

UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la
Communication
Département d'Informatique et Technologies de l'Information



Mémoire
MASTER ACADEMIQUE
Domaine: Mathématiques et Informatique
Filière: Informatique
Option: Gestion Et Sécurité Des Réseaux

Présenté par
ELHELLI Imane

Thème

**une approche basée sur l'algorithme PSO pour
le conception de VLAN dans les grandes
infrastructures**

soutenue le 15/06/2025

Khelili Khalida Farida	MAA	Président
Mesied Abdeslam	MCB	Examineur
BEKKARI Fouad	MAA	Rapporteur

Année universitaire: 2024/2025

Remerciements et Dédicaces

*À moi-même qui ai persévéré et, par la grâce de Dieu, ai atteint ce
moment.*

*À ma chère mère, mon soutien et ma force, et source de mon inspiration et
de mes prières.*

*À mon honorable professeur encadrant, tous mes remerciements pour vos
précieux conseils et votre soutien.*

*Et à ma chère amie, ma compagne de route et d'âme, ma gratitude
éternelle pour ta présence à mes côtés*

ElhelliImane

Résumé:

Dans un contexte de croissance rapide des infrastructures réseau, la segmentation logique des réseaux via les VLANs (Virtual Local Area Networks) est devenue essentielle pour améliorer la sécurité, la performance et la gestion des systèmes informatiques. Cependant, la conception optimale des VLANs dans des environnements complexes constitue un problème d'optimisation difficile. Ce mémoire propose une approche basée sur l'algorithme ParticleSwarmOptimization (PSO), une métaheuristique inspirée du comportement collectif des essaims, pour résoudre le problème de partitionnement des VLANs. Après une présentation détaillée des concepts fondamentaux des VLANs et des algorithmes métaheuristiques, l'étude modélise le problème de segmentation sous forme matricielle et adapte l'algorithme PSO pour produire des solutions optimales.

Une application a été développée en Python pour implémenter et visualiser les résultats de cette approche. Les résultats expérimentaux montrent l'efficacité de PSO en termes de réduction du nombre de VLANs et d'optimisation des communications inter-VLAN, tout en maintenant une bonne évolutivité du réseau.

Mots-clés : VLAN, PSO, Optimisation, Métaheuristiques, Segmentation réseau, Sécurité, Réseaux Informatiques.

المخلص :

في ظل التوسع الكبير في شبكات المؤسسات، أصبحت عملية تقسيم الشبكة إلى شبكات محلية افتراضية (VLANs) أمرًا ضروريًا لتعزيز الأمان وتحسين الأداء وتبسيط إدارة البنية التحتية الرقمية. غير أن تحقيق التقسيم الأمثل لهذه الشبكات يُعد مشكلة صعبة تتطلب حلولاً ذكية

تقترح هذه المذكرة استخدام خوارزمية التحسين بواسطة سرب الجسيمات (PSO)، وهي خوارزمية ميتاهيوريستية مستوحاة من سلوك أسراب الطيور، لحل مشكلة تقسيم الشبكة إلى VLANs بطريقة مثلى. بعد عرض شامل لمفاهيم VLAN والخوارزميات الميتاهيوريستية، تم تحويل المشكلة إلى نموذج رياضي باستخدام المصفوفات، ثم تم تكييف خوارزمية PSO لتوليد حلول فعالة.

تم تطوير تطبيق بلغة Python لتنفيذ ومتابعة العملية وتحليل النتائج. وقد أظهرت التجارب فعالية الخوارزمية في تقليل عدد VLANs وتحسين التواصل بين أجزاء الشبكة مع الحفاظ على سهولة التطوير والتوسع.

الكلمات المفتاحية: الشبكات المحلية الافتراضية، التحسين بواسطة سرب الجسيمات، الميتاهيوريستيك، التقسيم الشبكي، الأمن المعلوماتي، الأمثلية.

Table des matières:

INTRODUCTION GENERALE:	1
CHAPITRE 1 : LES RESEAUX VIRTUELS VLAN	4
1. INTRODUCTION:	4
2. DEFINITION DES RESEAUX ET LEURS TYPES	4
3. RESEAUX LOCAUX VIRTUELS (VLAN) : CONCEPTS, TYPES ET OBJECTIFS	6
4. LIMITES DES INFRASTRUCTURES TRADITIONNELLES ET IMPORTANCE DES VLAN	7
5. MECANISMES DE CONTROLE D'ACCES DANS LES VLAN ET LEUR IMPORTANCE	8
6. CONCLUSION:	9
CHAPITRE 2: INTRODUCTION SUR LES PROBLEMES DIFFICILE ET LES METAHERUSTIQUES	11
1. INTRODUCTION:	11
2. PROBLEMES D'OPTIMISATION COMBINATOIRE:	11
3. LES METAHEURISTIQUES	11
4. LES METHODES BASEES SUR UNE SEULE SOLUTION	14
5. LES METHODES BASEES SUR UNE POPULATION DES SOLUTIONS	19
6. CONCLUSION	28
CHAPITRE 3 : CONCEPTION D'UNE SOLUTION BASEE SUR L'ALGORITHME PSO POUR LE DESIGN DE VLAN	30
1. INTRODUCTION	30
2. MODELISATION DU PROBLEME DE PARTITIONNEMENT DES VLANs	30
3. JUSTIFICATION DU CHOIX DE PSO	30
4. ADAPTATION DE L'ALGORITHME PSO AU PROBLEME DE VLAN	30
5. COMPLEXITE ALGORITHMIQUE	32
6. CONCLUSION :	33
CHAPITRE 4 : L' IMPLEMENTATION	35
1. INTRODUCTION	35
2. ENVIRONNEMENT DE DEVELOPPEMENT ET OUTILS UTILISES	35
3. DEMARCHE:	37
4. INTERFACE DE APPLICATION	38
5. RESUME	43
6. CONCLUSION :	43
CONCLUSION GENERALE:	46
BIBLIOGRAPHIE	46

Listedes figures

FIGURE 1 RESEAU INFORMATIQUE	4
FIGURE 2 TYPES DE RESEAUX INFORMATIQUES	5
FIGURE 3 TYPES DE RESEAUX INFORMATIQUES	5
FIGURE 4 L'ALGORITHME DE DESCENTE NE PEUT PAS « REMONTER » POUR SORTIR D'UN OPTIMUM LOCAL	15
FIGURE 5 PRINCIPE DEMARCHE DES ALGORITHMES GENETIQUES	21
FIGURE 6 PRINCIPE DE L'OPERATEUR DE CROISEMENT	23
FIGURE 7 PRINCIPE DE FONCTIONNEMENT DE L'OPERATEUR DE MUTATION	24
FIGURE 8 PRINCIPE DEMARCHE LE PROCESSUS	25
FIGURE 9 PSEUDO CODE DE LA METHODE PSO	26
FIGURE 10 EXEMPLES DE TOPOLOGIES DE VOISINAGE POUR PSO	27
FIGURE 11 VECTEUR DE LONGUEUR N CORRESPOND VLAN	31
FIGURE 12 MATRICE Z DE LA FONCTION FITNESS	31
FIGURE 13 PSEUDOCODE DE L'ALGORITHME ADAPTE	32
FIGURE 14 INTERFACE PRINCIPALE	38
FIGURE 15 PARAMETRES DE L'ALGORITHME	38
FIGURE 16 LANCEMENT DE L'OPTIMISATION	39
FIGURE 17 SUIVI DE L'OPTIMISATION	39
FIGURE 18 RESULTATS FINAUX	40
FIGURE 19 GENERATION DU RAPPORT	41
FIGURE 20 RAPPORT D'OPTIMISATION DU PARTITIONNEMENT VLAN	42
FIGURE 21 TOPOLOGIE DU RESEAU AFFECTEE AUX VLAN	42
FIGURE 22 AFFECTATION DES VLAN AUX APPARIELS	43
FIGURE 23 CONFIGURATION VLAN SUGGEREE	43

Introduction générale:

Dans un monde en constante évolution numérique, les réseaux informatiques jouent un rôle fondamental dans le bon fonctionnement des entreprises, des institutions publiques et des infrastructures critiques. Cependant, avec l'augmentation exponentielle des équipements connectés et la complexité croissante des systèmes d'information, la gestion efficace des réseaux devient un enjeu majeur.

Les Réseaux Locaux Virtuels (VLANs) se présentent comme une solution puissante pour améliorer la sécurité, la performance et la flexibilité des architectures réseau. Grâce à la segmentation logique qu'ils offrent, les VLANs permettent de regrouper des utilisateurs selon des critères fonctionnels plutôt que physiques, tout en réduisant le trafic inutile et en simplifiant l'administration.

Cependant, la mise en place de VLANs efficaces dans de grandes infrastructures réseau pose des problèmes complexes de partitionnement, souvent assimilables à des problèmes NP-difficiles. Ces problèmes nécessitent des méthodes d'optimisation avancées pour aboutir à des solutions viables dans un temps raisonnable.

C'est dans ce contexte que les algorithmes métaheuristiques, en particulier l'algorithme d'optimisation par essaim de particules (OEP) ou PSO, le PSO est conçu pour explorer et exploiter de manière efficace les espaces de recherche et éviter les optima locaux.

Problématique :

Comment segmenter un réseau informatique complexe en un nombre optimal de VLANs, tout en minimisant les coûts de communication inter-VLAN et en respectant les contraintes de taille et de sécurité ?

Objectif de l'étude :

Cette étude vise à concevoir et implémenter une approche basée sur l'algorithme PSO pour optimiser la création de VLANs dans de grandes infrastructures, en évaluant ses performances par rapport aux méthodes traditionnelles.

Organisation du travail

Dans le chapitre 1 on va introduire les notions de base liées au VLANs, la définition, les types des VLAN et Mécanismes de contrôle d'accès dans les VLAN et leur importance

Dans le chapitre 2 on va introduire les notions de base des metaheuristique leur définition les types des metaheuristiques ou on va donner un aperçu sur l'optimisation par essime de particules PSO

Le chapitre 3 consacré pour la conception le l'approche proposée, la définition d'une solution la modélisation des opérations la spécification de la fonction objective

Le chapitre 4 et pour les choix techniques et l'implémentation des lustrations représentent les interfaces de l'application et des results optenus

Chapitre 1 : Les réseaux virtuels VLAN

Chapitre I : Les réseaux virtuels VLAN

1. Introduction:

Dans un monde où la connectivité est devenue essentielle, les réseaux informatiques jouent un rôle central dans le fonctionnement des entreprises et des institutions. Cependant, les infrastructures réseau traditionnelles présentent des limites en termes de flexibilité, de sécurité et de gestion. Les Réseaux Locaux Virtuels (VLAN) émergent comme une solution innovante pour répondre à ces défis. Ce chapitre explore les fondements des réseaux informatiques, introduit les concepts clés des VLAN et analyse les limites des infrastructures traditionnelles pour souligner l'importance des VLAN dans les architectures réseau modernes .

2. Définition des réseaux et leurs types

2.1. Définition d'un réseau informatique:

Un réseau informatique est un ensemble d'équipements interconnectés permettant l'échange de données et de ressources. Il facilite la communication entre ordinateurs, serveurs, imprimantes et autres périphériques, offrant ainsi une infrastructure essentielle pour le partage d'informations et la collaboration. Les réseaux peuvent être classés selon leur étendue géographique, leur architecture ou leur mode de transmission. Ils sont conçus pour répondre à des besoins spécifiques, allant de la simple connexion de quelques ordinateurs à la gestion de vastes infrastructures interconnectées à l'échelle mondiale .[ABGL01]

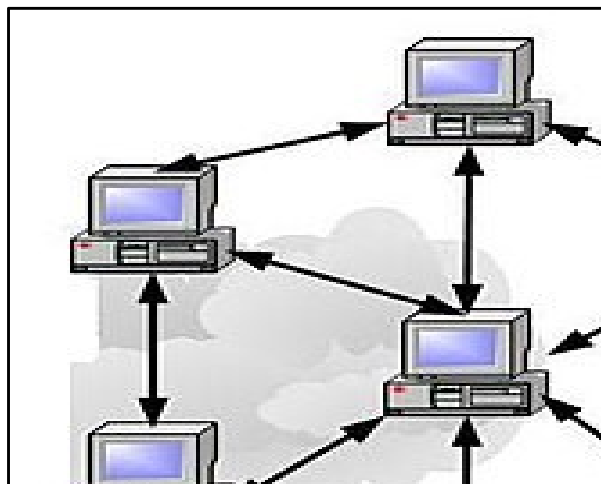


Figure 1réseau informatique

2.2. Types de réseaux informatiques

Les réseaux informatiques se classent principalement selon leur étendue géographique :

- **Réseau Local (LAN) :** Couvre une zone restreinte, comme un bureau ou un bâtiment. Il permet un haut débit de transmission et une faible latence, facilitant le partage de ressources locales.[ACE+02]
- **Réseau Métropolitain (MAN) :** S'étend sur une ville ou une agglomération, interconnectant[AD05] plusieurs LANs pour former un réseau plus vaste.
- **Réseau Étendu (WAN) :** Relie des réseaux sur de vastes zones géographiques, souvent à l'échelle nationale ou internationale. Il utilise des technologies comme MPLS ou le SD-WAN pour assurer la connectivité .

