

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université du 20 Août 1955 Skikda

N° d'ordre : D 012117015 D



Faculté des sciences de la technologie
Département de génie civil

Thème

**IDENTIFICATION DES PARAMÈTRES MÉCANIQUES ET
DYNAMIQUES SOUS SOLLICITATIONS COMPLEXES
PAR LES MÉTHODES INVERSES**

THÈSE

Présentée en vue de l'obtention du diplôme de
Docteur en Sciences

Discipline : Génie civil

Option : Géotechnique

Présentée par : Goudjil Kamel

Directeur de Thèse : Pr. SBARTAI Badreddine Université

Badji Mokhtar Annaba

Devant le jury composé de :

Jury	Grade	Affiliation	Qualité
Messast Salah	Professeur	Université de Skikda	Président
Sbartai Badreddine	Professeur	Université d'Annaba	Rapporteur
Bouabaz Mohamed	Professeur	Université de Skikda	Examineur
Meksaouine Mohamed	Professeur	Université d'Annaba	Examineur

Dédicaces

**À Mes parents
Ma femme et
Mes deux fils
Iyed et
Mouaïd Billah**

Remerciements

En premier lieu, je tiens à remercier mon directeur de thèse, le professeur Sbartai Badreddine pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses multiples conseils et pour le temps qu'il a consacré à diriger cette recherche.

Je suis heureux de pouvoir exprimer ici mes remerciements à chacun des membres du jury qui ont bien accepté d'évaluer ce travail. Je remercie donc tout d'abord le professeur Messast Salah d'avoir accepté de présider le jury de thèse. Je tiens à l'assurer de ma profonde reconnaissance pour l'intérêt qu'il porte à ce travail. Je remercie le professeur Bouabaz Mohamed chef de département de génie civil université de Skikda, pour l'honneur qu'il m'a fait d'accepter d'être examinateur de mon travail, pour le temps consacré à la lecture de cette thèse. Je tiens à remercier le professeur Meksaouine Mohamed, pour l'honneur qu'il m'a fait pour sa participation à mon jury de thèse en qualité d'examineur et pour le temps qu'il a consacré à la lecture de cette thèse.

Je remercie mon ami Khaled Khelil, professeur à l'université de Souk Ahras pour son soutien et ses conseils.

Je remercie mes parents et tous les membres de la famille pour leurs soutiens qui m'ont été bien utiles durant ma thèse.

Résumé : - Plusieurs études ont appliquées les méthodes métaheuristiques telles que les réseaux neurones(ANN), l'algorithme génétique et l'algorithme d'optimisation par essaim de particules dans la prédiction et l'optimisation des paramètres du sol. Le but de ce travail est de déterminer les valeurs de la vitesse de cisaillement du sol par optimisation en utilisant l'algorithme génétique NSGA-II et l'algorithme multi-objectif par essaim de particules à partir d'un tassement post-liquéfaction. Les résultats obtenus montrent que les deux méthodes utilisées ont réussi d'optimiser les valeurs de Vs avec des erreurs très acceptables.

Mots clé : algorithme génétique multi-objectif NSGA-II, l'algorithme d'optimisation multi-objectif par essaim de particules (MOPSO), liquéfaction, vitesse de cisaillement du sol Vs, tassement.

الخلاصة: طبقت عدة دراسات طرق (métaheuristiques) مثل شبكات الخلايا العصبية (ANN) والخوارزميات الجينية (GA) وطريقة التحسين بواسطة سرب الجسيمات (PSO) للتنبؤ وتحسين معالم الجيوتقنية للتربة. الهدف من هذا العمل هو تحديد قيم سرعة موجة القص (Vs) عن طريق التحسين باستخدام الخوارزمية الجينية (NSGA-II) وخوارزمية سرب الجسيمات متعددة الأهداف (MOPSO) من تسوية ما بعد التسييل. أظهرت النتائج التي تم الحصول عليها أن الطريقتين المستخدمتين في هذا العمل نجحت في تحسين قيم سرعة موجة القص (Vs) بأخطاء مقبولة جدا.

الكلمات المفتاحية: الخوارزمية الجينية (NSGA II)، طريقة التحسين بواسطة سرب جسيمات متعدد الأهداف (MOPSO)، التسييل، سرعة موجة القص (Vs) وتسوية التربة

Abstract: - Several studies applied metaheuristic methods such as neuronal networks (ANN), genetic algorithms (GA) and particle swarm optimization (PSO) to predict and optimize soil parameters. The aim of this study is to optimize shear wave velocity of soil using genetic algorithm NSGAII and multi-objective particle swarm optimization (MOPSO) from settlement post-liquefaction. The results show that our methods have successfully employed to optimize the shear wave velocity (vs) with acceptable test errors.

Key words: Genetic Algorithm (NSGA II), Multi-Objective Particle Swarm Optimization Algorithm (MOPSO), liquefaction, Shear Wave Velocity and Settlement.

Table des matières

Introduction générale	1
1. Étude bibliographique	4
1.1. Introduction	4
1.2. Applications des algorithmes d'intelligence artificielle en génie civil.....	4
1.2.1. Comportement des fondations (capacité portante, tassement, ..., etc)	4
1.2.2. Les modèles constitutifs	5
1.2.3. Propriétés géotechniques des sols et des roches.....	5
1.2.4. La liquéfaction.....	5
1.2.5. Analyse de la stabilité des talus.....	6
1.2.6. Domaine de l'interaction sol structure	6
1.2.7. Optimisation des constructions et ouvrages géotechnique.....	6
1.3. Conclusion	7
2. Méthodes d'optimisation	9
2.1. Introduction	9
2.2. Problème d'optimisation.....	9
2.3. Métaheuristiques.....	12
2.4. Métaheuristiques d'optimisation mono-objectif.....	14
2.4.1. Algorithme génétiques	14
2.4.2. Méthode de recuit simulé (Simulated annealing).....	17
2.4.3. Méthode de recherche avec tabous (Tabu Search).....	19
2.4.4. Algorithmes de colonies de fourmis.....	20
2.4.5. Méthode d'essaim de particules (Particles Swarm Optimization)	21
2.5. Métaheuristiques d'optimisation multi-objectif	25
2.5.1. Concepts d'optimisation multi-objectif	26
2.5.2. Méthode VEGA (Vector Evolutionary Genetic Algorithm)	27
2.5.3. Méthode lexicographique	28
2.5.4. Méthodes basées sur l'approche de Pareto.....	29
2.5.5. Méthode MOGA (Multiple Objective Genetic Algorithm)	29
2.5.6. Méthode NSGA (Non dominated Sorting Genetic Algorithm)	31
2.5.7. Méthode NPGA (Niche Pareto Genetic Algorithm)	33
2.5.8. Méthode PAES (Pareto Archived Evolution Strategy)	33
2.5.9. Méthode SPEA (Strength Pareto Evolutionary Algorithm).....	35
2.5.10. Méthode NSGA II (Non-Dominated Sorting Genetic Algorithm II).....	37
2.5.11. Méthode MOPSO (Multiobjective Particles Swarm Optimization)	40

2.6.	Conclusion	46
3.	Évaluation du potentiel de liquéfaction des sols	47
3.1.	Introduction	47
3.2.	Explication physique du phénomène de liquéfaction	47
3.3.	Sols pulvérulents lâches.....	49
3.4.	Sols pulvérulents denses	50
3.5.	Caractéristiques influençant la résistance à la liquéfaction	51
3.5.1.	Nature du sol	51
3.5.2.	État du sol et des contraintes	53
3.5.3.	Nature de la sollicitation sismique	54
3.5.4.	Niveau de la nappe phréatique	54
3.6.	Méthodes d'évaluation du risque de liquéfaction.....	55
3.6.1.	Approche par contraintes cycliques (Seed et Idriss, 1971)	55
3.6.2.	Évaluation de CRR par les méthodes simplifiées basées sur l'essai SPT	61
3.6.3.	Méthodes d'évaluation de CRR basées sur l'essai CPT.....	65
3.6.4.	Méthodes d'évaluation de CRR basées sur la vitesse de cisaillement (Vs).....	68
3.6.5.	Évaluation du CRR par l'essai de dilatomètre de Marchetti (DMT)	71
3.7.	Conséquences de la liquéfaction.....	74
3.7.1.	Tassement post-liquéfaction.....	74
3.7.2.	Méthodes d'évaluation de tassement post-liquéfaction (S).....	75
3.8.	Conclusion	79
4.	Application des méthodes d'optimisation pour identifier Vs	80
4.1.	Introduction	80
4.2.	Description du site	80
4.3.	Formulation du problème	81
4.3.1.	Problème d'optimisation mono-objectif 1.....	81
4.3.2.	Problème d'optimisation mono-objectif 2.....	85
4.3.3.	Problème d'optimisation multi-objectif	88
4.4.	Conclusion	102
	Conclusion générale	103
	Références bibliographiques.....	105

Liste des figures

Figure 2.1- Espace de décision et objectifs dans POM (Talbi, 2009)	11
Figure 2.2-Illustration du croisement d'un point.	16
Figure 2.3-Principe de fonctionnement d'AG	17
Figure 2.4 -Analogie physique de recuit simulé.	18
Figure 2. 5 - Fonction de probabilité pour le processus de recuit	19
Figure 2.6 – Faculté d'une colonie de fourmis pour trouver le chemin le plus court, fortuitement	21
Figure 2.7- Principe de fonctionnement de PSO (Talbi, 2009).....	22
Figure 2.8 – Différents types de voisinage associé aux particules (Talbi, 2009).....	23
Figure 2.9- Mouvement d'une particule et la mise à jour de sa vitesse (Talbi, 2009).....	24
Figure 2.10- Concepts principaux de l'approche avec Pareto.....	26
Figure 2.11- Principe de fonctionnement de VEGA (Deb, 2001).....	27
Figure 2.12- Principe lexicographique dans POM (Talbi, 2009).....	28
Figure 2.1- a) Exemple d'un problème de minimisation à deux objectifs.....	30
Figure 2. 2- Principe de tri en front (NSGA) (Deb K. , 2001).....	32
Figure 2. 3-Illustration graphique de la grille adaptative utilisée par PAES (Coello et al, 2007).....	34
Figure 2.4- Schéma de la procédure NSGA-II (Deb et al., 2000).....	38
Figure 2.5- Le concept de distance (Deb et al., 2000).....	39
Figure 2.6- Stratégie de maintien de la diversité (Talbi, 2009).....	42
Figure 2.7- Principe de l'archive contrôleur (Coello et al, 2004).....	46
Figure 3.1- Simulation en laboratoire de l'effet d'un séisme sur une couche de sol.....	48
Figure 3.2- Comportement d'un sol dans un état lâche et dense lorsque soumis à un	49
Figure 3. 3- Résultats des essais triaxiaux sur un sable à l'état lâche	50
Figure 3.4- Résultats des essais triaxiaux sur un sable à l'état dense	53
Figure 3.5- Éventail des granulométries pour les sols susceptibles à la liquéfaction.	54
Figure 3.6- Influence de la contrainte horizontale effective sur le nombre de cycles nécessaire afin d'atteindre la rupture.....	55
Figure 3.7- Déformation volumétrique en fonction de la déformation axiale pour des essais.	58
Figure 3. 8- Variation du coefficient réducteur de contrainte en fonction de la magnitude	59
Figure 3.9- Différentes courbes de facteur d'échelle de magnitude MSF.	61
Figure 3.10- Variation du facteur de contrainte statique de cisaillement $K \propto$	64

Figure 3.11- Variation de CRR en fonction de $(N_1)_{60}$	65
Figure 3.12- Variation de CRR en fonction de $(N_1)_{60}$	70
Figure 3.13- Différentes formules de CRR en fonction de $(V_s)_1$	71
Figure 3. 14- Courbe simplifiée pour le calcul du CRR basée sur VS pour Mw=7.5 (Andrus et Stokoe, 2000)	73
Figure 3. 15- Variation de CRR en fonction de KD (Monaco et al. 2005)	74
Figure 3.16- Variation de CRR en fonction KD pour un sable propre.....	74
Figure 3.17- Variation de CRR en fonction ED pour un sable propre.....	75
Figure 3.18- Tassement post-liquéfaction autour de la pile du viaduc dans le Port Island après le séisme de Kobe de 1995 (Idriss et Boulanger, 2008).....	75
Figure 3. 19- Variation de γ_{max} en fonction de εV (Ishihara and Yoshimine 1992).	78
Figure 3. 20- Variation de FS en fonction de γ_{max} (Ishihara and Yoshimine 1992).	78
Figure 3. 21- Variation de FS en fonction de εV (Idriss et Boulanger, 2008).	79
Figure 4.1- la route d'accès de la plage de Moss Landing (Idriss et Boulanger, 2008).	81
Figure 4.2-Vue aérienne de Moss Landing State Beach.	82
Figure 4.3- Procédure suivie dans l'optimisation mono-objectif par l'algorithme génétique .	83
Figure 4.4- Variation de la fonction fitness en fonction de nombre de génération.....	84
Figure 4.1- Évolution de l'algorithme génétique 2 en fonction du nombre de génération.	88
Figure 4.6- Corrélation entre les valeurs cibles et optimisées de Vs.	89
Figure 4.7- Processus d'optimisation suivi par l'algorithme NSGA-II.....	92
Figure 4.8- Front de Pareto	93
Figure 4.9- L'évolution de la population en 2 ^{ème} génération	94
Figure 4.10- L'évolution de la population en 5 ^{ème} génération	94
Figure 4.11- L'évolution de la population en 15 ^{ème} génération	95
Figure 4.12- L'évolution de la population en 600 ^{ème} génération	95
Figure 4.13- Corrélation entre les valeurs réelles et valeurs optimisées	96
Figure 4.14- Front de Pareto pour le cas de Vs en fonction de $\sigma v'$	97
Figure 4.15- Corrélation entre les valeurs cibles et optimisées par NSGA II (2 ^{ème} cas).....	98
Figure 4.16- Méthode d'Optimisation Multiobjectifs par Essaim de Particules.....	100
Figure 4.17- Évolution de la population vers le front de Pareto par MOPSO (1 ^{er} cas).....	101
Figure 4.18- Corrélation entre les valeurs réelles et valeurs optimisées	102
Figure 4.19- Évolution de la population vers le front de Pareto par MOPSO (2 ^{ème} cas).	102
Figure 4.20- Corrélation entre les valeurs cibles et les valeurs optimisées par MOPSO.....	103

Listes des tableaux

Tableau 2.1- Classification des méthodes métaheuristiques	13
Tableau 3.1- Facteurs de corrections de la résistance de pénétration de SPT	63
Tableau 3.2- Valeurs critiques de la résistance de pointe q_0 selon la méthode chinoise	67
Tableau 3.3- Détermination de C_p selon le type de sol et son indice de plasticité.....	68
Tableau 4.1- Valeurs des paramètres de l'algorithme génétique	84
Tableau 4.2- Comparaison entre valeurs cibles et optimisées de (V_s).....	85
Tableau 4.3- Variantes obtenus par algorithme génétique mono-objectif ga.....	86
Tableau 4.4- Valeurs du tassement dans chaque point.....	87
Tableau 4.5- Corrélation entre les valeurs cibles et optimisées de V_s	89
Tableau 4.6- Formules reliant V_s aux paramètres géotechniques.....	90
Tableau 4.7- Paramètres de NSGA-II	91
Tableau 4.8- Comparaison entre les valeurs cibles et optimisées par NSGA II.	96
Tableau 4.9- Corrélation entre valeurs cibles et optimisées de V_s	98
Tableau 4.10- Paramètres du problème et de MOPSO	99
Tableau 4.11- Comparaison entre valeurs cibles et optimisées par MOPSO.....	101
Tableau 4.12- Comparaison entre valeurs cibles et optimisées par MOPSO (2 ^{ème} cas).....	103

Liste des symboles et abréviations

ANN	Artificial neuronal network
ACO	Ant colony optimization
PSO	Particle swarm optimization
GA	Genetic algorithm
NSGA-II	Non-Sorting Genetic Algorithm
MOPSO	Multi-Objective Particles Swarm Optimization
TDM	Tuned Mass Damper
CPT	Cone penetration test
SPT	Standard penetration test
BPT	Becker penetration test
ANN	Artificial neuronal network
ACO	Ant colony optimization
PSO	Particle swarm optimization
GA	Genetic algorithm
SA	simulated annealing
TS	taboo search
HS	Harmony strategy
POM	Problème d'optimisation multi-objectif
VEGA	Vector Evaluated Genetic Algorithm
MOGA	Multiple Objective Genetic Algorithm
NPGA	Niched Pareto Genetic Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
PAES	Pareto Archived Evolution Strategy
EV	Evolutionary algorithms
SPEA	Strength Pareto Evolutionary Algorithm
CSR	Rapport de contrainte cyclique
CRR	Rapport de résistance du sol à la liquéfaction
SPT	Essai de pénétration standard

CPT	Essai de pénétration du cône
V_s	Vitesse de l'onde de cisaillement
DMT	Essai dilatomètre de Marchetti
ID	Indice de densité
ru	Rapport de pression interstitielle
σ'_{3c}	Contrainte de confinement effective initiale
w_L	Limite de liquidité
w_N	Teneur en eau
IL	Indice de liquidité
K_o	Coefficient de pression des terres au repos
OCR	Rapport de surconsolidation
τ_{cyc}	Contrainte de cisaillement cyclique.
a_{max}	Amplitude maximale de l'accélération horizontale au niveau de la surface du sol.
σ_{v0}	Contrainte verticale totale
σ'_{v0}	Contrainte verticale effective
r_d	Coefficient réducteur des contraintes avec la profondeur.
g	Constante gravitationnelle (m/s ²)
MSF	Facteur d'échelle de magnitude
$CRR_{7.5}$	Rapport de résistance cyclique pour une magnitude $M=7.5$.
K_σ	Coefficient de correction de la contrainte de cisaillement statique initiale
K_α	Coefficient de correction de la contrainte de confinement verticale effective.
C_σ	Facteur de correction
P_a	Pression atmosphérique égale à 100 kpa.
D_r	Densité relative du sol
K_α	Facteur de contrainte statique de cisaillement
N_m	Résistance à la pénétration par mesure standard ;
C_N	Correction de la contrainte de sols sus-jacents ;
C_E	Correction du taux d'énergie de la masse ;
C_B	Correction du diamètre du forage ;
C_R	Correction de la longueur du train de tiges ;

C_s	Correction du carottier avec ou sans revêtement.
q_c	Résistance de pointe
q_{c1N}	Résistance de pointe corrigée
$(q_{c1N})_{cs}$	Résistance de pénétration du CPT normalisée équivalente d'un sable propre
N	Nombre des coups non corrigés adoptés dans le SPT
$(N_1)_{60}$	Valeur corrigée de SPT-N
$(N_1)_{60cs}$	Résistance de pénétration du SPT normalisée équivalente d'un sable propre
R_f	Rapport de résistance de frottement en pourcentage
d_w	Niveau d'eau.
d_u	Profondeur du sol considéré
V_{S1}^*	Limite supérieure de V_{S1} pour que la liquéfaction se produise
K_{cs}	Facteur de correction de content des fines,
K_{a1}	Facteur d'âge pour corriger les grandes valeurs de V_{S1} Provoquées par le vieillissement.
K_D	Indice de contrainte horizontale
E_D	Module dilatomètre
$(V_{S1})_{csa1}$	Vitesse d'onde de cisaillement corrigé
S_i	Tassement au point i
R^2	Coefficient de détermination et
MAPE	Pourcentage d'erreur absolue moyenne
$CRR_{7.5}$	Rapport de résistance cyclique pour une magnitude $M=7.5$.
K_σ	Coefficient de correction de la contrainte de cisaillement statique initiale
K_α	Coefficient de correction de la contrainte de confinement verticale effective.

Introduction générale

Le calcul des ouvrages géotechniques se contraint parfois par le manque des données telles que les propriétés physiques ou mécaniques des sols, c'est pourquoi plusieurs chercheurs ont pensé à l'analyse inverse en utilisant les méthodes d'intelligence artificielle (IA) telles que les algorithmes génétiques, méthode d'optimisation des particules d'essaim et les réseaux de neurones qui sont connues par leurs efficacités dans la prédiction et l'optimisation des paramètres du sol. Ces méthodes peuvent être couplées aux méthodes numériques ou analytiques. Par exemple, Papon (2010) a optimisé l'angle de frottement, la cohésion et le module d'élasticité à partir des essais pressiométriques réalisés in situ afin de prédire le tassement d'une fondation superficielle. Levasseur et al (2008) ont optimisé le module de cisaillement ainsi que l'angle de frottement du sol, pour ce faire, les auteurs ont utilisé l'algorithme génétique couplé à la méthode des éléments finis. Sur le même principe, Mendjel et Messast (2009) ont identifié le coefficient de perméabilité. D'autres chercheurs, vu la complexité du comportement des fondations dans les sols qui est incertain et pas encore entièrement compris, ont appliqué les techniques de l'intelligence artificielle (IA) pour prédire le comportement des fondations. Par exemple, les ANNs et ses méthodes hybrides (PSO-Based ANN, GA-Based ANN) ont été largement utilisées pour modéliser les capacités portante sous une charge axiale et latérale des fondations sur pieux en compression et en soulèvement, y compris les pieux battus (Ahmad et al 2007 ; Ardalan et al. 2009 ; Das et Basudhar, 2006 ; Pal et Deswal, 2008 ; Shahin, 2010), puits foncés (Goh et al. 2005 ; Shahin, 2010). Les ANN ont également été utilisés pour prédire le tassement des pieux (Alkroosh et Nikraz, 2011b ; Ismail and Jeng, 2011 ; Pooya Nejad et al., 2009). D'autres chercheurs ont utilisé les techniques de AI pour trouver la conception optimale des dispositifs TDM (Tuned Mass Damper) qui sert à diminuer les vibrations provoquées par les tremblements de terre sur les structures. Parmi lesquelles, l'algorithme de colonie de fourmis ACO (Farshidianfar et al., 2013), GA (F. Khoshnoudian et al., 2015) et PSO (Leung et Zhang, 2009). D'autres applications des ANN dans le domaine de l'interaction sol structure où les auteurs ont utilisé cette méthode pour prédire les impédances dynamiques du sol (Sbartai et Goudjil, 2012).

D'autres applications des méthodes d'intelligence artificielle dans l'optimisation est la prédiction de la vitesse de cisaillement du sol (V_s) vu l'importance de ce dernier ainsi que le module de cisaillement qui sont deux paramètres essentiels pour effectuer une analyse dynamique. Parmi lesquelles : ANN (Shahoo Maleki et al., 2014), ACO (Hadi Fattahi et al., 2016) et PSO (Mehdi Assari et al., 2017).

a) L'objectif de la thèse

Comme c'est mentionné ci-dessus, le module de cisaillement maximal (G_{max}) et la vitesse de l'onde de cisaillement (V_s) sont deux paramètres essentiels dans l'analyse dynamique et l'étude de l'interaction sol-structure. Le paramètre (V_s) peut être obtenu par des essais au laboratoire ou par des essais in situ. Cependant, les essais au laboratoire exigent des échantillons intacts, ce qui n'est pas le cas toujours et les essais in situ nécessitent un coût d'investissement très élevés des équipements. Par conséquent, il est important de contourner les essais en utilisant les méthodes d'intelligence artificielle qui se basent sur le développement des bases de données.

L'objectif principal de cette thèse est d'identifier les valeurs du paramètre V_s (vitesse de cisaillement du sol) à partir d'un tassement post-liquéfaction en utilisant une analyse inverse.

Pour ce faire, nous avons essayé de répondre à la question suivante :

- ✓ Quelles sont les valeurs de V_s qui induisent un tassement observé (S_{obs}) ?

Ce tassement observé est pris d'un cas réel.

La réponse à cette question est de trouver à partir d'un tassement observé, les valeurs de V_s . Ce qui fait de notre étude un problème inverse. Nous avons d'abord, utilisé un algorithme génétique couplé à la méthode simplifiée de V_s (Yi, F. 2010). Les résultats obtenus sont très différents des valeurs réelles. Alors, nous avons reformulé notre question comme suit :

- ✓ Quelles sont les valeurs de V_s qui induisent un tassement égal au tassement observé et qui correspondent aux caractéristiques réelles du profil du sol en question ?

Ce problème peut être résolu par deux approches :

- Ajouter des données au problème mono-objectif en utilisant l'algorithme génétique standard.
- Ou traiter le problème en tant que multi-objectif en ajoutant un deuxième objectif.

Pour la première approche, nous avons ajouté le tassement dans chaque point. Les valeurs des tassements sont tirées de (Idriss et Boulanger, 2008). Par contre la deuxième approche est de transformer le problème à un problème multi-objectif. Pour cela, nous avons choisi deux méthodes métaheuristiques : l'algorithme génétique multi-objectif NSGAI et l'algorithme d'optimisation multi-objectif par essaim de particules MOPSO. Ces deux méthodes ont fait l'objet de plusieurs études dans différentes disciplines d'ingénierie. Cependant, cette étude présente la première tentative d'exploration du potentiel d'application de ces méthodes pour optimiser la vitesse de cisaillement du sol (v_s).

b) Organisation de la thèse

Cette thèse est constituée de quatre grands chapitres, précédés d'une introduction générale en plus d'une conclusion générale.

Le premier chapitre dresse un état de l'art non exhaustif sur les applications des méthodes métaheuristiques en génie civil d'une façon générale et particulièrement en géotechnique.

Le deuxième chapitre est consacré à la présentation des méthodes d'optimisation mono-objectif, et multi-objectif en accordant une importance particulière aux méthodes utilisées dans notre étude ; à savoir, l'algorithme génétique standard, l'algorithme génétique multi-objectif NSGA-II et l'algorithme d'optimisation multi-objectif par essaim de particules (MOPSO).

Le troisième chapitre décrit le phénomène de la liquéfaction du sol y compris ses méthodes d'évaluation et ses conséquences notamment le tassement post-liquéfaction.

Le quatrième chapitre constitue le noyau de cette thèse. C'est dans cette partie que nous avons exposé les résultats obtenus par les méthodes d'optimisation choisies ainsi que ses interprétations.

1. Étude bibliographique

1.1. Introduction

Tous les problèmes géotechniques sont liés étroitement avec le sol et/ou la roche qui représentent le support de toute construction que ce soit en profondeur (tunnel, fondations, pieux,..., etc.) ou en surface (pont, bâtiment, rail, route,..., etc). Par leurs natures que ce soit le sol ou la roche, ils exposent un comportement varié et imprévisible dû à leurs compositions structurales. Modéliser le comportement de tels matériaux est complexe et dépasse souvent la capacité des méthodes classiques. Les méthodes d'intelligence artificielle dont les métaheuristiques font parties sont plus adéquates et appropriées pour modéliser le comportement complexe de tels matériaux (sol et roche). Durant les dernières années, les métaheuristiques ont été largement appliquées dans le domaine de génie civil. Dans ce chapitre, nous allons exposer les différentes applications des métaheuristiques en génie civil.

1.2. Applications des algorithmes d'intelligence artificielle en génie civil

Dans cette partie, nous allons exposer les différentes applications des méthodes d'intelligence artificielle les plus utilisées, à savoir, les réseaux de neurones artificiels (**ANN**), l'algorithme génétique (**GA**), le recuit simulé (**SA**), l'algorithme de colonie de fourmis (**ACO**), l'algorithme d'optimisation par essaim de particules (**PSO** et **MOPSO**), algorithme de recherche avec tabous (**TS**), l'algorithme de stratégie harmonie (**HS**). Pour que l'enchaînement des idées soit clair, l'organisation du chapitre sera basée sur le domaine d'application.

1.2.1. Comportement des fondations (capacité portante, tassement,..., etc)

Le comportement des fondations (profonde et superficielle) dans les sols est complexe, incertain et pas encore entièrement compris. Ce fait a encouragé de nombreux chercheurs à appliquer les méthodes d'intelligence artificielle (IA) pour prédire le comportement des fondations. Par exemple (Ardalan et al. 2009 ; Pal and Deswal, 2008) ont utilisé les réseaux de neurones (ANN) pour estimer la capacité portante des pieux. Shahin (2010) a également utilisé les réseaux de neurones pour prédire les capacités portantes sous charge axiale et latérale des fondations sur pieux. Par contre Momeni et al (2014) ont utilisé l'algorithme génétique pour estimer la capacité portante des pieux. Chen et al(2012) ont aussi estimé la capacité portante en utilisant la méthode PSO. Les réseaux de neurones artificiels ont également été utilisé pour prédire le tassement des pieux (Alkroosh et Nikraz, 2011b ; Ismail and Jeng, 2011 ; Pooya Nejad et al., 2009).

1.2.2. Les modèles constitutifs

La modélisation constitutive du comportement du sol joue un rôle important dans le traitement des problèmes liés à la mécanique des sols et à l'ingénierie des fondations. Au cours des trois dernières décennies, de nombreux chercheurs ont consacré énormément d'efforts pour modéliser le comportement des sols. Cependant, des modèles constitutifs proposés basés sur les théories d'élasticité et de plasticité ont une capacité limitée à simuler correctement le comportement des sols. Ceci est dû à la complexité de la formulation, l'idéalisation du comportement des sols et paramètres empiriques excessifs (Adeli, 2001). À cet égard, de nombreuses méthodes ont été proposées comme étant fiables et une alternative pratique pour modéliser le comportement constitutif des sols : **ANN** (Banimahd et al., 2005; Chen et al., 2010; Fu et al., 2007; Garaga and Latha, 2010; Johari et al., 2011; Najjar and Huang, 2007), **GA** (Levasseur et al., 2007), **PSO** (T. Knabe et al., 2013).

1.2.3. Propriétés géotechniques des sols et des roches

Propriétés géotechniques des sols sont contrôlées par des facteurs tels que la minéralogie, histoire des contraintes, l'indice des vides et la pression interstitielle. Les interactions de ces facteurs sont difficiles à établir uniquement par des méthodes statistiques traditionnelles à cause de leurs interdépendances. Sur la base de l'application des algorithmes d'intelligence artificielle, des méthodologies ont été développées pour estimer plusieurs propriétés du sol, y compris l'indice de compression, résistance au cisaillement, coefficient de perméabilité, coef d'humidité, compactage du sol, pression latérale des terres, vitesse de cisaillement du sol et autres. Parmi lesquelles : **ANN** (Celik and Tan, 2005 ; Jianping et al., 2011 ; Park and Lee, 2011, B. DERRAS et A. BEKKOUCHE, 2010), **GA** (Levasseur et al., 2008 ; D. Mendjel et S. Messast, 2009), **MOPSO** (M. Kashif Gill et al., 2006), **NSGA II** (K. Goudjil et B. Sbartai, 2017).

1.2.4. La liquéfaction

La liquéfaction est un phénomène très dangereux qui peut causer des dommages énormes à la plupart des structures de génie civil. Bien que le mécanisme de liquéfaction soit bien connu, la prédiction du potentiel reste très complexe. Ce fait a attiré de nombreux chercheurs pour examiner la pertinence des techniques d'**ANN** pour prédire le potentiel de liquéfaction (Alavi et Gandomi, 2012 ; Baziar et Ghorbani, 2005 ; Hanna et al., 2007; Javadi et al., 2006; Samui et Sitharam, 2011; Shuh-Gi et Ching-Yinn, 2009).

1.2.5. Analyse de la stabilité des talus

La complexité de l'analyse de la stabilité des talus avec une surface de glissement critique non circulaire a attiré l'attention de nombreux chercheurs d'utiliser les algorithmes d'intelligence artificielle qui sont réputées par leurs efficacités afin de déterminer le facteur minimum global de sécurité pour les problèmes complexes. Parmi les quelles : **ANN** (Cho, 2009 ; Das et al., 2011a; Ferentinou et Sakellariou, 2007; Lee et al., 2008; Samui and Kumar, 2006; Wang et Sassa, 2006), **GA** (Goh, A.T.C., 2000 ; McCombie, P. et Wilkinson, P., 2002 ; Zolfaghari, A. R et al., 2005 ; Jianping, S. et Li, J., 2008), **PSO** (Cheng, Y. M. et al., 2007 ; Gordan, B et al., 2004 ; Syed Gous Andrab et al., 2017 ; Wei, Y. et al., 2016), **ACO** (Chenghua, W et al., 2003; Kahatadeniya, K. S. et al., 2009; Li et al., 2004) et **SA** (Cheng, Y. M., 2003 ; Shouju, L. et al., 2003).

1.2.6. Domaine de l'interaction sol structure

Pour diminuer les vibrations provoquées par les tremblements de terre sur les structures, plusieurs techniques ont été développées. Parmi lesquelles un dispositif sous forme d'un amortisseur de masse à accord multiples appelé TDM (Tuned Mass Damper) dont plusieurs résultats publiés indiquent l'effet de l'interaction sol structure sur sa conception. À cet égard, plusieurs chercheurs ont utilisé des méthodes d'intelligence artificielle pour déterminer la conception optimale de TDM. Parmi lesquelles : **ACO** (A. Farshidianfar et al., 2013 ; Viana, F. A. C. et al., 2008), **GA** (Faramarz Khoshnoudian et al., 2015 ; Mohebbi, M. et al., 2013 ; Pourzeynali, S et al., 2007), **PSO** (A.Y.T. Leung et Haijun Zhang, 2009 ; Amini, F. et al., 2013). Autre application d'**ANN** dans le domaine ISS, c'est l'optimisation des impédances dynamiques du sol (B. Sbartai et K. Goudjil, 2012).

1.2.7. Optimisation des constructions et ouvrages géotechnique

Pour construire une fondation, un mur de soutènement ou n'importe quel ouvrage, que ce soit en béton armé tels que les fondations, les murs de soutènement ou en acier tels que les treillis et la charpente métallique, l'objectif est le même : trouver un bon équilibre entre l'économie et la sécurité qui sont les principaux objectifs des concepteurs. À cet égard plusieurs métaheuristiques ont été utilisées afin de trouver la conception optimale. Parmi ces Algorithmes, **GA** dont ses applications en génie civil remontent à 1986, lorsque Goldberg et Samtani ont utilisé l'AG pour la conception optimale d'un système de treillis à 10 barres (Goldberg, D.E. et Samtani, M.P, 1986). Jusqu'à présent, GA et ses variantes ont été utilisées avec succès dans l'optimisation des problèmes d'ingénierie structurelle, la conception des ponts

à longue portée, l'optimisation de la topologie des treillis (Marano, G.C et al., 2011 ; Sgambi, L et al., 2012 ; Sahab, M.G et al., 2013 ; Li, J.P, 2015). Également pour Le recuit simulé (**SA**) qui a été utilisé pour la planification des systèmes ferroviaires à grande vitesse et l'optimisation des bâtiments (Costa, A.L et al., 2013 ; Junghans, L., Darde, N., 2015).

L'algorithme d'optimisation de colonie de fourmis (**ACO**) a également été appliqué pour l'optimisation des poutres en béton armé et dans des problèmes d'ingénierie routière (Sahab, M.G et al., 2013 ; Dias, J.C et al., 2014). **PSO** est un algorithme qui a été utilisé récemment dans l'optimisation des constructions de génie civil telles que la conception de bâtiments de grande taille, l'optimisation des treillis (Gholizadeh, S. et Fattahi, F, 2014 ; Kaveh, A et al., 2014). Autres applications de PSO l'optimisation d'une fondation en béton armé et un mur de soutènement (M. KHAJEHZADEH et al., 2011).

Autres applications des méthodes de l'intelligence artificielle commençant par **-ANN** : mur de soutènement (Kung et al., 2007 ; Yildiz et al., 2010), stabilisation des sols (Das et al., 2011b ; Park and Kim, 2011), excavation (Soroush et al., 2006). **GA** : barrage (Syed Gous Andrab et al., 2017). **PSO** : barrage et systèmes de distribution d'eau (Syed Gous Andrab et al., 2017 ; Montalvo, I et al., 2014)

1.3. Conclusion

Dans ce chapitre, nous avons dressé un aperçu sur les applications des méthodes d'optimisation dans le domaine de génie civil. Les méthodes publiées dans ce domaine ne peuvent pas être décrites dans quelques pages, c'est pourquoi nous nous sommes contenté d'exposer les plus récents et seulement celles appliquées en génie civil. D'après cette recherche, nous avons constaté que les réseaux de neurones sont les plus utilisés surtout dans les problèmes de la prédiction. Contrairement aux ANN, les métaheuristiques qui comprennent (PSO, GA, SA, TS, et toutes les méthodes d'optimisation multi-objectif) sont largement appliquées dans l'optimisation des conceptions des ouvrages et des structures. Dans le domaine de la prédiction, ces méthodes se présentent sous des méthodes hybrides avec les réseaux de neurones ANN. Nous avons essayé à partir de cette recherche bibliographique de mettre en relief les applications des méthodes utilisées dans cette thèse, à savoir, NSGA-II et MOPSO, en géotechnique. Pour la méthode MOPSO, nous avons trouvé que celles citées ci-dessus. Par contre, la méthode NSGA-II n'a pas été appliquée dans le domaine géotechnique. D'après Xin-She et al (2016), la méthode NSGA-II a été largement appliquée dans l'optimisation des réseaux de signalisation routière. D'après cette recherche bibliographique, nous pouvons dire qu'aucune

méthode des deux n'a été utilisée pour optimiser le paramètre dynamique V_s . Dans le chapitre suivant, la plupart des méthodes citées dans ce chapitre seront bien détaillées.

2. Méthodes d'optimisation

2.1. Introduction

L'ingénieur de génie civil aujourd'hui est obligé de développer des conceptions sécurisées et économiques. Ces conceptions sont influencées par plusieurs facteurs (techniques, économiques, environnementaux, géométriques, et géotechnique,..., etc.). Le manque des informations précises telles que les données géotechniques est un problème parmi d'autres qui contraint le géotechnicien à faire sa tâche convenablement (évaluation du tassement des fondations, la capacité portante et l'estimation du facteur de sécurité vis-à-vis de la liquéfaction,..., etc.). Tels problèmes peuvent souvent être considéré comme des problèmes d'optimisation dans lesquels on cherche à minimiser (ou maximiser) une fonction ou plusieurs fonctions par rapport à l'ensemble des paramètres en question (données manquantes par exemple). Le calcul différentiel fournit par les méthodes classiques pour déterminer les maximums et les minimums des fonctions objectifs nécessitent que ces dernières soient différentiable et continues dans l'intervalle de définition ainsi que les contraintes s'ils existent. Dans le cas contraire, cela conduit à un ensemble de fonctions simultanées qui, en général, sont non linéaires, multimodales et discontinues. La solution d'un tel système est compliquée. Ce qui a conduit les chercheurs à proposer des méthodes de résolution de plus en plus performantes, parmi lesquelles on peut citer les méthodes métaheuristiques. Une métaheuristique est un algorithme d'optimisation utilisé pour la résolution des problèmes d'optimisation difficile pour lesquels on ne connaît pas de méthode classique plus efficace. Elles sont des algorithmes stochastiques itératifs, qui font évoluer une solution ou une population de solutions vers un optimum global. Elles sont souvent utilisées pour leur facilité d'implémentation, de manipulation et leurs flexibilités (adaptables à tout type de problème d'optimisation). Néanmoins, elles sont le plus souvent employées pour des problèmes d'optimisation difficile, où des méthodes d'optimisation classiques (méthodes déterministes, notamment) présentent leurs limites.

2.2. Problème d'optimisation

L'optimisation est une discipline mathématique qui consiste à trouver le maximum ou le minimum d'une fonction soumis ou non à des contraintes. Un problème d'optimisation peut s'écrire sous la forme générale suivante :

Minimiser

$$f_m(x_i), \quad m = 1, \dots, M; \quad (2.1)$$

satisfaisant

$$g_j(x) \geq 0, \quad j = 1, \dots, J \quad (2.2)$$

$$h_k(x) = 0, \quad k = 1, \dots, K; \quad (2.3)$$

$$x_i \in S, \quad i = 1, \dots, n; \quad (2.4)$$

Où m ($m \geq 1$) est le nombre d'objectifs, $x_i = (x_1, \dots, x_n)$ est le vecteur représentant les variables de décision, et S représente l'ensemble des solutions réalisables associée à des contraintes d'égalité h_k et d'inégalité g_j . f_m est le vecteur des objectifs à optimiser.

L'espace de recherche S représente l'espace de décision ou l'espace de paramètres de POM. Par contre, l'espace dans lequel appartient le vecteur objectif est appelé l'espace objectif (figure 2.1).

La classification des problèmes d'optimisation dépend des paramètres qu'ils les composent. Par exemple, on dit que le problème est linéaire si les équations (2.1), (2.2) et (2.3) sont linéaires, dans le cas contraire ou une des trois équations est non linéaire le problème dit non linéaire. Également, le problème dit problème d'optimisation avec contrainte si une des équations (2.2) et (2.3) existent. Dans le cas contraire, le problème dit sans contraintes. Le problème d'optimisation généralement est classé par rapport au nombre de fonctions 'objectifs'. Si $i=1$, le problème dit mono-objectif, par contre si $i \geq 2$, le problème dit multi-objectif. C'est cette classification qui nous conduit vers le choix des méthodes de résolution. Si le problème est linéaire, il suffit d'utiliser une méthode déterministe de type gradient qui nécessite seulement que la fonction objectif soit dérivable. Dans le cas d'un problème non linéaire avec une seule fonction objectif (problème mono-objectif), une méthode évolutionnaire telle que l'algorithme génétique standard peut facilement résoudre le problème. Par contre, dans un problème non linéaire multi-objectif avec des contraintes, la tâche devient plus complexe et les méthodes classiques ou même les méthodes évolutionnaires mono-objectif ne peuvent pas trouver les solutions escomptées.

