

N° d'ordre : /2013/DM

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ HADJ LAKHDAR BATNA

FACULTÉ DE TECHNOLOGIE

DÉPARTEMENT DE MÉCANIQUE

Mémoire présenté pour l'obtention du diplôme de

MAGISTÈRE

En

Mécanique

Option : **CONSTRUCTION MECANIQUE**

Par

BETTAYEB SOUAD

**Optimisation des Conditions de Coupe
par la Méthode Colonie de Fourmis**

Soutenu le : / / Devant le jury composé de

D _r MAZOUZ HAMOUDI	Professeur, Université de Batna	Président
D _r ASSAS MEKKI	Professeur, Université de Batna	Rapporteur
D _r MANAA RABAH	M.C.A, Université de Batna	Co- rapporteur
D _r BEN MOHAMMED BRAHIM	Professeur, Université de Batna	Examineur
D _r DJEBAILI HAMID	Professeur, Université de khenchela	Examineur

Année universitaire 2012 / 2013

DEDICACE
ET
REMERCIEMENTS

DEDICACE

Je tien à dédier le fruit de ce modeste travail à ceux qui me sont les plus chers au monde, mon père et ma mère.

Malheureusement je crains d'oublier de citer certaines personnes; j'espère qu'elles ne m'en tiendront pas grief, et je peux leur assurer qu'elles ont une place particulière dans mon cœur.

Je dédie tout particulièrement :

A mes frères.

A mes sœurs.

Ainsi qu'à toute la famille.

A mes amis et spécialement : Wafa, Fadhila, Karima, Radia, Rahwa, Samar, Souad, Yasmina, Souad, Soumia, Amel, Mohammed, Nabil et Abd esatar.

A tous mes collègues de promotion.

Et à tous ceux que j'aime.

Souad

REMERCIEMENTS

Ce travail n'aurait pu aboutir sans l'aide d'Allah qui m'a donné la force de réaliser ce travail.

Mes remerciements vont en tout premier lieu à mon rapporteur de mémoire **Docteur Assas Mekki**, Professeur à l'université de Batna, directeur du laboratoire **L.R.P** Batna, pour tout ce qu'il a fait pour moi.

Mes remerciements vont aussi à mon Co- rapporteur **Docteur Manaa Rabeh**, Professeur à l'université de Batna pour ces judicieux conseils et incessants encouragements.

Je tiens aussi à remercier **Docteur Mazouz Hamoudi**, Professeur à l'université de Batna, directeur de la poste graduation Batna, pour l'intérêt qu'il a bien voulu porter à ce travail ainsi que pour l'honneur qu'il me fait de présider le jury d'examen.

Je remercie également **Docteur Ben Mohamed Brahim**, Professeur à l'université de Batna et **Docteur Djebaili Hamid**, Professeur à l'université de Khenchela, pour le grand honneur qu'ils me font en participant à mon jury.

Ensuite, Je tiens à remercier **Docteur Ameddah hacene**, Docteur à l'université de Batna, que j'exprime ma profonde gratitude pour son aide sans faille, sa disponibilité et sa rigueur lors de la direction de ce travail de mémoire.

Je remercie **Docteur Tibermacine Toufik**, Docteur à l'université de Biskra, **Docteur Chebah Mohammed**, Docteur à l'université de Biskra, **L'enseignant Debbache Mohammed**, Enseignant à l'université de Biskra, **L'enseignant Zalouf Mouloud**, Enseignant à l'université de Djelfa, pour avoir bien voulu se rendre disponibles pour apporter leurs aides ou pour me prodiguer des conseils.

Je remercie **Docteur Brioua Mourad**, Professeur à l'université de Batna et vice recteur de la poste de graduation du département de mécanique Batna.

Je remercie tous les membres du laboratoire **L.R.P** d'université de Batna.

Encore Je remercie tous les membres de laboratoire de génie mécanique d'université de Biskra, le chef de département **Docteur Moummi Noureddine**, Professeur à l'université de Biskra.

Mes remerciements et ma grande gratitude s'adressent particulièrement à **Monsieur Moussa Benaouf**, technicien au laboratoire de corrosion du département de mécanique.

Je remercie **Docteur Naoun Mahieddine**, Docteur à l'université de Batna, pour leur encouragement.

Je tiens aussi à remercier tous ceux qui ont, de près ou de loin, aidé à rendre ce travail possible, que ce soit par des idées ou par des encouragements.

Enfin merci aux personnes qui liront ces quelques pages.

A vous tous, je dis merci.

TABLE DES MATIERES

Table des matières.....	i
Liste des Figures.....	iii
Liste des Tableaux.....	iv
Liste des symboles.....	v
Liste des abréviations.....	viii
Introduction Générale.....	01

CHAPITRE I

ETAT DE L'ART

SUR LES ALGORITHMES EVOLUTIONNAIRES

I.1. Introduction.....	04
I.2. Algorithmes évolutionnaires (EA).....	04
I.2.1. Méthodes Heuristiques.....	05
I.2.2. Méthodes Méta-heuristiques.....	05
I.2.2.1. Algorithmes génétiques (GA).....	06
I.2.2.2. Optimisation par essaim de particules (PSO).....	11
I.3. Etat de l'art sur les algorithmes évolutionnaires.....	14
I.4. Conclusion.....	19

CHAPITRE II

OPTIMISATION PAR COLONIE DE FOURMIS

II.1. Introduction.....	20
II.2. Optimisation par colonies de fourmis.....	20
II.2.1. Comportement des fourmis.....	21
II.2.2. Similarités et différences entre les fourmis virtuelles et réelles.....	24
II.2.3. Expériences.....	26
II.3. Principe de l'algorithme colonie de fourmis.....	29
II.3.1. Le problème du voyageur de commerce (PVC).....	29
II.3.2. Système de fourmis AS (Ant System).....	30
II.3.3. Fonctionnement de l'algorithme (PVC).....	33
II.4. Conclusion.....	35

CHAPITRE III

OPTIMISATION DES CONDITIONS DE COUPE

III.1. Introduction	36
III.2. Problème d'optimisation	36
III.3. Optimisation mono-objective	37
III.4. Critères d'optimisation	37
III.4.1. Temps d'usinage	38
III.5. Contraintes d'optimisation	39
III.6. Conclusion	40

CHAPITRE IV

APPLICATION ET DISCUSSION DES RESULTATS

IV.1. Introduction	41
IV.2. Cas d'application	41
IV.2.1. Algorithme proposé	41
IV.2.2. Algorithme de colonie de fourmis.....	41
IV.2.3. Fonction-objectif.....	48
IV.2.4. Contraintes d'optimisation	48
IV.2.5. Application	48
IV.3. Résultats	50
IV.4. Discussion et interprétation des résultats	52
IV.5. Comparaison	52
Conclusion	54
Recommandations	55
Références bibliographiques	56

LISTE DES FIGURES

ET TABLEAUX

LISTE DES FIGURES

Figure 1	Les techniques d'optimisation conventionnelles et non conventionnelles...	03
Figure I.1	Fonctionnement courant d'un algorithme génétique.....	08
Figure I.2	Exemple du codage binaire d'une solution potentielle avec 3 paramètres...	09
Figure I.3	Modèle de la roulette – wheel	09
Figure I.4	Méthodes de recombinaison (croisement) utilisées par l'AG.....	10
Figure I.5	Méthodes de mutation utilisées par le AG	11
Figure I.6	Schéma de l'évitement d'un prédateur par un banc de poissons.(a) le banc forme un seul groupe, (b) les individus évitent le prédateur en formant une structure en "fontaine", (c) le banc se reforme.....	12
Figure II.1	Dépôt de la phéromone déposée par une fourmi.....	22
Figure II.2	Fourmis suivant une piste de phéromone.....	22
Figure II.3	Auto-organisation dans les systèmes biologiques : (a) Une colonie de fourmis qui ramène de la nourriture vers le nid (b) un essaim d'abeilles (c) un vol groupé d'oiseau (d) une formation de bactéries.....	23
Figure II.4	Pont binaire de Deneubourg.....	26
Figure II.5	Recherche de nourriture par une colonie de fourmis : (a)début de l'expérience, (b) fin de l'expérience.....	28
Figure II.6	Effet de la coupure d'une piste de phéromone.....	29
Figure II.7	Algorithme de Colonies de fourmis de bas : « ant system »	34
Figure II.8	Le problème du voyageur de commerce optimisé par l'algorithme AS.....	34
Figure IV.1	Représentation graphique du processus ACO sous la forme d'un réseau multicouche au début du processus.....	42
Figure IV.2	Représentation graphique du processus ACO sous la forme d'un réseau multicouche à la fin du processus.....	45
Figure IV.3	Organigramme de l'algorithme colonie de fourmis	47
Figure IV.4	Grphe de la fonction objectif en fonction du nombre d'itérations.....	51

LISTE DES TABLEAUX

Tableau I.1	Tableau de récentes techniques utilisant (GA) pour déterminer les paramètres d'usinage optimal.....	17
Tableau IV.1	Paramètres d'usinage.....	49
Tableau IV.2	Paramètres de programme.....	49
Tableau IV.3	Résultats trouvés par le programme ACO.....	50
Tableau IV.4	Comparaison des résultats.....	53

LISTE DES SYMBOLES

ET

ABREVIATIONS

LISTE DES SYMBOLES

Symbole	Désignation	Unité
T_U	Temps de production	min
t_m	Temps machine	min
t_{cs}	Temps de changement d'outil	min
t_R	Temps de retour de l'outil	min/passe
t_h	Temps de montage et démontage de la pièce	min
T	Durée de vie de l'outil	min/arrête
L	Longueur de la pièce à usiner	mm
D	Diamètre extérieur de la pièce	mm
V_c	Vitesse de coupe	m/min
f	Avance par tour	m/tour
d_c	Profondeur de coupe pour une passe	mm
K	Constante de l'équation de la durée de vie	
a_1, a_2, a_3	Constantes	
P_{max}	Puissance maximale pour toutes les opérations	kW
Ra_{max}	Rugosité maximale de surface	μm
SF	Rugosité de la surface finie	μm
SR	Rugosité de surface ébauche	μm
θ_{max}	Température maximale de coupe	$^{\circ}\text{C}$
F_{max}	Force maximale de coupe	N
r_1, r_2	Constantes aléatoires	
w	Constante d'inertie	

c_1, c_2	Constantes d'accélération	
$V_i^{(k)}$	Vitesse actuelle.	m/min
$X_i^{(k)}$	Position actuelle dans l'hyperespace de recherche.	mm
$Pbest_i$	La meilleure position trouvée par la $i^{ème}$ particule.	
$Pbest_g$	La meilleure solution trouvée par l'essaim.	
d_{ij}	La distance entre chaque arc (i, j)	mm
η_{ij}	Visibilité	m^{-1}
p_{ij}^k	Probabilité	
α, β, ς	Paramètres de contrôles	
ρ	Coefficient d'évaporation des phéromones	
Q	Constante	
$b_i(t)$	Nombre de fourmis dans la ville i à l'instant t	
m	Nombres total des fourmis dans la colonie	
$\tau_{ij}(t)$	Quantité de phéromones déposée sur l'arc (i, j) , à l'instant t	
N_i^k	L'ensemble des villes que la fourmi k , placée sur la ville i , n'a pas encore visité	
$T_k(t)$	Le tour réalisé par la $k^{ème}$ fourmi dans l'intervalle de temps $[t, t + n]$	
$L_k(t)$	La longueur réalisée par la $k^{ème}$ fourmi dans l'intervalle de temps $[t, t + n]$	mm
$\Delta\tau_{ij}(t)$	Quantité de phéromones déposée par la $k^{ème}$ fourmi sur l'arc (i, j) dans l'intervalle de temps $[t, t + n]$	
$\Delta\tau_{ij}^k(t)$	Quantité de phéromones déposée par la fourmi k , sur l'arc (i, j) dans l'intervalle de temps $[t, t + n]$	
$\tau_{ij}(0)$	Quantité de phéromones initiale	
$f(x)$	Une fonction- objectif	
$g_i(x), h_j(x)$	Ensemble des contraintes du problème	

it	Nombres d'itérations
it_{max}	Nombres d'itérations maximales
k	Une fourmi
i	Nombres de variables
n	Nombres maximaux de variables
j	Les valeurs de variables
p	Nombres maximaux des valeurs de variables
x_{ij}	Des valeurs
$T_{u\ best}$	La meilleure valeur de la fonction-objectif parmi les solutions prises par les k fourmis
$T_{u\ worst}$	La mauvaise valeur de la fonction-objectif parmi les solutions prises par les k fourmis
$\tau_{ij}^{(old)}$	Quantité de phéromone déposée à l'itération précédente.

LISTE DES ABREVIATIONS

Symboles	Désignations	
	Anglais	Français
ACO	Ant colony optimization	Optimisation par colonies de fourmis
AS	Ant system	Système de fourmis
EA	Evolutionary algorithms	Algorithmes évolutionnaires
GA	Genetic Algorithms	Algorithmes génétiques
GM	Gradient method	Méthode de gradient
PSO	Particle swarm optimization	Optimisation par essaim de particules
SA	Simulated annealing	Recuit simulé
P_{best_i}		La meilleure position rencontrée par chacun des particules
P_{best_g}		La meilleure solution trouvée par l'essaim
CPSO	The cellular particle swarm optimization	Optimisation par essaim de particules cellulaire
SMD		Surface maximale disponible
TS	Tabu search	Recherche tabou
MCACO	Modified continuous ACO	Algorithme de colonies de fourmis modifiées en continu
FEGA	Float encoded GA	Flottant codé GA
HC	Hill climbing)	Méthode des courses de côtes
NM	Newton's method	Méthode de newton
2BP		Bin-Packing en deux dimensions
ANFIS	Adaptive neuro-fuzzy inference system	Système adaptative d'inférence neuro-floue
C.E.U		Cellule élémentaire d'usinage
TSP	Travelling Salesman Problem	Problème du voyageur de commerce (PVC)

INTRODUCTION GENERALE

INTRODUCTION GENERALE

L'obtention des formes requises d'une pièce par enlèvement de matière est la seule méthode qui permet de produire des pièces mécaniques de formes complexes avec des tolérances très serrées. Encore à ce jour le procédé de fabrication est le plus répandu malgré les progrès réalisés par les procédés de mise en forme des matériaux (formage, moulage...) [27].

Le tournage est l'un des processus d'usinage les plus utilisés en fabrication où le besoin industriel est en progression rapide et permanent en termes de qualité technique ou économique. Ce besoin ne peut être satisfait que par un choix judicieux des paramètres de coupe [7].

La sélection des paramètres optimaux de coupe (vitesse de coupe, avance par tour et profondeur de passe) est une question très importante pour tous les processus d'usinage. En effet elle permet d'améliorer certaines performances de coupe (la qualité des produits d'usinage, les coûts d'usinage et le temps de production), en prenant en compte des limites de performances de la machine-outil (effort de coupe, puissance absorbée par la coupe, etc..), de l'outil de coupe utilisé (matière, géométrie, etc..) et de la pièce usinée (matière, dimensions, montage, etc.)

Dans la pratique, en atelier, les paramètres de coupe sont choisis à partir des bases de données d'usinage ou des manuels spécialisés, mais ils ne considèrent pas les aspects économiques de l'usinage. Les conditions de coupe réglées par de telles pratiques sont trop loin d'être optimales. Par ailleurs, les techniques d'optimisation, conventionnelles et non conventionnelles figure (1) [15], ont été utilisées en tant que méthodes pour obtenir des paramètres d'usinage optimisés. Pour l'optimisation d'un critère d'usinage, soit le temps minimal de production ou le taux maximum de profit est utilisé comme fonction-objectif soumis à des contraintes [23].

L'optimisation des paramètres d'usinages se complique quand un grand nombre des contraintes sont incluses, de sorte qu'il est difficile de résoudre ce problème par les méthodes non conventionnelles. Malgré cette difficulté, l'algorithme génétique (GA) [6, 13,14, 15] (qui a été largement appliquée), l'optimisation par essaim de particules (PSO) [7,16] et l'algorithme de colonie de fourmis (ACO) [21- 23], ont pu trouver de solutions a ces problèmes.

Dans notre travail de recherche nous nous intéresserons aux algorithmes colonies de fourmis.

Les algorithmes de colonies de fourmis forment une classe des méthodes méta-heuristiques récemment proposées pour les problèmes d'optimisation des conditions de coupe. Une colonie d'agents simples (Les fourmis) communiquent indirectement via des modifications dynamiques de leur environnement (Les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective [2].

Les fourmis ont la possibilité de déposer la phéromone sur le terrain et de suivre, en probabilité, la phéromone déposée précédemment par d'autres fourmis. En déposant cette substance chimique, les fourmis laissent une trace sur leur passage. En détectant cette trace, les autres fourmis de la colonie peuvent suivre le chemin découvert par d'autres fourmis à trouver de la nourriture. Pour trouver le plus court chemin d'obtenir de la nourriture, ces fourmis peuvent toujours suivre les sentiers de phéromone. De ce comportement de recherche coopérative sur les fourmis réelles s'inspire le nouveau paradigme de calcul pour l'optimisation des systèmes de la vie réelle et il est adapté à la résolution de problème d'optimisation [23].

Le travail réalisé et présenté dans ce mémoire s'articule de la façon suivante :

Le premier chapitre est consacré à une recherche bibliographique sur quelques méthodes méta-heuristiques inspirées de la biologie, avec une synthèse bibliographique sur ces méthodes.

Dans le deuxième chapitre, nous présentons la méthode évolutionnaire appelée optimisation par colonie de fourmis.

Dans le troisième chapitre nous présentons le modèle mathématique utilisé par J.S.Agapiou [12] pour l'optimisation des paramètres d'usinage.

Dans le dernier chapitre nous chercherons à trouver des solutions plus performantes pour le problème d'optimisation des paramètres d'usinage par la méthode colonie de fourmis.

Une comparaison de nos résultats avec ceux trouvés par M .Assas et al [6] en utilisant les algorithmes génétiques, ceux de J.S .Agapiou [12] en utilisant la méthode du gradient et ceux de M .Assas et al [7] en utilisant l'algorithme essaim de particules afin de montrer l'efficacité de la présente méthode.