Chaque type de réseau répond à des besoins spécifiques en termes de portée, de vitesse et de connectivité .[ANT02]



Figure 2 Types de réseaux informatiques

3. Réseaux Locaux Virtuels (VLAN) : concepts, types et objectifs

3.1. Définition et concepts des VLAN

Un Réseau Local Virtuel (VLAN) est une segmentation logique d'un réseau physique, permettant de regrouper des utilisateurs ou des équipements en fonction de critères spécifiques, indépendamment de leur emplacement physique. Cette approche améliore la gestion du réseau, la sécurité et l'efficacité du trafic. Les VLANs permettent de créer des domaines de diffusion distincts, réduisant ainsi le trafic inutile et améliorant la performance globale du réseau .[AtMG02]

3.2. Types de VLAN

Les VLAN peuvent être classés selon différents critères :

VLAN statiques (par port) : Les ports du commutateur sont assignés manuellement à un VLAN spécifique. Cette méthode est simple à mettre en œuvre mais manque de flexibilité.[BEG04]

VLAN dynamiques (par adresse MAC ou IP) : L'affectation des ports aux VLAN se fait automatiquement en fonction de l'adresse MAC, du protocole ou d'autres critères. Cette méthode offre une plus grande flexibilité mais nécessite une configuration plus complexe .[AUB+05]

Cette classification permet une flexibilité dans la gestion des réseaux et une adaptation aux besoins spécifiques des organisations[CGJ+00] .

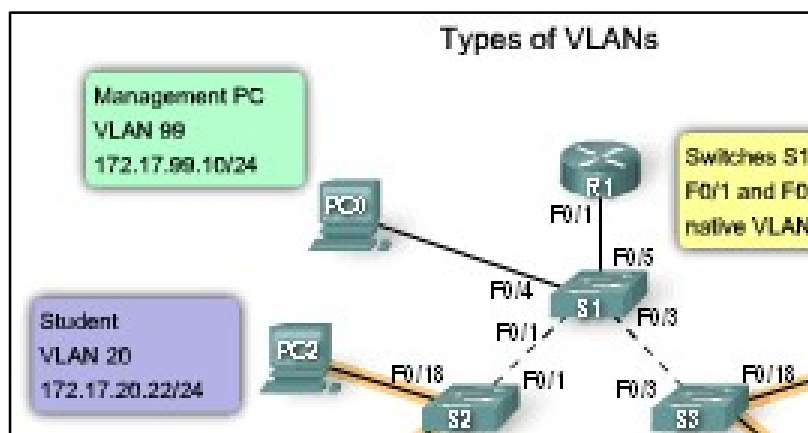


Figure 3 Types de VLAN

3.3. Objectifs des VLAN

Les VLAN visent à :

Améliorer la sécurité : En isolant les segments du réseau, les VLAN limitent la propagation des menaces et permettent une gestion plus fine des droits d'accès.[Glo95]

Optimiser la bande passante : En réduisant les domaines de diffusion, ils minimisent le trafic inutile, améliorant ainsi la performance du réseau.[KE95]

Faciliter la gestion du réseau : Les VLAN permettent une administration centralisée et une configuration simplifiée, notamment lors de changements organisationnels ou de déplacements d'utilisateurs[SHMD05] .

4. Limites des infrastructures traditionnelles et importance des VLAN

4.1. Limites des infrastructures réseau traditionnelles

Les réseaux traditionnels présentent plusieurs inconvénients :

- **Manque de flexibilité :** Toute modification nécessite souvent des interventions physiques, ce qui peut entraîner des interruptions de service.[Zam06]
- **Sécurité limitée :** L'absence de segmentation facilite la propagation des attaques et rend difficile la mise en place de politiques de sécurité granulaires.
- **Gestion complexe :** L'administration de grands réseaux devient rapidement difficile sans segmentation logique, rendant les tâches de maintenance et de configuration plus ardues .[WA04]

4.2. Importance des VLAN dans l'architecture réseau moderne

L'intégration des VLAN dans les architectures réseau modernes offre :

- **Une meilleure sécurité :** En isolant les différents segments, les VLAN limitent les accès non autorisés et permettent une application plus stricte des politiques de sécurité.[Tal99]
- **Une gestion simplifiée :** Les modifications peuvent être effectuées logiquement sans interventions physiques, réduisant ainsi les coûts et les temps d'arrêt.[SunMS05]
- **Une évolutivité accrue :** Les VLAN facilitent l'ajout de nouveaux utilisateurs ou services sans perturber l'ensemble du réseau, permettant une adaptation

rapide aux besoins changeants de l'organisation .[Mel99]

5. Mécanismes de contrôle d'accès dans les VLAN et leur importance

5.1. Définition du contrôle d'accès (Access Control)

Le contrôle d'accès désigne l'ensemble des techniques et des mécanismes permettant de réguler et de restreindre l'accès aux ressources réseau, notamment au sein des VLAN (Virtual Local Area Networks). Il s'agit d'assurer que seuls les utilisateurs et les appareils autorisés puissent communiquer ou accéder aux segments spécifiques du réseau.[LMS02]

Ce contrôle est fondamental pour la sécurité des réseaux, puisqu'il permet de prévenir les accès non autorisés, de protéger les données sensibles, et de garantir une gestion optimale des ressources réseau.

5.2. Méthodes de contrôle d'accès dans les réseaux VLAN

Plusieurs méthodes sont couramment utilisées pour implémenter le contrôle d'accès dans les VLAN:

- Port Security :

Cette technique consiste à limiter l'accès à un port de commutateur (switch) en autorisant uniquement certains appareils identifiés par leur adresse MAC. Cela permet d'éviter que des appareils non autorisés ne se connectent au réseau via un port donné.[Rai06]

Par exemple, on peut configurer un port pour accepter un nombre limité d'adresses MAC ou pour bloquer automatiquement les adresses inconnues.[Sch60]

- Listes de contrôle d'accès (ACLs) :

Les ACLs sont des règles définies pour filtrer le trafic réseau en fonction de critères précis tels que les adresses IP, les protocoles ou les ports utilisés. Elles peuvent être appliquées pour limiter ou autoriser le trafic entre VLANs ou à l'intérieur d'un même VLAN.[FKTY03]

Cette méthode est très flexible et permet une granularité fine dans la gestion des accès.[GL05]

- Authentification 802.1X :

Le protocole 802.1X offre un mécanisme d'authentification dynamique basé sur l'identification des utilisateurs ou des appareils avant leur accès au réseau. Il s'intègre généralement avec un serveur RADIUS et renforce la sécurité en empêchant l'accès aux utilisateurs non authentifiés.[DFJ54]

5.3. Importance du contrôle d'accès dans la protection du réseau et la gestion des ressources

Le contrôle d'accès est un pilier essentiel de la sécurité réseau, notamment dans les environnements VLAN, car il permet :

- De limiter les risques d'attaques et d'intrusions malveillantes, en réduisant la surface d'attaque.
- D'organiser et de réguler la circulation des données pour éviter la congestion et le trafic inutile.
- D'améliorer la performance globale du réseau par une meilleure gestion des flux.

De simplifier l'administration en facilitant la mise en place de politiques de sécurité cohérentes et efficaces.[CMRR02]

6. Conclusion:

Ce chapitre a mis en lumière l'importance des VLAN dans la modernisation des infrastructures réseau. En répondant aux limitations des architectures traditionnelles, les VLAN offrent une solution efficace pour améliorer la sécurité, la flexibilité et la gestion des réseaux. La compréhension de ces concepts est essentielle pour toute organisation souhaitant optimiser son infrastructure informatique.

Chapitre 2 : Introduction sur les problèmes difficiles et les métaheuristiques

Chapitre 2: introduction sur les problèmes difficiles et les métaheuristiques

1. introduction:

Dans le cadre des méthodes modernes de résolution des problèmes d'optimisation, les métaheuristiques occupent une place centrale, notamment dans les domaines où les approches exactes deviennent impraticables à cause de la complexité combinatoire. Ce chapitre présente les principales familles de métaheuristiques utilisées dans l'optimisation discrète, en mettant l'accent sur les méthodes inspirées de la nature. Une attention particulière est accordée à l'algorithme (PSO), qui constitue le cœur de notre approche pour la conception de VLANs. Nous détaillerons sa structure, ses mécanismes fondamentaux, ainsi que ses avantages par rapport à d'autres méthodes classiques.

2. Problèmes d'optimisation combinatoire:

- Méthodes de résolutions:

Il existe généralement deux approches pour la résolution des problèmes difficiles ; approche résolution exacte et approche résolution approximative ; dans la première on cherche à trouver la solution exacte d'un problème donné, les algorithmes de type B&X représentent cette approche.[CT03]

Pratiquement l'application d'un tel mode de résolution est limitée pour les petites instances des problèmes difficiles. La deuxième tendance est la résolution approximative, là on cherche à trouver des bonnes solutions en se basant sur des méthodes stochastiques dites heuristique ; parmi ces heuristiques on trouve des méthodes assez générales pour être appliquées sur un grand spectre des problèmes ces méthodes sont les métaheuristiques.[FK97]

3. Les métaheuristiques

3.1. Définition:

Le mot métaheuristiques est composé de deux mots ; le mot méta qui est un préfixe a la signification "au-delà" ou bien "dans un niveau supérieur " ; et le mot heuristique qui vient du verbe heuriskein et qui signifie 'trouver'

Dans littérature on trouve des définitions qui expliquent bien la notion de métaheuristiques citons :

“A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions.”[Bou02]

“A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a constructive method.”[Can98]

“A **metaheuristic** is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem. Examples of metaheuristics include simulated annealing (SA), tabu search (TS), iterated local search (ILS), evolutionary algorithms (EC), and ant colony optimization (ACO).”

De ces définitions on peut conclure que les métaheuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale, leur but est de faire explorer l'espace de recherche d'une manière efficace pour atteindre des solutions presque optimales, elles sont en générale non déterministes (pas de garantie), généralement elles contiennent des techniques pour échapper de l'optimum local, leurs concepts de base peuvent être décrit de manière abstraite (indépendamment d'un problème spécifique). Même si on utilise des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques resteront contrôlées par une stratégie de niveau supérieur.[AtMG02]

3.2. Classification

On peut classer les métaheuristiques selon plusieurs points de vue, par conséquent on peut trouver un métaheuristiques qui appartient aux différentes classes selon le critère de la classification [BPSV01], donc on a comme critères :

Le nombre des solutions manipulées à la fois : selon un tel point on distingue deux

classes les métaheuristiques basées sur la manipulation (évaluation et modification) d'une solution à la fois, là la notion de voisinage est pertinente, on trouve dans cette classe la méthode recuit simulé, recherche avec tabou, GRASP, ILS, VNS et autre.[FK99]

Dans l'autre coté on trouve les métaheuristiques basées sur la manipulation d'un ensemble des solutions à la fois dite généralement population des solutions ; citons dans cette classe les algorithmes génétiques, optimisation par colonies des fourmis et optimisation par essime des particules.[KGV83]

Type de parcours : les métaheuristiques se sont des méthodes qui explorent l'espace de recherche soit avec le déplacement dans l'ensemble de voisinage ou avec l'échange de l'état des individus d'une population. Ce déplacement forme une sorte de parcours qui peut être continu comme dans RS et la recherche avec tabou, ou discontinu comme dans ILS, GRASP, AG, ACO.[WA04]

Emploie de mémoire : la mémoire signifie l'utilisation de l'historique de la recherche pour déduire l'état courante de l'optimisation, dans ce contexte on peut distinguer les métaheuristiques dites "son mémoire" où l'état actuel de la recherche est déterminé uniquement suivant l'état précédant, c'est le cas de la descente récursive, le recuit simulé, recherche avec tabou et autres. Par contre il existe des méthodes qui utilisent l'historique pour guider la recherche telle que la recherche avec tabou, la mémoire peut être figurée selon différentes formes, par exemple on peut considérer que l'ensemble des individus d'un AG forme une mémoire, le phéromone dans ACO aussi et les positions dans le PSO.[Zam06]

La Source de la méthode : on trouve généralement deux classes, les méthodes inspirées de la nature telle que le RS, les algorithmes génétiques, l'ACO et PSO, et dans l'autre coté on trouve des méthodes qui ont des origines non naturelles, citons la recherche avec tabou, GRASP, VNS et ILS.[Wid01]

Nombre et structure de voisinage : le voisinage est l'ensemble des solutions (états) atteignables à partir de la solution (état) courante, le voisinage peut être fixe ou variable, les métaheuristiques selon ce critère sont classifiées en deux : les métaheuristiques avec un voisinage fixe, c'est le cas de la majorité des heuristique, ou

les métaheuristiques avec un voisinage variable, c'est le cas de la méthode VNS (recherche avec voisinage variable). [SDPT03]

