur de la date théorique doit correspondre à celle trouvée par l'algorithme.

On affecte les mêmes conditions initiales à toutes les conduites. Le tableau 3.1 résume la valeur de chaque paramètre d'entrée.

Tableau 3.1 Valeurs typiques des paramètres d'entrées

A	N (t_0)	C_r	C_b	R	t_0
0.05	0.1	50,000	1,000	0.1	1960

3.2. VALIDATION DE L'ALGORITHME GÉNÉTIQUE

La date de remplacement théorique est obtenue par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 1960 + 78 = 2038$.

Pour la simulation, les paramètres de l'algorithme génétique sont : N (le nombre de conduites), M (le nombre d'individus de la population) et N_{gen} (le nombre maximal de génération). Lorsqu'on lance les simulations, on obtient les résultats illustrés aux figures 3.1.

On peut constater que la date obtenu par l'algorithme génétique est la même que la date théorique. Et on observe que le coût total du système diminue jusqu'à se stabiliser sur la figure 3.1a.

On refait une autre simulation avec les valeurs du tableau 3.1, cette fois-ci avec $t_0 = 1950$. La figure 3.1b montre que la date théorique attendue (2028) est celle donnée par l'algorithme.

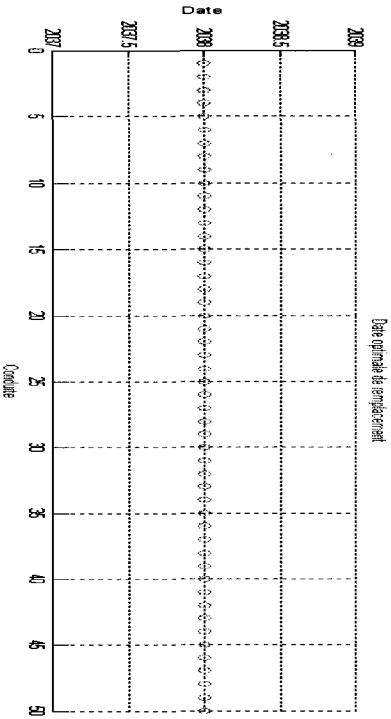


Figure 3.1a : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

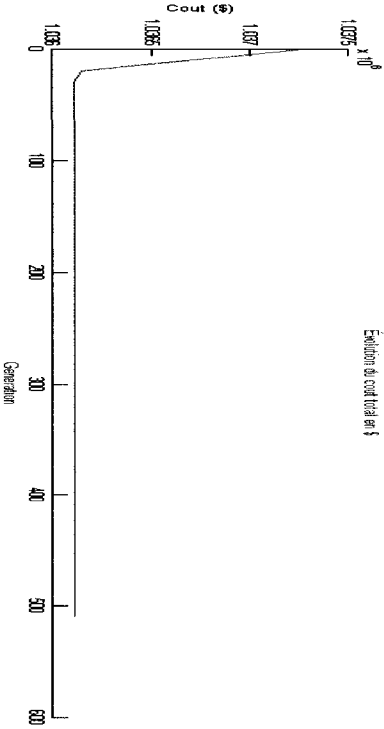


Figure 3.1a : Coût total du système en fonction des générations
($N = 50$, $M = 20$, $N_{gen} = 500$)

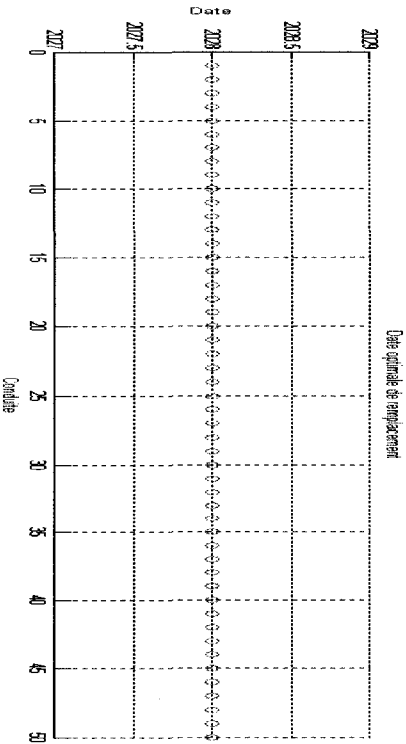


Figure 3.1b : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

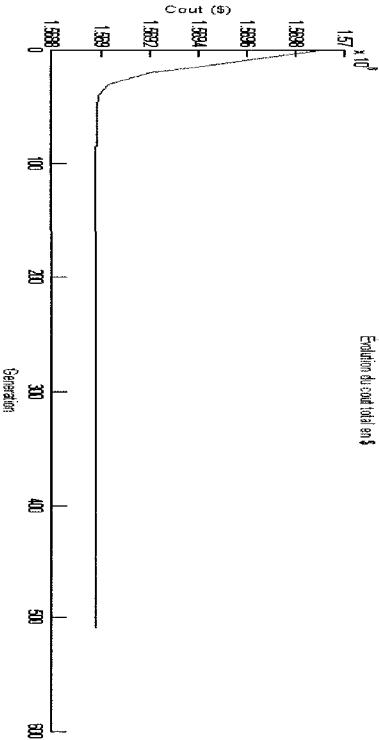


Figure 3.1b : Coût total du système en fonction des générations
($N = 50$, $M = 20$, $N_{gen} = 500$)

3.2.1. Influence du coefficient A

Les dates de remplacement théoriques sont obtenues par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 2037$ pour $A = 0.05064$ et $t_{\text{théorique}} = 2025$ pour $A = 0.06$. Les résultats de la simulation sont illustrés aux figures 3.2.

La figure 3.2a montre que la date diminue de 1 an lorsque A croît de 0.00064.

La figure 3.2b montre que la date diminue de 13 ans lorsque A croît de 0.01. Cependant, au bout de 500 générations, la valeur optimale est atteinte pour toutes les conduites, excepté la première. On relance la simulation jusqu'à 1000 générations et la valeur optimale est alors atteinte pour toutes les conduites sur la figure 3.2c. La figure 3.2d montre qu'en augmentant la taille de la population et en conservant le nombre de génération à 500, les conduites atteignent toutes leur valeur optimale.

3.2.2. Influence du coefficient R

Les dates de remplacement théoriques sont obtenues par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 2037$ pour $R = 0.095$ et $t_{\text{théorique}} = 2036$ pour $R = 0.09$. Les résultats de la simulation sont illustrés aux figures 3.3.

On voit sur la figure 3.3a que la date théorique a diminué de 1 an lorsque R a diminué de 0.5%, et sur la figure 3.3b que la date théorique a diminué de 2 ans lorsque R a diminué de 1%, ce que le modèle analytique prévoit.

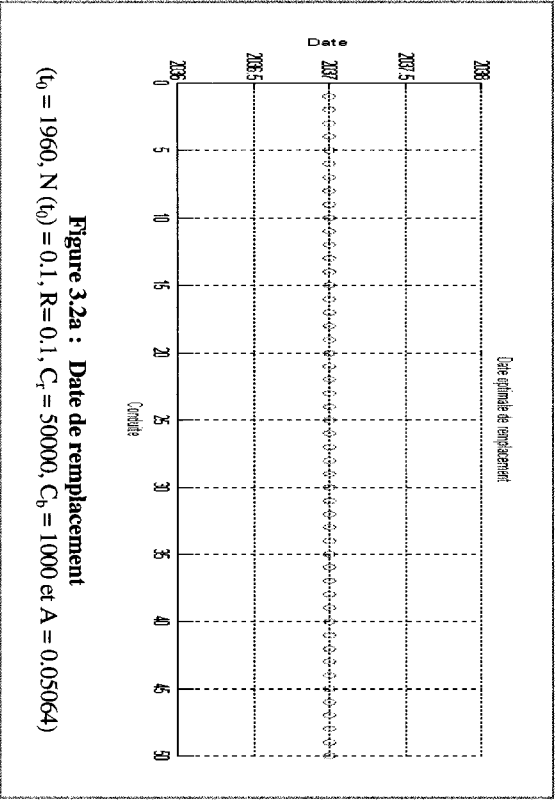


Figure 3.2a : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05064$)

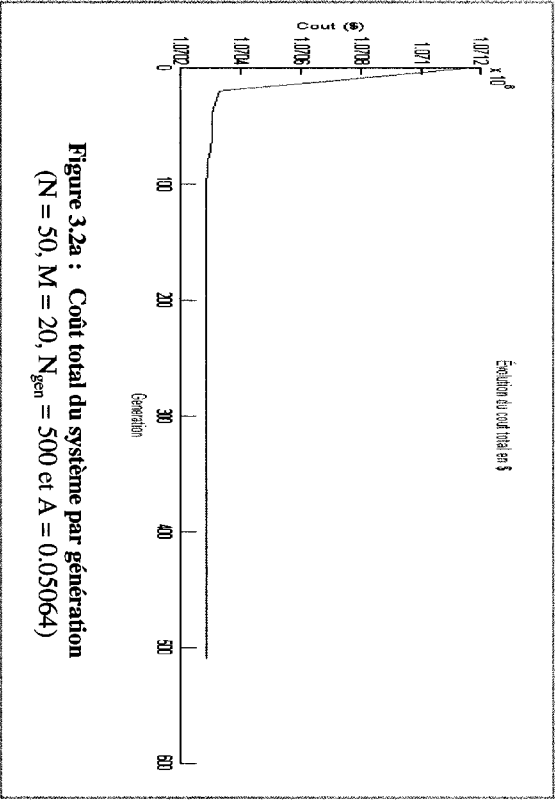


Figure 3.2a : Coût total du système par génération
($N = 50$, $M = 20$, $N_{gen} = 500$ et $A = 0.05064$)

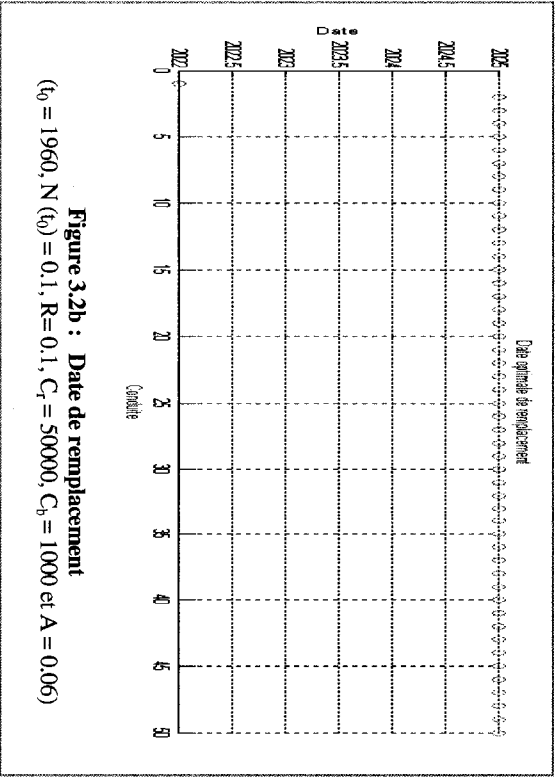


Figure 3.2b : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.06$)

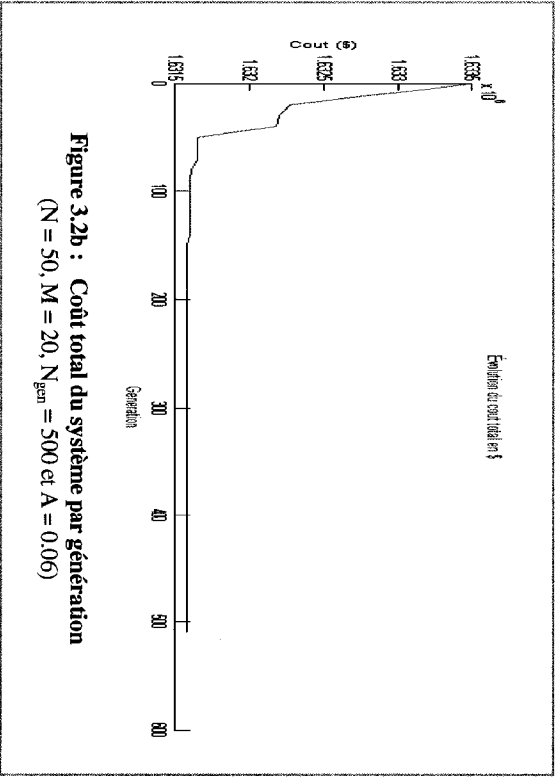
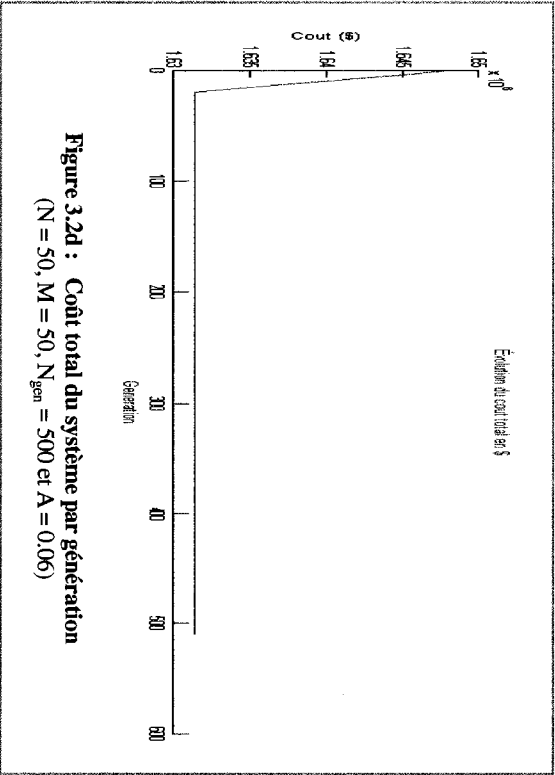
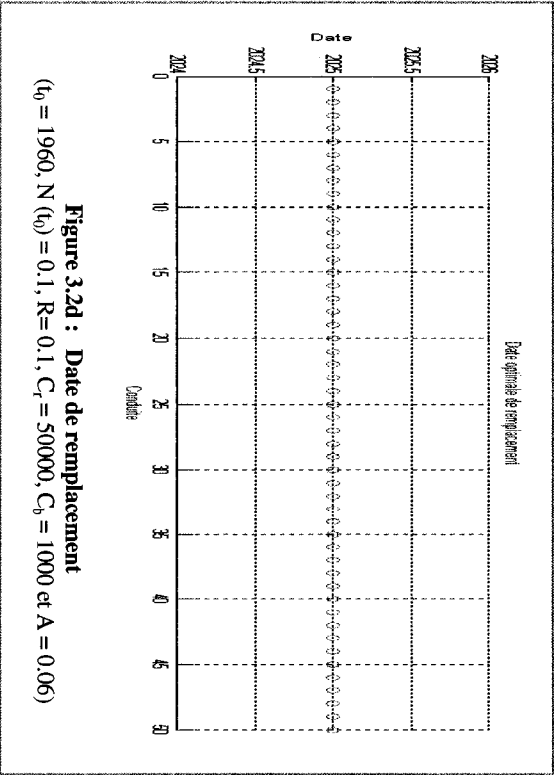
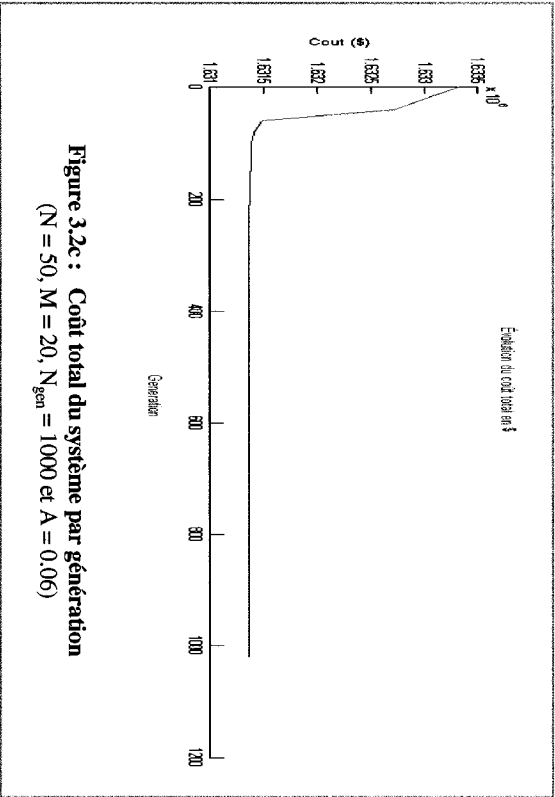
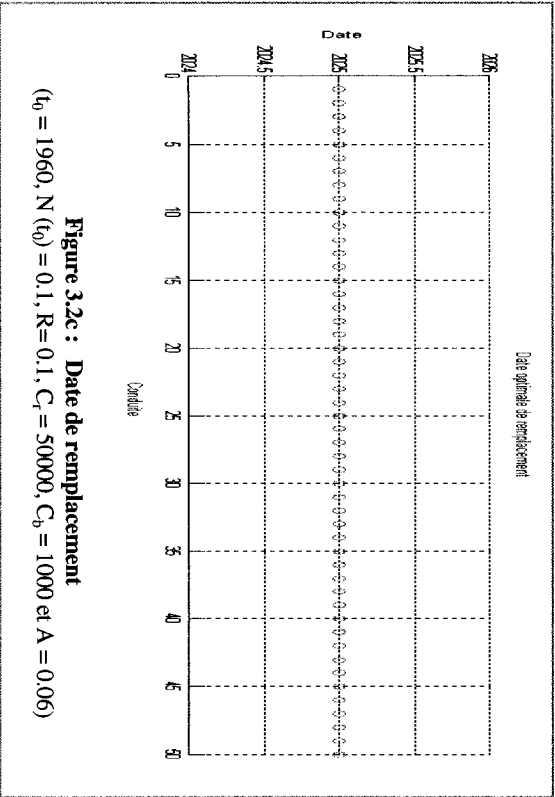
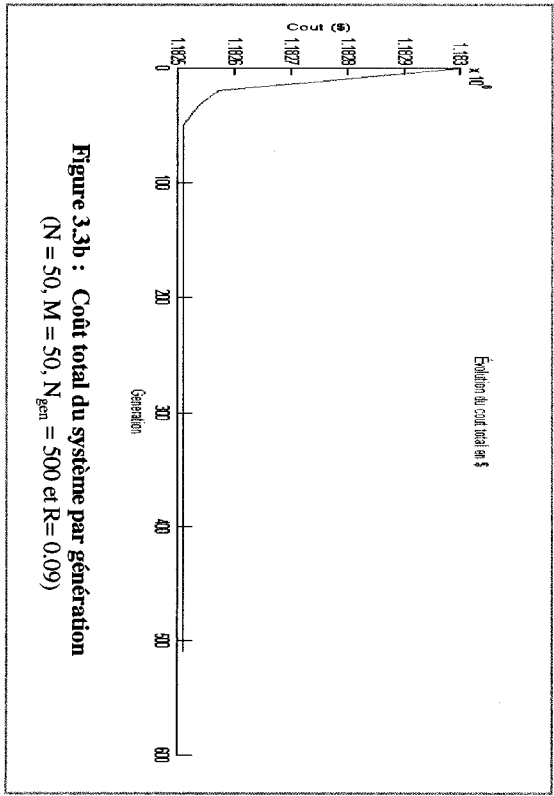
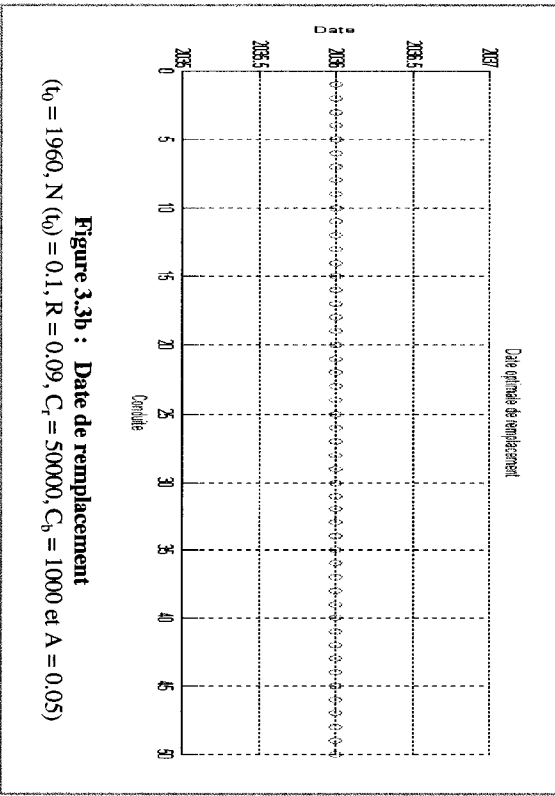
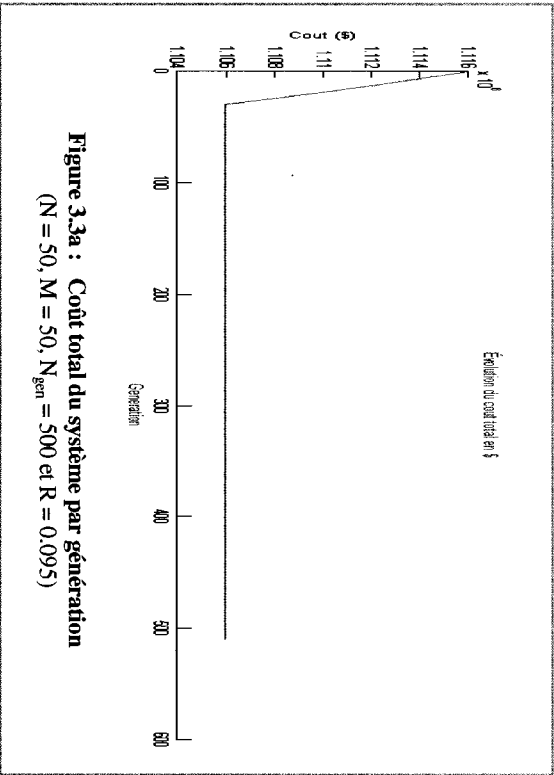
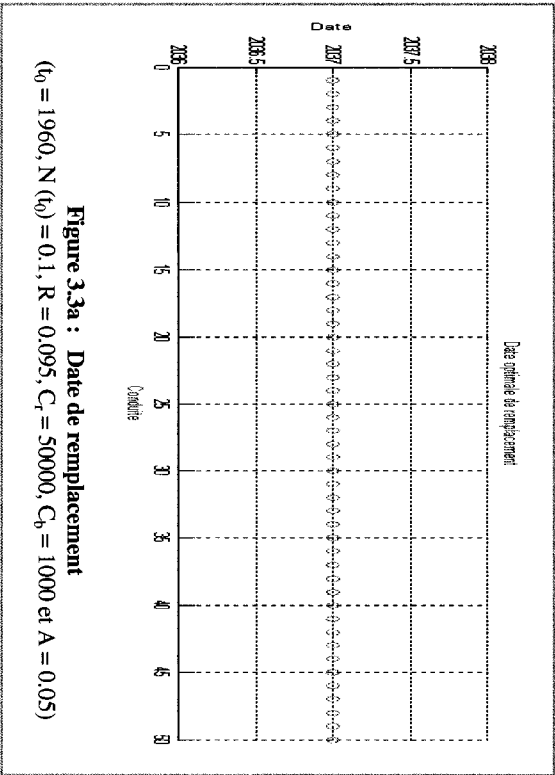


