REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــعـــــــليـــم العـــــــــــالي و البـــــــــــحث العـــــــــــلـــمـــــــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــامعة غـــردايـــــــــــة

Université de Ghardaia

كـــليـــــة العـــلـــــــوم والتكنولوجيا

Faculté des Sciences et de Technologie

قســم الرياضيات و الإعلام الآلـــي

Département des Mathématiques et Informatique

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Informatique

Spécialité : Systèmes Intelligents pour l'Extraction de Connaissances (SIEC)

Par : HADJ KOUIDER Abdelhay

CHIHANI Hammou

Sujet

Détection des communautés suspectes dans un réseau social à base de Clustering hybridé PSO-SMA

Soutenu publiquement, le 28 Juin 2018, devant le jury composé de :

M. BETKA Messaoud	: Président
M. BOUHANI Abdelkader	: Encadreur
M. ADJILA Abderrahmane	: Examinateur
M. OULEDMEHREZ Abdelkader	: Examinateur

Année Universitaire 2017/2018

Remerciements

Nous tenons à remercier avant tout DIEU le tout puissant qui nous a donné la foi et le courage pour réaliser ce travail.

Nous exprimons notre profonde gratitude à notre promoteur M. BOUHANI Abdelkader pour la qualité du sujet qu'il nous a proposé et la confiance qu'il nous a accordé, ainsi que tous les conseils qu'il nous a prodigué durant la période de réalisation de ce travail.

Nous aimerions également remercier Mr.BELLAOUAR Slimane, Mr.OULAD NAOUI Slimane, Mr.KECHIDA et Mr.KERACHE Abdelaziz pour tous leurs précieux conseils et leurs observations lors des Sciences de suivi des travaux des étudiants.

Nous exprimons notre reconnaissance à toute personne ayant contribué de près ou de loin à notre formation et à l'élaboration de ce travail.

Résumé

La détection des communautés a une grande importance dans l'analyse des réseaux sociaux, où le système est souvent représenté sous forme d'un graphe, Dans ce travail, nous essayons d'élaborer une méthode de clustering basée sur l'optimisation discrète de l'essaimage de particules (PSO) pour la détection des communautés dans un réseau social.

Cette méthode peut détecter automatiquement le nombre de communautés dans le réseau analysé, la fonction objective (fitness) utilisée pour cela est la modularité, à maximiser. Après la détermination des communautés dans le réseau, et en sachant leurs centroïdes, on calcule la similarité pour chacun avec un profil suspect modèle afin de juger la communauté entière.

A la fin on applique un système multi-agents pour améliorer les résultats la détection des communautés.

Mots-clés :

Réseau social, Optimisation de l'essaim de particules, Clustering, Partitionnement de graphe, système Multi-agents.

Abstract

The detection of communities has of great importance in the analysis of social networks where the system is often represented by a graph. In this work, we try to develop a method of clustering based on the Particle swarm optimization (PSO) for the detection of communities in a social network.

This method can automatically detect the number of communities in the network analyzed, the objective function (fitness) used for this, is the modularity function, which we try to maximize. After determining the groups in the network, and knowing their centroids, we calculate the similarity with a typical suspicious profil in order to judge the whole group.

At the end we apply a multi-agent system to improve the results of community detection.

Keywords:

Social network, Particle swarm optimization, Clustering, Graph Partitioning, Multi-Agent System.

ملخص

إن الكشف عن التجمعات على شبكات التواصل الاجتماعي له أهمية كبيرة في تحليل هذه الأخيرة والتي يتم تمثيلها عادة على شكل مخططات.

من خلال هذا العمل، نحاول تطوير طريقة للتجميع بطريقة تحسين سرب الجسيمات (PSO) للكشف عن التجمعات في شبكة اجتماعية. يمكن لهذه الطريقة أن تكشف تلقائيًا عدد التجمعات في الشبكة التي تم تحليلها، مع استعمال النمطية (modularité) كدالة الهدف (fonction objective) للبرنامج ، والتي نسعى إلى رفع قيمتها. بعد تحديد المجموعات الموجودة في الشبكة، وتحديد مراكزها التي تمثلها، يتم إجراء حساب التشابه (similarité)مع حساب مشبوه نموذجي من أجل الحكم على المجموعة بأكملها.

في النهاية يتم إدخال تقنية نظام تعدد الوكلاء (SMA) لتحسين نتائج اكتشاف المجتمع.

كلمات مفتاحية:

شبكة اجتماعية، تحسين بسرب الجسيمات، تجميع، تقسيم المخططات، نظام تعدد الوكلاء.

TABLE DE MATIERES

TABLE DES FIGURES

TABLE DES TABLEAUX

Introduction Générale

Les réseaux sociaux sont aujourd'hui l'un des moyens interactifs les plus populaires pour communiquer, partager et diffuser une quantité considérable d'informations sur le web.

Malheureusement, ce type de média est de plus en plus utilisé par des utilisateurs malintentionnés, qui profitent de leurs anonymats et de l'ignorance de ses risques par la plupart des internautes, pour publier et échanger des contenus illégaux et suspects (images, vidéos, textes ...), cela pourra affecter la sécurité des individus, des institutions, et même des pays. Ce qui rend ces réseaux comme idéales plateformes pour les organisations suspectes en les permettant de mener librement ses activités.

Afin d'offrir une bonne prévention pour les internautes, il est nécessaire de faire une analyse intelligente de données circulants via ces médias. Et pour le faire, il faut commencer par identifier les individus ou les communautés qui peuvent présenter une source de risque dans ces réseaux.

Parmi les approche appliquées pour faire une telle analyse c'est d'identifier les groupes d'utilisateurs qui interagissent plus fréquemment les uns avec les autres et partagent des intérêts communs et des désirs similaires, Cette identification est appelée : Détection de communautés ; si on représente un réseau social comme un graphe, on trouve que dans une communauté les nœuds (individus) sont fortement liés, et peu connectés vers les autres communautés.

L'idée est de faire regrouper tous les individus qui partagent les mêmes caractéristiques dans des groupes (communautés), en d'autre terme les individus qui ont une importante similarité entre eux. Ensuite localiser parmi ces communautés les quelles qui ont un degré de suspicion en se référant à un profil suspect qui est auparavant constitué.

Par analogie, la détection communautaire dans les réseaux sociaux peut être simuler dans les graphes à un regroupement les nœuds soit par leurs attributs ou soit par la force des relations entre eux afin de former des sous-graphes fortement intra liés et faiblement inter liés, donc on revient au problème classique de partition de graphes ; où un certain nombre d'approches ont été proposées pour le résoudre.

En plus des approches mathématiques, la découverte de structures communautaires dans les réseaux peut être considérée comme un problème d'optimisation [32], où de nombreux algorithmes inspirés de la nature sont proposés au cours des dernières décennies, ces algorithmes

sont caractérisés par des possibilités de recherche globale, et d'une capacité de convergence rapide, et ont pris de l'ampleur à travers des études théoriques et empiriques.

Parmi ces algorithmes méta-heuristiques, le paradigme de l'optimisation de l'essaimage de particules (PSO), qui provient du comportement des animaux sociaux, tels que les oiseaux migrateurs qui voyagent sur des longues distances et qui doivent optimiser leurs efforts pour arriver à leurs objectifs en utilisant l'intelligence de l'essaim.

L'algorithme PSO optimise un problème en employant un groupe de particules. Chaque particule est une solution candidate au problème. Les solutions candidates sont mises à jour avec des règles simples apprises par les particules. En raison de son efficacité et de sa mise en œuvre facile, PSO est répandu dans le domaine de l'optimisation, et diverses variantes ont été proposées.

L'objectif de notre étude et d'élaborer un algorithme basé sur la technique méta-heuristique PSO implémenté dans un système multi-agent SMA. Cet algorithme aura comme résultat un ensemble de communautés suspectes dans un réseau social.

Nous organiserons notre document comme suit :

Le chapitre I présente la revue des concepts et définitions de base de différentes notions utilisées dans notre projet.

Le chapitre II, présente un état de l'art sur les contributions des chercheurs concernant les techniques qu'on va les utiliser ultérieurement dans la partie pratique.

Le chapitre III décrit notre contribution, ainsi que l'approche appliquée pour résoudre ce problème avec une discussion des résultats obtenus.

Une conclusion termine notre travail.

Chapitre 1

Notions Préliminaires

1 Introduction

Dans ce chapitre, nous donnons les notions préliminaires et les concepts de base qui seront utilisés par la suite dans ce mémoire.

2 Les réseaux sociaux et leurs dangers sur la société

2.1 Introduction

À l'ère du Web 1.0 la plupart des sites étaient commerciaux et communicatifs, avec un faible contenu d'informations et souvent statiques, l'ère du Web 2.0, se caractérise par un contenu dynamique personnalisé que l'internaute contribue à enrichir grâce aux facilités d'édition octroyées par les plateformes de blogs et de wikis, et qu'il partage au sein de communautés. L'internaute passe ainsi d'une position de consommateur à celui d'acteur à part entière, il est davantage impliqué, ce qui va entrainer l'apparition de communautés regroupées par centres d'intérêts que l'on appelle : les réseaux sociaux ou social networking.

Bien que les réseaux sociaux sont bénéfiques, peuvent être parfois très dangereux, et portent beaucoup de mal aux internautes et à la société.

2.2 Définition d'un réseau social

Un réseau social est un ensemble de sites qui permettent de mettre en relation des personnes (amis, connaissances, collègues), rassemblés en fonction de centres d'intérêt communs (musique, cinéma, Sport, …) ou encore d'intérêt professionnel.

Chaque utilisateur peut créer un profil et construire un réseau personnel qui le connecte aux autres utilisateurs, comme il peut ajouter du contenu sur le réseau social, qu'il s'agisse d'une image, d'une vidéo, d'un texte, ou d'un son. Les contenus sont commentés par les amis de la personne. Enfin, le profil peut être plus ou moins publique. Il est possible de n'être visible que de ses amis, ou d'avoir un profil visible par tout le monde.

Les réseaux sociaux les plus connus sont Facebook, Twitter, LinkedIn, Instagram, et YouTube.

2.3 Quelques chiffres relatifs

D'après le site smartinsights.com, en 2018 , le nombre d'utilisateurs d'internet dans le monde est de 4,021 milliards, en augmentation de 7% par an, le nombre d'utilisateurs de médias sociaux dans le monde est de 3.196 milliards, en augmentation de 13% par an, le nombre d'utilisateurs de téléphones mobiles est de 5,135 milliards, en hausse de 4% par an.

Le site statista.com a donné des nouvelles statistiques sur les réseaux sociaux les plus populaires dans le monde le mois d'avril 2018, classés par nombre d'utilisateurs actifs (en millions)

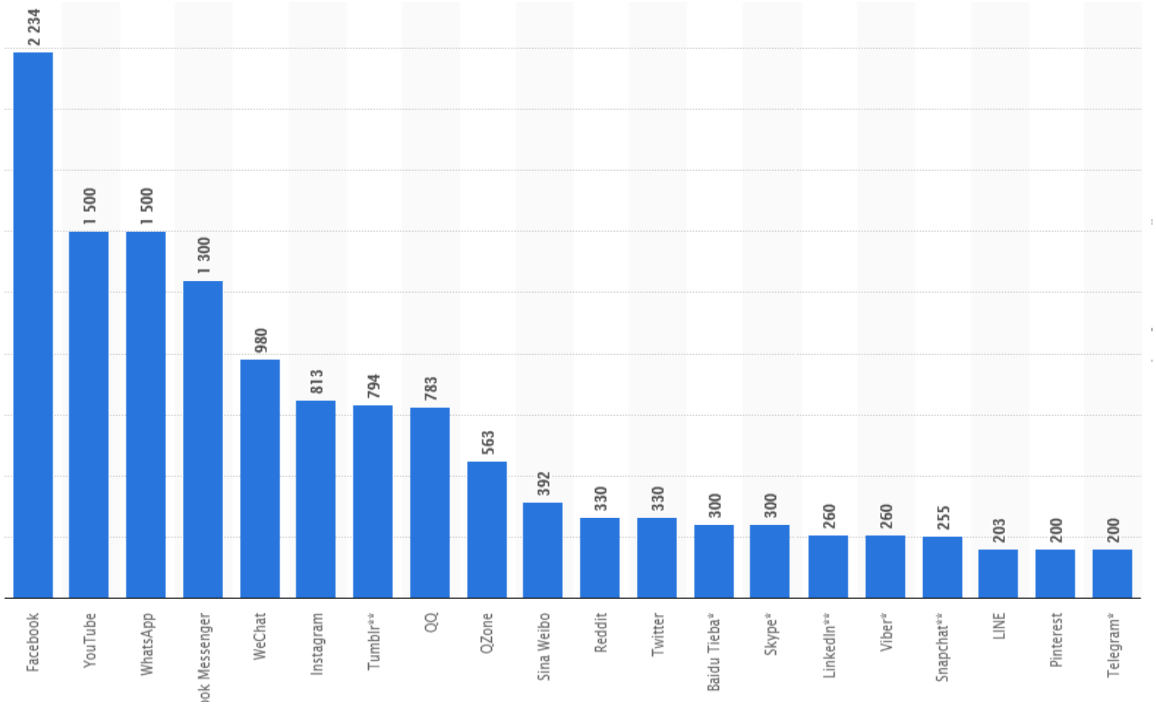


Figure 1: Nombre des utilisateurs actifs sur les Réseaux sociaux les plus populaires (En millions)

2.4 Principaux dangers dus aux réseaux sociaux

Les réseaux sociaux accueillent des millions d'utilisateurs chaque jour. Plusieurs entre eux sont devenus plus en plus dépendants de ces outils, cependant, cet engouement pour les réseaux sociaux n'est pas sans danger.