Les méthodes d'optimisation peuvent être divisées en méthodes classiques (déterministes) et méthodes métaheuristiques. En ce qui concerne, les méthodes classiques (déterministes) se sont des méthodes qui n'utilisent aucun concept aléatoire, et se déterminent par une exploration systématique de l'espace de recherche. Elles nécessitent des hypothèses sur la fonction 'objectif', telles que la continuité et dérivabilité en tout point du domaine de recherche. Les

méthodes classiques se divisent en deux groupes : les méthodes d'exploration directes et les méthodes d'exploration indirecte. Les méthodes d'exploration directes recherchent les optima locaux en se déplaçant dans une direction qui dépend du gradient de la fonction. Par contre, les méthodes

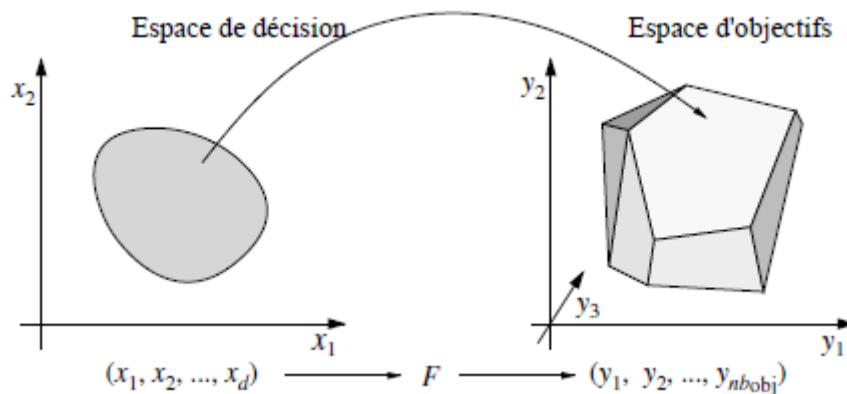


Figure 2.1- Espace de décision et objectifs dans POM (Talbi, 2009)

indirectes cherchent à atteindre les extrema locaux en résolvant des systèmes d'équations, souvent non linéaires, obtenus par l'annulation du vecteur gradient de la fonction. Parmi ces méthodes, on trouve : Les méthodes de gradient, la méthode de Newton, et la méthode de Nelder Mead. Toutes ces méthodes sont largement détaillées dans (Fletcher, 1987).

En réalité, les problèmes d'optimisation sont généralement complexes. Leur complexité réside dans les points suivants :

- Fonction objectif et contraintes sont non linéaires.
- Fonctions peuvent être discontinues.
- Fonctions objectives multimodales (plusieurs optimums locaux).

Donc la résolution des problèmes d'optimisation qui ont les caractéristiques ci-dessus, nécessite des méthodes qui n'ont pas besoin de calculer la dérivée de la fonction, peuvent surmonter le problème d'optimum local et qui ne demande pas un temps de calcul prohibitif. Les méthodes appropriées pour un tel problème s'appellent les métaheuristiques. Dans ce qui suit, nous décrivons les méthodes métaheuristiques utilisées dans l'optimisation mono-objectif et multi-objectif en accordant une importance particulière aux méthodes utilisées dans cette thèse ; à savoir la méthode de l'algorithme génétique, la méthode d'optimisation par essaim des particules, ainsi que les méthodes d'optimisation multi- objectif NSGA II et MOPSO.

2.3. Les méthodes métaheuristiques

Le mot métaheuristique est un mot grec composé de deux mots : heuristique et meta. Le mot heuristique signifie trouver et le suffixe meta signifie au-delà (dans un niveau supérieur). Les métaheuristiques sont des stratégies qui visent à explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales en utilisant des mécanismes aléatoires permettant d'éviter les optimums locaux. Elles peuvent être classées selon différentes notions.

- a) **Notion de parcours et de population** : La notion de parcours ou une solution unique représente les algorithmes qui font évoluer une seule solution dans l'espace de recherche à chaque itération où la notion de voisinage est capitale. Les plus connus sont le recuit simulé, la recherche avec tabous et la recherche à voisinage variable. Par contre celles fondées sur la notion de population se sont des métaheuristiques qui manipulent un ensemble de solutions en parallèle, à chaque itération. Telles que les algorithmes génétiques, l'optimisation par essaims particuliers et les algorithmes de colonies de fourmis.
- b) **Notion de mémoire** : La méthode de recherche locale consiste à passer d'une solution à une autre proche sans prendre en considération l'historique de la recherche. Ces méthodes sont dites sans mémoire, par contre beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche) pour dresser l'optimisation aux itérations subséquentes dans le but de trouver les meilleures solutions. Parmi ces méthodes, la méthode de recherche avec tabous (Tabu Search en anglais).
- c) **Notion de la fonction objectif (statique ou dynamique)** : Il est également possible de classer les métaheuristiques selon la fonction objectif utilisée, c'est-à-dire la nature de la fonction. La fonction objectif peut être statique (qui demeure constante tout au long de l'optimisation) ou dynamique (quand la fonction objectif est modifiée au cours de la recherche).
- d) **Notion du nombre de fonction** : Les problèmes d'optimisation sont généralement multi-objectif, c'est pourquoi les algorithmes évolutionnaires ainsi que les algorithmes d'essaim particuliers qui font partie des métaheuristiques, sont classés en méthodes mono-objectif et multi-objectif. Les algorithmes mono-objectif sont des méthodes qui optimisent une seule fonction objectif telles que l'algorithme génétique, le recuit simulé et la recherche avec tabous, ...etc. En ce qui concerne, les algorithmes multi-objectif,

on trouve : l'algorithme génétique de tri non-dominé (NSGA II), multi-objectif par d'essaim particuliers (MOPSO), SPEA II (Strength Pareto Evolutionary Algorithm II),...etc.

- e) **Notion de l'hybridation** : Les métaheuristiques peuvent être composées de plusieurs méthodes se répartissant les tâches de recherche. On parle dans ce cas de la taxinomie des métaheuristiques hybrides qui se sépare en deux parties : une classification hiérarchique et une classification plate (Talbi, 2009). La classification est applicable aux méthodes déterministes aussi bien qu'aux métaheuristiques.

Il est difficile de citer toutes les méthodes métaheuristiques. Mais, nous avons essayé de dresser un tableau contenant celles qui sont les plus utilisées.

Tableau 2.1- Classification des méthodes métaheuristiques

Méthodes métaheuristiques			
Algorithmes évolutionnaires		Méthodes d'optimisation par essais	
Mono-Objectif	Multiobjectifs	Mono-Objectif	Multiobjectifs
Algorithme génétique (GA), Recuit simule (Simulated Anealing SA), Recherche Taboue (Tabu Search TS)	VEGA(Vector Evaluated Genetic Algorithm), MOGA(Multiple Objective Genetic Algorithm), NPGA(Niched Pareto Genetic Algorithm), SPEA(Strength Pareto Evolutionary Algorithm), PAES(Pareto Archived Evolution Strategy), NSGA I et II(Non dominated Sorting Genetic Algorithm)	Optimisation par essaim de particules(PSO), Algorithme de colonie de fourmis (ACO)	Optimisation Multiobjectifs par essais de particules MOPSO

2.4. Métaheuristiques d'optimisation mono-objectif

Les métaheuristiques d'optimisation peuvent être classées selon différents critères, dans cette thèse, c'est le nombre d'objectif que nous avons choisi. Une métaheuristique mono-objectif est une méthode qui optimise un problème avec un seul objectif. Parmi ces méthodes, l'algorithme génétique, la méthode de recherche avec tabous, optimisation par essaim de particules et algorithme de colonie de fourmis,... etc. Dans ce qui suit, nous allons décrire quelques-unes.

2.4.1. Algorithme génétiques

Les algorithmes génétiques ont été développés par J. Holland dans les années 1970 (Université du Michigan, États-Unis) pour comprendre les processus adaptatifs des systèmes naturels (Holland., 1975). Ensuite, ils ont été appliqués à l'optimisation et l'apprentissage des machines dans les années 1980 (Goldberg, 1989). L'AG représente une classe très populaire des algorithmes évolutionnaires (EV). Traditionnellement, les AG sont associées à l'utilisation d'une représentation binaire, mais aujourd'hui, on peut trouver des AG qui utilisent d'autres types de représentations. Une GA applique généralement un opérateur de croisement à deux solutions qui jouent un rôle majeur, plus un opérateur de mutation qui modifie de façon aléatoire les contenus individuels pour promouvoir la diversité. Les AG utilisent une sélection probabiliste qui est à l'origine la sélection proportionnelle. Le remplacement (sélection de survivant) est générateur, c'est-à-dire que les parents sont remplacés systématiquement par les descendants. L'opérateur de croisement est basé sur le n-point ou le croisement uniforme alors que la mutation est une modification aléatoire. Des valeurs de probabilité fixes sont appliquées aux deux opérateurs de reproduction (p_c et p_m).

2.4.1.1. Concepts communs pour les algorithmes évolutionnaires

Les principaux composants de recherche pour la conception d'un algorithme évolutionnaire sont les suivants : Représentation, initialisation de la population, fonction objectif, stratégie de sélection, stratégie de reproduction et critères d'arrêt.

2.4.1.1.1. Représentation

Il s'agit d'un composant de recherche commun pour toutes les métaheuristiques. La solution codée est appelée chromosome tandis que les variables de décision dans une solution (chromosome) sont des gènes. Les valeurs possibles des variables (gènes) sont les allèles et la position d'un élément (gène) dans un chromosome est appelée locus.

2.4.1.1.2. *Fonction objectif*

Dans la communauté d'algorithmes évolutionnaires, le terme fitness se réfère à la fonction 'objectif'.

2.4.1.1.3. *Stratégie de sélection*

Le mécanisme de sélection est l'un des principaux composants de recherche dans les AEs. La stratégie de sélection détermine quels individus sont choisis pour l'accouplement (reproduction). La stratégie de sélection peut être effectuée sous différentes formes : sélection de roulette, La sélection universelle stochastique, sélection de tournoi et sélection par rang.

- **Sélection par rang** : Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation, le hasard n'entre donc pas dans ce mode de sélection. En fait, si n individus constituent la population, la sélection appliquée consiste à conserver les k meilleurs individus (au sens de la fonction d'évaluation) suivant une probabilité qui dépend du rang (et pas de la fonction d'évaluation).

- **Probabilité de sélection proportionnelle à l'adaptation** : Appelé aussi « roulette » ou « roue de la fortune », pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème. Afin de sélectionner un individu, on utilise le principe de la roue de la fortune biaisée. Cette roue est une roue de la fortune classique sur laquelle chaque individu est représenté par une portion proportionnelle à son adaptation. On effectue ensuite un tirage au sort homogène sur cette roue.

- **Sélection par tournoi** : Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation.

- **Sélection universelle stochastique** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité $1/P$ d'être sélectionné, où P est le nombre total d'individus dans la population.

2.4.1.1.4. *Stratégie de reproduction*

Une fois que le choix des individus pour former les parents est effectué, le rôle de la phase de reproduction est l'application d'opérateurs de variation tels que la mutation et croisement.

- **Croisement** : Croisement (Crossover) est un opérateur qui permet la combinaison des individus (parents) pour générer de nouveaux individus. Les opérateurs de croisement dans les algorithmes génétiques implémentent un mécanisme qui mélange les gènes des parents. Le plus connu est le **croisement de n points**. Il divise deux solutions à n positions et alternativement les assemble à une nouvelle (figure 2.2).

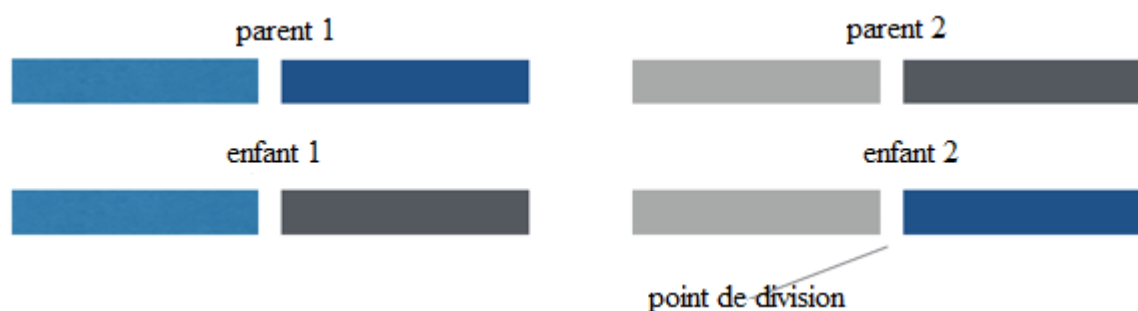


Figure 2.2-Illustration du croisement d'un point.

Le deuxième type de croisement est le **croisement arithmétique**, également connu sous le nom de croisement intermédiaire. Il calcule la moyenne arithmétique de toutes les solutions parentales. Il y a aussi le **croisement uniforme** utilise un coefficient 0,5 pour choisir au hasard un peu de l'un ou l'autre des parents. La question qui se pose, laquelle des solutions parentales participe à la génération de nouvelles solutions. Plusieurs algorithmes génétiques simplifient cette étape et choisissent les parents au hasard pour l'opération de croisement avec une distribution uniforme.

- **Mutation** : Les opérateurs de mutation sont des opérateurs unaires agissant sur un seul individu. Les mutations représentent de petits changements chez certains individus de la population. La probabilité p_m définit la probabilité de muter chaque élément (gène) de la représentation. En général, de petites valeurs sont recommandées pour cette probabilité ($p_m \in [0,001, 0,01]$) (Talbi, 2009). Certaines stratégies initialisent la probabilité de mutation à $1/k$ où k est le nombre de variables de décision, c'est-à-dire dans la moyenne d'une seule variable est modifiée. Il existe plusieurs types de mutation. Parmi lesquels, **mutation gaussienne** qui ajoute un nombre au hasard à chaque entrée vectorielle d'un individu. Ce nombre aléatoire est tiré d'une distribution gaussienne. **Mutation uniforme** remplace la valeur de la solution (gène) choisie par une valeur aléatoire uniforme sélectionnée entre les limites supérieures et inférieures spécifiées par l'utilisateur pour ce gène. Troisième type de mutation est l'**adaptatif réalisable** qui génère de façon aléatoire des directions adaptatives par rapport à la dernière génération réussie ou non. Une longueur d'étape est choisie dans chaque direction afin que les contraintes soient satisfaites.

Le processus de l'algorithme génétique peut être illustré par la figure (2.3).

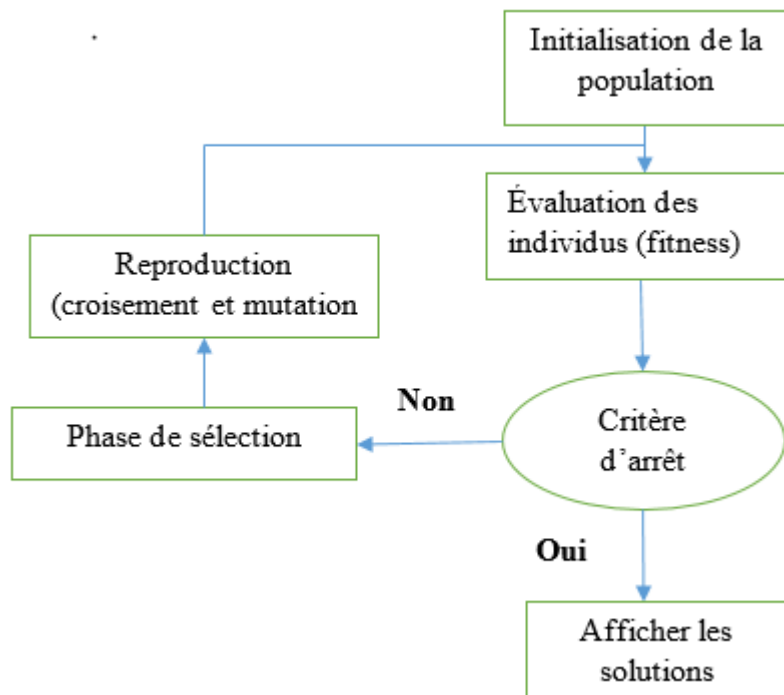


Figure 2.3-Principe de fonctionnement d'AG

2.4.2. Méthode de recuit simulé (Simulated annealing)

S. Kirkpatrick et ses collègues, spécialistes en physique statistique, (Kirkpatrick et al. 1983) ont proposé de traiter les problèmes rencontrés dans leurs travaux par la méthode du recuit simulé. Le recuit simulé (**Simulated annealing en anglais**) est une méthode métaheuristique utilisée dans l'optimisation globale. Elle est souvent utilisée lorsque l'espace de recherche est discret (par exemple, le problème du voyageur TSP). Le nom et l'inspiration proviennent du recuit dans la métallurgie, une technique impliquant le chauffage et le refroidissement contrôlé d'un matériau pour augmenter la taille de ses cristaux et réduire leurs défauts. La méthode est une adaptation de l'algorithme Metropolis-Hastings, une méthode de Monte Carlo pour générer des états d'échantillonnage d'un système thermodynamique.

2.4.2.1. Analogie physique

Le métal possède une configuration cristalline où les atomes sont assemblés d'une manière régulière. Néanmoins, si le métal a refroidi trop vite, sa structure cristalline peut présenter des défauts. Le recuit consiste à réchauffer le métal pour redonner l'énergie à ses atomes de sortir de leurs positions (figure 2.4.a), puis à le laisser refroidir lentement pour permettre aux atomes de revenir vers de meilleures positions (figure 2.4.b). Cette notion de refroidissement lent implémentée dans l'algorithme de recuit simulé est interprétée comme une diminution lente de la probabilité d'accepter de pires solutions à mesure que l'espace de solution est exploré

(accepter des solutions pires est une propriété fondamentale des métaheuristiques car elle permet une recherche plus étendue des solutions optimales).

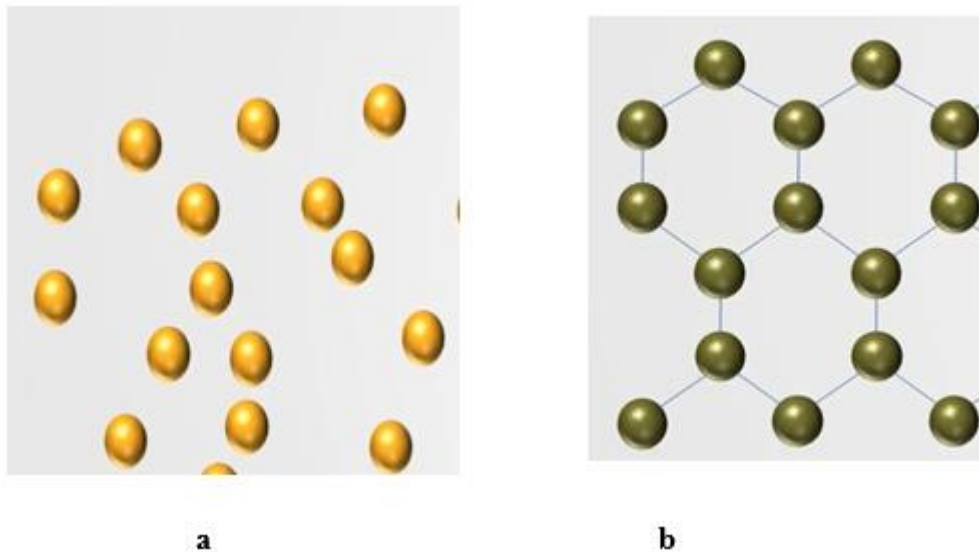


Figure 2.4 -Analogie physique de recuit simulé.

2.4.2.1.1. Principe du recuit simulé

La méthode du recuit simulé consiste à :

- a- Attribuer une température virtuelle au système T,
- b- Choisir les paramètres au hasard et mesurer le score obtenu,
- c- Modifier légèrement les paramètres et mesurer de nouveau le score obtenu,
- d- Si le coût est meilleur (plus bas), on conserve ses nouveaux paramètres, dans le cas contraire, on conserve ses nouveaux paramètres avec une probabilité proportionnelle à la température $e^{\frac{-\delta E}{T}}$ (figure 5),
- e- Refroidir un petit peu le système,
- f- Recommencer à partir de l'étape c jusqu'à un critère d'arrêt soit atteint.

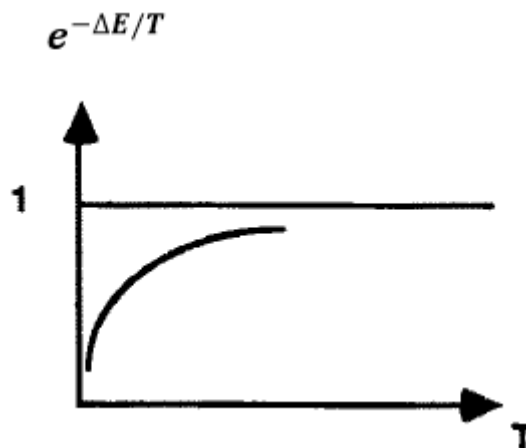


Figure 2. 5 - Fonction de probabilité pour le processus de recuit

2.4.3. Méthode de recherche avec tabous (Tabu Search)

Méthode de recherche avec tabous est une méthode métaheuristique qui a été proposée par (Glover, 1986) et formalisé par (Glover, 1989,1990). Cette méthode utilise un algorithme de recherche locale pour progresser itérativement d'une solution potentielle x à une solution améliorée x' dans le voisinage de x .

Les méthodes de recherche locales sont souvent bloquées dans les optimums locaux. Afin d'éviter ces pièges et d'explorer les régions de l'espace de recherche qui seraient laissées inexplorées par d'autres procédures de recherche locales. La recherche taboue explore soigneusement le voisinage de chaque solution en utilisant les structures de mémoire. Ces structures de mémoire forment ce qu'on appelle la liste des tabous d'où vient le nom de la méthode. Cette liste représente un ensemble de règles et des solutions interdites, utilisées pour filtrer les solutions qui seront admises au voisinage de x .

Les structures de mémoire utilisées dans la recherche taboue peuvent être divisées en trois catégories :

- Mémoire à court terme : la liste des solutions récemment considérées. Si une solution potentielle apparaît sur la liste des tabous, elle ne peut pas être revisitée jusqu'à ce qu'elle atteigne un point d'expiration.
- Mémoire intermédiaire : règles d'intensification visant à polariser la recherche vers des zones prometteuses de l'espace de recherche
- Mémoire à long terme : règles de diversification qui conduisent la recherche dans de nouvelles régions (c'est-à-dire concernant les réinitialisations lorsque la recherche est bloquée dans un plateau ou une impasse sous-optimale).

Les mémoires à court terme, à moyen terme et à long terme peuvent se chevaucher dans la pratique. Dans ces catégories, la mémoire peut encore être différenciée par des mesures telles que la fréquence et l'impact des modifications apportées. La mémoire à court terme seule peut être suffisante pour obtenir une solution supérieure à celles trouvées par les méthodes de recherche locales conventionnelles, mais des structures intermédiaires et à long terme sont souvent nécessaires pour résoudre des problèmes plus difficiles.

La recherche Tabu est souvent comparée à d'autres méthodes métaheuristiques, telles que le recuit simulé, les algorithmes génétiques, les algorithmes d'optimisation de la colonie de fourmis, ...etc.

2.4.4. Algorithmes de colonies de fourmis

Les algorithmes de colonies de fourmis ('**Ant Colony Algorithms**' en anglais) sont des algorithmes métaheuristiques d'optimisation. Cette approche, proposée par Colorni et al (1991) se tente de simuler la capacité collective de résoudre certains problèmes. Ce processus a été observé dans une colonie de fourmis, dont les membres sont équipés individuellement avec des capacités cognitives limitées. Leur succès ont incité, en particulier, les entomologistes d'analyser la collaboration existante entre les fourmis dans la recherche de nourriture à l'extérieur de la fourmilière. Il est remarquable que les fourmis suivent toujours le même chemin, et ce chemin est le plus court possible. Ce contrôle est le résultat d'un mode de communication indirecte. Chaque fourmi dépose sur son chemin, une substance chimique, appelée "phéromone". Tous les membres de la colonie perçoivent cette substance et changent leurs directions vers les zones les plus "odorantes".

Il résulte notamment d'un comportement comme celui des fourmis de retrouver rapidement le chemin le plus court, si celui-ci est bloqué accidentellement par un obstacle (figure 2.6). Bien que ce comportement ait été pris comme point de départ pour modéliser l'algorithme, Dorigo et al (2006) ont proposé un nouvel algorithme pour la solution du problème TSP. Depuis ces travaux, la méthode a été étendue à beaucoup d'autres problèmes d'optimisation.

La figure (2.6) illustre les étapes à suivre des fourmis dans leur recherche de la nourriture. Ce processus est décrit par les expressions ci-dessous :

- Les vraies fourmis suivent un chemin entre le nid et une source de nourriture.

- Un obstacle apparaît sur le chemin, les fourmis choisissent de tourner à gauche ou à droite, avec des probabilités égales ; La phéromone est déposée plus rapidement sur le chemin le plus court.
- Toutes les fourmis ont choisi le chemin le plus court.

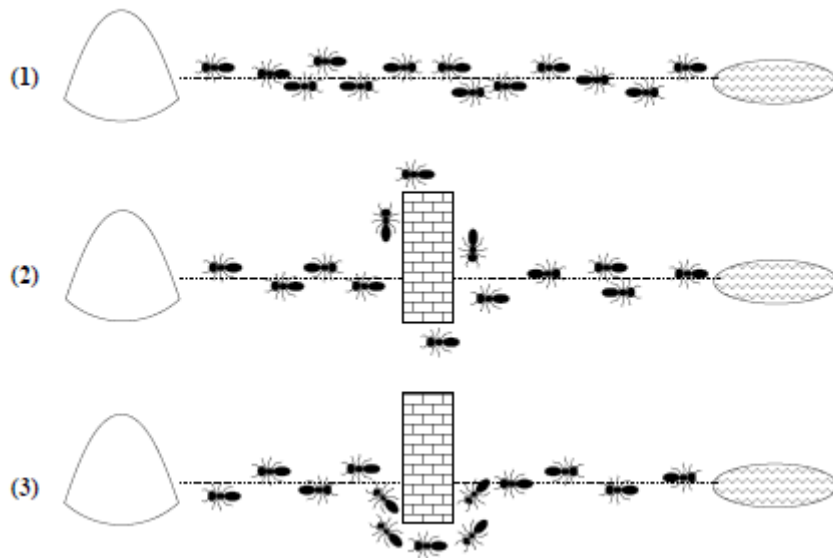


Figure 2.6 – Faculté d'une colonie de fourmis pour trouver le chemin le plus court, fortuitement bloqué par un obstacle (Johann Dreo et al., 2006)

Les algorithmes de colonies de fourmis ont plusieurs caractéristiques intéressantes ; à mentionner en particulier le parallélisme intrinsèque élevé, la flexibilité (une colonie de fourmis est capable pour s'adapter aux modifications de l'environnement), la robustesse (une colonie est capable à maintenir son activité même si certains individus échouent), la décentralisation (Une colonie n'obéit pas à une autorité centralisée) et l'auto-organisation (une La colonie se trouve une solution qui n'est pas connue à l'avance). Ces caractéristiques ci-citées fait des algorithmes de colonie de fourmis une méthode d'optimisation très puissante qui peut être utilisée dans le cas des problèmes difficiles d'optimisation.

2.4.5. Méthode d'essaim de particules (Particles Swarm Optimization)

L'optimisation des essaims de particules est une autre méthode stochastique à base de population inspirée de l'intelligence des essaims (Kennedy et Eberhart, 1995). Elle imite le comportement social des organismes naturels tels que le vol d'oiseaux et les bancs de poissons pour repérer les endroits avec assez de nourriture. Effectivement, dans ces essaims, un comportement coordonné utilisant des mouvements locaux émerge sans aucun contrôle central. À l'origine, PSO a été conçu avec succès pour une optimisation des problèmes continus. Sa

première application aux problèmes d'optimisation a été proposée par Kennedy et Eberhart (1995).

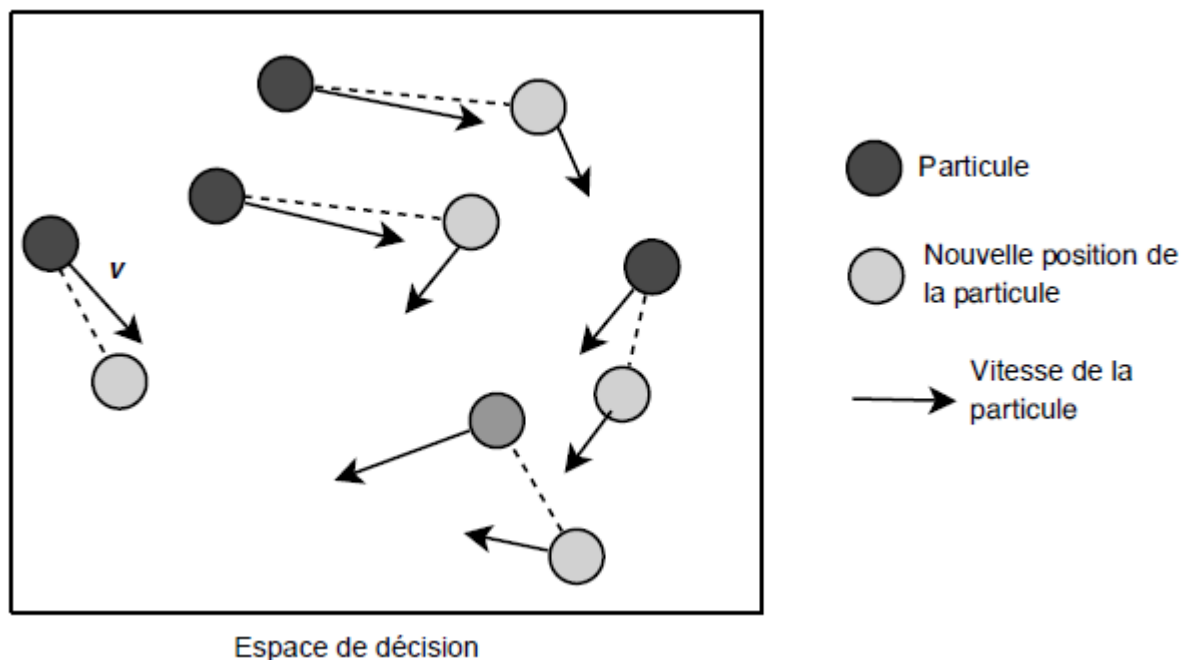


Figure 2.7- Principe de fonctionnement de PSO (Talbi, 2009)

Dans le modèle de base, un essaim se compose de N particules volant dans un espace de recherche de dimension D . Chaque particule i est une solution au problème, elle est représentée par le vecteur x_i dans l'espace de décision. Une particule a sa propre position et vitesse. Ce qui signifie la direction de vol qui mène vers la prochaine étape de la particule (figure 2.7). L'avantage de cette méthode réside dans la coopération qui existe entre les particules. Le succès de certaines particules influence le comportement des autres. Chaque particule ajuste successivement sa position x_i vers l'optimum global selon les deux facteurs : Sa meilleure position visitée (p_{best_i}) désignée par $p_i = (p_1, p_2, \dots, p_D)$ et la meilleure position visité par l'essaim entier (plus généreux) (ou l_{best} , la meilleure position pour un sous-ensemble donné de l'essaim) noté p_g . Le vecteur $(p_g - x_i)$ représente la différence entre la position actuelle de la particule i et la meilleure position de son voisinage.

2.4.5.1. Typologie de voisinage

Le voisinage doit être défini pour chaque particule. Ce voisinage dénote l'influence sociale entre les particules. Il y a plusieurs formes de définir un voisinage. Traditionnellement, deux méthodes sont utilisées :

- **Méthode gbest** : Dans la méthode gbest, le voisinage est défini comme étant la population entière (figure 2.8).
- **Méthode lbest** : la méthode lbest représente le meilleur voisinage local, elle est donnée sous une topologie associée à l'essaim. Par conséquent, le voisinage d'une particule est l'ensemble des particules directement connectées. Le voisinage peut être vide dans lequel les particules sont isolés (c'est-à-dire $C2 = 0$). La figure (2.8) montre trois topologies différentes : Graphique complet, Graphique en anneau et graphique du petit monde.

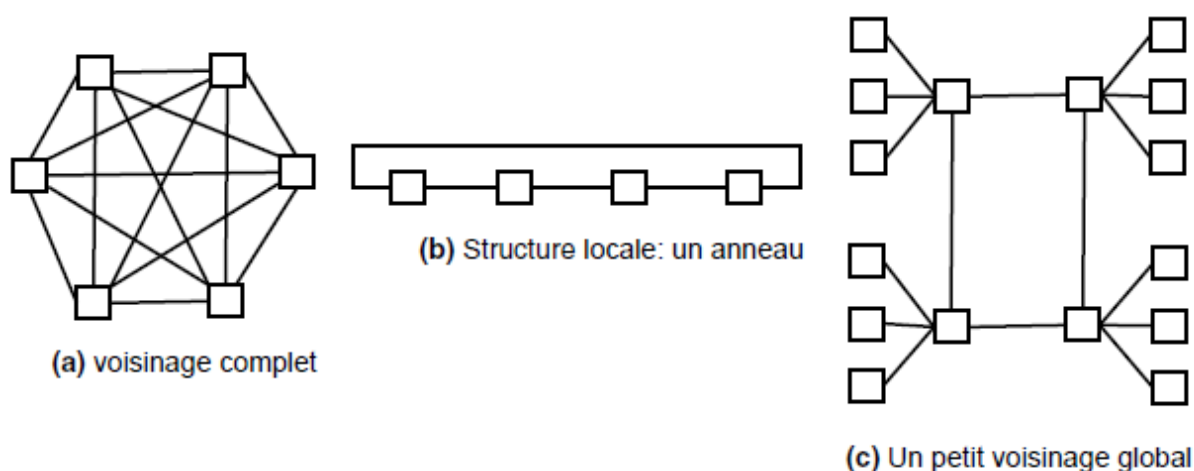


Figure 2.8 – Différents types de voisinage associé aux particules (Talbi, 2009).

Selon le voisinage utilisé, un leader (c'est-à-dire lbest ou gbest) représente la particule qui sert à guider la recherche d'une particule vers de meilleures régions de l'espace de décision.

Une particule est définie par trois vecteurs :

- Le vecteur X représente la position actuelle (emplacement) de la particule dans l'espace de recherche.
- Le vecteur P enregistre l'emplacement de la meilleure solution trouvée jusqu'à présent par la particule.
- Le vecteur V représente la vitesse de déplacement de la particule (direction).

Un essaim de particules peut être considéré comme un groupe d'individus (Particules dans PSO) dont les mises à jour de tous les individus s'effectuent en parallèle ; Chaque nouvelle position de l'individu dépend uniquement de l'ancienne valeur et son voisinage, et toutes les cellules sont mises à jour en utilisant les mêmes règles. À chaque itération, chaque particule appliquera les opérations suivantes :

- La vitesse qui définit la quantité de changement appliquée à la particule est définie comme suit :

$$V_i(t) = V_i(t - 1) + r_1 c_1 (p_i - x_i(t - 1)) + r_2 c_2 (p_g - x_i(t - 1)) \quad (2.5)$$

Où r_1 et r_2 sont deux variables aléatoires dans la plage $[0, 1]$. Les constantes c_1 et c_2 représentent les facteurs d'apprentissage. Le paramètre c_1 est le facteur d'apprentissage cognitif qui représente l'attraction d'une particule envers son propre succès. Or, le paramètre c_2 est le facteur social d'apprentissage qui représente l'attraction d'une particule envers le succès de ses voisins. La vitesse définit la direction et la distance que la particule devrait effectuée (figure 2.9).

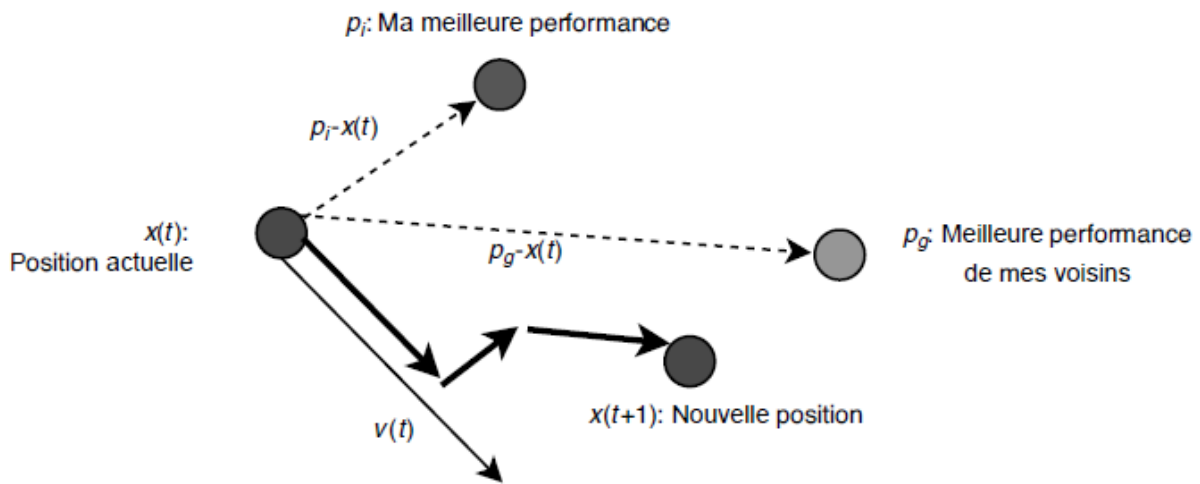


Figure 2.9- Mouvement d'une particule et la mise à jour de sa vitesse (Talbi, 2009)

La vitesse V_i est limitée à une valeur maximale $[-V_{max}, +V_{max}]$. Si la vitesse v_i dépasse V_{max} , il sera réinitialisé à V_{max} . Dans la procédure de mise à jour de la vitesse, un poids d'inertie w est généralement ajouté à par rapport à la vitesse précédente.

$$V_i(t) = W \times V_i(t - 1) + r_1 c_1 (p_i - x_i(t - 1)) + r_2 c_2 (p_g - x_i(t - 1)) \quad (2.6)$$

Le paramètre d'inertie w contrôle l'impact de la vitesse précédente sur la vitesse actuelle. Pour des valeurs élevées du poids d'inertie, les vitesses actualisées seront beaucoup plus élevées. Ainsi, le poids d'inertie représente un compromis entre l'exploration globale et l'exploitation locale. Une grande valeur de poids d'inertie favorise l'exploration de l'espace de recherche (c.-à-d., diversifier la recherche dans l'espace de recherche entier). Tandis qu'un poids d'inertie plus petit encourage l'exploitation locale (c'est-à-dire, intensifier la recherche dans la région actuelle).

- Mettre à jour la position : chaque particule actualise ses coordonnées dans l'espace de décision, en utilisant l'expression suivante :

$$x_i(t) = x_i(t - 1) + v_i(t) \quad (2.7)$$

Ensuite, il se déplace à la nouvelle position.

- Mettre à jour les meilleures positions trouvées : chaque particule mettra à jour la meilleure solution locale :

$$\text{If } f(x) < pbest_i, \text{ alors } p_i = x_i$$

En outre, la meilleure solution globale de l'essaim est mise à jour comme suit :

$$\text{If } f(x_i) < gbest, \text{ alors } g_i = x_i$$

Par conséquent, à chaque itération, chaque particule changera sa position selon sa propre expérience et celle de ses voisines. Comme tout concept d'intelligence, les agents (particules pour PSO) échangent des informations en partageant leurs expériences. Le comportement de l'essaim émerge de l'interaction de ces particules. Dans PSO, le partage global de l'information est réalisé grâce à la meilleure solution globale (gbest).

2.5. Métaheuristiques d'optimisation multi-objectif

Le développement de métaheuristiques multi-objectif avancées a été largement étudié depuis la fin des années 1980. C'est une question importante dans la résolution des problèmes d'optimisation multi-objectif (POM) complexes et de grande taille. La solution optimale pour les POM n'est pas une solution unique comme celle des problèmes d'optimisation mono-objectif, mais un ensemble de solutions définies comme des solutions optimales de Pareto.

Une solution est dite Pareto optimale s'il n'est pas possible d'améliorer un objectif donné sans détériorer au moins un autre objectif. Cet ensemble de solutions représente les solutions de compromis entre les différents objectifs. L'objectif principal de la résolution d'un problème multi-objectif est d'obtenir l'ensemble optimal de Pareto et, par conséquent, le front de Pareto.

Comparé à l'optimisation mono-objectif, la difficulté à résoudre les POM réside dans les points suivants :

- Il n'existe pas de définitions couramment utilisées sur l'optimalité globale d'une solution comme dans l'optimisation mono-objectif. La relation d'ordre entre les solutions d'un problème de POM est partielle et le choix final dépend du décideur.
- Le nombre de solutions optimales de Pareto augmente en fonction de la taille du problème et principalement du nombre d'objectifs considérés.

Avant de présenter les différentes métaheuristiques multi-objectives, nous devons d'abord expliquer quelques concepts relatifs aux problèmes d'optimisation multi-objectif.

2.5.1. Concepts d'optimisation multi-objectif

Cette section couvre les principaux concepts d'optimisation multi-objectif, tels que la notion de dominance, Optimalité de Pareto, ensemble de solutions optimales de Pareto et front de Pareto. Dans ces définitions, nous supposons que le problème est un problème de minimisation.

2.5.1.1. La domination de Pareto

Soient $u = [u_1, u_2, \dots, u_n]^t$ et $v = [v_1, v_2, \dots, v_n]^t$ deux vecteurs de solutions. Nous disons que le vecteur u domine v si et seulement si :

$$\forall i \in \{1, 2, \dots, k\}, f_i(u) \leq f_i(v) \wedge \exists i \in \{1, 2, \dots, N\}, f_i(u) < f_i(v) \quad (2.8)$$

Dans le processus d'optimisation multi-objectif, le concept de dominance du Pareto est utilisé afin de comparer et ranger le vecteur de variables des décisions : u domine v dans le sens du Pareto, signifie que $F(u)$ est mieux que $F(v)$ pour tous les objectifs, et il y a au moins une fonction objectif pour laquelle $F(u)$ est strictement meilleure que $F(v)$ (Goldberg, 1989).