Figure 3.2b : Coût total du système par génération
($N = 50$, $M = 20$, $N_{gen} = 500$ et $A = 0.06$)





3.2.3. Influence du taux $N(t_0)$

Les dates de remplacement théoriques sont obtenues par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 2037$ pour $N(t_0) = 0.095$ et $t_{\text{théorique}} = 2033$ pour $N(t_0) = 0.09$.

Les résultats de la simulation sont illustrés aux figures 3.4.

On voit sur la figure 3.4a que la date théorique a diminué de 1 an lorsque $N(t_0)$ a augmenté de 0.005, et sur la figure 3.4b que la date théorique a diminué de 5 ans lorsque $N(t_0)$ a augmenté de 0.025, ce que le modèle analytique prévoit.

3.2.4. Influence du coût C_b

Les dates de remplacement théoriques sont obtenues par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 2037$ pour $C_b = 1050$ et $t_{\text{théorique}} = 2030$ pour $C_b = 1500$. Les résultats de la simulation sont illustrés aux figures 3.5.

La figure 3.5a montre que la date théorique a diminué de 1 an lorsque C_b a augmenté de 50. Cependant, la première conduite donne une date de 2036. En augmentant le nombre de génération à 700, comme c'est le cas sur la figure 3.5b, toutes les conduites ont une date de 2037 (résultat attendu). La figure 3.5c montre que, pour le cas où C_b augmente de 500, la date trouvée est de 2030. Ceci correspond aux valeurs théoriques.

3.2.5. Influence du coefficient C_r

Les dates de remplacement théorique sont obtenues par l'équation E9. Avec les valeurs du tableau 3.1, $t_{\text{théorique}} = 2037$ pour $C_r = 47500$. Les résultats de la simulation sont illustrés aux figures 3.6. Ils sont en accord avec l'analyse théorique.

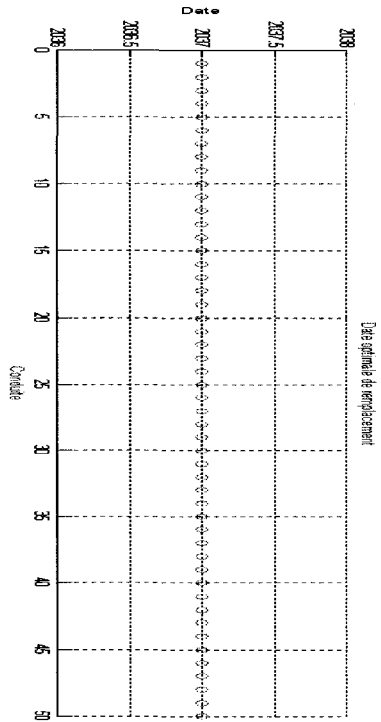


Figure 3.4a : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.105$, $R = 0.1$, $C_f = 50000$, $C_b = 1000$ et $A = 0.05$)

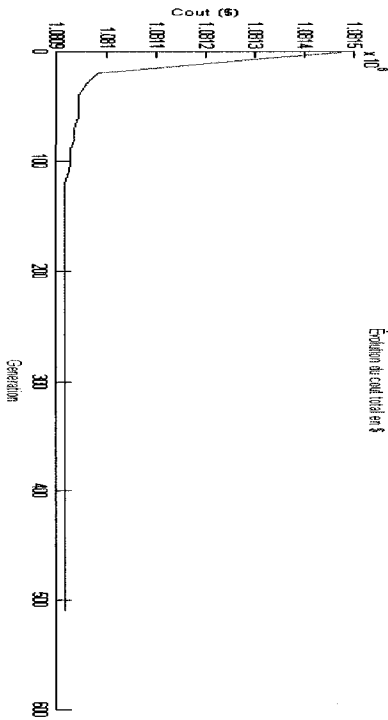


Figure 3.4a : Coût total du système par génération
($N = 50$, $M = 50$, $N_{gen} = 500$ et $N(t_0) = 0.105$)

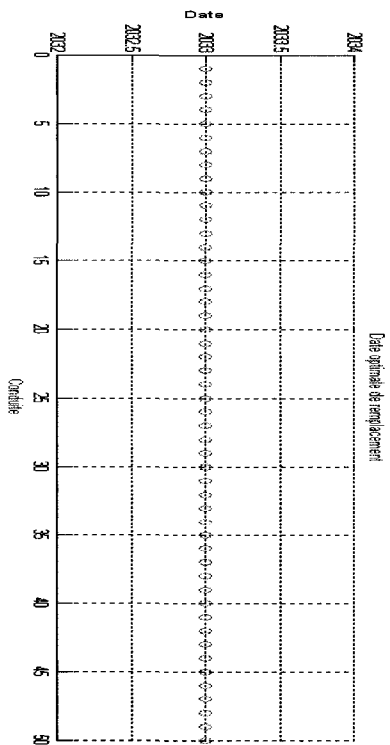


Figure 3.4b : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.125$, $R = 0.1$, $C_f = 50000$, $C_b = 1000$ et $A = 0.05$)

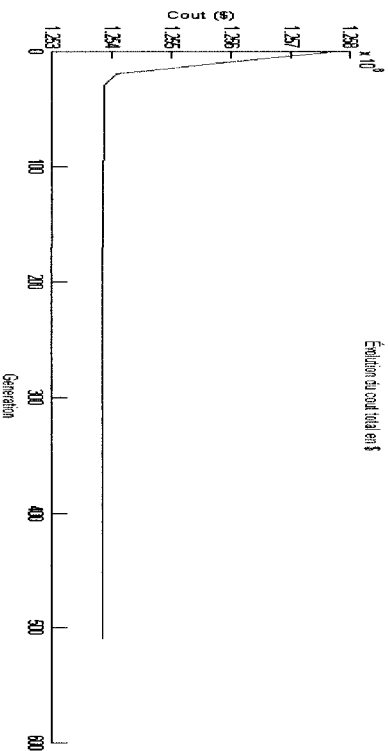


Figure 3.4b : Coût total du système par génération
($N = 50$, $M = 50$, $N_{gen} = 500$ et $N(t_0) = 0.125$)

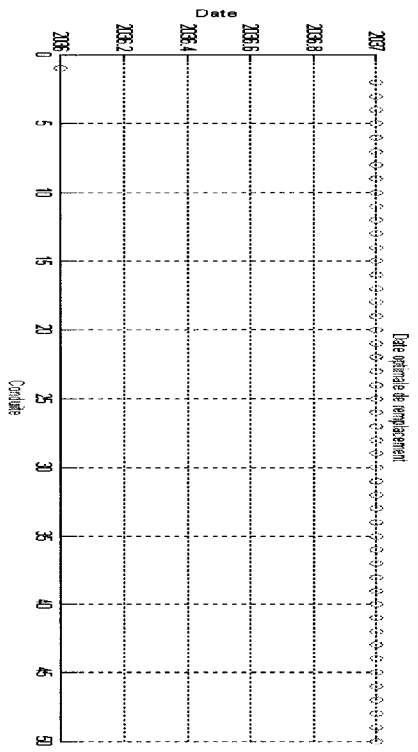


Figure 3.5a : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1050$ et $A = 0.05$)

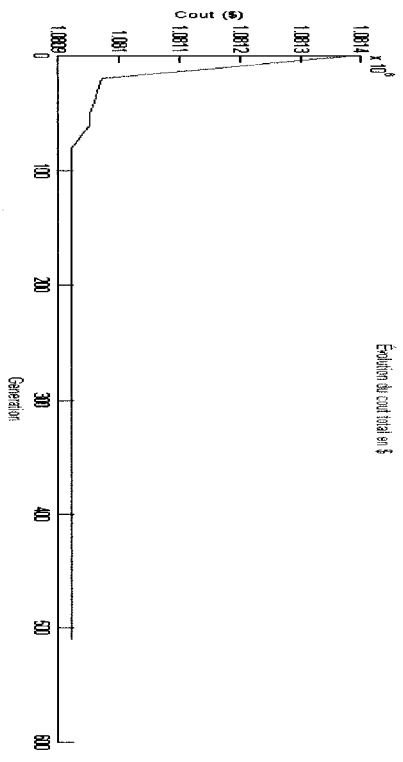


Figure 3.5a : Coût total du système par génération
($N = 50$, $M = 50$, $N_{gen} = 500$ et $C_b = 1050$)

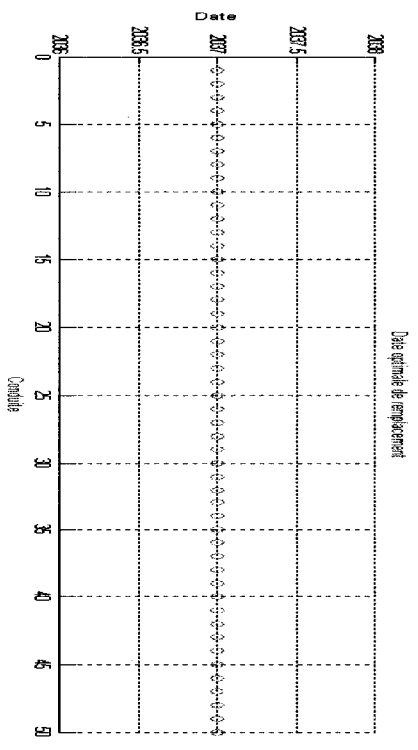


Figure 3.5b : Date de remplacement
($t_0 = 1960$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1050$ et $A = 0.05$)

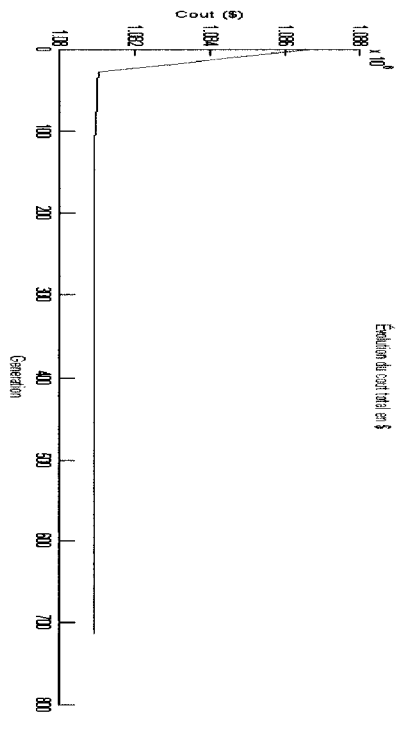
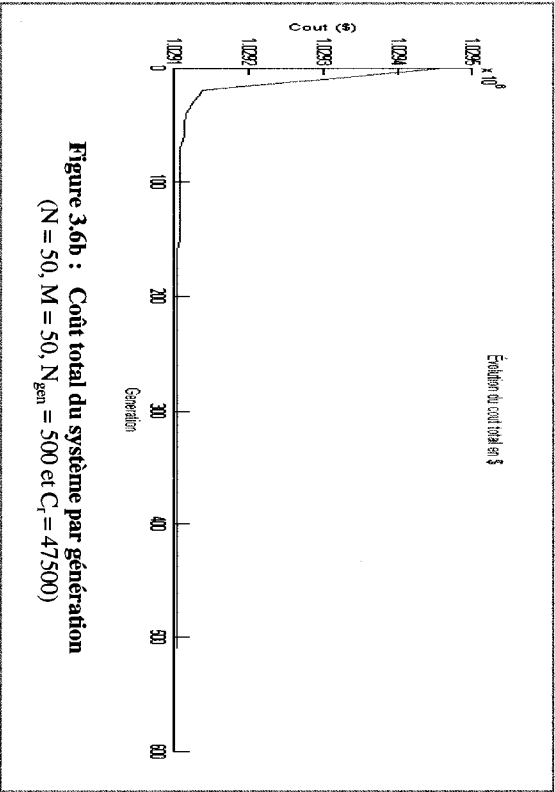
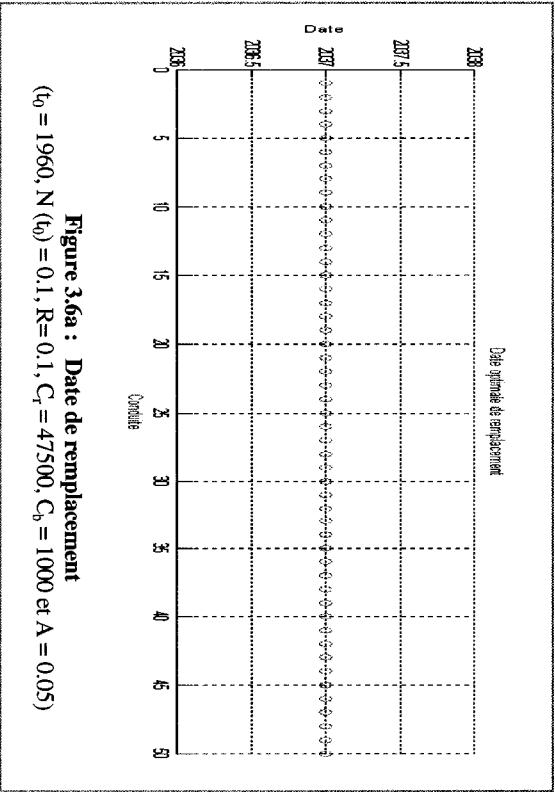
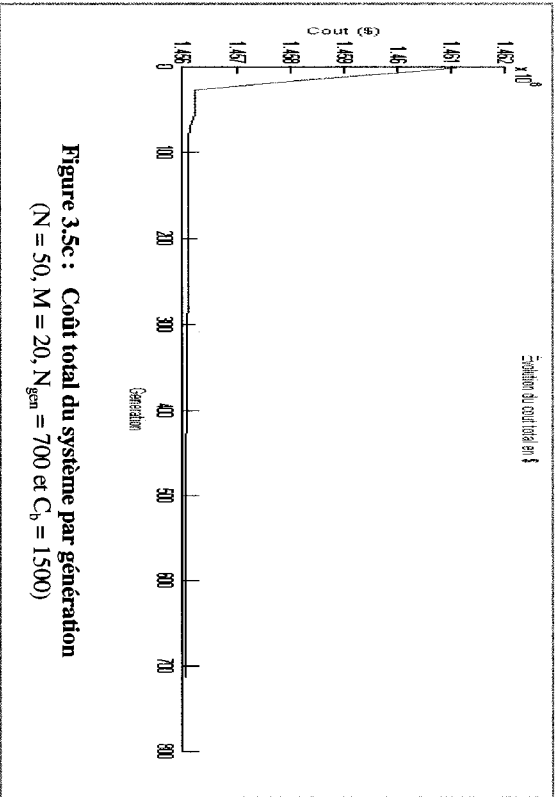
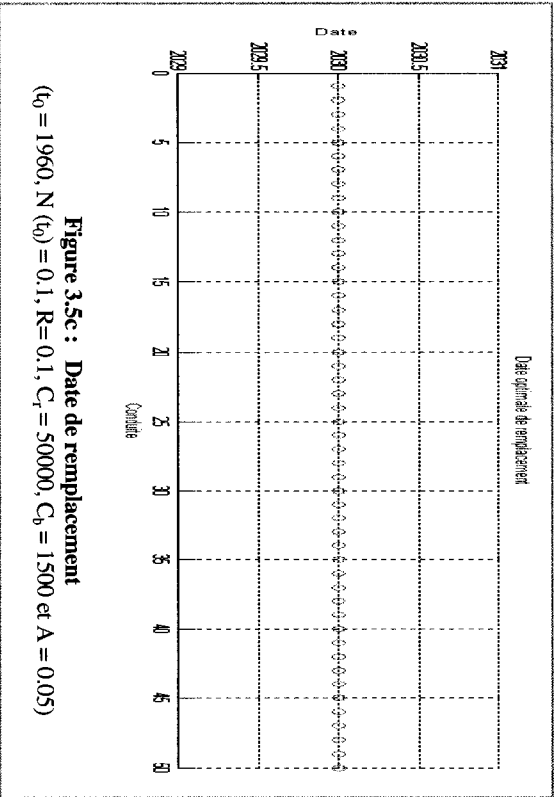


Figure 3.5b : Coût total du système par génération
($N = 50$, $M = 50$, $N_{gen} = 700$ et $C_b = 1050$)



3.2.6. Choix d'un taux de bris linéaire

Si le taux de bris est linéaire, alors la date optimale est donnée par l'équation E11.