4. Les méthodes basées sur une seule solution

Appelées aussi méthodes de trajectoire, le processus de recherche forme une sorte de chemin ou un parcours le long de leur progression, il se déplace d'une solution vers une autre dans *l'espace de recherche* ; ce déplacement est possible grâce à la notion de voisinage. Les sections suivantes donnent une vision sur les méthodes de recherche locale les plus connues, de la descente récursive qui représente une méthode de base, au recuit simulé et la recherche avec tabou. [HGH99]

4.1. La descente récursive :

C'est un exemple typique des méthodes de recherche locale, elle progresse à travers l'ensemble des solutions X par le choix de meilleure solution voisine de la solution courante et ainsi de suite, ce processus s'interrompt lorsqu'un minimum local est atteint .[CTA05]

Cette méthode est caractérisée par sa simplicité mais elle a deux inconvénients :

- Suivant la taille et la structure du voisinage $N(s)$ considérées, la recherche de la meilleure solution voisine est un problème qui peut être aussi difficile que le problème (P) initial.

elle est incapable de progresser au-delà du premier minimum local rencontré. Par contre les problèmes d'optimisation combinatoire comportent en générale plusieurs optima locaux pour lesquels la valeur de la *fonction objectif* peut être forte éloignée de la valeur optimale.[BPSV01]

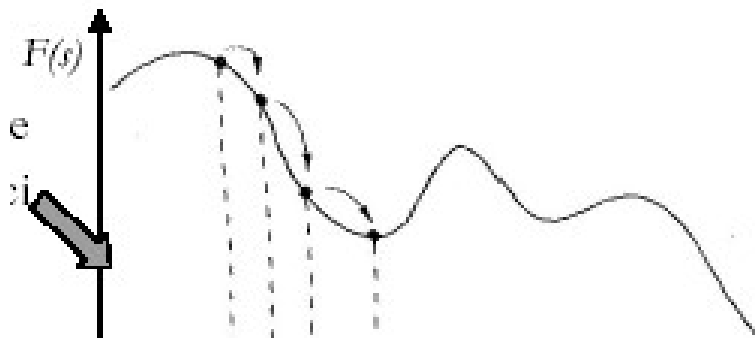


Figure 4 L'algorithme de descente ne peut pas « remonter » pour sortir d'un optimum local

Donc l'idée est de faire accepter des solutions moins bonnes (pour échapper du minimum local); en plus on exploite une portion de voisinage d'une solution pour faire diminuer la taille du problème; ce que fait les autres méthodes de recherche locale.[FK97]

4.2. Le recuit simulé

Le recuit simulé (SA) est parmi les plus vieilles métaheuristiques et sûrement l'un des premiers algorithmes qui ont une stratégie explicite pour échapper des minimums locaux [BR03]. Les origines du recuit simulé remontent aux expériences réalisées par Metropolis et al. dans les années 50 pour simuler l'évolution d'un tel processus de recuit physique [Wid01] ; Metropolis et al. utilisent une méthode stochastique pour générer une suite d'états successifs du système en partant d'un état initial donné. Tout nouvel état est obtenu en faisant subir un déplacement (une perturbation) aléatoire à un atome quelconque. Soit ΔE la différence d'énergie occasionnée par une telle perturbation. Le nouvel état sera accepté si l'énergie du système diminue ($\Delta E < 0$). Sinon, il sera accepté avec une probabilité définie par :

$$p(\Delta E, T) = \exp^{(-\Delta E / (C_b \cdot T))}$$

Où T est la température du système et C_b est un constant physique connu sous le nom de *constant de Boltzmann*.

Le recuit simulé est une méthode de recherche locale dont le mécanisme de recherche est calqué sur l'algorithme de Metropolis et autres, et les principes du recuit

thermodynamique ; il a été la première fois présenté comme un algorithme de recherche pour les problèmes d'optimisation combinatoire par S. Kirkpatrick et al [KGV83]. L'idée fondamentale est de permettre au processus de recherche de se déplacer vers des solutions ayant une qualité plus mauvaise que la qualité de la solution courante (mouvements ascendants) afin de s'échapper de minimum local. La probabilité de faire un tel déplacement est diminuée pendant la recherche.

L'idée consiste à utiliser l'algorithme de Metropolis et al. Pour atteindre des états avec une énergie aussi faible que possible. Le refroidissement progressif d'un système de particules est simulé en faisant une analogie entre l'énergie du système et la fonction objectif du problème d'une part, et entre les états du système et les solutions admissibles de l'autre part. Noté que chaque état du système représente une solution potentielle du problème. [CMRR02]

- Déroulement

Initialement on donne au système une très haute température ; puis on le refroidit petit à petit. Le refroidissement du système doit se faire très lentement pour avoir l'assurance d'atteindre un état d'équilibre à chaque température T .

Le voisinage $N(s)$ d'une solution S_x s'apparente à l'ensemble des états atteignables depuis l'état courant en faisant subir des déplacements infinitésimaux aux atomes du système physique.[LS01]

A chaque itération, une seule solution voisine s' est générée et elle est acceptée si elle est meilleure que la solution courante () dans le cas contraire on a les cas :

si T grande, $\exp^{(-\Delta E/T)}$ est de l'ordre de 1, on garde toujours le mouvement, même s'il est mauvais.[RPE01]

- si T très petit, $\exp^{(-\Delta E/T)}$ est de l'ordre de 0, donc les mouvements qui augmentent l'énergie (la différence) seront disqualifiés.

Méthode : on tire au hasard dans l'intervalle $[0,1[$, si le nombre est $< \exp^{(-\Delta E/T)}$, on garde sinon on jette.[SunMS05]

La meilleure solution trouvée est mémorisée dans la variable s^* [Wid01].

La méthode recuite simulée donne des très bons résultats pour le problème de

voyageur de commerce mais pas pour le problème d'affectation ; leur performance liée au schéma de refroidissement.[DS00]

4.3. La Recherche avec Tabou:

La recherche tabou ou recherche avec tabou est l'une des métaheuristiques les plus utilisées dans le domaine de l'optimisation des problèmes difficiles, elle est introduite par Glover (1986) basé sur ses premières idées formulées dans (1977). La méthode de recherche tabou utilise l'historique de la recherche pour implémenter un mécanisme permettant d'échapper de l'optimum local, et aussi d'éviter les cycles dans la recherche ce qui résulte une bonne exploration de l'espace de recherche.[DFJ54]

4.3.1. Principe et démarche:

Comme une vue générale la méthode de recherche avec tabou se comporte comme la descente récursive, elle applique une recherche locale basée sur la règle d'amélioration de la fonction objectif, c'est-à-dire choisir de l'ensemble des voisins de la solution courante, la meilleure solution et qu'elle doit être mieux que la solution courante ; si on atteint un optimum local, la recherche taboue permet d'accepter la meilleur solution qui appartient à l'ensemble des voisines même si elle a une qualité inférieure de la solution actuelle, en autre mots on a choisi le moins mauvais ;[BPSV01] La recherche tabou ne s'arrête donc pas au premier optimum trouvé.[Fos02] Une telle méthode basée sur une telle stratégie peut fortement tomber dans une recherche cyclique ; c'est-à-dire si un minimum local s se trouve au fond d'une vallée profonde est atteint, il sera impossible de ressortir de celle-ci en une seule itération, et un déplacement de la solution s vers une solution $s' \in N(s)$ avec $f(s') > f(s)$ pourra provoquer le déplacement inverse à l'itération suivante, puisqu'en général $s \in N(s')$ et $f(s) < f(s')$; dans ce cas on a un cycle de longueur 2

Pour régler ce problème, une mémoire à court terme est mise en application, cette mémoire stocke les dernières meilleures solutions déjà visitées sous forme d'une liste dite liste taboue, où tout mouvement vers une solution qui appartient à cette liste est interdit. À chaque itération, la meilleure solution de l'ensemble des solutions permis (solutions voisines de la solution courante et n'appartient pas à la liste tabou) est choisie comme une nouvelle solution courante. [CGJ+00]

En plus, cette solution est ajoutée à la liste de tabou et une des solutions qui étaient déjà dans la liste de tabou est enlevée (habituellement dans un ordre de FIFO) [BR03].

L'utilisation de la mémoire permet d'éviter tout cycle de longueur inférieur ou égal à la taille de la liste tabou, par conséquent elle oblige le processus de recherche d'explorer des nouvelles régions dans l'espace de recherche. Le pseudo-code de l'algorithme tabou classique .[Sch60]

La mémorisation des solutions entières sera trop coûteuse en terme d'espace et en terme de gestion de la mémoire et ne sera pas la plus efficace. Pour cela la liste tabou mémorise des attributs des solutions au lieu de solutions complètes. Les attributs sont habituellement des composants des solutions, des mouvements, ou des différences entre deux solutions. Noté que plus d'un attribut peut être considéré, dans ce cas une liste taboue est présentée pour chacun d'eux. L'ensemble d'attributs et les listes tabou correspondantes définissent les conditions taboues qui sont employés pour filtrer le voisinage d'une solution et pour produire l'ensemble des solutions permis. Le stockage des attributs des solutions au lieu des solutions complètes est beaucoup plus efficace, mais il présente une perte d'information, car la mémorisation d'un attribut dans la liste de tabou peut interdire plus d'une solution [BR03], un exemple simple de mémorisation des attributs est présenté dans [HGH99].

4.3.2. Fonction d'aspiration

A un moment donné les mécanismes employés réduisent l'espace de recherche et interdisent le processus de recherche d'explorer des zones qui peuvent être intéressantes, pour cette raison un mécanisme dit aspiration ou fonction d'aspiration est utilisé ; il permet de ressortir une solution de la liste taboue et l'utilise comme une solution candidate à la prochaine itération. Cette solution devra satisfaire un critère d'aspiration ($f(s') < A(f(s))$) [Wid01] ; la fonction la plus simple consistera à enlever le statut tabou d'un mouvement si ce dernier permet d'atteindre une solution de qualité supérieure à celle de la meilleure solution trouvée jusqu'alors [HGH99].

4.3.3. Intensification et diversification

L'intensification et la diversification sont deux techniques (parmi d'autres) utilisées pour améliorer la recherche avec tabou. L'intensification consiste à explorer en détaille une région de l'espace de recherche qui comprend des solutions qui partagent quelques propriétés ; l'idée est d'élargir l'ensemble de voisinage de la solution courante. Par contre la diversification consiste à diriger l'exploration vers des régions inexplorées de l'espace de recherche ; il faut simplement faire redémarrer le processus de recherche avec une solution initiale aléatoire ou choisie judicieusement dans une région non encore visitée de l'ensemble des solutions admissibles [Wid01].

4.3.4. Applications de la recherche tabou

On a dit que la méthode de recherche tabou est l'une des métaheuristiques les plus utilisées, elle a été appliquée dans plusieurs domaines, dans le domaine de la télécommunication (conception des réseaux...), la conception (conception assistée par ordinateur, conception des réseaux de transport,...), le routage (routage des véhicules, voyageur de commerce...), optimisation des graphes (coloration des graphes..), les problèmes d'allocation (problème d'assignement quadratique...) et autres [Glo95].

5. Les méthodes basées sur une population des solutions

Les métaheuristiques basées sur la manipulation d'une population des solutions essayent d'augmenter la qualité moyenne d'un ensemble des solutions via un mécanisme défini. Généralement ces méthodes sont inspirées de la nature, du domaine de la biologie, l'éthologie.[Fos02]

On peut distinguer entre deux approches utilisées par les méthodes basées sur une population des solutions : la première est l'utilisation des mécanismes d'évolution naturels cette approche est représentée par les AGs, EV, et autres. La deuxième est l'utilisation de l'intelligence collective, cette approche est représentée par ACO, PSO et autres. Dans les deux sections suivantes on va donner un exemple de chacune de ces stratégies qui sont les AGs et l'ACO. [WA04]

5.1. Les Algorithmes Génétiques

5.1.1. définition

Les AGs appartiennent à un ensemble des méthodes dites algorithmes évolutifs, le terme « algorithmes évolutifs » englobe toute une classe des métaheuristiques. Ces algorithmes sont basés sur le principe du processus d'évolution naturelle. Un algorithme évolutif typique est composé de trois éléments essentiels [HGH99]: une population constituée de plusieurs individus chacun représente une solution potentielle du problème donné.[Mel99]

1. un mécanisme d'évaluation de l'adaptation de chaque individu de la population à l'égard de son environnement extérieur.
2. un mécanisme d'évolution composé d'opérateurs permettant d'éliminer certains individus et de produire de nouveaux individus à partir des individus sélectionnés ; les AGs s'appuient sur des techniques dérivées de la génétique : croisements, mutations, sélection, etc.

Les algorithmes génétiques (GAs) ont été inventés par John Holland dans les années 60 et ont été développés par Holland et ses étudiants et collègues à l'université du Michigan dans les années 60 et les années 70.