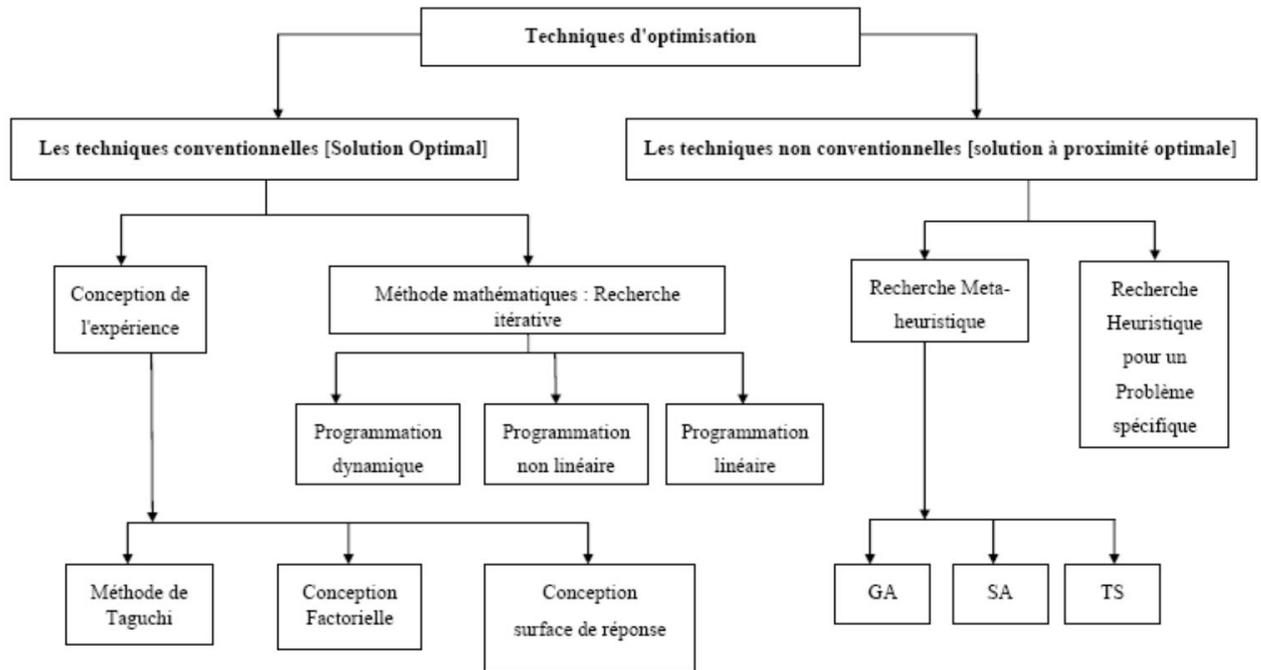


Figure1 - Les techniques d'optimisation conventionnelles et non conventionnelles [15]

CHAPITRE I

ETAT DE L'ART

SUR LES ALGORITHMES EVOLUTIONNAIRES

CHAPITRE I

ETAT DE L'ART

SUR LES ALGORITHMES EVOLUTIONNAIRES

Table des matières

I.1. Introduction.....	04
I.2. Algorithmes évolutionnaires (EA).....	04
I.2.1. Méthodes Heuristiques	05
I.2.2. Méthodes Méta-heuristiques.....	05
I.2.2.1. Algorithmes génétiques (GA).....	06
I.2.2.2. Optimisation par essaim de particules (PSO).....	11
I.3. Etat de l'art sur les algorithmes évolutionnaires	14
I.4. Conclusion.....	19

I.1. Introduction

En recherche opérationnelle, et plus précisément dans le domaine de l'optimisation des problèmes délicats, la majorité des méthodes sont inspirées par de telles études, et notamment par la biologie. Parmi les domaines de la biologie fertiles en inspiration, l'éthologie (étude du comportement des animaux) qui a récemment donné lieu à plusieurs avancées significatives, dont la conception de systèmes de fourmis artificielles. Ces systèmes sont notamment étudiés en robotique, en classification ou encore en optimisation [1,2].

Les études réalisées par les éthologistes ont mis en évidence que certains comportements collectifs des insectes sociaux sont auto-organisés notamment lorsqu'il s'agit de la recherche de nourriture. L'auto-organisation caractérise des processus au cours desquels des structures émergent au niveau collectif, à partir d'une multitude d'interactions simples entre insectes, sans être codées explicitement au niveau individuel [3].

Nous allons présenter dans ce chapitre quelques méthodes méta-heuristiques inspirées de la biologie et utilisant des phénomènes d'auto-organisation. L'algorithme colonie de fourmis, sera décrit au chapitre suivant.

I.2. Algorithmes évolutionnaires

Les Algorithmes évolutionnaires (EA) constituent une discipline impliquant la simulation par un ordinateur du processus de l'évolution naturelle.

Ils sont inspirés de la génétique et des mécanismes de la sélection naturelle basés sur la théorie de l'évolution de Darwin, selon laquelle la vie est une compétition où seuls les mieux adaptés survivent et se reproduisent. Ils empruntent les paradigmes de l'évolution biologique tels que la sélection, le croisement et la mutation pour chercher la solution du problème.

Les EA utilisent la notion de "population d'individus", dans laquelle chaque individu représente une solution potentielle de l'espace de recherche du problème donné. Ce sont des méthodes d'optimisation globales. Leur robustesse et leur souplesse permettent d'aborder les problèmes les plus raides. De plus, leur capacité à travailler sur des espaces de recherche non standards (non continus) ainsi que leur faible besoin d'information sur le problème (seulement la fonction coût) offrent les perspectives les plus originales et un large champ d'application. Ils ont

donc été appliqués avec succès à de nombreux problèmes où les algorithmes classiques d'optimisation sont incapables de produire des résultats satisfaisants.

En outre, en tant qu'algorithme à base de population, leur parallélisation est aisée : il suffit de distribuer l'évaluation de la fonction coût sur autant de processeurs que d'individus de la population. Leur principal inconvénient est leur coût. Ils nécessitent en effet un grand nombre d'évaluations pour aboutir à l'optimum : c'est le prix qu'ils doivent payer au fait de ne pas utiliser d'autre information sur la fonction et de s'appliquer à de très larges classes de problèmes, aussi chaotiques soient ils [4].

Les EA ont été appliqués dans le domaine de la robotique pour l'optimisation des trajectoires ou l'évitement des obstacles, aussi dans divers problèmes tel que la biochimie, l'électrotechnique et la magnétique...

Un certain nombre de travaux ont été effectués sur l'application de ces algorithmes dans le domaine de la mécanique concernant le domaine d'usinage. Nous en citons quelques uns [6, 7, 13, 14, 15, 16, 21, 22, 23].

I.2.1. Méthodes Heuristiques

Une heuristique est une méthode approchée se réclamant simple, rapide et adaptée à un problème particulier. Sa capacité à solutionner un problème difficilement résolvable par des méthodes exactes est nuancée par le fait qu'elle n'offre aucune garantie quant à la qualité de la solution calculée. Ce défaut n'est pas toujours un problème lorsque seule une approximation de la solution optimale est recherchée [3].

I.2.2. Méthodes Méta-heuristiques

Le terme méta-heuristique est utilisé par opposition aux heuristiques particulières pour un problème donné. Les méta-heuristiques peuvent être utilisées pour plusieurs types de problèmes, tandis qu'une heuristique est adaptée à un problème donné.

Les méta-heuristiques ont également comme caractéristiques communes leur caractère stochastique, ainsi que leur inspiration, une analogie avec d'autres domaines de recherche (la biologie, la physique, etc.) [5].

Les méta-heuristiques sont généralement itératives, c'est-à-dire qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation, et directes, c'est-à-dire qu'elles n'utilisent pas l'information du gradient de la fonction-objectif [2].

Il existe un grand nombre de méta-heuristiques, parmi elles : Monte Carlo, recuit simulé, recherche tabou, algorithme génétique, essaim de particule et colonie de fourmis [3].

I.2.2.1. Algorithmes génétiques GA

a. Introduction

Les algorithmes génétiques (GA) sont des méthodes utilisées dans les problèmes d'optimisation. Ils tirent leur nom de l'évolution biologique des êtres vivants dans le monde réel [8]. Le concept d'algorithme génétique a été proposé par [John Holland (1975)] pour décrire les systèmes adaptatifs. Par la suite, l'ouvrage [Goldberg (1989)] a fortement participé à leur diffusion [5]. Ces algorithmes cherchent à simuler le processus de la sélection naturelle et de la génétique dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par C. Darwin. Dans un environnement, « *les individus* » les mieux adaptés tendent à vivre assez longtemps pour se reproduire alors que les plus faibles ont tendance à disparaître [8].

Bien qu'utilisant le hasard, les GA ne sont pas purement aléatoires. Ils exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance [9].

Par analogie avec l'évolution naturelle, les GA font évoluer un ensemble de solutions candidates, appelé une « population d'individus ».

Un « individu » n'est autre qu'une solution possible du problème à résoudre. Chaque individu de cette population se voit attribuer une fonction appelée fonction d'adaptation (*fitness*) qui permet de mesurer sa qualité ou son poids ; cette fonction d'adaptation peut représenter la fonction objectif à optimiser. Ensuite, les meilleurs individus de cette population sont sélectionnés, subissent des croisements et des mutations et une nouvelle population de solutions est produite pour la génération suivante. Ce processus se poursuit, génération après génération, jusqu'à ce que le critère d'arrêt soit atteint, comme par exemple le nombre maximal de générations [8].

Leurs champs d'application sont très vastes. Ils sont utilisés en (mécanique, génie civil, chimie, informatique etc.). Les raisons de ce grand nombre d'applications sont la simplicité et l'efficacité de ces algorithmes [10].

b. Principes et généralités

Les algorithmes génétiques sont généralement utilisés pour rechercher les optima d'un critère d'optimisation défini sur un espace de recherche. Leur mise en œuvre nécessite :

- ❖ Un codage des données qui associe à chaque point de l'espace de recherche une structure de données spécifique, appelée génotype ou ensemble de chromosomes, qui caractérisera chaque individu de la population.
- ❖ La définition d'une fonction d'adaptation à maximiser, définie sur l'espace de recherche. Elle est également appelée fonction d'évaluation ou encore fitness. Elle se définit en fonction du critère d'optimisation du problème : les extrema recherchés du critère correspondent aux maxima de cette fonction.
- ❖ Une population initiale, regroupant un ensemble d'individus, dont dériveront les futures générations.
- ❖ Des opérateurs d'évolution de la population, permettant l'exploration, plus ou moins large, de l'espace de recherche.
- ❖ Un processus de sélection des individus les mieux adaptés, qui sera appliqué sur chaque nouvelle génération d'individus [3].

c. Fonctionnement d'algorithme génétique

Le fonctionnement d'un GA est basé sur les phases suivantes:

- 1. Initialisation** : Une population initiale de N chromosomes est tirée aléatoirement.
- 2. Évaluation** : Chaque chromosome de la population initiale est décodé, puis évalué.
- 3. Sélection** : Création d'une nouvelle population de chromosomes par l'utilisation d'une méthode de sélection appropriée (regroupement des chromosomes par paires).
- 4. Reproduction** : Possibilité de croisement et de mutation au sein de la nouvelle population.

5. Retour à la phase d'évaluation (étape 2) tant que la condition (ou critère d'arrêt) du problème n'est pas satisfaite [9,10].

La figure (I.1) illustre les principales étapes d'un algorithme génétique :

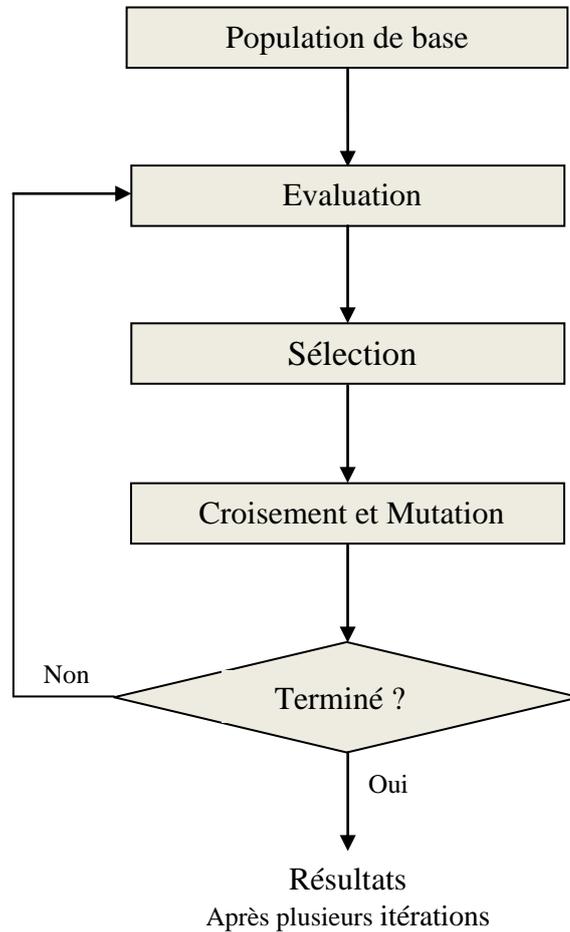


Figure I.1 - Fonctionnement courant d'un algorithme génétique [3]

d. Représentation

Le codage binaire est le code le plus utilisé [8]. Cela permet d'établir une connexion entre les valeurs de la variable et les individus de la population de manière à imiter la connexion qui existe en biologie entre le génotype et le phénotype [10]. Chaque individu est représenté par un vecteur binaire (ou chaîne de bits), dont chaque élément prend la valeur 0 ou 1. Ce vecteur est une concaténation des paramètres à optimiser, chaque paramètre étant transformé en une série binaire. La figure (I.2), présente un exemple du codage binaire d'une solution avec 3 paramètres, chaque paramètre est représenté par une série binaire de 4 chiffres {0,1}[5].

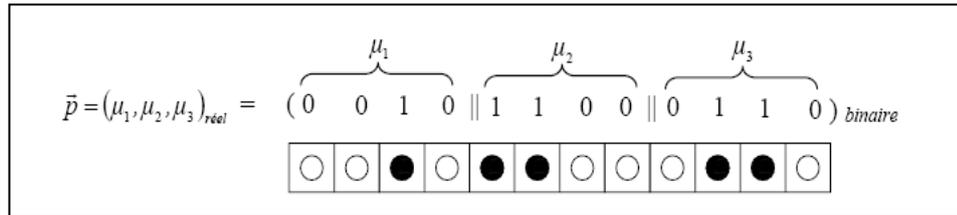


Figure I.2 - Exemple du codage binaire d'une solution potentielle avec 3 paramètres [4]

e. Evolution

L'évolution de la population se fait à l'aide des opérateurs de sélection, la recombinaison (Croisement) et de mutation [7].

❖ Sélection des parents

La sélection est un opérateur clé sur lequel repose en partie la qualité d'un algorithme génétique. Dans cette étape, les chromosomes de la population actuelle sont sélectionnés pour être les parents de la génération suivante. En accord avec la théorie de l'évolution de Darwin, les meilleurs individus doivent survivre et en créer les nouveaux [4]. Les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître [10]. Il existe plusieurs méthodes pour choisir les meilleurs individus, par exemple la sélection proportionnelle, la sélection par tournoi, la sélection par rang, la sélection selon l'état d'équilibre, etc [4]. La méthode la plus connue et la plus utilisée est la sélection par roulette (roulette - wheel) de Goldberg (1989) [10], figure (I.3) représente un modèle de roulette - wheel [7].

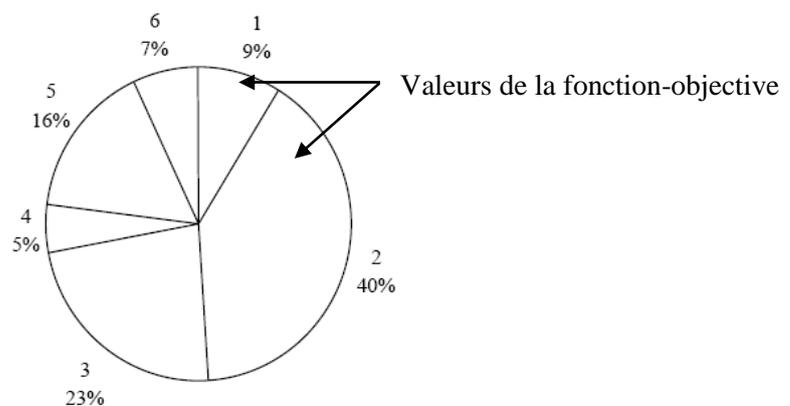


Figure I.3 - Modèle de la roulette -wheel[7]

❖ Croisement

Le croisement est l'opérateur principal des GA. C'est un opérateur génétique relatif à plusieurs individus parents (souvent deux). Son rôle consiste à combiner les génotypes des individus pour en produire un nouveau (engendrer des individus enfants mieux adaptés que leurs parents) figure (I.4) [4]. Les plus souvent, deux enfants sont créés par le croisement de deux parents sélectionnés [3] On distingue plusieurs types de croisements possibles. Les plus utilisés sont :

- Croisement à 1 point.
- Croisement à multipoints.
- Croisement uniforme [4].

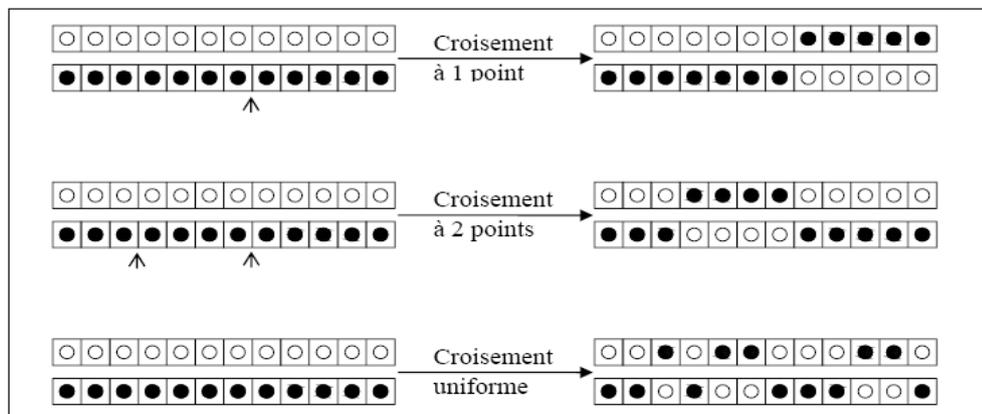


Figure I.4 - Méthodes de recombinaison (croisement) utilisées par l'GA [4]

❖ Mutation

Nous définissons une mutation comme étant l'inversion d'un bit dans un chromosome. Classiquement, l'opérateur de mutation modifie aléatoirement les symboles d'un génotype avec une faible probabilité, égale au taux de mutation. D'autre part, la mutation assure une recherche locale aléatoire autour de chaque individu. Dans cet ordre d'idée, la mutation peut améliorer considérablement la qualité des solutions découvertes [9].