Le tableau 3.2 résume la valeur de chaque paramètre d'entrée.

Tableau 3.2 Valeurs des paramètres d'entrée pour un taux de bris linéaire

Paramètres	A	$N(t_0)$	C_r	C_b	R	t_0
Essai 1	0.15	0.2	40,000	2,000	0.1	1960
Essai 2	0.15	0.2	10,000	1,000	0.1	1980

Avec les valeurs de paramètres du tableau 3.2, la date de remplacement théorique est égale à 2024 pour l'essai 1 et 2012 pour l'essai 2. Les résultats de la simulation sont illustrés aux figures 3.7a et 3.7b. Sur la figure 3.7a, on peut constater que la date optimale simulée est égale à 2026, et sur la figure 3.7b la date optimale simulée est égale à 2013.

On observe que dans le cas d'un taux de bris linéaire, l'algorithme ne converge pas vers la solution optimale du problème même après plusieurs autres simulations. On conclut que l'algorithme ne garantit pas l'atteinte de l'optimum.

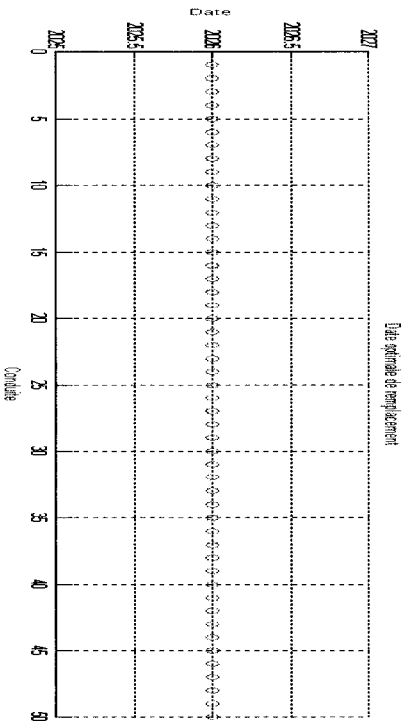


Figure 3.7a : Date de remplacement pour un modèle de taux de bris linéaire ($t_0 = 1960$, $N(t_0) = 0.2$, $R = 0.1$, $C_i = 40000$, $C_b = 2000$ et $A = 0.15$)

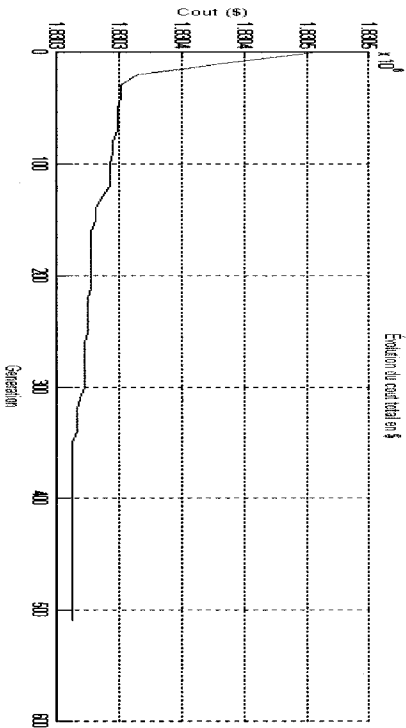


Figure 3.7a : Coût total du système par génération ($N = 50$, $M = 20$, $N_{gen} = 500$)

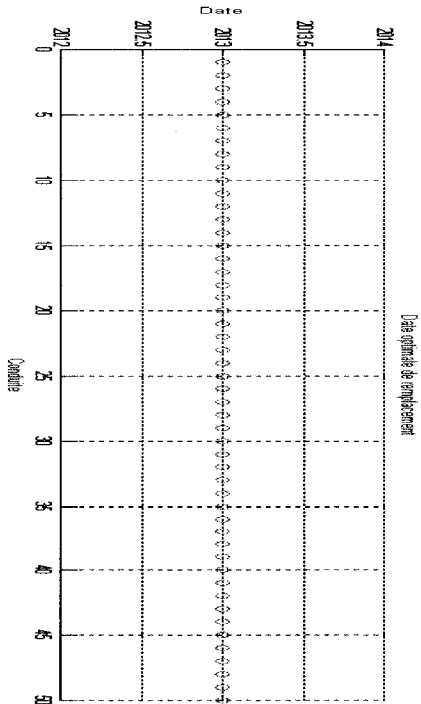


Figure 3.7b : Date de remplacement pour un modèle de taux de bris linéaire ($t_0 = 1980$, $N(t_0) = 0.2$, $R = 0.1$, $C_i = 10000$, $C_b = 1000$ et $A = 0.15$)

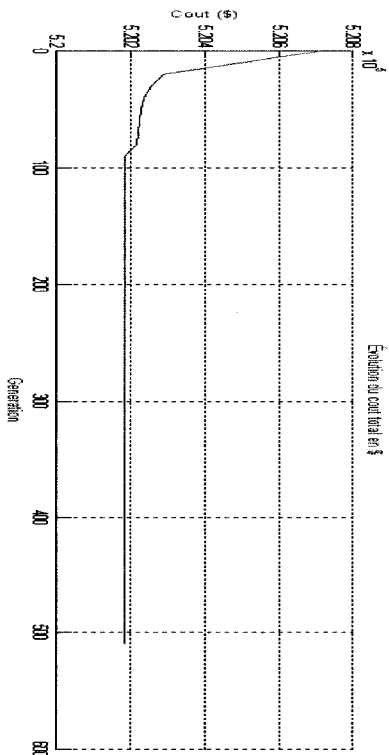


Figure 3.7b : Coût total du système par génération ($N = 50$, $M = 20$, $N_{gen} = 500$)

3.3. INFLUENCE DES OPÉRATEURS ET PARAMÈTRES DE L'ALGORITHME

On va étudier les effets de l'opérateur de création du meilleur des meilleurs individus et des différents opérateurs de croisement (uniforme et 2-points). Le taux de bris est donné par l'équation E2. Pour toutes les simulations, la date théorique optimale est 2028.

3.3.1. Opérateur de création du meilleur des meilleurs individus

Cet opérateur permet de faire converger la solution plus rapidement, en diminuant la dispersion de la solution autour de la valeur attendue. Un cas illustratif de l'apport de cet opérateur est démontré. Pour cela, deux simulations vont être faites, l'une avec l'opérateur et l'autre sans l'opérateur, en partant de la même matrice de population. Les résultats de la simulation sont illustrés aux figures 3.8.

Les résultats des simulations montrent que l'opérateur de création du meilleur des meilleurs individus permet d'accélérer la convergence vers la solution optimale. Les figures 3.8a et 3.8b montrent que l'atteinte de l'optimum n'est pas garantie sans opérateur. Les figures 3.8c et 3.8d montrent que l'augmentation du nombre de générations permet de faire tendre le coût minimal simulé vers le coût minimal théorique. Cependant sur la figure 3.8d, les conduites ont des dates oscillantes autour de la valeur optimale théorique lorsque l'opérateur est absent; et elles atteignent les dates optimales théoriques plus rapidement lorsque l'opérateur est présent.

* dates simulées (avec opérateur) o dates simulées (sans opérateur) * coût minimal théorique

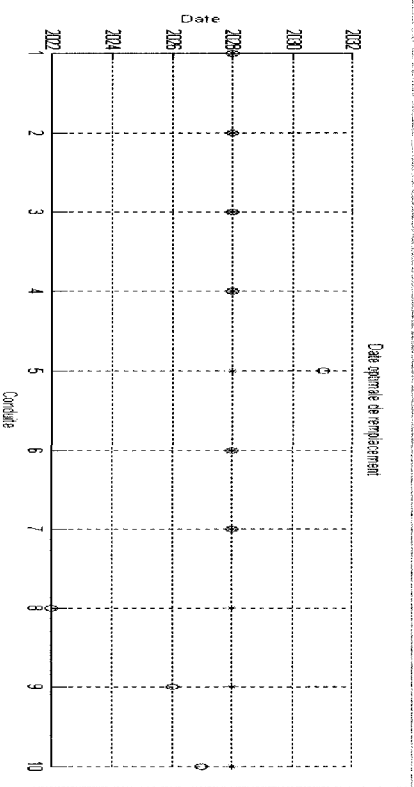


Figure 3.8a : Date de remplacement
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

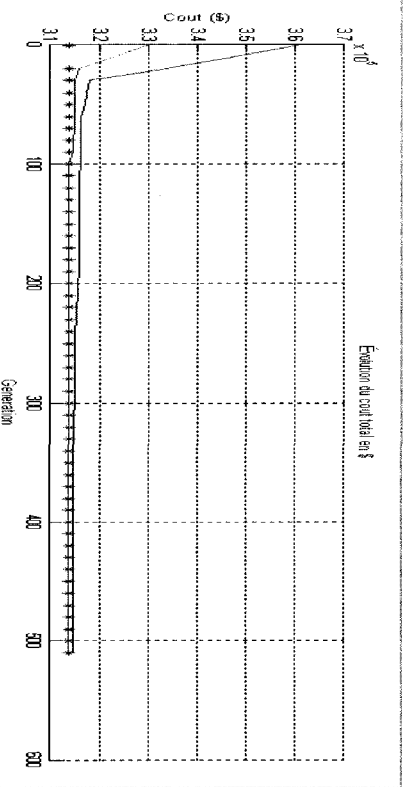


Figure 3.8a : Coût total du système par génération
($N = 10$, $M = 5$, $N_{gen} = 500$)

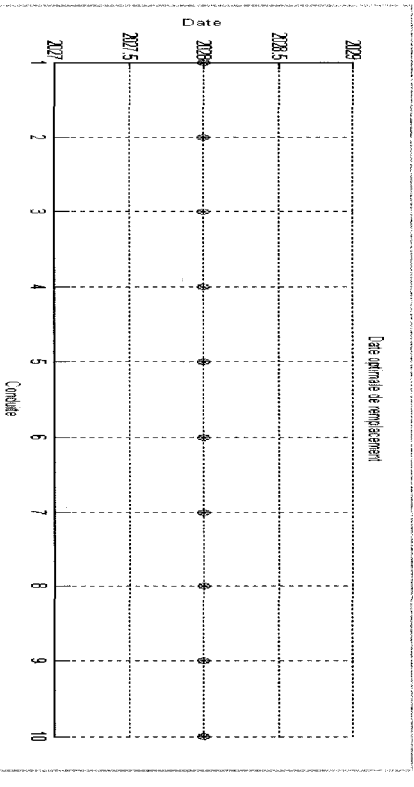


Figure 3.8b : Date de remplacement
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

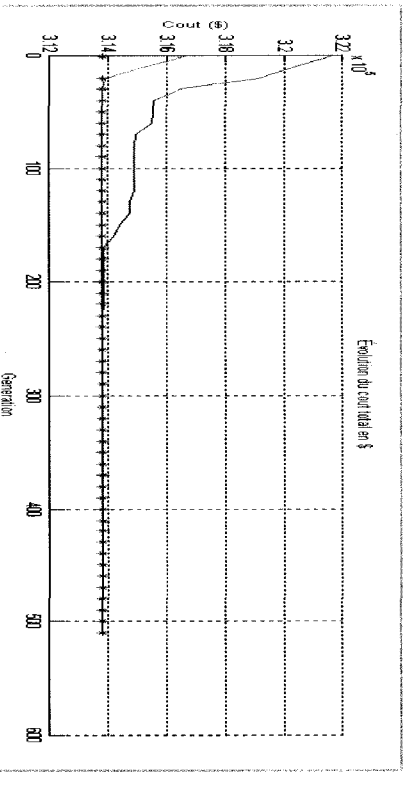


Figure 3.8b : Coût total du système par génération
($N = 10$, $M = 5$, $N_{gen} = 500$)

* dates simulées (avec opérateur) o dates simulées (sans opérateur) * coût minimal théorique

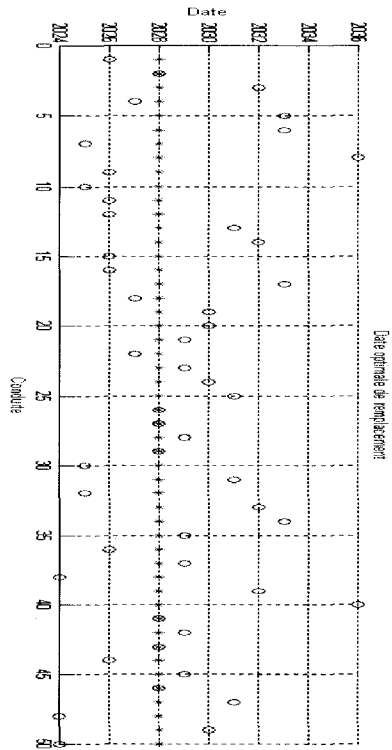


Figure 3.8c : Date de remplacement
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

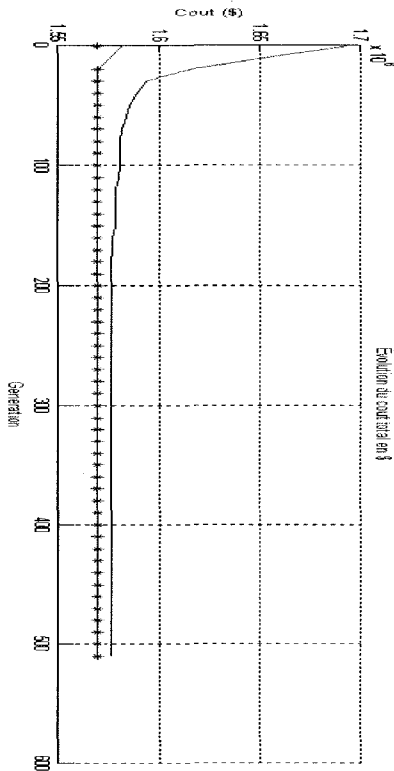


Figure 3.8c : Coût total du système par génération
($N = 50$, $M = 20$, $N_{\text{gen}} = 500$)

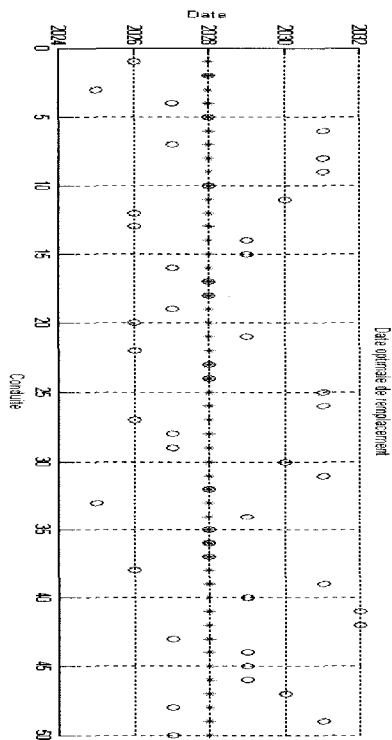


Figure 3.8d : Date de remplacement
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

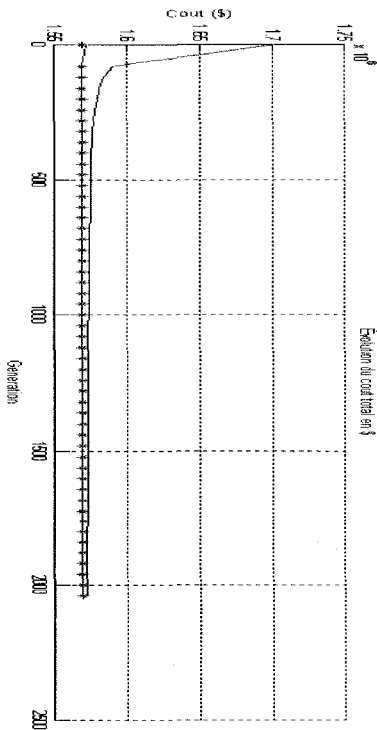


Figure 3.8d : Coût total du système par génération
($N = 50$, $M = 20$, $N_{\text{gen}} = 2000$)

3.3.2. Opérateurs de croisement

Deux types de croisement ont été testés : le croisement 2-points et le croisement uniforme. Les figures 3.9 à 3.11 montrent qu'il n'y a pas une stratégie de croisement meilleure qu'une autre en terme de convergence.

3.3.3. Influence du nombre d'individus M

Les résultats des simulations sont présentés aux figures 3.9. On constate que l'augmentation de la valeur de M fait converger la solution vers l'optimum pour un même nombre maximal de génération N_{gen} .

3.3.4. Influence du nombre de conduites N

Les résultats des simulations sont présentés aux figures 3.10. La valeur de N n'influence pas la convergence vers l'optimum. En effet, l'augmentation de N n'a pas perturbé l'atteinte de l'optimum pour une même valeur de M et N_{gen} .

3.3.5. Influence du nombre maximal de génération N_{gen}

Les résultats des simulations sont présentés aux figures 3.11. On constate que lorsque N_{gen} augmente, il y a plus de chance d'atteindre la solution optimale du problème.

En somme, la convergence vers l'optimum va être plus efficace en augmentant soit le nombre d'individus M, soit le nombre maximal de génération N_{gen} .