Les réseaux sociaux sont devenu une opportunité pour les délinquants et les fraudeurs, ils leurs permettent de capturer facilement des milliers de victimes potentielles. Les menaces sont nombreuses, à savoir, la cyber intimidation, harcèlement criminel, menace, incitation au suicide, diffusion de contenu illicite, compromettre, promouvoir la haine, le préjudice moral et physique et autres [1].

D'après [5] les crimes les plus fréquents sur les réseaux sociaux sont :

Les crimes sexuels (39,8% des cas). Tel qu'une majorité significative de cas dont les victimes sont des personnes mineures (57,2% de la catégorie « crimes sexuels »).

En deuxième lieu on trouve 16,4% de crimes liées au harcèlement, Il s'agit de répétition d'actes, pendant un certain temps, qui amènent la victime à craindre raisonnablement pour sa sécurité, comme le fait de suivre une personne ou de communiquer avec elle de façon répétée ; de surveiller la maison ou le milieu de travail ; ou de menacer directement une personne ou un membre de sa famille de manière à lui faire craindre pour sa sécurité ou pour celle d'une de ses connaissances.

Toujours d'après [5], les fraudes représentent un acte commis dans le but de réaliser un gain potentiel pour autant procéder à un vol d'identité et les atteintes aux biens et vie privés des personnes arrivent en 4ème et 5ème position avec respectivement 9,8% et 5,1% des affaires recensées.

Avec la naissance de terrorisme on peut parler d'une nouvelle catégorie : échange d'informations logistiques pour les attentats, le recrutement de nouveaux militants, diffusion de contenus lié à ce contexte.

Cette dernière catégorie constitue actuellement la plus grande préoccupation des secteurs de lute contre-terrorisme dans le monde entier, vu le grand volume et la nature complexe de données circulant dans le web social.

2.5 Communauté virtuelle

Le terme communauté est définie par le dictionnaire **SENSAGENT** du journal **LEPARISIEN** comme **«** groupe social dont les membres vivent ensemble, ont des intérêts communs. »

LAROUSSE le défini par : « Ensemble de personnes unies par des liens d'intérêts, des habitudes communes, des opinions ou des caractères communs »

Dans notre cas, la définition qui nous convient est celle de **LAROUSSE** avec un sens virtuel de la communauté.

« Une communauté virtuelle est un groupe de personnes qui communiquent par l'intermédiaire de courriels, internet en particulier via des forums, courrier, téléphone, pour des raisons professionnelles, sociales, éducatives ou autres » [11]. Le mot virtuel est employé pour signifier qu'il ne s'agit pas de communication face à face.

Certaines communautés sont purement virtuelles, d'autres se prolongent dans la réalité. Le Web permet de gérer des plannings, organiser des réunions, et passer des informations entre les individus de la communauté.

2.6 La détection de communautés

Le rôle de la détection de communautés dans un réseau social est de surligner des groupes qui sont formés implicitement, pas forcément par leurs choix, mais aussi en partageant des intérêts communs.

2.7 Sources de données sur les réseaux sociaux :

Les réseaux sociaux sont devenus une plateforme importante pour connecter les utilisateurs, partager les informations, et une source précieuse de données. Ainsi, la disponibilité de telles données représente une opportunité pour les gens afin d'étudier et d'analyser ces réseaux.

On peut distinguer deux types de données à extraire des réseaux sociaux [12]:

2.7.1 Données explicites

Sont l'ensemble des informations explicites fournies par les réseaux sociaux sur les utilisateurs, ou les données qui sont incorporées dans les informations fournies, à savoir, Les données dont l'utilisateur a fournis lui-même, ou a participé à leur création d'une façon explicite on trouve par exemple :

a. Données de service :

Ensemble de données qu'un utilisateur fournit au réseau social pour créer son compte tel que le nom de l'utilisateur, la date de naissance, le pays, etc.

b. Données publiées :

C'est ce que l'utilisateur publie sur son profil de réseau social. Cela peut inclure des commentaires, des photos postées, des entrées postées, des légendes, liens partagés, etc.

c. Données confiées :

C'est ce que l'utilisateur publie sur les profils d'autres utilisateurs. Ce pourrait inclure des commentaires, des légendes, des liens partagés, etc.

d. Données incidentes :

C'est ce que les autres utilisateurs de réseaux sociaux publient à propos de l'utilisateur. Cela peut inclure des photos, des commentaires, des notes, etc.

2.7.2 Données implicites

C'est l'ensemble d'informations non fournisses d'une façon explicite. Mais d'autres personnes ou les administrateurs de ces réseaux qui peuvent les déduire en basant sur le comportement de l'utilisateur et les différentes relations et contributions dans le réseau. Par conséquent, dans cette catégorie, les utilisateurs de réseaux sociaux sont considérés être passifs puisque l'information inférée est extraite d'activités antérieures ou données précédemment publiées.

a. Données comportementales :

Les réseaux peuvent collecter ces informations sur l'utilisateur, en étudiant ses habitudes et ses comportements dans le réseau.

b. Données dérivées :

Ce sont les données sur l'utilisateur qui peuvent être déduites de toutes les autres données. Ce n'est pas lié à l'habitude de l'utilisateur. Par exemple, l'adresse IP peut être utilisée pour déduire l'emplacement réel des utilisateurs, les relations que l'utilisateur a créées, et le type des amis avec lesquels il partage les informations, peuvent nous donner une information sur la personnalité de l'utilisateur lui-même.

3 Optimisation par Essaim de Particules (Particle Swarm Optimization PSO)

3.1 Présentation :

Le PSO est une méta-heuristiques, il a été proposé en 1995 par Kennedy et Eberhart. [2].

Les métas-heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels nous ne connaissons pas de méthodes classiques plus efficaces. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, d'où le qualificatif ***méta***. Leur capacité à optimiser un problème à partir d'un nombre minimal d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la meilleure solution trouvée. Cependant, du point de vue de la recherche opérationnelle, ce constat n'est pas forcément un désavantage, puisque l'on préfère toujours une approximation de l'optimum global trouvée rapidement à une valeur exacte trouvée dans un temps rédhibitoire. [13].

Le PSO est un processus d'optimisation analogique naturel qui cherche une solution à un problème d'optimisation basé sur le modèle du comportement de l'essaim biologique. Semblable au phénomène naturel, par exemple lors du déplacement des essaimes des oiseaux migrateurs qui

voyagent sur des longues distances et doivent enfaite optimiser leurs efforts (minimiser les distances des parcours, trouver les meilleures sources de nourriture, ...) en utilisant l'intelligence de l'essaim, en déplaçant sous la forme de la lettre V.

Un tel comportement permet à chaque élément (particule) de l'essaim de bénéficier des autres individus. Cette idée consiste qu'un groupe d'individus peu intelligent puisse posséder une organisation globale complexe et intelligente.

Le mouvement de ses oiseaux en essaim est complexe, sa dynamique obéit à des règles et des facteurs bien spécifiques, il s'agit de :

- Chaque individu dispose d'une certaine intelligence « limitée » (qui lui permet de prendre une décision).
- Chaque individu doit connaitre sa position locale et disposer d'information locale de chaque individu se trouvant dans son voisinage.
- Obéir à ces trois règles simples, « rester proche des autres individus », « aller dans une même direction » ou « voler à la même vitesse ».

Tous ses facteurs et règles sont indispensables pour le maintien de la cohérence dans l'essaim.

Une population de candidats de la solution est déplacée à travers l'espace de recherche pour obtenir une bonne solution au problème. Dans chaque étape de calcul, la position de chaque individu est recalculée.

3.2 Le principe général de fonctionnement d'un algorithme PSO

L'optimisation des essaims de particules est similaire à un algorithme génétique, en ce que le système est initialisé avec une population de solutions aléatoires. Puis chaque solution potentielle se voit assigner une vitesse aléatoire, et les solutions potentielles, appelées particules, sont ensuite transportées dans l'hyperespace voire le figure 2.

Chaque particule garde trace de ses coordonnées dans l'hyperespace qui sont associées à la meilleure solution calculée par la fonction objective (***fitness***) atteinte jusqu'à présent. (La valeur de cette forme physique est également stockée.) Cette valeur est appelée ***pbest***. Une autre "meilleure" valeur est également suivie.

La version "globale" de l'optimiseur d'essaim de particules garde trace de la meilleure valeur globale, et de sa localisation, obtenue jusqu'ici par n'importe quelle particule dans la population ; c'est ce qu'on appelle ***gbest***.

Le concept d'optimisation des essaims de particules consiste, à chaque pas de temps, à changer la vitesse de chaque particule vers son ***pbest*** et le ***gbest***.

L'accélération est pondérée par un terme aléatoire, avec des nombres aléatoires séparés générés pour l'accélération vers ***pbest*** et ***gbest***.[2]

Donc, pour appliquer la PSO il faut définir un espace de recherche constitué de particules et une fonction objective à optimiser. Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum.

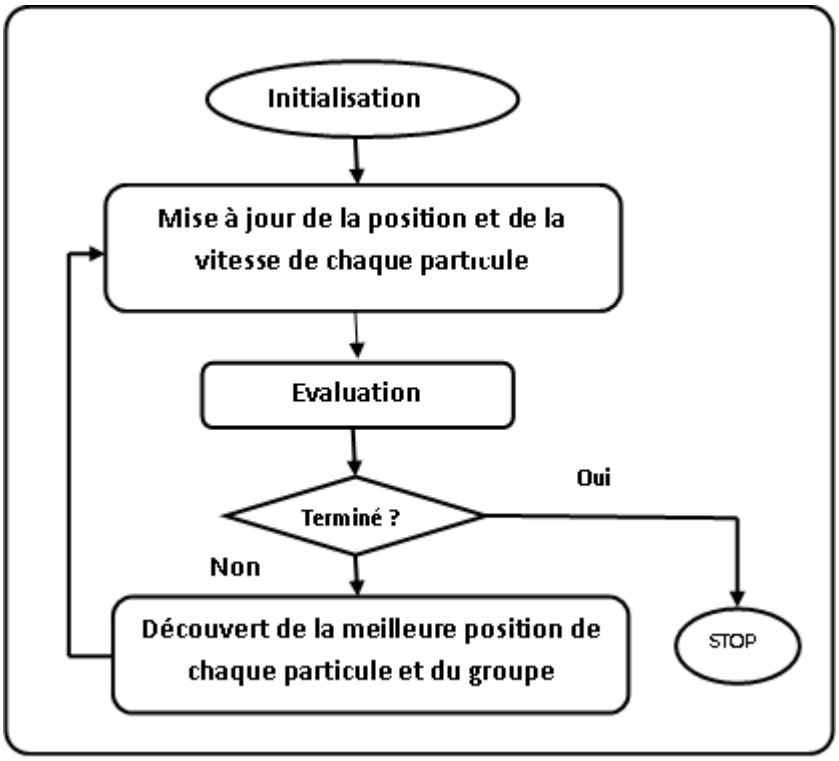


Figure 2: Organigramme de fonctionnement de l'algorithme PSO

Chacune de ces particules est dotée :

- D'une position, c'est-à-dire ses coordonnées dans l'ensemble de définition.
- D'une vitesse qui permet à la particule de se déplacer. De cette façon, au cours des itérations, chaque particule change sa position. Cette vitesse évolue en fonction de son meilleur voisin, de sa meilleure position, et de sa position précédente. C'est cette évolution qui permet de converger vers la valeur optimale.

- D'un voisinage, c'est l'ensemble de particules qui interagissent directement sur la particule, en particulier celle qui a la meilleure performance.

A tout instant, chaque particule connait :

- Sa meilleure position visitée. On retient essentiellement la valeur du fitness calculée ainsi que ses coordonnées.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnancement optimal.
- La valeur qu'elle donne la fonction objectif car à chaque itération il faut une comparaison entre la valeur du critère donnée par la particule courante et la valeur optimale.