2.5.1.2. Pareto optimal

Une solution Pareto optimal est une solution qui se trouve sur le front Pareto appelée aussi : solution efficace, non inférieure ou non dominée.

Soit : $[x_1, x_2, \dots, x_n]^t$ un vecteur de décision ou solutions avec x appartient à X (l'espace de recherche réalisable). x est dit Pareto optimal, s'il n'existe pas une solution y domine x .

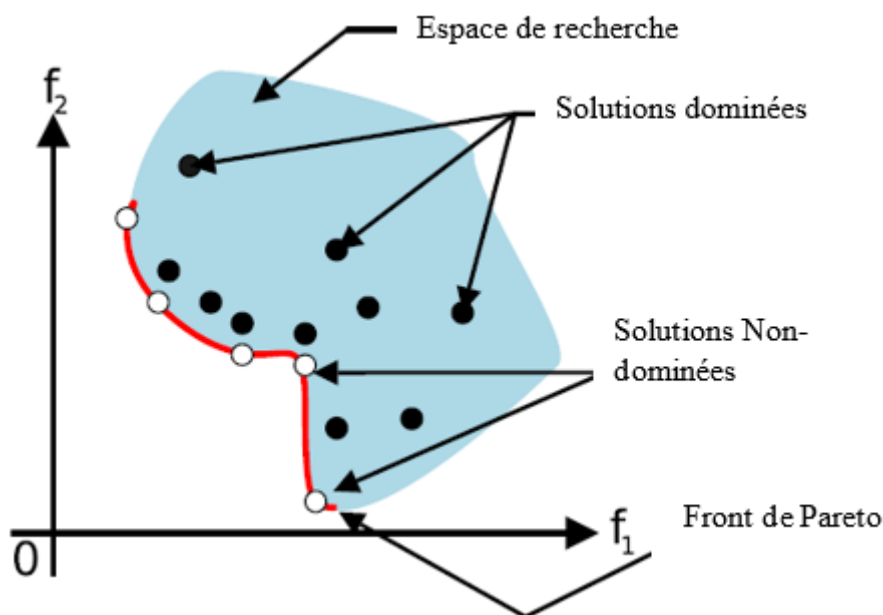


Figure 2.10- Concepts principaux de l'approche avec Pareto.

2.5.1.3. Front de Pareto

Le front (frontière) de Pareto est l'ensemble des solutions optimales de Pareto (les points non dominés). Le front de Pareto appelé aussi surface de compromise ou l'ensemble des solutions efficaces. La figure (2.10) est un exemple du front de Pareto pour un problème de minimisation de deux fonctions 'objectifs'.

2.5.2. Méthode VEGA (Vector Evolutionary Genetic Algorithm)

VEGA (Vector Evolutionary Genetic Algorithm) est un algorithme génétique multi-objectif proposé par Shaffer (1985). Cet algorithme consiste à évaluer un vecteur d'objectifs. Chaque élément du vecteur représente une fonction 'objectif'. VEGA constitue une extension de l'algorithme génétique standard GAs. Étant donné que VEGA est une méthode d'optimisation multi-objectif, par conséquent un certain nombre d'objectifs (par exemple M) doivent être optimisées, Schaffer a divisé au hasard la population à chaque génération en M sous-populations égales. Chaque sous-population reçoit une fitness basée sur une fonction objectif différente. De cette façon, chacune des M fonctions est utilisée pour évaluer certains membres de la population. La figure (2.11) illustre le système d'évaluation de fitness pour cinq fonctions objectives. La population est divisée en cinq divisions égales. Chaque individu de la 1^{ère} sous-population est attribuée une fitness basée uniquement sur la 1^{ère} fonction objectif, tandis que chaque individu de la 2^{ème} sous-population reçoit une fitness basée seulement sur la 2^{ème} fonction, et ainsi de suite.

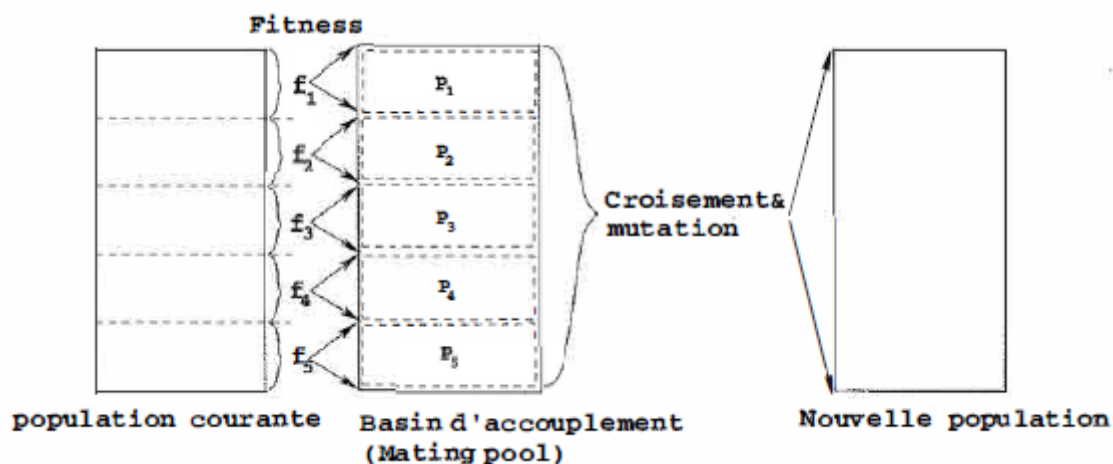


Figure 2.11- Principe de fonctionnement de VEGA (Deb, 2001).

Afin de réduire le biais positif dans la population, l'auteur a pensé de mélanger la population avant qu'elle ne soit répartie en sous-populations égales. Une fois que chaque solution reçoit

une fitness, l'opérateur de sélection restreint entre les solutions de chaque sous-population est appliqué jusqu'à ce que la sous-population soit remplie. Ensuite, la population qui s'est formée par l'ensemble des sous-populations est soumis aux opérateurs de reproduction (croisement et mutation). Étant donné que l'opérateur de sélection est appliqué dans chaque sous-population à part, cela conduit à de bonnes solutions correspondant à une seule fonction d'objectif à chaque exécution. Schaffer a utilisé l'opérateur de sélection proportionnelle (Goldberg, 1989).

2.5.3. Méthode lexicographique

Dans cette approche, la recherche est effectuée selon un ordre de préférence donné à un objectif défini par le décideur (Fishburn, 1974). Cet ordre définit le niveau de signification des objectifs. Supposons que les indices des fonctions objectifs indiquent également leur priorité, la fonction f_1 a la plus grande priorité. Ensuite, un ensemble de problèmes mono-objectifs sont résolus de façon séquentielle en tenant compte de la meilleure solution de chaque fonction objectif déjà traitée. Alors, le premier problème résolu sera formulé de la manière suivante :

$$\text{Min} f_1(x), \text{ avec } x \in S \quad (2.9)$$

Après que la 1^{ère} fonction objectif soit traitée, l'algorithme passe à la 2^{ème} en prenant la meilleure solution de l'objectif précédent comme une contrainte d'égalité. Par conséquent le problème s'écrit mathématiquement comme suit :

$$\text{Min} f_2(x), \text{ avec } x \in S \text{ et } f_1(x) = f_1(x_1^*) \quad (2.10)$$

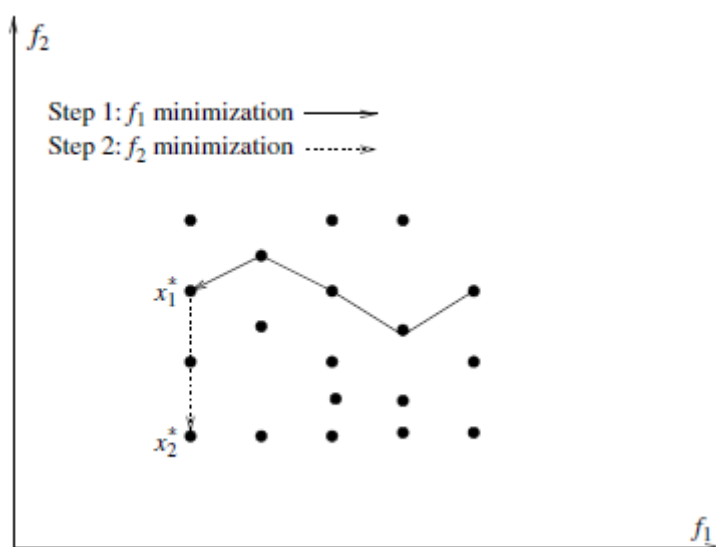


Figure 2.12- Principe lexicographique dans POM (Talbi, 2009)

Les mêmes critères d'arrêt et le processus sont itérés jusqu'à ce que le traitement de la dernière fonction soit traité. La figure 4.15 illustre le déroulement de cette procédure dans le cas d'un problème à deux objectifs.

2.5.4. Méthodes basées sur l'approche de Pareto

Les approches basées sur la dominance utilisent le concept de dominance dans l'affectation de fitness, contrairement aux autres approches qui utilisent une fonction de scalarisation ou traitent les différents objectifs séparément. Cette idée introduit initialement dans les AG par Goldberg (1989). Le principal avantage des approches basées sur la dominance est qu'elles n'ont pas besoin de transformer les POM en un problème mono-objectif. En une seule exécution, ils sont capables de générer un ensemble diversifié de solutions Pareto optimales. Depuis le début des années 90, les intérêts concernant les POM avec l'approche Pareto poussent toujours. La plupart des approches de Pareto utilisent des algorithmes évolutionnaires d'optimisation multi-objectif. On peut mentionner les algorithmes couramment utilisées : NSGA-II (nondominated sorting genetic algorithm), and SPEA2 (strength Pareto evolutionary algorithm), MOGA (Multiple Objective Genetic Algorithm),..., etc. Les métaheistiques à base de population semblent particulièrement adaptées pour résoudre les POM, car ils traitent simultanément un ensemble de solutions. Cela permet de trouver plusieurs membres de l'ensemble optimal de Pareto dans une seule séquence de l'algorithme. En outre, les métaheuristiques de Pareto sont moins sensibles à la forme du front de Pareto (Continuité, convexité).

2.5.5. Méthode MOGA (Multiple Objective Genetic Algorithm)

MOGA est un algorithme d'optimisation multi-objectif qui a été proposé par Fonesca et Fleming (1993). MOGA est le premier algorithme génétique multi-objectif qui a utilisé la notion de dominance pour classer les individus de la population. MOGA diffère d'une GA standard dans la manière d'évaluation de la population. (C'est-à-dire, comment affecter la valeur de fitness à chaque individu). Le reste de l'algorithme est le même que celui d'un GA classique. Le principe de la méthode est comme suit : D'abord, la population est classée selon la notion de dominance. Toutes les solutions non-dominées auront le rang 1 (figure 2.13.b), celles qui sont dominées seulement par les membres du rang 1, sont attribuées d'un rang calculé par l'équation suivante :

$$r(i) = 1 + n_i \quad (2.11)$$

Où $r(i)$ est le rang de la solution i et n_i est le nombre de solutions qui dominant la solution i

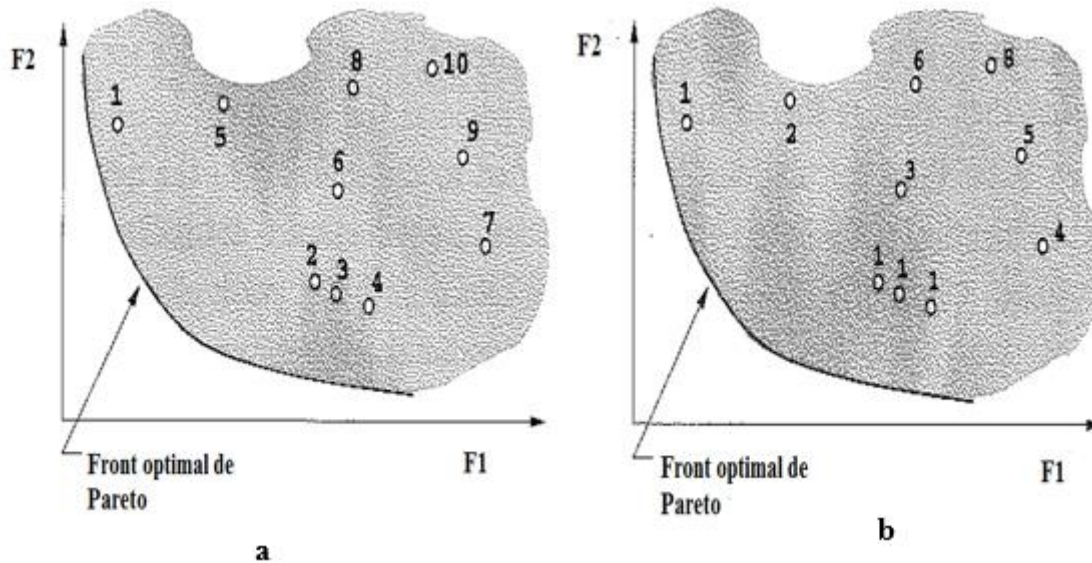


Figure 2.13- a) Exemple d'un problème de minimisation à deux objectifs,

b) Concept de tri de non-dominance par le rang utilisé par MOGA (Deb K. , 2001).

De cette façon, les solutions non-dominées reçoivent un rang égal à 1 car aucune solution dominerait une solution non dominée dans une population. Par conséquent, dans n'importe quelle population, il doit y avoir au moins une solution avec un rang égal à un et le rang maximum d'un membre de la population ne peut être supérieur à N (la taille population).

Une fois que le classement est effectué, une fitness brute est attribuée à chaque solution en fonction de son rang. Pour ce faire, d'abord, les rangs sont triés en ordre croissant. Ensuite, une fitness brute est attribuée à chaque solution en utilisant une fonction linéaire. Par la suite, une nouvelle affectation basée sur l'influence de la proximité des individus appelé fonction de partage (sharing function en anglais), cette dernière conduit à la détermination d'une fitness appelé fitness partagée (shared fitness). Cette technique encourage la diversité dans les solutions non-dominées.

2.5.5.1. Le processus d'évaluation de la fitness selon MOGA

Étape n°1 : Trier les solutions de la population selon la notion de dominance et donner le rang 1 à toutes les solutions non-dominées. Donner une valeur au paramètre σ_{share} ,

Étape n°2 : Déterminer n_i de chaque solution (n_i représente le nombre d'individus qui domine la solution i). Déterminer le rang ($r(i)$) de chaque solution i en utilisant l'équation suivante : $r(i) = 1 + n_i$,

Étape n°3 : Calculer la valeur de fonction fitness moyenne pour chaque individu du premier jusqu'au dernier rang en utilisant l'équation :

$$F_i = N - \sum_{k=1}^{r_i-1} u(k) - 0.5(u(r_i) - 1) \quad (2.12)$$

Étape n°4 : Pour chaque rang, calculer $d_{i,j}$ et nc_i (les deux paramètres représentent respectivement la distance euclidienne entre deux points et le nombre de niche de chaque individu) en utilisant les équations ci-dessous

$$d_{ij} = \sqrt{\sum_{k=1}^M \left(\frac{f_k^i - f_k^j}{f_k^{max} - f_k^{min}} \right)^2} \quad (2.13)$$

$$Sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}} \right)^\alpha, & \text{if } d \leq \sigma_{share}; \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

$$nc_i = \sum_{j=1}^{u(r_i)} Sh(d_{ij}) \quad (2.15)$$

Étape n°5 : Calculer la valeur de fitness partagée (shared fitness) en utilisant

$$F'_j = F_j / nc_j. \quad (2.16)$$

Pour préserver la même fitness moyenne, la fitness partagée (shared fitness) doit être normalisée comme suit :

$$F'_j = \left(\frac{F_j U(r)}{\sum_{k=1}^{u(r)} F'_k} \right) \times F'_j \quad (2.17)$$

Le principal avantage de MOGA est qu'il est efficace et relativement facile à mettre en œuvre. Néanmoins, sa performance dépend fortement d'une sélection appropriée du paramètre de partage (sharing).

2.5.6. Méthode NSGA (Non dominated Sorting Genetic Algorithm)

N. Srinivas et Kalyanmoy Deb (1994) ont proposé une autre variante de l'approche de Goldberg appelée NSGA (Non dominated Sorting Genetic Algorithm). Cet algorithme est basé sur plusieurs couches de classifications des individus. Afin de maintenir la diversité de la population, la méthode de fonction de partage (sharing function) introduite dans la méthode de

MOGA, a été utilisée. Par conséquent, le processus de NSGA débute par trier les individus selon la notion de dominance. Les solutions non-dominées sont classées dans le front n°1. Ensuite, ce groupe de solutions classées est ignoré et une autre couche de solutions non-dominées sera insérée dans le front n°2 et ainsi de suite. Le processus se poursuit jusqu'à ce que tous les individus de la population soient classés. De toute évidence, les pires solutions sont celles qui appartiennent au dernier front.

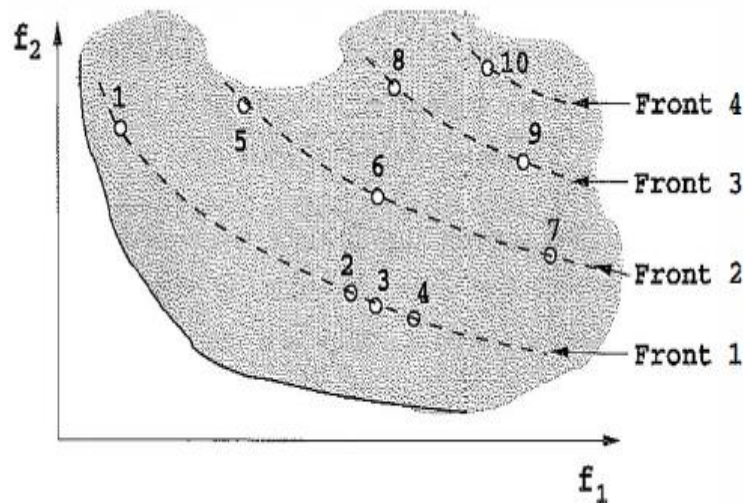


Figure 2. 14- Principe de tri en front (NSGA) (Deb K. , 2001).

La figure (2.14) illustre le principe de classement selon la notion de non-dominance utilisée par NSGA où on voit que la population est classée en quatre fronts distincts. En examinant la figure (2.14), il est clair que les solutions du premier front sont les meilleures en termes de proximité du véritable front optimal de Pareto. Ainsi, il est logique d'assigner la meilleure fitness aux solutions du meilleur front non dominé et ensuite affecter une fitness progressivement pire aux solutions de fronts supérieurs. Dans ce qui suit, nous décrivons l'affectation exacte NSGA Procédure adoptée par Srinivas et Deb (1994).

La procédure d'affectation de fitness commence à partir du premier ensemble non-dominé et procède successivement à des ensembles dominés. Chaque solution i du premier front est affectée d'une fitness égale à $F = N$ (N est la taille de la population). Cependant, afin d'atteindre le deuxième objectif, la diversité des solutions dans un front doit également être maintenue. Dans la méthode NSGA, la méthode de fonction de partage (sharing function) est utilisée à cette fin. La méthode de la fonction de partage est utilisée front par front en commençant toujours par le meilleur, c'est-à-dire le front n°1. Le rôle de fonction de partage est de pénaliser les solutions qui sont très proches en donnant une nouvelle fitness appelée fitness partagée.

Pour passer au deuxième front, nous notons fitness partagée minimale dans le premier front, puis une valeur de fitness légèrement inférieure à cette valeur est attribuée aux solutions du front n°2. Cela garantit qu'aucune solution dans le premier front n'a une fitness partagée pire que la fitness assignée aux solutions du front n°2. Cette procédure se poursuit jusqu'à ce que toutes les solutions reçoivent une fitness partagée.

2.5.7. Méthode NPGA (Niche Pareto Genetic Algorithm)

Horn et ses collègues (1994) ont proposé un algorithme génétique multi-objectif basé sur le concept de dominance. Cette méthode diffère des méthodes précédentes en opérateur de sélection. La méthode NPGA (Niche Pareto Genetic Algorithm) utilise la sélection de tournoi binaire. Le principe de cette méthode réside dans le processus de sélection. Durant la sélection de tournoi, deux individus i et j sont choisis aléatoirement de la population parents p . Ensuite, ils sont comparés à une sous-population T d'une taille $t_{dom} (\ll N)$. Après, les deux solutions i et j sont comparées à l'ensemble des solutions de la sous-population en utilisant le concept de dominance. Si un des deux domine toutes les solutions de la sous-population, par contre l'autre solution est dominée au moins par une solution de la sous-population, alors le premier sera choisi. Maintenant, si les deux solutions sont soit dominées par au moins une solution ou non-dominées par aucune solution de sous-population. Les deux seront vérifiées avec la population enfants qui est vide au début de l'algorithme. Chaque solution est placée dans la population enfants et son count niche est calculé. La solution avec le count niche le plus petit sera choisi. Ensuite deux autres solutions seront choisies au hasard pour sélectionner une autre solution qui va participer à la reproduction en leur appliquant les opérateurs de croisement et de mutation.

2.5.8. Méthode PAES (Pareto Archived Evolution Strategy)

La méthode (PAES) a été proposée par Joshua et David (2000). PAES consiste en une stratégie d'évolution (1 + 1) (c'est-à-dire un parent unique qui génère un seul enfant) en combinaison avec un archive historique qui enregistre des solutions non dominées précédemment trouvées. Le processus de PAES est très simple, il commence par l'initialisation d'un seul chromosome (la solution courante) qui est évalué en utilisant le vecteur de fonctions 'objectifs'. Ensuite, l'opérateur de mutation est appliqué à cette solution pour donner une nouvelle solution. Les deux solutions (solution courante et la copie mutée) sont comparées en utilisant le concept de dominance par rang (dominance rank). L'acceptation est simple si une solution domine l'autre. Mais dans le cas où aucune solution ne domine, la nouvelle mutée est comparée aux solutions

non-dominées de l'archive. Si encore aucune solution ne domine l'autre, c'est la solution moins encombrée qui gagne.

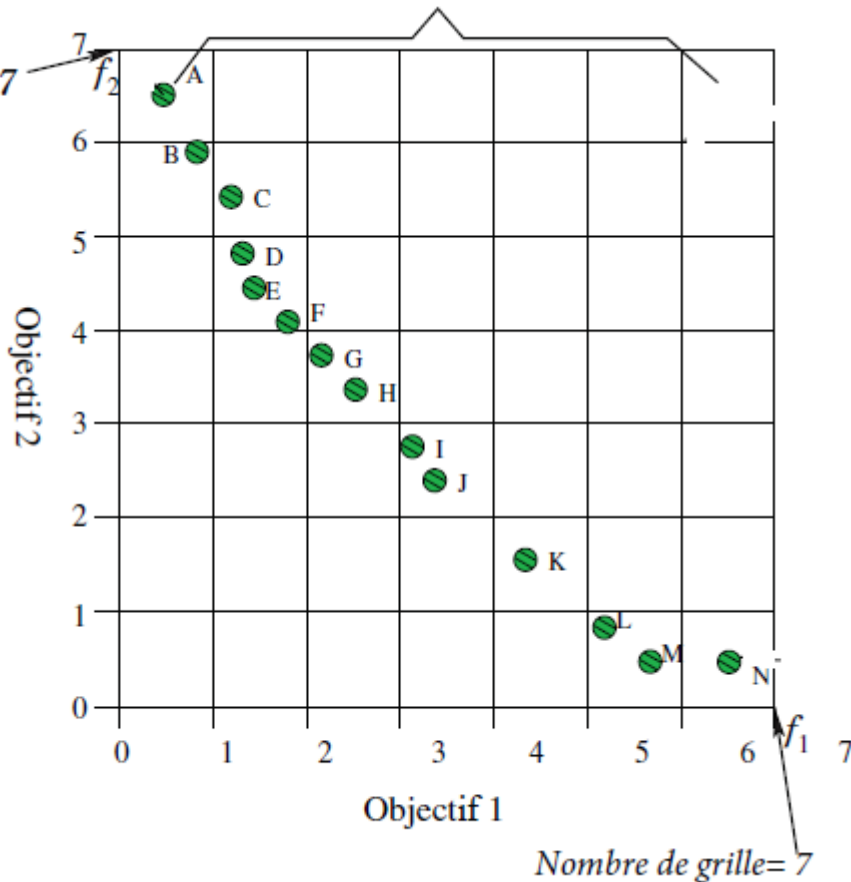


Figure 2. 15-Illustration graphique de la grille adaptative utilisée par PAES (Coello et al, 2007).

Cette procédure est utilisée pour garder la diversité, elle divise l'espace objectif de manière répétitive. Chaque solution est placée dans un certain emplacement de grille en fonction des valeurs de ses objectifs (qui sont utilisés comme "coordonnées" ou "situation géographique"). L'archive sert deux objectifs distincts. Tout d'abord, il stocke et met à jour toutes les solutions non-dominées générés. Deuxièmement, durant le processus, il est utilisé comme un paramètre d'aide au choix précis entre la solution actuelle et au vecteur des solutions stockées en agissant comme une approximation du front de solutions non-dominées. Sans ce processus, l'algorithme est incapable de différencier entre bonnes et mauvaises solutions

Une carte de cette grille est maintenue, indiquant le nombre de solutions qui résident dans chaque emplacement de grille. La figure 2.19 montre une illustration graphique de la grille adaptative de PAES.

2.5.9. Méthode SPEA (Strength Pareto Evolutionary Algorithm)

Zitzler et Thiele (1998) ont proposé un algorithme évolutif élitiste, qu'ils ont appelé SPEA (Strength Pareto Evolutionary Algorithm). Cet algorithme introduit le concept de l'élitisme en maintenant explicitement une population externe \bar{P} . Cette population stocke un nombre fixe des solutions non dominées. À chaque génération, des solutions non dominées récemment trouvées, sont comparées à la population extérieure existante et les solutions résultantes non dominées sont préservées. SPEA fait plus que préserver les élites ; il utilise aussi ces élites pour participer aux opérations génétiques avec les membres de la population actuelle dans l'espoir d'influencer la population à s'orienter vers le front de Pareto.

Cet algorithme commence par une population créée au hasard P_0 de taille N et une population externe vide \bar{P}_0 avec une capacité maximale de \bar{N} . À chaque génération, les meilleures solutions non dominées de la population P_t sont copiées dans la population externe \bar{P}_t . Par conséquent, les solutions dominées de la population externe sont supprimées.

Si ce processus se poursuit plusieurs générations, il existe un risque que la population externe soit surpeuplée avec des solutions non dominées. C'est pourquoi, la taille de la population externe a été limitée à (\bar{N}) . Autrement dit, lorsque la taille de la population extérieure est inférieure à (\bar{N}) , toutes les élites sont conservées dans la population externe. Cependant, lorsque la taille est supérieure à (\bar{N}) pas toutes les élites peuvent être stockées dans la population externe. Pour choisir les membres qui doivent être supprimés, les auteurs ont proposé une technique de regroupement (clustering method).

Une fois que les nouvelles élites sont conservées pour la prochaine génération, l'algorithme attribue à l'ensemble des individus des fitness (appelée Strength). Pour les individus de la population \bar{P}_t , il utilise l'équation (2.18), par contre pour les individus de la population P_t , il utilise l'équation (2.19)

$$S_i = \frac{n_i}{N + 1} \quad (2.18)$$

$$F_j = 1 + \sum_{i \in \bar{P}_t} S_i \quad (2.19)$$

Après que l'affectation de fitness soit terminée, le processus se reprend en commençant par la sélection de type tournoi de tous les individus (c'est-à-dire de la population $P_t \cup \bar{P}_t$) pour

former les parents de la population P_{t+1} . Donc, les opérateurs de reproduction (croisement et mutation) sont appliqués aux individus sélectionnés afin de générer la nouvelle.

Ceci représente seulement une seule génération, le processus se continue jusqu'à un critère d'arrêt soit atteint.

2.5.9.1. Algorithme SPEA

Étape n°1 : Trouver le meilleur ensemble F1 (P_t) non dominées et copier les dans la population externe \bar{P}_t .

Étape n°2 : Trier les individus de la population externe modifiée selon le concept de non dominance en supprimant les solutions dominées.

Étape n°3 : Si $|\bar{P}_t| > \bar{N}$ utilisez la technique de clustering pour réduire la taille à \bar{N} . Sinon, passer à l'étape suivante.

Étape n°4 : Attribuer une fitness (Strength) à chaque solution $i \in \bar{P}_t$ en utilisant l'équation (2.18). Ensuite, attribuer une fitness (Strength) à chaque solution $j \in P_t$ en utilisant l'équation (2.19).

Étape 5 : Appliquer une sélection de tournoi en se basant sur les fitness attribuées dans l'étape précédente et les opérateurs (croisement et mutation) pour une nouvelle population P_{t+1} de taille (N) à partir de la population ($\bar{P}_t \cup P_t$) de la taille ($\bar{N} + N$).

Cet algorithme se poursuit jusqu'à ce qu'un critère d'arrêt soit satisfait.

2.5.9.2. Algorithme de regroupement (Clustering algorithm)

La technique de Clustering est un concept de l'élitisme qui réduit la taille de la population externe \bar{P}_t de taille N^* afin de ne pas dépasser la taille fixée au début de l'algorithme \bar{N} .

Au début, chaque solution dans \bar{P}_t est considérée comme résidant dans un groupe (cluster) séparé. Ainsi, il existe initialement \bar{N} clusters. Par la suite, les distances entre chaque paire de groupe (cluster) sont calculées. En général, la distance d_{12} entre deux groupes C_1 et C_2 est définie comme la distance euclidienne moyenne de toutes les paires de solutions ($i \in C_1$ and $j \in C_2$).

$$d_{12} = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j) \quad (2.20)$$

La distance $d(i, j)$ peut être calculée dans l'espace de décision ou des objectifs, les auteurs de la méthode préfèrent l'espace des objectifs. Une fois que toutes les distances sont calculées,

les deux groupes qui ont la plus petite distance, sont combinés pour donner un seul groupe (cluster). Ce processus de fusion des groupes se poursuit jusqu'à ce que le nombre de groupes dans la population externe est réduit à \bar{N} . Comme les groupes fusionnés contiennent plus qu'une solution, alors un tri de non-dominance est effectué pour choisir une seule solution et la solution dominée sera supprimée.

2.5.10. Méthode NSGA II (Non-Dominated Sorting Genetic Algorithm II)

Deb et ses étudiants ont suggéré un algorithme génétique élitiste basé sur un tri non-dominé appelé NSGA-II (Deb et al., 2000). NSGA-II utilise également un mécanisme qui préserve la diversité. Dans la plupart des aspects, cet algorithme n'a pas beaucoup de similitude avec le NSGA original, mais les auteurs ont gardé le nom NSGA-II pour mettre en évidence son origine.

NSGA-II commence d'abord par créer aléatoirement une population initiale P_0 de N individus. Cette population est triée en différents fronts selon la notion de non-dominance. Ensuite, les individus de chaque front sont attribués une fitness que représente la distance d'encombrement (Crowding distance). Après avoir attribué à tous les individus deux qualités qui sont le numéro du front (le rang) et la valeur de distance d'encombrement, l'algorithme commence la génération de la population enfants en utilisant la sélection par tournoi à base des deux qualités susmentionnées. Autrement dit, deux individus appartiennent aux deux rang différent, celui qui a le plus petit rang gagne. Par contre si les deux individus appartiennent au même rang, celui qui a la plus grande distance d'encombrement gagne. Les individus sélectionnés sont soumis aux opérateurs de reproduction (croisement et mutation) pour créer la population enfant Q_0 de N individus. Les deux populations sont combinées pour former une population R_t de taille $2N$. Ensuite, la même procédure de classement qui a été faite à la population initiale se refait à cette population de $2N$ individus pour former une seule population de N individus. Alors, l'algorithme commence à construire la population P_{t+1} par les meilleurs individus, c'est-à-dire les individus appartenant aux fronts de rang les plus faibles. Pour le dernier front, il y a généralement plus de solutions que de places restantes dans la nouvelle population P_{t+1} . Les individus sont alors triés selon leur distance d'encombrement et par ordre décroissant. Ce scénario est illustré à la figure (2.16).

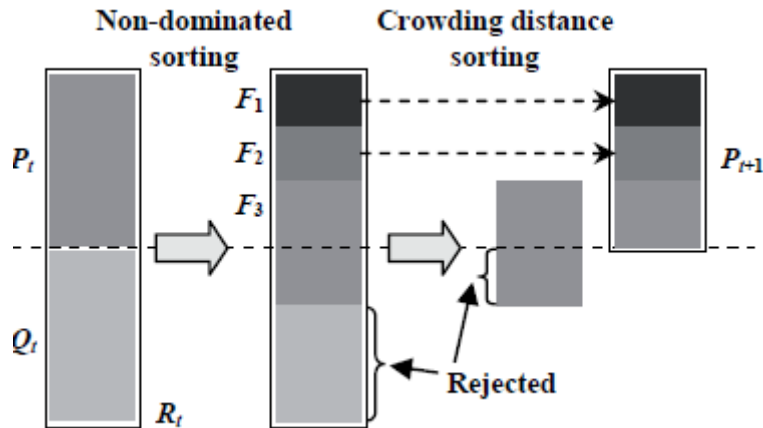


Figure 2.16- Schéma de la procédure NSGA-II (Deb et al., 2000).

Dans ce qui suit, nous présentons l'algorithme étape par étape. Dans un premier temps, une population aléatoire P_t de taille N est créée. Cette dernière est classée en différents fronts de non-domination. Chaque solution est affectée d'une fitness égale à son front de non-domination (1 est le meilleur front). Ensuite, une sélection de tournoi basée sur la notion de non-dominance et la distance d'encombrement est effectuée. Les individus sélectionnés sont soumis aux opérateurs de reproduction (croisement et mutation) pour créer une population enfants Q_t de taille N

La procédure de NSGA-II est décrite dans ce qui suit :

2.5.10.1. Algorithme NSGA-II

Étape n°1 : Combiner les deux populations (parents et enfants) pour créer $R_t = P_t \cup Q_t$

Effectuer un tri non-dominé R_t et identifier les différents fronts : $F_i, i= 1, 2, \dots$, etc.

Étape n°2 : Définir nouvelle population $P_{t+1} = \emptyset$

Jusqu'à $|P_{t+1}| + |F_i| < N$, effectuer $P_{t+1} = P_{t+1} \cup F_i$ et $i=i+1$.

Étape n°3 : Si $|P_{t+1}| + |F_i| > N$, effectuer la sélection basée sur la distance d'encombrement (Crowding distance) (Noter par $<_c$) pour le front i afin de garder seulement un nombre égale à $(N - |F_{i-1}|)$ individus.

Étape 4 : Créer une population enfants Q_{t+1} de P_{t+1} en utilisant un opérateur de sélection de tournoi basé sur la notion de dominance et la distance d'encombrement. Les individus sélectionnés soumis aux opérateurs de reproduction (croisement et mutation).

Les étapes décrites ci-dessus représente seulement une seule génération, donc le processus se poursuit jusqu'à un des critères d'arrêt soit atteint.

2.5.10.2. Opérateur de sélection par tournoi basé sur le concept d'encombement

L'opérateur de comparaison surpeuplé ($<_c$) compare deux solutions et renvoie le gagnant du tournoi. Dans cette méthode NSGA II chaque solution i possède deux attributs :

- Un rang de non-domination dans la population
- Une distance d'encombement d_i (crowding distance) dans la population.

2.5.10.2.1. Distance d'encombement (Crowding distance)

Distance d'encombement est une mesure qui estime de la densité des solutions entourant une solution particulière i dans une population, nous prenons la distance moyenne de deux solutions de chaque côté de la solution i dans l'espace des objectifs. Cette quantité sert d'estimation du périmètre de l'hypercube formé en utilisant les voisins les plus proches comme sommets.

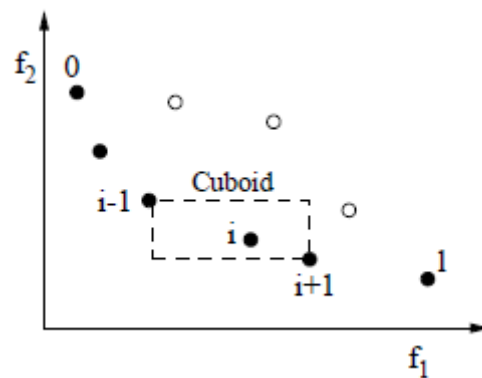


Figure 2.17- Le concept de distance d'encombement
(Crowding distance) (Deb et al., 2000).

La figure (2.17) illustre la distance d'encombement de la solution i . La distance d'encombement se calcule par l'équation (2.21). Sur la base de ces deux attributs, nous pouvons définir l'opérateur de sélection de tournoi à base d'encombement comme suit :

- **Définition :** Opérateur de sélection de tournoi basé sur la distance d'encombement : une solution i gagne un tournoi avec une autre solution j si l'une des conditions suivantes est vraie :
 1. Si le rang de la solution i est meilleur que celui de la solution j , c'est-à-dire $r_i < r_j$.
 2. Si elles ont le même rang, mais la solution i a une meilleure distance d'encombement que la solution j , c'est-à-dire $r_i = r_j$ et $d_i > d_j$

$$d_j = \sum_{m=1}^M d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{max} - f_m^{min}} \quad (2.21)$$

Où f^{max} et f^{min} sont respectivement la valeur maximale et minimale de la fonction f_m

$f_m^{I_{j+1}^m}$ est la valeur de la fonction objectif m de la solution j qui appartient au front I

L'équation ci-dessus donne la valeur de la distance d'encombrement de la solution j en fonction de ses deux voisines ($j+1$) et ($j-1$) pour la fonction m . Si, le problème contient deux objectifs seulement, on calcule la valeur de la distance d'encombrement pour la 1^{ère} fonction. Ensuite, on recalcule la distance d'encombrement pour la 2^{ème} fonction en ajoutant la valeur de la 1^{ère} fonction. Et ainsi de suite pour le reste des fonctions si nous avons plus de deux fonctions objectifs.

2.5.11. Méthode MOPSO (Multiobjective Particles Swarm Optimization)

Plusieurs études de la méthode PSO (Particles Swarm Optimization) sur des problèmes multi-objectif sont apparues et de nouvelles variantes spécialisées de la méthode ont été développées (Reyes-Sierra et Coello, 2006) après les résultats prometteurs obtenus par PSO dans l'optimisation des problèmes mono-objectif. Il n'est pas possible d'exposer toutes les variantes de PSO qui ont traité les problèmes multiobjectifs dans un espace limité. Dans cette partie du chapitre, nous allons essayer de présenter brièvement plusieurs variantes en accordant une importance particulière à celle développée par Coello et al (2004). Cette dernière appelée MOPSO (Multiple Objective with Particles Swarm Optimization), c'est cette méthode que nous avons utilisée dans notre thèse.

Pour les variantes de PSO multiobjectif, on peut distinguer deux approches fondamentales : La première approche consiste en algorithmes qui considèrent chaque fonction objectif séparément. Dans ces approches, chaque particule est évaluée uniquement pour une fonction objectif à la fois et la détermination de meilleures solutions est réalisée de la même manière que le cas d'optimisation mono-objectif. Le défi principal dans cette approche est la manipulation appropriée de l'information provenant de chaque objectif pour guider les particules vers les solutions optimales de Pareto. La deuxième approche consiste en algorithmes basés sur la notion de Pareto. Ces algorithmes produisent les meilleurs positions non-dominées (appelés leaders) qui sont utilisées pour guider les particules. Dans ces approches, la détermination des

leaders n'est pas facile, car il peut y avoir de nombreuses solutions non dominées dans le voisinage d'une particule. Mais un seul doit être choisi pour participer à la mise à jour de la vitesse. Ainsi, deux questions primordiales doivent être prises en considération, lors de l'extension de PSO pour les problèmes multi-objectifs.

- Comment sélectionner les particules utilisées comme guide (*leader*) pour d'autres particules afin de favoriser les solutions non dominées ?
- Comment conserver les solutions non dominées trouvées par le processus de recherche, durant toutes les itérations précédentes ?

En ce qui concerne la sélection de leader, les algorithmes de la 1ère approche ont gardé la sélection du guide telle que celle de la méthode standard de PSO, (c'est-à-dire, pbest et gbest, car les fonctions objectifs sont agrégées en une seule). Par contre les approches basées sur la notion de Pareto ont considéré toute solution non dominée, localement ou globalement (en se basant sur la notion de non-dominance) comme nouveau leader. Ensuite, une de ces solutions doit être choisie. De cette façon, une mesure de qualité est nécessaire pour la sélection des leaders. Évidemment, une telle caractéristique peut être définie de plusieurs manières différentes. L'une des méthodes qui peuvent être utilisées est la densité de voisinage. Figure (2.18) illustre les différentes techniques de densité de voisinage.

La façon la plus simple de sauvegarder les solutions non dominées trouvées le long du processus de recherche est d'utiliser un archive externe. Toutefois, ce choix n'est pas toujours valable, puisque la taille de l'archive des solutions non-dominées peut dépasser la taille de l'essaim. La solution à ce problème varie d'une approche à l'autre. Cela va être évoqué dans la suite de cette partie.