Contrairement aux stratégies d'évolution et programmation évolutionnaire, le but original de Holland n'était pas de concevoir des algorithmes pour résoudre des problèmes spécifiques, mais plutôt pour étudier formellement le phénomène de l'adaptation, et comment il se produit dans la nature ; et pour développer des méthodes dont les mécanismes de l'adaptation normale pourraient être importés dans les systèmes informatiques. En 1975 Holland présente dans son livre *Adaptation in Natural and Artificial Systems* les algorithmes génétiques comme une abstraction de l'évolution biologique, et il a donné un cadre théorique pour l'adaptation basé sur les algorithmes génétiques [Mel99].

5.1.2. Déroulement

Dans les algorithmes génétiques, on essaie de simuler le processus d'évolution d'une population. Au début on génère une population des solutions d'une manière aléatoire, cette population forme la population initiale. Le degré d'adaptation de chaque individu à l'environnement (exprimé par la valeur de la fonction coût dite

fonction fitness) est calculé ; après, des couples des individus P_1 et P_2 (appelés parents) sont sélectionnés en fonction de leurs adaptations ; ensuit un opérateur de croisement est appliqué sur P_1 et P_2 avec une probabilité P_c et génère des couples C_1 et C_2 (dites enfants). D'autres individus P sont sélectionnés en fonction de leur adaptation ; un opérateur de mutation est appliqué sur P avec une probabilité P_m (P_m est généralement très inférieur à P_c) et génère des individus mutés P_0 . Le niveau d'adaptation des enfants (C_1, C_2) et des individus mutés P_0 sont ensuite évalués avant de les insérer dans la nouvelle population ; l'insertion est faite par le choix de N individus parmi la population des parents et la population des enfants. En itérant ce processus jusqu'à un critère d'arrêt est atteint. Un critère d'arrêt peut être un nombre fixe d'itérations ou lorsque la population n'évolue plus ou moins suffisamment rapide [AD05]

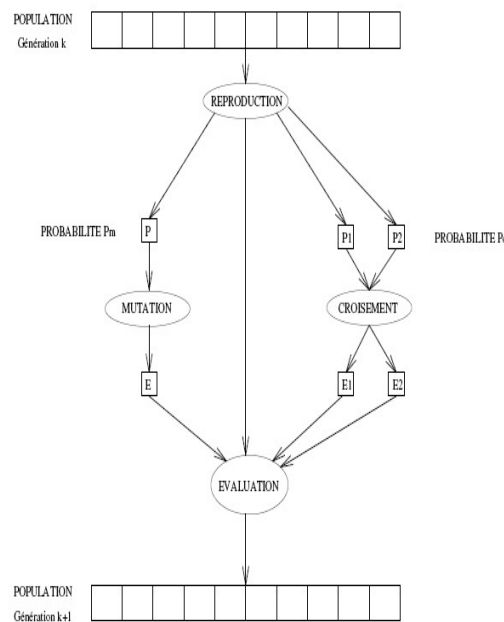


Figure 5 principe démarche des algorithmes génétiques

5.1.3. Le codage utilisé par les algorithmes génétiques

Un algorithme génétique repose sur l'utilisation d'un codage des informations de problème à résoudre, afin qu'il puisse appliquer les opérateurs génétiques sur ces informations ; il existe plusieurs manières pour coder les données du problème parmi lesquels on trouve le code binaire naturel, le code binaire de Gray et le code réel. [FK99]

- Le codage en binaire :

Dans ce type de codage, pour un individu on code ses variables et on les concatène par exemple, la chaîne binaire 1000| 0110| 1101, correspond à un individu défini par 3 variables (8, 6, 13) en codage binaire naturel sur 4 bits chacune. [CMRR02] C'est le codage le plus utilisé pour plusieurs raisons : pour des raisons historiques, ce codage a été utilisé par J. Holland (le fondateur des AGs) et ses étudiants ; plusieurs résultats théoriques sont basés sur ce codage, et il est facile à mettre en place les opérateurs génétiques avec ce codage. Cependant si la longueur de la chaîne augmente la performance de l'algorithme diminuera. [GLPF01]

- Le codage réel

Il s'agit de concaténation des variables x_i d'un individu x . par exemple un individu x (25, 31, 8) est codé 25| 31| 8 ; ce codage plus précis que le codage binaire, l'espace de recherche est le même que l'espace du problème, l'évaluation de la fonction coût est plus rapide ; mais son alphabet est infini et il a besoin d'opérateurs appropriés. [BR04]

- Les opérateurs génétiques

Dans un algorithme génétique il existe trois principaux opérateurs : l'opérateur de sélection, l'opérateur de croisement, et de mutation. Ces opérateurs jouent un rôle prépondérant dans la possible réussite d'un AG. [DFJ54]

a) L'opérateur de sélection

C'est le choix des individus pairs qui vont participer à la reproduction de la génération suivante. Il existe plusieurs méthodes pour la reproduction. Comme il y a plusieurs méthodes pour sélectionner un individu ; la méthode la plus connue et utilisée est la roue de loterie biaisée (roulette wheel) de Goldberg (1989) ; selon cette

méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. Donc si on considère la fitness d'un chromosome particulier étant donné $f(d(c_i))$, la probabilité avec laquelle il sera réintroduit dans la nouvelle population de taille N est :

$$\frac{f(d(c_i))}{\sum_{j=1}^N f(d(c_j))}$$

Dans ce cas les individus ayant une grande fitness ont donc plus de chance d'être sélectionnés [VY01] ; cette stratégie pose le problème de convergence prématurée si la population contient un super individu qui ne sera pas forcément un optimum global ou proche de l'optimum global.

Il existe d'autres méthodes, par exemple celle du *tournoi* (*tournamentselection*) : on tire deux individus aléatoirement de la population et on reproduit le meilleur des deux dans la nouvelle population. On refait cette procédure jusqu'à ce que la nouvelle population soit complète. Cette méthode donne de bons résultats [VY01].

b) L'opérateur de croisement

Cet opérateur permet de créer des nouveaux individus ; il permet de quelque sorte l'échange d'information entre les individus.

Au début on tire au hasard deux individus de la population courante, et on définit aléatoirement un ou plusieurs points de croisement. Ces derniers marquent des sites de croisement. Après selon une probabilité P_c , les segments des deux parents sont alors échangés autour de ces sites (voir figure) ; le résultat est la création de deux nouveaux individus (dans le cas d'un site de croisement). Noté qu'un individu sélectionné ne subit pas forcément l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité. Plus cette probabilité est élevée et plus la population subira de changement [AD05].

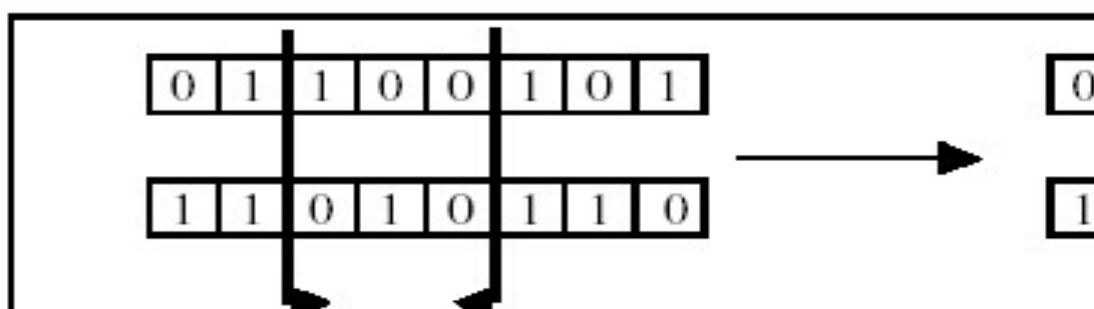


Figure 5 principe de l'opérateur de croisement

c) L'opérateur de mutation

La mutation consiste à changer aléatoirement un composant (gène) de l'individu. Cette opération est faite avec une certaine probabilité P_m (généralement inférieure de la probabilité de croisement P_c), par exemple si on utilise un codage binaire, un composant qui porte la valeur 1, on lui affecte une valeur 0 et vice versa (voir la figure).

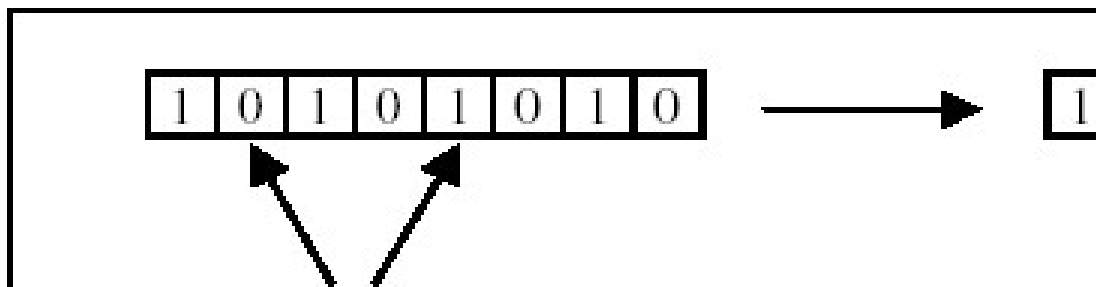


Figure 6 principe de fonctionnement de l'opérateur de mutation

La mutation est un mécanisme qui protège les algorithmes génétiques contre les pertes prématurées d'informations pertinentes, ces informations qui ont pu être perdues lors de l'opération de croisement. La mutation maintient aussi la diversité de l'algorithme pour l'exploration des nouvelles régions [AD05].

5.2. L'optimisation paresseuse de particules

5.2.1. Définition

L'optimisation par essaim de particule est une méthode d'optimisation basée population, elle est introduite pour la première fois par Dr. Russell C, Eberhart et Dr. James Kennedy en 1995. Qu'ils ont inspirés cette méthode de la nature [KE95].

La méthode se base sur un ensemble d'individus qui font l'exploration de l'espace de recherche avec un mécanisme d'interaction ; on appelle chaque individu une particule, chaque particule est caractérisée par deux propriétés principales :

- sa position courante (solution actuelle appartenant à l'espace de recherche)
- sa vitesse

Chaque particule survient toujours à deux positions, la meilleure position qu'elle a connue dans son évaluation et la meilleure position connue dans son voisinage [Set05].

A la différence dans les algorithmes génétiques qui utilisent des mécanismes génétiques dans leurs évolutions l'OEP fait l'appel à un mécanisme coopératif utilise les informations (position et vitesse) pour améliorer les solutions visitées.

5.2.2. Démarche

Au début ; l'essaim (l'ensemble de particules) est initialisé aléatoirement C'est-à-dire associer à chaque particule une solution initiale ensuite à chaque itération, chaque particule fait deux opérations principales, une adaptation de sa vitesse(1) et une adaptation de sa position(2), Les deux opérations faites appellent à la position courante, la vitesse courante, la meilleure position connue par la particule et la meilleure position connue par toute l'ensemble de voisinage de la particule (la figure), les adaptations suivent les équations suivantes:

$$V_i(t) = V_i(t-1) + P_1 (X_{ipbest} - X_i) + P_2 (X_{gbest} - X_i) \dots\dots\dots(1)$$

$$X_i(t) = X_i(t-1) + V_i(t). \dots\dots\dots(2)$$

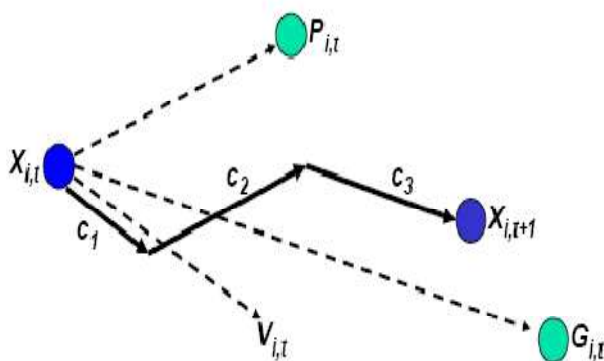


Figure 7 principe démarche le processus

Le processus stop lorsque la condition d'arrêt est atteinte, cette condition peut être un nombre maximal d'itérations, une fitness ou si l'évolution de processus ne fait pas augmenter la qualité des solutions avec un algorithme [Set05].

5.2.3. Les paramètres

Comme tous les métaheuristiques, l'efficacité de l'optimisation est fortement liée aux paramètres choisis l'heure de l'implémentation, pour l'optimisation par essaim de

particule, on trouve des paramètres numériques tel que les coefficients P_1 et P_2 et des paramètres généraux tel que la façon d'initialisation de la population, le nombre des particules et la structure et nombre des particules qu'ils construisent l'ensemble de voisinage de chaque particule de la population.

On a P_1 et P_2 peut être définie comme

$$P_1 = R_1 C_1$$

$$P_2 = R_2 C_2 \quad \text{ou} \quad R_1 \text{ et } R_2 \in [0, 1[.$$

Et C_1, C_2 sont des constants avec $C_1 + C_2 \leq 4$.