La figure(I.5) montre un exemple de mutation binaire.

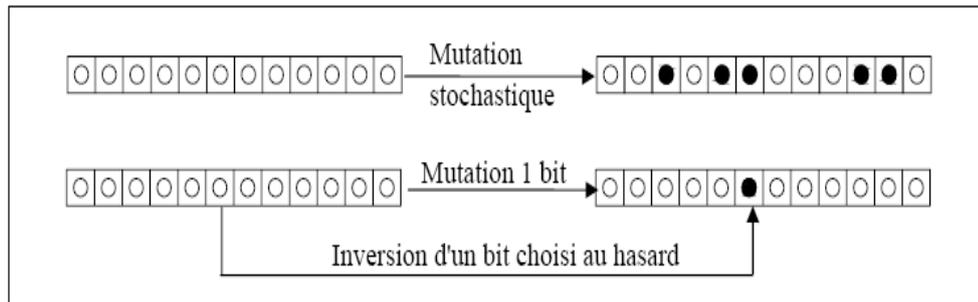


Figure I.5 - Méthodes de mutation utilisées par l'GA[4]

- Mutation stochastique.
- Mutation 1 bit [4].

I.2.2.2. Optimisation par essaim de particules PSO

a. Introduction

L'optimisation par essaim de particules ("Particle Swarm Optimization", PSO) est une méta-heuristique d'optimisation, inventée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995[11].

Heppner, un zoologiste, était intéressé à découvrir les règles fondamentales qui ont permis à un grand nombre d'oiseaux de s'assembler synchroniquement, souvent changeant la direction soudainement, dispersant et regroupant. Ce scientifique a eu l'idée que le modèle de simulation doit être fondé sur la manipulation des distances interindividuelles ; c'est-à-dire, le comportement du groupe a été pensé pour être une fonction des efforts des oiseaux de maintenir une distance optimale entre eux-mêmes et leurs voisins.

Ces algorithmes sont inspirés des essaims d'insectes (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs [7], comme les bancs de poissons, on peut observer des dynamiques de déplacements relativement complexes, alors que les individus eux-mêmes n'ont accès qu'à des informations limitées, comme la position et la vitesse de leurs plus proches voisins. On peut par exemple observer qu'un banc de poissons est capable d'éviter un prédateur : d'abord en se divisant en deux groupes, puis en reformant le banc original figure (I.6), tout en maintenant la cohésion du banc.

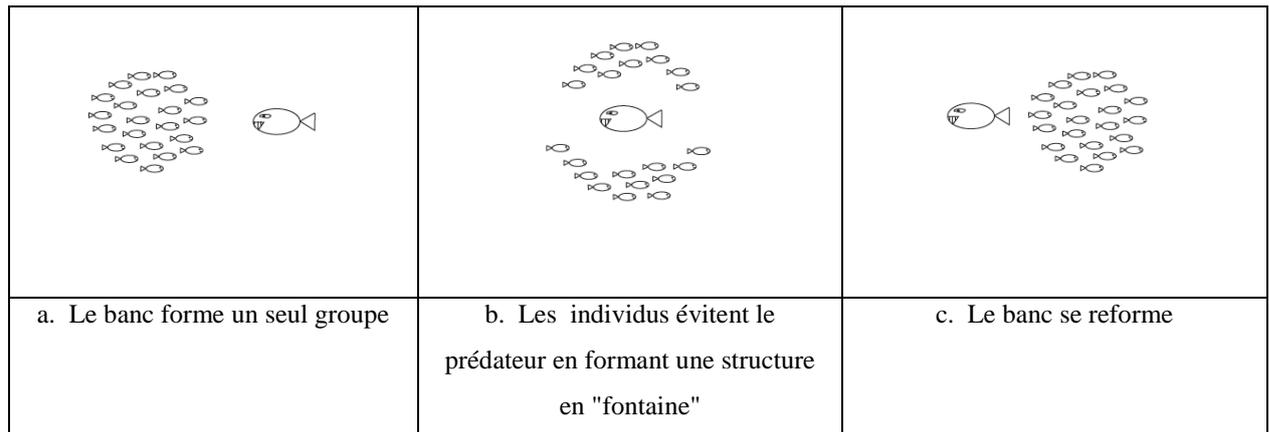


Figure I.6 - Schéma de l'évitement d'un prédateur par un banc de poissons [2]

Les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

Dans la théorie au moins, les différents membres du groupe peuvent profiter des découvertes et de l'expérience précédente de tous les autres membres du groupe pendant la recherche de la nourriture. Cette relation suggère que le partage social de l'information entre les membres offre un avantage évolutionnaire : Cette hypothèse était fondamentale au développement de l'optimisation par essaim de particules.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de la meilleure position trouvée par tout le groupe. Les nouvelles vitesses et directions de la particule seront définies en fonction de trois tendances ; la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers la meilleure position trouvée par le groupe.

Le PSO peut résoudre une variété de problèmes d'optimisation difficiles. Il a montré une convergence plus rapide que d'autres algorithmes. Un autre avantage du PSO est qu'il a très peu de paramètres à ajuster, qui le rend particulièrement facile à mettre en application.

b. Représentation

Tout comme les GA, les algorithmes à essaim de particules débutent par une population initiale des individus, appelée en PSO(essaim) mais la représentation des individus (particules)

est plus simple que celle en GA. En PSO, l'essaim est représenté comme un groupe de particules initialisées aléatoirement par des positions dans le domaine de recherche et par des vitesses. La tâche essentielle confiée aux particules de l'essaim est de faire balayer le domaine des solutions faisables en cherchant la meilleure position qui est la solution optimale. Donc, les particules de chaque génération sont les mêmes en changeant seulement leurs caractéristiques, position et vitesse.

c. Evolution

La reproduction des populations en PSO est faite toujours après l'évaluation des individus (particules) du présent essaim et la comparaison des valeurs pour chaque particule. Donc, on ne parle pas ici de la recombinaison ou de la mutation.

Le modèle de l'algorithme du PSO est basé sur les deux facteurs suivants:

- La mémoire autobiographique, qui se rappelle la meilleure position ($Pbest_i$) rencontrée par chacun des particules.
- L'information annoncée, qui est la meilleure solution ($Pbest_g$) trouvée par l'essaim.

A chaque génération et après la sélection, les vitesses de déplacement des particules doivent être ajustées par l'équation standard suivante [7] :

$$V_i^{(k+1)} = w \cdot V_i^{(k)} + c_1 \cdot r_1 \cdot (Pbest_i - X_i^{(k)}) + c_2 \cdot r_2 \cdot (Pbest_g - X_i^{(k)}) \quad (I-1)$$

r_1, r_2 : Des constantes aléatoires dans l'intervalle (0, 1).

w : Constante d'inertie initialisé dans l'intervalle (0, 1).

$V_i^{(k)}$: Vitesse actuelle.

$X_i^{(k)}$: Position actuelle dans l'hypermespace de recherche.

$Pbest_i$: La meilleure position trouvée par la $i^{ième}$ particule.

$Pbest_g$: La meilleure solution trouvée par l'essaim.

c_1, c_2 : Constantes d'accélération, elles ont le rôle de contrôler le mouvement des particules à chaque itération.

Le premier terme de cette équation montre la tendance de la particule à suivre son chemin, le deuxième montre sa tendance à revenir vers la meilleure position qu'elle a rencontrée et le troisième terme indique sa tendance à aller vers la meilleure solution trouvée par l'essaim.

Après l'ajustement des vitesses, chaque particule doit effectuer un déplacement pour atteindre sa nouvelle position selon l'équation suivante [7] :

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)} \quad (\text{I-2})$$

I.3. Etat de l'art sur les algorithmes évolutionnaires

L'optimisation des régimes de coupe pour la fabrication des pièces mécaniques consiste à chercher les valeurs optimales des paramètres de coupe (vitesse de coupe, vitesse d'avance, profondeur de passe, etc.), en tenant compte d'un ou de plusieurs critères (le coût de production, le temps de production, la productivité, la précision d'usinage, etc.).

L'intérêt de l'optimisation des régimes de coupe réside dans la réduction du prix de revient des pièces à usiner, l'augmentation de la productivité, l'amélioration de la précision dimensionnelle des pièces et la simplification des programmes d'usinage sur les Machines-Outils à Commande Numérique [6].

Une synthèse sur les techniques évolutionnaire, à savoir algorithme génétique (GA) essaim de particules (PSO), l'optimisation par colonies de fourmis (ACO) pour l'optimisation des régimes de coupe sera présentée.

Dans la littérature les GA ont été largement appliqués par les chercheurs pour optimiser les paramètres d'usinage.

La technique d'optimisation par l'algorithme génétique (GA) à été utilisée par **M. Assas** et al (2001) [6] pour résoudre un problème d'optimisation des conditions de coupe multicritère dans le processus de tournage. Les auteurs optimisent le coût et le temps de production simultanément.

Les résultats de calcul montrent une nette diminution des coûts et du temps d'usinage par rapport à la méthode du simplexe [6] et celle du gradient [12].

Le travail développé par **Franci Cus, Joze Balic** (2003) [13] détermine les paramètres d'usinage optimaux qui minimisent le coût, le temps et la qualité d'un produit final en fraisage par les algorithmes génétiques. Les résultats expérimentaux de l'approche proposée sont comparés avec les résultats tirés de la littérature récente : l'algorithme du recuit simulé et les

approches de programmation linéaire. L'algorithme génétique prouve son efficacité et peut être intégré dans un système de fabrication intelligent pour résoudre des problèmes d'optimisation des usinages complexes.

L'article de **Ramon Quiza Sardiñas** et al (2006) [14] présente une technique d'optimisation multi-objectif basée sur l'algorithme génétique afin de déterminer les paramètres de coupe (la profondeur de coupe, l'avance et la vitesse de coupe) en processus de tournage et d'optimiser simultanément la durée de vie de l'outil et le temps de fonctionnement qui sont deux objectifs contradictoires.

Dans le tableau ci-dessous nous présentons quelques états de l'art réalisés par **Norfadzlan Yusupet** al(2012) [15].

Nu	Auteurs/Année	Les paramètres de processus	Processus d'usinage	Performance d'usinage	Remarques
1	Singh and Rao (2007)	Vitesse de coupe et avance	Le tournage dur	Rugosité de surface	L'optimisation par GA donne des valeurs minimales de la rugosité de surface.
2	Parent et al. (2007)	Vitesse de coupe, avance et profondeur de coupe	Fraisage en bout	Coûts de production	Le GA a été proposé pour trouver un usinage optimal des paramètres du procédé.
3	Sankar et al. (2007)	Vitesse de coupe, avance et profondeur de coupe	Tournage multi-passe dressage et perçage	Coûts de production	La modification des opérateurs génétiques améliore la recherche d'une manière plus efficace que l'algorithme génétique classique.
4	Zhang and Chen (2009)	Vitesse de rotation, vitesse d'avance, profondeur de coupe, largeur de coupe	Fraisage	Durée de vie, coûts de production	Les paramètres de fraisage les plus optimisés ont été obtenus.

5	Xie and Pan (2010)	Vitesse de coupe, vitesse d'avance et profondeur de coupe	Tournage multi-passe	Coûts de production	L'algorithme proposé est efficace dans la recherche des paramètres d'usinage optimal, réduisant considérablement le coût de production unitaire.
6	Bharathi and Baskar (2010)	Nombre de passes, vitesse de coupe, avance et profondeur de coupe	Tournage multi-passe	Coût de production, le taux d'enlèvement du métal	D'après les expériences, le GA n'a pas donné de meilleurs résultats par rapport aux PSO et SA dans l'opération de trois passes.
7	Saffar and Razfar (2010)	Vitesse de coupe, vitesse d'avance et l'angle de coupe radial	Fraisage en bout	Force de coupe	Les résultats obtenus indiquent que les paramètres optimisés sont capables d'usiner la pièce avec plus de précision et avec une meilleure finition de surface.
8	Xu et al. (2010)	Vitesse d'avance, profondeur de coupe, largeur de coupe	Fraisage	Force de coupe, durée de vie et la rugosité de la surface usinée, le taux d'enlèvement du métal Ti6Al4V	Les résultats d'optimisation montrent que le système d'optimisation peut améliorer la productivité du fraisage.
9	Yongzhi et al. (2010)	La profondeur Axiale et radiale de coupe, et l'angle hélicoïdal	Fraisage à grande vitesse	Force de coupe, le taux d'enlèvement du métal	Le résultat montre qu'il est possible de sélectionner l'optimal pour obtenir la force de coupe minimale, relativement au bon taux d'enlèvement de métal.
10	Zain et al.	Angle radial	Fraisage en	Rugosité de	Le projet d'intégration de

	(2011c)	de coupe, la vitesse de coupe et d'avance	bout	surface	la SA et GA donne un plus faible nombre d'itérations par rapport aux techniques conventionnelles de SA et GA .
11	Zain et al. (2010a)	Vitesse de coupe, vitesse d'avance et l'angle radial de coupe	Fraisage en bout	Rugosité de surface	Le GA est capable d'estimer les paramètres du processus optimaux comparés aux données expérimentales, par utilisation d'un modèle de régression.
12	Xie and Guo (2011)	Profondeur de coupe, la vitesse de coupe et d'avance	Tournage multi-passe	Coûts de production	La démarche d'optimisation peut trouver les meilleurs résultats que d'autres algorithmes proposés pour réduire de manière significative les coûts de production unitaires.
13	Ganesan et al. (2011)	Profondeur de coupe, la vitesse de coupe et d'avance	Tournage multi-passe	Le temps de production	Le GA et le PSO ont été employés pour trouver les paramètres d'usage optimal pour le profil en continu.
14	An et al. (2011)	Vitesse de coupe, vitesse d'avance, la profondeur de coupe, et nombre de passes	Fraisage multi-passe	Coûts de production	La méthode permet d'obtenir une réduction des coûts de production unitaires par rapport aux résultats de la littérature.

Tableau I.1 - Tableau de techniques récentes utilisant (GA) pour déterminer les paramètres d'usage optimal [15]

M. Assas et al (2005) [7] introduisent une technique d'optimisation par essaim de particules (PSO) pour résoudre le problème d'optimisation des conditions de coupe, afin de minimiser le temps et le coût de production simultanément, appliquée sur une opération de tournage. Les résultats montrent une nette amélioration des coûts de production et du temps par rapport à la méthode du gradient GM et l'algorithme génétique GA.

Liang Gao et al (2012) [16] proposent dans leur article une nouvelle méthode basée sur l'optimisation par essaim de particules cellulaire (CPSO) pour résoudre le problème d'optimisation des paramètres de coupe (la profondeur de coupe, l'avance et la vitesse de coupe) du processus de fraisage multi-passe. L'objectif est de minimiser le temps total de production. Les résultats d'un cas d'application illustrent l'efficacité de l'approche proposée que la plupart des variantes de(PSO).

Les algorithmes de colonies de fourmis forment une classe des méta-heuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. L'emploi de l'algorithme colonies de fourmis se généralise à plusieurs domaines [2].

Dans le cadre d'un problème biomédical concernant les pathologies du vieillissement oculaire **J. Dréo** (2004) [2] apporte une nouvelle approche basée sur l'optimisation de colonies de fourmis dans le processus de recalage d'images d'angiographies rétiniennes et ceci par automatisation du suivi des lésions de l'œil nécessitant en effet une étape d'optimisation.

En mathématiques appliquées et informatiques, **A. Ben Mohamed** (2009) [3] a utilisé une hybridation d'une méthode méta-heuristique (ACO) basée sur l'heuristique (SMD) pour résoudre le problème de Bin-Packing en deux dimensions (2BP) qui consiste à déterminer le nombre minimum de grands rectangles identiques, dits "bins", nécessaires pour ranger un ensemble de petits rectangles, dits "objets». **A. Ben Mohamed** a utilisé l'heuristique (SMD) pour choisir la surface disponible maximale la plus adéquate pour placer l'objet en cours. L'algorithme colonie de fourmis a été exploité afin de faire varier l'ordre dans lequel les pièces (objets) sont placées dans les bins. Les résultats de simulation montrent la supériorité de l'hybridation par rapport à l'algorithme génétique et la recherche tabou.

Dans le domaine aérien, **C.Edmond Bichot, J.Marc Alliot** [20] présentent dans leur article la méthode d'optimisation par colonies de fourmis et son application au problème du découpage de l'espace aérien. Les résultats obtenus par cette méthode ont été comparés avec

ceux obtenus par la méthode du recuit simulé et d'une méthode classique d'optimisation. Le meilleur découpage obtenu est celui de la méthode recuit simulé.

Dans le domaine mécanique, les auteurs [21-23], utilisent la technique de colonies de fourmis pour résoudre les problèmes d'optimisation des conditions de coupe.

Dans le travail de **K. Vijayakumar** et al (2003) [21], les auteurs ont proposés une nouvelle technique d'optimisation basée sur l'algorithme de colonies de fourmis (ACO) pour résoudre des problèmes d'optimisation multi-passe en tournage. Les paramètres d'usinage sont déterminés en minimisant le coût de production, soumis à diverses contraintes d'usinage. Les résultats de calcul montrent une diminution de coût d'usinage, en les comparant avec la méthode génétique et celle du recuit simulé.

Dans la recherche de **J.Wu, Y.Yao** (2008) [22], la technique méta-heuristique, basée sur l'algorithme de colonies de fourmis (ACO) modifiée en continu (MCACO), a été proposée pour trouver les paramètres optimaux d'usinage en tournage multi-passe, telles que la vitesse de coupe, l'avance et la profondeur de coupe afin de minimiser le coût de production unitaire. D'après les résultats de l'expérimentation, il a été constaté que la technique proposée améliore le coût unitaire de production par rapport aux autres méthodes (flottant codé GA(FEGA), (SA), méthode des courses de côtes (HC) et de Newton (NM)).