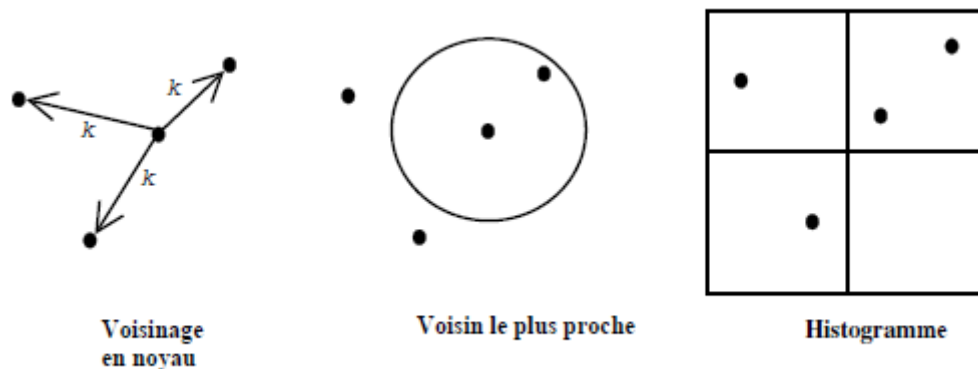


Figure 2.18- Stratégie de maintien de la diversité (Talbi, 2009).

2.5.11.1. Les différentes extensions de PSO

2.5.11.1.1. La métaphore de l'essaim de Ray et Liew (2002)

Cet algorithme utilise également la dominance de Pareto pour sélectionner les solutions non-dominées qui doivent être insérées dans l'archive externe. Pour maintenir la diversité, les auteurs ont utilisé le concept de distance d'encombrement (Crowding distance) utilisé dans les algorithmes évolutionnaires tels que NSGA II et PASE.

2.5.11.1.2. L'algorithme de K. E. Parsopoulos et M. N. Vrahatis (2002)

Les auteurs de cette méthode combinent tous les objectifs du problème en une seule fonction-objectif. Ils ont utilisé, pour l'agrégation des objectifs, une fonction linéaire. Les auteurs ont présenté trois méthodes d'ajustement des coefficients de pondération ; l'une est basée sur la fixation préalable de ces coefficients, où ils sont déterminés pendant l'exécution, l'autre est basée sur l'agrégation dynamique où ces poids peuvent être modifiés au cours de l'exécution et une troisième appelée approche *bang bang* où les poids peuvent être brusquement modifiés durant l'exécution.

2.5.11.1.3. PSO de voisinage dynamique proposé par Hu et Eberhart (2002)

Dans cet algorithme, un seul objectif est optimisé à la fois en utilisant un schéma similaire à la méthode lexicographique (Fishburn, 1974). L'ordre lexicographique tend à être utile uniquement lorsque peu de fonctions objectives sont utilisées (deux à trois) et il peut être sensible à l'ordre des objectifs. L'idée du voisinage dynamique est nouvelle dans ce contexte.

2.5.11.1.4. J. E. Fieldsend et S. Singh (2002)

J. E. Fieldsend et S. Singh adoptent une structure de données appelée « arbre dominé » pour stocker les solutions non dominées trouvées le long de la recherche. La sélection d'un leader pour une particule de l'essaim, dans cette technique, est basée sur la structure définie par l'arbre

dominé. Un point est, tout d'abord, composé par l'arbre sur la base des relations de dominance, et puis la plus proche particule (dans l'espace des solutions faisables) de ce point est choisie comme leader. D'autre part, un ensemble des meilleurs scores personnels non dominés, pour chaque membre de l'essaim, est maintenu afin d'alimenter l'archive externe.

2.5.11.1.5. Mostaghim et J. Teich (2003)

Dans cette dernière, l'archive externe contenant les solutions non dominées est mis à jour à partir des meilleurs scores locaux (*pbest*). Afin de choisir un leader pour une particule, une valeur sigma est affectée à chaque particule de l'essaim et de l'archive externe. Chaque particule de l'essaim choisi comme leader la solution de l'archive externe dont la valeur sigma est la plus proche de celle de cette particule.

2.5.11.2. Description de la méthode MOPSO

Coello, Toscano Pulido, et Salazar Lechuga (Coello et al, 2004) ont proposé PSO multi-objectif (MOPSO), une approche basée sur la notion de Pareto. Dans MOPSO, les solutions non-dominées trouvées le long du processus de recherche sont stockées dans un archive externe. En outre, l'espace de recherche est divisé en hypercube. Chaque hypercube est attribué une fitness qui est inversement proportionnelle au nombre de particules contenant de cet hypercube. Ensuite, la sélection classique de la roulette est utilisée pour choisir l'hypercube qui contient le leader. En outre de l'archive externe, Les auteurs ont introduit un opérateur de mutation variable. Le détail de la méthode MOPSO est donné ci-dessous.

2.5.11.2.1. L'algorithme principal de MOPSO

Étape n°1 : Initialiser la population POP

a- Pour $i=1$: Max / *MAX = nombre de particules *

b- Initialise POP(i)

Étape n°2 : Initialiser la vitesse de chaque particule : a- Pour $i=1$: MAX,

b- $V(i)=0$

Étape n°3 : Évaluer chaque particule de POP,

Étape n°4 : Stocker les solutions non-dominées dans l'archive REP,

Étape n°5 : Générer des hypercubes de l'espace de recherche exploré jusqu'à présent et localiser les particules en utilisant ces hypercubes comme un système de coordonnées où les coordonnées de chaque particule sont définies selon les valeurs de ses fonctions objectifs.

Étape n°6 : Initialiser la mémoire de chaque particule (cette mémoire sert de guide pour parcourir l'espace de recherche. Cette mémoire est également stocker dans l'archive)

Étape n°7 : Tant que le nombre des itérations n'a pas été atteint :

a) Mettre à jour la vitesse de chaque particule en utilisant l'équation suivante :

$$V(i) = W \times V(i) + R_1(pbest(i) - POP(i)) + R_2(REP(h) - POP(i)) \quad (2.22)$$

b) Calculer la nouvelle position de chaque particule en utilisant l'équation suivante :

$$POP(i) = POP(i) + V(i) \quad (2.23)$$

Où W est un paramètre d'inertie, R_1 et R_2 sont des nombres aléatoire dans l'intervalle $[0, 1]$. $Pbest(i)$ est la meilleure position que la particule a eue. $REP(h)$ est une valeur prise de l'archive, l'indice h est sélectionné de la manière suivante : les hypercubes qui ont plus d'une particule, on leurs attribue une fitness égale un x (x est une valeur > 1 , les auteurs ont pris la valeur de 10) divisé par le nombre de particules incluses dans l'hypercube. Une fois un hypercube, suivant son évaluation, est sélectionné, un leader est aléatoirement choisi.

Étape n°8 : Si la position de solution i , se trouve hors les frontières : 1) Affecter la variable de décision la valeur inférieur ou supérieur ou multiplier V par -1 pour que la recherche soit la direction opposée.

Étape n°9 : Évaluer chaque particule de la nouvelle population POP

Étape n°10 : Mettre à jour le contenu de REP.

Cette mise à jour consiste à insérer toutes les solutions non-dominées en les comparant avec les solutions de l'archive (REP). Si l'archive est complet et il y a assez de solutions non-dominées, un deuxième critère de rétention est appliqué : Les particules situées dans des zones moins peuplées de l'espace de recherche seront prioritaires par rapport à celles qui sont dans des régions peuplées.

Étape n°11 : Lorsque la position de la particule est meilleure que $pbest$ (la position contenue dans sa mémoire), alors $pbest$ prend cette nouvelle position comme suit :

$$PBEST(i) = POP(i) \quad (2.24)$$

2.5.11.2.2. Dépôt externe (Archive externe)

L'objectif principal du dépôt (ou archive) est de sauvegarder les solutions non-dominées trouvées le long du processus de recherche. L'archive externe se compose de deux parties principales : le contrôleur d'archive et la grille (grid).

- Le contrôleur d'archive : Le rôle du contrôleur d'archive est de décider si une certaine solution doit être ajoutée à l'archive ou non. Le processus décisionnel est le suivant : Les solutions non-dominées trouvées à chaque itération sont comparées par rapport au contenu de l'archive. Cette archive au début de l'algorithme est supposée vide. Donc la solution non-dominée de la première exécution sera automatiquement insérée à l'archive (voir cas 1, figure 2.19). Après l'exécution de la première génération, si la nouvelle solution générée est dominée par une des solutions de l'archive, alors la solution est automatiquement rejetée (cas 2, figure 2.19). Si aucune d'elles n'est dominée par l'autre, la solution est stockée dans la liste de l'archive. S'il y a des solutions dans l'archive dominées par cette nouvelle solution, alors ces solutions seront éliminées de l'archive (voir cas 3 et 4, figure 2.19). Enfin, si l'archive atteint sa capacité maximale, la procédure de la grille adaptative est effectuée.
- la grille (grid) : Pour obtenir des fronts de Pareto bien répartis, MOPSO utilise une variation de la grille adaptative proposée dans (Knowles et Corne, 2000). L'idée de base est d'utiliser une archive externe pour stocker toutes les solutions non-dominées. Si la solution non-dominée qui doit être insérée dans l'archive est située en dehors des limites actuelles de la grille, alors cette dernière doit être recalculée et chaque solution doit être relocalisée.

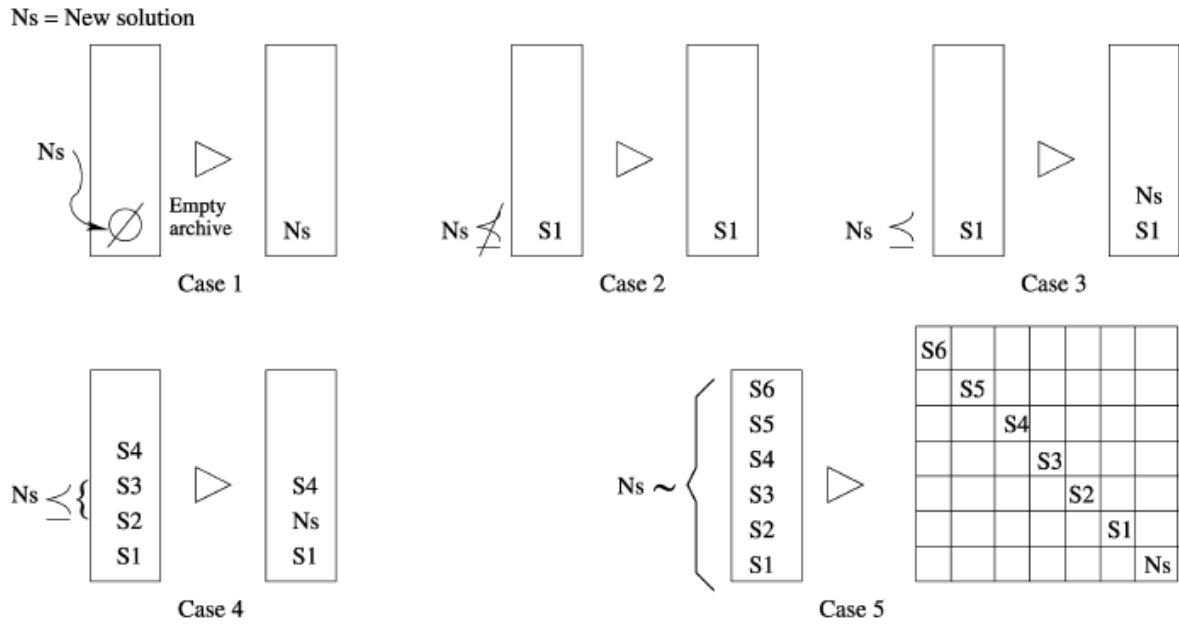


Figure 2.19- Principe de l'archive contrôleur (Coello et al, 2004)

2.5.11.2.3. Utilisation d'un opérateur de mutation

PSO est connu par sa vitesse de convergence très élevée. Cependant, une telle vitesse peut être dans le contexte de l'optimisation multiobjectif, car un algorithme basé sur PSO peut converger vers un faux front de Pareto (c'est-à-dire l'équivalent d'un optimum local). C'est pourquoi les auteurs ont développé un opérateur de mutation variable. Au début du processus de recherche, cet opérateur de mutation affecte toute la population, puis d'une itération à l'autre, le pourcentage de population affecté par la mutation commence à diminuer en utilisant une fonction non linéaire.

2.5.11.2.4. Manipulation des contraintes

Coello et al ont également ajouté un schéma relativement simple pour gérer les contraintes. À chaque fois que deux individus sont comparés, le processus commence par la vérification de leurs contraintes. Si les deux sont réalisables, la notion de dominance qui décide qui gagne. Si l'un des deux est réalisable par contre l'autre non. Le réalisable domine. Si les deux sont non réalisables, celui qui a moins de violation de contrainte gagne.

2.6. Conclusion

Dans ce chapitre, nous avons passé en revue plusieurs méthodes d'optimisation mono-objectif et multi-objectif, en accordant une importance particulière aux méthodes utilisées dans notre thèse ; à savoir, l'algorithme génétique, la méthode d'optimisation par essaim des particules,

ainsi que les méthodes d'optimisation multi-objectif NSGA II et MOPSO. À travers ce chapitre, nous avons constaté que toutes les métaheuristiques que ce soit mono-objectif ou multi-objectif se différencient en trois points : La façon d'attribuer la fitness, la technique de maintien de la diversité et la technique de préserver les élites. Par exemple la méthode MOGA se diffère de GAs dans la manière d'estimation de fitness. Dans MOGA, la fitness d'un individu est représentée dans la valeur de son rang dans la population. Par contre pour GAs, la fitness est tout simplement la valeur donnée par la fonction 'objectif'. NSGA et NSGA II est un autre exemple où la différence se résume dans la manière de maintenir la diversité. NSGA utilise la fonction de partage, or NSGA II utilise le concept d'encombrement. En ce qui concerne l'élitisme, nous pouvons comparer AGs et PAES. AGs ne préserve pas les élites trouvés durant la recherche, par contre PAES utilise un archive externe pour stocker les solutions non-dominées (élites). C'est toujours un point ou plus parmi les trois qui différencient une par rapport à l'autre. D'après (Talbi, 2009), l'innovation dans ce domaine réside dans la modification apportée à l'un des trois points susmentionnés.

3.Évaluation du potentiel de liquéfaction des sols

3.1. Introduction

La liquéfaction est un phénomène qui se produit sous sollicitation sismique. Lors du séisme, les contraintes et déformations créées, dans un dépôt de sol lâche saturé, peuvent lui causer la perte partielle ou totale de sa capacité portante. Dans ces conditions le sol se comporte comme un liquide d'où le terme liquéfié. Les sols susceptibles à la liquéfaction sous chargement cyclique étant limoneux à sableux. La liquéfaction des sables saturés est la cause de la majorité des dégâts subis aux constructions. Les séismes de Nigata au Japon en 1964 et de la ville de Mexico en 1985 sont des exemples pour lesquels la liquéfaction des sables a causé des dommages considérables.

Le phénomène de liquéfaction a été observé en simulant l'effet du tremblement de terre au laboratoire sur une couche de sol en appliquant un chargement cyclique sur des échantillons du sol. Un grand nombre de ces essais ont été réalisés au cours des 30 dernières années.

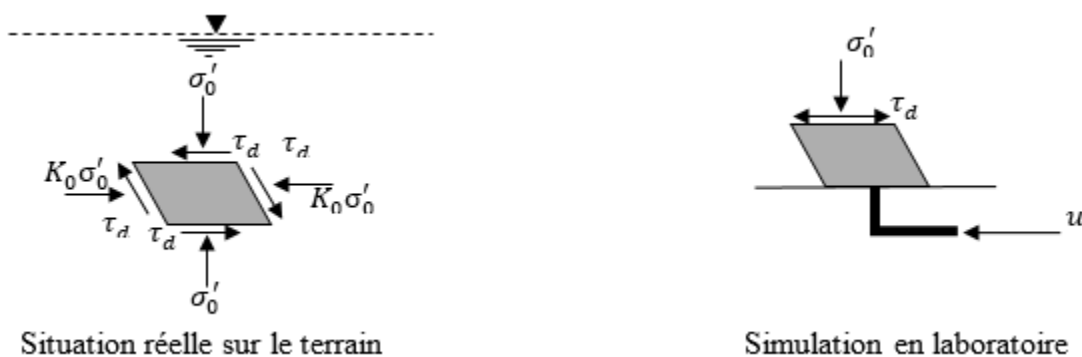


Figure 3.1- Simulation en laboratoire de l'effet d'un séisme sur une couche de sol (Peker, 1984).

3.2. Explication physique du phénomène de liquéfaction

Il existe plusieurs définitions pour expliquer le phénomène de liquéfaction. La majorité des auteurs le définit qu'il est associée aux sols saturés lâches sollicités dans des conditions non drainés. Lorsque le sol est soumis à une augmentation des contraintes, sous un chargement statique ou dynamique, cet effet conduit à une augmentation des pressions interstitielles et par conséquent, diminution des contraintes effectives. Alors Lorsque les contraintes effectives deviennent trop faibles pour maintenir la résistance du sol face aux sollicitations appliquées, le sol se liquéfie.

Selon leur état de compacité (ou indice de densité ID) et de la structure des particules, les sols peuvent se contracter (état lâche) ou se dilater (état dense) lorsqu'ils sont soumis à des contraintes déviatoriques.

Lorsque les contraintes de cisaillement (statiques ou cycliques) appliquées à un sol pulvérulent sont supérieures à la résistance du sol, l'ajout d'une charge au sommet d'une bute ou tout simplement l'érosion lente d'une pente en bordure d'une rivière sont deux exemples qui illustrent une augmentation de la charge quasi statique pouvant conduire à la liquéfaction du sol. D'autre part, l'application de charges cycliques (par exemple, lors d'un dynamitage ou au passage de camions lourds) peut engendrer les contraintes de cisaillement supérieures à la résistance du sol. Ces vibrations peuvent aussi mener à la liquéfaction.

Bowles (1984) a illustré le comportement d'un sol dense et celui d'un sol lâche lors d'un essai de compression triaxiale conventionnelle. La figure (3.2) montre qu'en augmentant la contrainte déviatorique ($\sigma_1 - \sigma_3$), le sol lâche se déforme (ϵ) plus rapidement que le sol dense. En d'autres termes, le sol dense nécessite un chargement plus élevé pour obtenir la même déformation qu'un sol lâche. Cette figure montre aussi que le sol dense va prendre de l'expansion (ϵ_v positif, ϵ_v représente la variation de volume sur le volume initial) lorsque l'on augmente le chargement. Au contraire, le sol lâche va d'abord se contracter (ϵ_v négatif) et ensuite se dilater (ϵ_v positif). Dans ces cas, on observe que la résistance ultime et l'indice des vides convergent vers une valeur unique appelée état critique. L'indice des vides devient alors indépendant de la déformation axiale et des conditions préalables.

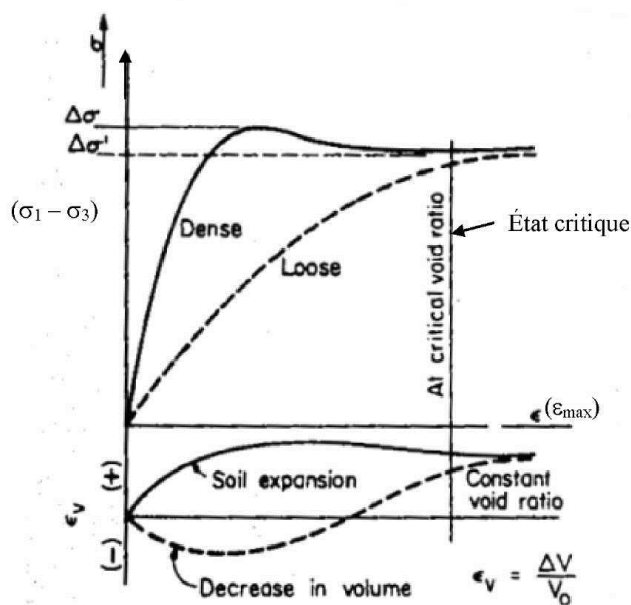


Figure 3.2- Comportement d'un sol dans un état lâche et dense lorsque soumis à un chargement en cellule triaxiale conventionnelle (tirée de Bowles, 1984).

3.3. Sols pulvérulents lâches

Les sols pulvérulents se caractérisent par un indice des vides élevé. Pendant une sollicitation externe, les grains tendent alors à se réarranger vers un état plus dense, en créant une contraction du milieu poreux. Si le sol est saturé et que les conditions sont non drainées, cette contraction peut générer une augmentation des pressions interstitielles (u) et par conséquent une diminution de la contrainte effective ($\sigma' = \sigma - u$).

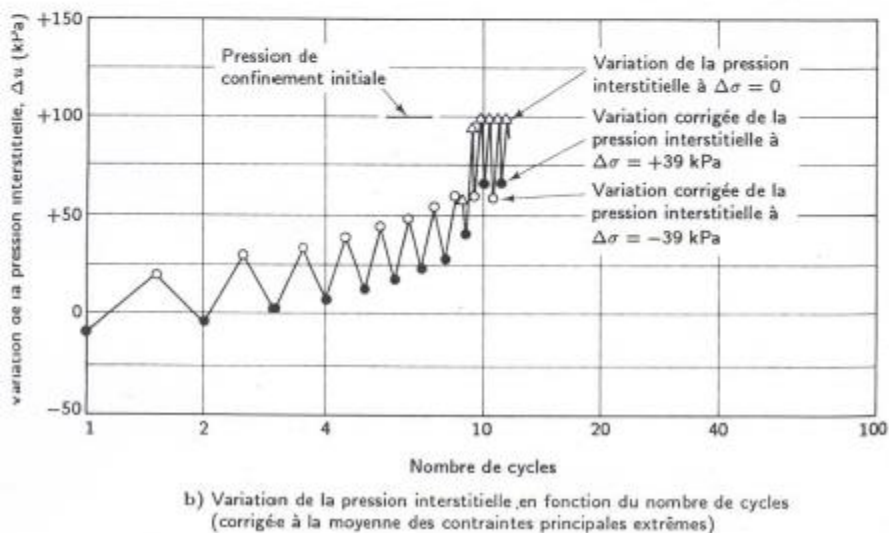
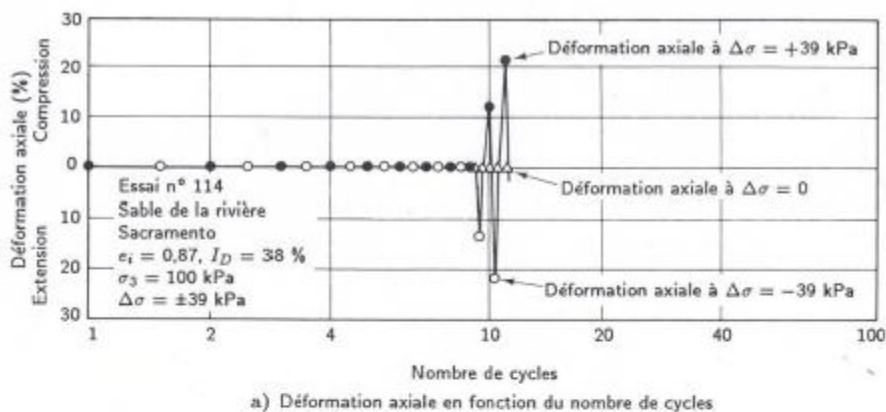


Figure 3.3- Résultats des essais triaxiaux sur un sable à l'état lâche, (Holtz et Kovacs, 1991)

Pour mieux comprendre la réponse du sol face aux chargements cycliques plusieurs essais ont été menés en laboratoire. Tels que les travaux de Seed et de ses étudiants, qui ont effectué des essais triaxiaux cycliques non drainés, consolidés sous contraintes isotropes et anisotropes (Seed et Lee, 1966 ; Seed et Idriss, 1982). La figure (3.3) montre le comportement d'un sable lâche (indice de densité $I_D = 38\%$) lors de l'application de charges triaxiales cycliques dans des conditions non drainées. Le graphique 3.3a montre que la déformation axiale n'apparaît qu'à partir du 9^{ème} cycle,

malgré l'augmentation graduelle des pressions interstitielles lors des cycles précédents (graphique 2.3b). Les déformations débutent (au 9^{ème} cycle) lorsque les surpressions interstitielles (Δu) ont augmenté jusqu'à atteindre environ 100 % de la contrainte de confinement effective initiale (σ'_{3c}). Après 9-10 cycles, le rapport de pression interstitielle ($ru = \Delta u / \sigma'_{3c}$) est donc près de 1 (Kramer et Seed, 1988). C'est à ce moment que la liquéfaction débute (Seed et Lee, 1966). Par contre, des expériences ont montré que le sol peut devenir instable à des valeurs de ru bien inférieures à l'unité.

3.4. Sols pulvérulents denses

Les sols pulvérulents denses, dans les conditions drainées, commencent d'abord à se dilater et les contraintes effectives peuvent rester constantes jusqu'à atteindre un état stationnaire. Lorsque l'environnement est non drainé, le sol ne peut pas se dilater (volume constant). Il subit plutôt une diminution des pressions d'eau et par conséquent, une augmentation des contraintes effectives. L'application de charges cycliques sur les sols relativement denses peut produire dans certains cas un comportement inverse, proche de celui des sols lâches, c'est-à-dire qu'ils tendent à se contracter (mais à volume constant).

La figure (3.4) montre les résultats d'un essai triaxial à chargement cyclique sur un sol relativement dense ($ID = 78\%$) dans des conditions non-drainées. Lors des dix premiers cycles, les pressions interstitielles se développent, sans engendrer de déformation axiale importante dans l'échantillon. Le même comportement a été observé pour les sols lâches (figure 3.3). La différence entre les deux types de sols est remarquable quand les surpressions (Δu) sont maximales et que les contraintes effectives (σ') sont momentanément nulles. Dans le cas du sol lâche, l'échantillon montre des déformations axiales importantes en peu de temps (douzième cycle). La figure (3.4) montre que le sol dense résiste mieux aux contraintes durant quelques cycles (cycles 12 à 20), bien que des contraintes effectives proches de zéro.

Les mécanismes engendrant une augmentation des pressions interstitielles dans les sols denses (dilatants) ne sont pas encore bien compris. Mitchell et Soga (2005) ont émis l'hypothèse que l'augmentation des pressions interstitielles est due à une déstructuration du sol causée par les contraintes cycliques appliquées. Une fois déstructuré, le sol dense réagirait comme un sol contractant et les pressions interstitielles pourraient se développer.

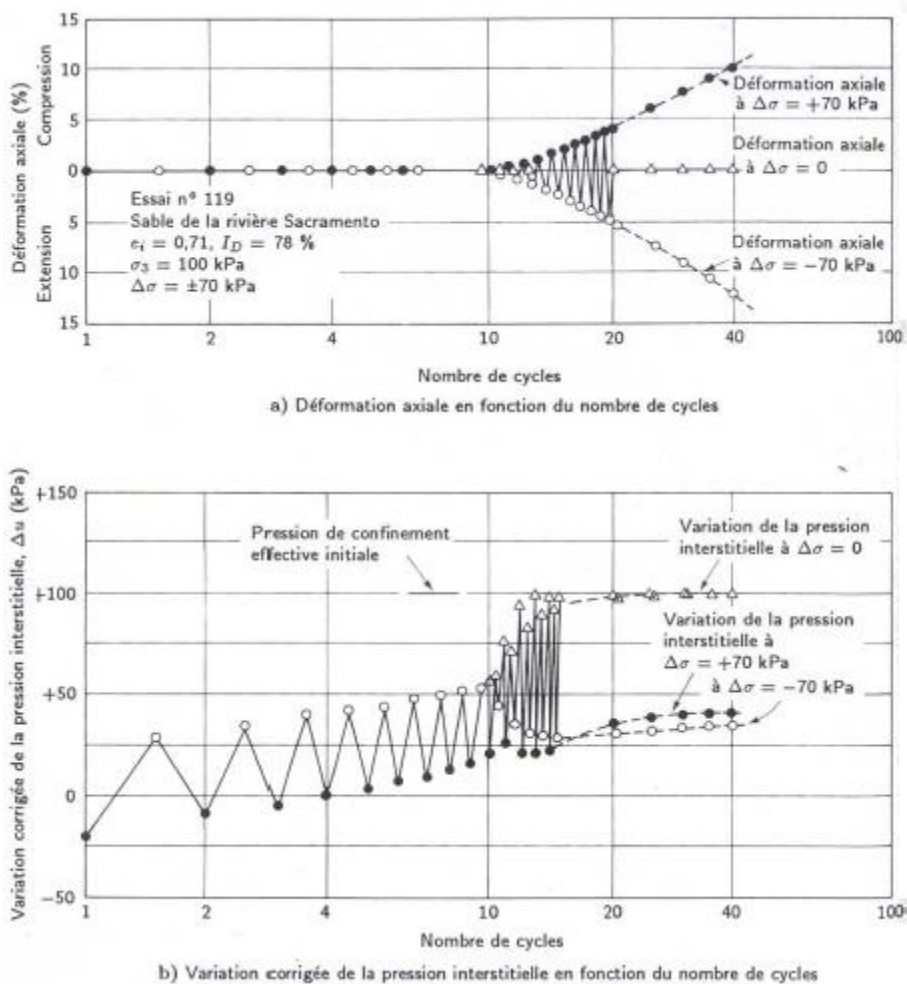


Figure 3.4. Résultats des essais triaxiaux sur un sable à l'état dense (Holtz et Kovacs, 1991)

3.5. Caractéristiques influençant la résistance à la liquéfaction

Dans les parties ci-dessus, on a vu que l'indice de densité I_D (lâche ou dense) et la nature du chargement (statique ou dynamique) influencent la réponse du sol et son risque de liquéfaction. Ces facteurs sont en effet très importants. Mais, il y a d'autres facteurs qui causent la liquéfaction. Les facteurs influençant la liquéfaction seront présentés dans la section qui suit.

3.5.1. Nature du sol

La composition du sol influence les changements de volume lors d'une variation des contraintes. La forme, la grosseur et la distribution de la taille des grains sont aussi des caractéristiques qui peuvent influencer le potentiel de liquéfaction. Par exemple, des particules sphériques résistent moins bien au cisaillement que des particules angulaires. Une granulométrie peu étalée augmente généralement l'indice des vides et ce qui conduit à l'augmentation du potentiel de liquéfaction.

Wang (1979) propose quatre critères permettant d'identifier les sols susceptibles à la liquéfaction. Ces critères, basés sur la nature du sol, ont été nommés « Chinese criteria ».

Contenu en particules argileuses ($\leq 5 \mu\text{m}$)	$\leq 15 \%$
Limite de liquidité (w_L)	$\leq 35 \%$
Teneur en eau naturelle (w_N)	$\geq 0.9 L_L$ (implique condition saturée)
Indice de liquidité (IL)	≤ 0.75

Ces critères s'appliqueraient aux sables même s'ils n'ont pas de limite de liquidité (L_L) ni d'indice de liquidité.

Hunt (1986) propose un fuseau de granulométrie pour les sols susceptibles à la liquéfaction. Ce fuseau est présenté à la figure (3.5).

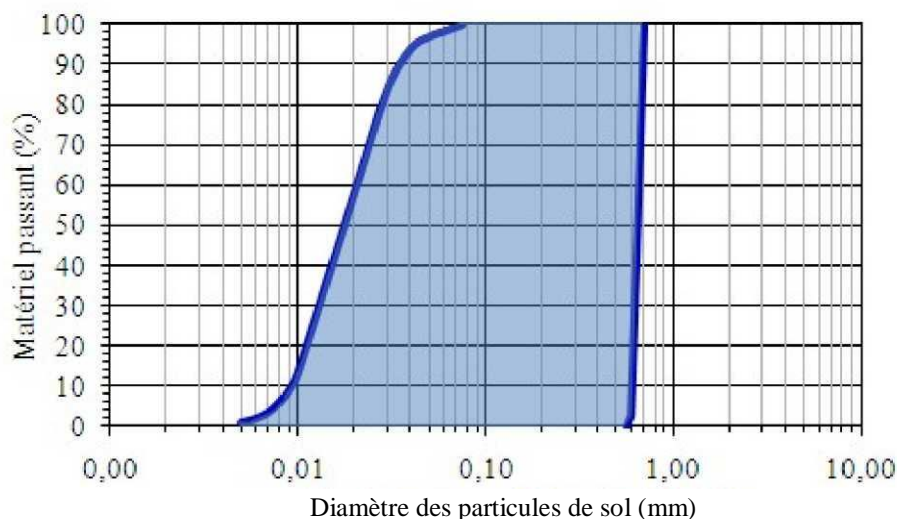


Figure 3.5- Éventail des granulométries pour les sols susceptibles à la liquéfaction (Hunt, 1986).

La présence de silt dans les sols sableux a un impact sur le potentiel de liquéfaction. Des études menées par Koester (1994) ont indiqué que la résistance au cisaillement des sables diminue avec l'augmentation de la fraction de particules fines sans cohésion, jusqu'à un pourcentage de 20 à 30 % de la masse totale. Ce pourcentage serait critique car c'est à partir de cette valeur que la susceptibilité à la liquéfaction serait grande. Polito (1999) a constaté que pour différents types de sable, le pourcentage de particules de silt « critique » variait généralement entre 25 et 50 %. Sous ce pourcentage « critique », le comportement du matériau serait dominé par une interaction sable-sable et au-delà de la valeur critique, par une interaction silt-silt.

Le mode de déposition des sédiments et l'âge de ceux-ci sont également deux facteurs influents la manière dont les couches se superposent. Ceci a un impact sur la résistance offerte par le sol. Quant

à l'âge du dépôt, un sol plus vieux aura tendance à offrir une résistance supérieure à un sol jeune, étant donné qu'il a bénéficié de plus de temps pour se tasser et se densifier.

3.5.2. État du sol et des contraintes

En général, plus le sol est compacté, moins il est susceptible à la liquéfaction. La valeur du coefficient de pression des terres au repos (K_0) et le rapport de surconsolidation (OCR) ont aussi un impact sur la susceptibilité des sols à la liquéfaction.

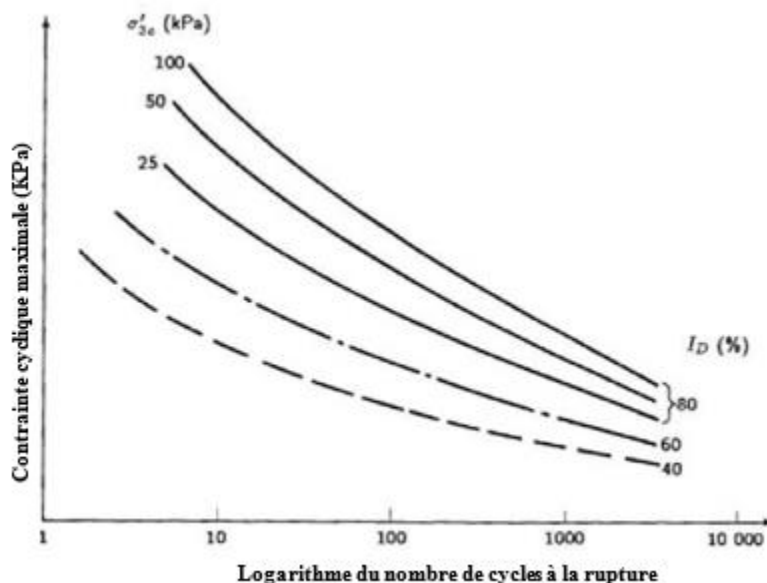


Figure 3.6- Influence de la contrainte horizontale effective sur le nombre de cycles nécessaire afin d'atteindre la rupture d'un sol (Holtz et Kovacs, 1991).

Si la valeur de ces deux paramètres augmente, un plus grand nombre de cycles de cisaillement sera nécessaire pour atteindre la rupture du sol (Holtz et Kovacs, 1991).

La figure (3.6) montre l'influence de l'indice de densité (I_D) et de la contrainte de confinement (σ'_3) sur la rupture d'un sable saturé soumis à un chargement cyclique non drainé. La figure montre que plus l'indice de densité (I_D) grand, la résistance du sable au chargement cyclique est grande. Sur cette figure, il est aussi possible d'observer que le nombre de cycles nécessaire afin d'atteindre la rupture serait influencé par la contrainte de confinement (σ'_3).

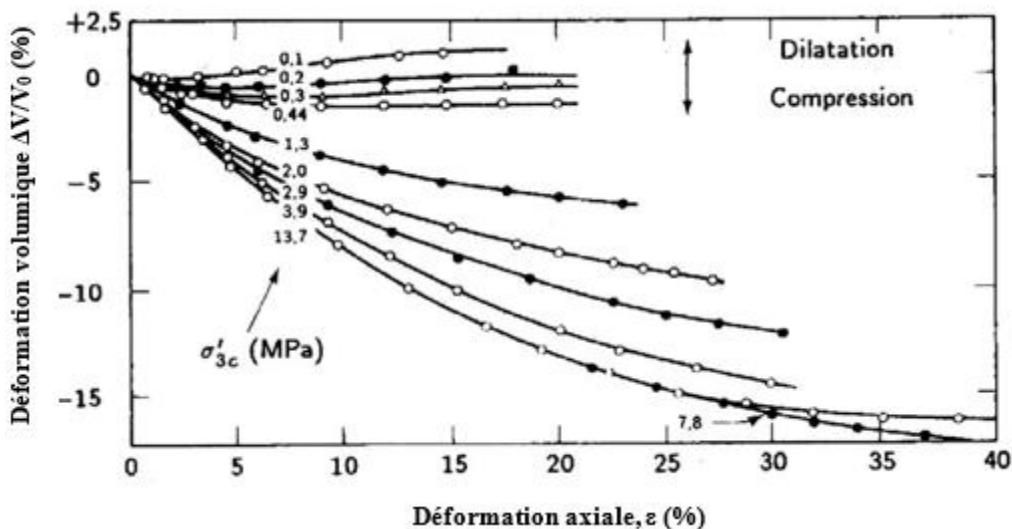


Figure 3.7- Déformation volumétrique en fonction de la déformation axiale pour des essais triaxiaux drainés faits sur un sable lâche (I_D initial = 38 %) de la rivière Sacramento (selon Lee, 1965 ; tirée de Holtz et Kovacs, 1991).

Lee (1965) à partir d'une étude expérimentale sur l'influence de la contrainte de confinement (σ'_3) sur le comportement drainé d'un sable lâche ($I_D = 38\%$). Il a constaté que le sable lâche soumis à des contraintes de confinement faibles, il pouvait se comporter comme un sable dense, c'est-à-dire qu'il se dilatait. La figure (3.7) montre la transition contraction-dilatation pour le sable lâche sous différentes contraintes de confinement. Le même sable dans un état plus dense ($I_D \approx 100\%$) peut se comporter comme un sable lâche (avec contraction) lorsque les contraintes de confinement sont très élevées.

3.5.3. Nature de la sollicitation sismique

Le facteur principal qui conduit un sol sableux saturé à la rupture due à la liquéfaction, c'est le séisme. Par conséquent si les caractéristiques de ce dernier autrement dit la magnitude, l'accélération sismique, la durée, le nombre de cycles et la fréquence d'oscillation du tremblement de terre varient, la réponse sera différente. Ce qui montre l'influence du type de séisme sur le phénomène de liquéfaction.

3.5.4. Niveau de la nappe phréatique

La profondeur de la nappe phréatique est un élément primordial à prendre en compte lors de l'analyse du potentiel de liquéfaction. Un sol ne doit pas nécessairement être complètement saturé pour qu'il y ait risque de liquéfaction, mais il doit contenir suffisamment d'eau pour qu'il y ait compression de celle-ci lors de la densification du sol soumis aux chargements. Yegian et al. (2007) ont étudié l'effet du degré de saturation sur le potentiel de liquéfaction d'un sable à l'aide d'essais

sur une table sismique. Ils ont constaté qu'en diminuant le degré de saturation, il était possible de réduire significativement le rapport des pressions interstitielles (ru) atteint lors de sollicitation sismique. Comme résultats de leurs essais, ils ont démontré qu'en diminuant le degré de saturation de 100 % à 86 %, le rapport ru maximal diminuait de 1 à 0,63 et 0,72 pour des profondeurs de 11 cm et de 21 cm, respectivement (pour une hauteur totale de l'échantillon de 34 cm).

3.6. Méthodes d'évaluation du risque de liquéfaction

Une fois qu'un sol est susceptible de se liquéfier sur la base de divers critères de sensibilité tels que mentionnés dans Kramer et Seed (1988), l'étape suivante du processus est l'évaluation du potentiel de liquéfaction. Les facteurs majeurs qui contrôlent le potentiel de liquéfaction d'un sol pulvérulent saturé sont l'intensité et la durée du tremblement de terre, la densité relative et la contrainte de confinement effective du sol. Plusieurs approches sont utilisées pour évaluer le potentiel de liquéfaction, y compris (i) approche par contraintes cycliques (par exemple, Seed et Idriss, 1971) (ii) approche par déformations cycliques (par exemple, Dobry et al., 1982) et (iii) approche par intensité d'Arias (par exemple, Kayen et Mitchell, 1997). Seulement la 1^{ère} approche sera discutée dans les sections suivantes.

3.6.1. Approche par contraintes cycliques (Seed et Idriss, 1971)

Cette approche est la procédure la plus largement utilisée dans la pratique pour estimer la résistance à la liquéfaction des sols sableux. Dans cette approche, l'évaluation du potentiel de liquéfaction se fait par la comparaison de deux paramètres : le rapport de contrainte cyclique (CSR) et le rapport de résistance du sol à la liquéfaction (CRR). CSR peut être évalué par la procédure simplifiée qui a été développée par Seed et Idriss (1971). Par contre la résistance à la liquéfaction du sol CRR peut être évaluée par plusieurs méthodes simplifiées basées sur les essais in-situ. Parmi ces méthodes : Méthodes basées sur l'essai de pénétration standard (ex., Seed and Idriss, 1971; Seed et al., 1985; Youd et al., 2001; Idriss et Boulanger, 2006), celles basées sur l'essai de pénétration au cône (ex., Robertson et Campanella, 1985; Robertson et Wride, 1998; Juang et al., 2003; Idriss et Boulanger, 2006), mesure de la vitesse des ondes de cisaillement (ex., Andrus et Stokoe, 2000) et l'essai au dilatomètre de Marchetti (ex., Monaco et Marchetti, 2007; Tsai et al., 2009). Seulement quelques méthodes parmi susmentionnées seront discutées dans les sections suivantes.