* dates simulées (croisement 2-points) o dates simulées (croisement uniforme) * coût minimal théorique

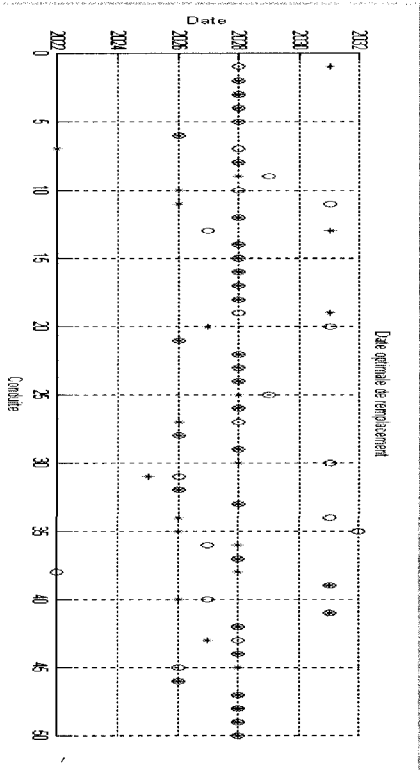


Figure 3.9a : Influence du nombre d'individus M
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

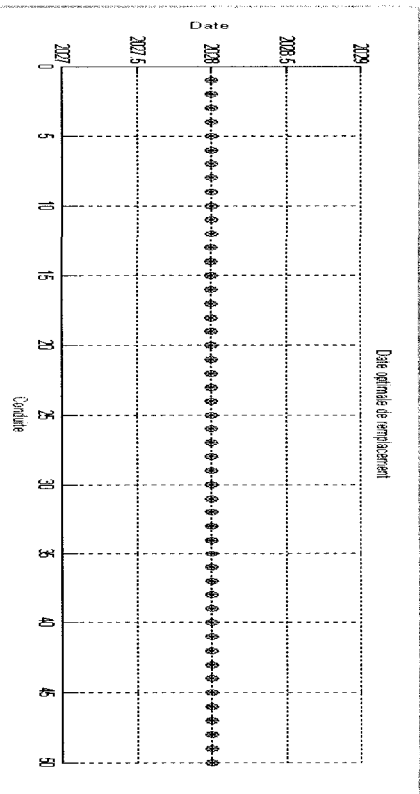


Figure 3.9b : Influence du nombre d'individus M
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

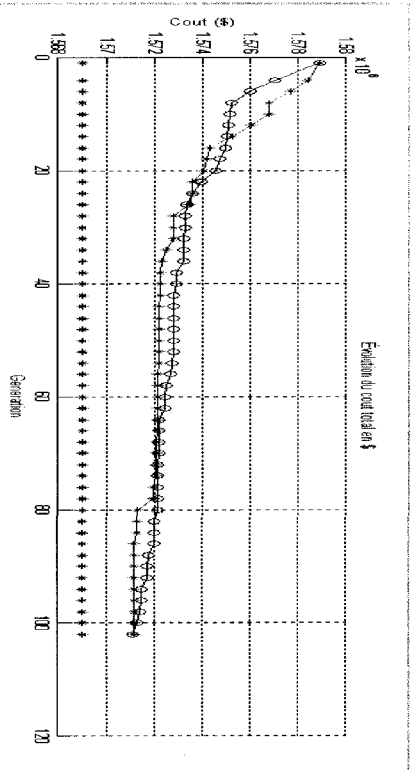


Figure 3.9a : Coût total du système par génération
($N = 50$, $M = 5$, $N_{gen} = 100$)

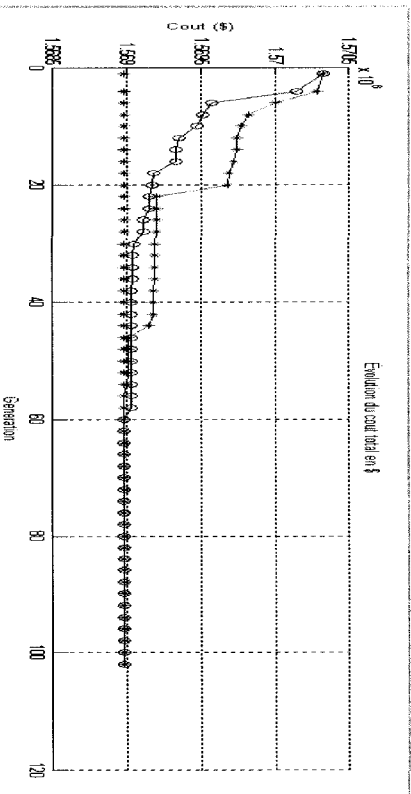
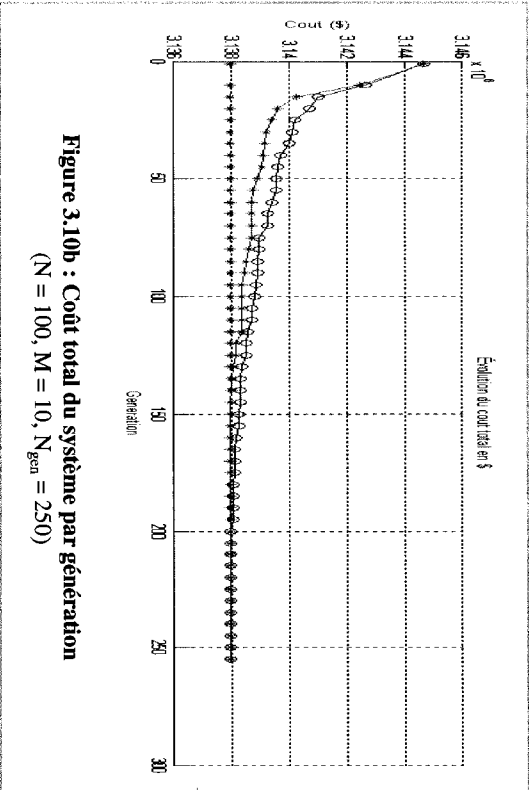
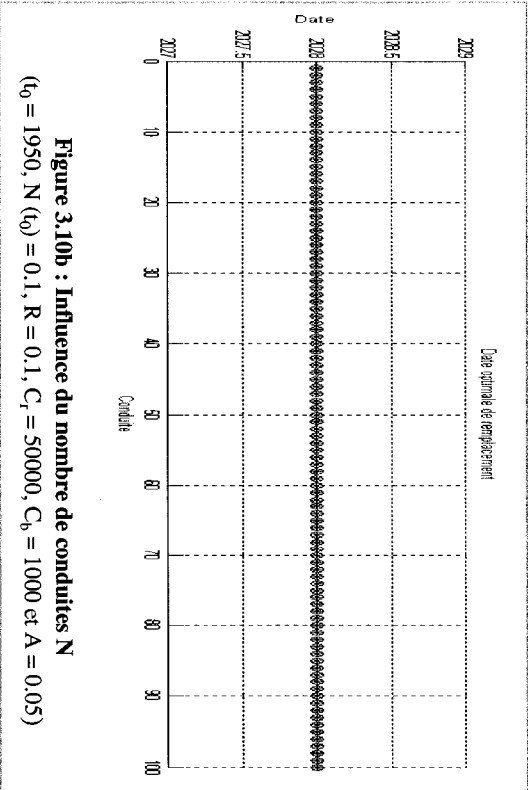
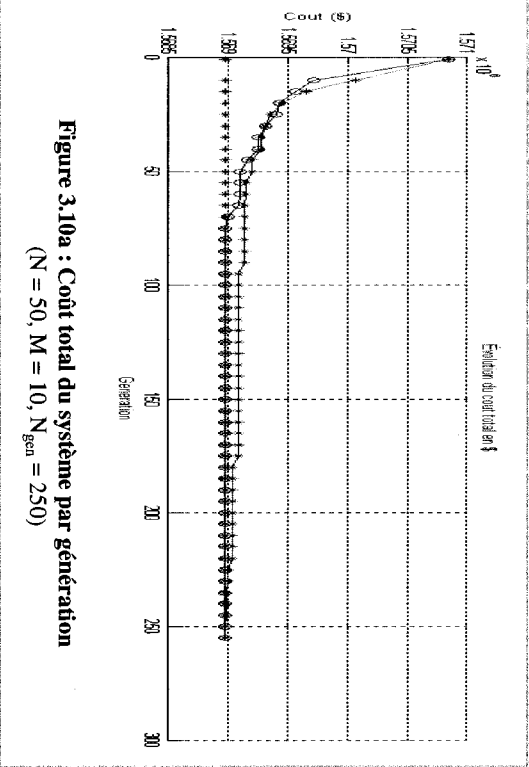
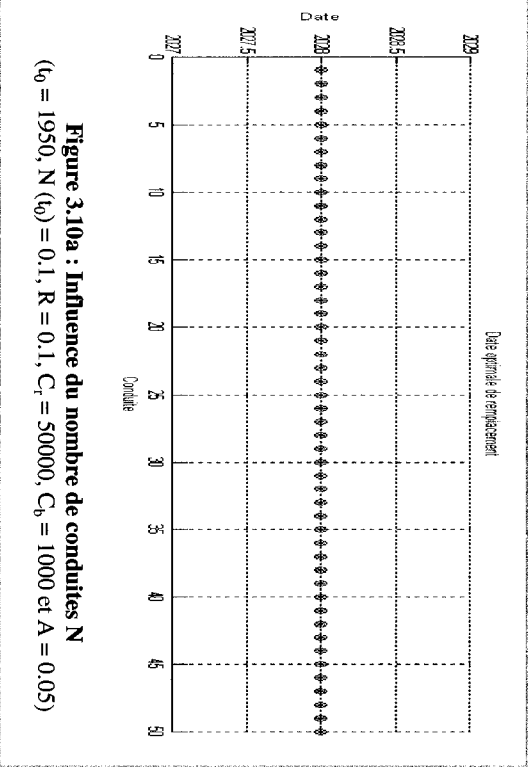


Figure 3.9b : Coût total du système par génération
($N = 50$, $M = 20$, $N_{gen} = 100$)

* dates simulées (croisement 2-points) o dates simulées (croisement uniforme) * coût minimal théorique



* dates simulées (croisement 2-points) o dates simulées (croisement uniforme) * coût minimal théorique

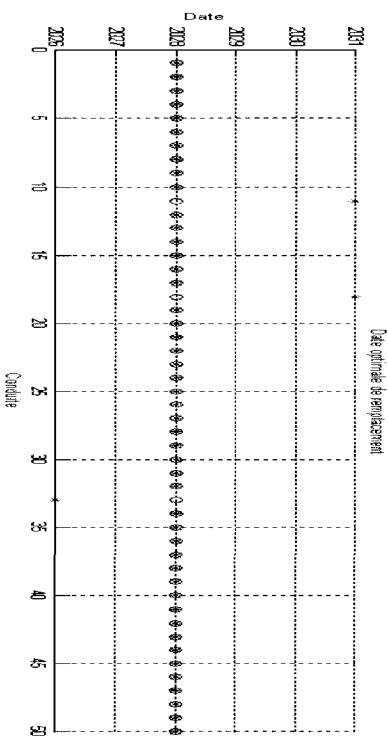


Figure 3.11a : Influence du nombre de génération maximal N_{gen}
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

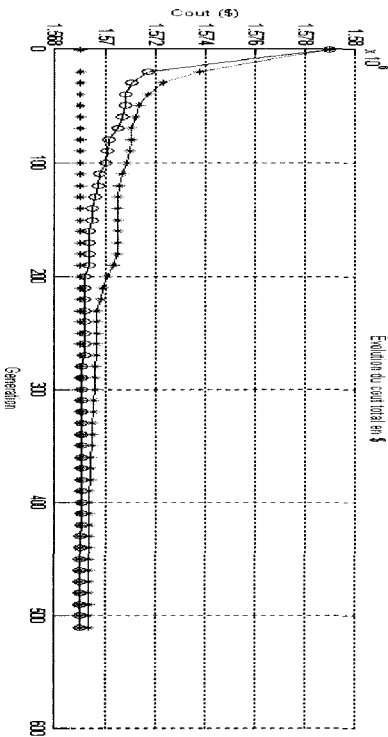


Figure 3.11a : Coût total du système par génération
($N = 50$, $M = 5$, $N_{gen} = 500$)

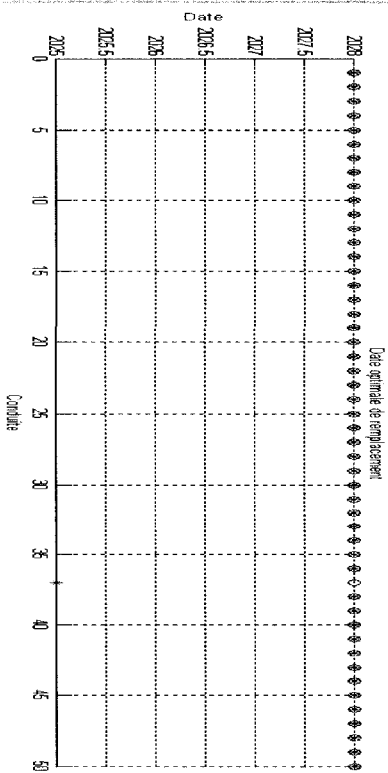


Figure 3.11b : Influence du nombre de génération maximal N_{gen}
($t_0 = 1950$, $N(t_0) = 0.1$, $R = 0.1$, $C_r = 50000$, $C_b = 1000$ et $A = 0.05$)

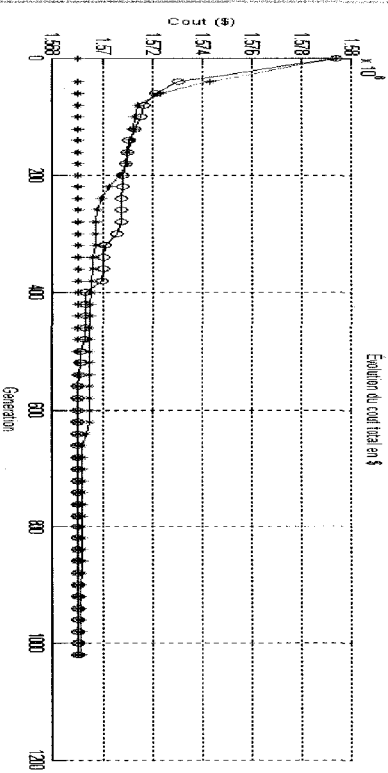


Figure 3.11b : Coût total du système par génération
($N = 50$, $M = 5$, $N_{gen} = 1000$)

En conclusion, la mise en œuvre de l'algorithme génétique dépend surtout du type de problème étudié. En effet, le codage des solutions est adapté au problème posé. Les opérateurs usuels de l'algorithme sont les opérateurs de sélection, croisement et mutation. Bien qu'il existe des méthodes pour implémenter ces différents opérateurs, en pratique, il y a toujours une possibilité d'ajouter des variantes à ces méthodes afin d'améliorer la convergence vers la solution optimale. L'opérateur de croisement favorise l'exploration de l'espace de recherche, tandis que la sélection permet de faire évoluer la population vers les meilleurs individus. Cependant, lorsque la taille de la population est petite, il peut arriver que certains individus favorisés par le hasard se répandent au détriment des autres; ce phénomène s'appelle la dérive génétique. L'opérateur de mutation prend alors tout son intérêt car il permet de contrebalancer ce phénomène en introduisant constamment dans la population de nouveaux individus.

Dans des problèmes de maximisation ou minimisation des coûts, il est très important de toujours conserver dans la nouvelle population, le meilleur individu de l'ancienne génération, car le brassage des individus parents peut créer des individus enfants dont aucun ne soit trouvé meilleur que le meilleur des parents.

Enfin, les équations théoriques de Shamir et Howard confirment la justesse des résultats obtenus par l'algorithme génétique. Cependant, comme on l'a constaté pour un taux de bris linéaire, l'atteinte de l'optimum n'est pas toujours garantie.

CHAPITRE 4

DEUXIÈME MODÈLE

Dans le premier modèle, les variables de décision sont les dates optimales de remplacement des conduites de l'ensemble du réseau, obtenues pour un horizon de temps fixe sans contraintes budgétaires. Ce modèle permet une planification financière des interventions de remplacement, car le coût total minimal du système et les budgets annuels requis peuvent être déterminés. Cependant, dans la pratique, les gestionnaires n'ont pas toujours les fonds suffisants pour intervenir sur les conduites du réseau. D'où l'utilité du deuxième modèle qui permet de prendre en compte les restrictions budgétaires afin d'évaluer l'impact de celles-ci sur la programmation des interventions faite à l'aide du premier modèle.

Ainsi, un premier scénario sans contraintes budgétaires va être retenu afin d'utiliser la programmation obtenue comme base pour les scénarios avec contraintes budgétaires. Ensuite deux scénarios avec contraintes budgétaires vont être étudiés. Dans le premier scénario, on va considérer uniquement le remplacement comme option d'intervention, et on va imposer au système des contraintes budgétaires annuelles. Dans le second scénario, on va considérer les deux options d'intervention qui sont la réhabilitation et le remplacement, sans imposer des contraintes budgétaires annuelles. Pour éviter que le système ne converge vers 100% de réhabilitation car celle-ci est moins dispendieuse, on va imposer une contrainte budgétaire au niveau du coût total du système.

4.1. SCÉNARIO SANS CONTRAINTES

On va utiliser le premier modèle pour analyser l'impact des stratégies de renouvellement d'un réseau de 50 conduites. On subdivise le réseau en 5 groupes de 10 conduites, chaque groupe ayant les mêmes conditions initiales. Le coût de remplacement est fixé à 50 000 \$ pour toutes les conduites. Deux scénarios sont étudiés : le remplacement à 100% ou la réhabilitation à 100% du réseau. En effet, étant donné que le système est sans contrainte budgétaire et que la fonction économique vise à minimiser le coût total du système, lorsque les deux options de remplacer et de réhabiliter sont possibles pour toutes les conduites, les simulations montrent que la solution converge toujours vers une réhabilitation du réseau à 100%. Les coûts de réhabilitation étant inférieurs aux coûts de remplacement expliquent cette convergence.

4.1.1. Choix de remplacer toutes les conduites du réseau

Chaque groupe de conduites a une date optimale de remplacement déterminée. En effet, les conditions initiales sont les mêmes dans chaque groupe. Le choix de procéder à cette subdivision du réseau vient du fait qu'en pratique, il est toujours possible de rassembler les conduites d'un réseau en groupes de conduites ayant la même date optimale de remplacement, même si à l'intérieur du groupe les conditions initiales des conduites sont différentes. Les résultats de la simulation sont présentés aux figures 4.1. Le coût minimal total est de 2 190 100 \$.

4.1.2. Choix de réhabiliter toutes les conduites du réseau

Un scénario envisageable est de réhabiliter systématiquement toutes les conduites au lieu de les remplacer. Dans ce scénario, on fixe le coût d'une réhabilitation à 75 % du coût de remplacement, soit 37 500 \$ pour toutes les conduites, tout en conservant leurs conditions initiales. Les résultats des simulations sont présentés aux figures 4.2. Le coût minimal total est de 1 851 900 \$.

On constate que la réhabilitation demande une intervention anticipée sur les conduites (voir tableau 4.1). En effet, il ne faut pas arriver à la fin de la durée de vie utile (date optimale de remplacement) pour réhabiliter une conduite, mais il faut intervenir plus tôt sur celle-ci. La réhabilitation permet également de réduire le coût minimal du système. Cependant, il faut être conscient que la durée de vie d'une réhabilitation est moins longue que celle d'un remplacement et par conséquent, elle est une solution qui permet d'alléger les investissements à moyen terme tout en augmentant les niveaux de service requis.

Les figures 4.1a et 4.1b montrent les répartitions des budgets annuels respectivement pour chaque scénario. On a ainsi pour le réseau, la planification des interventions sur 32 ans ainsi que les budgets annuels en dollars actualisés requis pour la mettre en œuvre.

Tableau 4.1: Scénarios sans contraintes budgétaires

Lot	Remplacement					Réhabilitation				
	1	2	3	4	5	1	2	3	4	5
2007	-	-	-	-	-	100%	100%	-	-	-
2008	100%	-	-	-	-	-	-	-	-	-
2009	-	-	-	-	-	-	-	-	-	-
2010	-	-	-	-	-	-	-	-	-	-
2011	-	-	-	-	-	-	-	-	-	-
2012	-	-	-	-	-	-	-	100%	-	-
2013	-	100%	-	-	-	-	-	-	-	-
2014	-	-	-	-	-	-	-	-	-	-
2015	-	-	-	-	-	-	-	-	-	-
2016	-	-	-	-	-	-	-	-	-	-
2017	-	-	-	-	-	-	-	-	100%	-
2018	-	-	100%	-	-	-	-	-	-	-
2019	-	-	-	-	-	-	-	-	-	-
2020	-	-	-	-	-	-	-	-	-	-
2021	-	-	-	-	-	-	-	-	-	-
2022	-	-	-	-	-	-	-	-	-	100%
2023	-	-	-	100%	-	-	-	-	-	-
2024	-	-	-	-	-	-	-	-	-	-
2025	-	-	-	-	-	-	-	-	-	-
2026	-	-	-	-	-	-	-	-	-	-
2027	-	-	-	-	-	-	-	-	-	-
2028	-	-	-	-	100%	-	-	-	-	-
2029	-	-	-	-	-	-	-	-	-	-
2030	-	-	-	-	-	-	-	-	-	-
2031	-	-	-	-	-	-	-	-	-	-
2032	-	-	-	-	-	-	-	-	-	-
2033	-	-	-	-	-	-	-	-	-	-
2034	-	-	-	-	-	-	-	-	-	-
2035	-	-	-	-	-	-	-	-	-	-
2036	-	-	-	-	-	-	-	-	-	-
2037	-	-	-	-	-	-	-	-	-	-
2038	-	-	-	-	-	-	-	-	-	-

On observe pour chaque lot de conduites les réponses du modèle face à un choix exclusif de remplacement ou de réhabilitation. Pour cette dernière, toutes les interventions sont anticipées.