F.Cüset al (2009) [23] utilisent l'optimisation (ACO) en tournage multi-passe. Dans cette étude, les performances d'usinage modélisées sont le temps de fabrication, la qualité de coupe et le coût de production. L'approche proposée utilise le système adaptatif d'inférence neuro-floue (ANFIS) pour obtenir la fonction objectif, et un algorithme (ACO) pour obtenir la valeur optimale objectif. De l'expérience, il a été constaté que le projet de l'approche ANFIS-ACO surpasse les (GA) et (SA).

I.4. Conclusion

Dans ce chapitre nous avons mis en revue plusieurs techniques d'optimisations évolutionnaires (GA, PSO et ACO) dans le but de situer notre choix de la technique (Optimisation par colonie de fourmis) dans le domaine de la mécanique et précisément dans l'optimisation des conditions de coupe.

Dans le deuxième chapitre nous aborderons la méthode colonie de fourmis d'une manière plus détaillée.

CHAPITRE II

OPTIMISATION

PAR COLONIE DE FOURMIS

CHAPITRE II

OPTIMISATION

PAR COLONIE DE FOURMIS

Table des matières

II.1. Introduction	20
II.2. Optimisation par colonies de fourmis	20
II.2.1. Comportement des fourmis	21
II.2.1.1. La communication	21
II.2.1.2. Stigmergie	22
II.2.1.3. L'auto-organisation	22
II.2.2. Similarités et différences entre les fourmis virtuelles et réelles	24
II.2.2.1. Points communs	24
II.2.2.2. Différences	25
II.2.3. Expériences	26
II.2.3.1. Pont binaire de Deneubourg	26
II.2.3.2. Expérience du double pont binaire	27
II.2.3.3. Effet de la coupure d'une piste de phéromone	28
II.2.3.4. Conclusions	29
II.3. Principe de l'algorithme colonie de fourmis	29
II.3.1. Le problème du voyageur de commerce (PVC)	29
II.3.1.1. Formalisation	30
II.3.2. Système de fourmis AS (Ant System)	30
II.3.2.1. Conventions	30
II.3.2.2. Choix d'implémentation	31
II.3.2.3. Choix des transitions	31
II.3.2.4. Mise à jour des phéromones	31
II.3.2.5. Quantités de phéromones déposées	32
II.3.3. Fonctionnement de l'algorithme (PVC)	33
II.4. Conclusion	35

II.1. Introduction

Les insectes sociaux en général, et les fourmis en particulier, résolvent naturellement des problèmes relativement complexes. Les biologistes ont étudié comment les fourmis arrivent à résoudre collectivement des problèmes trop complexes, notamment les problèmes de choix lors de l'exploitation de sources de nourriture [2].

Marco Dorigo, de l'université Libre de Bruxelles a inventé en 1991 l'algorithme à colonies de fourmis lorsqu'il a observé des fourmis dans leur chemin de recherche de la nourriture. Celles-ci ont la capacité de trouver le chemin le plus court entre leur nid et une source de nourriture, en contournant les obstacles qui jonchent leur chemin.

L'idée générale de l'algorithme de colonie de fourmis est d'imiter le comportement coopératif d'une colonie de fourmis naturelles à l'aide des fourmis artificielles se déplaçant à travers le graphe qui représente le problème à résoudre. Le principe est le suivant : les fourmis cherchent de la nourriture et se déplacent de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière une substance chimique appelée phéromone. Cette substance a la propriété de s'évaporer au cours du temps et a pour but de guider les fourmis vers leur objectif. Une fois cet objectif atteint (dans notre cas, la nourriture trouvée), les fourmis rentrent au nid en empruntant le même chemin qu'à l'aller, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, on va donc constater l'émergence du plus court chemin vers la nourriture grâce au renforcement de la trace de phéromone.

II.2. Optimisation par colonies de fourmis

L'optimisation par colonies de fourmis (ACO) est une nouvelle méthode approchée qui s'ajoute à la classe des méta-heuristiques telles que les algorithmes génétiques, essaim de particules, la recherche avec tabou, le recuit simulé etc.

L'optimisation par colonies de fourmis étudie les systèmes artificiels qui prennent l'inspiration du comportement de vraies colonies de fourmis [24]. Toutes ces idées abstraites sont inspirées des travaux de la Biologiste Jean Louis Deneubourg 1983 sur le comportement des fourmis [25].

L'algorithme de colonies de fourmis a été à l'origine principalement utilisé pour produire des solutions quasi-optimales au problème du voyageur de commerce, puis plus généralement, aux problèmes d'optimisation combinatoire comme applications aux problèmes de coloration de graphe, applications aux réseaux de télécommunications, le traitement d'image [5,18], et aussi dans notre cas aux problèmes d'usinages [21, 22, 23].

II.2.1. Comportement des fourmis

II.2.1.1. La communication

Les mécanismes de communication sont très élaborés chez les fourmis. Ils sont sonores, tactiles, visuels ou chimiques. Dans certaines espèces la vue est faible et même parfois nulle, mais le toucher, l'odorat et le goût sont très développés.

a. Les pistes de phéromones

➤ La communication chimique

Les moyens de communication utilisés par les fourmis sont donc nombreux. Toutefois leur efficacité est insignifiante comparée à celle de la communication chimique [17]. En marchant du nid à la source de nourriture et vice-versa (ce qui dans un premier temps se fait essentiellement de façon aléatoire), les fourmis déposent au passage sur le sol une substance odorante grâce à une glande située dans leur abdomen, appelée *phéromones* que seules les fourmis peuvent sentir figure (II.1). Cette substance permet ainsi donc de créer une piste chimique, sur laquelle les fourmis s'y retrouvent figure (II.2). En effet, d'autres fourmis peuvent détecter les phéromones grâce à des capteurs sur leurs antennes.

Les phéromones ont un rôle de marqueur de chemin : quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones. Cela leur permet de retrouver le chemin vers leur nid lors du retour. D'autre part, les odeurs peuvent être utilisées par les autres fourmis pour retrouver les sources de nourritures trouvées par leurs congénères. Ce comportement permet de trouver le chemin le plus court vers la nourriture lorsque les pistes de phéromones sont utilisées par la colonie entière [25].

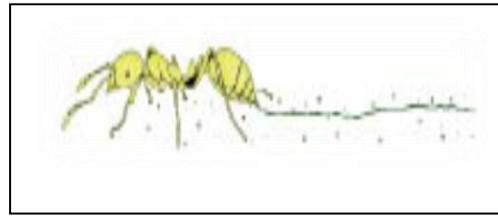


Figure II.1 - Dépôt de la phéromone déposée par une fourmi [17]

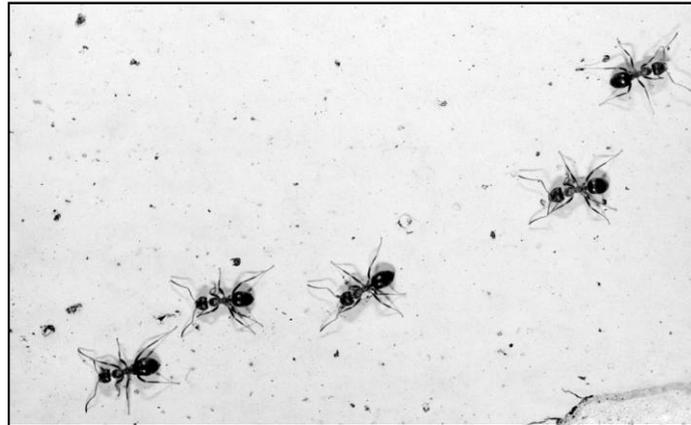


Figure II.2 - Fourmis suivent une piste de phéromone [2]

II.2.1.2. Stigmergie

C'est la possibilité d'échanger des informations par le biais de modifications de l'environnement, il s'agit d'une forme de communication indirecte entre les individus d'un groupe. C'est par ce biais que les fourmis interagissent entre elles, notamment grâce aux phéromones.

II.2.1.3. L'auto-organisation

Les fourmis prennent des décisions localement et ne sont pas soumises à des ordres hiérarchiques. L'auto-organisation de la population contribue à l'émergence d'une solution vers laquelle elle n'a été jamais orientée. C'est notamment grâce à une vision locale de l'environnement que les algorithmes de colonies de fourmis sont robustes (ils sont performants, même en cas de défaillance de certains individus) et flexibles (une colonie peut s'adapter à un nouvel environnement) [3].

Nous trouvons ainsi, dans la nature, plusieurs exemples qui donnent des idées pour le principe d'auto-organisation figure (II.3). À titre d'exemple, nous pouvons citer les vols groupés d'oiseaux figure (II.3c). Ces comportements se trouvent aussi chez des organismes plus simples unicellulaires, comme les bactéries figure (II.3d). De nouvelles recherches ont établi que les bactéries utilisent des molécules pour communiquer entre elles. Elles utilisent un réseau composé de liens de cellule à cellule. Elles sont capables de détecter des changements dans leur environnement, s'associer avec des bactéries d'une même espèce, faire des alliances bénéfiques avec d'autres espèces, et prendre l'avantage sur des espèces concurrentes [5].

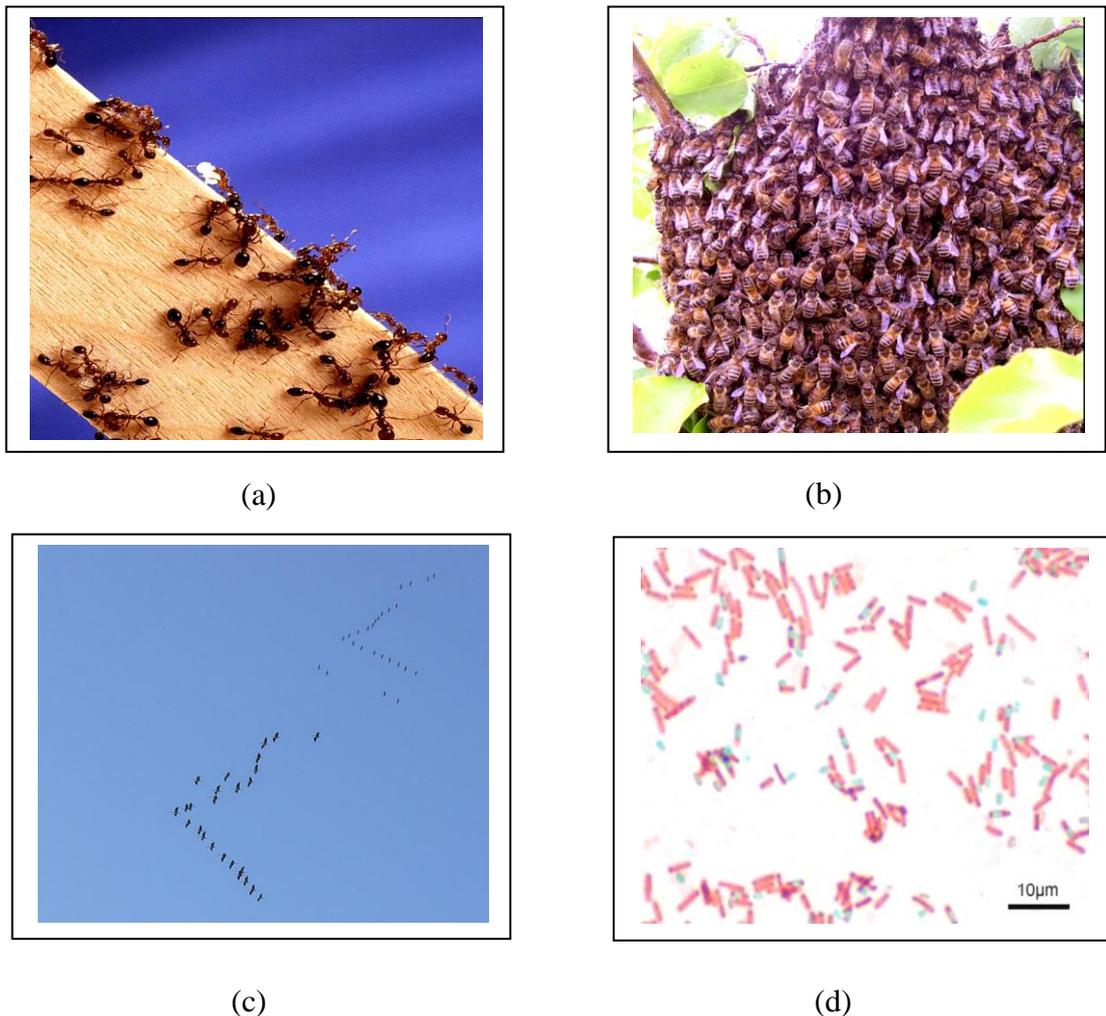


Figure II.3 - Auto-organisation dans les systèmes biologiques : (a) Une colonie de fourmis qui ramène de la nourriture vers le nid (b) Un essaim d'abeilles (c) Un vol groupé d'oiseau (d) Une formation de bactéries [5]

II.2.2. Similarités et différences entre les fourmis virtuelles et réelles

Les fourmis virtuelles ont une double nature. D'une part, elles modélisent les comportements abstraits de fourmis réelles, et d'autre part, elles peuvent être enrichies par des capacités que ne possèdent pas les fourmis réelles, afin de les rendre plus efficaces que ces dernières. Nous allons maintenant synthétiser ces ressemblances et différences.

II.2.2.1. Points communs

❖ Colonie d'individus coopérants

Comme pour les fourmis réelles, une colonie virtuelle est un ensemble d'entités non-synchronisés, qui se rassemblent ensemble pour trouver une "bonne" solution au problème considéré. Chaque groupe d'individus doit pouvoir trouver une solution même si elle est mauvaise.

❖ Pistes de phéromones

Ces entités communiquent par le mécanisme des pistes de phéromone. Cette forme de communication joue un grand rôle dans le comportement des fourmis : son rôle principal est de changer la manière dont l'environnement est perçu par les fourmis, en fonction de l'historique laissé par ces phéromones.

❖ Évaporation des phéromones

La méta-heuristique ACO comprend aussi la possibilité d'évaporation des phéromones. Ce mécanisme permet d'oublier lentement ce qui s'est passé avant. C'est ainsi qu'elle peut diriger sa recherche vers de nouvelles directions, sans être trop contrainte par ses anciennes décisions.

❖ Recherche du plus petit chemin

Les fourmis réelles et virtuelles partagent un but commun : recherche du plus court chemin reliant un point de départ (le nid) à des sites de destination (la nourriture).

❖ Déplacement locaux

Les vraies fourmis ne sautent pas des cases, tout comme les fourmis virtuelles. Elles se contentent de se déplacer entre sites adjacents du terrain.

❖ Choix aléatoire lors des transitions

Lorsqu'elles sont sur un site, les fourmis réelles et virtuelles doivent décider sur quel site adjacent se déplacer. Cette prise de décision se fait au hasard et dépend de l'information locale déposée sur le site courant. Elle doit tenir compte des pistes de phéromones, mais aussi du contexte de départ (ce qui revient à prendre en considération les données du problème d'optimisation combinatoire pour une fourmi virtuelle).

II.2.2.2. Différences

Les fourmis virtuelles possèdent certaines caractéristiques que ne possèdent pas les fourmis réelles :

❖ Elles vivent dans un monde non-continu

Leurs déplacements consistent en des transitions d'état.

❖ Mémoire (état interne) de la fourmi

Les fourmis réelles ont une mémoire très limitée. Tandis que nos fourmis virtuelles mémorisent l'historique de leurs actions. Elles peuvent aussi retenir des données supplémentaires sur leurs performances.

❖ Qualité de la solution

Les fourmis virtuelles déposent une quantité de phéromone proportionnelle à la qualité de la solution qu'elles ont découverte.

❖ Retard dans le dépôt de phéromone

Les fourmis virtuelles peuvent mettre à jour les pistes de phéromones de façon non immédiate : souvent elles attendent d'avoir terminé la construction de leur solution.

Ce choix dépend du problème considéré bien évidemment.

❖ Capacités supplémentaires

Les fourmis virtuelles peuvent être pourvues de capacités artificielles afin d'améliorer les performances du système. Ces possibilités sont liées au problème et peuvent être :

1. l'anticipation : la fourmi étudie les états suivants pour faire son choix et non seulement l'état local.
2. le retour en arrière : une fourmi peut revenir à un état déjà parcouru car la décision qu'elle avait prise à cet état a été mauvaise [25].

II.2.3. Expériences

II.2.3.1. Pont binaire de Deneubourg

L'expérience montre un nid d'une colonie de fourmis, qui est séparé d'une source de nourriture par un pont à deux voies de même longueur. On laisse évoluer les fourmis sur le pont, on trace ainsi en fonction du temps, le graphe du nombre de fourmis empruntant chaque branche.

Le résultat de l'expérience est exposé à la figure (II.4). L'illustration (II.4a) représente la configuration physique de l'expérience. Le graphique (II.4b) indique l'évolution de ce système en fonction du temps : on constate que les fourmis ont tendance à emprunter le même chemin (par exemple celui du haut) après une dizaine de minutes.

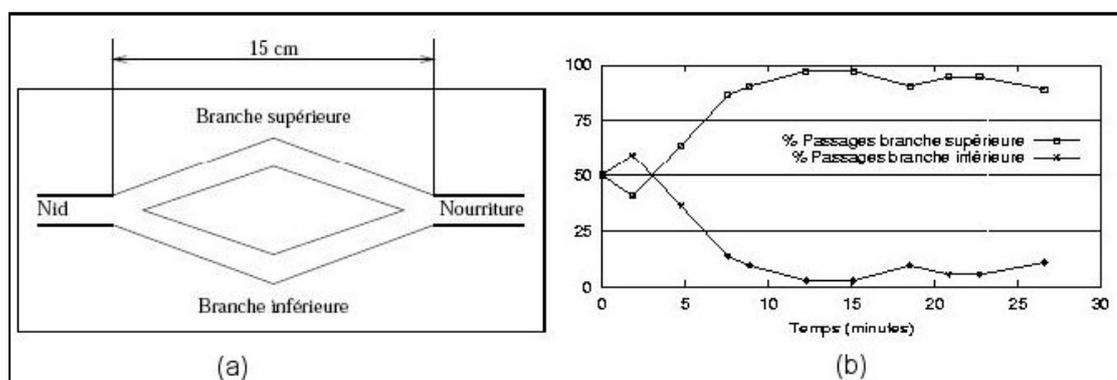


Figure II.4 - Pont binaire de Deneubourg [25]

a. Explication

Au départ, il n'y a pas de phéromone sur le pont. Donc, chaque branche peut être choisie par une fourmi avec la même probabilité. Néanmoins, dans notre exemple, après une certaine période, des variations aléatoires ont fait qu'un peu plus de fourmis ont choisi le chemin du haut plutôt que celui du bas.