3.6.1.1. Rapport de contrainte cyclique (CSR)

CSR représente l'effort de cisaillement cyclique moyen dans une couche (τ_{cyc}) normalisé par rapport à la contrainte effective due au sol sus jacent (σ'_{v0}). Il est fonction de l'accélération en surface maximale (a_{max}), de la profondeur au point considéré (z) et des contraintes verticales totales (σ_{v0}) et effectives (σ'_{v0}). Pour un tremblement de terre d'amplitude $M_w = 7,5$, CSR est généralement présenté par l'équation de Seed et Idriss (1971).

$$CSR = \frac{\tau_{cyc}}{\sigma'_{v0}} = 0.65 \times \left(\frac{a_{max}}{g} \right) \times \left(\frac{\sigma_{v0}}{\sigma'_{v0}} \right) \times r_d \quad (3.1)$$

Où

- τ_{cyc} : Contrainte de cisaillement cyclique.
- a_{max} : Amplitude maximale de l'accélération horizontale au niveau de la surface du sol.
- σ_{v0} et σ'_{v0} : Contraintes verticales totale et effective des sols sus-jacents.
- r_d : Coefficient réducteur des contraintes avec la profondeur.
- g : Constante gravitationnelle (m/s^2).

Le facteur de 0,65 est un facteur empirique, basé sur des observations, et visant à représenter la valeur moyenne de la contrainte de cisaillement générée lors d'un tremblement de terre. Il a été estimé que cette valeur valait 65 % de la contrainte maximale (qui correspond au maximum de l'accélération horizontale, PGA).

Le rapport de contrainte cyclique CSR est calculé pour une magnitude $M=7.5$. Par conséquent, pour un séisme de $M \neq 7.5$, un coefficient d'ajustement doit être introduit dans l'équation (3.1). Ce dernier est le facteur d'échelle de magnitude MSF.

3.6.1.1.1. Coefficient de réduction de contrainte

r_d est un coefficient de réduction de contrainte qui tient compte de la flexibilité du profil du sol. Plusieurs formules ont été établies, mais c'est Seed et Idriss en 1971 qui ont introduit au premier ce coefficient comme une fonction qui dépend uniquement de la profondeur (z). En 1999 Idriss a effectué plusieurs analyses suite à des réponses de différents sites et il a conclu que le paramètre r_d pourrait être exprimé de manière adéquate comme une fonction qui dépend de la profondeur (z) et de la magnitude du séisme (M) simultanément et il a modifié l'expression de r_d comme suit :

$$\ln(r_d) = \alpha(z) + \beta(z) \times M_w \quad (3.2a)$$

$$\alpha(z) = -1.012 - 1.126 \sin\left(\frac{z}{11.73} + 5.133\right) \quad (3.2b)$$

$$\beta(z) = 1.06 + 0.118 \cdot \sin\left(\frac{z}{11.28} + 5.142\right) \quad (3.2c)$$

Où z est la profondeur en m et M_w est la magnitude. Ces équations sont applicables pour $z \leq 34$ m. Pour $z > 34$ m, l'équation suivante est plus appropriée :

$$r_d = 0.12 \exp(0.22M) \quad (3.2d)$$

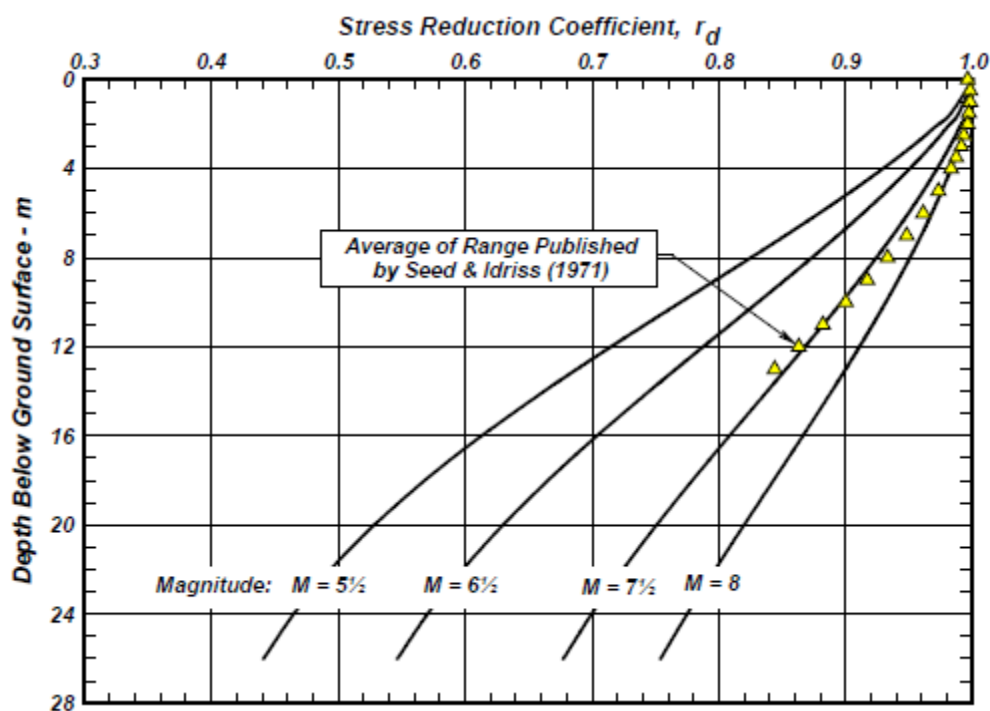


Figure 3. 8- Variation du coefficient réducteur de contrainte en fonction de la magnitude M (Idriss et Boulanger, 2006)

Les courbes de r_d calculées en utilisant l'équation (3.2) pour $M = 5.5, 6.5, 7.5$ et 8 sont présentées à la figure 3.8.

3.6.1.1.2. Facteur d'échelle de magnitude MSF

Le facteur d'échelle de magnitude, MSF, a été utilisé pour ajuster CSR induite due à un tremblement de terre de magnitude M à une valeur de CSR équivalente pour une magnitude sismique, $M = 7.5$. MSF est définie comme suit :

$$MSF = CSR_M / CSR_{7.5} \quad (3.3)$$

Ainsi, MSF fournit une représentation approximative des effets de la durée de tremblement de terre ou du nombre équivalent de contraintes cycles. Plusieurs formules ont été établies comme montré dans la figure ((3.9). Mais la plus utilisée est celle d'Idriss (1999) donnée par l'équation (3.4).

$$MSF = 6.9 \exp\left(\frac{-M}{4}\right) - 0.058 \leq 1.8 \quad (3.4)$$

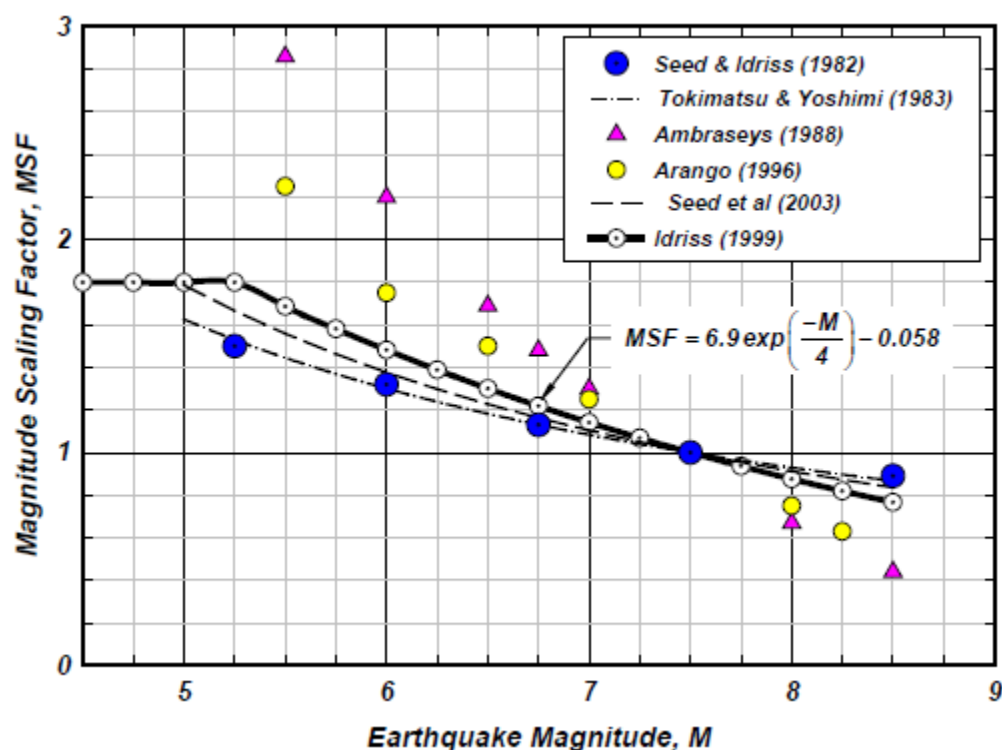


Figure 3.9- Différentes courbes de facteur d'échelle de magnitude MSF (Idriss et Boulanger, 2006).

3.6.1.2. Rapport de résistance cyclique (CRR)

Le but principal de l'atelier de travail qui a été organisé par le centre de recherche national (NCR) en 1985 est de revoir les procédures d'évaluation de la résistance à la liquéfaction. Une méthode plausible pour évaluer CRR doit être tirée d'une analyse profonde sur des échantillons intacts par des essais au laboratoire. Malheureusement, les états de contraintes in situ généralement ne peuvent pas être rétablis en laboratoire, et des échantillons de sols granulaires récupérés avec des techniques typiques de forage et d'échantillonnage sont trop troublés pour donner des résultats significatifs.

Seulement grâce à des techniques d'échantillonnage spécialisées, telles que le sol gelé qu'on peut obtenir des échantillons suffisamment non perturbés. Le coût de ces procédures est généralement prohibitif. Pour éviter les difficultés associées à la qualité des échantillons et des essais de laboratoire, les essais in-situ sont devenus plus pratiques pour l'évaluation de liquéfaction.

Plusieurs essais in-situ ont été utilisés pour l'évaluation de résistance à la liquéfaction, y compris l'essai de la pénétration standard (SPT), essai de pénétration du cône (CPT), vitesse de l'onde de cisaillement (V_s) et l'essai dilatomètre de Marchetti (DMT).

Les chercheurs dans leurs méthodes simplifiées ont essayé de représenter CRR en fonction d'un paramètre de résistance normalisée d'un essai in situ (par exemple, $(N_1)_{60}$, V_{s1} , q_{c1} ...). Le rapport de résistance cyclique (CRR) obtenu par ces dernières méthodes simplifiées représente la valeur de CRR pour des conditions bien particulières telles que le profil du sol est horizontal et la contrainte de confinement verticale prise égale à 100 kpa. Dans des cas différents, l'expression de CRR doit être modifiée (équation 6) pour tenir compte de ces effets.

$$CRR = CRR_{7.5} \times K_\sigma \times K_\alpha \quad (3.5)$$

Où

$CRR_{7.5}$: Rapport de résistance cyclique pour une magnitude $M=7.5$.

K_σ : Coefficient de correction de la contrainte de cisaillement statique initiale

K_α : Coefficient de correction de la contrainte de confinement verticale effective.

Avant d'exposer les différentes méthodes d'évaluation de CRR, nous allons d'abord aborder les différentes procédures proposées par les chercheurs pour le calcul de K_σ et K_α .

3.6.1.2.1. Facteur de correction de contrainte verticale de confinement K_σ

Le but de (NCR) est de revoir les différentes procédures d'évaluation de CRR, ce qui conduit aux formules de tous les paramètres qui influent sur CSR et CRR. Parmi, les résultats du conseil est la modification de la formule de K_σ .

Youd et al(2001) ont proposé la formule de K_σ comme suit :

$$K_\sigma = \left(\frac{\sigma'_v}{P_a} \right)^{f-1} \quad (3.6)$$

Où

K_σ : Facteur de correction de contrainte verticale effective.

P_a : Pression atmosphérique égale à 100 kpa.

f : Exposant dépendant de la densité relative du sol (D_r)

Pour $40\% \leq D_r \leq 60\%$, $0,7 \leq f \leq 0,8$

Pour $60\% \leq D_r \leq 80\%$, $0,6 \leq f \leq 0,7$

(Recommandations adoptées par le comité NCEER).

Après trois ans, c'est-à-dire en (2004), Boulanger et Idriss ont modifié la valeur de K_σ en fonction de D_r et les résistances de pénétration normalisées (N_1)₆₀, (q_{C1N}) des essais SPT et CPT.

$$K_\sigma = 1 - C_\sigma \ln\left(\frac{\sigma'_V}{P_a}\right) \leq 1 \quad (3.7)$$

$$C_\sigma = \frac{1}{18.9 - 17.3D_r} < 0.3 \quad (3.8.a)$$

$$C_\sigma = \frac{1}{18.9 - 2.55\sqrt{(N_1)_{60}}} < 0.3 \quad (3.8.b)$$

$$C_\sigma = \frac{1}{37.3 - 8.27(q_{C1N})^{0.264}} < 0.3 \quad (3.8.c)$$

3.6.1.2.2. Facteur de correction de contrainte de cisaillement statique initiale K_α

C'est un facteur qui tient compte de la forme du terrain. Toutes les formules proposées pour l'évaluation de CRR ont été établies pour des terrains plats.

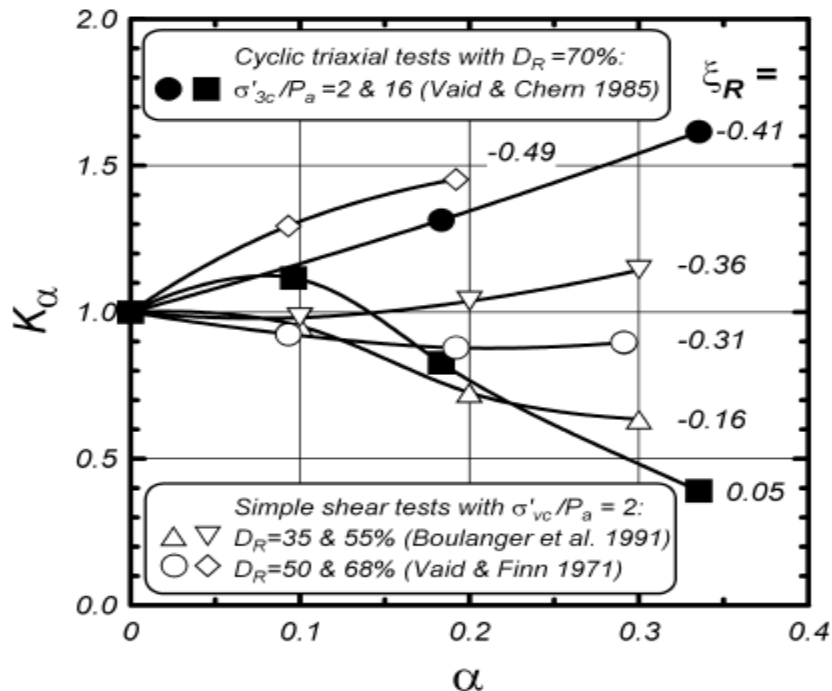


Figure 3.10- Variation du facteur de contrainte statique de cisaillement K_α (Idriss et Boulanger, 2003).

Idriss et Boulanger (2003a) ont établi des expressions en utilisant les données de la figure (3.10) qui sont basées principalement sur un essai de cisaillement direct.

$$K_\alpha = a + b \cdot \exp\left(\frac{-\xi_R}{c}\right) \quad (3.9)$$

$$a = 1267 + 636\alpha^2 - 634\exp(\alpha) - 632\exp(-\alpha) \quad (3.10)$$

$$b = \exp(-1.11 + 12.3\alpha^2 + 1.31 \cdot \ln(\alpha + 0.0001)) \quad (3.11)$$

$$c = 0.138 + 0.126\alpha + 2.52\alpha^3 \quad (3.12)$$

$$\alpha = \frac{\tau_s}{\sigma'_{vc}} \quad (3.13)$$

L'indice ξ_R est calculé à partir des résistances de pénétration comme suit :

$$\xi_R = \frac{1}{Q - \ln\left(\frac{100(1 + 2K_0)\sigma'_{vc}}{3P_a}\right)} - \sqrt{\frac{(N_1)_{60}}{46}} \quad (3.14)$$

$$\xi_R = \frac{1}{Q - \ln\left(\frac{100(1 + 2K_0)\sigma'_{vc}}{3P_a}\right)} - (0.478(q_{c1N})^{0.264} - 1.063) \quad (3.15)$$

Avec $q_{c1N} \geq 21$. En outre, α et ξ_R sont limités comme indiqués dans les expressions ci-dessous :

$$\alpha \leq 0.35$$

$$-0.6 \leq \xi_R \leq 0.1$$

3.6.2. Évaluation de CRR par les méthodes simplifiées basées sur l'essai SPT

L'essai SPT a été le plus utilisé au fil des années, ce qui a permis de construire une base de données relativement riche. Ce test repose sur le nombre de coups nécessaire pour enfoncer un carottier dans le sol d'une profondeur de trente centimètres (N_1-60cs). L'indice « 1 » définit que le nombre de coups est normalisé pour une contrainte de confinement verticale de 100 kPa. L'indice « 60 » implique que la valeur a été normalisée pour une efficacité de 60 % de l'énergie totale fournie par le marteau lors de l'enfoncement de la tige. L'indice « cs », quant à lui, implique que la valeur est normalisée pour un sable propre (clean sand), c'est-à-dire sans particule fine. Certaines relations ont été proposées afin d'effectuer ces normalisations (ex. Youd et al. 2001). Avant d'exposer les différentes méthodes basées sur l'essai SPT, nous devons d'abord expliquer quelques paramètres communs tels que : $(N_1)_{60}$.

Après avoir introduit les corrections mentionnées ci-dessus $(N_1)_{60}$ est donné comme suit :

$$(N_1)_{60} = N_m \times C_N \times C_E \times C_B \times C_R \times C_S \quad (3.16)$$

Où

N_m : résistance à la pénétration par mesure standard ;

C_N : correction de la contrainte de sols sus-jacents ;

C_E : correction du taux d'énergie de la masse ;

C_B : correction du diamètre du forage ;

C_R : correction de la longueur du train de tiges ;

C_S : correction du carottier avec ou sans revêtement.

Le tableau (3.1) résume le calcul de ces facteurs de corrections

Tableau 3.1- Facteurs de corrections de la résistance de pénétration de SPT

Facteur	Équipement variable	Notation	Correction
Contrainte de sols sus-jacents	σ'_{v0}	C_N	$(P_a/\sigma'_{v0})^{0.5}$ où $C_N \leq 2$ et $P_a = 100$ kPa
Rapport d'énergie	<ul style="list-style-type: none"> · Mouton en Donut · Mouton de sécurité · Mouton automatique 	$C_E = ER / 60\%$	0,5 à 1,0 0,7 à 1,2 0,8 à 1,3
Diamètre de forage	<ul style="list-style-type: none"> · 65 mm à 115 mm · 150 mm · 200 mm 	C_B	1,00 1,05 1,15
Longueur du train de tiges	3 m à 4 m 4 m à 6 m 6 m à 10 m 10 m à 30 m > 30 m	C_R	0,75 0,85 0,95 1,00 < 1,00
Méthode du carottage	<ul style="list-style-type: none"> - Carottage standard - Carottage sans revêtement 	C_S	1,00 1,1 à 1,3

3.6.2.1. Méthode de Youd et al (2001)

Youd et al (2001) ont proposé une mise à jour de la courbe de CRR (figure 3.11) proposée par Seed et al (1985). La formule déduite de cette courbe est donnée par l'équation ci-dessous :

$$CRR_{7.5} = \frac{1}{34 - (N_1)_{60}} + \frac{(N_1)_{60}}{135} + \frac{50}{[10 \cdot (N_1)_{60} + 45]^2} - \frac{1}{200} \quad (3.17)$$

Cette équation est valable pour $(N_1)_{60} \leq 30$, pour $(N_1)_{60} > 30$, le sol est considéré non liquéfiable. Pour tenir compte de l'effet des particules fines, l'équation ci-dessus a été améliorée comme suit :

$$CRR_{7.5} = \frac{1}{34 - (N_1)_{60cs}} + \frac{(N_1)_{60cs}}{135} + \frac{50}{[10 \cdot (N_1)_{60cs} + 45]^2} - \frac{1}{200} \quad (3.18)$$

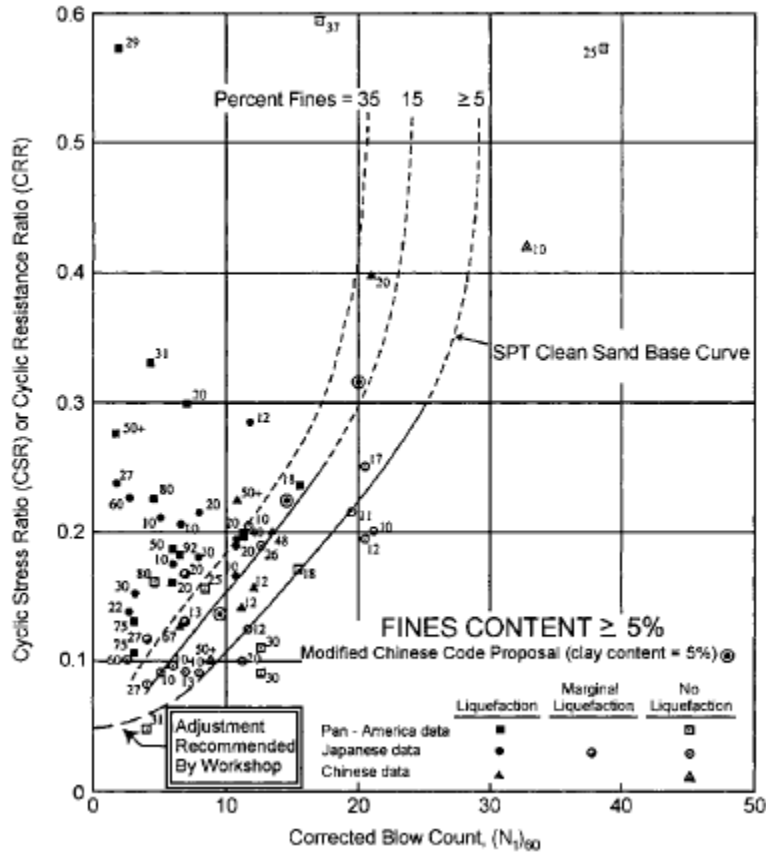


Figure 3.11- Variation de CRR en fonction de $(N_1)_{60}$ (courbes modifiée de Seed et al. 1985, tirée de Youd et al. 2001)

Où

$$(N_1)_{60cs} = \alpha + \beta(N_1)_{60} \quad (3.19)$$

Avec α et β des coefficients déterminés par les relations suivantes :

$$\alpha = 0, \text{ pour } FC \leq 5\% \quad (3.20.a)$$

$$\alpha = \exp \left[1.76 - \left(\frac{190}{FC^2} \right) \right], \text{ pour } 5\% < FC < 35\% \quad (3.20.b)$$

$$\alpha = 5, \text{ pour } FC \geq 35\% \quad (3.20.c)$$

$$\beta = 1, \text{ pour } FC \leq 5\% \quad (3.21.a)$$

$$\beta = \left[0.99 + \left(\frac{FC^{1.5}}{1000} \right) \right], \text{ pour } 5\% < FC < 35\% \quad (3.21.b)$$

$$\beta = 1.2, \text{ pour } FC \geq 35\% \quad (3.21.c)$$

3.6.2.2. Méthode d'Idriss et Boulanger (2006)

En se basant sur les modifications apportées aux paramètres r_d (Seed et al (2001)), MSF (Idriss, 1999), K_σ (Boulanger and Idriss (2004)) et les courbes de limites publiées dans (Youd et al., 2001) comme indiqué dans la figure (3.12), Idriss et Boulanger (2006) ont proposé une nouvelle formule de CRR pour les sols pulvérulents donnée par l'équation (3.22).

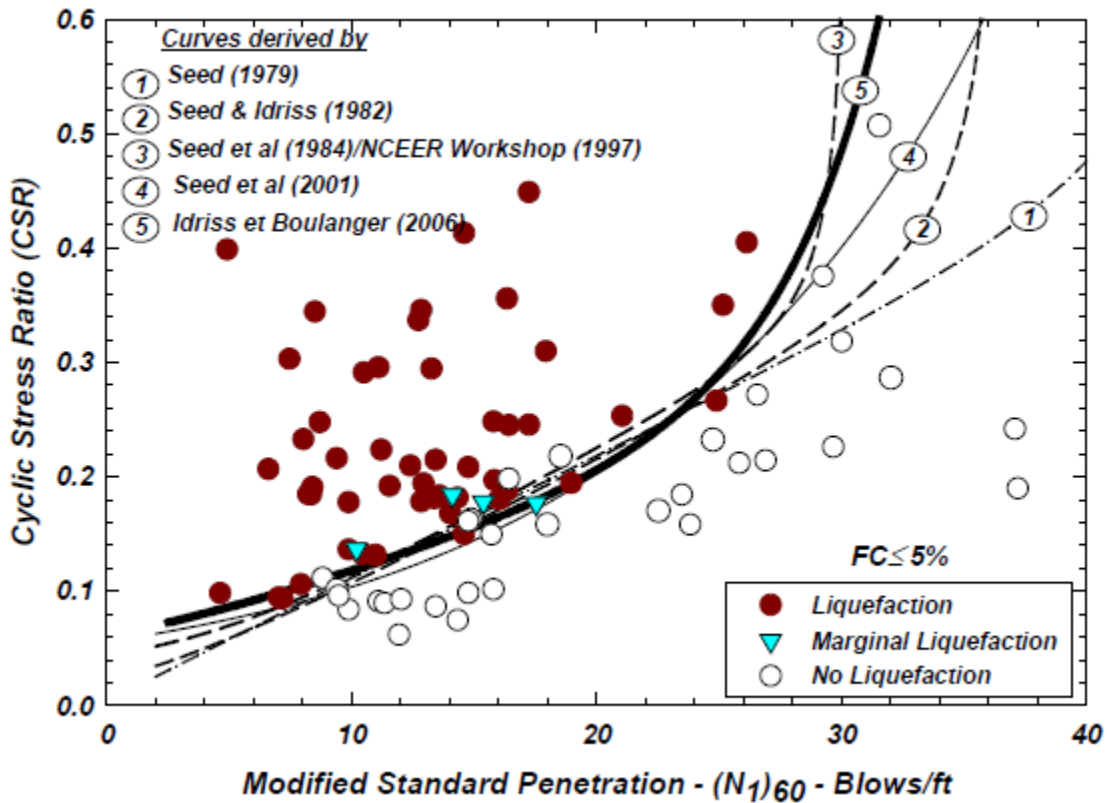


Figure 3.12- Variation de CRR en fonction de $(N_1)_{60}$ (Idriss et Boulanger, 2006)

$$CRR_{7.5} = \exp \left[\left(\frac{(N_1)_{60cs}}{14.1} \right) + \left(\frac{(N_1)_{60cs}}{126} \right)^2 - \left(\frac{(N_1)_{60cs}}{23.5} \right)^3 + \left(\frac{(N_1)_{60cs}}{25.4} \right)^4 - 2.8 \right] \quad (3.22)$$

Avec

$$(N_1)_{60cs} = (N_1)_{60} + \exp \left[1.63 + \frac{9.7}{FC + 0.1} - \left(\frac{15.7}{FC + 0.1} \right)^2 \right] \quad (3.23)$$

3.6.3. Méthodes d'évaluation de CRR basées sur l'essai CPT

Vu qu'il a une bonne précision et d'une excellente reproductibilité, l'essai de pénétration au cône est devenu le plus largement utilisé par les géotechniciens. CPT donne des mesures en continu de la résistance du sol, contrairement à l'essai SPT, ce qui le rend préféré lors de l'évaluation d'un profil très variable selon la profondeur. Plusieurs études ont été menées afin d'utiliser la résistance de pénétration de CPT pour l'évaluation du rapport de la résistance de liquéfaction (CRR). Dans ce qui suit, nous allons exposer quelques méthodes simplifiées basées sur les paramètres de résistance de CPT.

3.6.3.1. Méthode d'Olsen et al. (1997)

Dans la méthode d'évaluation de CRR à partir des mesures de CPT proposée par (Olsen et al. 1997), R_{CR} est calculé de la façon suivante :

$$CRR = 0,00128 \left[\frac{q_c}{(\sigma'_v)^{0,7}} \right] - 0,025 + 0,17R_f - 0,028R_f^2 + 0,0016R_f^3 \quad (3.23)$$

Où

$$R = \left(\frac{f_s}{q_c} \right) \times 100 \quad (3.24)$$

q_c : Résistance de pointe exprimée en atm.,

σ'_v : Contrainte effective exprimée en atm ;

R_f : Rapport de résistance de frottement en pourcentage défini par l'équation (3.24).

3.6.3.2. Méthode de Juang et al (2003)

Dans la méthode de Juang, R_{CR} est exprimé de la façon suivante :

$$CRR = C_\sigma \exp[-2,957 + 1,264((q_{c1N})_{cs}/100)^{1,25}] \quad (3.25)$$

Où

$$C_\sigma = -0,016(\sigma'_v/100)^3 + 0,178(\sigma'_v/100)^2 + 0,063(\sigma'_v/100) + 0,903 \quad (3.26)$$

$$(q_{c1N})_{cs} = q_{c1N} \times (2,429I_c^4 - 16,943I_c^3 + 44,551I_c^2 - 51,497I_c + 22,802) \quad (3.27)$$

$$q_{c1N} = \frac{q_c}{(\sigma'_v)^{0,5}} \quad (3.28)$$

$$I_c = [(3.47 - \text{Log}_{10} q_{c1N})^2 - (\text{Log}_{10}(F) + 1.22)^2]^{0.5} \quad (3.29)$$

Où q_c σ'_v sont respectivement la résistance de pointe et la contrainte verticale effective données en unité (atm). Et F représente la résistance de frottement normalisée donnée comme suit :

$$F = \frac{f_s}{(q_c - \sigma_v)} \quad (3.30)$$

Où f_s résistance de frottement et σ_v contrainte verticale totale en kpa.

3.6.3.3. Méthode Chinoise

Cette méthode est basée sur une condition : si la résistance de pointe mesurée d'un CPT est inférieure à une valeur critique calculée, le sol ne se liquéfiera pas. Elle est traduite par l'expression :

$$q_c \leq (q_c)_{cr} \quad (3.31)$$

Où

$(q_c)_{cr}$ est une valeur critique de la résistance de pointe, elle est calculée à partir de l'équation

$$q_{c,cr} = q_0 \times C_w \times C_u \times C_p \quad (3.32)$$

q_0 est la résistance de pointe dépendante de l'échelle du séisme (tableau 3.2).

Tableau 3.2- Valeurs critiques de la résistance de pointe q_0 selon la méthode chinoise

Échelle du séisme	M=7,0	M=8,0	M=9,0
q_0	4,6 ~ 5,5	10,5 ~ 11,8	16,4 ~ 18,2

L'équation (3.32) suppose que la nappe phréatique et la contrainte due aux sols sus jacents n'ont un effet qu'à partir de 2 m de profondeur.

$$C_w = 1 - 0,065(d_w - 2) \quad (3.33)$$

$$C_u = 1 - 0,05(d_u - 2) \quad (3.34)$$

Où d_w est le niveau d'eau et d_u est la profondeur du sol considéré.

Pour le calcul de C_p , on se réfère au tableau 3.3.

Tableau 3.3- Détermination de C_p selon le type de sol et son indice de plasticité

Type de sol	Sable	Sable silteux	
Indice de plasticité, I_L	$I_L \leq 3$	$3 < I_L \leq 7$	$7 < I_L \leq 10$
C_p	1,0	0,6	0,8

Pour l'évaluation de l'intensité de la liquéfaction, on adopte l'équation suivante :

$$L_{LE} = \sum_{i=1}^n \left(1 - \frac{q_{ci}}{q_{c,cri}} \right) \times d_i \times W_i \quad (3.35)$$

Ainsi, l'évaluation sera :

- Si $I_{LE} = 0$ Pas de liquéfaction
 Si $0 < I_{LE} < 5$ La liquéfaction est très improbable
 Si $5 < I_{LE} < 15$ La liquéfaction est probable
 Si $I_{LE} > 15$ La liquéfaction est presque certaine

3.6.3.4. Méthode d'Idriss et Boulanger (2006)

Cette réévaluation des procédures basées sur le CPT ont incorporés les ajustements et les révisions de paramètres qui sont semblable à ceux précédemment décrits pour la réévaluation du SPT par les mêmes auteurs. Ainsi, toutes les valeurs de CSR et q_{c1N} ont été recalculés en utilisant les paramètres (r_d , MSF, K_σ et C_N) modifiés.

La nouvelle formule de CRR, en utilisant les considérations susmentionnées, est donnée par l'équation

$$CRR_{7.5} = \exp \left[\frac{q_{c1N}}{540} + \left(\frac{q_{c1N}}{67} \right)^2 - \left(\frac{q_{c1N}}{80} \right)^3 + \left(\frac{q_{c1N}}{114} \right)^4 - 3 \right] \quad (3.36)$$

Pour tenir compte des effets des particules fines, Idriss et Boulanger (2008) ont utilisé une approche similaire à celle utilisée pour l'approche SPT. En conséquence, la résistance de pointe normalisée $(q_{c1N})_{cs}$ peut être calculée comme suit :

$$(q_{c1N})_{cs} = q_{c1N} + \Delta q_{c1N} \quad (3.37)$$

Où

$$\Delta q_{c1N} = \left(5.4 + \frac{q_{c1N}}{16}\right) \cdot \exp\left(1.63 + \frac{9.7}{FC + 0.01} - \left(\frac{15.7}{FC + 0.01}\right)^2\right) \quad (3.38)$$

Ce qui donne un CRR calculé comme suit :

$$CRR_{7.5} = \exp\left[\frac{(q_{c1N})_{cs}}{540} + \left(\frac{(q_{c1N})_{cs}}{67}\right)^2 - \left(\frac{(q_{c1N})_{cs}}{80}\right)^3 + \left(\frac{(q_{c1N})_{cs}}{114}\right)^4 - 3\right] \quad (3.39)$$

3.6.4. Méthodes d'évaluation de CRR basées sur la vitesse des ondes de cisaillement (V_s)

En suivant la procédure traditionnelle pour la correction des résistances standards de pénétration et au cône (Olsen, 1997 ; Robertson et Wride, 1997 ; Youd et al., 1997 ; Robertson et Wride, 1998), on peut corriger V_s par rapport à une contrainte de référence (Eq. 3.40, Sykora 1987 ; Robertson et al. 1992) :

$$V_{s1} = V_s \left(\frac{P_a}{\sigma'_v}\right)^{0,25} \quad (3.40)$$

Où

P_a : contrainte de référence, 100 kPa ou approximativement la pression atmosphérique

L'équation (3.40) suppose que $\sigma'_h = K_0 \times \sigma'_v$ et $k_0 \approx 0,5$ pour les sols déjà liquéfiés. L'équation (3.40) implique que V_s est mesurée dans deux directions, celle du déplacement des particules et celle des contraintes principales, et que l'une de ces directions est verticale.

Comme la direction de la propagation des ondes et celle du déplacement des particules sont différentes par rapport aux contraintes dans le sol pour chaque méthode sismique adoptée in-situ, quelques variations dans les mesures de V_s sont attendues. Ces variations sont minimisées en réalisant des essais avec au moins une composante majeure de la propagation des ondes ou du déplacement des particules dans la direction verticale.

La valeur de CSR séparant l'état de liquéfaction de l'état de non-liquéfaction pour une vitesse corrigée V_{s1} est notée CRR. Les relations existantes entre V_{s1} et CRR sont regroupées sur la figure (3.16).

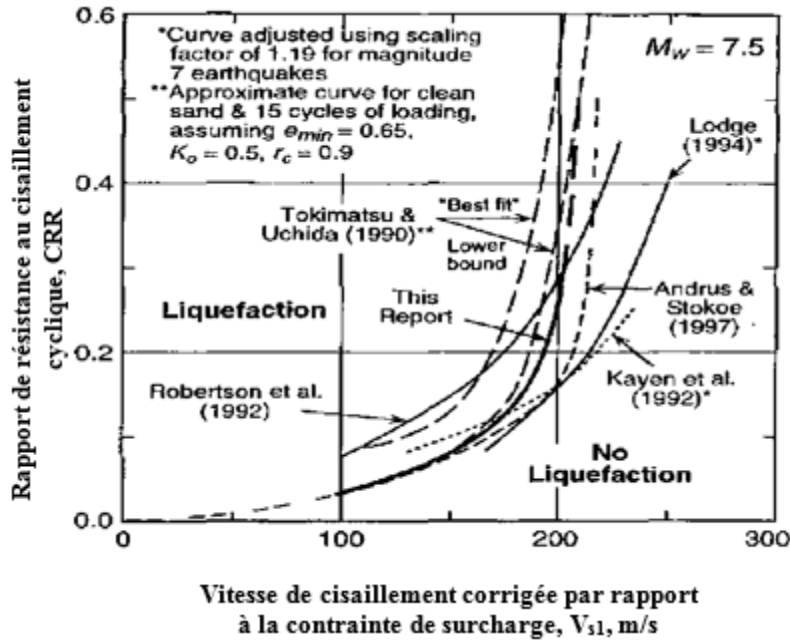


Figure 3.13- Différentes formules de CRR en fonction de la vitesse de cisaillement corrigée, V_{s1} (Andrus et Stokoe, 2000)

3.6.4.1. Méthode d'Andrus et Stokoe (1997)

La relation proposée par Andrus et Stokoe (1997) (figure 3.13) a été présentée dans le workshop NCEER en 1996. Ils ont construit des courbes pour des sols non cimentés, âge holocène avec différents pourcentages de fines. Ces courbes sont basées sur une relation modifiée entre V_{s1} et CSR pour des déformations en cisaillement cyclique proposée par (Dobry, 1981). L'équation modifiée est de la forme :

$$CRR_{7.5} = \left[0.022 \left(\frac{V_{s1}}{100} \right)^2 + 2.8 \left(\frac{1}{V_{s1}^* - V_{s1}} - \frac{1}{V_{s1}^*} \right) \right] \quad (3.41)$$

Où

$$V_{s1}^* = 215 \text{ m/s}, \quad \text{pour sable avec } FC \leq 5\% \quad (3.42.a)$$

$$V_{s1}^* = 215 - 0.5(FC - 5) \text{ m/s}, \quad \text{sable avec } 5\% < FC \leq 35\% \quad (3.42.b)$$

$$V_{s1}^* = 200 \text{ m/s}, \quad \text{pour sable et limon avec } FC \geq 35\% \quad (3.42.c)$$

V_{s1}^* = limite supérieure de V_{s1} pour que la liquéfaction se produise.

3.6.4.2. Méthode d'Andrus et al (2004)

La figure (3.14) présente la courbe de CRR en fonction de la vitesse d'onde de cisaillement corrigé $(V_{S1})_{csa1}$ montré par Andrus et Stokoe (2000). Cette courbe pour un pourcentage de fines (FC) inférieur ou égal à 5 %, correspondant à un sable propre. Andrus et al(2004) ont proposé un ajustement des courbes établies pour des sols comportant plus de 5% de fines.

$$CRR = 0.022 \left[\frac{(V_{S1})_{csa1}}{\sigma'_{vo}} \right]^2 + 2.8 \left[\frac{1}{215 - (V_{S1})_{csa1}} - \frac{1}{215} \right] \quad (3.42)$$

$(V_{S1})_{csa1}$: vitesse d'onde de cisaillement corrigé donné comme suit :

$$(V_{S1})_{csa1} = K_{a1}(V_{S1})_{cs} = K_{a1}K_{cs}V_{S1} \quad (3.43)$$

Où

K_{cs} : Facteur de correction de content des fines,

K_{a1} : Facteur d'âge pour corriger les grandes valeurs de V_{S1} provoquées par le vieillissement.