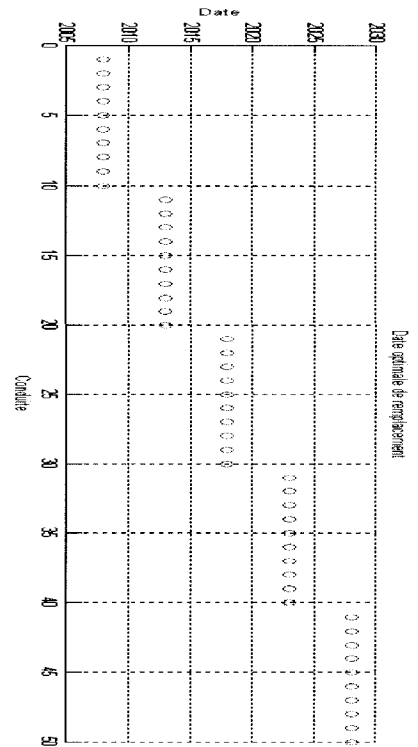


Figure 4.1a : Date de remplacement (scénario 100% de remplacement)
($N = 10$, $M = 50$, $N_{gen} = 1000$)

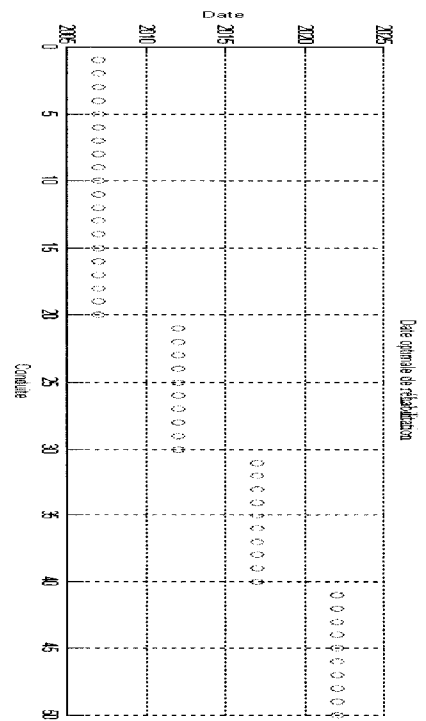


Figure 4.2a : Date de réhabilitation (scénario 100% de réhabilitation)
($N = 50$, $M = 50$, $N_{gen} = 1000$)

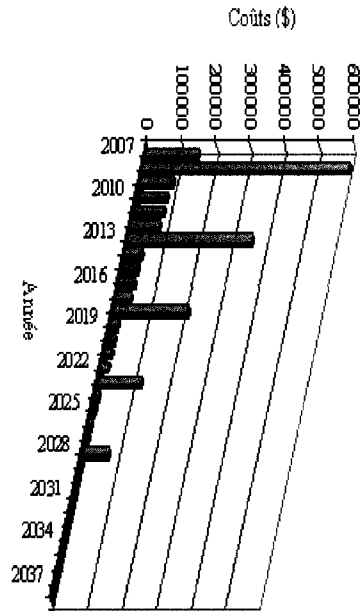


Figure 4.1a : Répartition des budgets annuels sur 32 ans

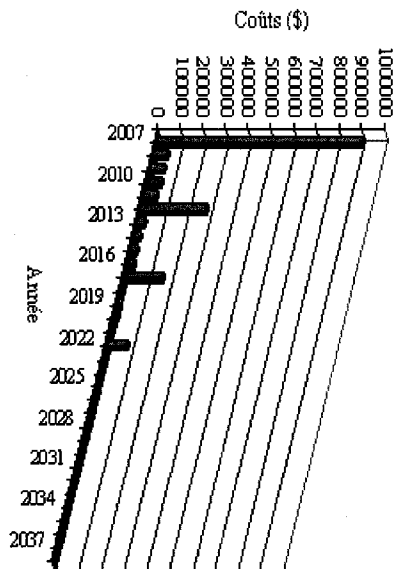


Figure 4.2b : Répartition des budgets annuels sur 32 ans

Dans le modèle 1 sans contraintes budgétaires, on peut calculer les budgets annuels requis pour couvrir les investissements sur l'horizon de temps choisi. Connaissant les enveloppes budgétaires annuelles pour intervenir sur le réseau, les gestionnaires ne disposent pas toujours du capital nécessaire pour renouveler leurs conduites. Le deuxième modèle, détaillé au chapitre suivant, prend alors en compte les restrictions budgétaires.

4.2. PREMIER SCÉNARIO AVEC DES CONTRAINTES BUDGÉTAIRES ANNUELLES

Le remplacement est l'unique option d'intervention. On impose au système des contraintes budgétaires annuelles. Pour cela, on répartit le budget annuel disponible à chaque année sur l'horizon de temps, de manière homogène en dollars courants.

On travaille sur le même réseau de 5 lots de 10 conduites, ayant chacun leur date de remplacement. L'effet des restrictions budgétaires va pouvoir être quantifiés à partir des variables de décision qui sont les proportions $\alpha_{k,t}$ de chaque lot k de conduites à remplacer à une date t sur l'horizon de temps fixé.

Les résultats des simulations sont présentés aux figures 4.3 à 4.5. La première simulation faite avec le modèle 1 donne un coût minimal total de 2,19 M\$ pour le remplacement complet du réseau d'ici 32 ans. La répartition des coûts annuels requis est aussi donnée par le modèle 1. On observe des pics budgétaires aux dates de remplacement de chaque lot k de conduites sur la figure 4.3, le budget annuel moyen étant de 68 440 \$.

Sachant que l'on ne dispose pas du budget annuel requis aux périodes de pic financier, à l'aide du deuxième modèle, on va imposer une répartition homogène du budget annuel en dollars courants de 1,5 fois le coût de 2,19 M\$ sur 32 ans, soit environ 100 000 \$ à chaque année sur 32 années. Une seconde simulation faite avec le modèle 2 et on peut observer sur la figure 4.3 que des pics financiers sont absents et les budgets annuels tendent à s'homogénéiser. Même s'il faut plus investir pendant les 10 premières années, les budgets annuels requis sont moins divergents que ceux du premier modèle.

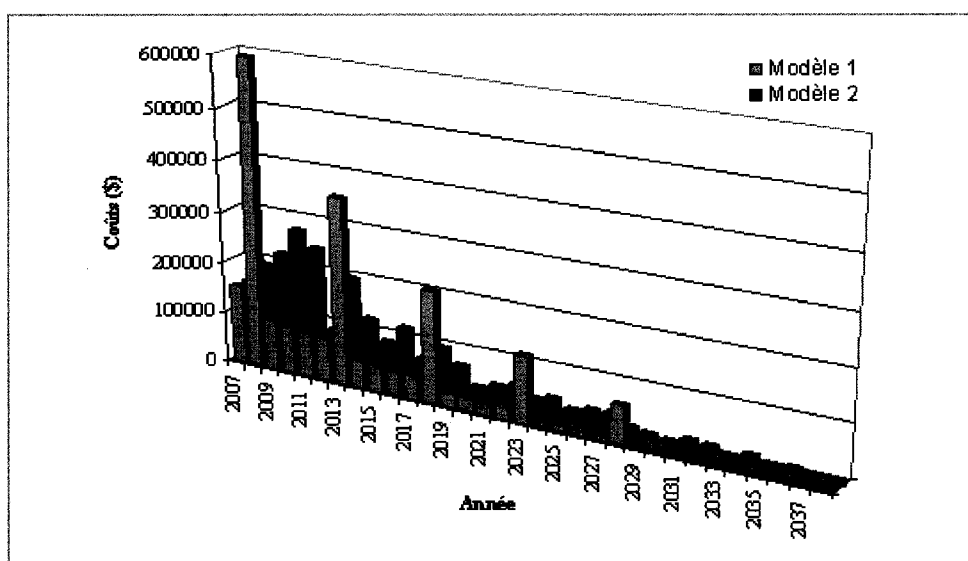


Figure 4.3 : Budgets annuels en dollars courants

La figure 4.4 montre l'effet des contraintes budgétaires sur les dates de remplacement. On constate que les dates de remplacement des conduites sont avancées ou reculées dans le temps afin de balancer le pourcentage des conduites remplacées chaque année pour homogénéiser les budgets annuels.

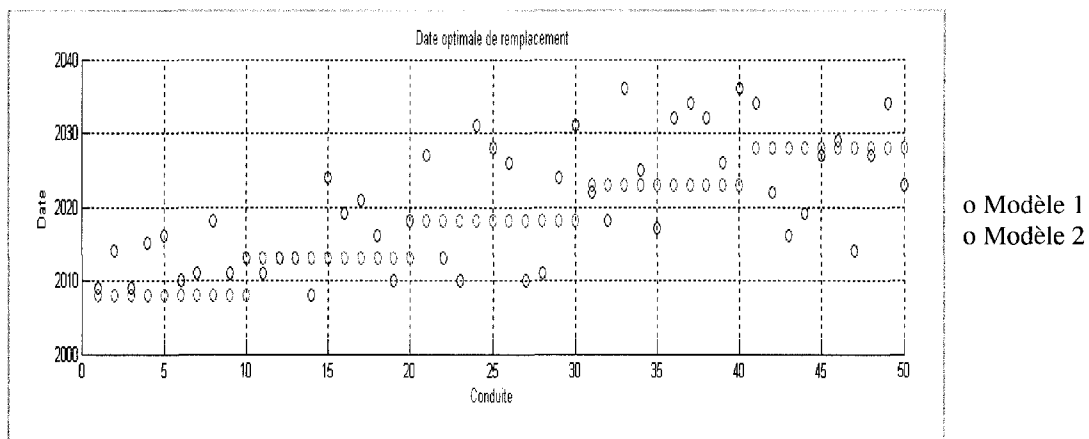


Figure 4.4 : Date de remplacement avec contraintes budgétaires annuelles
($M = 50$, $N = 50$, $N_{\text{gen}} = 1000$)

Les proportions $\alpha_{k,t}$ de chaque lot k de conduites sont notées dans le tableau 4.2.

Tableau 4.2 : Proportions des lots de conduites à remplacer sur l'horizon de temps

Lot	Remplacement (modèle 1)					Remplacement (modèle 2)				
	1	2	3	4	5	1	2	3	4	5
2007	-	-	-	-	-	-	-	-	-	-
2008	100%	-	-	-	-	-	10%	-	-	-
2009	-	-	-	-	-	20%	-	-	-	-
2010	-	-	-	-	-	10%	10%	20%	-	-
2011	-	-	-	-	-	20%	10%	10%	-	-
2012	-	-	-	-	-	-	-	-	-	-
2013	-	100%	-	-	-	10%	20%	10%	-	-
2014	-	-	-	-	-	10%	-	-	-	10%
2015	-	-	-	-	-	10%	-	-	-	-
2016	-	-	-	-	-	10%	10%	-	-	10%
2017	-	-	-	-	-	-	-	-	10%	-
2018	-	-	100%	-	-	10%	10%	-	10%	-
2019	-	-	-	-	-	-	10%	-	-	10%
2020	-	-	-	-	-	-	-	-	-	-
2021	-	-	-	-	-	-	10%	-	-	-
2022	-	-	-	-	-	-	-	-	10%	10%
2023	-	-	-	100%	-	-	-	-	-	10%
2024	-	-	-	-	-	-	10%	10%	-	-
2025	-	-	-	-	-	-	-	-	10%	-
2026	-	-	-	-	-	-	-	10%	10%	-

	Remplacement (modèle 1)					Remplacement (modèle 2)				
2027	-	-	-	-	-	-	-	10%	-	20%
2028	-	-	-	-	100%	-	-	10%	-	-
2029	-	-	-	-	-	-	-	-	-	10%
2030	-	-	-	-	-	-	-	-	-	-
2031	-	-	-	-	-	-	-	20%	-	-
2032	-	-	-	-	-	-	-	-	20%	-
2033	-	-	-	-	-	-	-	-	-	-
2034	-	-	-	-	-	-	-	-	10%	20%
2035	-	-	-	-	-	-	-	-	-	-
2036	-	-	-	-	-	-	-	-	20%	-
2037	-	-	-	-	-	-	-	-	-	-
2038	-	-	-	-	-	-	-	-	-	-

On peut constater que la moyenne des pourcentages des conduites à remplacer tend à s'harmoniser sur l'horizon de temps. Par ailleurs, le coût total du système donné par le modèle 2 est de l'ordre de 2,29 M\$. La figure 4.5 montre l'évolution du coût total du système. Celui-ci ne diminue pas sans cesse à cause des restrictions budgétaires.

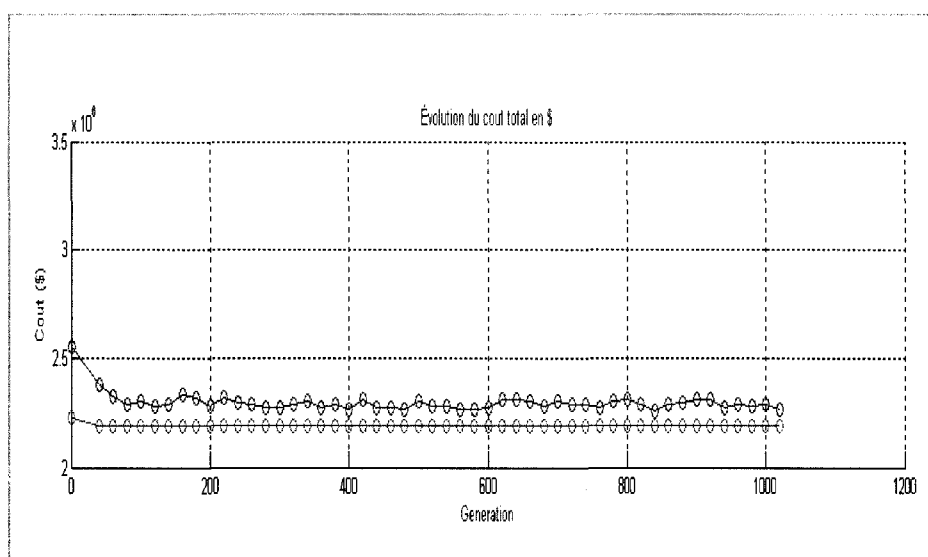


Figure 4.5 : Coût total du système avec contraintes budgétaires annuelles
($M = 50$, $N = 50$, $N_{\text{gen}} = 1000$)

En fonction des conditions initiales des conduites et des restrictions budgétaires, il est possible qu'aucune solution respectant les budgets imposés ne soit trouvée. Par conséquent, le modèle 2 permet de trouver la solution qui respecte au mieux les contraintes financières imposées.

4.3. DEUXIÈME SCÉNARIO AVEC UNE CONTRAINTE SUR LE COÛT TOTAL DU SYSTÈME

On n'impose aucune contrainte sur les budgets annuels. Les options d'intervention sont la réhabilitation et le remplacement. On impose que le coût total ne doit pas descendre en deçà d'une certaine valeur fixée. En effet, en l'absence de cette contrainte, la solution convergerait vers la réhabilitation systématique de toutes les conduites. Le but de ce scénario est d'étudier le comportement du modèle en imposant un coût total.

La simulation est faite avec le modèle 1 sans contrainte converge vers la solution de réhabilitation à 100% du réseau. Ce modèle donne le coût minimal du système lorsque toutes les conduites sont réhabilitées sur un horizon de temps donné. On choisit d'imposer un coût total supérieur à ce dernier, mais inférieur au coût total qu'a donné le modèle 1 lorsque les conduites sont remplacées à 100%. Une seconde simulation avec ce coût total fixé à respecter est alors exécutée.

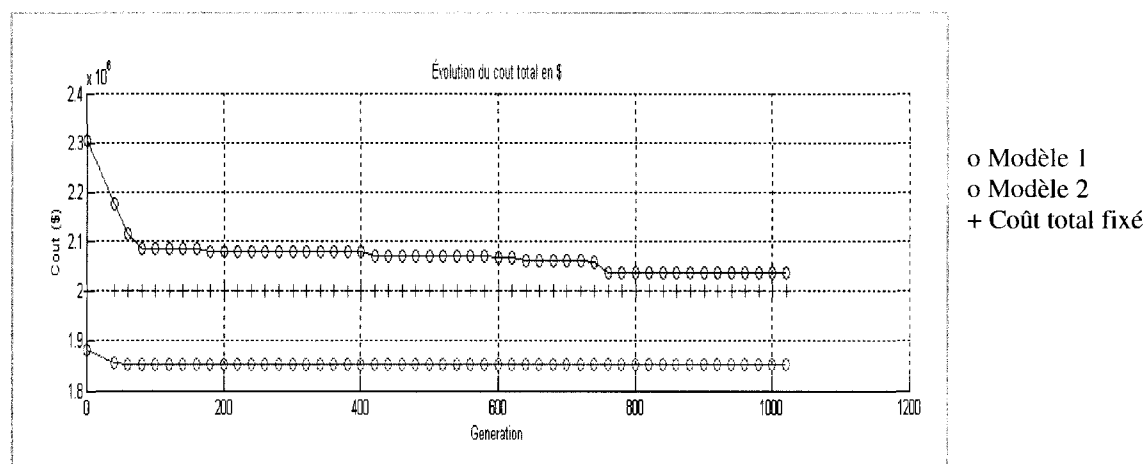


Figure 4.6 : Coût total du système avec contrainte sur le coût total
($M = 50$, $N = 50$, $N_{\text{gen}} = 1000$)

La figure 4.6 montre l'évolution du coût total en fonction des générations. Et on constate que ce coût converge vers le coût total fixé à 2 M\$. La solution du modèle 2 au bout de 1 000 générations donne un coût total de 2,13 M\$. La figure 4.7 montre les dates des interventions sur les conduites. Dans le tableau 4.3, on peut observer quelles proportions de conduites sont remplacées $\alpha_{k,t}$ ou réhabilitées $\beta_{k,t}$ dans chaque lot k .

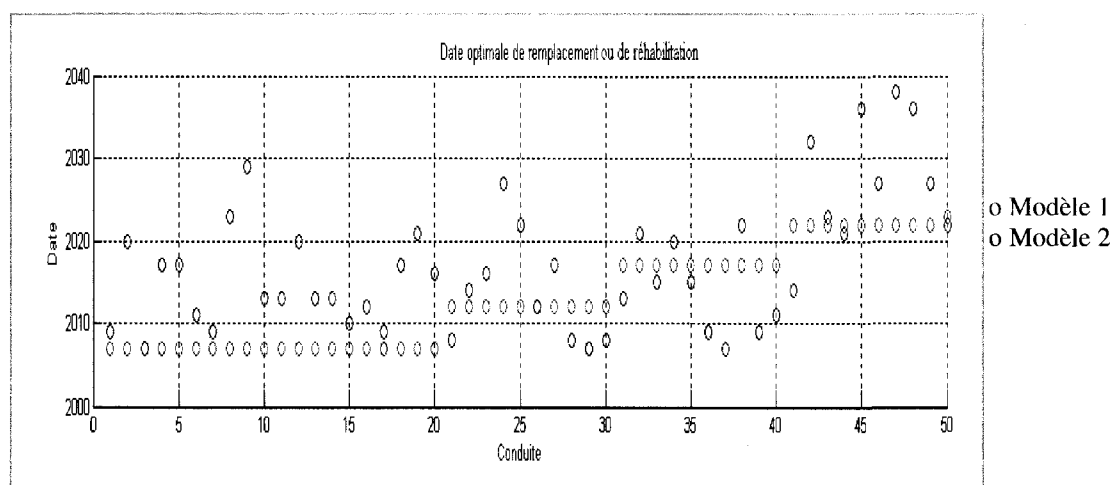


Figure 4.7 : Date de remplacement ou de réhabilitation avec contrainte sur le coût total
($M = 50$, $N = 50$, $N_{\text{gen}} = 1000$)

	Réhabilitation (modèle 1)					Remplacement ou réhabilitation (modèle 2)				
2035	-	-	-	-	-	-	-	-	-	-
2036	-	-	-	-	-	-	-	-	-	0-20%
2037	-	-	-	-	-	-	-	-	-	-
2038	-	-	-	-	-	-	-	-	-	10-0%

Note : X = date optimale de remplacement

4.4. DISCUSSIONS SUR LES RÉSULTATS OBTENUS

La mise en œuvre de la solution avec des contraintes budgétaires par l'algorithme génétique est un processus qui s'est avéré complexe. Tout d'abord, la population finale du premier modèle a été choisie comme la population initiale du deuxième modèle. Ensuite, chaque individu (solution) est évalué à l'aide d'une nouvelle fonction d'évaluation. Pour cela, il y a plusieurs possibilités qui s'offrent :

- garder une fonction d'évaluation qui minimise le coût total du système et rejeter les solutions qui ne rencontrent pas les contraintes budgétaires;
- ajouter à la fonction d'évaluation qui minimise le coût total du système un coût de pénalité pour discriminer les solutions qui ne rencontrent pas les contraintes budgétaires;
- adapter la fonction d'évaluation à la contrainte.