Il existe d'autres paramètres tel que :

- Le nombre de particule.
- La vitesse maximale
- Le facteur d'inertie.

```

Si on prend les valeurs suivantes :
N nombre de particules
 $X_i$  position de la particule  $P_i$ 
 $V_i$  vitesse de la particule  $P_i$ 
pbesti meilleure fitness obtenue pour la particule  $P_i$ 
gbest meilleure fitness obtenue dans tout le voisinage de  $P_i$ 
Xpbesti position de la particule  $P_i$  pour la meilleure fitness
Xgbest position de la particule ayant la meilleure fitness de toutes
 $P_1$  et  $p_2$  valeurs aléatoires positives
Initialisation aléatoire de la population
Tan que la condition d'arrêt n'est pas atteinte
    Pour i de 1 à N faire
        Si ( $f(X_i) > pbest_i$ ) alors
             $pbest_i \leftarrow f(X_i)$ 
             $X_{pbest_i} \leftarrow X_i$ 
        finsi
        Si ( $f(X_i) > gbest$ ) alors
             $gbest \leftarrow f(X_i)$ 
             $X_{gbest} \leftarrow X_i$ 
        fin si
    fin pour
    Pour i de 1 à N faire
         $V_i \leftarrow V_{i-1} + p_1(X_{pbest_i} - X_i) + p_2(X_{gbest} - X_i)$ 
         $X_i \leftarrow X_{i-1} + V_i$ 
    fin pour
fin tan que
    
```

Figure 8 pseudo code de la méthode PSO

5.2.4. Topologies des voisinages

Il y a 3 topologies principales de voisinage utilisées dans PSO (voir la figure) : cercle, rayon et étoile. Le choix de la topologie de voisinage détermine quel individu à employer pour le X_{gbest} de X_i .

Dans la topologie de cercle, chacun individuel est connecté socialement à ses k voisins topologiques les plus proches (X_{gbest} de X_i = mieux résultat individuel parmi ses k voisins plus proches, k égale typiquement 2). La topologie de rayon, isole efficacement des individus les uns des autres, car l'information doit être communiquée par un individu focal.

La topologie d'étoile est meilleure connue sous le nom de meilleure topologie globale, ici chaque individu est relié à chaque autre individu (X_{gbest} de X_i = mieux différents résultats dans la population) [Set05].

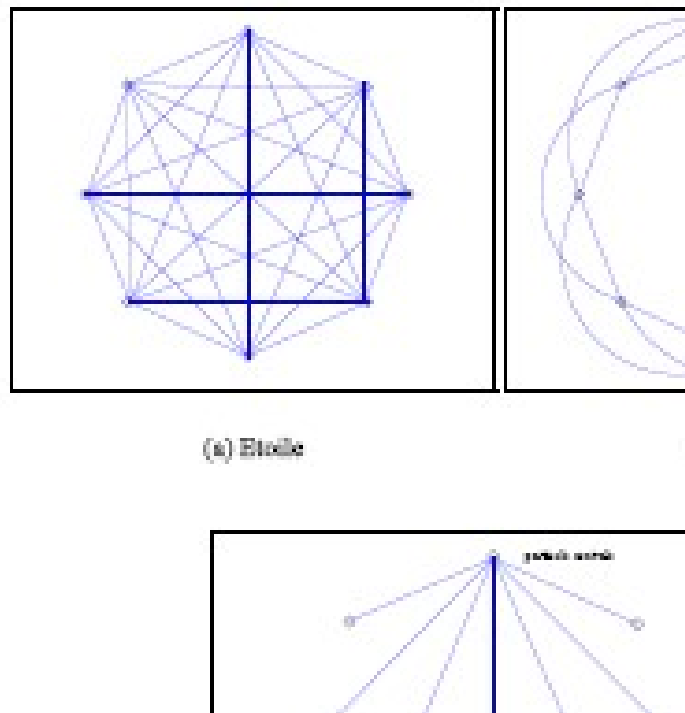


Figure 9 Exemples de topologies de voisinage pour PSO

5.2.5. Les avantages et les inconvénients

L'avantage majeur de l'optimisation par essaim de particules est qu'elle donne des bons résultats avec un nombre réduit de calcul par rapport d'une méthode basée population telle que les AGs, l'OEP exécute un nombre inférieur d'opération pour

avoir des solutions avec la même qualité de celle obtenus par les AGs, ou peut être mieux.[BR04]

Dans l'autre côté l'OEP souffre du problème de la convergence prématurée, c'est-à-dire la recherche converge rapidement vers un optimum local loin de l'optimum global en termes de qualité. En plus l'OEP nécessite un effort additionnel lors du traitement des problèmes de nature discontinu.[AtMG02]

6. Conclusion

Les métaheuristiques se sont des méthodes générales qu'ils peuvent être appliqués sur un grand spectre des problèmes difficiles pour donner des solutions efficaces dans un temps raisonnable. Ces méthodes sont des méthodes stochastiques, elles partagent le même principe, générer un(ou plusieurs) solution(s) et tester que cette (ces) solution(s) est (sont) bonne(s) ou non. Les métaheuristiques se distinguent dans la manière d'appliquer ce principe et dans les techniques additionnelles pour échapper d'un optimum local ou pour se déplacer dans l'espace de recherche.

On peut classer les métaheuristiques selon plusieurs critères : nombre des solutions manipulées, source de la méthode, type de voisinage utilisé et autre. Mais le critère le plus utilisé est le nombre des solutions manipulées par la méthode où on distingue deux grandes classe, les méthodes de recherche locale qui manipulent une solution à la fois et les méthodes basées population qui manipulent un ensemble des solutions à la fois. Cette classification influe considérablement sur la manière d'appliquer des autres techniques telles que le parallélisme et l'hybridation afin d'augmenter l'efficacité des métaheuristiques.

Chapitre 3 : Conception de l'algorithme PSO pour les VLANs

Chapitre 3 : Conception d'une solution basée sur l'algorithme PSO pour le design de VLAN

1. Introduction

La segmentation des réseaux informatiques en VLANs (Virtual Local Area Networks) vise à améliorer la sécurité, la performance et la gestion du trafic dans les environnements d'entreprise. Cependant, le problème de découpages optimaux de réseaux complexes demeure un défi computationnel. Ce chapitre présente la conception d'une approche basée sur l'optimisation par essaim des particules, adapté pour résoudre efficacement le problème de partitionnement des VLANs.

2. Modélisation du problème de partitionnement des VLANs

Le problème peut être modélisé comme un problème de factorisation matriciel. Considérons un réseau composé de N machines (ou hôtes), dont les communications sont décrites par une matrice d'interconnexion $C = [C_{ij}]$, représentant si les deux hôtes i et j sont connectés entre eux ou non.

L'objectif est de regrouper ces machines en K VLANs de telle sorte que le nombre totale des VLAN soit minimum.

3. Justification du choix de PSO

3.1 Comparaison avec d'autres algorithmes

Contrairement aux algorithmes comme GA ou ACO, PSO présente une convergence rapide, nécessite peu de paramètres à ajuster, et est bien adapté aux problèmes discrets avec grands espaces de recherche.

3.2 Avantages spécifiques de PSO

PSO permet une exploration efficace de l'espace des solutions tout en maintenant une exploitation locale via les composantes sociales et cognitives.

3.3 Études antérieures similaires

Des recherches récentes ont appliqué PSO avec succès dans le design de réseaux, notamment pour le placement des routeurs ou la segmentation de sous-réseaux.

4. Adaptation de l'algorithme PSO au problème de VLAN

4.1 Codage des solutions

Chaque solution est un vecteur de longueur N où l'élément i correspond au VLAN :

Soit P_i un vecteur booléen qui représente l'ensemble des VLANs du taille N

Donc on a
$$\begin{cases} P[j] = 1 \text{ si le Vlan contien au moin un hote} \\ P[j] = 0 \text{ si non} \end{cases}$$

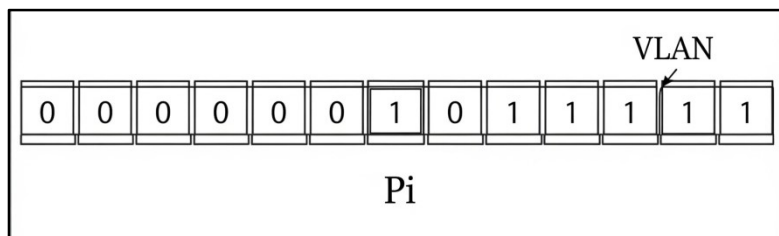


Figure 10 vecteur de longueur N correspond VLAN

4.2 Fonction de coût fitness

La formulation traditionnelle du problème d'optimisation du schéma de contrôle d'accès VLAN est la suivante. Les données initiales de ce problème sont données par la matrice A [n,n].

Il s'agit de trouver une matrice Z [n,k] telle que la condition (1) soit satisfaite, et que la dimension k soit minimale. Formellement, cette déclaration peut être écrite comme suit :

$$\begin{cases} A = Z \otimes Z^T \\ n \times (n \times k) \end{cases}$$

Où $Dim(Z, q) \rightarrow \min$ est une fonction qui détermine la q-ième dimension de la matrice Z

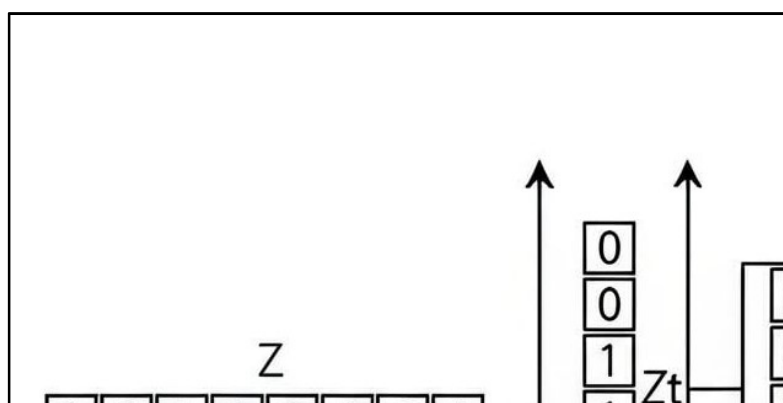


Figure 11 matrice Z de la fonction fitness

4.3 Mécanismes de calcul

On va modéliser la soustraction entre les vecteurs qui représentent les positions avec le XOR binaire donc si $C = A - B$ alors $C_i = A_i \text{ XOR } B_i$

La multiplication d'un vecteur par une paramètre est le choix d'une partie du vecteur L'addition c'est une addition binaire ordinaire

4.4 Pseudocode de l'algorithme adapté

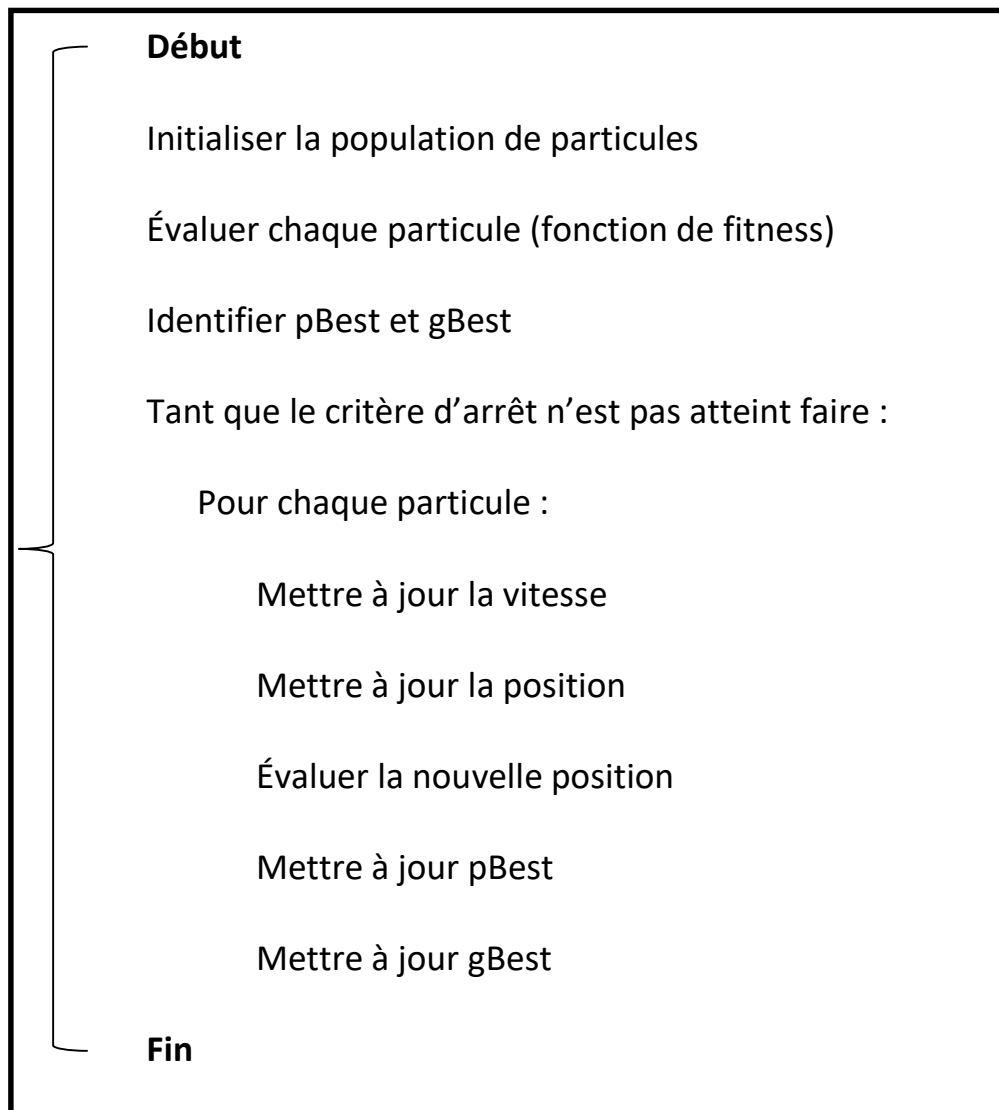


Figure 12 Pseudocode de l'algorithme adapté

5. Complexité algorithmique

5.1 Analyse temporelle

La complexité temporelle de PSO est $O(N \times D \times T)$ où :

- N : nombre de particules

- D : dimension du problème
- T : nombre d'itérations

5.2 Analyse spatiale

La complexité en mémoire est principalement déterminée par la taille de la population et les dimensions des vecteurs position/vitesse.