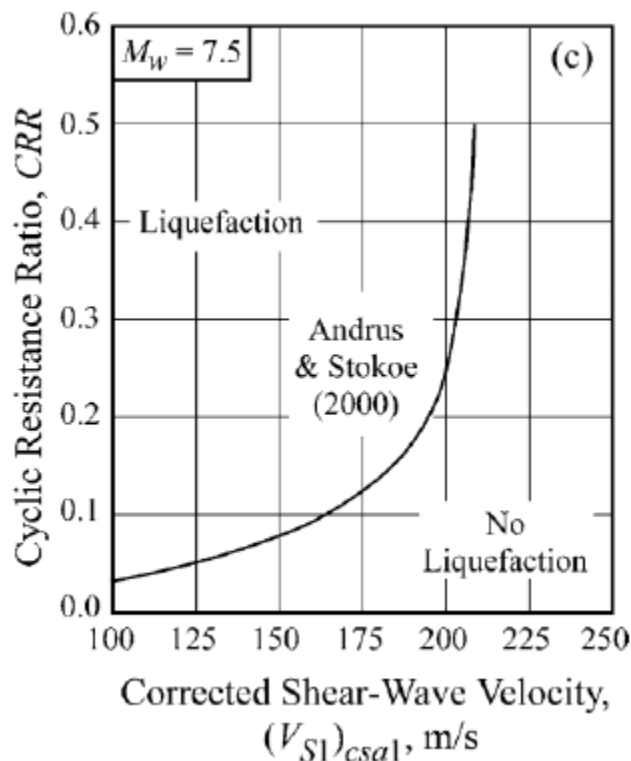


Figure 3. 14- Courbe simplifiée pour le calcul du CRR basée sur VS pour $M_w=7.5$ (Andrus et Stokoe, 2000)

Andrus et Stokoe (2004) ont assumé $K_{a1} = 1$ pour tous les sols d'Holocène-âge.

$$K_{cs} = 1, \text{ pour } FC \leq 5\% \quad (3.44.a)$$

$$K_{cs} = 1 + (FC - 5)T, \text{ pour } 5\% < FC \leq 35\% \quad (3.44.b)$$

$$K_{cs} = 1 + 30T, \text{ pour } FC > 35\% \quad (3.44.c)$$

Avec

$$T = 0.009 - 0.0109 \left(\frac{V_{s1}}{100} \right) + 0.0038 \left(\frac{V_{s1}}{100} \right)^2 \quad (3.45)$$

3.6.5. Évaluation du CRR par l'essai de dilatomètre de Marchetti (DMT)

Bien que les méthodes simplifiées basées sur SPT, CPT et Vs soient bien développées et largement utilisées, l'utilisation de l'essai du dilatomètre (DMT) pour l'évaluation de la résistance à la liquéfaction a reçu une plus grande attention au cours des dernières années (par exemple Monaco et al. 2005 ; Monaco et Marchetti, 2007 ; Tsai et al. 2009).

Les méthodes simplifiées basées sur l'essai de dilatomètre (DMT) utilisent deux paramètres : l'indice de contrainte horizontale (K_D) et le module dilatomètre (E_D) comme indice pour évaluer la résistance de liquéfaction des sols. Plus précisément, les courbes de limites (CRR- K_D et CRR- E_D) sont établies sur la base des courbes de limites qui ont déjà été développées sur l'essai de pénétration standard (SPT) et l'essai de pénétration au cône (CPT). Un élément clé dans le développement des courbes de limites est la corrélation entre (K_D et/ou E_D) et les résistances de pénétrations corrigées ($(N_{160})_{cs}$) et ($(q_{c1N})_{cs}$) des essais SPT et CPT.

3.6.5.1. Méthode de Monaco et al (2005)

Monaco et al. (2005) ont proposé une nouvelle courbe de CRR basée sur une étude des corrélations entre la résistance de pointe du cône (q_c) et la densité relative (Dr), entre le nombre de coups (N) et entre Dr et l'indice de contrainte horizontale K_D .

La figure (15) montre les différentes propositions y compris celle de Monaco et al (2005). Le modèle basé sur cette courbe est exprimé comme suit :

$$CRR_{7.5} = 0.0107K_D^3 - 0.0741K_D^2 + 0.2169K_D - 0.1306 \quad (3.46)$$

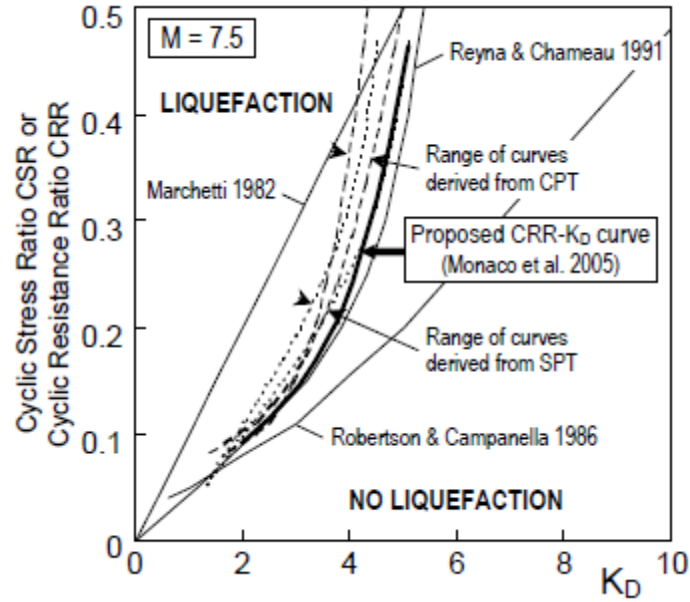


Figure 3. 15- Variation de CRR en fonction de K_D (Monaco et al. 2005)

3.6.5.2. Méthode de Grasso et Maugeri (2006)

Les auteurs de cette méthode ont utilisé l'essai dilatomètre de Marchetti sismique SDMT qui peut fournir deux paramètres K_D et V_S . En se basant sur les mesures SDMT de San Giuseppe La Rena, l'indice K_D a été modifié en donnant trois nouvelles formules de CRR.

$$CRR_{7.5} = 0.0908K_D^3 - 1.0174K_D^2 + 3.8466K_D - 4.5369 \quad (3.47)$$

$$CRR_{7.5} = 0.0308e^{(0.6054K_D)} \quad (3.48)$$

$$CRR = 0.0111K_D^{2.5307} \quad (3.49)$$

3.6.5.3. Méthode de Pai-Hsiang Tsai et al (2009)

Contrairement aux méthodes précédentes, Tsai et al (2009) ont utilisé le module dilatomètre E_D en plus de l'indice de contrainte horizontale K_D pour l'évaluation de CRR. Par conséquent, en se basant comme a été mentionné ci-dessus sur les corrélations entre les paramètres des essais in situ (SPT et CPT) deux nouvelles formules ont été proposées.

$$CRR_{7.5} = \exp \left[\left(\frac{K_D}{8.8} \right)^3 - \left(\frac{K_D}{6.5} \right)^2 + \left(\frac{K_D}{2.5} \right) - 3.1 \right] \quad (3.50)$$

$$CRR_{7.5} = \exp \left[\left(\frac{E_D}{49} \right)^3 - \left(\frac{E_D}{36.5} \right)^2 + \left(\frac{E_D}{23} \right) - 2.7 \right] \quad (3.51)$$

Les deux courbes qui représentent les formules ci-dessus sont montrées dans les figures (16 et 17).

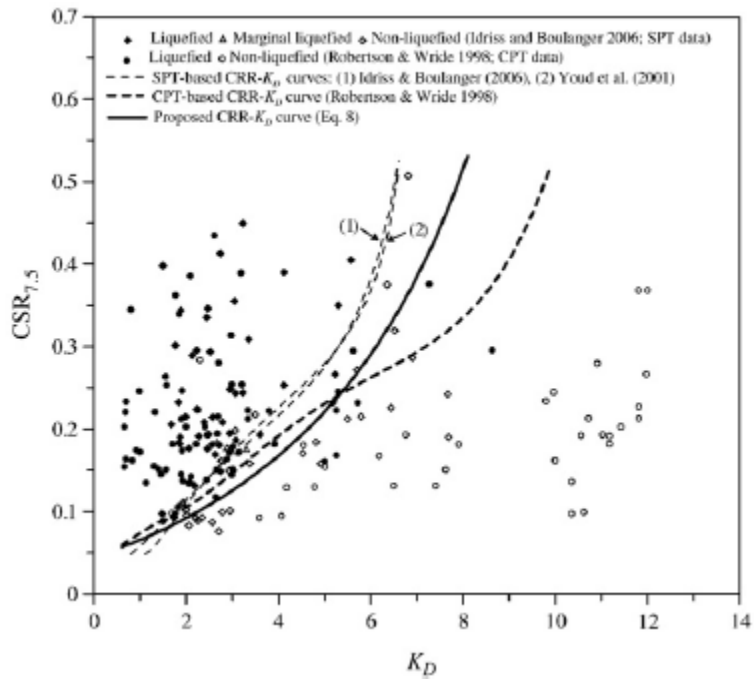


Figure 3.16- Variation de CRR en fonction K_D pour un sable propre (Tsai et al. 2009)

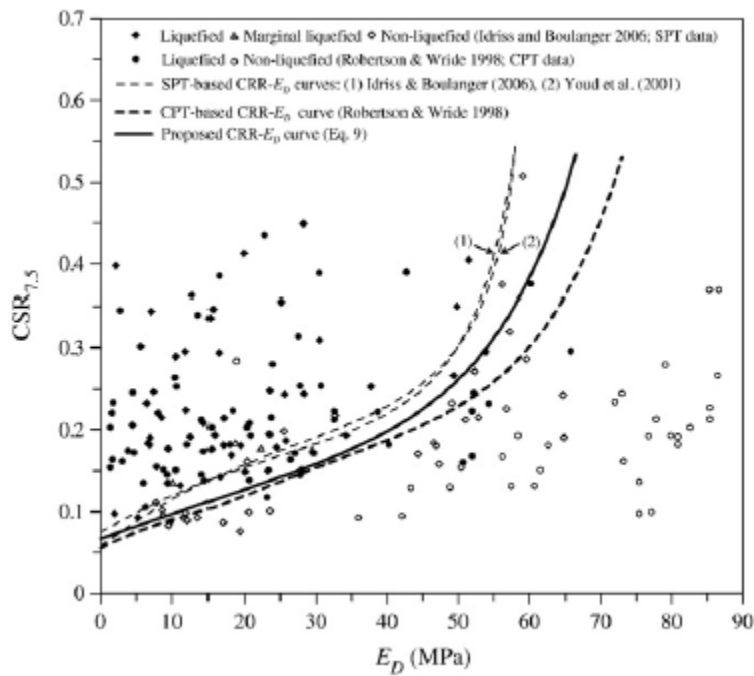


Figure 3.17- Variation de CRR en fonction E_D pour un sable propre (Tsai et al. 2009)

3.7. Conséquences de la liquéfaction

La liquéfaction est un phénomène très dangereux qui cause de multiples conséquences. Ces dernières dépendent des conditions de site, l'intensité et la durée du séisme et la nature de structures se trouvant dans ce site. De telles conséquences comprennent la déformation ou l'instabilité des masses des sols, l'augmentation des pressions latérales des terres contre les murs de soutènement, la perte de la capacité portante des fondations superficielles et profondes, flottation de conduites enterrées ou des tunnels, et tassement causé par la reconsolidation des sols liquéfiés. La figure 3.18 illustre le tassement qui s'est produit après le séisme de Kobe de 1995, au Japon.

Dans cette thèse, c'est le tassement dû à la liquéfaction qui va être détaillé dans les sections ci-dessous.



Figure 3.18- Tassement post-liquéfaction autour de la pile du viaduc dans le Port Island après le séisme de Kobe de 1995 (Idriss et Boulanger, 2008)

3.7.1. Tassement post-liquéfaction

Il est bien connu que les dépôts de sable lâche ont tendance à se densifier lorsqu'ils sont soumis aux secousses du tremblement de terre. Pour le dépôt de sable saturé, l'excès de pression d'eau interstitielle créé pendant le tremblement de terre, provoque la perte de résistance du sol ce qui conduit à la liquéfaction. Après l'agitation, l'excès d'eau se dissipe vers une zone où la pression de l'eau est relativement faible, habituellement à la surface du sol. Cette dissipation est habituellement accompagnée d'une reconsolidation du sable lâche (Ishihara et Yoshimine 1992). La

reconsolidation se manifeste à la surface du sol sous forme d'un déplacement verticale, habituellement appelé tassement post-liquéfaction ou tassement sismique.

Les tassements post-liquéfaction ont été observés après presque tous les grands tremblements de terre. Après le séisme de Niigata 1964, des tassements importants ont donné lieu à d'énormes destructions de constructions partout dans la ville de Niigata. L'évaluation et la prévision du tassement post-liquéfaction est devenu une tâche primordiale pour la conception des constructions situées dans des zones susceptibles à la liquéfaction. Plusieurs méthodes ont été développées pour étudier le tassement post-liquéfaction. Elles peuvent varier des méthodes semi-empiriques simplifiées aux méthodes complexes (analyse dynamique non linéaire). Les méthodes semi-empiriques sont basées sur les procédures simplifiées d'analyse de la liquéfaction. Il est commun dans toutes les procédures simplifiées d'estimer le tassement post-liquéfaction à partir de la déformation volumétrique.

3.7.2. Méthodes d'évaluation de tassement post-liquéfaction (S)

Plusieurs recherches ont été effectuées sur le calcul du tassement induit par liquéfaction en utilisant les essais in situ (SPT, CPT et V_s). Parmi ces méthodes simplifiées basées sur les essais in situ : Tokimatsu et Seed, 1987 ; Zhang et al. 2002 ; Idriss et Boulanger, 2008 ; Yi, 2010).

Comme nous avons mentionné ci-dessus, les méthodes simplifiées se basent dans leurs calculs du tassement post-liquéfaction sur des relations qui relient la déformation volumique au facteur de sécurité. Dans ce qui suit, nous allons exposer deux méthodes.

3.7.2.1. Méthode simplifiées SPT et CPT d'Idriss et Boulanger (2008)

Idriss et Boulanger ont utilisé les résultats d'Ishihara et Yoshimine (1992). Ces derniers ont observé que les déformations volumétriques (ε_v) qui se produisent lors de la reconsolidation post-liquéfaction des échantillons de sable ont été directement liées aux déformations de cisaillement maximales (γ_{max}) développées au cours du chargement cyclique non drainée et à la densité relative initiale (D_R) du sable, comme illustré à la figure 102. En se servant des corrélations (équation 1 et 2, Idriss et Boulanger 2006) et des courbes données dans la figure 103, Idriss et Boulanger ont proposé de nouvelles formules (équations 1 et 2) représentées par la figure 105. Cette figure montre la corrélation du facteur de sécurité de liquéfaction (FS) en fonction de la déformation volumétrique (ε_v). Pour les sites à terrain plat, loin de toute face libre (p. Ex., Rives, remblais), il est raisonnable de supposer que le maximum de la déformation volumique qui survient après un séisme égale au déplacement verticale. Ce qui implique que le tassement (le déplacement vertical total) égale à la somme des déformations dans chaque sous couche.

$$D_R = \sqrt{\frac{(N_1)_{60}}{46}} \quad (3.52)$$

$$D = 0.478(q_{c1N})^{0.264} - 1.063 \quad (3.53)$$

Les étapes de la procédure d'évaluation du tassement (Settlement)

Étape n° 1 : Calcul du facteur de sécurité de la liquéfaction (FS)

$$FS_{liq} = CRR/CSR$$

Étape n° 2 : Calcul de la déformation maximale (γ_{max})

$$\begin{cases} \gamma_{max} = 0, & \text{for } FS_{liq} \geq 2 \\ \gamma_{max} = \min(\gamma_{lim}, 0.035(2 - FS_{liq}) \cdot ((1 - F_\alpha)/(FS_{liq} - F_\alpha))) & \text{for } 2 > FS_{liq} > F_\alpha \\ \gamma_{max} = \gamma_{lim} & \text{for } FS_{liq} < F_\alpha \end{cases} \quad (3.54)$$

Où

$$\text{Méthode} \quad \gamma_{lim} = 1.859(1.1 - \sqrt{(N_1)60/46})^3 \geq 0 \quad (3.55)$$

$$\text{SPT} \quad F_\alpha = 0.032 + 0.69\sqrt{(N_1)60cs} - 0.13(N_1)60cs ; (N_1)60cs \geq 7 \quad (3.56)$$

$$\text{Méthode} \quad \gamma_{lim} = 1.859(2.163 - 0.478qc1cs^{0.264})^3 \geq 0 ; qc1Ncs \geq 69 \quad (3.57)$$

$$\text{CPT} \quad F_\alpha = -11.74 + 8.34(qc1Ncs)^{0.264} - 1.371(qc1Ncs)^{0.528} \quad (3.58)$$

Étape n° 3 : Calcul de déformation volumique post-liquéfaction (ε_v)

$$\text{Méthode SPT} \quad \varepsilon_v = 1.5 \cdot \text{Exp}(-0.369\sqrt{(N_1)60cs}) \cdot \min(0.08, \gamma_{max}) \quad (3.59)$$

$$\text{Méthode} \quad \varepsilon_v = 1.5 \cdot \text{exp}(2.551 - 1.147(qc1Ncs)^{0.264}) \cdot \min(0.08, \gamma_{max}) \quad (3.60)$$

$$\text{CPT} \quad \text{Avec } qc1Ncs \geq 21$$

Étape n° 4 : Calcul du tassement (S)

$$S_{v,1D} = \int_0^{Z_{max}} \varepsilon_v \cdot dz \quad (3.61)$$

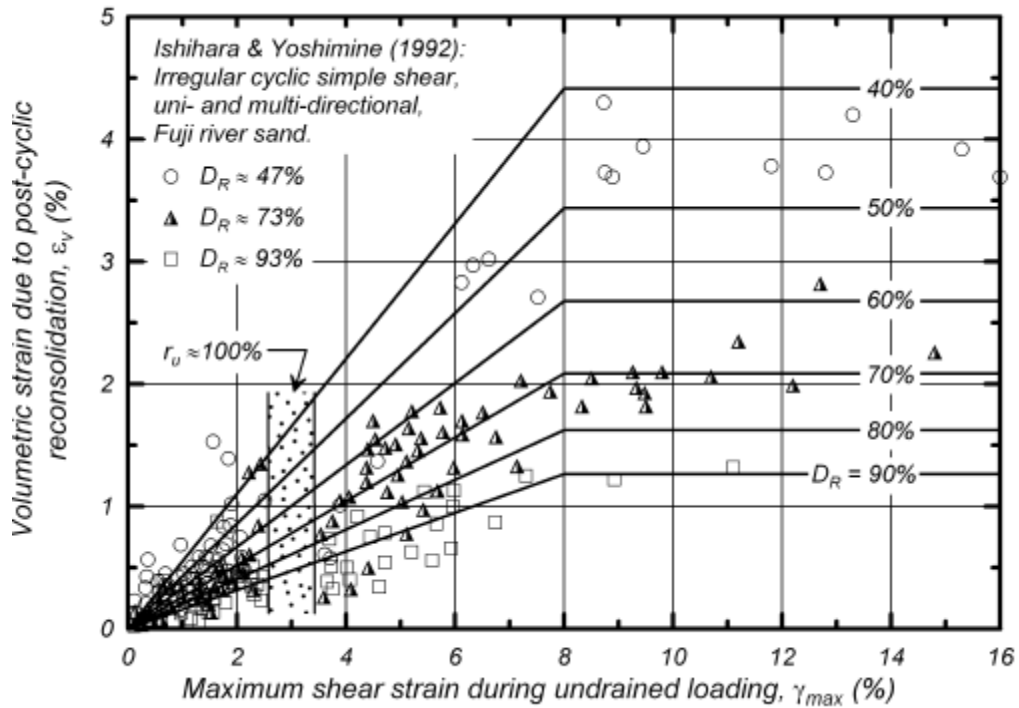


Figure 3. 19- Variation de γ_{max} en fonction de ϵ_v (Ishihara and Yoshimine 1992).

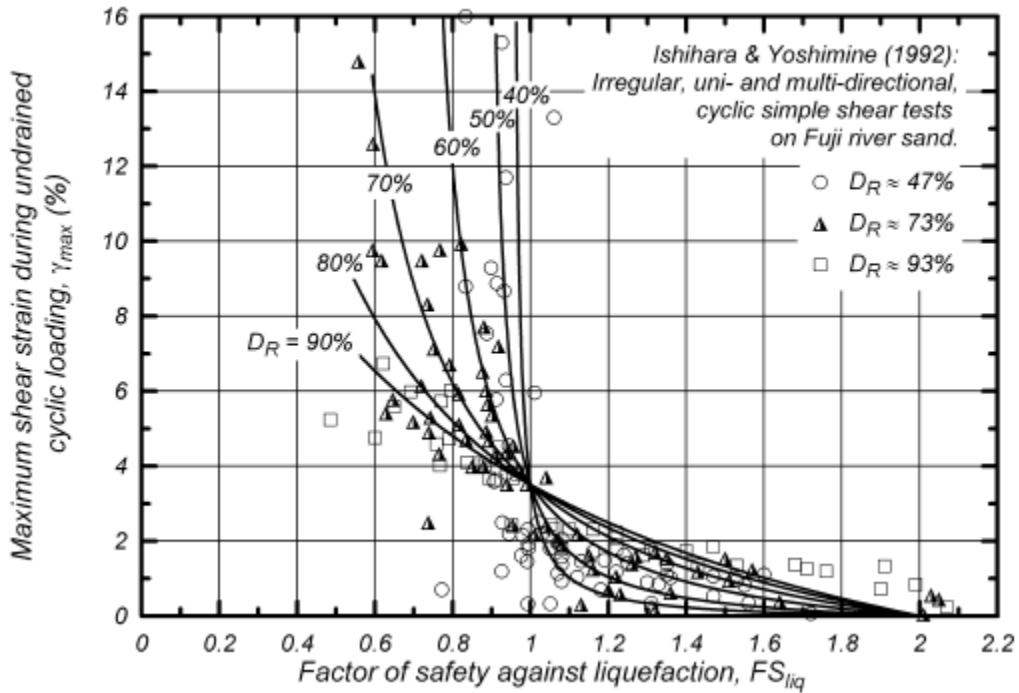


Figure 3. 20- Variation de FS en fonction de γ_{max} (Ishihara and Yoshimine 1992).

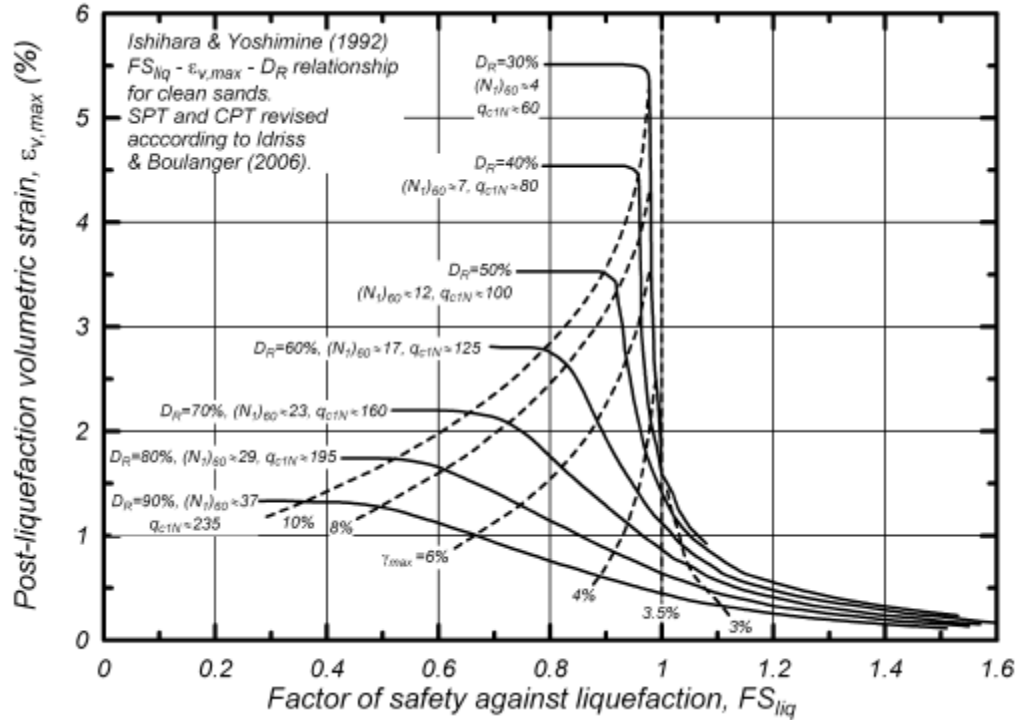


Figure 3. 21- Variation de FS en fonction de ε_V (Idriss et Boulanger, 2008).

3.7.2.2. Méthode de F.Yi (2010) basée sur la vitesse de cisaillement V_s

Yi(2010) a développé une méthode basée sur V_s pour estimer le tassement post-liquéfaction. Dans son développement, il a utilisé la méthode (Andrus et Stokoe 2000 ; Andrus et al. 2004) pour le calcul du facteur de sécurité de la liquéfaction (sections 3.6.4.1 et 3.6.4.2). Yi (2009b) a établi une relation entre V_s et D_r (équation 3.62). C'est à partir de cette relation que Yi (2010) a estimé la déformation volumique (ε_v) en fonction de la vitesse de cisaillement corrigée $(V_{s1})_{CS}$. Le détail de la méthode est donné ci-dessous.

$$DR = 17.974 \left[\frac{(V_{s1})_{CS}}{100} \right]^{1.974} \quad (3.62)$$

Les étapes de la procédure d'évaluation du tassement (Settlement)

Étape n° 1 : Calcul du facteur de sécurité

$$FS_{liq} = \frac{CRR_M}{CSR}$$

Où CSR : Rapport de contrainte cyclique calculé par la formule Seed et Idriss(1971)

CRR_M : Rapport de résistance cyclique calculé par la méthode (Andrus et al. 2004)

Étape n° 2 : Calcul de la déformation de cisaillement maximum, γ_{max}

$$\begin{cases} \gamma_{max} = 0, & \text{for } FS_{liq} \geq 2 \\ \gamma_{max} = 0.035(2 - FS_{liq}) \cdot ((1 - F_{\alpha}) / (FS_{liq} - F_{\alpha})) & \text{for } 2 > FS_{liq} > F_{\alpha} \\ \gamma_{max} = \infty & \text{for } FS_{liq} < F_{\alpha} \end{cases} \quad (3.63)$$

Où F_{α} est la valeur limite de F_{liq} , donné comme suit :

$$F_{\alpha} = 0.032 + 0.836((vs1)_{cs}/100)^{1.976} - 0.190((vs1)_{cs}/100)^{3.952} \quad (3.64)$$

Étape n° 3 : Calcul de la déformation volumique ε_v

$$\varepsilon_v = 1.5 \exp(-0.449((vs1)_{cs}/100)^{1.976}) \cdot \min(0.08, \gamma_{max})$$

Étape n° 4 : Tassement post-liquéfaction

$$S_{v,1D} = \int_0^{z_{max}} \varepsilon_v \cdot dz$$

3.8. Conclusion

Dans ce chapitre, nous avons exposé le problème de la liquéfaction en expliquant le sens physique du phénomène ainsi que ses méthodes d'évaluation. Les méthodes d'évaluation basées sur les essais in situ sont largement utilisées jusqu'à nos jours, vu les résultats encourageants publiés dans divers journaux. Nous avons également dressé un aperçu sur les conséquences du phénomène en décrivant seulement le tassement et ses méthodes de calcul car il fait partie de notre travail y compris la méthode de Yi(2010) qui a été couplée avec l'algorithme génétique multi-objectif NSGA-II et l'algorithme multi-objectif MOPSO. Dans le chapitre suivant, nous allons étudier un cas réel d'un tassement post-liquéfaction en appliquant les deux méthodes susmentionnées pour déterminer les valeurs du paramètre V_s .

4. Application des méthodes d'optimisation pour identifier V_s

4.1. Introduction

Dans ce chapitre, nous allons exposer les différents résultats obtenus par les méthodes d'optimisation : NSGA-II, l'algorithme génétique standard (AG) et multi-objectif par essaim de particules (MOPSO) sur un cas réel d'un tassement post-liquéfaction. Ce dernier a été observé dans la route d'accès de la plage de Moss Landing après le séisme de Loma Prieta 1989 (Idriss et Boulanger, 2008).

Pour déterminer les valeurs de la vitesse de cisaillement V_s de profil du sol en question, à partir d'un tassement observé dû au phénomène de liquéfaction, nous avons couplé une méthode simplifiée basée sur les données de la vitesse de cisaillement (Yi, F., 2010) avec trois méthodes métaheuristiques, à savoir l'algorithme génétique standard (AG), multi-objectif d'optimisation de type NSGA- II et multi-objectif d'optimisation par essaim de particules (MOPSO). Le choix de chaque méthode sera justifié dans la suite de ce chapitre. Pour la compréhension et l'enchaînement des idées de notre travail, une description du site d'où nous avons pris les données est nécessaire.

4.2. Description du site

Le tassement que nous avons utilisé dans notre étude représente le tassement observé post-liquéfaction dû au séisme de 1989 à Loma Prieta. Ce tassement observé s'est produit dans la route d'accès de la plage de Moss Landing. La figure (4.1) montre l'endroit où le tassement s'est produit et la figure (4.2) montre la situation géographique de Moss Landing.



Figure 4.1- la route d'accès de la plage de Moss Landing (Idriss et Boulanger, 2008).

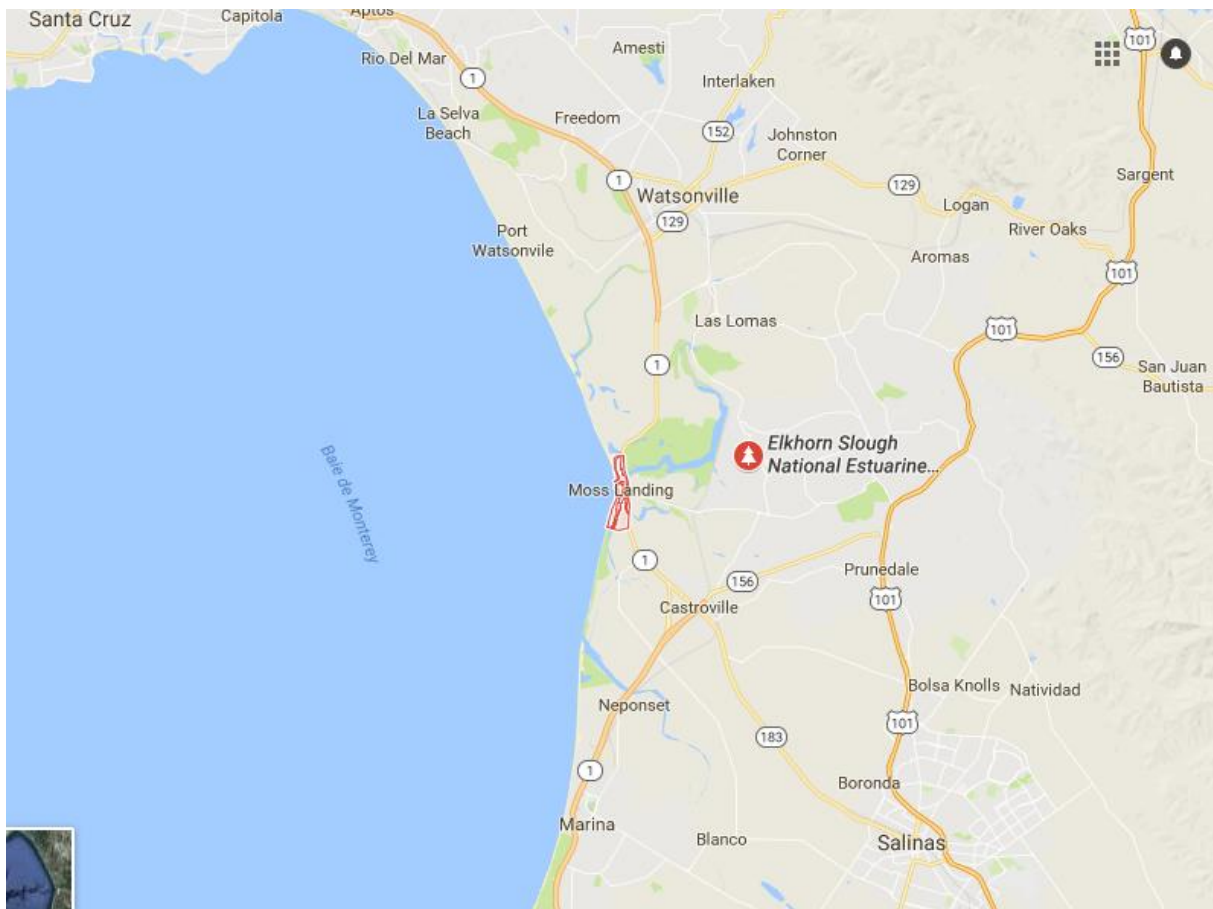


Figure 4.2-Vue aérienne de Moss Landing State Beach
(https://en.wikipedia.org/wiki/Moss_Landing,_California)

4.3. Formulation du problème

4.3.1. Problème d'optimisation mono-objectif 1

Au début de ce chapitre, nous avons parlé d'un tassement réel produit dans la route d'accès de la plage de Moss Landing. Ce tassement réel qui vaut (0.3 m), nous l'avons utilisé pour effectuer une analyse inverse afin de déterminer les valeurs du paramètre dynamique Vs. Notre problématique est formulée par la réponse à la question suivante :

- ✓ Quelles sont les valeurs du paramètre Vs qui induisent un tassement égal ou proche au tassement observé ?

Selon la question ci-dessus, le seul objectif visé est de trouver les valeurs de Vs qui induisent un tassement de 0.3. C'est ce qui fait de cette étude un problème d'optimisation mono-objectif. Pour ce faire, nous avons couplé l'algorithme génétique avec la méthode simplifiée Vs. Le modèle mathématique du problème d'optimisation mono-objectif est formulé comme suit :

$$\left\{ \begin{array}{l} \min F_{obj} = \left[\left(\sum_{i=1}^n S_i \right) - S_{obs} \right]^2 \Big)^{0.5} \\ x_{min} < x < x_{max} \end{array} \right. \quad (4.1)$$

Où

x_{min} et x_{max} : Limites inférieure et supérieure du vecteur des solutions,

F_{obj} : Fonction 'objectif' donnée par l'équation (4.2).

Pour effectuer cette analyse, nous avons programmé la méthode simplifiée de Vs et l'algorithme génétique standard en utilisant le logiciel Matlab. À chaque itération, l'algorithme génétique fait appel à la méthode simplifiée Vs pour évaluer les solutions. Ces solutions sont ensuite améliorées en employant les opérateurs génétiques (croisement et mutation) jusqu'à ce qu'un des critères d'arrêt soit atteint. Les étapes de notre analyse sont présentées sous forme d'un organigramme montré par la figure (4.3).

Algorithme Génétique Mono-objectif

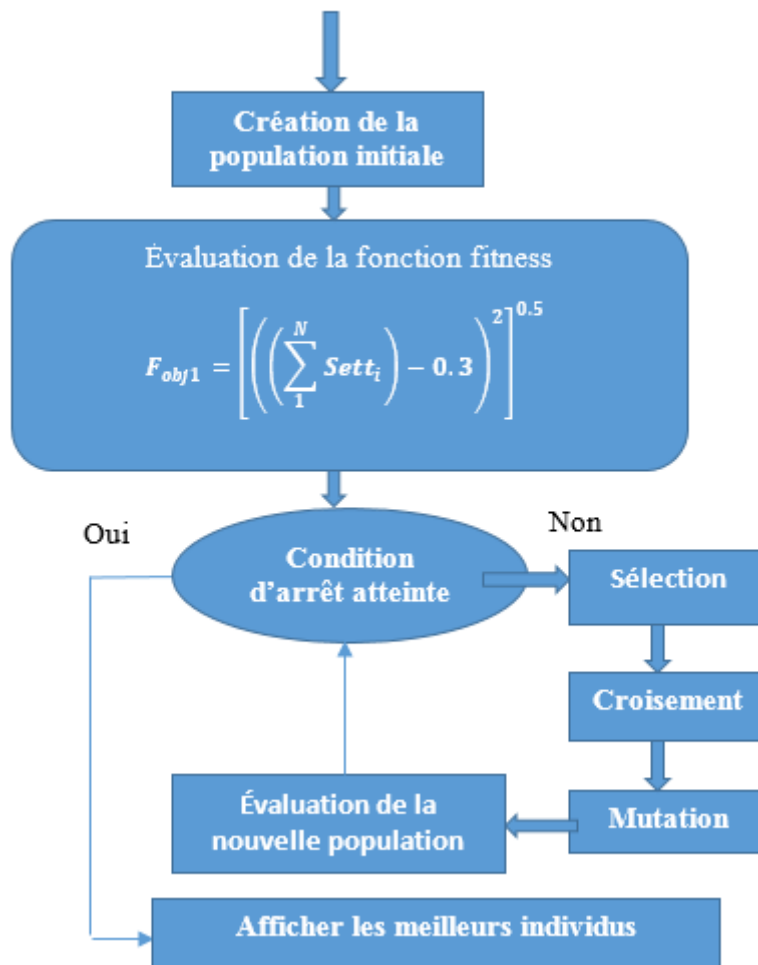


Figure 4.3- Procédure suivie dans l'optimisation mono-objectif par l'algorithme génétique.

$$F_{obj} = \left[\left(\left(\sum_{i=1}^n S_i \right) - S_{obs} \right)^2 \right]^{0.5} \quad (4.2)$$

Où

n : Nombre de variable (dans notre étude, n=12),

S_i : Tassement au point i,

S_{obs} : Tassement observé (dans notre étude =0.3).

Le profil du site de notre exemple est stratifié (Idriss et Boulanger, 2008), c'est pourquoi nous avons considéré la valeur de la vitesse de cisaillement de chaque profondeur comme une variable. Ce qui nous donne un problème avec 12 variables.

Tableau 4.1- Valeurs des paramètres de l'algorithme génétique

Taille de la population	Nombre de variables	Nombre de génération	Bornes	Tolérance
800	12	500	[100 ; 400]	0.00000001

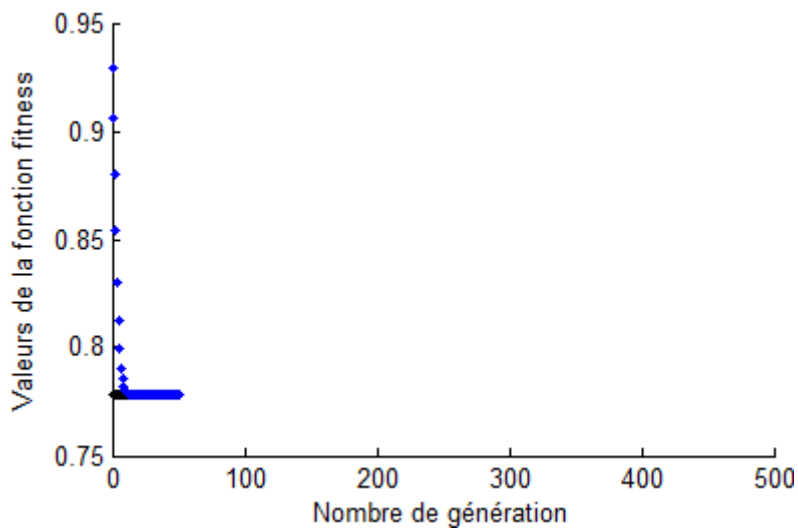


Figure 4.4- Variation de la fonction fitness en fonction de nombre de génération.

La figure (4.4) montre que l'algorithme génétique converge dès les premières générations vers l'optimum global. Afin de tester l'efficacité et l'efficacité de notre algorithme, nous avons établi un tableau comparatif (4.2) entre les valeurs cibles et optimisées. Les résultats obtenus indiquent que l'erreur (MAPE) entre les valeurs cibles et optimisées est très grand.

L'algorithme génétique est reconnu d'être efficace dans l'optimisation. Cependant, dans notre étude les résultats n'étaient pas acceptables. Cela est dû à la complexité de notre problème et à l'insuffisance des données pour le traiter comme un problème mono-objectif. La complexité de notre problème réside dans les points suivants :

- Nombre de variable à optimiser est élevé,
- Le profil du sol en question est stratifié,
- Nous avons fixé deux bornes pour tous les points de Vs ($x_{min}=100$ et $x_{max}= 400$ m /s).
- Une seule donnée pour tous les points (tassement observé, $S_{obs}=0.3$ m)

Tableau 4.2- Comparaison entre valeurs optimisées de (Vs) et valeurs réelles obtenues par l'algorithme génétique.

Profondeur(m)	Valeurs cibles (m/s)	Valeurs optimisées (m/s)	MAPE
1.8	146,61	108,736	25,833
2.6	105,46	112,986	7,137
3.4	120,25	114,986	4,377
4.1	130,03	114,986	11,569
4.9	134,29	114,986	14,375
5.6	151,59	114,986	24,147
6.4	183,09	114,986	37,197
7.2	188,43	114,986	38,977
7.9	192,53	114,986	40,276
9.4	183,12	114,986	37,207
10.2	169,56	114,986	32,185
11	172,31	114,986	33,268
Tassement	0,3156	0,4762	50,88

Pour toutes ces données, nous pouvons trouver plusieurs combinaisons différentes qui donnent toutes presque le même tassement global. Pour mettre en évidence l'explication que nous avons avancée ci-dessus, nous avons effectué plusieurs scénarios en modifiant les limites inférieure et supérieure du paramètre Vs. Les résultats sont résumés dans le tableau (4.3.). Ce dernier confirme ce que nous avons avancé ci-dessus. Ce résultat ne signifie rien sur l'efficacité des algorithmes génétiques qui sont reconnu par leurs efficacités dans l'optimisation. C'est à cause des données du problème qui sont insuffisantes. C'est pourquoi nous avons reformulé la question comme suit :

- ✓ Quelles sont les valeurs de Vs qui induisent un tassement global égal au tassement observé (0.3m) et qui doivent correspondre aux caractéristiques réelles du sol en question ?

Ce problème peut être résolu par deux procédures :

- Ajouter des données au problème mono-objectif en utilisant l'algorithme génétique standard.
- Ou traiter le problème en tant que multi-objectif en ajoutant un deuxième objectif.