La première solution a été rejetée car beaucoup d'individus risquent d'être éliminés et si les contraintes sont sévères, il peut arriver que l'algorithme ne trouve aucune solution. Pourtant, l'objectif est autant la recherche d'une solution exacte que la recherche d'une solution respectant au mieux les critères.

La deuxième solution a été rejetée car la proportion des solutions très insatisfaisantes peut devenir très grande par rapport au reste de la population.

La troisième solution a été retenue. Soit A une année dans l'horizon de temps choisi T , la figure 4.8 illustre la nouvelle fonction d'évaluation.

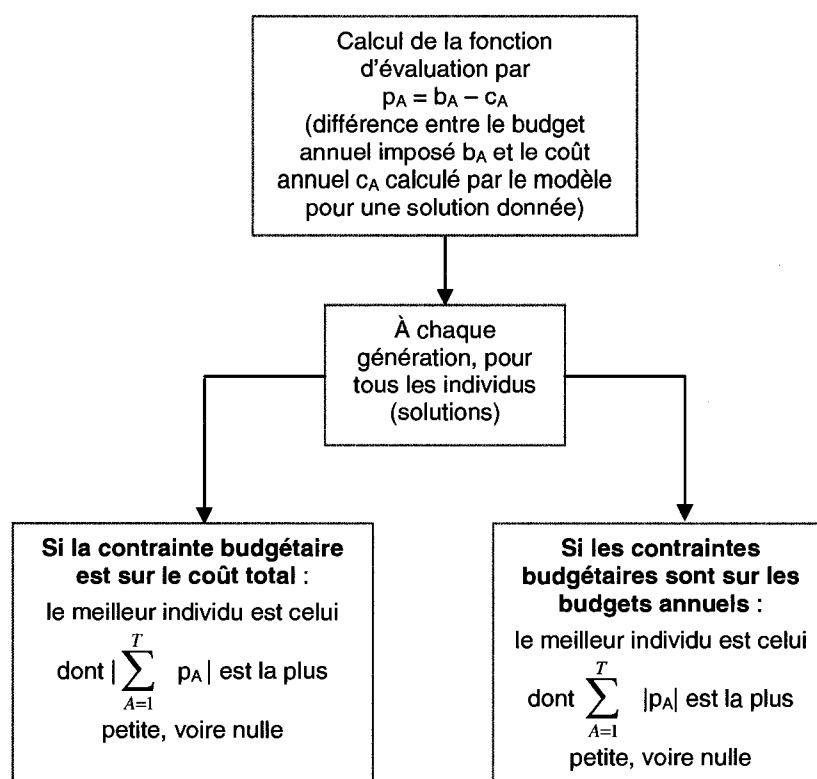


Figure 4.8 Adaptation de la fonction d'évaluation avec les contraintes budgétaires

L'objectif reste la minimisation du coût total du système. Cependant, la fonction d'évaluation sert à sélectionner dans la génération courante, non pas la solution qui a le plus petit coût total du système, mais la solution qui respecte au mieux les contraintes budgétaires. Elle permet ainsi de trouver une solution de "bon compromis" entre le respect des contraintes budgétaires et la minimisation du coût total du système.

Connaissant les dates obtenues par le premier modèle sans contraintes budgétaires, les nouvelles dates et les choix d'intervention pour chaque conduite venant du deuxième modèle avec contraintes budgétaires, on détermine les proportions des conduites à réhabiliter et à remplacer dans chaque groupe.

Avec un minimum de deux heures de temps de calcul pour simuler chaque scénario avec contraintes budgétaires, on obtient une nouvelle programmation des interventions. Les résultats sont quantifiables car ils donnent en fonction du budget disponible, les proportions de conduites à remplacer et à réhabiliter à chaque année dans chaque groupe. Et la décision revient aux gestionnaires de choisir dans chaque groupe les conduites sur lesquelles ils veulent intervenir tout au long de l'horizon. La pertinence des résultats est observable dans la figure 4.4 car, même si les interventions sur les conduites sont en général reportées dans le temps pour tous les groupes, on observe tout de même que, plus le groupe de conduites est critique, plus rapidement les interventions sont faites sur l'ensemble des conduites de ce groupe par rapport au reste des conduites du réseau.

Dans la mise en œuvre de la solution, les solutions qui ne rencontrent pas les contraintes fixées ne sont pas éliminées et aucun coût de pénalité n'est affecté à ces solutions. Ainsi la solution obtenue peut être la meilleure parmi des solutions ne respectant pas les contraintes.

CONCLUSION

À travers ce projet, l'algorithme génétique a été appliqué à des modèles inspirés de la démarche de Shamir et Howard (1979) en vue d'obtenir les dates optimales de remplacement des conduites (premier modèle), et de l'approche de Karaa et *al.* (1987) pour obtenir les proportions de conduites à remplacer ou réhabiliter sur l'horizon de temps afin de répondre au mieux à des restrictions budgétaires imposés (deuxième modèle).

Cette étude a permis d'évaluer les atouts et les limites de l'algorithme génétique utilisé comme méthode de calcul pour les deux problèmes d'optimisation posés. L'un des avantages majeurs observé de l'algorithme génétique est qu'il s'adapte à des problèmes où : soit la mise en œuvre de la solution par des méthodes non linéaires est complexe, soit il n'existe pas de méthode exacte de résolution. Comme on a pu le constater, aucun bagage mathématique n'est requis pour la mise en œuvre de l'algorithme. De plus, on a pu aussi observer que l'algorithme génétique est un outil robuste, dans la mesure où il peut fournir des solutions très bonnes mais pas nécessairement optimales comme c'est le cas dans l'utilisation d'un taux de bris linéaire. Cependant, la robustesse de l'algorithme génétique a un prix, dans la mesure où un très grand nombre de générations (ou de calculs de la valeur d'adaptation) peut être nécessaire avant d'atteindre la solution optimale. Ce nombre devient problématique lorsque le nombre d'individus augmente et le coût de calcul (en temps) de la valeur d'adaptation augmente. Par exemple, pour le

premier modèle, le temps de réponse était de l'ordre d'une dizaine (voir quinzaine) de minutes, tandis que pour le modèle, le temps de réponse était d'environ deux heures.

Cette étude a aussi permis de constater combien la vitesse de convergence est cruciale, et pour tout type de problème, il est important de trouver des méthodes alternatives pour accélérer la convergence de l'algorithme. En l'occurrence, pour cette étude, l'opérateur de création du meilleur des meilleurs individus a été créé.

En somme, l'algorithme génétique est un outil qui suscite un engouement et l'analyse faite au travers de cet outil mérite d'être poussée encore plus loin : d'une part, en essayant cet outil sur un réseau de conduites plus large (d'environ 200 conduites) pour observer la réaction d'un tel outil en terme d'efficacité de convergence et de temps de calculs; et d'autre part, une comparaison des résultats obtenus avec ceux des autres modèles peut être faite pour mieux évaluer la robustesse de cet outil (c'est-à-dire son efficacité à donner de "bonnes" solutions).

Par ailleurs, une analyse encore plus poussée peut être faite en rajoutant dans l'algorithme des contraintes au niveau des techniques d'intervention en imposant un taux d'utilisation pour chaque technique sur l'ensemble des conduites à l'intérieur de l'horizon de temps fixé. Par exemple, la Politique nationale de l'eau (engagement 45) exige depuis 2005, un taux d'utilisation des techniques de réhabilitation des conduites de 25% par rapport au remplacement.

De même, une recherche plus approfondie en introduisant plusieurs critères de décision peut être étudiée. Dans ce projet, seul le critère économique a été abordé, et des

scénarios d'intervention sous des contraintes budgétaires ont pu être élaborés. Cependant, dans un réseau d'eau potable, la performance hydraulique et la qualité de l'eau sont aussi des critères importants. On pourrait alors évaluer la capacité de cet algorithme à gérer par une formulation, monocritère ou multicritère, l'ensemble de ces objectifs dans le but d'introduire ces modules dans un système intégré d'aide à la décision pour le renouvellement des réseaux d'eau potable.

De nos jours, les modèles d'optimisation sont des outils utiles pour une planification à long terme des interventions sur les conduites d'eau potable. En effet, ils permettent d'une part d'élaborer différents scénarios d'interventions et d'autre part de mesurer l'impact des investissements insuffisants sur le réseau de conduites soumis à différentes autres contraintes. Bien que les modèles d'optimisation soient des outils puissants d'aide à la décision, les interventions sur les conduites d'aqueduc ne dépendent pas uniquement de l'état des conduites d'eau potable mais aussi des conditions des autres infrastructures adjacentes telles que les réseaux d'égouts et les chaussées. Ainsi, un arbre de décision prenant en compte les résultats du modèle d'optimisation et les facteurs externes (travaux sur les égouts et chaussées, choix politiques, problématiques particulières) viendra raffiner l'optimisation des plans d'interventions sur les conduites d'eau potable. En définitive, le modèle d'optimisation est une tentative d'automatiser la décision. Cependant, le choix revient toujours aux gestionnaires municipaux qui doivent se bâtir un arbre de décision pour chacune des conduites traitées.

BIBLIOGRAPHIE

- AFSHAR, M. H., MARINO, M. A., 2005 "A convergent genetic algorithm for pipe Network Optimisation". *Scientica Iranica*. 12:4. 392-401.
- BLOCH, T.A. 1998. *Storage and pumping system optimization using genetic algorithms*. 188 p. Mémoire de maîtrise, University of Toronto.
- CHAMBERS L., *Practical handbook of genetic algorithms*, Boca Raton [Florida]: CRC Press, 1995.
- CINGALA, E. 2003. *Algorithme génétique introduisant l'âge des individus*. Consulté en 2007, tiré de http://www.moteurprog.com/Tutoriaux/Tutorial.php?ID_tuto=99
- DANDY, G.C., ENGELHARDT, M. 2001. "Optimal scheduling of water pipe replacement using genetic algorithms". *Journal of Water Resources Planning and Management*. 127:4 . 214-223.
- DRDI, L. 2005. *Les algorithmes génétiques*. In Thèse de doctorat, INRS-ETE.
- GEN M. ET CHENG R., *Genetic algorithms and engineering design*, New York; Toronto: John Wiley & Sons, Inc., 1997.
- GEN M. ET CHENG R., *Genetic algorithms and engineering optimization*, New York; Toronto: Wiley, 2000.
- GOLDBERG D. E., *Genetic algorithms in search, optimization, and machine learning*, Reading, Mass.; Don Mill, Ont.: Addison-Wesley, 1989.
- HALHAL, D. ET AL. 1997. "Water Network Rehabilitation with Structured Messy Genetic Algorithm". *Journal of water resources planning and management*. 137-146.
- HAUPT R. L. ET HAUPT S. E., *Practical genetic algorithms*, New York; Toronto: Wiley, 1998.
- KARAA A. F. ET AL. 1987. "Budgeting of water distribution improvement projects ". *Journal of Water Resources Planning and Management*. Vol. 113, No.3.
- MATLAB. Version 6.5.
- MICHALEWICZ Z., *Genetic algorithms + data structures = evolution programs*, Berlin : Springer, 1996.

MINISTÈRE DES AFFAIRES MUNICIPALES ET DES RÉGIONS. *Guide d'élaboration d'un plan d'intervention pour le renouvellement des conduites d'eau potable et d'égout*. Consulté en avril 2007, tiré de http://www.mamr.gouv.qc.ca/infrastructures/infr_docu.asp

SHAMIR, U., HOWARD, D.D., 1979. "An analytic approach to scheduling pipe replacement". *Journal AWWA*. 248-258.

VOSE M. D., *The simple genetic algorithm: foundations and theory*, Cambridge, Mass. : MIT Press, 1999.

WILSON, L.L., MOSHAVEGH, F., FREY, J., KOLOKAS, L., SIMPSON, A.R. 2003. " Cost savings and enhanced reliability for main rehabilitation and replacement ". *Journal of American Water Works Association*. 95:8. 54-64.

WU, Z.Y. ET AL., 2001. "Using genetic algorithms to rehabilitate distribution systems". *Journal AWWA*. 74-85.

WU, Z.Y., SIMPSON, A.R. 2001. "Competent genetic-evolutionary optimization of water distribution systems". *Journal of Computing in Civil Engineering*. 15:02. 89-101.

WU, Z.Y., SIMPSON, A.R. 2002. "La recherche de limite auto-adaptative d'un algorithme genetique et son application aux systemes de distribution d'eau (A self-adaptive boundary search genetic algorithm and its application to water distribution systems)". *Journal of Hydraulic Research*. 40:2. 191-204.

ZALZALA A. M., FLEMING P.J., *Genetic algorithms in engineering systems*, London: Institution of Electrical Engineers, 1997.

ANNEXE 1

DÉVELOPPEMENT MATHÉTIQUE POUR LE CALCUL DE LA DATE

OPTIMALE

Les coûts totaux sont : $P_t(t_r) = \sum_{t=t_p}^{t_r} \frac{C_m(t)}{(1+R)^{t-t_p}} + \frac{C_r}{(1+R)^{t_r-t_p}}$.

On cherche t_r qui minimise $P_r(t_r) : \underset{t_r}{Min}[P_r(t_r)]$

Nous savons que nous pouvons remplacer $\sum$ par $\int$. Soit P_r la fonction dépendant de t_r dont nous recherchons le minimum. Soit P'_r la dérivée de la fonction P_r . Mathématiquement le minimum de la fonction P_r est une racine de sa dérivée. Par

ailleurs, nous savons aussi que $P_r(t_r) = \int_{t_p}^{t_r} P'_r(t) dt + \text{Constante}$

$$P_t(t_r) = \sum_{t=t_p}^{t_r} \frac{C_m(t)}{(1+R)^{t-t_p}} + \frac{C_r}{(1+R)^{t_r-t_p}}$$

$$P_t(t_r) = \int_{t=t_p}^{t=t_r} \frac{C_m}{(1+R)^{t-t_p}} dt + C_r \times \frac{1}{e^{(t_r-t_p)\ln(1+R)}}$$

$$P_t(t_r) = \int_{t=t_p}^{t=t_r} \frac{C_m}{(1+R)^{t-t_p}} dt + C_r \times e^{-(t_r-t_p)\ln(1+R)}$$

$$P_t(t_r) = \int_{t=t_p}^{t=t_r} \frac{C_m}{(1+R)^{t-t_p}} dt + \int_{t=t_p}^{t=t_r} -C_r \times \ln(1+R) e^{-(t-t_p)\ln(1+R)} dt + K,$$

avec K une constante

$$P_t(t_r) = \int_{t=t_p}^{t=t_r} \frac{C_m}{(1+R)^{t-t_p}} - C_r \times \ln(1+R) e^{-(t-t_p)\ln(1+R)} dt + K$$

$$P_t(t_r) = \int_{t=t_p}^{t=t_r} \frac{C_m}{(1+R)^{t-t_p}} - \frac{C_r \ln(1+R)}{(1+R)^{t-t_p}} dt + K$$

$$P_t(t_r) = \int_{t=tp}^{t=tr} \frac{C_m - C_r \ln(1+R)}{(1+R)^{t-tp}} dt + K$$

D'où $P_r'(t_r) = 0$ équivaut à $C_m - C_r \ln(1+R) = 0$.

Soit $C_m = C_b \times N(t) = C_r \ln(1+R)$

$$N(t) = N(t_0) e^{A(t-t_0)} = \frac{C_r \ln(1+R)}{C_b}$$

$$\text{D'où } e^{A(t-t_0)} = \frac{C_r \ln(1+R)}{N(t_0)C_b}$$

$$A(t-t_0) = \ln\left(\frac{C_r \ln(1+R)}{N(t_0)C_b}\right)$$

$$t = t_0 + \frac{1}{A} \ln\left(\frac{C_r \ln(1+R)}{N(t_0)C_b}\right) = t^*, \text{ } t^* \text{ étant la date optimale théorique}$$

Par ailleurs, $N(t_p) = N(t_0) e^{A(t_p-t_0)}$. Alors, $N(t_0) = N(t_p) e^{-A(t_p-t_0)}$

$$\text{Donc, } t^* = t_0 + \frac{1}{A} \ln\left(\frac{C_r \ln(1+R)}{N(t_p)C_b} e^{A(t_p-t_0)}\right) \text{ sachant que } \ln(a \times b) = \ln a + \ln b$$

$$t^* = t_0 + \frac{1}{A} \ln\left(\frac{C_r \ln(1+R)}{N(t_p)C_b}\right) + \frac{1}{A} \ln e^{A(t_p-t_0)}$$

$$t^* = t_0 + \frac{1}{A} \ln\left(\frac{C_r \ln(1+R)}{N(t_p)C_b}\right) + \frac{1}{A} A(t_p-t_0)$$

$$t^* = t_p + \frac{1}{A} \ln\left(\frac{C_r \ln(1+R)}{N(t_p)C_b}\right)$$

ANNEXE 2

CODES DE L'ALGORITHME GÉNÉTIQUE

Deux modèles ont été élaborés : Le modèle 1 (sans contraintes budgétaires) et le modèle 2 (avec des contraintes budgétaires). Ces deux modèles utilisent des fonctions communes exceptées certaines fonctions telles la fonction d'évaluation (qui permet de calculer le fitness de chaque individu), la fonction génération (qui permet d'exécuter l'algorithme de génération en génération) et la fonction compil (qui est la fonction principale d'exécution du programme). Les codes ont été écrits et compilés avec le logiciel MATLAB.

Les fonctions utilisées sont :

Pour le calcul des coûts

- 1- *budgets* : elle permet de répartir le budget global de manière homogène par année.
- 2- *cout_maint* et *cout_reempla* : elles permettent de calculer respectivement le coût de maintenance et le coût de remplacement d'une conduite à une année donnée.
- 3- *cout_tot* et *cout_tot_lin* : elles permettent de calculer les coûts totaux du système selon un modèle de dégradation exponentiel et selon un modèle linéaire.
- 4- *cout_annee* : elle permet de déterminer le budget annuel d'une solution.
- 5- *coef_R* : elle permet de définir le taux d'actualisation.

Pour les conditions initiales des conduites

6- *coef_A*, *date_pose*, *taux_def*, *cout_rep*, *cout_remp* et *cond_init_stat* : elles permettent de définir les conditions initiales selon les différents types de réseau.

7- *technique* : elle permet de définir les techniques d'intervention qui sont la réhabilitation et le remplacement et d'en fixer les coûts.

Pour le déroulement de l'algorithme

8- *pop_init* : elle permet de générer la population initiale.

9- *conv_bin_dec* et *conv_dec_bin* : elles permettent respectivement de convertir un nombre binaire en décimal et un nombre décimal en binaire.