6. Conclusion :

Dans ce chapitre, nous avons conçu une approche basée sur PSO pour la conception optimale des VLANs. Cette solution sera implémentée et testée dans le prochain chapitre afin d'évaluer sa performance dans des scénarios réels.

L'utilisation de l'algorithme PSO s'avère prometteuse pour résoudre le problème complexe du design VLAN dans des infrastructures à grande échelle. La prochaine étape consistera à valider cette approche à travers une implémentation concrète et une série d'expérimentations.

Chapitre4: L'implémentation

Chapitre 4 :L' implémentation

1. Introduction

Ce chapitre aborde l'aspect pratique de notre projet, qui vise à optimiser la segmentation des réseaux locaux virtuels (VLANs) à l'aide de l'algorithme d'optimisation par essaim de particules (PSO). Nous présenterons l'environnement de développement, l'interface graphique, les étapes d'exécution de l'algorithme, l'analyse graphique des résultats, et enfin, la génération de rapports et de configurations réseau prêtes pour une application pratique.

2. Environnement de Développement et Outils Utilisés

2.1 Langage de Programmation :

Le développement de l'application a été entièrement réalisé en **Python**, spécifiquement la version **3.9 ou ultérieure**. Ce choix a été motivé par la flexibilité du langage, sa syntaxe claire et concise, et la richesse de son écosystème de bibliothèques, particulièrement adaptées aux calculs scientifiques, à la manipulation de données, et au développement d'interfaces graphiques.

2.2 Bibliothèques Python Essentielles :

- **NetworkX** : Spécifiquement dédiée à la création, la manipulation et l'étude des graphes et des réseaux, NetworkX a été utilisée pour modéliser la topologie du réseau et visualiser l'affectation des dispositifs aux différents VLANs.
- **NumPy** : Indispensable pour les opérations numériques de haute performance, NumPy a été massivement utilisée pour la manipulation des matrices de connexion (Matrice A, A_i , F_i), les calculs complexes de la fonction d'aptitude (fitness function), et l'optimisation des performances des algorithmes numériques.
- **Matplotlib** : Cette bibliothèque a été employée pour la visualisation graphique des résultats. Elle a permis de générer des tracés variés, incluant

les représentations visuelles des matrices (A , A_i , F_i), l'historique de la fonction d'aptitude, et la topologie du réseau, facilitant ainsi l'analyse et l'interprétation des données.

- **Pandas** : Cette bibliothèque a joué un rôle crucial dans la gestion des données structurées, notamment pour le chargement des matrices de connexion à partir de fichiers CSV et la sauvegarde des résultats dans des formats lisibles.
- **Threading** : Pour assurer une interface utilisateur fluide et réactive, même pendant les calculs intensifs de l'algorithme PSO, la bibliothèque threading a été utilisée. Elle a permis d'exécuter le processus d'optimisation en arrière-plan (dans un thread séparé), évitant ainsi le blocage de l'interface graphique.
- **Tkinter** : Cette bibliothèque standard de Python a été la pierre angulaire pour la conception et l'implémentation de l'interface utilisateur graphique (GUI) de l'application. Elle a permis de créer une interface intuitive et interactive pour les utilisateurs.

2.3 Environnement de Développement Intégré (IDE) :

Le développement du code a été effectué à l'aide de Visual Studio Code. Cet IDE a fourni un environnement complet pour l'écriture du code, le débogage, la gestion des dépendances et l'organisation du projet.

2.4 Système exploitation :

L'application a été développée et testée sur un système d'exploitation Windows, assurant sa compatibilité et son bon fonctionnement dans cet environnement

3. Démarche:

- Création d'une matrice binaire symétrique A de taille $n \times n$ où chaque ligne contient au maximum 20 valeurs de 1.
 - Génération d'un ensemble de 30 vecteurs binaires P_i aléatoires, chacun contenant au plus 20 bits à 1.
 - Initialisation du vecteur $Lbest_i$ pour chaque position.
 - Calculer le $Gbest$ initial pour la population initiale
 - Chaque vecteur P_i est initialement son propre.
 - Tant que la condition d'arrêt n'est pas atteinte
- **Calcul de A_i pour chaque P_i :** $A_i = P_i \otimes P_i^T$ (produit booléen)
 - **Calcul de F_i :** $F_i = A - A_i$
 - **Mise à jour de $Lbest$ et $Gbest$:**
 - Si $F_i < Lbest_i$ alors $Lbest_i = P_i$
 - Si $F_i < Gbest$ alors $Gbest = P_i$
 - **Mise à jour des positions :**
 - $V_i = c1 \times (Lbest - P_i) + c2 \times (Gbest - P_i)$
 - $P_i = P_i + V_i$
 - Répéter jusqu'à atteindre le nombre maximal d'itérations.

Structure du programme Python

Fichier config.py contient les paramètres suivants :

MATRIX_SIZE = 30

MAX_ONES_PER_ROW = 20

NUM_PARTICLES = 50

C1 = 1.5

C2 = 1.5

MAX_ITERATIONS = 100

4. Interface de application

4.1 Interface principale

À l'exécution, l'interface principale affiche les outils de contrôle, les paramètres PSO, les résultats et la topologie du réseau.

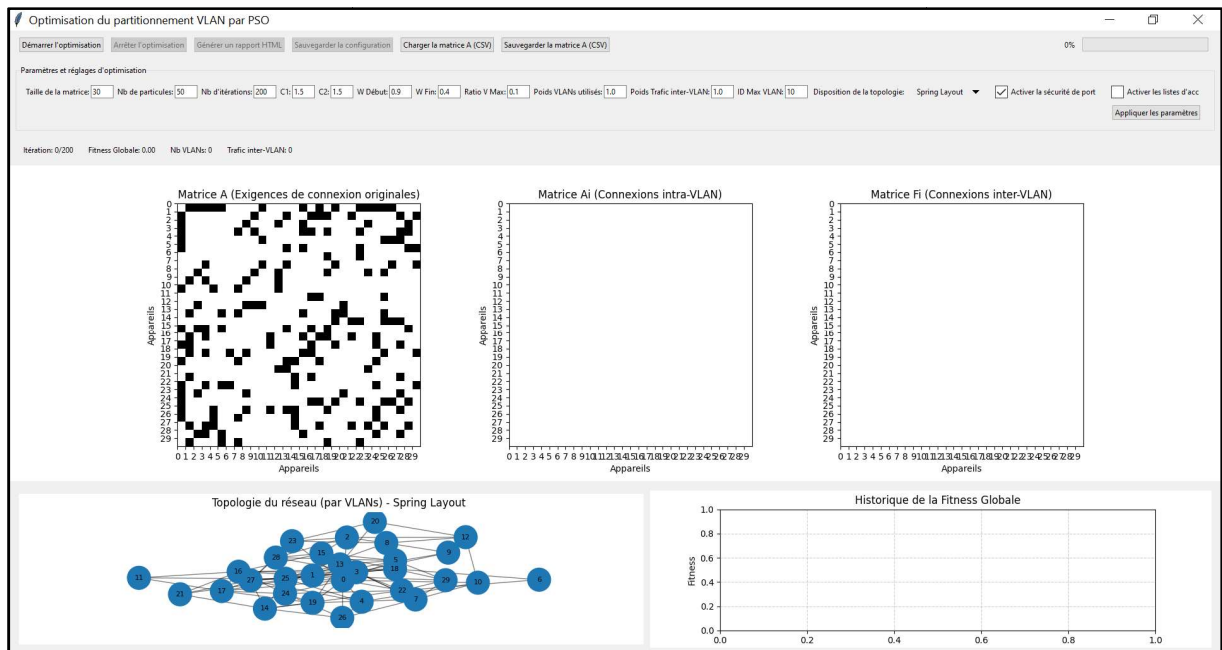


Figure 13 Interface principale

4.2 Paramètres de l'algorithme

L'utilisateur peut modifier les tailles de matrices, le nombre de particules, les itérations, $C1$, $C2$, les poids inertiels (W), la vitesse maximale, etc.

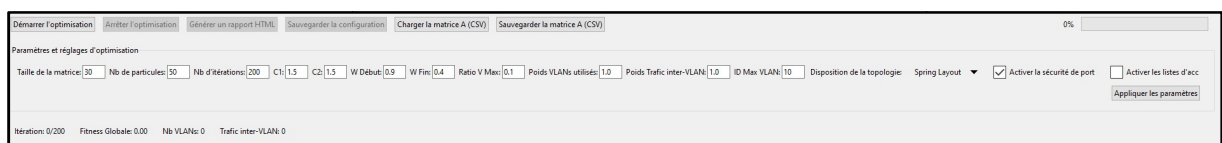


Figure 14 Paramètres de l'algorithme

4.3 Lancement de l'optimisation

Le bouton "Démarrer l'optimisation" exécute l'algorithme PSO dans un thread séparé.

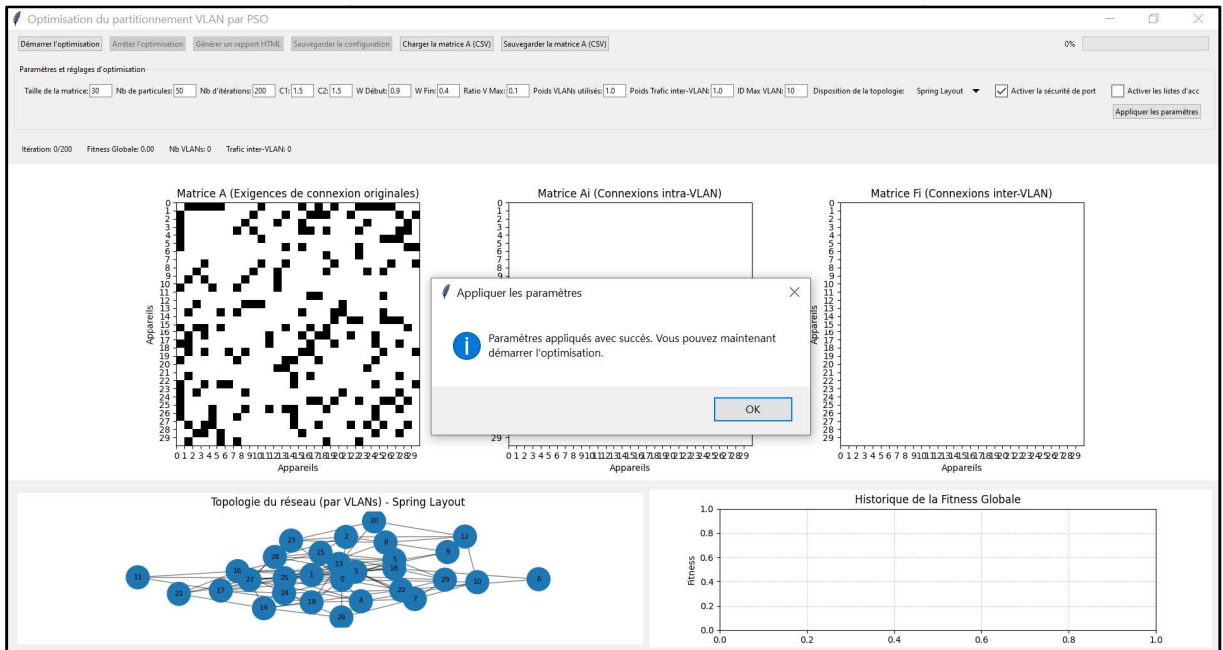


Figure 15 Lancement de l'optimisation

4.4 Suivi de l'optimisation

Affichage dynamique de la progression via :