Tableau 4.3- Variantes obtenus par algorithme génétique mono-objectif ga

Profondeur (m)	Valeurs cibles (m/s)	Variante 1	Variante 2	Variante 3
1,8	146,61	106,706	103,086	105,909
2,6	105,46	106,706	103,086	113,385
3,4	120,25	106,706	103,086	113,385
4,1	130,03	106,706	120,766	113,385
4,9	134,29	106,706	120,766	113,385
5,6	151,59	106,706	120,766	113,385
6,4	183,09	106,706	120,766	113,385
7,2	188,43	106,706	120,766	113,385
7,9	192,53	106,706	120,766	113,385
9,4	183,12	106,706	120,766	113,385
10,2	169,56	106,706	120,766	113,385
11	172,31	106,706	141,885	113,385
Tassement	0,3156	0,4774	0,4762	0,4770

4.3.2. Problème d'optimisation mono-objectif 2

Nous avons mentionné ci-dessus que la complexité de notre problème réside dans quatre points. Parmi lesquels, le manque de la valeur du tassement dans chaque point. L'exemple que nous avons utilisé dans notre travail pour optimiser le paramètre Vs est pris du rapport d'Idriss et Boulanger (2008). Ces auteurs l'ont utilisé pour valider les méthodes approchées d'évaluation du tassement post-liquéfaction. Les tassements du profil du sol en question obtenus par Idriss et Boulanger (2008) sont donnés dans le tableau (4.4).

Tableau 4.4- Valeurs du tassement dans chaque point.

Profondeur(m)	Tassement (m)
1.8	0.031
2.6	0.034
3.4	0.03
4.1	0.026
4.9	0.024
5.6	0.00
6.4	0.004
7.2	0.00
7.9	0.004
9.4	0.003
10.2	0.021
11	0.035

Dans cette partie, nous avons suivi presque la même procédure, c'est-à-dire la même méthode (algorithme génétique standard). La modification a porté sur la formule de la fonction 'objectif' où le tassement observé global est remplacé par le tassement dans chaque point du profil du sol. Le modèle mathématique du problème est formulé comme suit :

$$\begin{cases} \min F_{obj} \\ x_{min} < x < x_{max} \end{cases} \quad (4.3)$$

x_{min} et x_{max} : Limites inférieure et supérieure du vecteur des solutions,

F_{obj} : Fonction 'objectif' donnée par l'équation (4.3).

$$F_{obj} = \left[\frac{1}{n} \times \sum_{i=1}^n (S_{cible(i)} - S_{optimisé(i)})^2 \right]^{0.5} \quad (4.4)$$

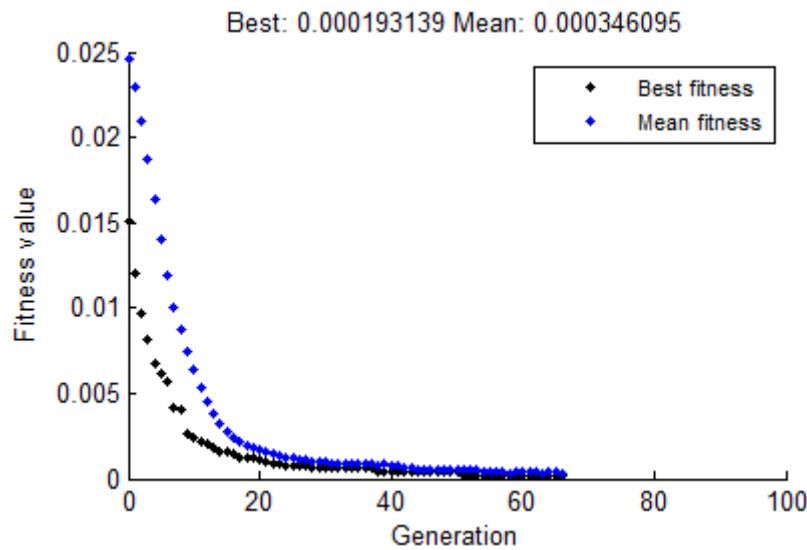


Figure 4.5- Évolution de l'algorithme génétique 2 en fonction du nombre de génération.

La figure (4.5) indique que 65 itération a été largement suffisantes pour que la variation moyenne de best fitness (F_{obj}) atteinte sa valeurs minimale. Afin de tester la performance de l'algorithme génétique dans l'optimisation de Vs après la modification portée au problème, une comparaison entre les valeurs cibles et optimisées est réalisée. Pour mettre en relief les résultats, les deux critères de mesure d'erreur à savoir, le coefficient de détermination R^2 et le pourcentage d'erreur absolue moyenne (MAPE) donnés par les équations (4.5) et (4.6), ont été calculés.

Les résultats exposés dans la figure (4.6) et le tableau (4.5) indiquent que l'algorithme génétique a donné de bons résultats avec des erreurs très acceptables (MAPE= 4.651 % et $R^2= 0.886$).

$$R^2 = 1 - \left[\frac{\sum_1^n (x_{op(i)} - x_{cible(i)})^2}{(\sum_1^n x_{cible(i)} - moy(x_{cible(i)}))^2} \right] \quad (4.5)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{x_{cible(i)} - x_{op(i)}}{x_{cible(i)}} \right| \quad (4.6)$$

Où

n : Nombre des variables,

$x_{op(i)}$: La valeur optimisée de i ,

$x_{cible(i)}$: La valeur cible de i .

Tableau 4.5- Corrélation entre les valeurs cibles et optimisées de Vs

Vs cibles	Vs optimisées	Erreur (MAPE) %
146,61	132,251	9,794
105,46	100	5,177
120,25	120,33	0,066
130,03	130,001	0,0222
134,29	134,268	0,016
151,59	151,572	0,0119
183,09	221,681	21,078
188,43	188,33	0,0533
192,53	229,945	19,434
183,12	183,054	0,036
169,56	169,438	0,0718
172,31	172,398	0,051
MAPE = 4,651 %		

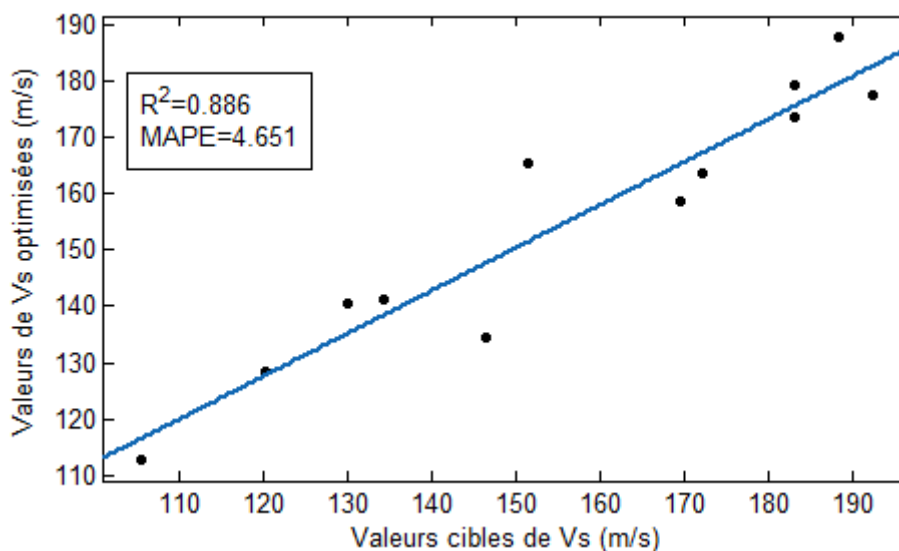


Figure 4.6- Corrélation entre les valeurs cibles et optimisées de Vs.

Dans la suite, nous allons procéder d'une manière différente à celle optée ci-dessus. Nous allons traiter le problème en tant qu'un problème multi-objectif.

4.3.3. Problème d'optimisation multi-objectif

Pour transformer un problème mono-objectif à un problème multi-objectif, il faut ajouter un deuxième objectif. Le but de ce deuxième objectif est d'orienter l'algorithme vers la zone où se trouvent les meilleures solutions. Dans notre cas, les deux objectifs sont :

- ✓ 1^{er} Objectif : Valeurs de la vitesse de cisaillement doivent induire un tassement global égal au tassement observé (0.3m).
- ✓ 2^{ème} Objectif : Valeurs de Vs doivent être proches des valeurs réelles du sol.

La première fonction 'objectif', c'est celle utilisée par la méthode d'algorithme génétique standard. Concernant la deuxième fonction 'objectif', nous avons opté pour une formule reliant Vs avec un autre paramètre géotechnique du sol.

Dans la littérature, beaucoup de formules ont été développées reliant Vs et les paramètres géotechniques tels que : le nombre de coup N-SPT, la résistance de pointe (qc) et la contrainte verticale effective σ'_v, \dots , etc. Dans Le tableau 4.6, nous avons donné quelques formules.

Dans cette étude, nous avons opté pour deux formules. La première est celle qui relie Vs avec le N-SPT de Hasancebi et Ulusay (2007) et la deuxième formule relie Vs avec σ'_v de Ku et al (2016).

Tableau 4.6- Formules reliant Vs aux paramètres géotechniques

Formules reliant Vs aux paramètres géotechniques	Type de sol	Référence
$V_s = 26.56\sigma'_v{}^{0.42}$	Tout type de sol	(Ku et al., 2016)
$V_{s1} = 60.3qc_1^{0.23}$	Sol pulvérulent	(Robertson, et al., 1992)
$V_s = 13.18qc^{0.192}\sigma'_v{}^{0.179}$	Sol pulvérulent	(Hegazy et Mayne, 1995)
$V_s = 90N^{0.309}$	Tout type de sol	(Hasancebi et Ulusay, 2007)

Le modèle mathématique qui représente un problème multi-objectif est formulé comme suit :

$$\begin{cases} \min f = y = \{f_1(x), f_2(x)\} \\ x = \{x_1, x_2, \dots, x_n\} \\ x_{min} < x < x_{max} \end{cases} \quad (4.7)$$

Où

x : Vecteur des solutions,

x_{min} et x_{max} : Limites inférieure et supérieure,

$f_1(x)$ et $f_2(x)$: Fonctions 'objectifs' données par les équations (4.9) et (4.10).

4.3.3.1. Application de l'algorithme génétique NSGA II

Le tableau (4.7) donne les valeurs des paramètres de la méthode NSGA-II. Cette dernière est une extension de l'algorithme génétique standard, c'est pourquoi les paramètres sont presque les mêmes en ajoutant seulement le coefficient de fraction de Pareto. Ce paramètre représente le pourcentage des individus qui seront affichés dans le front de Pareto.

Tableau 4.7- Paramètres de NSGA-II

Nombre de variable	12
Taille	100
Bornes inférieur et supérieur	[100 ; 400]
Génération	1000
Fraction du front de Pareto	0.35
Tolérance	0.000001
Probabilité de croisement	0.8
Probabilité de mutation	0.2

La figure (4.7) représente le processus d'optimisation suivi par la méthode NSGA-II

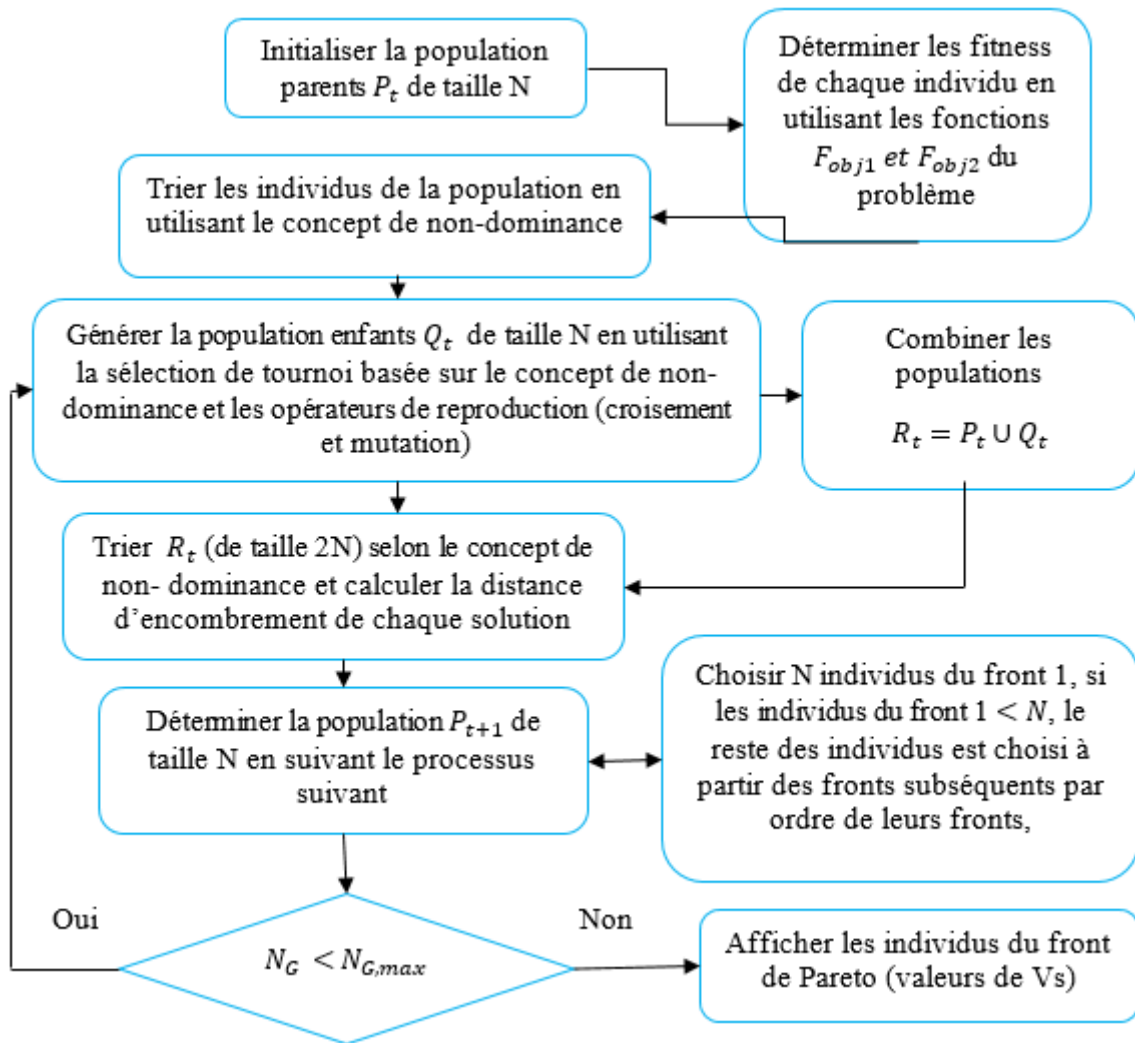


Figure 4.7- Processus d'optimisation suivi par l'algorithme NSGA-II

4.3.3.1.1 Vs en fonction de N-SPT

La formule de de Hasancebi et Ulusay (2007) est donnée par l'équation (4.8). La deuxième fonction 'objectif' donnée par l'équation (4.10) est utilisée comme une contrainte qui aide l'algorithme à explorer l'espace de recherche contenant les solutions optimales.

$$V_s = 90N^{0.309} \quad (4.8)$$

$$F_{obj1} = \left[\left(\left(\sum_1^n S_i \right) - S_{obs} \right)^2 \right]^{0.5} \quad (4.9)$$

$$F_{obj2} = [(x_i - V_s)^2]^{0.5} \quad (4.10)$$

Où

N : Nombre de coups de l'essai SPT,

x_i : Valeur de la variable x indice i

V_s : Valeur de Vs calculée par l'équation (4.8).

- **Front de Pareto**

Le front de Pareto montré dans la figure (4.8) représente les meilleures solutions de l'espace de recherche. Autrement dit, les solutions non dominées qui représentent un compromis entre plusieurs objectifs (Pour notre étude deux objectifs). D'après les valeurs données en abscisse de la figure (4.8) qui varient dans l'intervalle [0 – 0,08], nous pouvons dire que la fonction 'objectif' (F_{obj1}) a été bien minimisée. Par contre, les valeurs données en ordonnée de la fonction 'objectif' (F_{obj2}) varient dans l'intervalle [2-6]. Ce qui indique que le deuxième objectif a servi seulement comme une contrainte qui guide le processus de recherche vers la zone prometteuse.

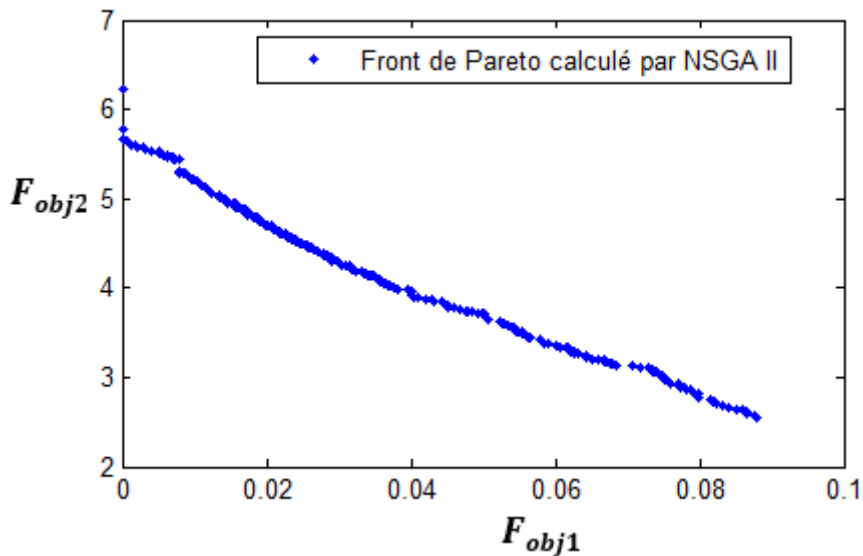


Figure 4.8- Front de Pareto

Les figures de (4.9) jusqu'à (4.12) montrent l'évolution de la population initiale vers le front de Pareto en utilisant les opérateurs de la méthode NSGA-II. Dans la figure (4.9), les individus sont un peu éloignés du front de Pareto, mais à partir de la cinquième génération, les individus commencent à converger vers les solutions optimales et à la (600)^{ème} génération (figure 4.12), ils deviennent de plus en plus proches du front.

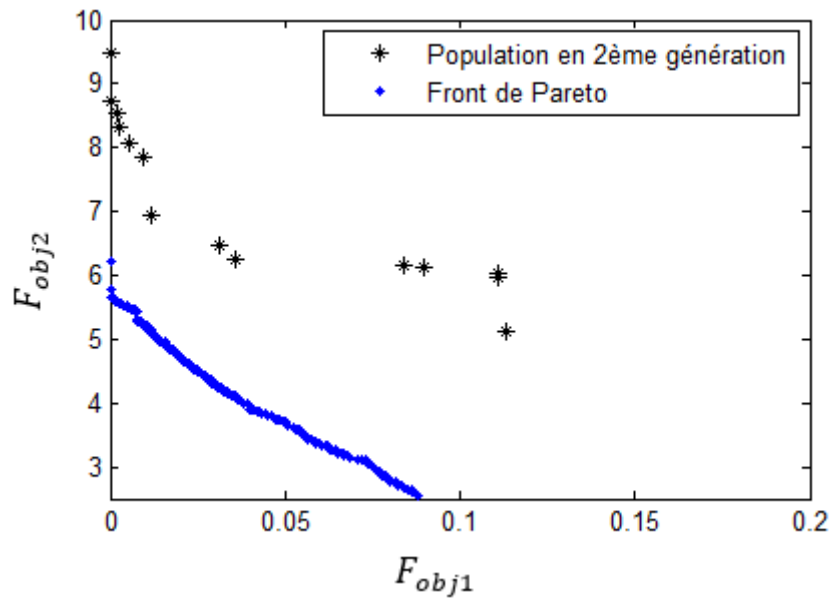


Figure 4.9- L'évolution de la population en 2^{ème} génération

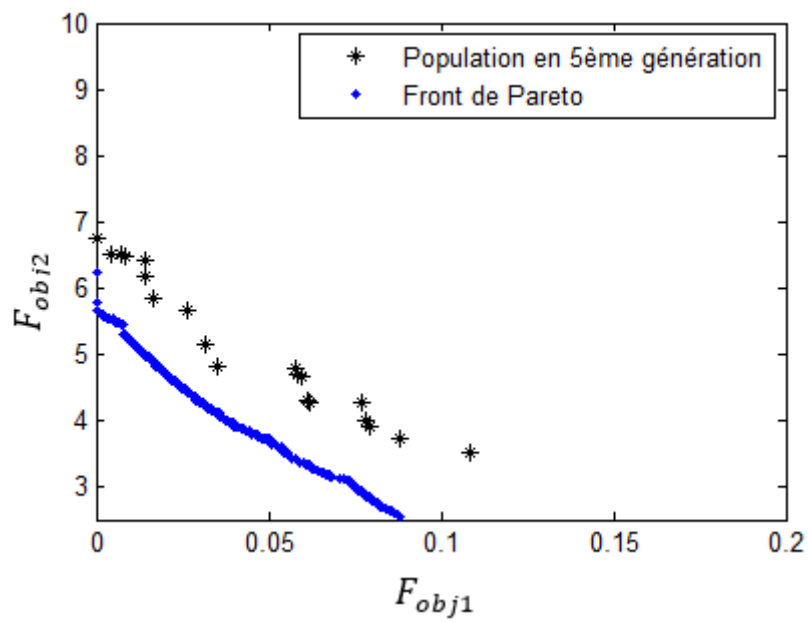


Figure 4.10- L'évolution de la population en 5^{ème} génération

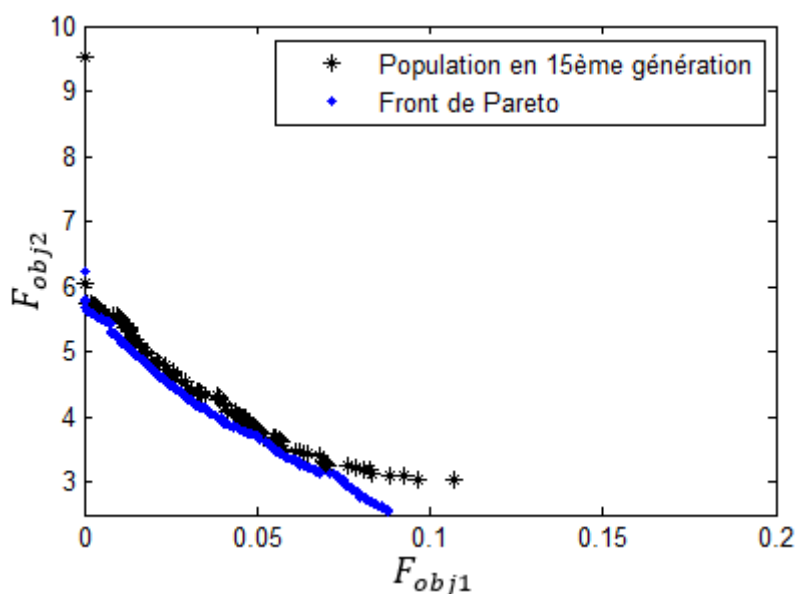


Figure 4.11- L'évolution de la population en 15^{ème} génération

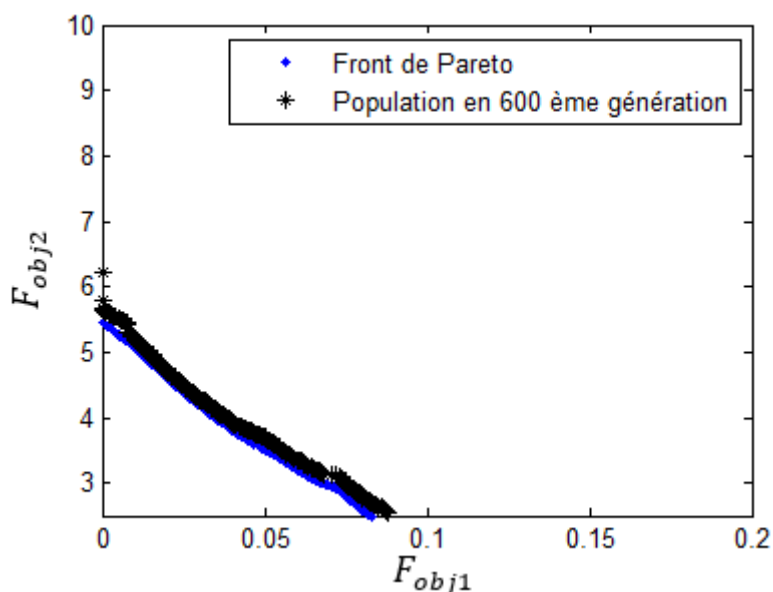


Figure 4.12- L'évolution de la population en 600^{ème} génération

Afin de vérifier l'efficacité de l'algorithme NSGA II, une comparaison entre les valeurs cibles et optimisées a été faite. Les résultats exposés dans la figure (4.13) et le tableau (4.8) indiquent que les erreurs ($R^2=0,9019$ et $MAPE= 5,95 \%$) sont très acceptables. En outre, l'erreur sur le tassement global obtenu est de $3,041 \%$. Ce qui prouve la performance de la méthode NSGA-II dans l'optimisation de Vs.

Tableau 4.8- Comparaison entre les valeurs cibles et optimisées par NSGA II.

h(m)	Vs cibles (m/s)	Vs optimisées (m/s)	MAPE
1.8	146,61	134,48	8,274
2.6	105,46	112,76	6,922
3.4	120,25	128,34	6,728
4.1	130,03	140,61	8,137
4.9	134,29	141,17	5,123
5.6	151,59	165,44	9,136
6.4	183,09	179,23	2,108
7.2	188,43	187,88	0,292
7.9	192,53	177,43	7,843
9.4	183,12	173,82	5,079
10.2	169,56	158,65	6,434
11	172,31	163,5	5,113
Tassement global	0,3156	0,3252	3.041

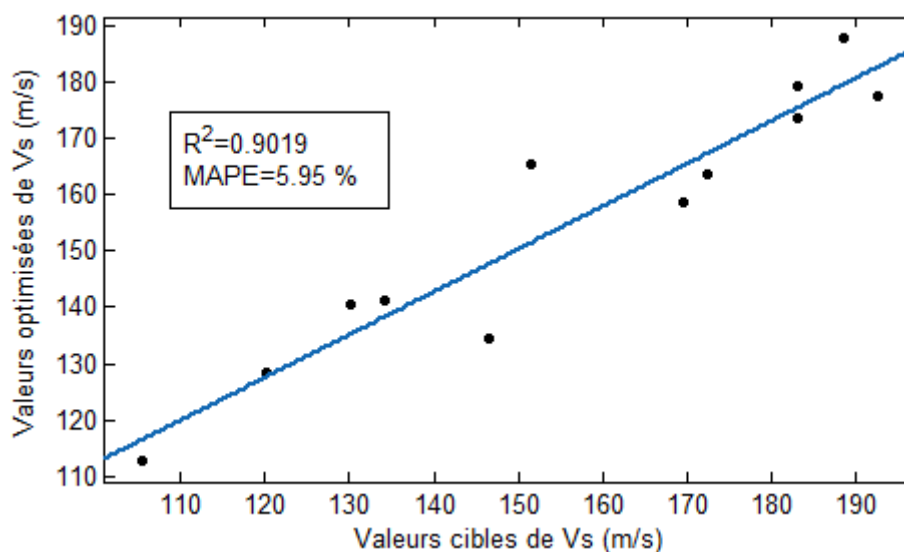


Figure 4.13- Corrélation entre les valeurs réelles et valeurs optimisées

Vu les résultats encourageants obtenus par la méthode NSGA-II en utilisant la formule reliant Vs et N-SPT, nous avons décidé d'utiliser un autre paramètre géotechnique. Pour cela, nous avons choisi la contrainte effective verticale (σ'_v) en utilisant la formule de Ku et al (2016).

4.3.3.2 Vs en fonction de σ'_v

La formule de Ku et al (2016) est donnée par l'équation (4.11). La deuxième fonction 'objectif' donnée par l'équation (4.12) est utilisée comme une contrainte qui aide la méthode NSGA-II à explorer la zone des solutions optimales.

$$V_s = 26.56(\sigma'_v)^{0.42} \quad (4.11)$$

$$F_{obj2} = [(x_i - V_s)^2]^{0.5} \quad (4.12)$$

Où σ'_v : Contrainte effective verticale,

V_s : Valeur de Vs calculée par l'équation (4.11).

- **Front de Pareto du deuxième cas**

Le front de Pareto qui se compose de 35 individus, est représenté par la figure (4.13). La figure indique les images des solutions non dominées. En abscisse, les valeurs de la fonction 'objectif 1' varient dans l'intervalle de [0.78 ; 0.88]. En ordonnée, les valeurs de la fonction 'objectif 2' varient dans l'intervalle de [4 ; 35]. Ce qui prouve que le deuxième objectif sert seulement comme un guide pour l'algorithme et que la fonction 'objectif 1' a été bien minimisée.

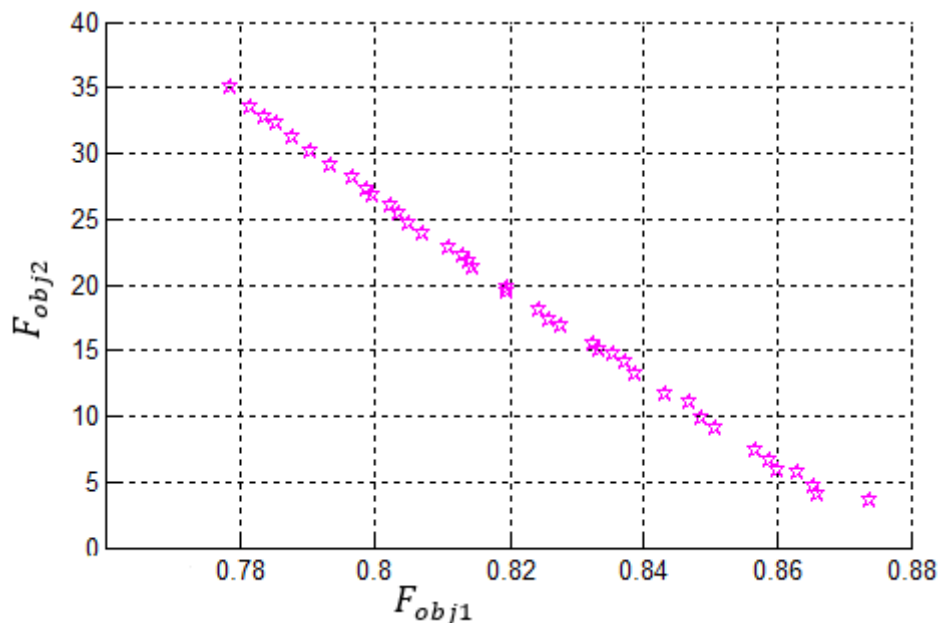


Figure 4.14- Front de Pareto pour le cas de Vs en fonction de σ'_v

- **Résultats et discussion du deuxième cas**

Pour le deuxième cas, nous avons également suivi la même démarche que celle du premier cas. Les résultats présentés dans la figure (4.15) et le tableau (4.9) indiquent que la méthode NSGA-II a également bien réussi à optimiser le paramètre Vs avec des erreurs satisfaisantes de ($R^2=0.738$ et $MAPE=7.885\%$).

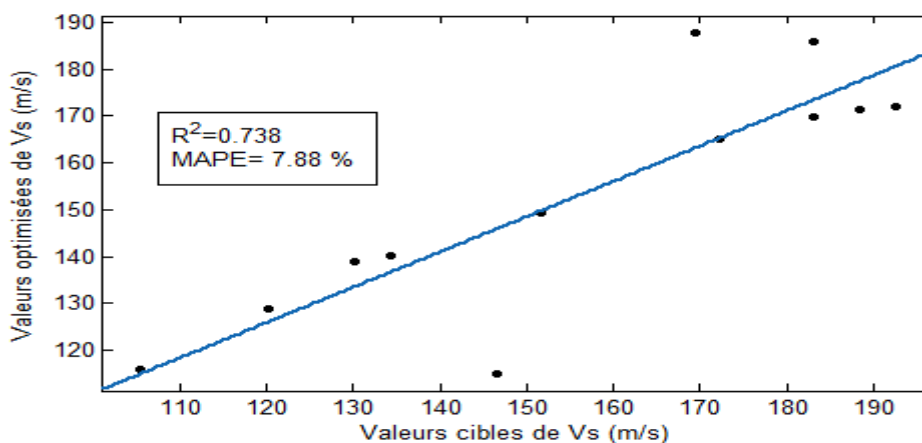


Figure 4.15- Corrélation entre les valeurs cibles et optimisées par NSGA II (2^{ème} cas)

Tableau 4.9- Corrélation entre valeurs cibles et optimisées de Vs

Valeurs cibles de Vs (m/s)	Valeurs optimisées de Vs (m/s)	MAPE
146,61	114,959	21,589
105,46	116,047	10,039
120,25	128,754	7,072
130,03	138,891	6,815
134,29	140,059	4,296
151,59	149,196	1,579
183,09	169,921	7,193
188,43	171,395	9,040
192,53	172,148	10,586
183,12	185,948	1,544
169,56	187,736	10,720
172,31	165,118	4,174
	MAPE=	7,887 %

Vu les résultats encourageants obtenus par la méthode NSGA-II et pour confirmer la performance des méthodes d'optimisation multi-objectif sur notre exemple, nous avons décidé d'appliquer une autre méthode d'optimisation de type multi-objectif. La méthode que nous avons choisie est l'algorithme multi-objectif par essaim de particules (MOPSO).

4.3.3.2. Application de la Méthode d'Optimisation Multi-objectif par Essaim de Particules (MOPSO)

MOPSO est une méthode métaheuristique inspiré du comportement du vol des oiseaux en groupe. Pour cette méthode, nous avons également utilisé deux cas où nous avons utilisé les mêmes paramètres que ceux de la méthode NSGA-II (le nombre de coups N-SPT et la contrainte effective verticale σ_v').

La procédure suivie dans cette partie est présentée sous forme d'algorithme montré par la figure (4.16) et les paramètres de la méthode sont donnés dans le tableau (4.10).

Tableau 4.10- Paramètres du problème et de MOPSO

Paramètres du problème et de l'algorithme MOPSO	
Taille de la population	200
Taille de l'archive	100
Taux de mutation	0.1
Nombre de génération	500
Poids d'inertie	0.5
Coefficient d'apprentissage personnel et global	2
Type de sélection	Sélection de tournoi basée sur le concept de dominance et d'encombrement
Nombre de grille	7

4.3.3.2.1. Vs en fonction de N-SPT

La relation qui relie Vs au nombre de coups N-SPT est donnée par l'équation (4.13). Cette relation est utilisée dans la deuxième fonction 'objectif ' comme une contrainte qui guide l'algorithme vers le front de Pareto.

$$V_s = 90N^{0.309} \quad (4.13)$$

$$F_{obj1} = \left[\left(\left(\sum_1^n S_i \right) - S_{obs} \right)^2 \right]^{0.5} \quad (4.14)$$

$$F_{obj2} = [(x_i - V_s)^2]^{0.5} \quad (4.15)$$

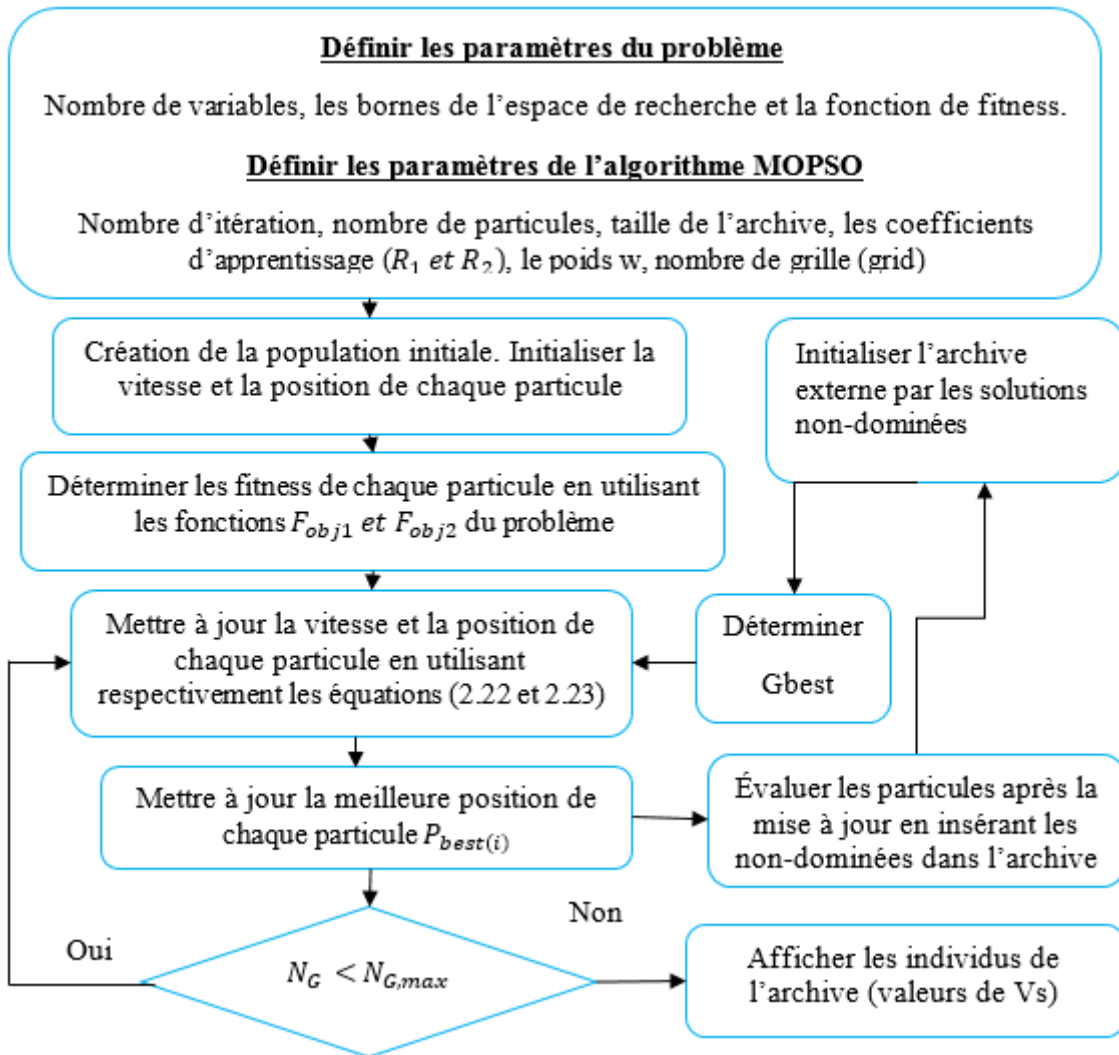


Figure 4.16- Méthode d'Optimisation Multiobjectifs par Essaim de Particules

- **Front de Pareto pour la méthode MOPSO**

Le front de Pareto de la méthode MOPSO représenté par La figure (4.13) se compose des meilleurs solutions stockées dans l'archive externe. Le nombre de ces solutions égale à la taille de l'archive (=100 dans notre étude). Les valeurs des fonctions 'objectifs' représentent les images des solutions non dominées. En abscisse, les valeurs de la fonction 'objectif 1' a été bien minimisée. En ordonnée, les valeurs de la fonction 'objectif 2' varient dans l'intervalle de [15 ; 70]. Ces valeurs nous confirment encore que la deuxième fonction 'objectif' sert uniquement à guider l'algorithme vers les solutions optimales.

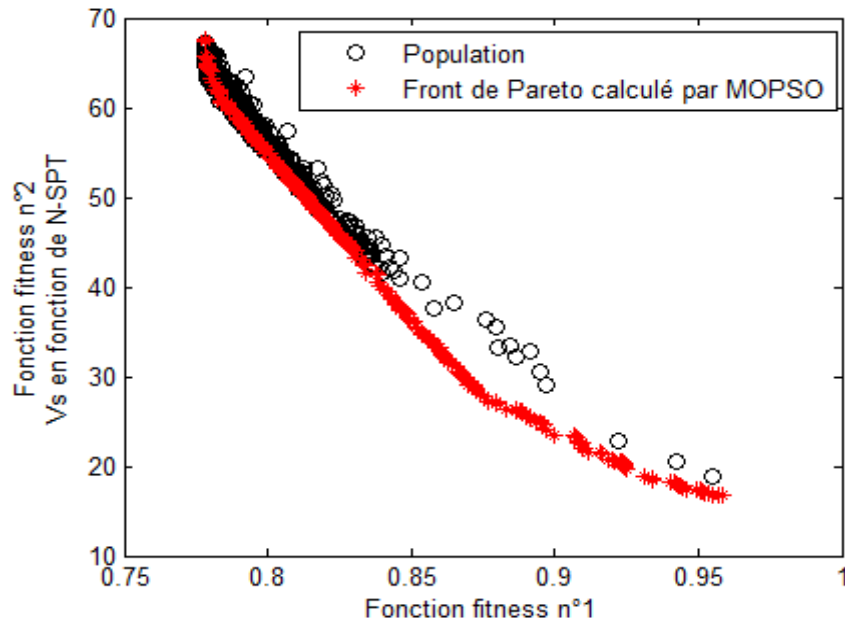


Figure 4.17- Évolution de la population vers le front de Pareto par MOPSO (1^{er} cas).

- **Résultats et discussion**

Les résultats présentés par la figure (4.18) et le tableau (4.11) indiquent que la méthode MOPSO a également bien réussi à trouver les bonnes solutions du paramètre Vs avec des erreurs très acceptables de ($R^2=0.783$ et $MAPE= 6,472 \%$). Ce qui prouve l'efficacité de la méthode MOPSO dans l'optimisation de Vs.