10- *vecteur_fitness* et *vecteur_fitness_cas2* : elles permettent de calculer le coût total du système pour le modèle 1 et le modèle 2.

11- *classement* : elle permet de classer par rang les individus en fonction de leur coût total.

12- *probabilite* : elle permet de calculer la probabilité de sélection en fonction du rang.

13- *choixAB* : elle permet de sélectionner deux parents pour le croisement.

14- *croisementAB* et *croisementABu* : elles permettent de croiser les parents respectivement selon un croisement en 2 points et un croisement uniforme.

15- *mutation* : elle permet de faire l'inversion d'un bit d'un individu.

16- *generation* et *generation_cas2* : elles sont les fonctions d'exécution pour le modèle 1 et le modèle 2.

17- *compil1* et *compil2* : elles sont les fonctions principales pour le modèle 1 et le modèle 2.

1- Définition de la fonction *budgets*

La fonction *budgets* est donnée par :

```
function an_hor = budgets(argent, periode, mode)

%an_hor = zeros(1,horizon);
val = argent / periode;
switch mode

% Premier modèle : mêmes couts par annee
case 0
    for i = 1:periode
        an_hor(i) = val ;
    end

% Deuxieme modèle : différents couts par annee
case 1

end
```

2- Définition des fonctions *cout_maint* et *cout_repla*

Les fonctions *cout_maint* et *cout_repla* sont données par :

```
%-----
% Cette fonction donne les couts de maintenance de chaque conduite
%-----

function cout_unit = cout_maint(id_cond, mat_cond_init, date)

x = [ ];
```

```

x = mat_cond_init;

a = 0; % cout de maintenance actualisé

    temp = x(6,id_cond)*x(2,id_cond)*exp(x(3,id_cond)*(date+2007-x(1,id_cond)));
    a = a + temp/(1 + x(4,id_cond))^(date); %i-2007);

cout_unit = a;

%-----
% Cette fonction donne les couts de remplacement de chaque conduite
%-----

function cout_unit = cout_repla(id_cond, mat_cond_init, date)

x = [ ];
x = mat_cond_init;

b = x(5,id_cond) / (1 + x(4,id_cond))^(date); % cout de remplacement actualisé

cout_unit = b;

```

3- Définition des fonctions *cout_tot* et *cout_tot_lin*

Les fonctions *cout_tot* et *cout_tot_lin* sont données par :

```

%-----
% Cette fonction donne les couts de rehabilitation de chaque conduite
%-----

function cout_unit = cout_tot(id_cond, mat_cond_init, date)

x = [ ];
x = mat_cond_init;

a = 0; % cout de maintenance actualisé
for i = 2007:2007+date
    temp = x(6,id_cond)*x(2,id_cond)*exp(x(3,id_cond)*(i-x(1,id_cond)));
    a = a + temp/(1 + x(4,id_cond))^(i-2007);
end

```

```

b = x(5,id_cond)/(1 + x(4,id_cond))^(date); % cout de remplacement actualisé

cout_unit = a + b;

%-----
% Cette fonction donne les couts de rehabilitation de chaque conduite
%-----

function cout_unit = cout_tot_lin(id_cond, mat_cond_init, date)

x = [ ];
x = mat_cond_init;

a = 0; % cout de maintenance actualisé
for i = 2007:2007+date
    temp = x(6,id_cond)*x(2,id_cond)*x(3,id_cond)*(i-x(1,id_cond));
    a = a + temp/(1 + x(4,id_cond))^(i-2007);
end

b = x(5,id_cond)/(1 + x(4,id_cond))^(date); % cout de remplacement actualisé

cout_unit = a + b;

```

4- Définition de la fonction *cout_annee*

La fonction *cout_annee* est donnée par :

```

function an_hor = budgets(argent, periode, mode)

%an_hor = zeros(1,horizon);
val = argent / periode;
switch mode

% Premier modèle : mêmes couts par annee
case 0
    for i = 1:periode
        an_hor(i) = val ;
    end

end

```

5- Définition de la fonction *coef_R*

La fonction *coef_R* est donnée par :

```
%-----
% Cette fonction donne le taux d'actualisation
%-----

function vecteur = coef_R (Nb_cond, taux)

for i =1 :Nb_cond
    vecteur(i) = taux;
end
```

6- Définition des fonctions *coef_A*, *date_pose*, *taux_def*, *cout_rep*, *cout_remp* et *cond_init_stat*

Les fonctions *coef_A*, *date_pose*, *taux_def*, *cout_rep*, *cout_remp* et *cond_init_stat* sont données par :

```
%-----
% Cette fonction donne les coefficients A de chaque conduite
%-----

function vecteur = coef_A(Nb_cond, min, max, mode)

switch mode

% Premier modèle : mêmes coefficients A
case 0
    for i =1 :Nb_cond
        vecteur(i) = 0.05;
    end

% Deuxième modèle : différents coefficients A
case 1
    for i =1 :Nb_cond
        vecteur(i) = min + (rand*(max-min));
    end
```

```
% Troisieme modele : 5 lots de conduites
```

```
case 2
```

```
    k = Nb_cond/5;
    vecteur(1,1:k) = 0.05;
    vecteur(1,1+k:2*k) = 0.06;
    vecteur(1,1+2*k:3*k) = 0.07;
    vecteur(1,1+3*k:4*k) = 0.08;
    vecteur(1,1+4*k:5*k) = 0.9;
```

```
end
```

```
%-----
% Cette fonction donne les dates de pose de chaque conduite
%-----
```

```
function vecteur = date_pose(Nb_cond, date, mode)
```

```
switch mode
```

```
% Premier modèle : même date d'installation
```

```
case 0
```

```
    for i = 1 : Nb_cond
        vecteur(i) = date;
    end
```

```
% Deuxième modèle : differente date de pose
```

```
case 1
```

```
    for i = 1 : Nb_cond
        vecteur(i) = date + round(rand*(2007-date));
    end
```

```
% Troisieme modele : 5 lots de conduites
```

```
case 2
```

```
    k = Nb_cond/5;
    vecteur(1,1:k) = 1960;
    vecteur(1,1+k:2*k) = 1945;
    vecteur(1,1+2*k:3*k) = 1965;
    vecteur(1,1+3*k:4*k) = 1970;
    vecteur(1,1+4*k:5*k) = 2002;
```

```

end

%-----
% Cette fonction donne les taux de défaillance de chaque conduite
%-----

function vecteur = taux_def(Nb_cond, min, max, mode)

switch mode

% Premier modèle : même taux de defaillance

case 0
    for i = 1 : Nb_cond
        vecteur(i) = 0.1;
    end

% Deuxième modèle : differente date de pose

case 1
    for i = 1 : Nb_cond
        vecteur(i) = min + (rand*(max-min));
    end

% Troisième modèle : 5 lots de conduites

case 2
    k = Nb_cond/5;
    vecteur(1,1:k) = 0.1;
    vecteur(1,1+k:2*k) = 0.04;
    vecteur(1,1+2*k:3*k) = 0.03;
    vecteur(1,1+3*k:4*k) = 0.02;
    vecteur(1,1+4*k:5*k) = 0.01;

end

%-----
% Cette fonction donne les couts de maintenance de chaque conduite
%-----

function vecteur = cout_rep(Nb_cond, min, max, mode)

```

```
switch mode
```

```
% Premier modèle : mêmes couts de remplacement
```

```
case 0
```

```
    for i =1 :Nb_cond
        vecteur(i) = 1000;
    end
```

```
% Deuxième modèle : differents couts de remplacement
```

```
case 1
```

```
    for i =1 :Nb_cond
        vecteur(i) = min + round(rand*(max-min));
    end
```

```
% Troisieme modele : 5 lots de conduites
```

```
case 2
```

```
    k = Nb_cond/5;
    vecteur(1,1:k) = 1000;
    vecteur(1,1+k:2*k) = 1250;
    vecteur(1,1+2*k:3*k) = 1500;
    vecteur(1,1+3*k:4*k) = 1750;
    vecteur(1,1+4*k:5*k) = 2000;
```

```
end
```

```
%-----
% Cette fonction donne les couts de remplacement de chaque conduite
%-----
```

```
function vecteur = cout_remp(Nb_cond, min, max, mode)
```

```
switch mode
```

```
% Premier modèle : mêmes couts de remplacement
```

```
case 0
```

```
    for i =1 :Nb_cond
        vecteur(i) = 50000;
    end
```

% Deuxième modèle : différents couts de remplacement

case 1

```
for i = 1:Nb_cond
    vecteur(i) = min + round(rand*(max-min));
end
```

% Troisième modèle : 5 lots de conduites

case 2

```
k = Nb_cond/5;
vecteur(1,1:k) = 50000;
vecteur(1,1+k:2*k) = 40000;
vecteur(1,1+2*k:3*k) = 30000;
vecteur(1,1+3*k:4*k) = 20000;
vecteur(1,1+4*k:5*k) = 10000;
```

end

function matrice = cond_init_stat(Nb_cond)

global date_theo;

matrice = [];

```
matrice(1,1:Nb_cond) = date_pose(Nb_cond, 1960, 1);
matrice(2,1:Nb_cond) = taux_def(Nb_cond, 0.01, 0.2, 1);
matrice(3,1:Nb_cond) = coef_A(Nb_cond, 0.01, 0.15, 1);
matrice(4,1:Nb_cond) = coef_R(Nb_cond, 0.1);
matrice(5,1:Nb_cond) = cout_remp(Nb_cond, 10000, 150000, 1);
matrice(6,1:Nb_cond) = cout_rep(Nb_cond, 500, 2000, 1);
matrice(7,1:Nb_cond) = cout_reh(Nb_cond, 2000, 10000, 1);
```

% calcul des dates théoriques de remplacement

```
fid=fopen('dates_sh_stat.txt','w');
for i = 1:Nb_cond
    date_theo(i) = matrice(1,i) + (1/matrice(3,i))*log( (matrice(5,i)*log(1+ matrice(4,i))) /
    (matrice(6,i)*matrice(2,i)));
    if (date_theo(i) < 2007)
        date_theo(i) = 2007;
        fprintf(fid,'%d\n',ceil(date_theo(i)));
    elseif (date_theo(i) > 2038)
        date_theo(i) = 2038;
```

```

    fprintf(fid,'%d\n',ceil(date_theo(i)));
else
    date_theo(i) = ceil(date_theo(i));
    fprintf(fid,'%d\n',ceil(date_theo(i)));
end
end
fclose(fid);

```

7- Définition de la fonction *technique*

La fonction *technique* est donnée par :

```

function tech = technique(N)

matrice = zeros(1,N);
cout_repla = 50000;
alpha = 0.75;
beta = 1;
for i = 1 : N

    decision = round(rand(1,2));
    if (decision(1) == 0 & decision(2) == 0)
        matrice(5,i) = cout_repla*alpha;
    elseif (decision(1) == 0 & decision(2) == 1)
        matrice(5,i) = cout_repla*beta;
    elseif (decision(1) == 1 & decision(2) == 0)
        matrice(5,i) = cout_repla*beta;
    else
        matrice(5,i) = alpha*cout_repla;
    end

end

tech = matrice(5,:);

```

8- Définition de la fonction *pop_init*

La fonction *pop_init* est donnée par :

```

%-----
% Cette fonction génère une population initiale à M individus
%-----

```

```
function matrice = pop_init(M,N)
```

```
matrice = round(rand(M,N));
```

9- Définition des fonctions *conv_bin_dec* et *conv_dec_bin*

Les fonctions *conv_bin_dec* et *conv_dec_bin* sont données par :

```
function decimal = conv_bin_dec (binaire)
```

```
taille = size(binaire,2);
```

```
decimal = 0;
```

```
for i=1:taille
```

```
    decimal = decimal + binaire(i)*2^(taille-i);
```

```
end
```

```
function vecteur = conv_dec_bin(date)
```

```
global N
```

```
vecteur = [];
```

```
%N = size(date,2);
```

```
for i = 1:N
```

```
    binaire = zeros(1,5);
```

```
    if (date(i) == 0)
```

```
        vecteur(1,1+5*(i-1):5*i)= binaire;
```

```
    else
```

```
        for j = 1:floor(log(date(i))/log(2))+1
```

```
            binaire(5-j+1) = mod(date(i),2);
```

```
            date(i) = floor(date(i)/2);
```

```
        end
```

```
        vecteur(1,1+5*(i-1):5*i)=binaire;
```

```
    end
```

```
end
```

10- Définition de la fonction *vecteur_fitness* et *vecteur_fitness_cas2*

Les fonctions *vecteur_fitness* et *vecteur_fitness_cas2* sont données par :

```

-----
% Cette fonction renvoie le fitness de chaque individu de la population et
% vérifie si les contraintes budgétaires sont respectées
% -----
function [vecteur, champion1] = vecteur_fitness(population, mat_cond_init)

global Bglobal
global budget
global M
global N
global faisabilite
global alpha

alpha = 5;
annee = [];
cout_annee = [];
t = [];
Ct = [];
somme_Ct = [];
somme_Ct_penalite = [];
champion = [];
temp = [];

for i=1:M

    k = 1; % incrémentation pour le vecteur annee et cout_annee

    individu = population(i,:); % sélection de la solution i

    % on calcule pour la premiere conduite les valeurs

    t(i,1) = conv_bin_dec(individu(1:alpha)); % annee optimale
    Ct(i,1) = cout_tot(1,mat_cond_init,t(i,1)); % cout total
    annee(i,k) = t(i,1); % annee optimal
    cout_annee(i,k) = Ct(i,1); % cout de l'annee
    somme_Ct(i) = Ct(i,1); % budget global trouve

    if (i == 1)

```

```

        champion(1,1) = t(1,1);
        cout_cond_min(1,1) = Ct(1,1);
    else
        if (Ct(i,1) < cout_cond_min(1,1))
            champion(1,i) = t(i,1);
            cout_cond_min(1,1) = Ct(i,1);
        end
    end
end

% on travaille sur les conduites restantes de 2 a N

for j=2:N

    t(i,j) = conv_bin_dec(individu(1 + alpha*(j - 1):alpha + alpha*(j - 1))); % annee optimale
    Ct(i,j) = cout_tot(j,mat_cond_init,t(i,j)); % cout total de la conduite concerne
    somme_Ct(i) = somme_Ct(i) + Ct(i,j);

    if (i == 1)
        champion(1,j) = t(1,j);
        cout_cond_min(1,j) = Ct(1,j);
    else
        if (Ct(i,j) < cout_cond_min(1,j))
            champion(1,j) = t(i,j);
            cout_cond_min(1,j) = Ct(i,j);
        end
    end
end

temp = k; % sauvegarde de la derniere position
for m = 1:k

    if (t(i,j) == annee(i,m))

        cout_annee(i,m) = cout_annee(i,m) + Ct(i,j);
        break
    else
        if (temp == m)

            k = k + 1;
            annee(i,k) = t(i,j);
            cout_annee(i,k) = Ct(i,j);
        end
    end
end
end

```

```

        end

    end

end

vecteur = somme_Ct; %somme_Ct_penalite;
champion1 = champion;

% -----
% Cette fonction renvoie le fitness de chaque individu de la population et
% vérifie si les contraintes budgétaires sont respectées
% -----
function [vecteur, sup_i, sup_i1, vect_int] = vecteur_fitness_cas2(population,
mat_cond_init,budgetglo)

global Bglobal2
global budget
global M
global N
global faisabilite
global alpha
global horizon

alpha = 5;
annee = [];
cout_annee = [];
t = [];
C_t = [];
somme_Ct = [];
somme_Ct_penalite = [];
vect_cout = [];
temp = [];

for i=1:M

    individu = population(i,:); % sélection de la solution i

    % on calcule pour la premiere conduite les valeurs

    t(i,1) = conv_bin_dec(individu(1:alpha)); % annee optimale

    % on change de cout par conduite a chaque individu-----

```

```

mat_cond_init(5,1:N) = technique(N);
%-----

C_t(i,1) = cout_tot(1,mat_cond_init,t(i,1)); % cout total

somme_Ct(i) = C_t(i,1); % budget global trouve

% on travaille sur les conduites restantes de 2 a N

for j=2:N

    t(i,j) = conv_bin_dec(individu(1 + alpha*(j - 1):alpha + alpha*(j - 1))); % annee optimale
    C_t(i,j) = cout_tot(j,mat_cond_init,t(i,j)); % cout total de la conduite concernee
    somme_Ct(i) = somme_Ct(i) + C_t(i,j);

end % end for
%-----end for-----

%-----
for r = 1:N
    vect_date(r) = t(i,r) + 2007;
end
%-----

[ct, cout_glo] = cout_annee(N,vect_date,mat_cond_init, horizon);
cout_glo;
%-----
%temp1 = find(t == max (t));
%pas = t(temp1);
budget_an_est = budgets(budgetglo, horizon, 0);
somme_temp1 = 0;
for ind = 1:horizon
    temp_an(ind) = budget_an_est(ind)- ct(ind);
    somme_temp1 = somme_temp1 + abs(temp_an(ind));
end

% calcul de la difference entre le cout annuel estimé et le cout reel
%-----

%-----
% Budget annuel comme critere
%-----

```

```

if (i == 1)
    minimum = abs(cout_glo - budgetglo);
    sup_i = 1;
    vect_remp_reh = mat_cond_init(5,1:N); % sauvegarde des couts de rehab/rempe
else
    if (abs(cout_glo - budgetglo) < minimum)
        minimum = abs(cout_glo - budgetglo);
        sup_i = i;
        vect_remp_reh = mat_cond_init(5,1:N); % sauvegarde des couts de rehab/rempe
    end
end

%-----
% Budget annuel comme critere
%-----

if (i==1)
    minimum1 = somme_temp1;
    sup_i1 = 1;
    vect_remp_reh1 = mat_cond_init(5,1:N); % sauvegarde des couts de rehab/rempe
else
    if (somme_temp1 < minimum1)
        minimum1 = somme_temp1;
        sup_i1 = i;
        vect_remp_reh1 = mat_cond_init(5,1:N); % sauvegarde des couts de rehab/rempe
    end
end

%-----

end

vect_int = vect_remp_reh;
%*****
%vect_int = vect_remp_reh1; % <=== ATTENTION
vecteur = somme_Ct;

```

11- Définition de la fonction *classement*

La fonction *classement* est donnée par :

```

%-----

```

```

% fonction de classement : Cette fonction classe les M individus ou vecteurs
% ou points de la population en fonction de leur cout par ordre
% decroissant. Par ex.: ici on renvoie un vecteur de taille M, qui
% contient les positions de chaque point dans la matrice population.
%-----

function vecteur_indice_ordre = classement(vecteur_fitness)

taille = size(vecteur_fitness,2);

vecteur_indice_ordre = zeros(taille, 1);

% on cherche un minimum de toutes les valeurs du fitness

val_min = min(vecteur_fitness) - 1;

i = 1;

pas = 0;

while(i <= taille)

    % on cherche le tableau d'indice de position du maximum

    temp = find(vecteur_fitness == max(vecteur_fitness));

    pas = pas + size(temp,1); % on incremente le pas de la taille du tableau

    vecteur_indice_ordre(i:pas) = temp(1:pas - i + 1); % on sauvegarde les positions

    vecteur_fitness(temp) = val_min; % on affecte le min pour ne pas resélectionner les
positions

    i = pas + 1;

end

```

12- Définition de la fonction *probabilité*

La fonction *probabilité* est donnée par :

```

%-----

```

```
% assignation de la probabilité : Cette fonction calcule la probabilité
% en fonction du rang (valeur décroissante) de leur fitness
%-----
```

```
function vecteur_proba = probabilite(vecteur)
```

```
tailleM = size(vecteur,2); % trouve la taille du vecteur de probabilité
```

```
vecteur_proba = zeros(tailleM,1);
```

```
pm = 2./tailleM;
```

```
for j = 1:tailleM
```

```
    vecteur_proba(j) = pm *(j - 1)/(tailleM -1);
```

```
end
```

13- Définition de la fonction *choixAB*.