Puisque les fourmis déposent des phéromones en avançant et puisqu'elles sont plus nombreuses en haut qu'en bas, le chemin du haut comportera plus de phéromones. Cette quantité supérieure de phéromone incite plus de fourmis à choisir la branche du haut, donc la quantité de phéromone déposée augmentera encore plus.

On en déduit que plus les fourmis suivent un chemin, plus ce chemin devient intéressant à suivre. Ainsi la probabilité avec laquelle une fourmi choisit un chemin, augmente avec le nombre de fourmis qui ont pris ce chemin précédemment.

II.2.3.2. Expérience du double pont binaire

On peut se demander à présent quel serait l'effet de l'augmentation de la longueur d'une des deux branches du pont. L'effet produit sera que la branche la plus courte sera sélectionnée [25].

Les fourmis qui reviendront au nid le plus rapidement, après avoir visité la source de nourriture, sont celles qui auront emprunté les deux branches les plus courtes. Ainsi, la quantité de phéromone présente sur le plus court chemin est relativement plus importante que celle présente sur le chemin le plus long. En sachant qu'une piste présentant une plus grande concentration en phéromone est plus attirante pour les fourmis, elle aura une plus grande probabilité d'être choisie. Le chemin le plus court sera renforcé par les passages successifs et deviendra à terme la piste empruntée par la majorité des insectes [3].

On peut généraliser cela à plus de deux chemins possibles : dans la figure (II.5a), on a utilisé un double pont avec quatre chemins possibles de différentes longueurs. On voit dans la figure (II.5b) que la plupart des fourmis finissent par choisir le chemin le plus court. Les expériences montrent que quand environ cent fourmis ont déjà emprunté le pont, plus de 90 pourcents d'entre elles sélectionnent le chemin le plus court : les fourmis convergentes donc assez rapidement [25].

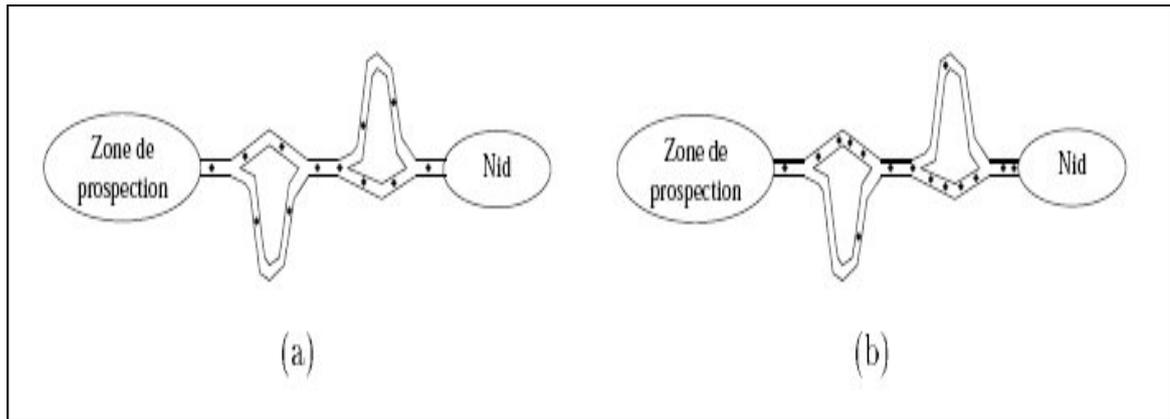


Figure II.5 - Recherche de nourriture par une colonie de fourmis :

(a) Début de l'expérience, (b) Fin de l'expérience [25]

II.2.3.3. Effet de la coupure d'une piste de phéromone

Une autre expérience qui illustre l'intelligence collective des fourmis concerne la coupure d'une piste de phéromone, voir figure (II.6) On suppose que les fourmis suivent une piste de phéromones sur laquelle est déposé un obstacle. Les fourmis choisissent alors de contourner l'obstacle par la gauche ou par la droite. Dans la mesure où aucun marquage n'existe autour de l'obstacle, les fourmis partiront à gauche ou à droite dans les mêmes proportions. Le chemin de droite étant plus court que celui de gauche, les fourmis qui l'empruntent, retrouveront plus vite la piste de phéromone déjà existante. À partir de ce moment, les fourmis qui arriveront devant l'obstacle préféreront suivre la piste de droite. Le nombre de fourmis passant par la droite augmentera proportionnellement avec la concentration de phéromones. De plus, l'évaporation des phéromones sera plus rapide sur la piste de gauche du fait de sa plus grande longueur [3].

La piste de gauche sera donc rapidement abandonnée, parce qu'elle en est beaucoup moins imprégnée : les fourmis passeront toutes très rapidement par la piste la plus courte [25].

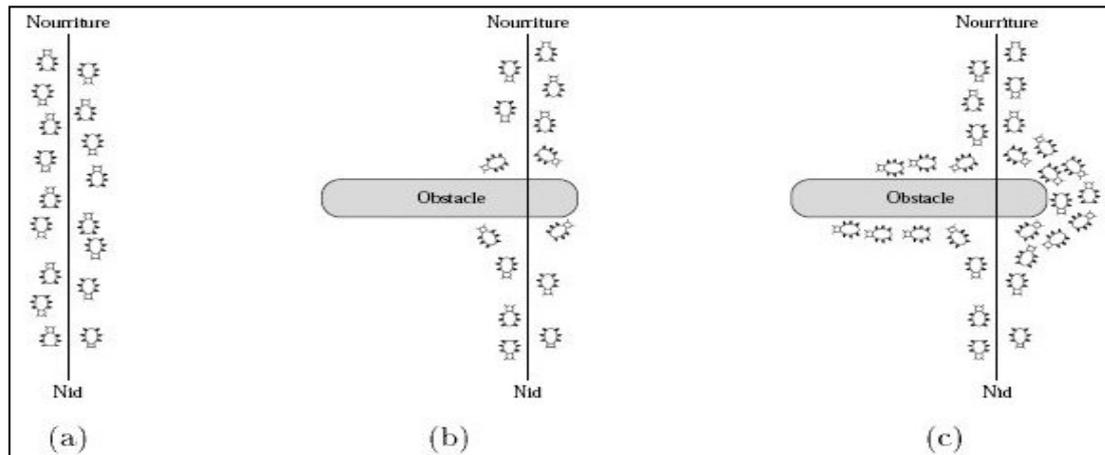


Figure II.6 - Effet de la coupure d'une piste de phéromone [25]

II.2.3.4. Conclusions

Il est intéressant de remarquer que bien qu'une seule fourmi soit capable de construire une solution (de trouver un chemin du nid à la nourriture), c'est seulement le comportement de *l'ensemble* de la colonie qui crée le chemin le plus court [25].

II.3. Principe de l'algorithme colonie de fourmis

Le problème du voyageur de commerce (« Travelling Salesman Problem », TSP) a fait l'objet de la première implémentation d'un algorithme de colonies de fourmis : le « Ant System » (AS). Le passage de la métaphore à l'algorithme est ici relativement facile et le problème du voyageur de commerce est bien connu et étudié. Le problème du voyageur de commerce consiste à trouver le trajet le plus court (désigné par « tournée » ou plus loin par « tour ») reliant n villes données, chaque ville ne devant être visitée qu'une seule fois. Le problème est plus généralement défini comme un graphe complètement connecté (X, U) , où les villes sont les nœuds X et les trajets entre ces villes, les arêtes U [2].

II.3.1. Le problème du voyageur de commerce (PVC)

Soit un graphe $G(X, U)$ tel que :

- Un ensemble fini X de nœuds représentant des villes.
- Un ensemble fini $U = \{(i, j) \mid i, j \in X\}$ d'arcs reliant les nœuds de X .
- Un ensemble de constantes d_{ij} représentant la longueur de chaque arc $(i, j) \in U$, c'est-à-dire la distance séparant la ville i de la ville j (avec $i, j \in X$).

On imagine alors un voyageur de commerce qui doit réaliser sa tournée en visitant une et une seule fois l'ensemble X des villes. Il souhaite déterminer la suite de villes qui minimisera la distance qu'il à parcourir.

II.3.1.1. Formalisation

➤ **Circuit hamiltonien**

Est un circuit qui passe exactement une fois par tous les sommets du graphe.

➤ **La longueur d'un circuit μ**

Est la somme des longueurs des arcs qui le composent, soit [25] :

$$L(\mu) = d_{u_q, u_1} + \sum_{i=1}^{q-1} d_{u_i, u_{i+1}} \quad (\text{II-1})$$

➤ **Le TSP (Traveling Salesman Problem)**

Est le problème consistant à trouver un circuit hamiltonien de longueur minimale sur le graphe $G(X, U)$.

II.3.2. Système de fourmis AS (Ant System)

II.3.2.1. Conventions

Les variables que nous devons définir pour comprendre la suite sont les suivantes :

- $b_i(t)$ (où $i \in X$) le nombre de fourmis dans la ville i à l'instant t
- $m = \sum_{i \in X} b_i$ leur nombre total, invariant dans le temps
- $\tau_{ij}(t)$ la valeur de τ_{ij} , à l'instant t
- $n = |X|$, le nombre de villes
- $\eta_{ij} = \frac{1}{d_{ij}}$, la visibilité d'une ville j quand on est placé sur la ville i , invariante dans le temps.

Précisons maintenant le comportement de l'ensemble de la colonie :

- A tout instant t , chaque fourmi choisit une ville de destination selon un choix défini.

- Toutes les fourmis se placent à l'instant $t + 1$ dans une ville de leur choix. On appelle une itération de l'algorithme (AS), l'ensemble de déplacements de l'ensemble de la colonie entre l'instant t et l'instant $t + 1$.
- Ainsi après n itérations, l'ensemble de la colonie aura effectué un circuit hamiltonien sur le graphe.
- De cette manière toutes les fourmis commenceront et finiront leur tour en même temps.

II.3.2.2. Choix d'implémentation

On précise que chaque fourmi a une mémoire implémentée par une liste de villes déjà visitées. Cela permet de garantir qu'aucune fourmi ne visitera deux fois une même ville au cours de sa recherche.

La mémoire de chaque fourmi est vidée lorsqu'elles ont terminé leur cycle.

II.3.2.3. Choix des transitions

Une fourmi k placée sur la ville i à l'instant t va choisir sa ville j de destination en fonction de la visibilité η_{ij} de cette ville et de la quantité de phéromones $\tau_{ij}(t)$ déposée sur l'arc reliant ces deux villes. Ce choix sera réalisé de manière aléatoire, avec une probabilité de choisir la ville j donnée par [25] :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^k(t)} [\tau_{il}(t)]^\alpha \cdot \eta_{il}^\beta} & \text{si } j \in N_i^k \\ 0 & \text{sinon} \end{cases} \quad (\text{II-2})$$

Où l'on définit l'ensemble N_i^k comme étant l'ensemble des villes que la fourmi k , placée sur la ville i , n'a pas encore visité à l'instant t dans le cycle courant. α et β sont deux paramètres qui contrôlent l'importance relative entre phéromones et visibilité. Ainsi si α est égal à 0, le choix se fera uniquement en fonction de la visibilité (si β est différent de zéro).

II.3.2.4. Mise à jour des phéromones

A la fin de chaque cycle (chaque fourmi a parcouru les n sommets qui composent le graphe), les variables des phéromones sont mises à jour selon la formule [2] :

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (\text{II-3})$$

Où $\rho \in [0, 1[$ est un coefficient qui définira la vitesse d'évaporation des phéromones sur les arcs entre l'instant t et l'instant $t+n$, et où $\tau_{ij}^k(t)$ représente la quantité de phéromone déposée par les fourmis dans ce même intervalle de temps sur l'arc (i, j) .

Le choix de ρ est important, en effet si ρ se rapproche trop de 1, on observe un effet de stagnation des phéromones sur les arcs, ce qui implique des inconvénients tel que le fait de voir les mauvaises solutions persister.

De même, choisir $\rho \approx 0$ implique une évaporation trop rapide des phéromones, donc amène la fourmi à un choix dépendant uniquement de la visibilité des nœuds.

II.3.2.5. Quantités de phéromones déposées

Appelons $T_k(t) = (u_{k_1}, \dots, u_{k_q})$ le tour réalisé par la $k^{\text{ème}}$ fourmi dans l'intervalle de temps $[t, t+n]$, et $L_k(t)$ sa longueur. $T_k(t)$ (donc $L_k(t)$ s'obtient en analysant la mémoire de la fourmi).

Soit $\Delta\tau_{ij}(t)$, la quantité de phéromones déposée par cette fourmi sur l'arc (i, j) dans ce même intervalle de temps. On le définit ainsi [25] :

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k(t)} & \text{si } (u \in T_k(t) \wedge u = (i, j)) \\ 0 & \text{sinon} \end{cases} \quad (\text{II-4})$$

Où Q est une constante. On voit bien ici que les phéromones sont régulées en fonction de la qualité de la solution obtenue car plus $L_k(t)$ est faible plus l'arc ne sera mis à jour en phéromones.

On peut maintenant définir le $\Delta\tau_{ij}(t)$ de la formule de mise à jour des phéromones ainsi [25] :

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (\text{II-5})$$

II.3.3. Fonctionnement de l'algorithme (PVC)

Nous allons maintenant préciser les différentes étapes de l'algorithme au cours du temps.

❖ Initialisation de l'algorithme

Les éléments s'agencent de la manière suivante au début de l'algorithme :

- Les m fourmis sont réparties aléatoirement sur les n villes.
- Pour chaque fourmi, la liste qui modélise sa mémoire contient sa ville de départ.
- Les pistes de phéromones sont initialisées comme suit : $\tau_{ij}(0) = c$, où c est une petite constante positive, qui ne peut être nulle (sinon, il y a un problème lors du calcul de formule (II-2))

❖ Fin d'un cycle

Après n itérations, nous sommes à l'instant t , toutes les fourmis ont terminé leur tour, chacune a une liste "mémoire" pleine et est revenue à sa propre ville de départ. À ce moment :

- Chaque fourmi calcule sa valeur $L_k(t)$.
- Les variables $\tau_{ij}^k(t)$ sont calculées conformément à la formule (II-2).
- Les variables de phéromone $\tau_{ij}(t)$ sont mises à jour suivant la formule (II-3). En d'autres termes, la fourmi refait son tour en sens inverse tout en déposant des phéromones.
- On observe quelle fourmi a trouvé le tour de longueur minimum (on recherche la fourmi k telle que $k = \min_{k=1}^m L_k(t)$). Si ce tour est meilleur que le meilleur tour jusqu'ici, on le mémorise.
- Les mémoires des fourmis (liste des villes visitées) sont effacées.
- Les fourmis recommencent un nouveau tour, toujours au départ de la ville sur laquelle elles avaient été placées au début de l'algorithme.

❖ Fin de l'algorithme

On arrête l'algorithme après un nombre de cycles égal à une constante NC_{max} . Si à partir d'un instant, toutes les fourmis font le même tour, l'algorithme s'interrompt: on est dans une

situation de stagnation où le programme arrête de chercher des alternatives. L'algorithme donne en retour le meilleur tour mémorisé [25].

Pour $t=1, \dots, t_{max}$

Pour chaque fourmi $k=1, \dots, m$

Choisir une ville au hasard

Pour chaque ville non visitée i

Choisir une ville j , dans la liste N_i^k des villes restantes, selon la formule (II-2)

Fin pour

Déposer une piste $\Delta\tau_{ij}^k(t)$ sur le trajet $T_k(t)$ conformément à Eq (II-4)

Fin pour

Évaporer les pistes selon la formule (II-3)

Fin pour

Figure II.7 - Algorithme de colonies de fourmis de bas : « Ant system » [2]

La figure (II.8) présente un exemple simplifié de problème du voyageur de commerce optimisé par un algorithme AS.

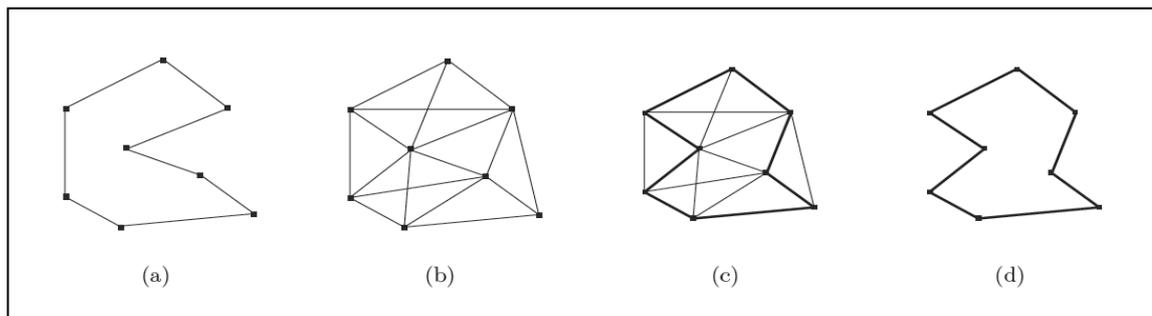


Figure II.8 - Le problème du voyageur de commerce optimisé par l'algorithme AS, les points représentent les villes et l'épaisseur des arêtes la quantité de phéromone déposée. (a) exemple de trajet construit par une fourmi, (b) au début du calcul, tous les chemins sont explorés, (c) le chemin le plus court est plus renforcé que les autres, (d) l'évaporation permet d'éliminer les moins bonnes solutions [2]

II.4. Conclusion

L'optimisation par colonies de fourmis (ACO) est une nouvelle méthode approchée basée sur l'étude des systèmes artificiels qui prennent l'inspiration du comportement de vraies colonies de fourmis.

Dans ce chapitre nous avons présenté l'essentiel de l'approche pour pouvoir l'utiliser en quatrième chapitre.