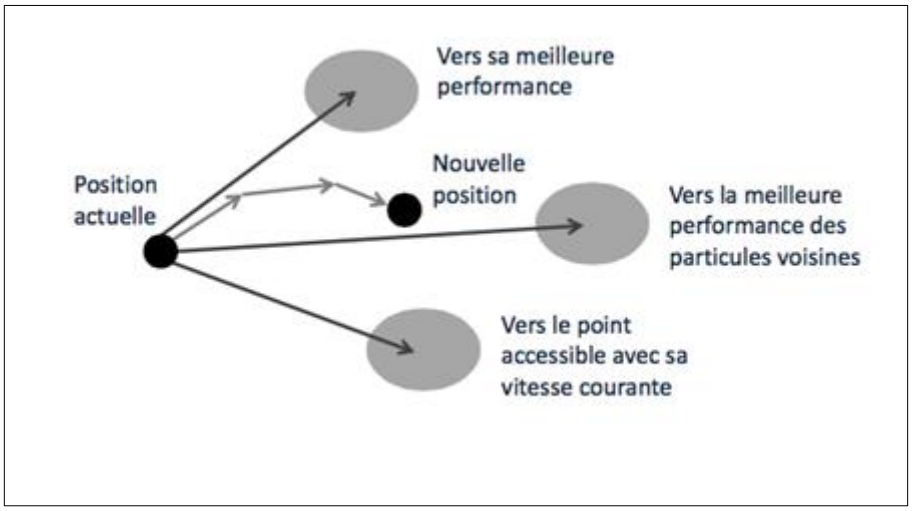


Figure 3: Déplacement d'une particule [47]

3.3 Notion de voisinage

Le voisinage d'une particule est le sous-ensemble de particules de l'essaim avec lequel elle a une communication directe. Ce réseau de rapports entre toutes les particules est connu comme *la topologie de l'essaim.*

3.4 Formulation

Dans un espace de recherche de dimension *D*, la particule *i* de l'essaim est modélisée par son vecteur position $\vec{X}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ et par son vecteur vitesse $\vec{V}\, i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. La qualité de sa position est déterminée par la valeur de la fonction *Objective* en ce point.

Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note $\vec{P}\, besti = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})$.

La meilleure position atteinte par les particules de l'essaim est notée $\vec{G}$ *best* .avec $\vec{G}$ best = ($gbest_1$, $gbest_2$, . . . , $gbest_D$).

Nous nous référons à la version globale de PSO, où toutes les particules de l'essaim sont considérées comme voisines de la particule *i*, d'où la notation $\vec{G}$ *best (Global best).*

Au départ, les particules sont initialisées d'une manière aléatoire dans l'espace de recherche du problème. Ensuite, à chaque itération, chaque particule se déplace, en combinant linéairement les trois composantes $\vec{X}_i, \vec{V}_i$ *et* $\vec{P}$ *best*$_i$

En effet, à l'itération *t+1,* le vecteur *vitesse* et le vecteur *position* sont calculés comme suite :

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_{1i,j}^{t} [\, pbest_{i,j}^{t} - x_{i,j}^{t} \,] + c_2 r_{2i,j}^{t} [\, gbest_{j}^{t} - x_{i,j}^{t} \,], j \in \{1,2,\ldots,D\} \quad (1)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \; j \in \{1,2,\ldots,D\} \quad (2)$$

Où : W est une constante, appelée coefficient d'inertie ; c_1 et c_2 sont deux constantes, appelées coefficients d'accélération ;

r_1 et r_2 sont deux nombres aléatoires tirés uniformément dans [0, 1], à chaque itération t et pour chaque dimension j.

Une fois le déplacement des particules effectué, les nouvelles positions sont évaluées et les deux vecteurs $\vec{P}$ *best*$_i$ et $\vec{G}$ *best* sont mis à jour, à l'itération (*t+1),* suivant les deux équations(3) (dans le cas d'une minimisation) et (4) (dans une version globale de PSO), respectivement.

$$\vec{P}best_i(t+1) = \begin{cases} \vec{P}best_i(t), si \quad f(\vec{x}_i(t+1)) \geq \vec{P}best_i(t) \\ \vec{x}_i(t+1) \quad \text{sin } on \end{cases} \quad (3)$$

$$\vec{G}best_i(t+1) = \arg\min_{\vec{P}best}(\vec{P}best_i(t+1)), 1 \leq i \leq N \quad (4)$$

3.5 Algorithme PSO

Après cette formulation, l'algorithme PSO de base peut être écrit comme suite :

Où N est le nombre des particules de l'essaim :

1 **Initialiser** aléatoirement N particules : position et vitesse.
2 **Evaluer** les positions des particules
3 Chaque particule i, $\vec{P}$ best$_i$ = $\vec{X}$ i
4 **Calculer** $\vec{G}$ best selon la formule **(4)**
5 **Tant que** le critère d'arrêt n'est pas satisfait faire
6 **Déplacer** les particules selon les formules **(1)** et **(2)**
7 **Evaluer** les positions des particules
8 **Mettre à jour** $\vec{P}$ best$_i$ et $\vec{G}$ best selon les formules **(3)** et **(4)**
9 Fin

Figure 4: Algorithme général de PSO

3.6 Variantes et hybridation de la PSO

3.6.1 Variantes de PSO

De nombreuses variantes de même algorithme de base PSO sont possibles juste en jouant sur les paramètres et les composantes de base de l'algorithme, Par exemple, sur la façon d'initialiser les particules et les vitesses, réduire la vitesse, mettre à jour P_{best} et G_{best} après la mise à jour de l'essaim, etc.

Dans ce contexte, et à titre d'exemple, trois modifications ont été proposé par [43] basées sur la taille de la population (particules) de telle sorte de rendre l'algorithme dynamique en ajoutant et supprimant les particules. Cette dynamique affecte considérablement les performances et augmente la capacité à trouver l'optimum global.

La deuxième idée est basée sur la stratégie dite *solution-sharing strategy*, qui permet aux meilleures particules de partager leurs informations et de mettre à jour leurs vitesses. La troisième idée porte sur la technique dite *searching range sharing* (SRS), qui empêche les particules de tomber dans un optimum local.

Une autre approche d'amélioration de la méthode PSO c'est *La PSO coopérative* qui a pour objectif de minimiser l'augmentation exponentielle de la difficulté d'optimisation pour les problèmes à dimensions élevées en considérant chaque dimension comme un problème unidimensionnel. Plusieurs stratégies de coopération ont été développées, où les petits essaims prennent en charge chaque dimension et une dimension transversale de communication permet à la solution globale de progresser vers l'objectif.

3.6.2 Hybridation

De nombreux algorithmes ont été proposés dans la littérature et utilisés pour résoudre les différents types de problèmes d'optimisation.

A titre d'exemple : l'algorithme génétique, le recuit simulé, la recherche taboue, la colonie de fourmis, PSO et autres. Mais les résultats de ses méthodes prouvent qu'aucune parmi eux n'est efficace à 100% pour la résolution de tous les problèmes d'optimisation.

En effet, chaque méthode a ses avantages et ses inconvénients (ou lacunes). Alors que les chercheurs ont envisagé la combinaison des méthodes de résolution des problèmes d'optimisation, afin de profiter des avantages de chaque méthode et de combler certaines de ses lacunes. Par conséquent, ils ont donné naissance à une nouvelle classe de méthodes de résolution de problèmes d'optimisation : c'est la classe des méthodes hybrides [37].

Parmi les hybridations les plus populaires avec la PSO et qui ont donné des résultats très prometteurs on peut citer à titre d'exemple :

Le PSO et l'algorithme génétique a donné naissance du MGPSO[46].
Le PSO et l'optimisation par colonies de fourmis a donné naissance du PSACO, [50]
Le PSO et le recuit simulé a donné naissance du BPSO[55].
Le PSO et la recherche tabou a donné naissance du HPSOTS[54].
Le PSO et la recherche à voisinage variable a donné naissance du VNPSO[53].
Le PSO et le système immunitaire.[29].
Le PSO et la méthode simplexe de Nelder-Mead.[26].
Le PSO et la méthode GRASP Greedy Randomized Adaptive Search Procedure).[51].
La PSO et 'algorithme Differential Evolution a donné naissance DEPSO[52].
La PSO et l'algorithme K-MENS a donné naissance KPSO[44], PSO-KM[45].

Les hybridations citées ne sont qu'à titre d'exemple par ce qu'ils existent en un nombre important de tentations de combinaison da la PSO avec d'autre méthodes.

4 Système Multi Agents SMA

4.1 Introduction

Un système multi-agent ou SMA est un système de plusieurs entités de négociation et de coordination, similaires ou différemment, spécialisées, des entités logicielles qui résolvent collectivement un problème. Ces entités sont appelées « agents »

Les systèmes multi-agents existent à la fois en biologie (systèmes multi-agents naturels) et en technologie. Un échantillon des systèmes multi-agents biologiques sont les colonies de fourmis.

En réalité, La notion d'agent est utilisée dans beaucoup de domaines : biologie, sociologie, psychologie cognitive, psychologie sociale, et informatique.

4.2 Un agent

Il existe plusieurs définitions pour un agent, selon [6] : "Un agent est tout ce qui peut être perçu comme percevoir son environnement à travers des capteurs et agissant sur cet environnement à travers des capteurs".

Une autre définition a été donnée par [7] : "Les agents sont des systèmes informatiques qui habitent un environnement dynamique complexe, ils sentent et agissent d'une manière autonome dans cet environnement, et réalisent ainsi un ensemble de buts ou de tâches pour lesquels ils sont daignés".

Une autre définition donnée par IBM pour un agent intelligent : "Les agents intelligents sont des entités logicielles qui réalisent des opérations à la place d'un utilisateur ou d'un autre programme, avec une sorte d'indépendance ou d'autonomie, et pour faire cela, ils utilisent une sorte de connaissance ou de représentation des buts ou des désirs de l'utilisateur."

4.3 Caractéristiques des agents

Un agent est caractérisé par :

- **Situé** : dans un environnement.
- **Autonome :** agir sans intervention externe.
- **Proactif** : prendre l'initiative au bon moment (opportuniste).
- **Réactif :** une réponse en bon moment.
- **Social** : interagir avec des autres agents.

4.4 Typologie des agents

En général on distingue deux grandes type d'agents, cette catégorisation dépend essentiellement du niveau de l'intelligence de l'agent, ces catégories sont :

4.4.1 Agent réactif

Ce type d'agent agit en se basant uniquement sur ses perceptions courantes. Il utilise un ensemble finis de règles afin de choisir son action, Il présente l'avantage d'être fort et simple mais en pratique il est très limité.

Une dérivation plus évoluée de ce type, ce sont des agents qui conservent une trace du monde en utilisant ses informations internes pour mettre à jour ses perceptions actuelles à savoir :

- L'état précédent de l'environnement.
- L'évolution de l'environnement.
- L'impact de ses actions.

Sont les agents **délibératifs**, qui possèdent :

- Une description de l'état actuel de son environnement.
- Des informations décrivant ses buts.
- Une projection sur le futur.
- Beaucoup plus de flexibilité

4.4.2 Agent BDI

Une architecture BDI est conçue en partant du modèle **"Croyance-Désir-Intention"**, en anglais "**Belief-Desire-Intention**", de la rationalité d'un agent intelligent.

Belief (Croyance) : Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement. Elles peuvent être incorrectes, incomplètes ou incertaines, et à cause de cela, elles sont différentes des connaissances de l'agent, qui sont des informations toujours vraies. Les croyances peuvent changer au fur et à mesure par l'agent,

Par sa capacité de perception ou par l'interaction avec d'autres agents, l'agent recueille plus d'informations.

Desire (Désir) : Les désirs d'un agent représentent les états de l'environnement, et parfois les états que l'agent aimerait voir réalisés lui-même.

Intention (Intention) : Les intentions d'un agent sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs. Même si tous les désirs d'un agent sont consistants, l'agent peut ne pas être capable d'accomplir tous ses désirs à la fois.

4.5 Systèmes multi agents SMA

Un système multi-agents est un système distribué composé d'un ensemble d'agents. Ces agents agissent sur autres paramètres qui sont :

- Un environnement E, c'est l'espace disposant généralement d'une métrique.
- Un ensemble d'objets O situé dans E. Ces objets peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble de relations R qui relient des agents entre eux.
- Un ensemble d'opérations Op permettant aux agents de percevoir, produire, consommer, transformer et manipuler des objets.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction de l'environnement envers les tentatives de modification

4.6 Caractéristiques d'un SMA

Généralement, un SMA se caractérise par :

- Les agents agissent et travaillent indépendamment les uns des autres.
- Chaque agent est une partie du système.
- Chaque agent travaille dans le but d'accomplir ses tâches.
- Chaque agent est capable de communiquer et d'interagir avec d'autres agents.
- Un agent coopère avec les autres agents lorsque c'est nécessaire.
- Un agent est capable de coordonner ses activités avec les autres agents.
- Les agents ont un but commun.
- Chaque agent a une vue partielle du SMA.