La fonction *choixAB* est donnée par :

```
%-----
% selection de A et B en fonction de leur probabilité : cette fonction
% permet de sélectionner deux parents en fonction de leur rang de
% probabilité selon la méthode de sélection par la roulette biaisée. Par
% ex.: ici on renvoie les positions (dans le vecteur de probabilité) des individus
%-----
```

```
function [indiceA, indiceB] = choixAB(vecteur_proba)
```

```
taille = size(vecteur_proba,1);
```

```
roulette = zeros(taille,1);
```

```
somme_cumulative = 0;
```

```
for i = 1:taille % obtention de la roulette
```

```
    somme_cumulative = somme_cumulative + vecteur_proba(i);
```

```
    roulette(i) = somme_cumulative;
```

```

end

% choix des indices des deux points avec la roulette au hasard

temp = [];

for i = 1:2 % pour deux points

    temp(i,1) = rand*1;

    aiguille = 1; % premiere tranche dans la roulette

    j = 1;

    while (temp(i,1) > roulette(j))

        j = j + 1;

        aiguille = j;

    end

    temp(i,1) = aiguille; % tranche ou se situe le pointeur dans la roulette

end

indiceA = temp(1,1);

indiceB = temp(2,1);

```

14- Définition des fonctions *croisementAB* et *croisementABu*

Les fonctions *croisementAB* et *croisementABu* sont données par :

```

%-----
% fonction de croisement : elle fait un croisement en deux points sur deux
% individus sélectionnés et donne un unique enfant
%-----

```

```

function enfant = croisementAB(A, B, N) % N n'est pas nécessaire

```

```

taille = size(A,2);

```

```

enfant = [];

% choix des deux points de croisement aléatoires

k1 = 1;

k2 = 1;

while (k1 == k2)

    k1 = 1 + round(rand*(taille - 1));

    k2 = 1 + round(rand*(taille - 1));

    if (k1 > k2)

        temp = k1;

        k1 = k2;

        k2 = temp;

    end

end

% croisement des parents pour donner un enfant

enfant(1,1:k1 - 1) = B(1:k1 - 1);

enfant(1,k1:k2 - 1) = A(k1:k2 - 1);

enfant(1,k2:taille) = B(k2:taille);

%-----
% fonction de croisement : elle fait un croisement en deux points sur deux
% individus sélectionnés et donne un unique enfant
%-----

function enfant = croisementABu(A, B, N)

taille = size(A,2);

```

```

enfant = [];

masque = round(rand(1,N));

for i =1:N

    if (i == 1)
        enfant(i) = A(i);
    else
        enfant(i) = B(i);
    end

end

```

15- Définition de la fonction *mutation*

La fonction *mutation* est donnée par :

```

%-----
% fonction mutation : Cette fonction opere une mutation aléatoire sur un
% individu en inversant une valeur de bit
%-----

function [individu_mute] = mutation (individu)

individu_mute = [];

longueur = size(individu,2);

point = 1 + round(rand*(longueur - 1));

% opération de mutation

if (individu(1,point) == 0)
    individu(1,point) = 1;
else individu(1, point) = 0;
end

individu_mute = individu;

```

16- Définition des fonctions *generation* et *generation_cas2*

Les fonctions *generation* et *generation_cas2* sont données par :

```
% -----
% Algorithme génétique
% -----

function [tab1, tab2] = generation(f, matrice, mat_cond_init) % f est la technique de
croisement

global Bglobal budget M N Ngen horizon date_theo
budget = []; % cout total du systeme
nb_eval = []; % nombre évaluation

pas = 1;

%pause = input('Tapez ENTRÉE');

population = matrice;

%[M,N] = size(population);

compteur = 1;

while (compteur <= Ngen) % on a 'Ngen' generations

    % recherche de l'indice du meilleur cout des solutions

    [cout, champion] = vecteur_fitness(population, mat_cond_init);

    temp = find(cout == min (cout));

    indice_minimum = temp(1); % sauvegarde de la position du meilleur individu

    % -----
    % tracé de la fonction de cout total en fonction des générations

    somme2 = 0;

    if (compteur == 1)
```

```

temp_champion = champion;
somme1 = 0;
for i = 1:N
    somme1 = somme1 + cout_tot(i,mat_cond_init,temp_champion(1,i));
end
budget(1) = somme1;
nb_eval(1) = 1;
else
    for i = 1:N
        if
(cout_tot(i,mat_cond_init,champion(1,i))<cout_tot(i,mat_cond_init,temp_champion(1,i)))
            temp_champion(1,i) = champion(1,i);
        end
        somme2 = somme2 + cout_tot(i,mat_cond_init,temp_champion(1,i));
    end
    if (compteur == pas*floor(Ngen/50))
        pas = pas + 1;
        budget(pas) = somme2;
        nb_eval(pas) = pas*floor(Ngen/50);
    end
end
end
%-----

% tableaux des evaluations en fonction des generations

tableau1(compteur) = cout(indice_minimum); % minimum

% on travaille sur la valeur optimale
cout_horizon = 0;
for i = 1:N

    temp2 = population(indice_minimum,:);
    an(i) = conv_bin_dec(temp2(1 + 5*(i - 1):5 + 5*(i - 1)));
    cout_horizon = cout_horizon + Cm(an(i)) + Cr(an(i));

end
tableau2(compteur) = cout_horizon;

% on classe dans un vecteur les indices de la plus grande valeur a la
% plus petite de la fonction objective

vecteur_indice_ordre = classement(cout);

```

```

% on calcule la probabilité de la plus grande valeur a la plus petite

vecteur_proba = probabilite(cout);

%===== sélection, croisement, population suivante =====

for i = 1:M

    [indA, indB] = choixAB(vecteur_proba);

    A = population(vecteur_indice_ordre(indA,1),:);

    B = population(vecteur_indice_ordre(indB,1),:);

    enfant = feval(f, A, B, 5*N);

    nouvelle_population(i,:) = enfant;

end

%===== fin croisement =====

%===== on effectue la mutation =====

% nombre d'individus a muter

nombre = round(0.01*N*10*M);

for i = 1:nombre

    id = 1 + round(rand*(M - 1)); % indice de l'individu qui mute

    individu_mute = mutation(nouvelle_population(id,:));

    nouvelle_population(id,:) = individu_mute;

end

%===== fin mutation =====

%===== conservation du meilleur gene ou elitisme =====

```

```

    indice1 = 1 + round(rand*(M - 1));

    nouvelle_population(indice1,:) = population(indice_minimum,:);

    indice2 = 1 + round(rand*(M - 1));

    while (indice2 == indice1)
        indice2 = 1 + round(rand*(M - 1));
    end

    nouvelle_population(indice2,:) = conv_dec_bin(temp_champion);

    population = nouvelle_population;

    %===== fin conservation =====

    % incrementation du compteur

    compteur = compteur + 1;

end

% on travaille sur la valeur optimale pour trouver les temps optimaux des
% conduites

for i = 1:N

    temp = nouvelle_population(indice1,:);
    an(i) = 2007 + conv_bin_dec(temp(1 + 5*(i - 1):5 + 5*(i - 1)));

end

an

[cout_an, cout_global] = cout_annee(N,an,mat_cond_init,horizon);
cout_global
temp = find(an == max (an));
date_max = an(temp(1));

%calcul budget global-----
somme3 = 0;
somme4 = 0;
for i = 1:N

```

```

        somme3 = somme3 + cout_tot(i,mat_cond_init,date_theo(i)-2007);
        somme4 = somme4 + cout_tot(i,mat_cond_init,an(i)-2007);
    end
    Bglobal_theo = somme3;
    Bglobal = somme4; %<== attention
    %-----

    subplot(3,1,2)
    hold on
    %plot(nb_eval,budget1,'c');
    %plot(nb_eval,budget,'g');

    %plot(nb_eval,Bglobal, 'g -')
    plot(nb_eval,Bglobal_theo, 'r *')
    tab1 = [nb_eval;budget]; % couts
    tab2 = an; % dates
    title('Évolution du cout total en $', 'color', 'blue', 'Linewidth', 2);
    xlabel('Generation', 'color', 'blue', 'Linewidth', 2);
    ylabel('Cout ($)', 'color', 'blue', 'Linewidth', 2);

    subplot(3,1,3)
    hold on

    axes_bar = 2007:1:date_max;
    bar(cout_an);
    plot(cout_global);

    % -----
    % Algorithme génétique
    %-----

function [tab1, tab2] = generation_cas2(f, matrice, mat_cond_init, budgetglo) % f est la
technique de croisement

global Bglobal2 Bglobal budget M N Ngen horizon date_theo
budget = []; % cout total du systeme
nb_eval = []; % nombre évaluation
temp_cout_int = [];
%nouvelle_population = [M,5*N];
pas = 1;

%pause = input('Tapez ENTRÉE');

```

```

population = matrice;

%[M,N] = size(population);

compteur = 1;

while (compteur <= Ngen) % on a 'Ngen' generations

    % recherche de l'indice du meilleur cout des solutions

    [cout, entier1, entier2, cout_int] = vecteur_fitness_cas2(population,
mat_cond_init,budgetglo);

    indice_minimum1 = entier1; % sauvegarde de la position du meilleur individu
    indice_minimum2 = entier2; % sauvegarde de la position du meilleur individu

    %-----sauvegarde de la meilleure solution (budget global) -----
    if (compteur == 1)
        temp_indice_minimum1 = entier1;
        temp_cout_indice_minimum1 = cout(temp_indice_minimum1);
        temp_cout_int1 = cout_int;
        %budget(1) = cout(temp_indice_minimum1); % <=== ATTENTION
        %*****
        nb_eval(1) = 1;
    else
        if (abs(cout(entier1) - budgetglo) < abs( temp_cout_indice_minimum1 - budgetglo))
            temp_indice_minimum1 = entier1;
            temp_cout_indice_minimum1 = cout(temp_indice_minimum1);
            temp_cout_int1 = cout_int;
        end
    end

    %-----sauvegarde de la meilleure solution (budget annuel)-----
    if (compteur == 1)
        temp_indice_minimum2 = entier2;
        temp_cout_int2 = cout_int;
        budget(1) = cout(temp_indice_minimum2);% <=== ATTENTION
        %*****
        tempor = population(entier2,:);
        for i = 1:N
            vect_date(i) = 2007 + conv_bin_dec(tempor(1 + 5*(i - 1):5 + 5*(i - 1)));
        end
    end
end

```

```

mat_cond_init(5,1:N) = temp_cout_int2;
[ct, ct_glo] = cout_annee(N,vect_date,mat_cond_init,horizon);
somme_tempor = 0;
budget_an_est = budgets(budgetglo, horizon, 0);
for i = 1:horizon
    somme_tempor = somme_tempor + abs(ct(i)-budget_an_est(i));
end
temp_somme_tempor = somme_tempor;

else
    tempor = population(entier2,:);
    for i = 1:N
        vect_date(i) = 2007 + conv_bin_dec(tempor(1 + 5*(i - 1):5 + 5*(i - 1)));
    end
    mat_cond_init(5,1:N) = cout_int;
    [ct, ct_glo] = cout_annee(N,vect_date,mat_cond_init,horizon);
    somme_tempor = 0;
    budget_an_est = budgets(budgetglo, horizon, 0);
    for i = 1:horizon
        somme_tempor = somme_tempor + abs(ct(i)-budget_an_est(i));
    end
    if (somme_tempor < temp_somme_tempor)
        temp_somme_tempor = somme_tempor;
        temp_indice_minimum2 = entier2;
        temp_cout_int2 = cout_int;
    end

end

%-----

% tracé de la fonction de cout total en fonction des générations

if (compteur == pas*floor(Ngen/50))
    pas = pas + 1;
    %budget(pas) = temp_cout_indice_minimum1;
    %*****
    budget(pas) = cout(temp_indice_minimum2); % <== ATTENTION
    nb_eval(pas) = pas*floor(Ngen/50);
end

%-----

```

```

% on classe dans un vecteur les indices de la plus grande valeur a la
% plus petite de la fonction objective

vecteur_indice_ordre = classement(cout);

% on calcule la probabilité de la plus grande valeur a la plus petite

vecteur_proba = probabilite(cout);

%===== sélection, croisement, population suivante =====

for i = 1:M

    [indA, indB] = choixAB(vecteur_proba);

    A = population(vecteur_indice_ordre(indA,1),:);

    B = population(vecteur_indice_ordre(indB,1),:);

    enfant = feval(f, A, B, 5*N);

    nouvelle_population(i,:) = enfant;

end

%===== fin croisement =====

%===== on effectue la mutation =====

% nombre d'individus a muter

nombre = round(0.01*N*10*M);

for i = 1:nombre

    id = 1 + round(rand*(M - 1)); % indice de l'individu qui mute

    individu_mute = mutation(nouvelle_population(id,:));

    nouvelle_population(id,:) = individu_mute;

end

```

```

%===== fin mutation =====

%===== conservation =====

indice1 = 1 + round(rand*(M - 1));

%nouvelle_population(indice1,:) = population(temp_indice_minimum1,:);
%*****
nouvelle_population(indice1,:) = population(temp_indice_minimum2,:);% <===
ATTENTION

population = nouvelle_population;

%===== fin conservation =====

% incrementation du compteur

compteur = compteur + 1;

end

% on travaille sur la valeur optimale pour trouver les temps optimaux des
% conduites

for i = 1:N

    %temp = nouvelle_population(temp_indice_minimum1,:);
    %*****
    temp = nouvelle_population(temp_indice_minimum2,:); % <=== ATTENTION
    an(i) = 2007 + conv_bin_dec(temp(1 + 5*(i - 1):5 + 5*(i - 1)));

end

% graphique du budget annuel
an

% temp_cout_int1 % <=== ATTENTION
%*****
temp_cout_int2

%mat_cond_init(5,1:N) = temp_cout_int1;% <=== ATTENTION

```

```

%*****
mat_cond_init(5,1:N) = temp_cout_int2;

[cout_an, cout_global] = cout_annee(N,an,mat_cond_init,horizon);
cout_an
cout_global

%calcul budget global-----
somme3 = 0;
for i = 1:N
    somme3 = somme3 + cout_tot(i,mat_cond_init,an(i)-2007);
end
Bglobal2 = somme3;
%-----

tab1 = budget; % couts
tab2 = an; % dates

subplot(3,1,2)
hold on

tab1 = [nb_eval;budget];% couts
%plot(nb_eval,budget,'c');
plot(nb_eval,budgetglo, 'r +');

title('Évolution du cout total en $', 'color', 'blue','Linewidth', 2);
xlabel('Generation', 'color', 'blue','Linewidth', 2);
ylabel('Cout ($)', 'color', 'blue','Linewidth', 2);

subplot(3,1,3)
hold on

axes_bar = 2007:1:horizon-1+2007;
bar(cout_an,0.4,'c');
plot(cout_global);

grid on

title('Couts annuels ($) sur 32 ans', 'color', 'blue','Linewidth', 2);
xlabel('Année', 'color', 'blue','Linewidth', 2);
ylabel('Cout ($)', 'color', 'blue','Linewidth', 2);

```

17- Définition des fonctions *compil1* et *compil2*

Les fonctions *compil1* et *compil2* sont données par :

```
function [val_pop, val_mat, Budgetglo] = compil1()

% comparaison des croisements

close all;

clear all

% generation de la population initiale

global Bglobal budget M N Ngen horizon date_theo

M = input('Entrer le nombre d individus de la population :');
N = input('Entrer le nombre de conduites :');
Ngen = input('Entrer le nombre maximal de génération :');
horizon = 32;

population = pop_init(M,5*N);
mat_cond_init = cond_init_stat(N);

f1 = @croisementAB;
[tab10, tab11] = generation(f1,population, mat_cond_init);
%pause = input('Tapez ENTRÉE');
%f2 = @croisementABu;
 %[tab20, tab21] = generation(f2, population, mat_cond_init);
 %pause = input('Tapez ENTRÉE');
 %f3 = @croisementAB;
 %[tab30, tab31] = generation(f3, population, mat_cond_init);

subplot(3,1,1)

hold on

plot(tab11, 'b o');
%plot(tab21, 'b *');
%plot(tab31, 'b +');
plot(date_theo, 'r o'); %<==
```

```

grid on
title('Date optimale de remplacement', 'color', 'blue', 'Linewidth', 2);
xlabel('Conduite', 'color', 'blue', 'Linewidth', 2);
ylabel('Date', 'color', 'blue', 'Linewidth', 2);

subplot(3,1,2)

hold on

tab100 = tab10(1,:);
tab101 = tab10(2,:);

plot(tab100,tab101, 'o-b');

%tab200 = tab20(1,:);
%tab201 = tab20(2,:);
%plot(tab200,tab201, '*-b');

grid on
title('Évolution du cout total en $', 'color', 'blue', 'Linewidth', 2);
xlabel('Generation', 'color', 'blue', 'Linewidth', 2);
ylabel('Cout ($)', 'color', 'blue', 'Linewidth', 2);

%plot(tab30, '+-b');

%-----entrees -----

val_pop = population;
val_mat = mat_cond_init;
Budgetglo = Bglobal;
%-----

% fonction maitresse compil2-----

close all;

clear all

% generation de la population initiale

global Bglobal2 budget M N Ngen horizon date_theo

```

```

[population,mat_cond_init,Bglo] = compil1;
disp(Bglo);

Bglobal2 = input('Entrer le budget global :');

pause = input(' Tapez ENTRÉE');

mat_cond_init(5,:);
f1 = @croisementABu;
[tab10, tab11] = generation_cas2(f1,population, mat_cond_init,Bglobal2);

%pause = input(' Tapez ENTRÉE');
%f2 = @croisementAB;
 %[tab20, tab21] = generation_cas2(f2, population, mat_cond_init,Bglobal2);

%pause = input(' Tapez ENTRÉE');
%f3 = @croisementABu;
 %[tab30, tab31] = generation_cas2(f3, population, mat_cond_init,Bglobal2);

subplot(3,1,1)

hold on

plot(tab11, 'g *');
%plot(tab21, 'b *');
%plot(tab31, 'b +');
%plot(date_theo, 'r o'); %<==

grid on
title('Date optimale de remplacement', 'color', 'blue', 'Linewidth', 2);
xlabel('Conduite', 'color', 'blue', 'Linewidth', 2);
ylabel('Date', 'color', 'blue', 'Linewidth', 2);

subplot(3,1,2)

hold on

tab100 = tab10(1,:);
tab101 = tab10(2,:);

plot(tab100,tab101, '*-g');

```

```
%tab200 = tab20(1,:);  
%tab201 = tab20(2,:);  
%plot(tab200,tab201, '*-b');
```

```
grid on  
title('Évolution du cout total en $', 'color', 'blue', 'Linewidth', 2);  
xlabel('Generation', 'color', 'blue', 'Linewidth', 2);  
ylabel('Cout ($)', 'color', 'blue', 'Linewidth', 2);
```

```
%tab300 = tab30(1,:);  
%tab301 = tab30(2,:);  
%plot(tab300,tab301, '*-b');
```

```
grid on  
title('Évolution du cout total en $', 'color', 'blue', 'Linewidth', 2);  
xlabel('Generation', 'color', 'blue', 'Linewidth', 2);  
ylabel('Cout ($)', 'color', 'blue', 'Linewidth', 2);
```

18- Exemple d'exécution du programme.

Les fichiers suivant sont créés lors du déroulement du programme : dates_sh_stat.txt (qui enregistre les dates théoriques) et histo.txt (qui enregistre les coûts annuels).

Voici les courbes obtenues par les simulations :