CHAPITRE III

OPTIMISATION

DES CONDITIONS DE COUPE EN TOURNAGE

Chapitre III

OPTIMISATION

DES CONDITIONS DE COUPE EN TOURNAGE

Table des matières

III.1. Introduction	36
III.2. Problème d'optimisation	36
III.3. Optimisation mono-objective	37
III.4. Critères d'optimisation	37
III.4.1. Temps d'usinage	38
III.5. Contraintes d'optimisation	39
III.6. Conclusion	40

III.1. Introduction

L'étude des phénomènes de coupe consiste à rechercher les relations entre les caractéristiques du matériau usiné, celles de l'outil et les paramètres fondamentaux caractérisant l'opération de coupe (forces, puissance consommée, usure d'outil et qualité de coupe). L'ensemble de ces données déterminent les performances du procédé.

Dans ce chapitre nous présentons le modèle mathématique utilisé par J.S.Agapiou [12] pour l'optimisation des paramètres d'usinage. J.S.Agapiou utilise comme fonction- objectif le (temps d'usinage) contenant plusieurs paramètres (la vitesse de coupe, l'avance, la profondeur de passe, la durée de vie de l'outil etc.) et un ensemble des contraintes (la puissance de coupe, l'état de surface, la température d'interface outil-copeau, l'effort de coupe etc.)

III.2. Problème d'optimisation

Le problème d'optimisation des conditions de coupe consiste à déterminer les valeurs des variables de coupe (V_c, f et d_c) pour chaque opération élémentaire d'usinage en prenant en compte les limites de performances des éléments de la cellule élémentaire d'usinage (C.E.U) qui se traduisent par des contraintes établies à partir de la lois de coupe et qui font intervenir les variables de coupe avec lesquelles le domaine des solutions réalisables est borné. D'une manière générale, la détermination des paramètres de coupe nécessite une bonne connaissance des facteurs d'incidence relatifs au couple outil-matière, à l'opération d'usinage et à l'environnement [7].

Les variables définissant les conditions de coupe en tournage sont des grandeurs géométriques et cinématique intervenant dans la dynamique de coupe et la nature du milieu dans lequel est effectuée la coupe (lubrification). Les variables retenues pour une opération de tournage sont :

- Pour les grandeurs géométriques :
 - L'avance par tour f (mm/tr)
 - La profondeur de passe d_c (mm)
- Pour les grandeurs cinématiques :
 - La vitesse de coupe V_c (m/mn) [26,7].

III.3. Optimisation mono-objectif

Un problème d'optimisation est exprimé comme une fonction- objectif à une ou plusieurs variables, pour être maximisée ou minimisée sous un certain nombre de contraintes indiquées. Les fonctions de contraintes définissent le domaine acceptable des variables pour le processus d'optimisation. De telles contraintes, pouvant être de types égalité ou inégalité, permettent la limitation des domaines sur les variables, dans la recherche de la (ou les) solution(s) optimale(s).

Un problème d'optimisation mono-objectif est souvent donné sous la forme suivante :

$$\left\{ \begin{array}{ll} \text{Minimiser } f(x) & f \in \mathbb{R}^n \\ \text{Sous les contraintes :} & \\ g_i(x) \leq 0 & i=1,\dots,n \\ h_j(x) = 0 & j=1,\dots,m \\ x \in S \subset \mathbb{R}^n & \end{array} \right.$$

Où:

$x \in \mathbb{R}^n$: Vecteur de composantes $(x_1, x_2, \dots, x_n)^T$ qui sont les inconnues du problème.

$f(x)$: Une fonction- objectif.

$g_i(x) \leq 0$: L'ensemble des contraintes d'inégalité du problème.

$h_j(x) = 0$: L'ensemble des contraintes d'égalité du problème.

III.4. Critères d'optimisation

Le coût d'usinage, le débit de copeau et le temps d'usinage sont les paramètres technoéconomiques d'usinage. Ce sont ces paramètres qui déterminent l'importance économique des pièces usinées [26].

L'analyse de la relation entre ces critères et les conditions de coupe conduit aux conclusions suivantes : plus les conditions de coupe sont réduites, plus les temps nécessaires à l'exécution des opérations sont élevés et par conséquent le coût de production est élevé. L'augmentation des conditions de coupe provoque une usure rapide des outils de coupe, d'où leur changement assez fréquent conduit à l'augmentation du temps et du coût de production [9].

III.4.1. Temps d'usinage

La production maximale sera atteinte lorsque le temps de fabrication unitaire sera minimal.

Le temps d'usinage unitaire est le temps nécessaire à la réalisation d'une passe sur une pièce. C'est la somme des temps machine, temps de changement de l'arrête, temps retour de l'outil et le temps de montage et démontage de la pièce. Il s'exprime par la relation [12] :

$$T_U = t_m + t_{cs} \left(\frac{t_m}{T} \right) + t_R + t_h \quad (\text{III-1})$$

Où t_m : Est le temps machine: (Temps de coupe avec une avance)

Puisque [12] :

$$\text{(En chariotage)} \quad t_m = \frac{\pi \cdot D \cdot L}{1000 \cdot V_c \cdot f} \quad (\text{III-2})$$

t_{cs} : Temps de changement d'outil y compris le temps de réaffûtage si nécessaire.

T : Durée de vie d'une arrête de l'outil de coupe.

t_R : Temps de retour rapide de l'outil pour la dernière passe.

t_h : Temps de montage et démontage de la pièce.

Où D et L sont respectivement le diamètre extérieur et la longueur de la partie usinée de la pièce.

La durée de vie de l'outil T est donnée par la loi de Taylor développée par la suite par Gilbert. Elle est donnée par l'équation suivante [12] :

$$T = K^{1/a_3} \cdot V_c^{-1/a_3} \cdot f^{-a_1/a_3} \cdot d_c^{-a_2/-a_3} \quad (\text{III-3})$$

Où a_1 , a_2 , et a_3 sont des exposants, K est un constant.

En remplaçant t_m et T par leurs expressions respectives (III-2) et (III-3) dans l'équation (III-1) et on posant :

$$A = \frac{\pi \cdot D \cdot L}{1000}$$

On obtient le temps d'usinage sous la forme suivante [12] :

$$T_U = A \cdot V_c^{-1} \cdot f^{-1} + A \cdot V_c^{(1-a_3)/a_3} \cdot f^{(a_1-a_3)/a_3} \cdot d_c^{(a_2/a_3)} \cdot K^{(-1/a_3)} \cdot t_{CS} + t_R + t_h \quad (\text{III-4})$$

III.5. Contraintes d'optimisation

En connaissant les limites des éléments de la C.E.U (Cellule élémentaire d'usinage) et les lois de coupe, nous pouvons formuler les contraintes par des égalités et des inégalités de fonctions des variables de coupe et des constantes de la C.E.U. Dans ce travail, nous considérons les limitations qui suivent [12] :

- L'avance f doit être comprise entre deux valeurs extrêmes programmables sur la machine-outil ; soit [12] :

$$\text{(En tournage)} \quad f_{min} \leq f \leq f_{max} \quad (\text{III-5})$$

- De même, la profondeur de passe doit être comprise entre deux extrêmes $d_{c \min}$ et $d_{c \max}$; soit [12] :

$$\text{(En tournage)} \quad d_{c \min} \leq d_c \leq d_{c \max} \quad (\text{III-6})$$

- La vitesse de coupe V_c doit être comprise dans les limites de validité du modèle de la loi de la durée de vie établie pour le couple outil-matière [12] :

$$\text{(En tournage)} \quad V_{c \min} \leq V_c \leq V_{c \max} \quad (\text{III-7})$$

Le processus de tournage est variable dans le temps par le changement au cours de l'usinage des paramètres de coupe ainsi que l'état de l'outil qui change par l'effet de l'usure. Pour avoir une optimisation globale du processus de tournage, l'ajustement des paramètres de coupe au cours de l'usinage doit être pris en considération. Or, cet ajustement ne sera possible qu'en faisant des mesures des variables du processus de coupe tels que : l'effort de coupe, la température et le niveau des vibrations. Les paramètres de coupe ne peuvent être ajustés au cours de l'usinage qu'à travers une commande numérique adaptative. Cette dernière doit aussi intégrer les contraintes liées à la physique et la mécanique du processus de coupe. On essaie de trouver les contraintes principales qui ont une grande influence sur les résultats d'usinage:

➤ **Limitation sur la puissance de coupe**

La puissance nécessaire à la coupe doit être inférieure à une puissance limite ; soit [12] :

$$\text{(En tournage)} \quad 0.0373 \cdot V_c^{0.91} \cdot f^{0.78} \cdot d_c^{0.75} \leq P_{max} \quad \text{(III-8)}$$

➤ **Limitation sur l'état de surface**

L'état de surface résultant d'usinage doit être inférieure à une rugosité maximale imposée soit [12] :

$$\text{(En tournage)} \quad 14.785 \cdot V_c^{-1.52} \cdot f^{1.004} \cdot d_c^{0.25} \leq Ra_{max} \quad \text{(III-9)}$$

➤ **Limitation sur la température de coupe**

Le facteur de contrôle est la température d'interface outil-copeau qui doit être inférieure à une température maximale ; soit [12] :

$$\text{(En tournage)} \quad 74.96 \cdot V_c^{0.4} \cdot f^{0.2} \cdot d_c^{0.105} - 17.8 \leq \theta_{max} \quad \text{(III-10)}$$

➤ **Limitation sur l'effort de coupe**

L'effort de coupe doit être inférieur à une force de coupe limite ; soit [12] :

$$\text{(En tournage)} \quad 844 \cdot V_c^{-0.1013} \cdot f^{0.725} \cdot d_c^{0.75} \leq F_{max} \quad \text{(III-11)}$$

Les constantes et les coefficients des modèles mathématiques des contraintes ci-dessus et de la durée de vie de l'outil sont obtenus expérimentalement par la méthode de planification des essais multi-factorielle [12]. Ces modèles ont été utilisés par la suite dans les travaux de M.Assas et al [9].

III.6. Conclusion

Nous avons présenté dans ce chapitre le critère de temps et l'ensemble des contraintes qui représentent les plus importantes limitations des performances à respecter pour résoudre le problème d'optimisation des conditions de coupe.

CHAPITRE IV

APPLICATION

ET DISCUSSION DES RESULTATS

Chapitre IV

APPLICATION ET DISCUSSION DES RESULTATS

Table des matières

IV.1. Introduction	41
IV.2. Cas d'application	41
IV.2.1. Algorithme proposé	41
IV.2.2. Algorithme de colonie de fourmis.....	41
IV.2.3. Fonction-objectif.....	48
IV.2.4. Contraintes d'optimisation	48
IV.2.5. Application	48
IV.3. Résultats	50
IV.4. Discussion et interprétation des résultats	52
IV.5. Comparaison	52
Conclusion	54
Recommandations	55
Références bibliographiques	56

IV.1. Introduction

Dans ce chapitre nous allons étudier l'application d'une méthode méta-heuristique à savoir colonie de fourmis afin d'optimiser les paramètres de coupes (vitesse de coupe et vitesse d'avance) sous des contraintes telles que (la puissance de coupe, l'état de surface, la température, l'effort de coupe, les limites d'avance et les limites de vitesse de coupe). Ensuite, nous comparerons les présents résultats avec d'autres résultats obtenus par des méthodes évolutionnaires et traditionnelle [6], [7], [12].

IV.2. Cas d'application

Le but de notre programme d'application est d'étudier les avantages de l'algorithme colonie de fourmis ACO par rapport à l'algorithme génétique GA, à celui de l'essaim de particules PSO et à la méthode du gradient GM en termes de convergence, ainsi que les améliorations apportées par le calcul évolutionnaire à l'optimisation des conditions de coupe.

Comme model mathématique d'optimisation nous avons pris celui traité par J.S.Agapiou [12] en utilisant la méthode du gradient GM.

IV.2.1. Algorithme proposé

Notre démarche d'implantation et de conception du model numérique d'optimisation s'inspire de l'algorithme général de voyageur de commerce (PVC), avec des réarrangements essentiels propres à notre problème d'étude (*minimiser le temps d'usinage*) figure (IV.3).

IV.2.2. Algorithme de colonie de fourmis

❖ Etape 1

1. Initialiser les constantes : it_{max} , m , α , ρ , $\tau_{ij}(0)$, ζ , d_c , P_{max} , $R_{a\ max}$, θ_{max} , F_{max} .
2. Générer les vitesses de coupe et des avances aléatoirement, avec des valeurs qui se situent dans la plage admissible (IV-17) et (IV-18).
3. Vérifier les contraintes de coupe supposées (IV-19), (IV-20), (IV-21) et (IV-22).
4. Initialiser tous les chemins (les valeurs de variables) avec une quantité de phéromones égales, (figure IV.1):

$$\tau_{ij}(0) = c, \quad c : \text{Constante.}$$

- Recommencer la recherche des solutions depuis le point de départ de la 1^{ère} couche jusqu'à la dernière couche.

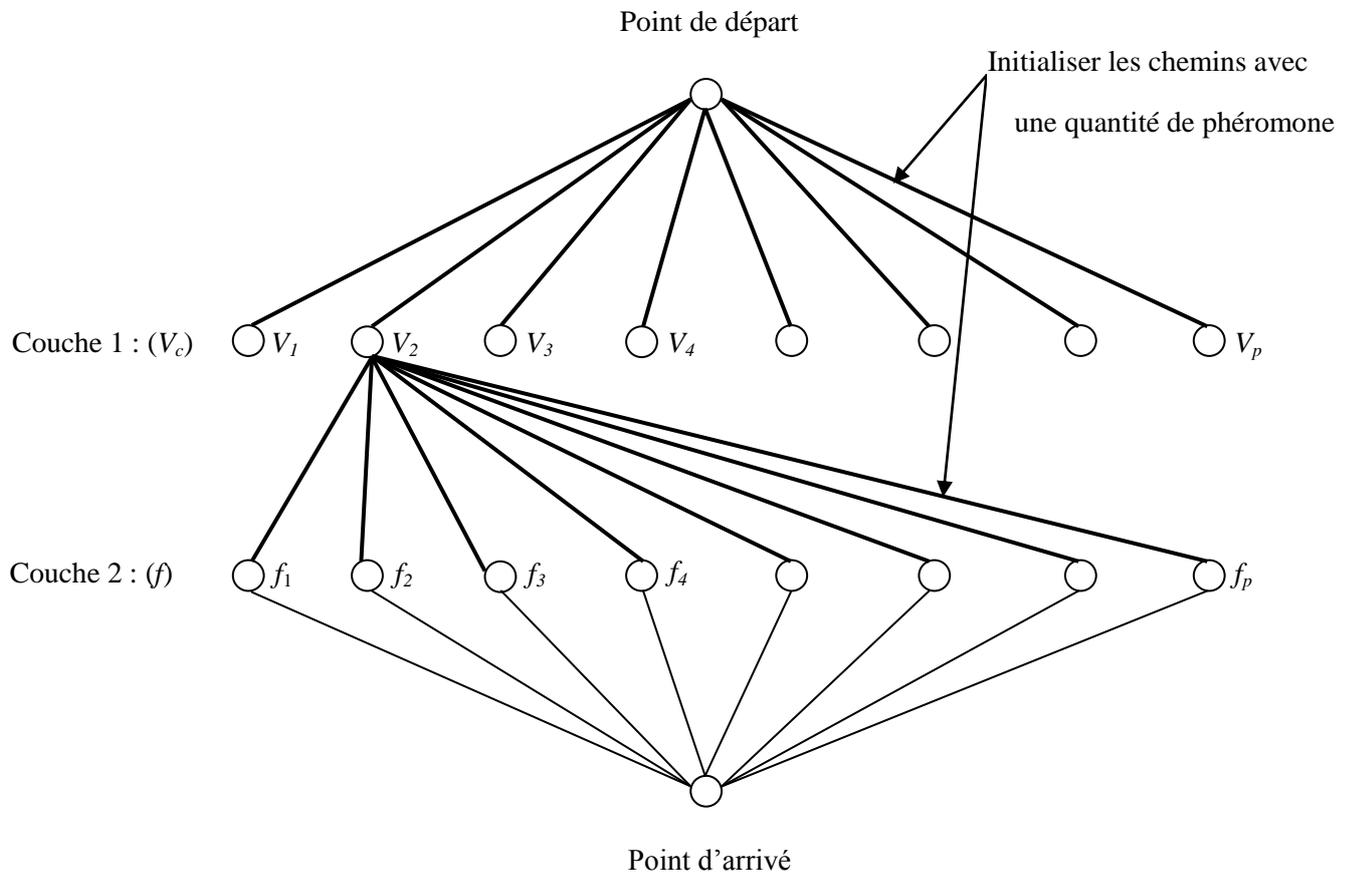


Figure IV.1 - Représentation graphique du processus ACO sous la forme d'un réseau multicouche au début du processus [28]

❖ **Etape 2**

- Calculer la probabilité (IV-1) pour choisir une seule valeur des variables (V_c, f) [28]:

$$P_{ij} = \frac{\tau_{ij}^\alpha}{\sum_{l=1}^p \tau_{il}^\alpha} \tag{IV-1}$$

$i=1,2,\dots,n$ (Nombres de variables)

$i=1,2$ (V_c, f)

$j=1,2,\dots,p$ (Les valeurs de variables)

2. Spécifier Le chemin (ou les valeurs de variables) choisi par la $k^{\text{ème}}$ fourmi qui peut être déterminée en utilisant des nombres aléatoires générés dans l'intervalle (0, 1). Pour cela, nous trouvons les intervalles de probabilité associées avec différents chemins (figure (IV.1)) basés sur les probabilités données par (IV-1). Le chemin spécifique choisi par la fourmi k sera déterminée en utilisant le processus de sélection roulette – wheel, dans l' (Etape 3-1).