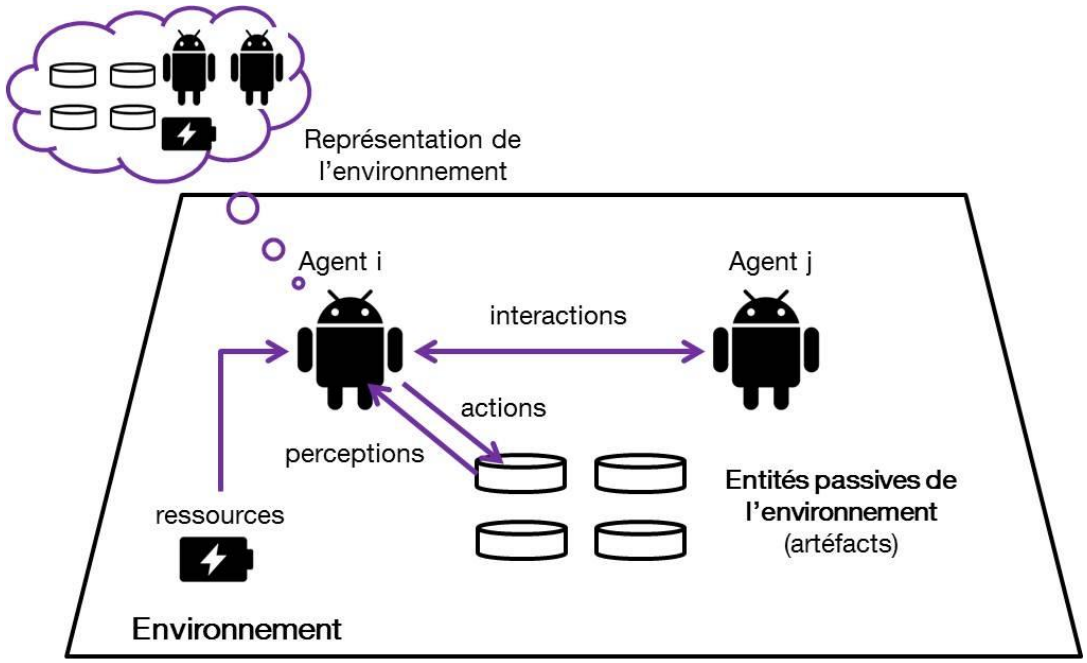


Figure 5: Modèle général d'un Système Multi-Agents [48]

4.7 Implémentation des SMAs

Tous d'abord, on pense au SMA dans les cas de programmes assez longs, avec des problèmes spécifiques (ex. parallélisme, synchronisation, interactions, ...) avec un besoin d'un certain niveau d'intelligence collectives et possibilité d'apprentissage,

Plusieurs plateformes existantes soit de développement ou de simulation tels que : GAMA, JASON, JADE, Repast, CorMAS, Netlogo, Mason, Madkit, JaCoMo, etc.), avec une variété de langages de programmation possibles (ex. Java ou Python).

On doit définir une topologie d'interaction (c.-à-d. l'environnement spatial, réseau social, …) pour savoir comment les agents communiquent.

On doit définir des mécanismes pour activer (pro activement) les agents, les créer, déployer et démarrer une simulation, les regrouper ou les dissocier (dynamiquement), etc.

On doit mettre en œuvre des objets permettant de visualiser le SMA, pour récupérer et stocker des données. [30]

4.8 Avantages de l'utilisation du SMA

L'utilisation des SMA peut apporter beaucoup d'avantages à la programmation, et à la résolution des problèmes complexes, on peut citer quelques avantages attribués aux SMA :

- ✓ Les SMA améliorent la performance du programme par un travail parallèle des agents.

✓ La facilité de « Passage à l'échelle d'une architecture », car plusieurs agents peuvent s'ajouter ou se retirer dynamiquement d'un système.

✓ Plus un logiciel est modulaire, plus la complexité et les coûts de développement diminuent.

✓ Les SMA reflètent la réalité, la majorité des problèmes dans la réalité sont distribués, ce qui s'adapte facilement aux SMA.

✓ Diversité, Les SMA peuvent avoir parmi les agents qui les constituent, une grande diversité, ce qui donne la possibilité aux concepteurs d'intégrer différents agents (réactifs, cognitifs …etc.)

✓ A chaque agent sa façon de résoudre les problèmes, donc un même problème peut avoir différentes solutions selon les agents.

5 Les graphes et la représentation des réseaux sociaux

5.1 Introduction

Trouver une représentation adéquate et efficace pour présenter et interpréter les données d'un réseau social est une tâche très importante dans les études et l'analyse de ses derniers.

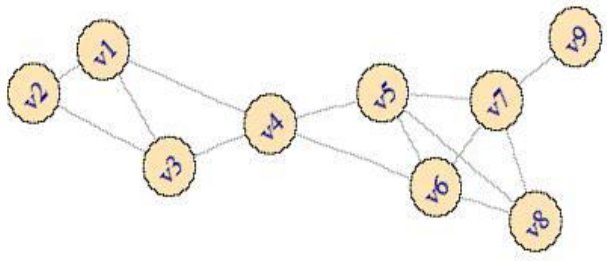
Les réseaux sociaux conçoivent des relations sociales en termes de nœuds et liens. Les nœuds sont les acteurs sociaux dans le réseau, et les liens sont les interactions ou des relations entre ces nœuds.

Les réseaux sociaux comme les graphes sont donc un ensemble de nœuds interconnectés, d'où les graphes sont des outils visuels puissants et des moyens formels pour représenter les réseaux sociaux.

5.2 Représentation graphique des réseaux sociaux

Un réseau social peut être représenté par un graphe G(V,E) où V (vertex) représente l'ensemble des sommets (Nœuds), E l'ensemble des arêtes (Edge).

Comme il peut être représenté à l'aide d'une matrice dite *matrice d'adjacence* (A_{ij}) qui indique les connexions entre les nœuds.



S	v1	v2	v3	v4	v5	v6	v7	v8	v9
v1	0	1	1	1	0	0	0	0	0
v2	1	0	1	0	0	0	0	0	0
v3	1	1	0	1	0	0	0	0	0
v4	1	0	1	0	1	1	0	0	0
v5	0	0	0	1	0	1	1	1	0
v6	0	0	0	1	1	0	1	1	0
v7	0	0	0	0	1	1	0	1	1
v8	0	0	0	0	1	1	1	0	0
v9	0	0	0	0	0	0	1	0	0

Figure 6 : Représentation graphique et par matrice d'adjacence [9]

5.3 Définitions de base sur les graphes

Soit un graphe G(V,E), tel que **V** l'ensemble des sommets [V_i, i = 1, …, n] et **E** l'ensemble des arêtes [$e_{ij} = e(V_i,V_j)$, i,j= 1,…, n].

5.3.1 Chemin

Un chemin permettant d'aller de v_i à v_j est la succession de sommets et arêtes reliant les deux sommets.

5.3.2 Boucle

Arête reliant un sommet à lui-même.

5.3.3 Voisinage

Le voisinage N_i d'un sommet c'est l'ensemble des nœuds qui lui sont directement connectés. Ex $N_3=3$.

5.3.4 Distance géodésique δij

Nombre d'arêtes du plus court chemin reliant deux sommets. Dans l'exemple précédant $\delta 1,7 = 3$.

5.3.5 Graphe connexe

Un graphe dite connexe s'il existe au moins un chemin entre chaque paire de nœuds. Autrement dit, Il n'y a pas de sommets isolés.

5.3.6 Diamètre d'un graphe

C'est La plus grande distance géodésique possible entre 2 sommets dans le graphe. Dans l'exemple le diamètre du graphe est égale à $\delta 2,9 = 5$.

5.3.7 Densité d'un graphe

Rapport entre le nombre d'arêtes observées et le nombre maximal d'arêtes possibles. La densité est nulle si tous les sommets sont isolés ; et égale à 1 si le graphe est complet (Il existe un lien entre chaque paire de sommets).

$$Densité = \frac{|E|}{\frac{|V|.(|V|-1)}{2}}$$

5.3.8 Graphes Non Orienté

Dans un graphe non orienté, l'ordre des sommets connectés d'un bord n'est pas important. Nous nous référons à chaque lien par un couple des nœuds i et j tels que e (i, j), i et j sont les nœuds d'extrémité du lien.

Les réseaux sociaux peuvent être modélisés comme des graphes non orientés lorsque les relations entre les nœuds sont mutuelles tel que la relation d'amitié sur Facebook.

5.3.9 Graphe Orienté

Un graphe orienté est défini par un ensemble de nœuds et un ensemble d'arêtes dirigées où l'ordre des deux nœuds est important : e (i, j) désigne le lien de i à j.

Pour indiquer graphiquement la direction des liens, les bords dirigés sont représentés par des flèches.

Les réseaux sociaux peuvent être modélisés comme des graphes orientés lorsque les relations entre les nœuds ne sont pas mutuelles (bidirectionnelles) tel que la relation « Suivant » sur Twitter, on dit que l'utilisateur i suit l'utilisateur j si e (i, j).

5.3.10 La pondération dans les graphes

Les poids dans un graphe représentent la force des relations entre les acteurs du réseau social. Lorsque les graphiques sont pondérés, cela signifie que leurs bords sont assignés avec un poids numérique, ***w***, qui peut fournir diverses indications telles que la capacité de liaison, la force de liaison, le niveau d'interaction ou la similarité entre les nœuds.

5.4 Partitionnement d'un graphe

5.4.1 Définition

Le partitionnement de graphe est la tâche qui consiste à diviser un graphe orienté ou non orienté en plusieurs parties.

Soit un graphe G(S;A). On peut chercher :

- Une partition de l'ensemble des sommets S.
- Une partition de l'ensemble des arêtes A.

En générale, on entend par partition d'un graphe la partition de l'ensemble de ses sommets.

5.4.2 Méthodes de partitionnement des graphes

Tous dépendent au problème à résoudre, ils existent plusieurs méthodes de partitionnement des graphes telles que [27] :

- Méthodes inertielles.
- Classification hiérarchique.
- Expansion de région.
- Méthode spectrale.
- Multi-niveaux.

- Variantes de l'algorithme de Kernighan-Lin.
- Analogie avec la mécanique du fluide : diffusion et percolation.
- Méta heuristiques.
- Méthodes hybrides.
- Autres…

5.4.3 Modularité d'un graphe

La modularité est une mesure pour la qualité d'un partitionnement des nœuds d'un graphe, ou d'un réseau, en communautés (sous graphes).

Elle est introduite par M. E. J. Newman [36], avec un principe simple, c'est qu'un bon partitionnement d'un graphe implique un nombre d'arêtes intra-communautaires important et un nombre d'arêtes intercommunautaires faible.

Elle est égale à la différence entre le nombre de liens présents dans un sous-groupe (ou communauté), et le nombre de liens attendus dans un graphe aléatoire. La valeur de la modularité est dans l'intervalle [-1;1].

On calcule une valeur de modularité pour un partitionnement donné c_i du graphe, on note souvent la valeur de la modularité par la lettre Q.

Elle se calcule avec la formule suivante :

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{K_i * K_j}{2m} \right] \Delta(c_i, c_j)$$

Avec :

A_{ij} : valeur de la matrice d'adjacence entre les sommets i et j,

k_i : Somme des poids des arêtes adjacentes à i,

m : Nombre d'arêtes du graphe.

Δ : est le Delta de Kronecker qui est à 1 si c_i et c_j sont égaux, et à 0 sinon.

6 Le regroupement des données (CLUSTERING)

6.1 Introduction

Le clustering ou bien Le regroupement des données est l'une des techniques d'analyse des données les plus populaires dans l'exploration de données. Elle est une partie fondamentale du Data Mining. Contrairement à la classification, le clustering est une méthode descriptive, par contre, la classification est prédictive.

Il s'agit d'un processus de partitionnement d'un ensemble de données non marquées en groupes (apprentissage non supervisé), chaque groupe contenant des objets qui sont similaires à un autre par rapport à une certaine mesure de ***similarité***. L'objectif du regroupement est de découvrir un nouvel ensemble de catégories dans l'ensemble de données. [14].

6.2 Fonctions et mesures de similarité

Le choix des mesures de similarité est une étape critique dans le Clustering. Il définit comment la similarité de deux éléments (X, Y) est calculée et influencera sur la forme des groupes.

Soit la fonction $S(x_i; x_j)$ qui compare les deux vecteurs x_i et x_j. Cette fonction devrait être symétrique (c à d $S(x_i; x_j) = S(x_j; x_i)$) et avoir une grande valeur lorsque x_i et x_j sont en quelque sorte «similaires» et constituent la plus grande valeur pour des vecteurs identiques [14].

Du point de vue mathématique, c'est par la différence de distance entre deux données qu'on mesure leur degré de similarité.

Une fonction de similarité qui donne des valeurs entre 0 et 1 est appelée ***dichotomique***.

Plusieurs variantes des fonctions de similarités peuvent être utilisé tout dépend du problème en question ainsi que la plage de données à traiter.

Nous donnons dans ce qui suit, les définitions de quelques exemples de mesures de similarité les plus connues, appliquées sur deux vecteurs X et Y, de dimension N.