- Barre de progression
- Graphiques (Fitness globale)
- Matrices Ai et Fi mises à jour

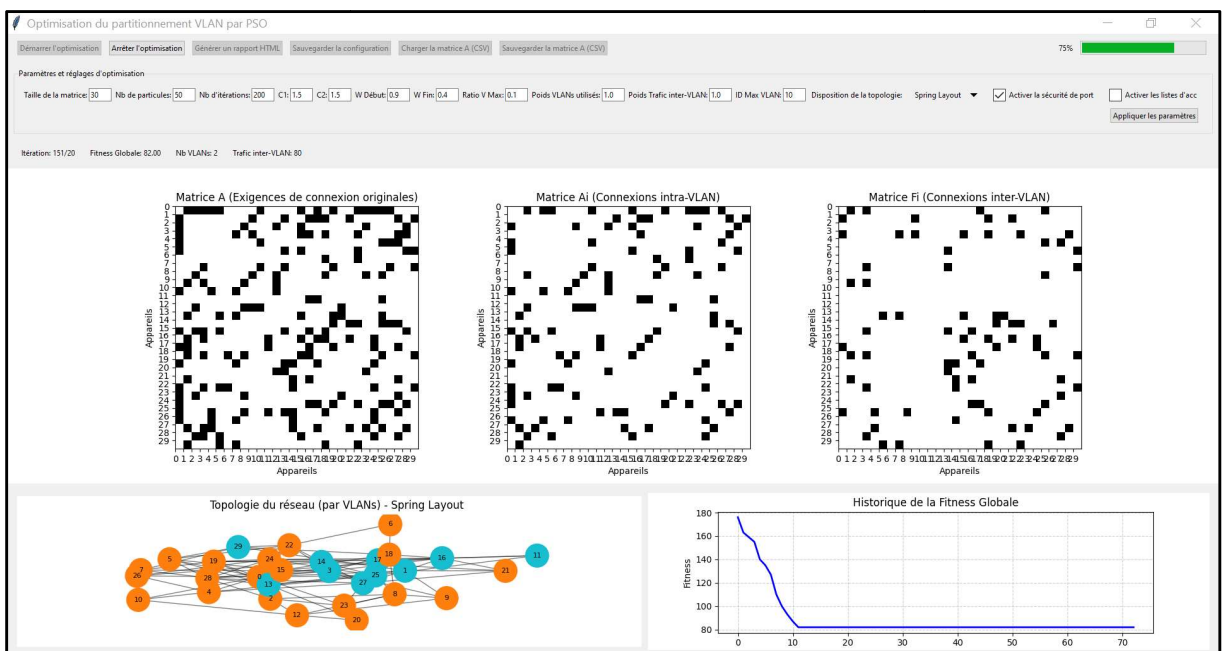


Figure 16 Suivi de l'optimisation

4.5 Résultats finaux

À la fin des itérations, l'application affiche :

- Meilleure valeur de fitness globale
- Meilleure attribution VLAN (Gbest)
- Matrices Ai et Fi finales
- Topologie colorée indiquant les VLANs

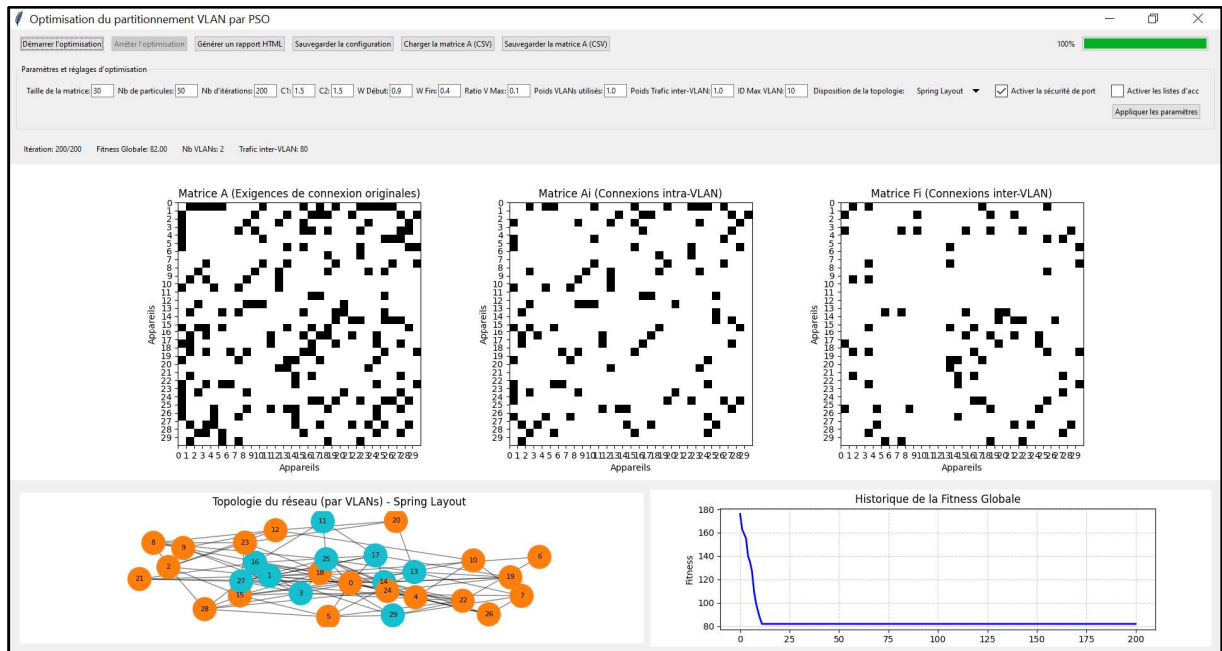


Figure 17 Résultats finaux

4.6 génération du Rapport

- L'application produit des rapports détaillés pour les résultats d'optimisation.
- L'utilisateur peut enregistrer le rapport sous un nom et un emplacement personnalisés.
- Le rapport inclut un résumé des résultats, des graphiques (matrices, évolution de la fitness, topologie), les attributions de VLANs, et une configuration Cisco proposée (VLANs, Port Security Access Lists)

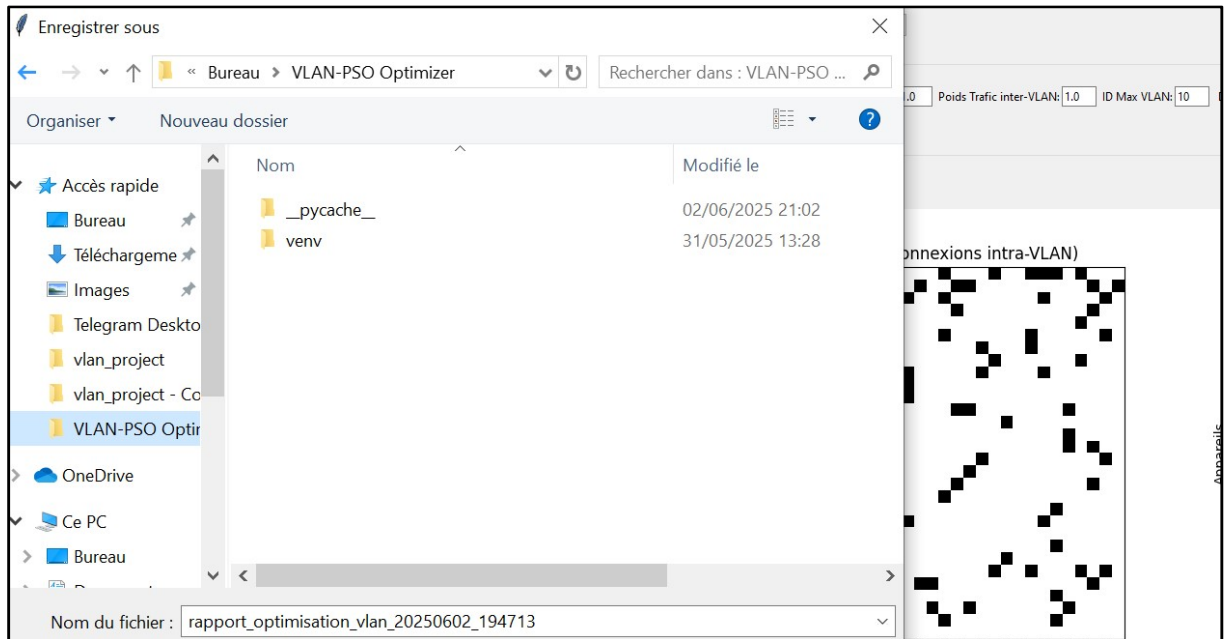


Figure 18 génération du Rapport

4.7 Contenu du Rapport :

- **complet** : Fournit un rapport regroupant tous les résultats d'optimisation en un seul endroit pour faciliter la révision et l'analyse.
- **Résumé des résultats finaux** : Affiche les indicateurs clés tels que la valeur de fitness finale, le nombre de VLANs utilisés et le volume des interconnexions.
- **Graphiques visuels** : Inclut des images de toutes les matrices, de l'évolution de la fitness et de la topologie résultat.
- **Attributions de VLANs détaillées** : Présente une liste des appareils et leurs attributions VLANs suggérées.
- **Configuration Cisco proposée** : Fournit un code prêt à l'emploi pour l'application des VLANs, de la sécurité des ports (Port Security) et des listes d'accès (Access Lists) (si activées)



Figure 19rapport d'optimisation du partitionnement VLAN

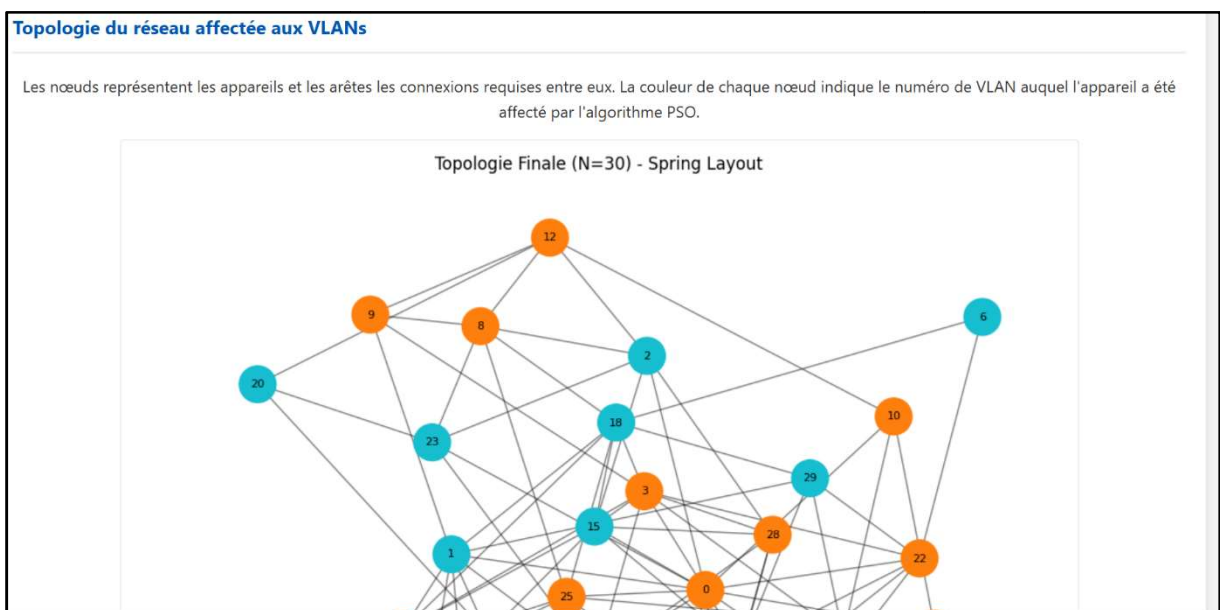


Figure 20topologie du réseau affectée aux VLAN

Affectation des VLANs aux appareils (Vecteur Gbest)

Cette section affiche le VLAN affecté à chaque appareil (de 0 à 29) :

```
Appareil 00:      VLAN 1
Appareil 01:      VLAN 9
Appareil 02:      VLAN 9
Appareil 03:      VLAN 1
Appareil 04:      VLAN 1
Appareil 05:      VLAN 1
Appareil 06:      VLAN 9
Appareil 07:      VLAN 1
```

Figure 21 affectation des VLAN aux appareils

Configuration VLAN suggérée (Exemple Cisco IOS)

Paramètres suggérés pour appliquer les affectations VLAN à un commutateur réseau, facilitant la configuration des

```
vlan 1
  name VLAN_1
!
vlan 9
  name VLAN_9
!
interface GigabitEthernet0/1
  switchport mode access
  switchport access vlan 1
  switchport port-security
  switchport port-security maximum 2
  switchport port-security violation restrict
```

Figure 22 configuration VLAN suggérée

5. Résumé

Cette application représente une solution efficace pour l'implémentation d'un design VLAN basé sur PSO. Elle permet une grande interactivité et adaptabilité aux différents scénarios réseau.