❖ Etape 3

1. Générer m nombres aléatoires r_1, r_2, \dots, r_m dans l'intervalle (0,1), un pour chaque fourmi. Déterminer la valeur ou le chemin supposé par la fourmi $k^{\text{ème}}$ pour la variable i pour laquelle l'intervalle de probabilité [trouvé à l' (Etape 2-.2)] inclut la valeur r_i .
2. Répéter l' (Etape 3-1) pour tous les variables $i=1,2$.
3. Evaluer la fonction- objectif (IV-2), correspondant aux chemins complets (des valeurs de x_{ij} choisis pour toutes les variables) [28].

$$i=1,2,\dots,n$$

$$j=1,2,\dots,p$$

$$k=1,2,\dots,m$$

$$T_u = 96,936 \cdot V_c^{-1} \cdot f^{-1} + 3,4716 \cdot 10^{-8} \cdot V_c^3 \cdot f^{0,16} \cdot d_c^{1,4} + 1,63 \quad (\text{IV-2})$$

T_u : La fonction- objectif (Tu : Temps d'usinage).

4. Déterminer $T_{u \text{ best}}$ et $T_{u \text{ worst}}$ de la fonction- objectif choisie par les différentes fourmis [28].

$$T_{u \text{ best}} = \min_{k=1,2,\dots,m\{T_u\}} \quad (\text{IV-3})$$

$$T_{u \text{ worst}} = \max_{k=1,2,\dots,m\{T_u\}} \quad (\text{IV-4})$$

$T_{u \text{ best}}$: La meilleure valeur de la fonction-objectif parmi les solutions prises par les k fourmis.

$T_{u \text{ worst}}$: La mauvaise valeur de la fonction- objectif parmi les solutions prises par les k fourmis.

❖ **Etape 4**

Nous supposons que toutes les fourmis retournent au point de départ et recommencent la recherche des nouvelles solutions [28].

1. Mise à jour des phéromones sur les différents chemins (solutions) figure (IV.2), comme suivant [28]:

$$\tau_{ij}^{(it)} = \tau_{ij}^{(old)} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (IV-5)$$

2. Les solutions mauvaises seront évaporées par [28]:

$$\tau_{ij}^{(it)} = \tau_{ij}^{(old)} = (1 - \rho)\tau_{ij}^{(it-1)} \quad (IV-6)$$

$\tau_{ij}^{(old)}$: Quantité de phéromone déposée à l'itération précédente.

$\rho \in [0, 1[$: Est un coefficient qui définira la vitesse d'évaporation des phéromones.

$\Delta\tau_{ij}^{(k)}$: Quantité de phéromone déposée sur les chemins (i, j) par la meilleure fourmi k .

$$\Delta\tau_{ij}^{(k)} = \frac{Q}{L_k} \quad (IV-7)$$

Et nous appliquons (IV-7) comme suit [28] :

$$\Delta\tau_{ij}^{(k)} = \begin{cases} \frac{\zeta T_{u\ best}}{T_{u\ worst}} ; & \text{si } (i, j) \in \text{la meilleure solution globale} \\ 0 & \text{si non} \end{cases} \quad (IV-8)$$

ζ : C'est un paramètre utilisé pour contrôler l'échelle de mise à jour globale de phéromone.

3. Pour les nouvelles valeurs de $\tau_{ij}^{(it)}$, aller à l' (Etape 2).

4. Répéter les (Etapes 2, 3 et 4) jusqu'à la convergence du processus, (toutes les fourmis prennent le même chemin (même solution)).
5. L'itération du processus s'arrête si le critère d'arrêt est satisfaisant [28].

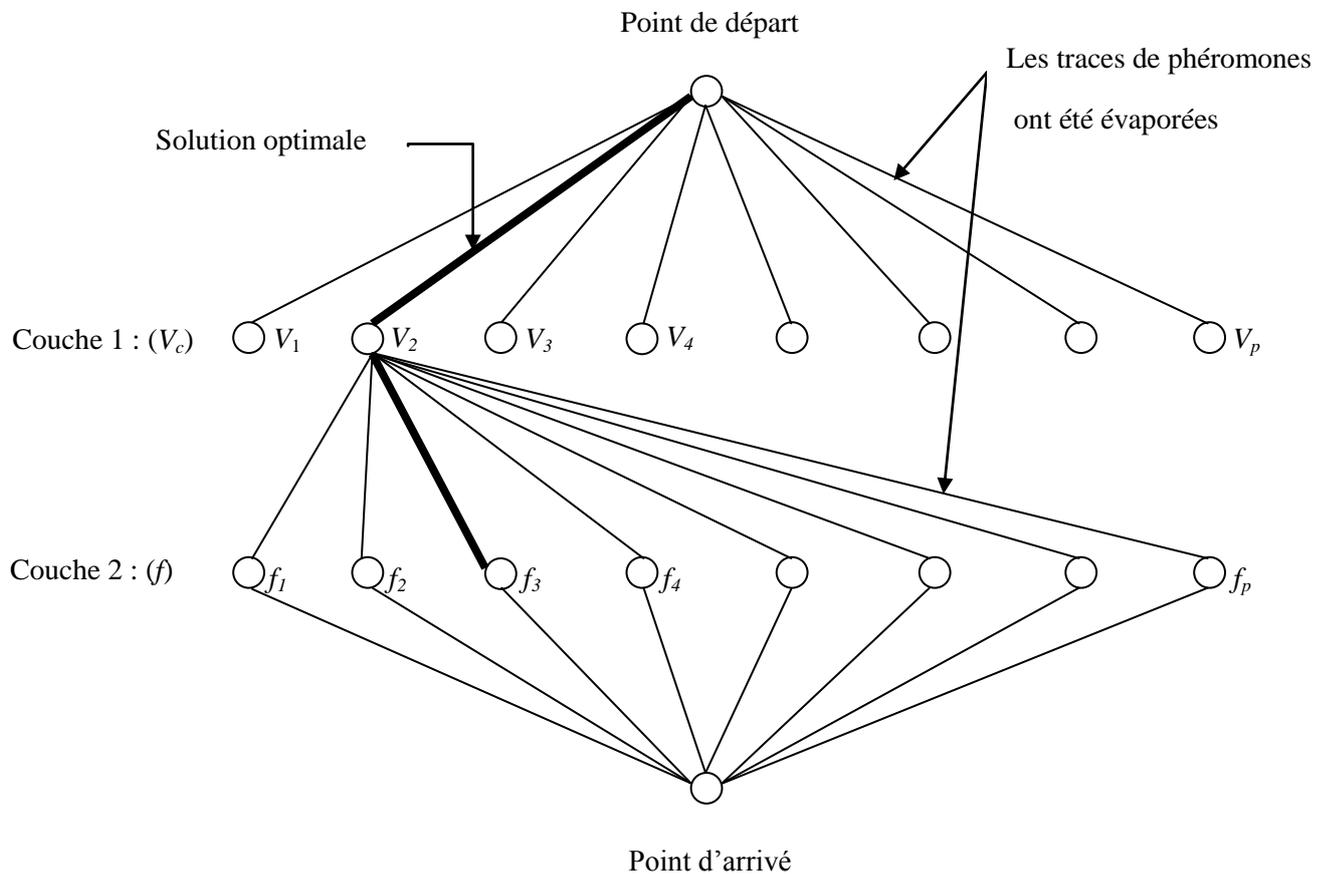


Figure IV.2 - Représentation graphique du processus ACO sous la forme d'un réseau multicouche à la fin du processus [28]

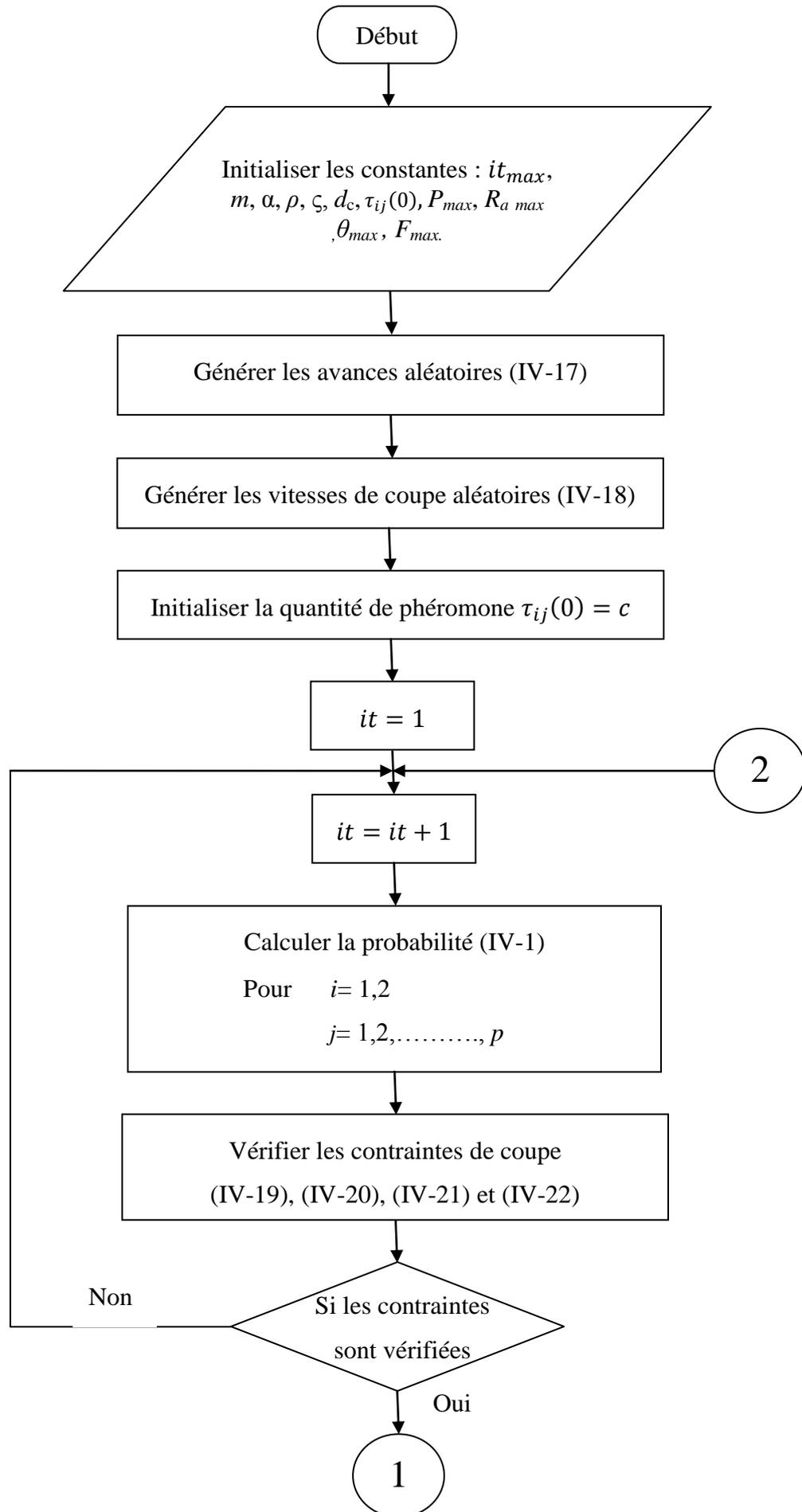
❖ Remarque

La mise à jour des phéromones pour les meilleures solutions est donnée par la formule [28] :

$$\tau_{ij}^{(it)} = \tau_{ij}^{(old)} + \sum_{k=1}^m \Delta\tau_{ij}^k$$

Où

$\tau_{ij}^{(old)}$: Quantité de phéromone déposée à l'itération qui doit conserver la même valeur, et qui ne sera pas influencée par l'évaporation.



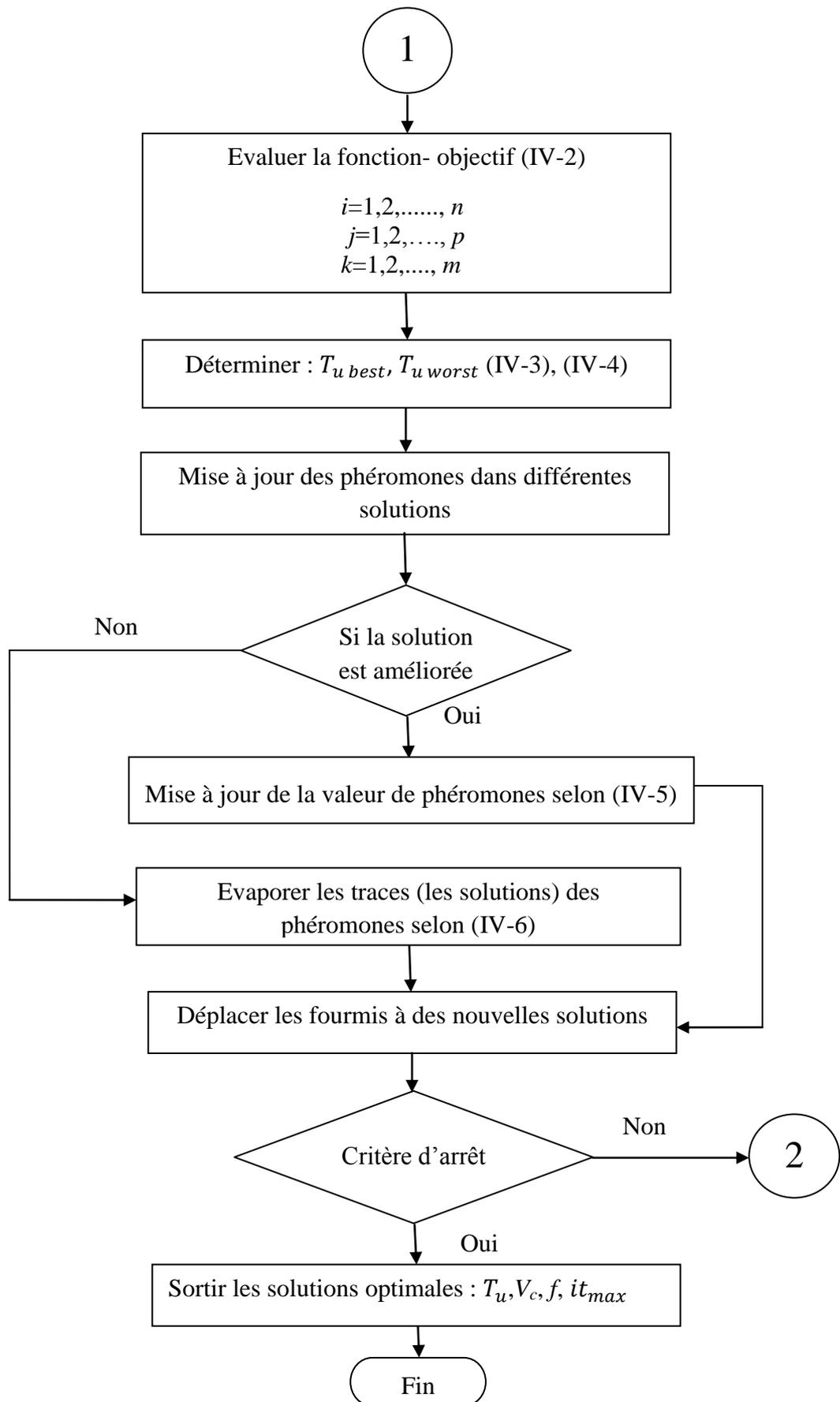


Figure IV.3 - Organigramme de l'algorithme colonie de fourmis

IV.2.3. Fonction-objective

La fonction- objectif est donnée sous la forme suivante [12] :

$$T_u = A.V_c^{-1}.f^{-1} + A.V_c^{(1-a_3)/a_3}.f^{(a_1-a_3)/a_3}.d_c^{(a_2/a_3)}.K^{(-1/a_3)}.t_{cs} + t_R + t_h \quad (IV-9)$$

IV.2.4. Contraintes d'optimisation

Dans ce travail, nous considérons les limitations suivantes [12] :

$$f_{min} \leq f \leq f_{max} \quad [mm/t_r] \quad (IV-10)$$

$$V_{cmin} \leq V_c \leq V_{cmax} \quad [m/min] \quad (IV-11)$$

$$d_{cmin} \leq d_c \leq d_{cmax} \quad [mm] \quad (IV-12)$$

$$0.0373.V_c^{0.91}.f^{0.78}.d_c^{0.75} \leq P_{max} \quad [kW] \quad (IV-13)$$

$$14.785.V_c^{-1.52}.f^{1.004}.d_c^{0.25} \leq R_{a_{max}} \quad [\mu m] \quad (IV-14)$$

$$74.96.V_c^{0.4}.f^{0.2}.d_c^{0.105} - 17.8 \leq \theta_{max} \quad [^\circ C] \quad (IV-15)$$

$$844.V_c^{-0.1013}.f^{0.725}.d_c^{0.75} \leq F_{max} \quad [N] \quad (IV-16)$$

IV.2.5. Application

Les paramètres utilisés pour l'application numérique sont mentionnés dans le tableau (IV.1) pour une opération de chariotage.

$$d_c = 1.27mm.$$

Paramètre	Val	Paramètre	Val	Paramètre	Val	Paramètre	Val
L [mm]	203	$f_{max}[mm/t_r]$	0.762	$t_R[min/passe]$	0.13	a_3	0.25
D [mm]	152	$SF_{max}[\mu m]$	2	$t_h[min/pièce]$	1.5	K	193.3
$V_{c\ min}[m/min]$	30	$SR_{max}[\mu m]$	8	$\theta_{max}[°C]$	500	$t_{cs}[min/arête]$	0.5
$V_{c\ max}[m/min]$	200	$P_{max}[kW]$	5	a_1	0.29	$C_0 [$/min]$	0.1
$f_{min}[mm/t_r]$	0.254	$F_{max}[N]$	1100	a_2	0.35	$C_t [$/arête]$	0.5

Tableau IV.1 - Paramètres d'usinage [12]

Le choix des paramètres du programme (ACO) sont mentionnés dans le tableau (IV.2).

Méthode	Nombres des fourmis	Coefficient d'évaporation	Phéromone initiale	Paramètre de contrôle	Paramètre de contrôle
	k	(ρ)	$\tau_{ij}(0) = c$	α	ζ
ACO	20	0.5	0.1	1	2
ACO	40	0.5	0.1	1	2
ACO	60	0.5	0.1	1	2
ACO	100	0.5	0.1	1	2
ACO	140	0.5	0.1	1	2
ACO	160	0.5	0.1	1	2

Tableau IV.2 - Paramètres de programme

Remplaçons maintenant les données du tableau (IV.1) dans les équations (IV-9), (IV-10), (IV-11), (IV-13), (IV-14), (IV-15) et (IV-16).