Tableau 4.11- Comparaison entre valeurs cibles et valeurs optimisées par MOPSO

Vs cibles (m/s)	Vs optimisées (m/s)	MAPE
146,61	126,324	13,836
105,46	108,209	2,606
120,25	132,185	9,925
130,03	123,963	4,666
134,29	140,383	4,537
151,59	152,861	0,839
183,09	164,273	10,278
188,43	171,890	8,778
192,53	173,887	9,683
183,12	181,153	1,074
169,56	177,767	4,840
172,31	160,927	6,606
	MAPE	6,472 %

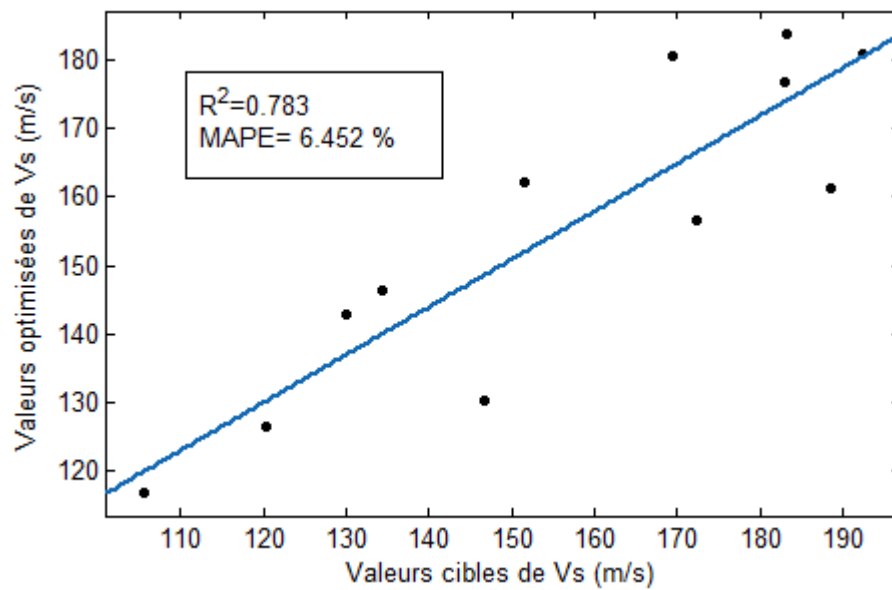


Figure 4.18- Corrélation entre les valeurs réelles et valeurs optimisées

4.3.3.2.2. Vs en fonction de σ'_v

Également dans ce cas, nous avons utilisé la formule qui relie Vs avec la contrainte verticale effective σ'_v donnée par l'équation (4.16) comme un guide à notre méthode.

$$V_s = 26.56(\sigma'_v)^{0.42} \tag{4.16}$$

$$F_{obj2} = [(x_i - V_s)^2]^{0.5} \tag{4.17}$$

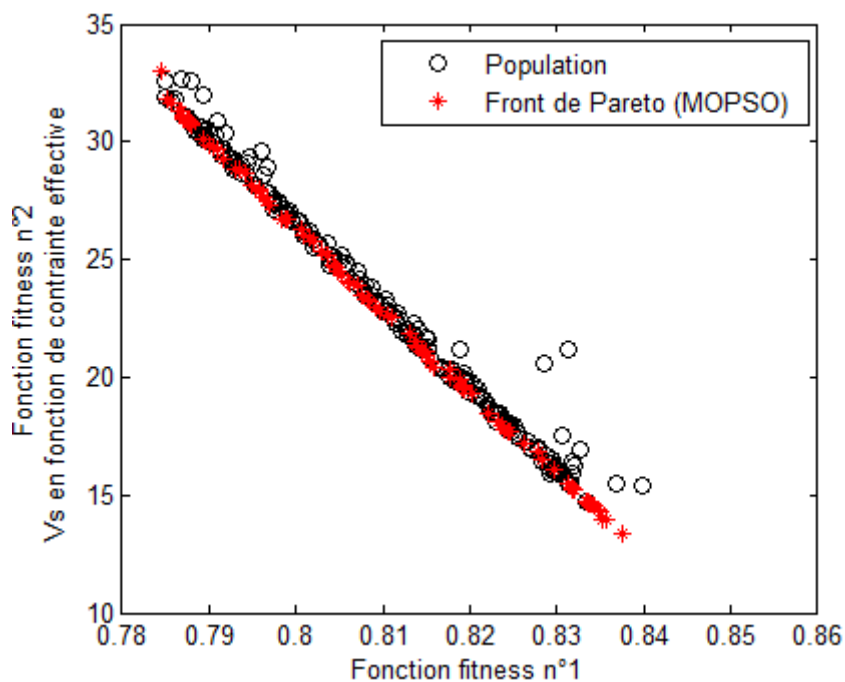


Figure 4.19- Évolution de la population vers le front de Pareto par MOPSO (2^{ème} cas).

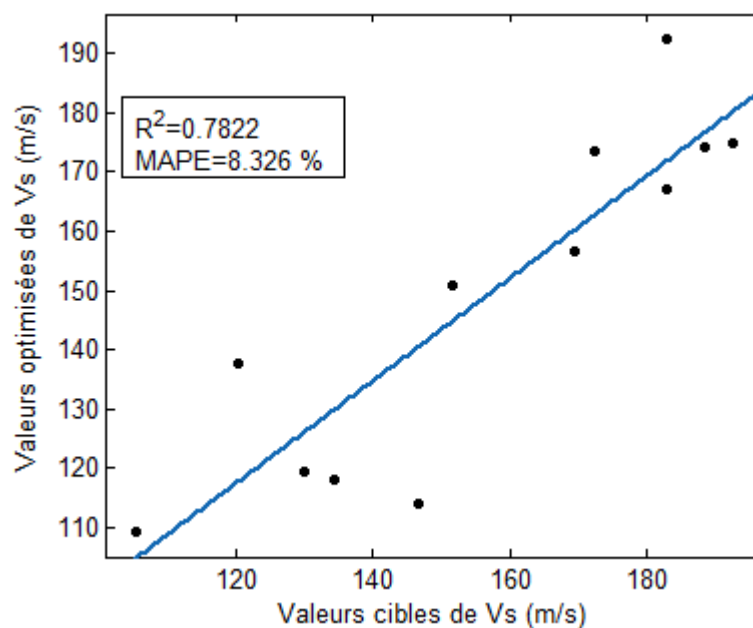


Figure 4.20- Corrélation entre les valeurs cibles et les valeurs optimisées par MOPSO

La figure (4.19) représente l'évolution des particules d'essaims vers le front de Pareto pour le 2^{ème} cas. Même pour ce cas la figure (4.20) montre également que MOPSO a donné de bons résultats avec des critères de performance ($R^2=0.7822$ et $MAPE=8.326\%$).

Tableau 4.12- Comparaison entre valeurs cibles et valeurs optimisées par MOPSO (2^{ème} cas)

H (m)	Vs cibles (m/s)	Vs optimisées (m/s)	MAPE
1.8	146,61	113,82	22,365
2.6	105,46	109,19	3,537
3.4	120,25	137,51	14,353
4.1	130,03	119,4	8,175
4.9	134,29	118,11	12,049
5.6	151,59	150,96	0,416
6.4	183,09	167,13	8,717
7.2	188,43	174,23	7,536
7.9	192,53	174,75	9,235
9.4	183,12	192,57	5,161
10.2	169,56	156,49	7,708
11	172,31	173,45	0,662
		MAPE	8,326 %

4.4. Conclusion

Dans ce chapitre, nous avons exposé les résultats obtenus par les différentes méthodes d'optimisation. Au début, nous avons traité le problème en tant qu'un problème mono-objectif. Pour ce type de problème l'algorithme génétique standard a donné de bons résultats en optimisation selon plusieurs études faites dans ce sens. Or dans notre cas, l'écart entre les valeurs cibles et optimisées était grand car tout simplement les données du problème sont insuffisantes pour le traiter comme un problème mono-objectif. C'est pourquoi, nous avons opté pour deux approches : Soit nous ajoutons des données au problème mono-objectif ou bien le traiter comme un problème multi-objectif en ajoutant un deuxième objectif.

Pour la première approche, nous avons utilisé l'algorithme génétique en modifiant le tassement global observé par le tassement dans chaque point. Les résultats obtenus par cette approche étaient très satisfaisants avec des erreurs très acceptables (MAPE= 4.651 % et $R^2= 0.886$). Ce qui prouve que l'algorithme génétique standard est efficace à condition que le problème ne soit pas trop complexe. Pour la deuxième approche, nous avons transformé le problème en multi-objectif en utilisant les méthodes d'optimisation multi-objectif NSGA-II et MOPSO. Pour transformer le problème mono-objectif à un problème multi-objectif un deuxième objectif doit être ajouté. Le rôle de ce dernier est de guider le processus de recherche vers la zone optimale. Pour ce faire, nous avons opté pour les formules qui relient Vs avec les paramètres géotechniques. Dans le premier cas, nous avons utilisé la formule qui relie Vs avec le nombre de coups (N-SPT) et la formule reliant Vs avec la contrainte verticale effective (σ'_v) pour le deuxième cas. Les résultats obtenus étaient très acceptables notamment ceux de NSGA-II avec des erreurs de ($R^2=0,9019$ et MAPE= 5,95 %) pour le premier cas et ($R^2=0.738$ et MAPE =7.885 %) pour le deuxième cas. La méthode MOPSO a également donné de bons résultats avec des erreurs de ($R^2=0.783$ et MAPE= 6,472 %) pour le premier cas et ($R^2 =0.7822$ et MAPE=8.326 %) pour le deuxième cas.

Les résultats obtenus par les deux méthodes prouvent l'efficacité et la capacité de ces méthodes pour l'optimisation du paramètre dynamique Vs.

Conclusion générale et perspectives

1. Conclusion générale

Partout et dans toutes les disciplines les problèmes sont généralement multi-objectif. En génie civil, l'ingénieur est obligé de développer des conceptions qui répondent à deux objectifs simultanément : Elles doivent être sécurisées et économiques. Ces conceptions sont influencées par plusieurs facteurs (techniques, économiques, environnementaux, géométriques, et géotechnique, ..., etc.). Le manque des informations précises telles que les données géotechniques est un problème parmi d'autres qui contraint le géotechnicien à faire sa tâche convenablement (évaluation du tassement des fondations, la capacité portante et l'estimation du facteur de sécurité vis-à-vis de la liquéfaction, ..., etc.). Tels problèmes peuvent souvent être considéré comme des problèmes d'optimisation dans lesquels, on cherche à minimiser (ou maximiser) une fonction ou plusieurs fonctions par rapport à l'ensemble des paramètres manquants.

Dans cette thèse, nous avons appliqué des méthodes d'optimisation pour déterminer les valeurs de la vitesse de cisaillement à partir d'un tassement post-liquéfaction. Pour ce faire, nous avons couplé la méthode simplifiée basée sur les mesures de V_s avec des méthodes d'optimisation. D'abord, nous avons commencé par l'algorithme génétique standard qui est une méthode d'optimisation mono-objectif.

D'après les résultats obtenus, l'algorithme génétique n'a pas réussi de trouver les bonnes solutions. Cela est dû à la complexité du problème qui peuvent être résumé dans les points suivants : Le nombre de variable élevé, le profil du sol stratifié, une seule donnée pour tous les variable (tassement). C'est pourquoi, nous avons opté pour deux approches : Soit nous ajoutons des données au problème mono-objectif ou bien le traiter comme un problème multi-objectif en ajoutant un deuxième objectif.

Pour la première approche, nous avons utilisé l'algorithme génétique en modifiant le tassement global observé par le tassement dans chaque point. Les résultats obtenus par cette approche étaient très satisfaisants avec des erreurs très acceptables ($MAPE= 4.651\%$ et $R^2= 0.886$). Ce qui prouve que l'algorithme génétique standard est efficace à condition que le problème ne soit pas trop complexe. Pour la deuxième approche, nous avons transformé le problème en multi-objectif en utilisant les méthodes d'optimisation multi-objectif NSGA-II et MOPSO. Pour transformer le problème mono-objectif à un problème multi-objectif un deuxième objectif doit être ajouté. Le rôle de ce dernier est

de guider le processus de recherche vers la zone optimale. Pour ce faire, nous avons opté pour les formules qui relient V_s avec les paramètres géotechniques. Nous avons utilisé la formule qui relie V_s avec le nombre de coups (N-SPT) pour le premier cas et la formule reliant V_s avec la contrainte verticale effective (σ'_v) pour le deuxième cas. Les résultats obtenus étaient très acceptables notamment ceux de NSGA-II avec des erreurs de ($R^2=0,9019$ et $MAPE= 5,95 \%$) pour le premier cas et ($R^2=0.738$ et $MAPE =7.885 \%$) pour le deuxième cas. La méthode MOPSO a également donné de bons résultats avec des erreurs de ($R^2=0.783$ et $MAPE= 6,472 \%$) pour le premier cas et ($R^2 =0.7822$ et $MAPE=8.326 \%$) pour le deuxième cas.

Les résultats obtenus par les deux méthodes prouvent l'efficacité et la capacité de ces méthodes pour l'optimisation du paramètre dynamique V_s .

2. Perspectives

Les résultats obtenus prouvent la performance des deux méthodes dans l'optimisation, ce qui rend leurs utilisations dans le domaine géotechnique très utiles. Dans cette thèse, NSGA-II et MOPSO ont été utilisées pour optimiser seulement le paramètre dynamique V_s , mais elles peuvent aussi optimiser facilement d'autres paramètres géotechniques :

- Tassement : post-liquéfaction, fondation (superficielle et profonde), remblai, ..., etc.
- Capacité portante des fondations profondes et superficielles, etc.
- Interaction sol-structure : Impédances dynamiques, conception optimales des dispositifs TDM (Tuned Damper Mass), ...etc.

Bibliographie

- Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, 16(2), 126-142.
- Alavi, A. H., & Gandomi, A. H. (2012). Energy-based numerical models for assessment of soil liquefaction. *Geoscience Frontiers*, 3(4), 541-555.
- Alkroosh, I., & Nikraz, H. (2011). Simulating pile load-settlement behavior from CPT data using intelligent computing. *Central European Journal of Engineering*, 1(3), 295-305.
- Amini, F., Hazaveh, N. K., & Rad, A. A. (2013). Wavelet PSO-Based LQR Algorithm for Optimal Structural Control Using Active Tuned Mass Dampers. *Computer-Aided Civil and Infrastructure Engineering*, 28(7), 542-557.
- Andrab, S. G., Hekmat, A., & Yusop, Z. B. (2017). A Review: Evolutionary Computations (GA and PSO) in Geotechnical Engineering. *Computational Water, Energy, and Environmental Engineering*, 6(02), 154.
- Andrab, S. G., Hekmat, A., & Yusop, Z. B. (2017). A Review: Evolutionary Computations (GA and PSO) in Geotechnical Engineering. *Computational Water, Energy, and Environmental Engineering*, 6(02), 154.
- Andrus, R. D., & Stokoe II, K. H. (2000). Liquefaction resistance of soils from shear-wave velocity. *Journal of geotechnical and geoenvironmental engineering*, 126(11), 1015-1025.
- Andrus, R. D., Piratheepan, P., Ellis, B. S., Zhang, J., & Juang, C. H. (2004). Comparing liquefaction evaluation methods using penetration-V S relationships. *Soil dynamics and earthquake engineering*, 24(9), 713-721.
- Andrus, R.D. & Stokoe II, K.H. (1997). Liquefaction resistance based on shear wave velocity. Proceeding of NCEER Workshop on Evaluation of Liquefaction Resistance of Soils, (pp. 89-128). Buffalo, NY: T.Y. Youd and I.M. Idriss, Eds
- Ardalan, H., Eslami, A., & Nariman-Zadeh, N. (2009). Piles shaft capacity from CPT and CPTu data by polynomial neural networks and genetic algorithms. *Computers and Geotechnics*, 36(4), 616-625.
- Badreddine, S., & Goudjil, K. (2012). Prediction of dynamic impedances functions using an Artificial Neural Network (ANN). In-*Applied Mechanics and Materials* (Vol. 170, pp. 3588-3593). Trans Tech Publications.
- Banimahd, M., Yasrobi, S. S., & Woodward, P. K. (2005). Artificial neural network for stress–strain behavior of sandy soils: Knowledge based verification. *Computers and Geotechnics*, 32(5), 377-386.
- Baziar, M. H., & Ghorbani, A. (2005). Evaluation of lateral spreading using artificial neural networks. *Soil Dynamics and Earthquake Engineering*, 25(1), 1-9.
- Boulanger, R. W., & Idriss, I. M. (2004). *Evaluating the potential for liquefaction or cyclic failure of silts and clays* (p. 131). Center for Geotechnical Modeling
- Boumédiène, D. & Abdelmalek, B. (2010). Estimation de la vitesse maximale du sol par la méthode des réseaux de neurones artificiels.
- Bowles, J.E. (1984). Physical and geotechnical properties of soils (2e edition). New York, NY: McGraw-Hill.
- Çelik, S., & Tan, Ö. (2005). Determination of preconsolidation pressure with artificial neural network. *Civil Engineering and Environmental Systems*, 22(4), 217-231.

- Chen, Y. R., Hsieh, S. C., & Liu, C. H. (2010). Simulation of stress-strain behavior of saturated sand in undrained triaxial tests based on genetic adaptive neural networks. *Electronic Journal of Geotechnical Engineering*, 15(Q), 1815-1834.
- Cheng, Y. M. (2003). Location of critical failure surface and some further studies on slope stability analysis. *Computers and geotechnics*, 30(3), 255-267.
- Cheng, Y. M., Li, L., Chi, S. C., & Wei, W. B. (2007). Particle swarm optimization algorithm for the location of the critical non-circular failure surface in two-dimensional slope stability analysis. *Computers and Geotechnics*, 34(2), 92-103.
- Chenghua, W., Xuyong, X., & Guangxin, L. (2003). Ant algorithm in search of the critical slip surface in soil slopes based on stress fields. *Chinese Journal of Rock Mechanics and Engineering*, 22(5), 813-819.
- Chern, S. G., & Lee, C. Y. (2009). CPT-based simplified liquefaction assessment by using fuzzy-neural network. *Journal of Marine Science and Technology*, 17(4), 326-331.
- Cho, S. E. (2009). Probabilistic stability analyses of slopes using the ANN-based response surface. *Computers and Geotechnics*, 36(5), 787-797.
- Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5). New York: Springer.
- Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3), 256-279.
- COELLO, Carlos A. Coello, PULIDO, Gregorio Toscano, et LECHUGA, M. Salazar. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 2004, vol. 8, no 3, p. 256-279.
- Costa, A. L., Cunha, M. D. C., Coelho, P. A., & Einstein, H. H. (2013). Solving high-speed rail planning with the simulated annealing algorithm. *Journal of Transportation Engineering*, 139(6), 635-642.
- Das, S. K., & Basudhar, P. K. (2006). Undrained lateral load capacity of piles in clay using artificial neural network. *Computers and Geotechnics*, 33(8), 454-459.
- Das, S. K., Biswal, R. K., Sivakugan, N., & Das, B. (2011). Classification of slopes and prediction of factor of safety using differential evolution neural networks. *Environmental Earth Sciences*, 64(1), 201-210.
- Das, S. K., Samui, P., & Sabat, A. K. (2011). Application of artificial intelligence to maximum dry density and unconfined compressive strength of cement stabilized soil. *Geotechnical and Geological Engineering*, 29(3), 329-342.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons.
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000, September). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature* (pp. 849-858). Springer, Berlin, Heidelberg.
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000, September). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International*

- Conference on Parallel Problem Solving From Nature* (pp. 849-858). Springer, Berlin, Heidelberg.
- Dias, J. C., Machado, P., Silva, D. C., & Abreu, P. H. (2014). An inverted ant colony optimization approach to traffic. *Engineering Applications of Artificial Intelligence*, 36, 122-133.
- Dobry R., Ladd R. S., Yokel F. Y., Chung R. M., Powell D., Prediction of pore water pressure buildup and liquefaction of sands during earthquakes by the cyclic strain method, Building science series 138, National Bureau of Standards, 1982.
- Dobry, R., Stokoe, K. H., Ladd, R. S., & Youd, T. L. (1981, October). Liquefaction susceptibility from S-wave velocity. In *Proc., ASCE Nat. Convention, In Situ Tests to Evaluate Liquefaction Susceptibility*, ASCE, New York.
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28-39.
- Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.
- Farshidianfar, A., & Soheili, S. (2013). Ant colony optimization of tuned mass dampers for earthquake oscillations of high-rise structures including soil–structure interaction. *Soil Dynamics and Earthquake Engineering*, 51, 14-22.
- Farshidianfar, A., & Soheili, S. (2013). Ant colony optimization of tuned mass dampers for earthquake oscillations of high-rise structures including soil–structure interaction. *Soil Dynamics and Earthquake Engineering*, 51, 14-22.
- Ferentinou, M. D., & Sakellariou, M. G. (2007). Computational intelligence tools for the prediction of slope performance. *Computers and Geotechnics*, 34(5), 362-384.
- Fieldsend, J. E., & Singh, S. (2002). A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence.
- Fishburn, P. C. (1974). Exceptional paper—Lexicographic orders, utilities and decision rules: A survey. *Management science*, 20(11), 1442-1471
- Fletcher, R. (1987). *Practical methods of optimization* John Wiley & sons. New York, 80
- Fonseca, C. M., & Fleming, P. J. (1993, June). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In *Icga* (Vol. 93, No. July, pp. 416-423).
- Fu, Q., Hashash, Y. M., Jung, S., & Ghaboussi, J. (2007). Integration of laboratory testing and constitutive modeling of soils. *Computers and Geotechnics*, 34(5), 330-345.
- Garaga, A., & Latha, G. M. (2010). Intelligent prediction of the stress–strain response of intact and jointed rocks. *Computers and Geotechnics*, 37(5), 629-637.
- Gholizadeh, S., & Fattahi, F. (2014). Design optimization of tall steel buildings by a modified particle swarm algorithm. *The Structural Design of Tall and Special Buildings*, 23(4), 285-301.
- Gill, M. K., Kaheil, Y. H., Khalil, A., McKee, M., & Bastidas, L. (2006). Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research*, 42(7).
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5), 533-549.
- Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, 1(3), 190-206.
- Glover, F. (1990). Tabu search—part II. *ORSA Journal on computing*, 2(1), 4-32.

- GOH, A. T. (2000). Search for critical slip circle using genetic algorithms. *Civil Engineering Systems*, 17(3), 181-211.
- Goh, A. T., Kulhawy, F. H., & Chua, C. G. (2005). Bayesian neural network analysis of undrained side resistance of drilled shafts. *Journal of Geotechnical and Geoenvironmental Engineering*, 131(1), 84-93.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning.
- Goldberg, D. E., & Samtani, M. P. (1986, February). Engineering optimization via genetic algorithm. In *Electronic computation* (pp. 471-482). ASCE.
- Gordan, B., Armaghani, D. J., Hajihassani, M., & Monjezi, M. (2016). Prediction of seismic slope stability through combination of particle swarm optimization and neural network. *Engineering with Computers*, 32(1), 85-97.
- Goudjil, K., & Sbartai, B. (2017). Optimization of Shear Wave Velocity (Vs) From a Post-Liquefaction Settlement Using a Genetic Algorithm Multi-Objective NSGA II.
- Grasso, S., Maugeri, M., & Monaco, P. (2006, April). Using KD and Vs from seismic dilatometer (SDMT) for evaluating soil liquefaction. In *Proc. of the Second International Conference on the Flat Dilatometer, Washington* (pp. 2-5).
- Hanna, A. M., Ural, D., & Saygili, G. (2007). Evaluation of liquefaction potential of soil deposits using artificial neural networks. *Engineering Computations*, 24(1), 5-16.
- Hardin, B. O., & Drnevich, V. P. (1972). Shear modulus and damping in soils: design equations and curves. *Journal of Soil Mechanics & Foundations Div*, 98(sm7).
- Hasancebi, N., & Ulusay, R. (2007). Empirical correlations between shear wave velocity and penetration resistance for ground shaking assessments. *Bulletin of Engineering Geology and the Environment*, 66(2), 203-213.
- Hegazy, Y. A., & Mayne, P. W. (1995, October). Statistical correlations between VS and cone penetration data for different soil types. In *Proceedings of the International Symposium on Cone Penetration Testing, CPT* (Vol. 95, pp. 173-178).
- Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press. *Ann Arbor*, 1(975), 1.
- Holtz, R.D. & Kovacs, W.D. (1991). Introduction à la géotechnique (édition en français traduite par Jean Lafleur). Montréal, Qc : Éditions de l'École Polytechnique de Montréal.
- Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference on* (pp. 82-87). IEEE.
- Hu, X., & Eberhart, R. (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* (Vol. 2, pp. 1677-1681). IEEE.
- Hu, X., & Eberhart, R. (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* (Vol. 2, pp. 1677-1681). IEEE.
- Hunt, R.E. (1986). Geotechnical Engineering Analysis and Evaluation. New York, NY: McGraw-Hill.
- Idriss, I. M. (1999). An update to the Seed-Idriss simplified procedure for evaluating liquefaction potential. *Proc., TRB Workshop on New Approaches to Liquefaction, Publ. n. FHWA-RD-99-165, Federal Highway Administration.*

- Idriss, I. M., & Boulanger, R. W. (2003, September). Relating K_α and K_σ to SPT blow count and to CPT tip resistance for use in evaluating liquefaction potential. In *Proc. of the 2003 Dam Safety Conference* (pp. 7-10).
- Idriss, I. M., & Boulanger, R. W. (2006). Semi-empirical procedures for evaluating liquefaction potential during earthquakes. *Soil Dynamics and Earthquake Engineering*, 26(2), 115-130.
- Idriss, I. M., & Boulanger, R. W. (2008). *Soil liquefaction during earthquakes*. Earthquake Engineering Research Institute.
- Ishihara, K., & Yoshimine, M. (1992). Evaluation of settlements in sand deposits following liquefaction during earthquakes. *Soils and foundations*, 32(1), 173-188.
- Ismail, A., & Jeng, D. S. (2011). Modelling load–settlement behaviour of piles using high-order neural network (HON-PILE model). *Engineering Applications of Artificial Intelligence*, 24(5), 813-821.
- Javadi, A. A., Rezaia, M., & Nezhad, M. M. (2006). Evaluation of liquefaction induced lateral displacements using genetic programming. *Computers and Geotechnics*, 33(4), 222-233.
- Jianping, J., Changhong, Y., & Guangyun, G. (2011). Prediction on compressibility
- Johari, A., Javadi, A. A., & Habibagahi, G. (2011). Modelling the mechanical behaviour of unsaturated soils using a genetic algorithm-based neural network. *Computers and Geotechnics*, 38(1), 2-13.
- Juang, C. H., Yuan, H., Lee, D. H., & Lin, P. S. (2003). Simplified cone penetration test-based method for evaluating liquefaction resistance of soils. *Journal of geotechnical and geoenvironmental engineering*, 129(1), 66-80.
- Junghans, L., & Darde, N. (2015). Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization. *Energy and Buildings*, 86, 651-662.
- Kahatadeniya, K. S., Nanakorn, P., & Neaupane, K. M. (2009). Determination of the critical failure surface for slope stability analysis using ant colony optimization. *Engineering Geology*, 108(1), 133-141.
- Kaveh, A. (2014). Ray Optimization Algorithm. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures* (pp. 233-276). Springer International Publishing.
- Kayen, R. E., & Mitchell, J. K. (1997). Assessment of liquefaction potential during earthquakes by Arias intensity. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(12), 1162-1174.
- Kennedy, R. (1995, November). J. and Eberhart, Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks IV*, pages (Vol. 1000).
- Khajehzadeh, M., Taha, M. R., El-Shafie, A., & Eslami, M. (2011). Modified particle swarm optimization for optimum design of spread footing and retaining wall. *Journal of Zhejiang university-science a*, 12(6), 415-427.
- Khoshnoudian, F., Ayyobi, P., Paytam, F., & Mirmo'azen, S. M. (2015). Genetic algorithm design of a soil–structure–damper system. *Proceedings of the Institution of Civil Engineers-Structures and Buildings*, 168(8), 593-607.
- Khoshnoudian, F., Ayyobi, P., Paytam, F., & Mirmo'azen, S. M. (2015). Genetic algorithm design of a soil–structure–damper system. *Proceedings of the Institution of Civil Engineers-Structures and Buildings*, 168(8), 593-607.

- Knabe, T., Datcheva, M., Lahmer, T., Cotecchia, F., & Schanz, T. (2013). Identification of constitutive parameters of soil using an optimization strategy and statistical analysis. *Computers and Geotechnics*, 49, 143-157.
- Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, 8(2), 149-172.
- Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, 8(2), 149-172.
- Koester, J.P. (1994). The influence of fines type and content on cyclic strength. Ground failures under seismic conditions, ASCE Geotechnical Special Publication No.44, 17-33.
- Kramer, S. L., & Seed, H. B. (1988). Initiation of soil liquefaction under static loading conditions. *Journal of Geotechnical Engineering*, 114(4), 412-430.
- Ku, T., Subramanian, S., Moon, S. W., & Jung, J. (2016). Stress dependency of shear-wave velocity measurements in soils. *Journal of Geotechnical and Geoenvironmental Engineering*, 143(2), 04016092.
- Kung, G. T., Hsiao, E. C., Schuster, M., & Juang, C. H. (2007). A neural network approach to estimating deflection of diaphragm walls caused by excavation in clays. *Computers and Geotechnics*, 34(5), 385-396.
- Lee, K.L. (1965). Triaxial compressive strength of saturated sand under seismic loading conditions. Thèse de Ph.D., University of California, Berkeley, USA. Lee.
- Lee, T. L., Lin, H. M., Jeng, D. S., & Lu, Y. P. (2008). Back-propagation neural network for assessment of high way slope failure in Taiwan. *Geotech.Eng.*39, 121_128.
- Leung, A. Y. T., & Zhang, H. (2009). Particle swarm optimization of tuned mass dampers. *Engineering Structures*, 31(3), 715-728.
- Leung, A. Y. T., & Zhang, H. (2009). Particle swarm optimization of tuned mass dampers. *Engineering Structures*, 31(3), 715-728.
- Levasseur, S., Malécot, Y., Boulon, M., & Flavigny, E. (2008). Soil parameter identification using a genetic algorithm. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(2), 189-213.
- Levasseur, S., Malécot, Y., Boulon, M., & Flavigny, E. (2008). Soil parameter identification using a genetic algorithm. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(2), 189-213.
- Li, J. P. (2015). Truss topology optimization using an improved species-conserving genetic algorithm. *Engineering Optimization*, 47(1), 107-128.
- Li, L., Chi, S. C., & Lin, G. (2004). The complex method based on ant colony algorithm and its application to the slope stability analysis. *CHINESE JOURNAL OF GEOTECHNICAL ENGINEERING-CHINESE EDITION*-, 26, 691-696.
- Marano, G. C., Quaranta, G., & Monti, G. (2011). Modified genetic algorithm for the dynamic identification of structural systems using incomplete measurements. *Computer-Aided Civil and Infrastructure Engineering*, 26(2), 92-110.
- McCombie, P., & Wilkinson, P. (2002). The use of the simple genetic algorithm in finding the critical factor of safety in slope stability analysis. *Computers and Geotechnics*, 29(8), 699-714.
- Mendjel, D., & Messast, S. (2009). Identification des paramètres hydromécaniques d'un sol non saturé par analyse inverse.

- Mitchell, J.K. & Soga, K. (2005). *Fundamentals of soil behavior* (3e édition). New York, NY: John Wiley Interscience.
- Mohebbi, M., Shakeri, K., Ghanbarpour, Y., & Majzoub, H. (2013). Designing optimal multiple tuned mass dampers using genetic algorithms (GAs) for mitigating the seismic response of structures. *Journal of Vibration and Control*, 19(4), 605-625.
- Monaco, P., & Marchetti, S. (2007, June). Evaluating liquefaction potential by seismic dilatometer (SDMT) accounting for aging/stress history. In *Proc. 4th Int. Conf. on Earthquake Geotechnical Engineering*.
- Monaco, P., Marchetti, S., Totani, G., & Calabrese, M. (2005, September). Sand liquefiability assessment by flat dilatometer test (DMT). In *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SOIL MECHANICS AND GEOTECHNICAL ENGINEERING* (Vol. 16, No. 4, p. 2693). AA BALKEMA PUBLISHERS.
- Montalvo, I., Izquierdo, J., Pérez-García, R., & Herrera, M. (2014). Water Distribution System Computer-Aided Design by Agent Swarm Optimization. *Computer-Aided Civil and Infrastructure Engineering*, 29(6), 433-448.
- Mostaghim, S., & Teich, J. (2003, April). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE* (pp. 26-33). IEEE.
- Najjar, Y. M., & Huang, C. (2007). Simulating the stress–strain behavior of Georgia kaolin via recurrent neuronal approach. *Computers and Geotechnics*, 34(5), 346-361.
- Nejad, F. P., Jaksa, M. B., Kakhi, M., & McCabe, B. A. (2009). Prediction of pile settlement using artificial neural networks based on standard penetration test data. *Computers and Geotechnics*, 36(7), 1125-1133.
- Olsen, R. S. (1997). “Cyclic liquefaction based on the cone penetration test.” Proc., NCEER Workshop on Evaluation of Liquefaction Resistance of Soils, Nat. Ctr. for Earthquake Engrg. Res., State Univ. of New York at Buffalo, 225–276.
- Pal, M., & Deswal, S. (2008). Modeling pile capacity using support vector machines and generalized regression neural network. *Journal of geotechnical and geoenvironmental engineering*, 134(7), 1021-1024.
- Papon, A. (2010). *Modélisation numérique du comportement des sols sous très grands nombres de cycles : homogénéisation temporelle et identification des paramètres* (Doctoral dissertation, Université de Nantes ; Ecole Centrale de Nantes).
- Park, H. I., & Kim, Y. T. (2011). Prediction of strength of reinforced lightweight soil using an artificial neural network. *Engineering Computations*, 28(5), 600-615.
- Park, H. I., & Lee, S. R. (2011). Evaluation of the compression index of soils using an artificial neural network. *Computers and Geotechnics*, 38(4), 472-481.
- Parsopoulos, K. E., & Vrahatis, M. N. (2002, March). Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM symposium on Applied computing* (pp. 603-607). ACM.
- Pecker, A. (1984). *Dynamique des sols*. Presses de l'école nationale des ponts et chaussées.
- Polito, C.P. (1999). The effects of nonplastic and plastic fines on the liquefaction of sandy soils. Thèse de Ph.D., Virginia Polytechnic institute and state university, Blacksburg, Virginia, USA.
- Pourzeynali, S., Lavasani, H. H., & Modarayi, A. H. (2007). Active control of high-rise building structures using fuzzy logic and genetic algorithms. *Engineering Structures*, 29(3), 346-357.

- Ray, T., & Liew, K. M. (2002). A swarm metaphor for multiobjective design optimization. *Engineering optimization*, 34(2), 141-153.
- Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3), 287-308.
- Robertson, P. K., & Campanella, R. G. (1985). Liquefaction potential of sands using the CPT. *Journal of Geotechnical Engineering*, 111(3), 384-403.
- Robertson, P. K., and Wride, C. E. (1998). "Evaluating cyclic liquefaction potential using the cone penetration test." *Can. Geotech. J.*, Ottawa, 35(3), 442-459.
- Robertson, P. K., Woeller, D. J., & Finn, W. D. L. (1992). Seismic cone penetration test for evaluating liquefaction potential under cyclic loading. *Canadian Geotechnical Journal*, 29(4), 686-695.
- Robertson, P.K., and Wride (Fear), C.E. 1997. Evaluation of cyclic liquefaction potential based on the CPT. In *Proceedings of the Discussion Special Technical Session on Earthquake Geotechnical Engineering*, 14th International Conference on Soil Mechanics and Foundation Engineering, Hamburg. Edited by P.S. Sêco e Pinto. A.A. Balkema. Rotterdam, the Netherlands. pp. 269-277.
- Sahab, M. G., Toropov, V. V., & Gandomi, A. H. (2013). A review on traditional and modern structural optimization: problems and techniques. *Metaheuristic Applications in Structures and Infrastructures*, 25-47.
- Sahab, M. G., Toropov, V. V., & Gandomi, A. H. (2013). A review on traditional and modern structural optimization: problems and techniques. *Metaheuristic Applications in Structures and Infrastructures*, 25-47.
- Samui, P., & Kumar, B. (2006). Artificial neural network prediction of stability numbers for two-layered slopes with associated flow rule. *The Electronic Journal of Geotechnical Engineering*.
- Samui, P., & Sitharam, T. G. (2011). Determination of liquefaction susceptibility of soil based on field test and artificial intelligence. *International Journal of Earth Science and Engineering*, 4(2), 216-222.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithm. In *Proceeding of the First International Conference of Genetic Algorithms and Their Application* (pp. 93-100).
- Seed, H. B., Tokimatsu, K., Harder, L. F., and Chung, R. M. (1985). The influence of SPT procedures in soil liquefaction resistance evaluations. *J. Geotech. Engrg. ASCE*, 111(12), 1425-1445.
- Seed, H.B. & Idriss I.M. (1971). Simplified procedure for evaluating soil liquefaction potential. *Journal Geotechnical Engineering*, 97(9), 1249-1273.
- Seed, H.B. & Idriss I.M. (1982). *Ground motions and soil liquefaction during earthquakes*. Berkeley, CA: Earthquake Engineering Research Institute.
- Seed, H.B. & Lee, K.L. (1966). Liquefaction of saturated sands during cyclic loading. *Journal of the Soil Mechanics and Foundations Division (ASCE)*, 92(6), 105-134.
- Sgambi, L., Gkoumas, K., & Bontempi, F. (2012). Genetic Algorithms for the Dependability Assurance in the Design of a Long-Span Suspension Bridge. *Computer-Aided Civil and Infrastructure Engineering*, 27(9), 655-675.
- Shahin, M. A. (2010). Intelligent computing for modeling axial capacity of pile foundations. *Canadian Geotechnical Journal*, 47(2), 230-243.
- Shahoo Maleki, Ali Moradzadeh, Reza Ghavami Riabi & Raoof Gholami (2014). Prediction of shear wave velocity using empirical correlations and artificial intelligence methods, *NRIAG Journal of Astronomy and Geophysics*, 3, 70-81.

- Shouju, L., Yingxi, L., Xiang, H., & Yujing, L. (2003). Global search algorithm of minimum safety factor for slope stability analysis based on annealing simulation [J]. *Chinese Journal of rock mechanics and engineering*, 2, 016.
- Soroush, A., Foroozan, R., Asadollahi, P., & Engineers, M. S. M. C. (2006). Simulation of 3D Effect of Excavation Face Advancement Using an ANN Trained by Numerical Models. *Electron. J. Geotech. Eng.*, 11.
- Srinivas, N., & Deb, K. (1994). Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3), 221-248.
- Sun, J., Li, J., & Liu, Q. (2008). Search for critical slip surface in slope stability analysis by spline-based GA method. *Journal of geotechnical and geoenvironmental engineering*, 134(2), 252-256.
- Sykora, D. W. (1987). Creation of a database of seismic shear wave velocities for correlation analysis. Geotech. Lab. Misc. Paper GL-87- 26, U.S. Army Engr. Waterways Experiment Station, Vicksburg, Miss.
- Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.
- Tokimatsu, K., & Seed, H. B. (1987). Evaluation of settlements in sands due to earthquake shaking. *Journal of Geotechnical Engineering*, 113(8), 861-878.
- Tsai, P. H., Lee, D. H., Kung, G. T. C., & Juang, C. H. (2009). Simplified DMT-based methods for evaluating liquefaction resistance of soils. *Engineering Geology*, 103(1), 13-22.
- Viana, F. A. C., Kotinda, G. I., Rade, D. A., & Steffen, V. (2008). Tuning dynamic vibration absorbers by using ant colony optimization. *Computers & Structures*, 86(13), 1539-1549.
- Wang, H. B., & Sassa, K. (2006). Rainfall-induced landslide hazard assessment using artificial neural networks. *Earth Surface Processes and Landforms*, 31(2), 235-247.
- Wang, W. (1979). Some findings in soil liquefaction. Report Water Conservancy and Hydro- Electric Power Scientific Research Institute, Pékin, Chine (pp. 1-17).
- Wei, Y., & Qiqiang, L. (2004). Survey on Particle Swarm Optimization Algorithm [J]. *Engineering Science*, 5(5), 87-94.
- Yang, X. S., Gandomi, A. H., Talatahari, S., & Alavi, A. H. (Eds.). (2012). *Metaheuristics in water, geotechnical and transport engineering*. Newnes.
- Yegian, M.K., Eseller-Bayat, E., Alshawabkeh, A. & Ali, S. (2007). Induced-partial saturation for liquefaction mitigation: experimental investigation. *Journal of Geotechnical and Geoenvironmental Engineering* (ASCE), 133(4), 372-380.
- Yi, F. (2010, July). Procedure to evaluate liquefaction-induced settlement based on shear wave velocity. In *Manuscript submitted to The 9th US national and 10th Canadian Conference on Earthquake Engineering (9US/10CCEE), Toronto, Canada*.
- Yi, F. F. (2010) Procedure to evaluate liquefaction-induced settlement based o shear wave velocity.
- Yildiz, E., Ozyazicioglu, M. H., & Ozkan, M. Y. (2010). Lateral pressures on rigid retaining walls: a neural network approach. *Gazi University Journal of Science*, 23(2), 201-210.
- Youd, T. L., & Idriss, I. M. (2001). Liquefaction resistance of soils: summary report from the 1996 NCEER and 1998 NCEER/NSF workshops on evaluation of liquefaction resistance of soils. *Journal of geotechnical and geoenvironmental engineering*, 127(4), 297-313

- Youd, T. L., Kayen, R. E., and Mitchell, J. K. (1997). "Liquefaction criteria based on energy content of seismograms." Proc., NCEER Workshop on Evaluation of Liquefaction Resistance of Soils, Nat. Ctr. For Earthquake Engrg. Res., State Univ. of New York at Buffalo, 217–224.
- Zhang, G., Robertson, P. K., & Brachman, R. W. (2002). Estimating liquefaction-induced ground settlements from CPT for level ground. *Canadian Geotechnical Journal*, 39(5), 1168-1180.
- Zitzler, E., & Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength Pareto approach.
- Zolfaghari, A. R., Heath, A. C., & McCombie, P. F. (2005). Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. *Computers and geotechnics*, 32(3), 139-152.