6. Conclusion :

Dans ce chapitre, nous avons détaillé l'implémentation pratique d'une solution basée sur l'algorithme PSO (Particle Swarm Optimization) pour

l'optimisation de la segmentation de réseaux en VLANs. Nous avons présenté la structure générale du programme, les différentes étapes de l'algorithme, ainsi que l'interface graphique développée pour faciliter l'interaction utilisateur et la visualisation des résultats.

Grâce à cette approche, il devient possible de générer automatiquement des configurations VLAN efficaces, réduisant les coûts de communication inter-VLAN tout en assurant une répartition optimale des périphériques. L'utilisation de PSO montre ainsi sa pertinence dans le domaine de la conception intelligente des réseaux informatiques. Le chapitre suivant se consacrera à l'expérimentation et à l'évaluation des performances de la solution proposée

.

Conclusion générale

Conclusion générale:

Notre motivation pour ce travail découle de l'émergence de réseaux informatiques massifs appartenant à de grandes institutions ou entreprises, souvent géographiquement dispersées. De là est née la nécessité de segmenter ces réseaux en un ensemble de réseaux virtuels, principalement pour des raisons de sécurité.

Cependant, et afin de réduire la complexité des configurations de sécurité, il est crucial de minimiser le nombre de réseaux virtuels. C'est ce qui nous a conduits à aborder ce problème comme une problématique d'optimisation visant à trouver la solution optimale.

Pour notre travail, nous avons eu recours à un algorithme d'intelligence en essaim : l'algorithme d'optimisation par essaim de particules (PSO). Nous l'avons choisi en raison de son efficacité prouvée dans la résolution de ce type de problèmes.

Au terme de ce travail, nous avons démontré l'efficacité de l'approche par Optimisation par Essaim de Particules (PSO) pour la conception optimale des VLANs dans les grandes infrastructures réseau. En combinant la puissance d'exploration de l'algorithme PSO avec une modélisation rigoureuse du problème de partitionnement, nous avons pu générer des topologies virtuelles optimisées.

Il est important de souligner que ce travail nécessite une analyse approfondie des résultats pour être amélioré. En effet, nous avons noté certaines lacunes dans l'algorithme, notamment un phénomène de convergence précoce vers un optimum local. De plus, la solution pourrait être étendue pour inclure la conception de réseaux virtuels basée sur les rôles.

Ce travail ouvre des perspectives intéressantes pour l'intégration d'autres techniques hybrides et l'application de cette méthode à d'autres domaines de l'optimisation réseau.

Bibliographie

- [ABGL01] K. Anstreicher, N. Brixius, J-P. Goux, J. Linderoth. **Solving Large Quadratic Assignment Problems on Computational Grids**. To appear in Mathematical Programming, available at http://www.optimizationonline.org/DB_HTML/2000/10/233.html, 2001.
- [ACE+02] M.G. Arenas, P. Collet, A.E. Eiben, M. Jelasity, J.J. Merelo, B. Paechter ; M. Preuss, and M. Schoenauer. **A framework for distributed evolutionary algorithms**. In Proceedings of PPSN VII, september 2002.
- [AD05] J.-M. Alliot and N. Durand. **algorithmes génétiques**. March 14, 2005
- [ANT02] E. Alba, A. J. Nebro, and J. M. Troya **Heterogeneous Computing and Parallel Genetic Algorithms** Journal of Parallel and Distributed Computing 62, 1362–1385 (2002)
- [AT02] E. Alba and M. Tomassini. **Parallelism and Evolutionary Algorithms**. IEEE transactions on evolutionary computation, vol. 6, no. 5, OCTOBER 2002
- [AtMG02] E. Alba and MALLBA Group. **MALLBA: A library of skeletons for combinatorial optimisation**. In R.F.B. Monien, editor, Proceedings EuroPar , volume 2400 of LNCS, pages 927-932 , Paderborn, 2002 Springer-Verlage.
- [AUB+05] Alet'cia P.F. Ara'ujo, Sebasti'an Urrutia, Cristina Boeres, Vinod E.F. Rebello, Celso C. Ribeiro. **Towards Grid Implementations of Metaheuristics for Hard Combinatorial Optimization Problems**. 17th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'05), 2005
- [BEG04] M. Belal and T. El-Ghazawi **PARALLEL MODELS FOR PARTICLE SWARM OPTIMIZERS**. *IJICIS, VOL. 4, NO. 1, January 2004*
- [Bou02] L. Bougé. **Si tous les ordinateurs du monde... Calcul scientifique vraiment très haute performance sur la grille mondiale**. Département info-télécoms ENS Cachan, Antenne de Ker Lann 25 octobre 2002
- [BPSV01] M. Birrtari, L. Paquete, T. Stutzle and K. Varrentrap. **Classification of metaheuristics and Design of Experiments for the analysis of component**. November 2001
- [BR03] C. Blum And A. Roli. **Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison**. ACM Computing Surveys, Vol. 35, No. 3, September 2003, pp. 268–308.
- [BR04] C. Boeres and V. Rebello. **Easygrid: Towards a framework for the automatic grid enabling of legacy mpi applications**. *Concurrency And Computation Practice And Experience*, 17(2):173.190, 2004.
- [Can98] E. Cantu-Paz. **A Survey of Parallel Genetic Algorithms**. Calculateurs parallèles, réseaux et systèmes répartis 10, 141-171, Hermès, Paris, 1998
- [CGJ+00] B. Carpenter, V. Getov, G. Judd, A. Skjellum and G. Fox. **MPJ: MPI-like Message Passing for Java**. 2000

- [CMRR02] V.-D. Cung, S.L. Martins, C.C. Ribeiro, and C. Roucairol. **Strategies for the Parallel Implemetation of Metaheuristics**. Essays and Surveys in Metaheuristics. C.C. Ribeiro, P. Hansen (Eds.), 263-308, Kluwer Academic Publishers, Norwell, MA, 2002.
- [CMT04] S. Cahon N. Melab E.-G. Talbi. **ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics**. Journal of Heuristics, Vol 10 :353-376, May 2004
- [CN05] T.G. Crainic, H. Nourredine, **Parallel Meta-Heuristics Applications**. Parallel Metaheuristics, E. Alba (Ed.), John Wiley & Sons, 2005.
- [CT03] T.G. Crainic and M. Toulouse. **"Parallel Strategies for Meta-heuristics"**, State-of-the-Art Handbook in Metaheuristics, F. Glover, G. Kochenberger (Eds.), 475-513, Kluwer Academic Publishers, Norwell, MA, 2003.
- [CTA05] C. Cotta, EG. Talbi, and E. Alba. **Parallel hybrid metaheuristics**. In Alba, E., ed.: Parallel Metaheuristics, a New Class of Algorithms. John Wiley (2005) 347–370
- [DFJ54] G. Dantzig, R. Fulkerson, And S. Johnson **solution of a large-scale traveling-salesman problem***The Rand Corporation, Santa Monica, California* (Received August 9, 1954)
- [DS06] J. Dréo et P. Siarry. **Métaheuristiques pour l’optimisation et auto-organisation dans les systèmes biologiques***Technique et Science Informatiques, Vol. 25, No. 2. (2006), pp. 221-240. - 1 January 2006*
- [DS00] M. Dorigo and T. Stutzle. **The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances**. Technical Report IRIDIA-2000.. To appear in *Metaheuristics Handbook*, F. Glover and G. Kochenberger (Eds.), International Series in Operations Research and Management Science, Kluwer, 2001.
- [FK97] I. Foster et C. Kesselman**Globus: A Metacomputing Infrastructure Toolkit**. Intl. J. of supercomputer application, 11(2) : 115-128,1997
- [FK99] I. Foster et C. Kesselman. **The Grid: Blueprint for a New Computing Infrastructure**. Morgan Kaufmann, 1999.
- [FKT01] I. Foster, et C. Kesselman et S. Tuecke. **The Anatomy of the Grid: Enabling Scalable Virtual Organization**. The International Journal of High Performance Computing Applications, 15(3), 2001
- [FKTY03] K. Fujisaway, M. Kojimaz, A. Takeda and M. Yamashita. **High Performance Grid and Cluster Computing for Some Optimization Problems** Research Report B-396, October 2003. Revised December 2003
- [Fos02] I. Foster. **What is the Grid? A Three Point Checklist**. Grid Today, 1(6), july 22 2002
- [GL05] WasuGlinkwamdee_ J. T. Linderoth**MW: A Software Framework for Combinatorial Optimization on ComputationalGrids** October 20, 2005
- [Glo95] F. Glover. **tabu search fundamentals and uses**. 1995

- [GLPF01] V. Getov, G. V. Laszewski, M. Philippsen and I. Foster **Multiparadigm Communications in Java for GridComputing**. *Communications of ACM*, 2001
<http://www.globus.org/cog/documentataion/papers/>
- [HGH99] J.-K Hao, P. Galinier and M. Habib. **Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes**. *Revue d'Intelligence Artificielle* Vol : No. 1999
- [KE95] J. Kennedy and R. Eberhart. **Particle Swarm Optimization**. In *Proceedings of IEEE International Conference on Neural Network*, volume IV., pages 1942–1948, 1995
- [KGV83] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi. **Optimization by Simulated Annealing** 13 May 1983, Volume 220, Number 4598 **SCIENCE**
- [LMS02] H. R. Lourenço, O. Martin, and T. Stützle. **Iterated Local Search**. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 321-353, Kluwer Academic Publishers, Norwell, MA, 2002.
- [LS01] J. Linderoth, J. W. Stephen **Computational Grids for Stochastic Programming Optimization** Technical Report 01-01 October 10, 2001.
- [Mal05] N. Meleb **contributions à la résolution des problèmes d'optimisation combinatoire sur grilles de calcule** Thèse Pd soutenue le 28 Novembre 2005.
- [Mel05] N. Meleb. **contributions à la résolution des problèmes d'optimisation combinatoire sur grilles de calcule** Thèse soutenue le 28 Novembre 2005.
- [Mel99] M. Melanie. **An Introduction to Genetic Algorithms**. A Bradford Book The MIT Press Cambridge, Massachusetts. London, England Fifth printing, 1999
- [Rai06] G. R. Raidl **A Unified View on Hybrid Metaheuristics**. Third International workshop, HM 2006, Gran Canaria, Spain, October 13-14, 2006
- [Rat04] C. Rattanapoka **P2P-MPI: A Peer-to-Peer Framework For Robust Execution of Message Passing Prallel Programs on Grids** Mémoire de DEA Univercité Louis-Pasteur de Strasbourg 2003/2004
- [RPE01] M. G. C. Resende, P. M. Pardalos and S. D. Eksioglu. **Parallel Metaheuristics for Combinatorial Optimization**. *Advanced Algorithmic Techniques of Parallel Computation with Applications*, R. Correa et al. (Eds.), 179-206, Kluwer, 2001.
- [Sch60] A. Schrijver **On the history of combinatorial optimization (till 1960)**
- [SDPT03] P. Siarry, J. Dréo, A. Pétrowski and É. Taillard. **métaheuristiques pour l'optimisation difficile** Éditions Eyrolles 2003 ISBN : 2-212-11368-4
- [Set05] M. Settles **An Introduction to Particle Swarm Optimization** Department of Computer Science, University of Idaho, Moscow, Idaho U.S.A 83844 November 7, 2005
- [SHMD05] H. Shimosaka, T. Hiroyasu, M. Miki, and J. Dongarra. **Optimization Problem Solving System using GridRPC**. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, VOL. 1, NO. 1, OCTOBER 2005

- [Ste 00] J. Stephen. **Solving Optimization Problems on Computational Grids** Wright November 17, 2000
- [SunMS05] **JXTA v2.3.x: Java™ Programmer's Guide** Sun Microsystems, Apr 7, 2005
- [Tal99] E-G. Talbi. **A Taxonomy of hybrid metaheuristics**. Journal of combinatorial optimization 1999.
- [VY01] T. Valléeet M. Yildizo-glu. **Présentation des algorithmes génétiques et de leurs applications en économie**. 7 septembre 2001, v. 1.2
- [WA04] S. Wagner, M. Affenzeller. **Heuristiclabb Grid– a flexible and extensible environment for parallel heuristic optimization**.*Proceedings of the 15th International Conference on Systems Science*.2004
- [Wid01] M. Widmer **les metaheuristiques : des outils performants pour les roblemes industriels**. *3e Conférence Francophone de MOdélisation et SIMulation “Conception, Analyse et Gestion des Systèmes Industriels” MOSIM'01 – du 25 au 27 avril 2001 - Troyes (France)*
- [Zam06] L. Zambito. **The Traveling Salesman Problem: A Comprehensive Survey**. Submitted as a project for CSE 4080, Fall 2006