Le problème d'optimisation des conditions de coupe prend cette forme:

$$\text{On a } A = \frac{\pi.D.L}{1000} = \frac{152 \times 203 \times \pi}{1000} = 96,936$$

$$\text{minimiser } T_u = 96,936 \cdot V_c^{-1} \cdot f^{-1} + 3,4716 \cdot 10^{-8} \cdot V_c^3 \cdot f^{0,16} \cdot d_c^{1,4} + 1,63 \quad (\text{IV-2})$$

$$\text{Telque } 0,254 \leq f \leq 0,762 \quad (\text{IV-17})$$

$$30 \leq V_c \leq 200 \quad (\text{IV-18})$$

$$0,0373 \cdot V_c^{0,91} \cdot f^{0,78} \cdot d_c^{0,75} \leq 5 \quad (\text{IV-19})$$

$$14,785 \cdot V_c^{-1,52} \cdot f^{1,004} \cdot d_c^{0,25} \leq 2 \quad (\text{IV-20})$$

$$74,96 \cdot V_c^{0,4} \cdot f^{0,2} \cdot d_c^{0,105} - 17,8 \leq 500 \quad (\text{IV-21})$$

$$844 \cdot V_c^{-0,1013} \cdot f^{0,725} \cdot d_c^{0,75} \leq 1100 \quad (\text{IV-22})$$

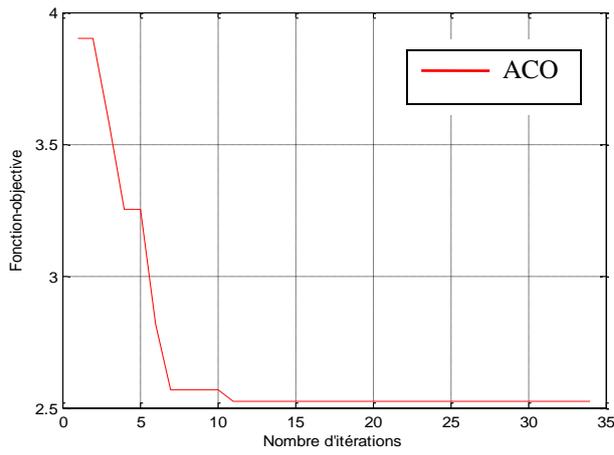
IV.3. Résultats

Les résultats trouvés en utilisant (ACO) sont représenté dans le tableau (IV.3).

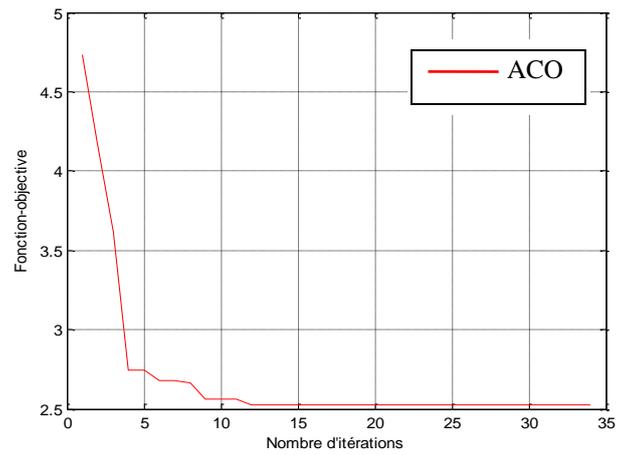
Nbr des fourmis	d_c (mm)	V_c (m/min)	f (mm/tr)	Tu (min)	Nbr d'itérations
20	1.27	159.81	0.663	2.52	11
40	1.27	159.81	0.663	2.52	12
60	1.27	168.69	0.663	2.48	13
100	1.27	168.69	0.663	2.48	10
140	1.27	168.69	0.663	2.48	09
160	1.27	168.69	0.663	2.48	08

Tableau IV.3 - Résultats trouvés par le programme ACO

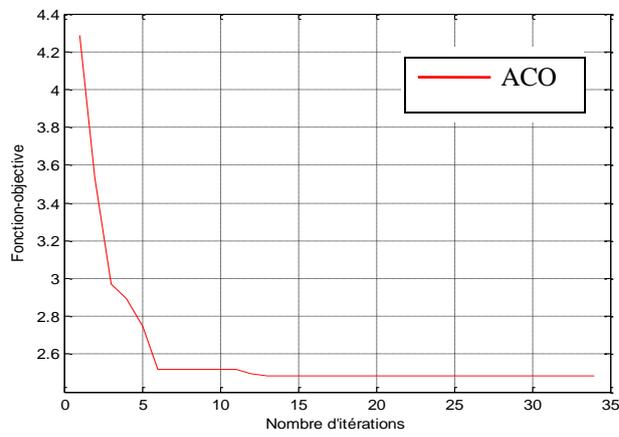
A partir du tableau (IV.3), nous avons tracé des courbes des valeurs de la fonction-objectif au cours de l'évolution d'itérations pour le programme ACO, figure (IV.4).



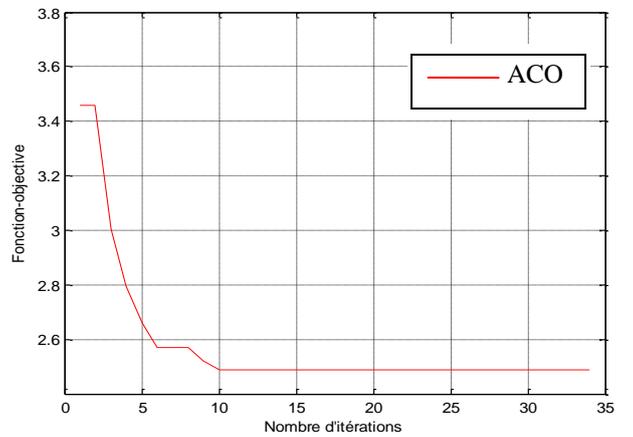
Fourmis = 20



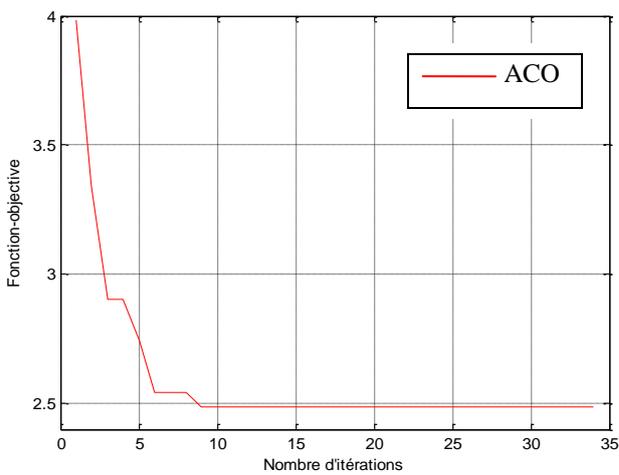
Fourmis = 40



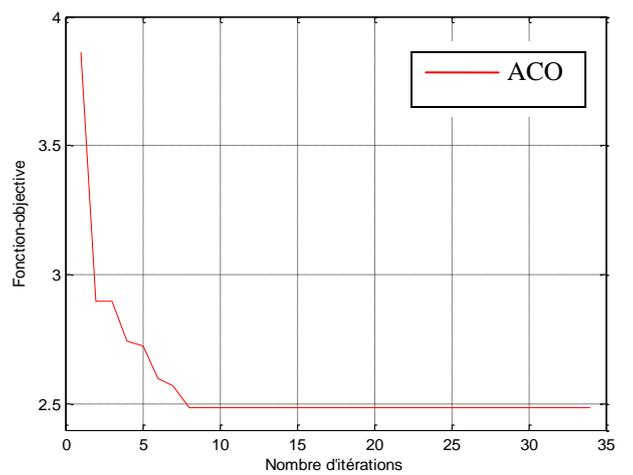
Fourmis = 60



Fourmis = 100



Fourmis = 140



Fourmis = 160

Figure IV.4 – Les graphes de la fonction-objective en fonction du nombre d'itérations

IV.4. Discussion et interprétation des résultats

Plusieurs essais ont été effectués avec un nombre de fourmis entre 20 et 160 fourmis évoluant pendant 34 générations.

Nous remarquons que :

Si nous varions le nombre de fourmis de 20 à 40 fourmis on obtient un temps d'usinage égal à 2,52 min.

Si nous augmentons le nombre de fourmis de 60 à 160 on obtient un temps d'usinage minimal stable égal à 2,48 min avec un nombre d'itérations minimales égales à 08 itérations.

Les résultats de l'approche proposée (ACO) ont été comparés à trois méthodes (GM [12], GA [6], PSO [7]).

Les résultats obtenus à partir des quatre techniques sont donnés dans le tableau ci-dessous (IV.4).

Les résultats obtenus montrent que la méthode proposée ACO surpasse de manière significative les trois autres méthodes.

De toute évidence, l'approche ACO fournit une approximation suffisante à l'optimisation de notre problème.

IV.5. Comparaison

Les résultats trouvés par la méthode du gradient [12], l'algorithme à essaim de particules [7], l'algorithme génétique [6], et ceux de l'algorithme colonie de fourmis pour une passe de chariotage sont mentionnées respectivement dans le tableau (IV.4).

NU	Algorithme	d_c (mm)	V_c (m/min)	f (mm/tr)	Tu (min)	Nombres d'itérations
1	gradient [12]	1,27	173	0,71	2,65	/
2	PSO [7]	1,27	124,02	0,762	2,744	20
3	génétique [6]	1,27	132,55	0,762	2,69	/
4	ACO	1,27	168.69	0,663	2,48	08

Tableau IV.4 - Comparaison des résultats

La présente approche nous a permis d'obtenir un temps d'usinage inférieur caractérisé par un pourcentage représentant une diminution de 6,85% en le comparant avec GM, de 10,64% par rapport au PSO et atteignant les 8,46% pour le GA.

De plus le nombre d'itérations, dans la présente approche, a diminué par rapport à la méthode de PSO.

CONCLUSION

CONCLUSION

Dans cette étude, nous avons travaillé aux frontières de deux domaines très différents qui sont l'optimisation des conditions de coupe et les algorithmes évolutionnaires.

Une technique d'optimisation non-conventionnelle (ACO) a été étudiée pour l'optimisation des paramètres d'usinage en tournage à passe unique. Les fourmis en colonie présentent, en effet, des comportements auto-organisés, où des interactions simples au niveau local permettent l'émergence d'un comportement global complexe.

En utilisant la fonction-objectif proposée par J.S. Agapiou avec un ensemble de contraintes pour résoudre le problème d'optimisation d'usinage, l'algorithme ACO a été mis en œuvre pour l'optimisation de l'opération de tournage.

Nos résultats de calcul pour un même cas d'application montrent une nette diminution du temps d'usinage en comparaison avec (GM, GA, et PSO) et aussi un faible nombre d'itérations par rapport au nombre d'itérations réalisées par l'algorithme essaim de particules.

La présente approche nous a permis d'obtenir un temps d'usinage inférieur d'un pourcentage représentant une diminution de 6,85% en le comparant avec celui obtenu par le GM. Par ailleurs les pourcentages de diminution atteignent respectivement les 10,64%, et 8,46% en comparaison avec les temps obtenus par le PSO et le GA.

De plus, le nombre d'itérations a diminué par rapport à la méthode de PSO de 20 à 8 itérations.

D'après les résultats obtenus nous concluons que, parmi les quatre méthodes appliquées dans ce travail, l'ACO s'avère la méthode qui surpasse toutes les autres méthodes en ce qui concerne le temps d'usinage et le temps de calcul.

RECOMMANDATIONS

Suite aux résultats obtenus, nous recommandons les travaux suivants :

- Une étude concernant la profondeur dans l'opération de tournage multi passe.
- Une étude mettant en œuvre plusieurs critères tels que, le temps, le coût d'usinage.

REFERENCES

BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

- [1] P. Siarry. Optimisation en traitement du signal et de l'image. Lavoisier (2007).
- [2] J. Dréo. Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues, application en génie biomédical. Thèse de doctorat. Université de Paris 12 - Val de Marne (2004).
- [3] A. Ben Mohamed. Résolution approchée du problème de bin-packing. Thèse de doctorat. Université du Havre (2009).
- [4] M. Tien Tho Do. Optimisation de forme en forgeage 3d. Thèse de doctorat. École des Mines de Paris (2006).
- [5] W. Tfaili. Conception d'un algorithme de colonie de fourmis pour l'optimisation continue dynamique. Thèse de doctorat. Université de Paris 12 - Val de Marne (2007).
- [6] M. Djenane. Optimisation des régimes de coupe a l'aide d'un critère combiné par la méthode du simplexe et la méthode des algorithmes génétiques. Mémoire de magistère. Université Hadj Lakhdar- Batna (2001).
- [7] T. Ameer. Optimisation des conditions de coupe en utilisant le calcule évolutionnaire. Mémoire de magistère. Université Mohamed Khider - Biskra (2005).
- [8] L. Drdi. Les algorithmes génétiques. Thèse de doctorat (2005).
- [9] M. Assas, M. Djenane. Optimisation des Conditions de Coupe Basée sur Critère Combiné par la Méthode des Algorithmes Génétiques. 16^{ème} Congrès Français de Mécanique, Nice, 1-5 septembre (2003).
- [10] M. Assas, M. Djenane. Optimisation des conditions d'usinage basée sur un critère combiné. Journées de Mécanique EMP. Alger. 23/24 Décembre (2001).
- [11] A. Touil, A. Benderradji. Optimisation des conditions d'usinage en fraisage. Mémoire fin d'étude. Université hadj lakhdar- Batna (2011).

- [12] J. S. Agapiou. The optimization of machining operations based on a combined criterion, Part 1 : The use of combined objectives in single-pass operations. *Journal of Engineering for Industry*. Vol 114. pp 500-507. November (1992).
- [13] F. Cus, J. Balic. Optimization of cutting process by GA approach. *Robotics and Computer Integrated Manufacturing* 19 ,113–121, (2003).
- [14] R. Q. Sardiñas, M. R. Santana, E. A. Brindis. Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence* 19, 127–133, (2006).
- [15] N. Yusup, A. M. Zain, S. Z. M. Hashim. Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007–2011). *International Journal Science Direct Expert Systems with Applications* 39 ,9909–9927 , (2012).
- [16] L. Gao, J. Huang, X. Li. An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process. *Applied Soft Computing* 12 ,3490–3499, (2012).
- [17] O. Roux. La mémoire dans les algorithmes à colonie de fourmis : Applications à l'optimisation et à la programmation automatique. Thèse de doctorat. Université du Littoral Côte d'Opale (2001).
- [18] S. Hernane, K. Belkadi. Deux stratégies parallèles de l'optimisation par colonie de fourmis. 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications - Tunisia (2007).
- [19] P. Nicolas, F. Saubion, I. Stéphan. Optimisation par colonies de fourmis pour la programmation logique étendue. Université d'angers 2, bd Lavoisier, 49045 Angers cedex 01.
- [20] Ch. E. Bichot, J. M. Alliot. Optimisation par colonies de fourmis appliquées au découpage de l'espace aérien européen en zones de qualification. 7, avenue Edouard Belin 31055 Toulouse cedex, France.
- [21] K. Vijayakumar, G. Prabhakaran, P. Asokan, R. Saravanan. Optimization of multi-pass turning operations using ant colony system. *International Journal of Machine Tools and Manufacture*, vol 43, no 15, pp 1633-1639. (2003).

- [22] J. Wu, Y. Yao. A modified ant colony system for the selection of machining parameters. Seventh International Conference on Grid and Cooperative Computing, pp89-94, DOI 10.1109/GCC.64 (2008)
- [23] F. Cūs, J. Balic, U. Züperl. Hybrid ANFIS-ants system based optimization of turning parameters. Journal of Achievements in Materials and Manufacturing Engineering, vol 36, no 1, pp 79-86 (2009).
- [24] V. C. Preda, Dj. Tali-Maamar. Optimisation par colonies de fourmis. (2004).
- [25] C. Andrea, L. Thé Van, M. Guillaume. Optimisation par colonies de fourmis. (2006).
- [26] Ch. Bettayeb, A. Guerrah. Optimisation non linéaire des conditions de coupe .Mémoire Fin d'étude. Université Kasdi Merbah - Ouargla (2009).
- [27] A. Debbache, M. B. Benglia. Optimisation multi-passe des régimes de coupe en tournage. Mémoire de master. Université Kasdi Merbah Ouargla (2011).
- [28] J. Wiley, S. Inc. Engineering optimization: Theory and practice. Fourth Edition (2009).

Résumé

Cette étude présente une nouvelle technique d'optimisation évolutionnaire, basée sur l'algorithme colonie de fourmis (ACO), afin de résoudre le problème d'optimisation des conditions de coupe dans l'opération de chariotage (tournage). Les paramètres d'usinage (vitesse de coupe, avance et profondeur de passe) sont déterminés en minimisant le temps de production soumis à diverses contraintes d'usinage. Le modèle mathématique utilisé par J.S. Agapiou est adopté dans ce travail pour l'optimisation des paramètres d'usinage. D'après les résultats obtenus, il est clair que l'approche proposée colonie de fourmis surpasse de manière significative les autres méthodes, telles que GA, GM et PSO.

Abstract

This study presents a new optimization evolutionary technique based on ant colony algorithm (ACO) to solve the problem of optimization of cutting conditions in the roughing operation (turning). The machining parameters (cutting speed, feed rate and depth of cut) are determined by minimizing the production time subjected to various processing constraints. The mathematical model used by J. S. Agapiou is adopted in this work for the optimization of machining parameters. From the results, it is clear that the proposed ant colony approach significantly outperforms other methods such as GA, GM and PSO.

ملخص :

في هذا العمل استعملنا تقنية جديدة متطورة تركز على خوارزمية مستعمرة النمل و ذلك لإيجاد الشروط المثالية لتشغيل مرحلة من مراحل الخراطة التي تضمن أقل وقت ممكن لانجازها وذلك تحت مجموعة من العوائق التي تمثل أهم الشروط الواجب إشباعها .

النموذج الرياضي المستعمل من قبل J. S. Agapiou اعتمدنا عليه في عملنا هذا لحل هذه المسألة. النتائج المتحصل عليها أظهرت أن طريقة مستعمرة النمل أبدت فعالية عالية في تحسين وقت انجاز مرحلة الخراطة مقارنة بطرق أخرى مثل GA, MG, PSO