6.2.1 Similarité par distances Euclidienne

La méthode classique pour les mesures de distance entre X et Y est la distance Euclidienne, elle est définie comme suite :

$$S(x, y) = 1 - \sqrt{\sum_{i=1}^{N} (xi - yi)^2}$$

6.2.2 Similarité du cosinus

Pour calculer la similarité du cosinus on doit calculer le produit scalaire des deux vecteurs et le diviser par le produit par les normes des deux vecteurs. Formellement :

$$S(x_i, x_j) = \frac{x_i^T . x_j}{\|x_i\| . \|x_j\|}$$

6.2.3 Mesure de corrélation de Pearson

La corrélation de Pearson normalisée est définie comme suite :

$$S(x_i, x_j) = \frac{(x_i - \bar{x}_i)^T . (x_j - \bar{x}_j)}{\|x_i - \bar{x}_i\| . \|x_j - \bar{x}_j\|}$$

6.2.4 Coefficient de Jaccard généralisé

Le coefficient de Jaccard est défini comme étant le quotient du cardinal de l'intersection par celui de l'union.

Il a été présenté dans [15] par (Strehl et Ghosh,) et il est défini ainsi :

$$S(x_i, x_j) = \frac{x_i^T . x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i^T . x_j}$$

6.2.5 Mesure du coefficient de dés

La mesure du coefficient de dés est similaire à la mesure Jaccard étendue, il est défini par :

$$S(x_i, x_j) = \frac{2x_i^T . x_j}{\|x_i\|^2 + \|x_j\|^2}$$

6.3 Choix du type de mesure de similarité

Le choix des mesures de similarité est très important, car il a une forte influence sur les résultats de la catégorisation. Pour la plupart des logiciels de clustering courants, la mesure de similarité par défaut est par la distance euclidienne.

Selon le ***type de données*** et les ***questions du chercheur***, d'autres mesures de similarité pourraient être préférées. Par exemple, la distance basée sur la corrélation est souvent utilisée dans l'analyse des données d'expression génique.

Si nous voulons identifier les groupes d'observations avec les mêmes profils globaux indépendamment de leurs grandeurs, alors nous devrions aller vers la distance basée sur la corrélation comme une mesure de similarité. C'est particulièrement le cas dans l'analyse des données d'expressions géniques, où nous pourrions vouloir considérer des gènes semblables quand ils sont « en haut » et « en bas » ensemble. C'est aussi le cas, en marketing, si l'on veut identifier un groupe d'acheteurs ayants la même préférence en termes d'articles, quel que soit le volume d'articles qu'ils ont acheté [16].

Pour mesurer la similarité ou la dissemblance entre les objets qui sont exprimés par des variables numériques telles que l'âge, la taille, des nombres, etc.., des distances telles que la distance euclidienne, la distance de Manhattan, la distance de Chebyshev, etc., sont utilisées [22].

Cependant, pour représenter des distances simples entre les variables, de même catégorie comme les couleurs, les familles d'animaux, etc., le choix se tournera vers la distance Jaccard ou Hamming [23].

6.4 Méthodes de Clustering

6.4.1 Clustering par partitionnement

Ce sont des méthodes de regroupement utilisées pour classer les observations, dans un ensemble de données, en plusieurs groupes en fonction de leur similarité. Les algorithmes nécessitent que l'analyste spécifie le nombre de grappes à générer.

Les méthodes de partitionnement aboutissent généralement à un ensemble de K clusters, chaque objet appartenant à un cluster. Chaque groupe peut être représenté par un centroïde ou un représentant de groupe [17]. L'algorithme le plus populaire de cette approche est le *K-Means*.

6.4.2 Méthodes hiérarchiques Agglomératives

Les méthodes de regroupement hiérarchique par agglomération sont les plus couramment utilisées.

Cet algorithme de classification hiérarchique peut être affiché sous la forme d'un arbre, appelé ***dendrogramme***.

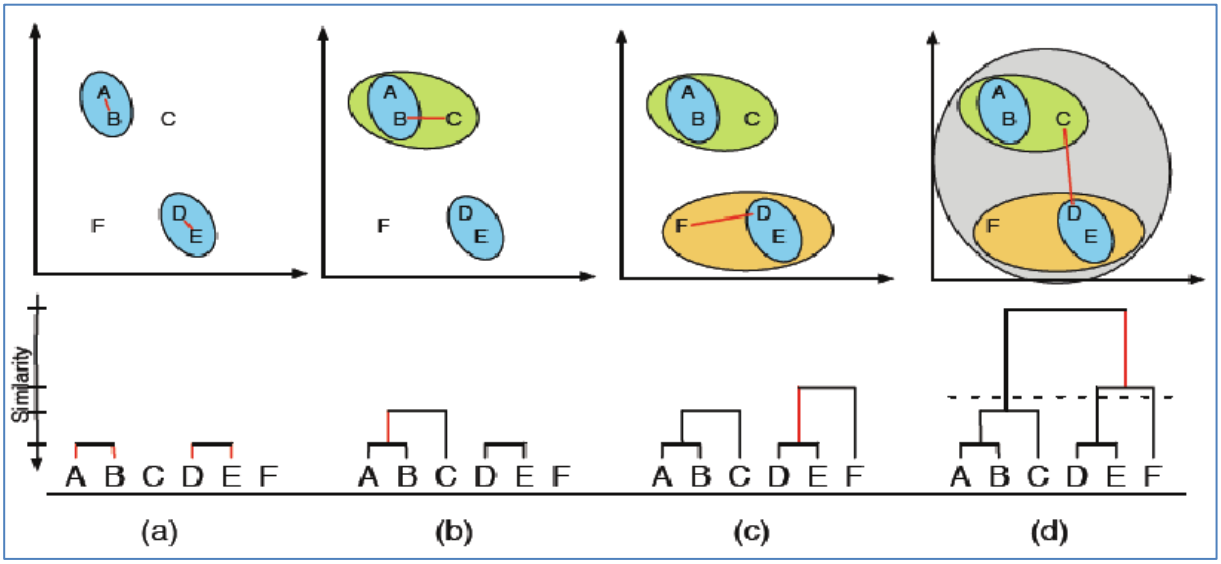


Figure 7: Dendrogramme et différentes étapes d'un algorithme hiérarchiques Agglomératives [24]

6.4.3 Clustering basé sur la densité (par voisinage dense)

Dans les méthodes de clustering basés sur la densité, les groupes sont définis comme des zones de densité plus élevée que le reste de l'ensemble de données. Les objets dans ces zones clairsemées - qui sont nécessaires pour séparer les clusters - sont généralement considérés comme des points de bruit et de frontière [17].

6.5 Avantage et inconvénients des algorithmes de Clustering

Les techniques hiérarchiques ont deux avantages fondamentaux. L'un est que le nombre de classes n'a pas besoin d'être spécifié à priori, et qu'elles sont indépendantes des conditions initiales. Cependant, le principal inconvénient des techniques de classification hiérarchique est qu'elles sont statiques ; ce qui signifie que les points de données affectés à un cluster ne peuvent pas être déplacés vers un autre cluster. En outre, ils peuvent ne pas séparer les clusters qui se chevauchent en raison d'un manque d'informations sur la forme ou la taille globale des clusters.

Avec le clustering par partitionnement, l'algorithme ne crée qu'un seul ensemble de clusters. Cette approche utilise le nombre de clusters souhaité et fixée au début pour créer l'ensemble final.

Les avantages des algorithmes hiérarchiques se trouvent dans les inconvénients des algorithmes par partitionnement, et vice versa. [10].

Il a été montré que la technique de clustering par partitionnement est bien adaptée au regroupement d'un grand ensemble de données en raison de leurs exigences de calcul relativement faibles [18] [19]. La complexité temporelle de cette technique est presque linéaire, ce qui la rend largement utilisable. L'algorithme de clustering de partitionnement le plus connu est l'algorithme K-means et ses variantes [20].

Bien qu'il s'agisse d'un algorithme de clustering simple et très utile, K-means souffre de beaucoup de lacunes ; La plus importante est que le résultat de regroupement dépend du choix initial du nombre et de des centres des clusters choisis au départ.

7 Conclusion

Dans ce chapitre nous avons passé en revue bref sur les principaux éléments de base de notre projet, tout d'abord, nous avons parlé des réseaux social et les principaux dangers liés à leurs utilisations, puis on a parlé sur la PSO comme l'une des méthodes méta heuristiques les plus prometteur : son principe général, sa formulation et son algorithme standard ainsi qu'un bref aperçu sur ses variantes et hybridations, Puis nous avons présenté le paradigme Multi agent et les définitions relatives : notion d'agent, leurs typologies, le système multi agent et son implémentation, Ensuite nous avons passé un tour d'horizon sur la notion de graphe et les méthodes de partitionnement des graphes ainsi que la fonction de mesure de qualité Modularité, et en fin nous avons présenté les concepts de base et les technique de clustering, la notion de similarité, et les critères de la choisir.

Nous allons donner dans le chapitre suivant un état de l'art sur les travaux réalisés qui nous semblent être bénéfiques dans l'élaboration de notre méthode.

Chapitre 2

Etat de L’art

1 Introduction

Nous présenterons dans ce chapitre un aperçu sur les travaux réalisés auparavant qui ont une relation à notre sujet.

Nous nous intéressons à trois points principaux, à savoir : les techniques de clustring à base de la PSO, l'utilisation des systèmes multi agents avec la PSO, et nous finissons par les techniques existantes pour la détection des communautés dans les réseaux, et particulièrement à celles qui sont à base de PSO.

2 Etat de l'art sur le Clustering à base de PSO

Beaucoup de recherches basées sur les métas heuristiques et les techniques bio-inspirés ont été faites pour le regroupement des données jusqu'à présent, et l'optimisation de l'essaim de particules a pris la part du lion de ces études.

Sunita Sarkar et Arindam Roy, Bipul Shyam Purkayasth dans [21] ont fait un Sandage sur les différents travaux réalisés au sujet de techniques de regroupement des données à base des algorithmes bio-inspiré et en particulier l'algorithme PSO.

D'après [21], les algorithmes basés sur les essaims (PSO) sont les plus prometteurs car ils sont en train de devenir une alternative aux techniques de regroupement conventionnelles.

Le clustering de données avec des algorithmes PSO a récemment démontré de bons résultats dans une grande variété de données réelles.

Ahmady fard et Modares, dans [49] ont examiné l'application de la PSO pour regrouper les vecteurs de données. Deux algorithmes ont été testés, à savoir un algorithme PSO standard et une approche hybride où les individus de l'essaim sont ensemencés par le résultat de l'algorithme K-means. Les deux approches PSO ont été comparées à la classification de K-means, ce qui montre que les approches PSO ont une meilleure convergence pour réduire les erreurs de quantification, et en général, des distances inter-cluster plus grandes et des distances intra cluster plus petites.

Van der Merwe , et AP Engelhrecht, dans [45] ont proposé un autre algorithme de clustering, qui est un hybride de PSO et de K-means, nommé algorithme PSO-KM. Dans cet algorithme PSO est initialement appliqué pour rechercher une solution globale. Lorsque cette solution est trouvée, l'algorithme de clustering K-means est utilisé pour une convergence plus rapide afin de terminer le processus de clustering.

Premalatha et al. dans [46] ont proposé une approche hybride d'optimisation de l'essaimage de particules (PSO) - algorithme génétique (GA) pour la classification des documents afin de surmonter le problème d'être piégé dans une zone optimale locale dans le cas de donnée de grande dimension, Ce mécanisme hybride des modèles PSO et GA améliore le processus de recherche, en améliorant la diversité ainsi que la convergence.

Dans cette méthode, l'opération de croisement de GA est appliquée pour échanger des informations entre deux particules et l'opération de mutation est appliquée à PSO pour augmenter la diversité de la population.

Stuti Karol, Veenu Mangat ont proposé dans [44] deux méthodes de clustring hybride entre K-Means, Fuzzy-C Means et PSO, Cette hybridation a donné de meilleurs résultats par rapport à d'autres algorithmes. Ces deux méthodes sont appelées KPSO et FCPSO.

FCPSO donne des résultats encore meilleurs que KPSO car il traite bien le chevauchement des documents La performance varie également pour les deux ensembles de données. Bien que la vitesse de convergence de KPSO soit meilleure que FCPSO, Mais ce dernier reste comme la meilleure technique, car elle donne les meilleurs résultats pour les mesures d'évaluation entropie et F-Measure qui sont des mesures externes standard et sont plus importantes pour juger la validité des groupes de documents.

Pour plus d'informations sur ce sujet, se référer aux documents suivants : [10] [21] [44]

3 Utilisation PSO-SMA

En premier coup, une grande similarité est observée entre le PSO et le SMA, cela parce qu'ils sont tous les deux des approches basées sur la population et accomplissent les tâches en coopération. Cependant, les agents sont assez différents des particules d'une manière très spécifique et cela pour les trois raisons suivantes [8] :

1. Une particule n'est pas considérée en général comme autonome car elle ne peut que se déplacer dans l'espace du problème selon l'algorithme principal, par contre un agent est capable d'explorer l'environnement avec beaucoup plus de flexibilité.
2. Les particules sont volontairement définies comme moins intelligents pour obtenir des performances de calcul en réduisant leurs capacités. D'autre part, les agents sont capables de faire l'apprentissage, comme forme d'intelligence, et c'est l'un des principales caractéristiques des agents.

3. Enfin, PSO applique l'exécution synchronisée afin de maintenir la simplicité dans la conception. Cependant, à cause des composants d'autonomie, d'apprentissage et de coopération, les agents dans SMA sont naturellement exécutés de manière asynchrone.

Par conséquent, une approche qui combine la simplicité du PSO avec l'autonomie et l'apprentissage du SMA va bénéficier des deux.

Plusieurs approches ont été proposées, en introduisant les agents en PSO afin de bénéficier des deux outils ; Nous présentons ici l'une de ces approches qui a été proposée par A.Raheel ,L.Yung-Chuan et R.Shahram dans [8] et ce résume comme suite :

En intégrant l'autonomie et l'apprentissage dans le PSO original, une nouvelle approche PSO modifiée est née, appelée Agent-based PSO (APSO), où chaque particule est maintenant appelée Agent.

L'environnement est lui-même modélisé en tant qu'agent et est chargé de fournir des informations supplémentaires sur l'espace de problème aux agents. Et en introduisant une valeur booléenne et l'associer à chaque point de l'espace de problème pour identifier si ce point a déjà été visité par un agent, si le point a été visité, alors il ne doit pas être une solution.

Comme dans le PSO d'origine, les particules recherchent la solution optimale en parcourant l'espace du problème dans un environnement multidimensionnel. Par conséquent, ces chercheurs visent dans cette approche, de limiter le nombre de points non visités qu'une particule peut éventuellement parcourir à l'avenir, cela peut augmenter l'efficacité dans un grand espace de problème.

L'environnement est transformé d'un environnement statique à un environnement dynamique où les points de l'espace du problème sont régulièrement marqués par des particules après leur visite pour la première fois, afin de réduire encore le risque que des particules visitent les points marqués, pour cela l'agent de l'environnement applique un algorithme de clustering basé sur la densité pour découvrir le nuage de points marqués.

Chaque agent peut demander des informations de cluster à l'agent d'environnement pour rester au courant aux changements éventuels.

4 Détection des communautés sur les réseaux sociaux

Différentes approches ont été proposées dans le domaine de la détection communautaire sur les réseaux sociaux, ces méthodes se basent souvent, selon l'étude menée par l'auteur de [25],

soit sur l'analyse des publications et les différentes données qui peuvent être explicites ou implicites fournisses par le réseau où bien sur les relations existantes entre les acteurs du réseau (relations d'amitié ou autres).

Une troisième approche hybride proposée par le même auteur, qui tente de combiner entre les deux autres approches en exploitant les liens entre les acteurs et les attributs décrivant les sommets.

La première méthode utilise l'apprentissage non supervisé, appelée aussi classification de données vectorielles, qui exploitent les attributs décrivant les objets, comme la classification hiérarchique ou le k-means.

La deuxième méthode consiste à créer une partition de sommets, en tenant compte des relations qui existent entre les sommets du graphe, de telle sorte que les communautés soient composées de sommets fortement connectés. Pour cela on utilise des méthodes qui optimisent une fonction de qualité pour évaluer la qualité d'une partition donnée, comme la modularité, la coupe ratio, la coupe min-max ou la coupe normalisée [25], les techniques hiérarchiques comme les algorithmes de division, les méthodes spectrales ou l'algorithme de Markov et ses extensions, et la méthode k-clique percolation [28].

Ces techniques de partitionnement de graphes sont très utiles pour détecter des composantes fortement connectées dans un graphe d'une façon générale.

Concernant les méthodes hybrides qui combinent les méthodes précédentes, [25] a proposé trois approches dans ce sens :

Partitionnement orienté relations appliqué sur un graphe valué (pondéré).

Dans cette méthode, les attributs sont utilisés pour obtenir un graphe valué. Il définit une distance portant sur les attributs textuels dis_T, par exemple la distance euclidienne ou la distance du cosinus, bien adaptée aux attributs textuels.

La valeur dis_T (d_i,d_j) est associée à chaque arête (v_i,v_j) de E. Puis, une méthode d'optimisation de la fonction de qualité (modularité) compatible avec les graphes valués, est utilisée pour partitionner l'ensemble des sommets V.

Partitionnement de données vectorielles appliqué sur la distance relationnelle

Dans cette méthode, les informations relationnelles sont utilisées pour définir une mesure de *dissimilarité* dis_s (v_i,v_j) entre chaque paire de sommets (v_i,v_j) dans le graphe.

Pratiquement, pour un graphe non pondéré la dissimilarité dis_s (vi,vj) peut être prise comme la longueur du chemin le plus court entre v_i et v_j où le chemin le plus court est le chemin qui a le plus petit nombre d'arêtes.

Dans le cas d'un graphe pondéré, la mesure de dissimilarité dis_s (vi,vj) peut être prise comme le minimum des sommes des poids des chemins entre v_i et v_j.

Toute technique d'apprentissage non supervisée peut être appliquée sur la matrice de dissimilarités ainsi obtenue.

Classification hybride

Dans cette méthode, une dissimilarité globale $dis_{TS}(v_i,v_j)$ entre deux sommets v_i et v_j est définie comme une combinaison linéaire de deux mesures de dissimilarité correspondant respectivement à chaque type d'information :

$$\boldsymbol{dis_{TS}(v_i,v_j)=\alpha dis_T(d_i,d_j)+(1-\alpha)dis_S(v_i,v_j)}$$

Où dis_T (d_i,d_j) est une dissimilarité définie sur les attributs, dis_S (v_i,v_j) est définie directement sur le graphe, et α est un paramètre compris entre 0 et 1.

La mesure dis_S (v_i,v_j) c'est La longueur du plus court chemin entre v_i et v_j, et dis_T (d_i,d_j) est la distance euclidienne ou la distance du cosinus calculées sur les attributs .

Ensuite, la partition peut être construite soit avec un algorithme de partitionnement de graphe appliqué sur le graphe étendu et valué par la dissimilarité globale, soit par une technique non supervisée d'apprentissage utilisant la dissimilarité globale.

L'article [33] a proposé un algorithme GDPSO pour la mise en grappe (cluster) des réseaux sociaux, en utilisant la technique PSO et la topologie de réseau pour diriger les mises à jour de l'état des particules. La stratégie est de guider les particules vers une région prometteuse. Certains petits opérateurs, tels que l'initialisation heuristique et la réorganisation de position, sont introduits pour accélérer la convergence comme : taille de l'essaim de particules, nombre d'itérations g_{max}, poids d'inertie x, facteurs d'apprentissage c1 et c2; les données d'entrée sont matrice d'adjacence du réseau et en sortie résulte la meilleure forme physique, de la structure communautaire du réseau; la fonction de fitness adoptée pour cette algorithme est la fonction Modularité proposée par Newman.

Des expériences sur des réseaux synthétiques et réels ont démontré que cet algorithme est efficace et prometteur.

5 Conclusion

Une grande importance a été donnée aux techniques d'optimisations par essaim de particules, les travaux réalisés dans ce sens sont très nombreux et diversifiées, notamment les problèmes de clustering, il est claire que cette technique pure ou hybride est promotrice, mais peu sont les travaux réalisés en utilisant cette technique au profil de partitionnement des graphes et la détection des communautés dans les réseaux sociaux.

Dans le chapitre suivant, nous allons implémenter un algorithme de détection des communautés dans un réseau social à base de PSO, puis nous introduisons le concept d'agents afin d'améliorer la qualité de clustering.

Chapitre 3

Contribution & Expérimentations

1 Introduction

Comme mentionné précédemment, l'objectif de notre étude et d'élaborer un algorithme basé sur la technique méta-heuristique PSO implémentée dans un système multi-agent SMA. Cet algorithme aura comme résultat un ensemble de communautés dans un réseau social.

Selon [34], un réseau social, est composé de deux variables : une variable structurelle, qui décrit les connexions entre les acteurs, une variable de composition, qui décrit chaque acteur de façon individuelle selon ses informations propres. Mais on peut définir une troisième variable, elle décrit les groupes d'acteurs, cette variable est dite d'affiliation et reflète ainsi l'appartenance à une communauté. Cette variable peut être inférée à partir des deux autres variables, et peut ainsi refléter des communautés détectées à travers la variable structurelle ou calculées grâce à des points communs du profil de la variable de composition.

A notre connaissance, peut d'approches ont essayé d'appliquer le clustering sur les réseaux sociaux comme des graphes, en utilisant la troisième variable. On trouve des approches de clustering des réseaux sociaux qui traitent les réseaux sociaux comme des graphes, où les nœuds représentent les utilisateurs (users) et les arêtes représentent les relations. Mais en réalité un réseau social contient plus d'informations qu'un graphe de relation : il contient également des informations sur chaque utilisateur, en plus de ses publications, par exemple, ses préférences de lecture, son âge, sa situation géographique et, en général, d'autres informations de contexte réseau.

Dans cette étude, nous définissons cette troisième variable comme base de clustering pour l'élaboration de l'algorithme de clustering dans les réseaux sociaux.

Nous considérons un réseau social comme un graphe noté G(V, E), V est l'ensemble des nœuds (individus) et E est l'ensemble des arêtes c.-à-d. les relations entre individus, ces arêtes sont pondérées par des similarités entre individus. Et nous considérons les communautés non chevauchées dans les graphes autrement dit, un sommet appartient seulement à une communauté.

Nous limitons cette similarité par une simple similarité entre les publications des individus, rien n'empêche d'élargir cette similarité pour contenir toute autre information sans aucune modification dans l'algorithme de clustering.

Méthodologie :

Pour aboutir à l'objectif, nous avons organisé notre démarche en phases à savoir :

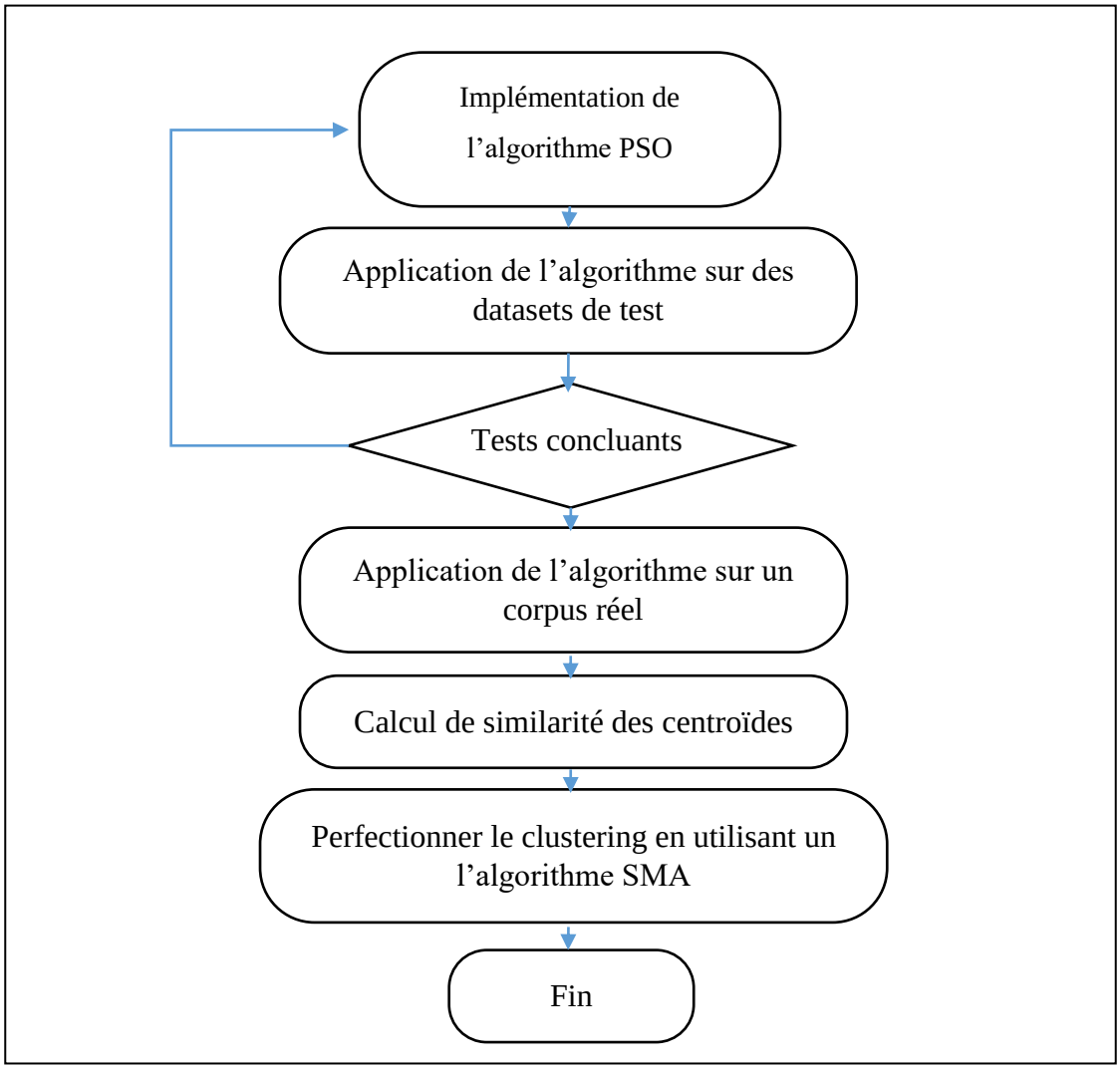


Figure 8: Méthodologie de travail

Ainsi le reste du chapitre sera organisé comme suit :

- Présentation de l'algorithme PSO proposé.
  - Description de l'algorithme.
  - Tests et comparaisons avec d'autres algorithmes
- Application de l'algorithme sur un corpus réel.
  - Préparation des données.
  - Résultats.
- Calcul de similarité des centroïdes des clusters avec le « profil modèle ».
- Description de l'architecture du système multi-agent proposé.
- Conclusion.

2 Plateforme de travail

Plateforme matérielle

Nos développements, tests et expérimentations ont été réalisés sur un PC marque DELL, CPU : Intel ® Core ™ I3-4170 3.70GHZ 3.70 GHZ, RAM : 8 GO, HDD : 1 TO. Carte graphique : Intel® HD Graphics 4400.

Plateforme logiciel

Cette machine tourne sous Windows 7 Professionnel 64 bits. Pour la partie prétraitement des données nous avons utilisé python 3.6, avec les packages :

- « Pickle » qui sert à lire/écrire les publications sous forme d'ensembles de mots.
- « shutil » pour la gestion des E/S fichiers et dossiers.
- « porterStemmer » c'est un stemmer (donne le stem d'un mot).
- « networkx » pour la gestion et la visualisation des graphes.
- « community » pour le clustering des graphes.

Pour éliminer les mots vides des publications, nous avons utilisé une liste Stopwords anglais téléchargée depuis : **https://code.google.com/archive/p/stop-words.**

Pour la partie implémentation de l'algorithme PSO, nous avons utilisé JAVA dans l'environnement ECLIPSE (Oxygen.1) September 2017 avec les packages :

- « jgrapht » org.jgrapht pour la gestion des graphes.
- « JADE » pour utilisation des SMA.

3 Algorithme PSO pour clustering des graphes

Dans la technique PSO, chaque individu est appelé « particule », a un vecteur de position et un vecteur de vitesse. Le vecteur de position simule une solution candidate au problème d'optimisation. Pour que la PSO recherche la solution optimale, une particule mettra à jour sa trajectoire au fil de temps avec quelques règles simples.

Si « n » est la taille de l'essaim de particules et « d » est la dimension de la particule, on a $V_i = (v_1, v_2, ..., v_d)$ et $X_i = (x_1, x_2, \ldots, x_d)$ sont les vecteurs de vitesse et de position de la $i^{ème}$ particule (i = 1, 2, ..., n).

Ensuite, la particule ajustera son statut selon les règles suivantes :

$$V_i\,(t+1) \leftarrow \omega V_i\,(t) + c1*r1(Pbest_i - X_i(t)) + c2*r2(Gbest - X_i(t)) \quad \textbf{(5)}$$

$$X_i(t+1) \leftarrow X_i(t) + V_i(t) \quad \textbf{(6)}$$

Où $V_i(t)$, $X_i(t)$ sont respectivement la vitesse et la position de la particule i à l'instant (t), $V_i(t+1)$, $X_i(t+1)$ sont respectivement la vitesse et la position de la particule i

A l'instant $(t+1).Pbest_i = (Xp_1, Xp_2, ..., Xp_d)$ est la meilleure position de la $i^{ème}$ particule et $Gbest = (Xg_1, Xg_2, ..., Xg_d)$ c'est la meilleure position du swarm. $c1$ et $c2$ sont les coefficients d'accélération respectivement cognitif et social. $r1$ et $r2$ sont des nombres aléatoires entre 0 et 1, ω est le coefficient d'inertie.

3.1 Description de l'algorithme

3.1.1 Initialisation du Swarm

Dans un algorithme PSO souvent, les particules sont initialisées aléatoirement. Notre proposition consiste à définir les noyaux initiaux des clusters en choisissant les nœuds qui ont un poids (avec pondération) > à un seuil donné.

Ensuite, pour tous les nœuds, trouver la liste des centroïdes (clusters) liés, les particules seront initialisées, pour chaque dimension (nœud), avec une valeur de cette liste. La vitesse pour chaque particule va être initialisée à 0.

```
1     Procedure Initialisation
2         Paramètres swarm_size=longueur(clusters)
3         i=0
4         Tantque i<swarm_size
5             Nouvelle particule P
6             Pour (j=0 ; j<dimension ; j++) :
7                 Si longeur(voisignage(j))>i Alors pos=voisinage(j)(i)
8                 Sinon pos(voisinage(j)(index aléatoire)
9                 P.position[j]=pos
10                P.vitesse[j]=0
11            P.fitness=modularite (P.position)
12            P.best=P.position
13            Si P.fitness est max Alors Gp=P
14            swarm←P
15            incrémenter i
16        Fin Tanque
17    Fin Proc.
```

Figure 9: Initialisation des particules

3.1.2 Exemple d'illustration

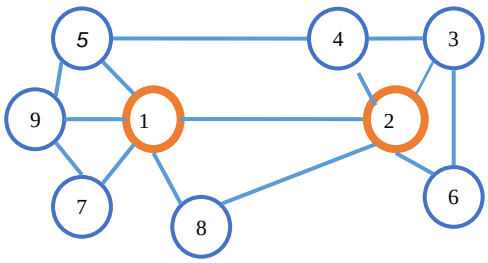


Figure 10: Graphe exemple

Pour un seuil =4 nous avons 2 centroïdes (possibles) : nœud 1 et 2.

Nœud	Centroïdes(Clusters)	
1	1	2
2	2	1
3	2	
4	2	
5	1	
6	2	
7	1	
8	1	2
9	1	

X_0	1	2	2	2	1	2	1	1	1
X_1	2	1	2	2	1	2	1	2	1
V_0	0	0	0	0	0	0	0	0	0
V_1	0	0	0	0	0	0	0	0	0

Fitness1 0,695512821

Fitness2 0,266941392

P_{best0}	1	2	2	2	1	2	1	1	1
P_{best1}	2	1	2	2	1	2	1	2	1

G_{best}	1	2	2	2	1	2	1	1	1

Figure 11: Initialisation des particules (exemple)

Dans l'exemple ci-dessus nous avons un essaim de 2 particules P_0 (x_{00}, x_{01}, x_{02},.., x_{0d}), P_1(x_{10}, x_{11}, x_{12},.., x_{1d}), d(dimension du graphe)=9. V_0=(0,0,…,0) (vitesse pour la particule P_0) V_1=(0,0,…,0) (vitesse pour la particule P_1), Gbest est la position de la particule où la fonction objective (fitness) est la plus optimale (ici c'est P_0), la modularité Q_0=0,266941392 pour P_0 et Q_1=0,695512821 pour P_1, Gbest sera mise à jour à chaque itération.

3.1.3 Fonction objective (fitness)

Soit $G = (V,E)$ un graphe et $\{C_1,\ldots,C_k\}$ une partition de V en k clusters. La question de savoir comment évaluer la qualité du clustering, c.-à-d. des structures des communautés trouvées par les algorithmes de clustering, se pose lorsque nous travaillons sur des ensembles de données du monde réel et que nous ne connaissons pas les communautés auparavant. Le concept de modularité Q proposé par Newman et Girvan [36] est souvent utilisé comme une mesure de qualification pour les communautés. Le calcul de la modularité est donné par la règle suivante :

$$Q = \frac{1}{2m}\sum_{vv' \in V}\left[A_{vv'} - \frac{d_v d_{v'}}{2m}\right]\sigma(C_v, C_{v'}) \qquad \textbf{(7)}$$

Où m =|E| est e nombre des arrêtes,

$A_{vv'}$: poids de l'arête (v,v'), d_v : somme des poids des arêtes de v.

Cv: cluster de v, $\sigma(C_v, C_{v''})$=1 si Cv=Cv', 0 sinon

a. Règles de mise à jour de l'état (position et vitesse) d'une particule.

Pour cet algorithme, les vecteurs de la vitesse et la position dans les règles (5) et (6), ont été redéfinis sous une forme discrète comme suit :

$$V_i(t+1) \leftarrow ג\left[\omega V_i(t) + c1r1\left(Pbest_i \ominus X_i(t)\right) + c2r2\left(Gbest \ominus X_i(t)\right)\right] \quad (8)$$

$$X_i(t+1) \leftarrow X_i(t) \circledast V_i(t+1) \quad (9)$$

Soit V une vitesse définie par V=(v_1,v2,…,vn), X une position définie par X=(x1,x2,…,xn), si V1,V2 et V3 sont des vitesses et X1,X2 sont des position, on définit les règles suivantes :

- V1⊖V2=V3 avec

$$v3_k = \begin{cases} 0 \text{ si } v2_k = v1_k \\ 1 \text{ sinon} \end{cases}$$

- X1⊛V1=X2 avec

$$x2_k = \begin{cases} x1_k \quad \text{si } v1_k = 0 \\ \text{Argmax}_p\,(Q/\, p \in \{\text{voisinage}(k)\} \text{ si } v1_k = 1 \end{cases}$$

- La fonction ג est définie :

$$ג(x) = \begin{cases} 0 \quad \text{si } x<1 \\ 1 \quad \text{sinon} \end{cases}$$

3.1.4 L'algorithme principale

La structure générale de notre algorithme est pratiquement la même pour tout algorithme PSO, la différence réside dans la partie initialisation et dans les procédures de mise à jour de la situation des particule.

Comme tout autre algorithme porte un nom pour le distinguer des autres, nous avons baptisé notre algorithme proposé, « VSPSO ».

```
1     Algorithme  VSPSO
2       Paramètres G(V,E)
3       swarm_size=N // N donné
4       initialisation(swarm_size)
5           Pour chaque particule P faire
6               calculer Q la valeur de la fonction objective // fitness
7               Si Q<Pbest Alors
8                   P.best=P.position
9                   P.fitness=Q
10              FinSi
11              Si Q<Gbest Alors Gp=p
12          Fin pour
13          Pour chaque particule P faire
14              mettre à jour vitesse(p)
15              mettre à jour position(p)
16          Fin pour
17    Fin VSPSO.
```

Figure 12: Algorithme VSPSO

3.2 Tests et comparaisons avec d'autres algorithmes

Dans cette section nous allons tester notre algorithme VSPSO sur des données réelles proposées par [33], en le comparant avec des algorithmes de détection de communauté à savoir : GDPSO[33], GA[38], MOGA[39], LPA[41], CNM[42], Informap[40]. Les résultats de test de ces algorithmes ont été extraits du [37], nous complétons le tableau de comparaison par les résultats de notre implantation PSO.

3.2.1 Présentation des datasets

Dataset	Nbr de nœuds	Nbr d'Arêtes	Lien de téléchargement
Karate	34	**78**	konect.uni-koblenz.de/networks/ucidata-zachary
Dolphin	62	**159**	www-personal.umich.edu/~mejn/netdata/dolphins.zip
Football	115	**613**	www-personal.umich.edu/~mejn/netdata/football.zip.
SFI	118	**200**	github.com/MessianNil/CommunityDetection/blob/master/Datasets/SFI/collaboration.zip
E-mail	1133	**5451**	deim.urv.cat/~alexandre.arenas/data/xarxes/email.zip
Netscience	1589	**2742**	www-personal.umich.edu/~mejn/netdata/netscience.zip
Power grid	4941	**6594**	konect.uni-koblenz.de/networks/opsahl-powergrid
PGP	10680	**24340**	konect.uni-koblenz.de/networks/arenas-pgp

Tableau 1 Liens de téléchargement des Datasets

3.2.2 Résultats des tests :

Selon [33], le paramètre k utilisé dans k-means et Ncut est égale à 4 pour **Karate**, à 5 pour **Dolphin**, à 12 pour **football**, à 7 pour **SFI**, à 4 pour **Email**, à 100 pour **Netscience**, à 200 pour **Power grid** et à 300 pour **PGP**. Pour les autres algorithmes le nombre de clusters n'est pas prédéfini.

Dataset	**Karate**	**Dolphin**	**Football**	**SFI**	**E-mail**	**Netscience**	**Power grid**	**PGP**
Algorithme	Qmax	Qmax	Qmax	Qmax	Qmax	Qmax	Qmax	Qmax
GDPSO	**0.4198**	**0.5285**	**0.6046**	**0.7506**	0.5487	0.9540	0.8382	0.8050
GA	**0.4198**	**0.5285**	0.5929	**0.7506**	0.3647	0.9086	0.7161	0.6576
MOGA	**0.4198**	0.5085	0.5280	0.7430	0.3283	0.8916	0.7035	—
LPA	0.4151	0.5258	0.6030	0.7341	0.2055	0.9266	0.7602	0.7949
CNM	0.3800	0.4950	0.5770	0.7335	0.4985	0.9555	0.9229	0.8481
Informap	0.4020	0.5247	0.6005	0.7334	0.5355	0.9252	0.8140	0.7777
k-Means	0.1429	0.4796	0.5783	0.4376	0.3681	0.6510	O.o.M*	O.o.M*
Ncut	**0.4198**	0.5068	0.6031	0.7478	0.4841	0.9293	0.8875	O.o.M*
VSPSO.	**0.4198**	**0.5285**	**0.6046**	**0.7506**	**0.5582**	**0.9185**	**0.8391**	**0.8140**
Classement	01**	01**	01**	01**	01	06	03	02

*Out of Memory **Avec d'autres

Tableau 2 : Résultats de la fonction de modularité selon les algorithmes et les datasets

On constate que notre algorithme converge rapidement vers une valeur de modularité (figure 14). C.-à-d. que l'influence de nombre d'itération sur la qualité du clustering est limitée. Par contre, l'influence de la taille de l'essaim est remarquable ainsi que l'initialisation de l'essaim.

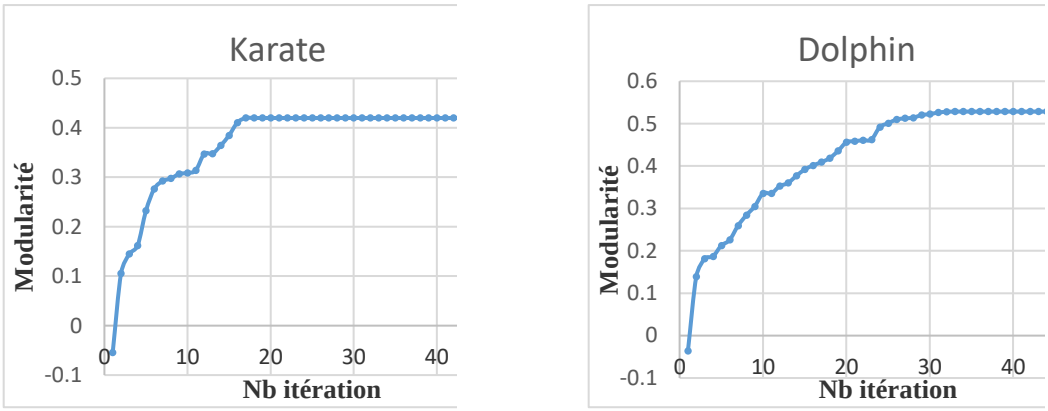


Figure 13 Exemple de convergence de la modularité

Les résultats du test indiquent que l'algorithme VSPSO est bien classé parmi les algorithmes cités ci_dessus. Il surpasse LPA, CNM, Informap, k-Means pour **Karate**, et il est aligné avec GDPSO, GA, MOGA et Ncut pour le même réseau.

Pour **Dolphin** et **SFI**, il est classé premier avec GDPSO et GA, il est de même rang avec GDPSO pour **Football.**

3.3 Récapitulation

Malgré sa version primaire, on remarque que notre implémentation donne de bons résultats en matière de clustering des graphes en se comparant à d'autres algorithmes.

On constate aussi que cet algorithme est très sensible à l'initialisation des particules, car les résultats du clustering se divergent rapidement entre deux initialisations distinctes. Même constatation pour le choix des paramètres de l'algorithme. L'influence de la taille de l'essaim est remarquable sur la précision du clustering, mais la variation de la valeur de la modularité n'est pas assez corrélée à cette taille. Une fois les paramètres de l'algorithme sont bien choisis, les particules ont une bonne initialisation, on remarque que cet algorithme a une convergence rapide vers la solution optimale.

Les résultats des tests obtenus nous permettent d'entamer la phase suivante qui est l'application de l'algorithme sur un corpus réel.

4 Application de l'algorithme sur un corpus réel

Après avoir testé notre algorithme sur des données des graphes réels de tailles variées. Les résultats de ces tests ont prouvé que la mise en œuvre de l'algorithme proposé pour le clustering (partitionnement) des graphes est réalisable et qu'il est capable de détecter des structures de communautés dans un corpus extrait d'un réseau social.

Pour se faire, nous avons téléchargé un corpus (Dataset) depuis ***http://snap.stanford.edu/data/*** pour 4.069.982 Users de Twitter avec leurs publications. Le corpus est composé de deux parties, un fichier qui contient le graphe de lien (qu'on considère dans notre cas comme un graphe non orienté), est un ensemble de Tweets sous forme de fichiers.

Etant donné que le corpus téléchargé est volumineux, Nous avons simplifié le corpus en éliminant les nœuds de degré 1 et les nœuds isolés. Ainsi la taille du corpus a été réduite.

4.1 Préparation des données

Avant de commencer l'application de l'algorithme sur un dataset réel, afin de détecter une éventuelle structure de communauté, il est nécessaire de procéder à un prétraitement des tweets en commençant d'abord par supprimer le bruit en procédant comme suit :

1. Convertir les tweets en minuscules.
2. Éliminer toutes les URL.
3. Supprimer "@username"
4. Remplace les hashtags avec exactement le même mot sans le hash.
5. Supprimez la ponctuation au début et à la fin des tweets. Remplacer plusieurs espaces avec un seul espace.
6. Supprimer les mots vides (stop words)
7. Appliquer un stemmer.
8. Fusionner tous les tweets d'un même user dans un seul fichier pour former un sac de mots pour chaque user.

Une fois les prétraitements sont achevés, on calcule la similarité entre des sacs de mots (Tweets) deux par deux selon le fichier des liens, on obtient ainsi un graphe de lien pondéré par ces mesures de similarité qui peut être partitionné avec notre algorithme.

4.2 Résultats

L'algorithme est appliqué à ce Dataset, la modularité de clustering est de l'ordre de : 0.8213, les données et les résultats de l'opération sont :

Nb nœuds	1982
Nb arêtes	2806
Nb clusters	26
Taille du plus grand cluster	239
Taille du plus petit cluster	11

Tableau 3 : Les résultats de l'application de l'algorithme sur un corpus réel

Le graphe de communautés résultantes du clustering est le suivant :

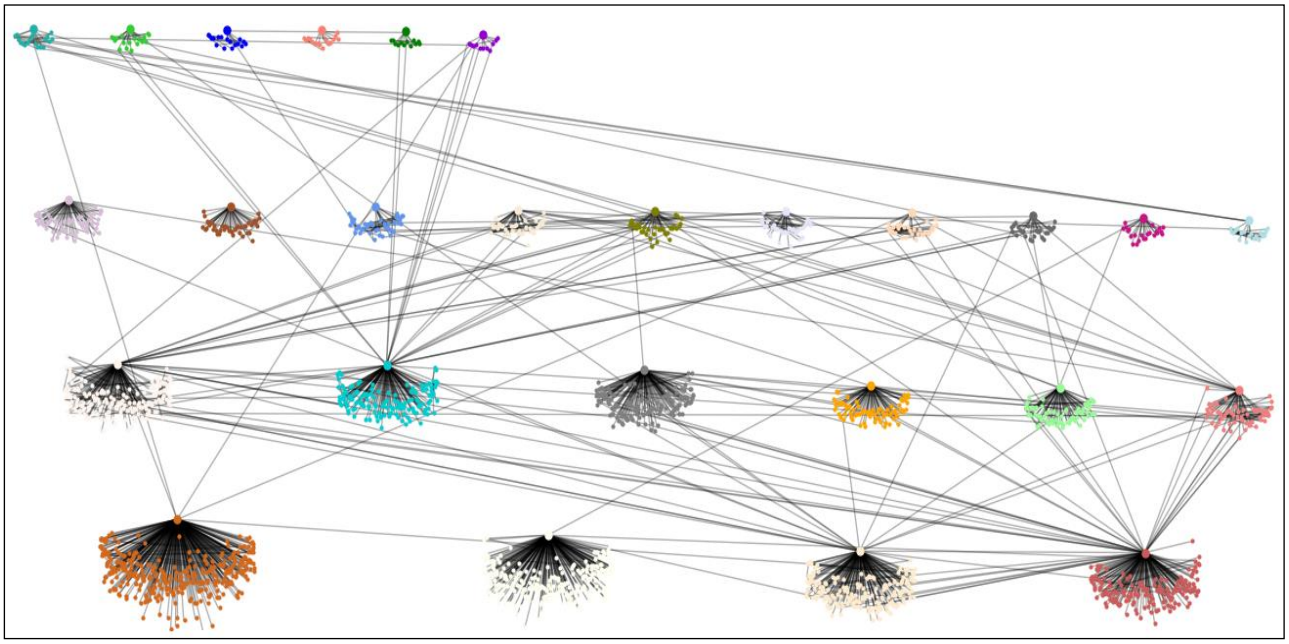


Figure 14 : Le graphe de communautés résultantes du clustering

4.3 Récapitulation

Vu le nombre des arêtes intra clusters qui est assez grand que ce des arêtes inter clusters, et la modularité de clustering (qui est proche de 1), nous pouvons dire que notre algorithme de clustering donne de bons résultats sur des données extraites d'un réseau social réel.

Pour confirmer ce résultat, nous avons utilisé la fonction « *best_partition()* » du package « *community* » sur python, pour partitionner le graphe en mettant tous les poids à 1. Nous avons, à l'aide de la fonction « *modularity*() » du même package, calculé la modularité de cette partition qui égale à 0.8664.

5 Calcul de similarité des centroïdes des clusters avec le « Profil modèle »

Afin de détecter quelles sont les communautés suspectes, il faut comparer (calculer la similarité) entre ses centroïdes et un « Profil modèle », ce dernier peut être conçu manuellement ou téléchargé depuis le web ou de toute autre source.

Dans cette phase il est possible de choisir n'importe quel « Profil modèle » qui décrit une forme de suspicion, pour notre cas nous avons téléchargé un document portant les mots les plus vulgaires en anglais depuis : **https://code.google.com/archive/p/badwordslist/downloads.**

Les communautés séparées sont automatiquement de même nature de suspicion.

La mesure de similarité utilisée ici est la mesure de Jaccard.

N° Cluster	Taille	Similarité
1	329	0,0024
2	231	0,0032
3	183	0,0090
4	176	0,0070
5	149	0,0074
6	146	0,0081
7	131	0,0079
8	76	**0,3070**
9	68	0,0072
10	63	0,0076
11	58	0,0090
12	45	**0,2893**
13	40	**0,2719**

N° Cluster	Taille	Similarité
14	36	0,0047
15	33	0,0076
16	32	0,0038
17	32	0,0097
18	27	0,0029
19	24	0,0024
20	22	**0,2893**
21	18	0,0110
22	17	0,0105
23	13	0.0029
24	12	0.0065
25	11	0.0013
26	10	0.0099

Tableau 4 : Résultats de similarité

Après les calculs, nous avons constaté que ces 26 clusters ont des similarités faibles avec notre « Profil modèle ».

Les quatre meilleurs clusters sont les clusters N° 8, 12, 20 et 13 qui ont des similarités respectivement 0.3070, 0.2893, 0.2893 et 0.2719, la chose qui signifie que ce sont les clusters les plus suspects relativement au « Profil modèle » choisi.

6 Description de l'architecture du système multi-agent proposé

Les résultats constatés lors de l'application de dit algorithme PSO sur des datasets de dimensions relativement importantes, montrent que cet algorithme souffre toujours de la malédiction de la dimensionnalité, malgré les améliorations proposées, aux déférentes étapes de l'algorithme, que nous nous attendions à ce qu'elles soient opérantes.

Pour améliorer la qualité de clustering de cet algorithme, nous proposons de le compléter par une hybridation SMA. Cette hybridation nous permet un gain considérable de temps vu le parallélisme des SMAs, en plus de la perfection du clustering attendu par ce SMA.

L'approche permet à une collection d'agents de collaborer pour produire un « meilleur » clustering, ce qui implique un certain nombre d'axe à développer :

- Le fonctionnement de clustering SMA souhaitée, (l'environnement SMA, la coordination et la communication inter-agent).
- Le second est de savoir comment définir la manière d'optimiser le clustering.

Ce SMA prend comme entrée les clusters produit lors de la première phase, ensuite par définition d'une manière d'optimisation (que nous allons détailler plus tard), les agents vont réagir pour obtenir leurs objectifs.

6.1 Processus de perfectionnement du clustering

La quasi-totalité des approches, qui utilisent les systèmes multi-agents (SMA) en hybridation dans le clustering, ont voulu profiter de la distribution et le parallélisme que les SMAs offrent, autrement dit les SMA n'affectent pas directement la qualité de clustering mais peut être en distribuant les particules par exemple dans le cas d'un PSO.

Notre approche vise à impliquer directement les SMAs dans le clustering. Etant donné que nous avons les clusters résultats du PSO de la première phase, nous allons considérer que chaque cluster sera représenté par un agent. Chaque agent va mettre les nœuds, qui lui semblent étranges de son cluster, à la disposition des agents qui vont les intégrer à leurs clusters, si ces nœuds semblent homogènes avec leurs clusters. Nous pouvons dire que c'est une sorte de mini bourse.

L'agent donneur décide si un nœud est étrange ou pas, en calculant initialement la modularité du cluster, ensuite il la recalcule en l'absence de ce nœud, si la variation de la modularité du cluster est négligeable alors le nœud est étrange, est peut-être mis à la disposition des autres agents.

De la même manière, l'agent récepteur décide si un nœud est peut-être de son cluster ou non, en examinant le gain qui porte l'intégration de ce nœud à son cluster.

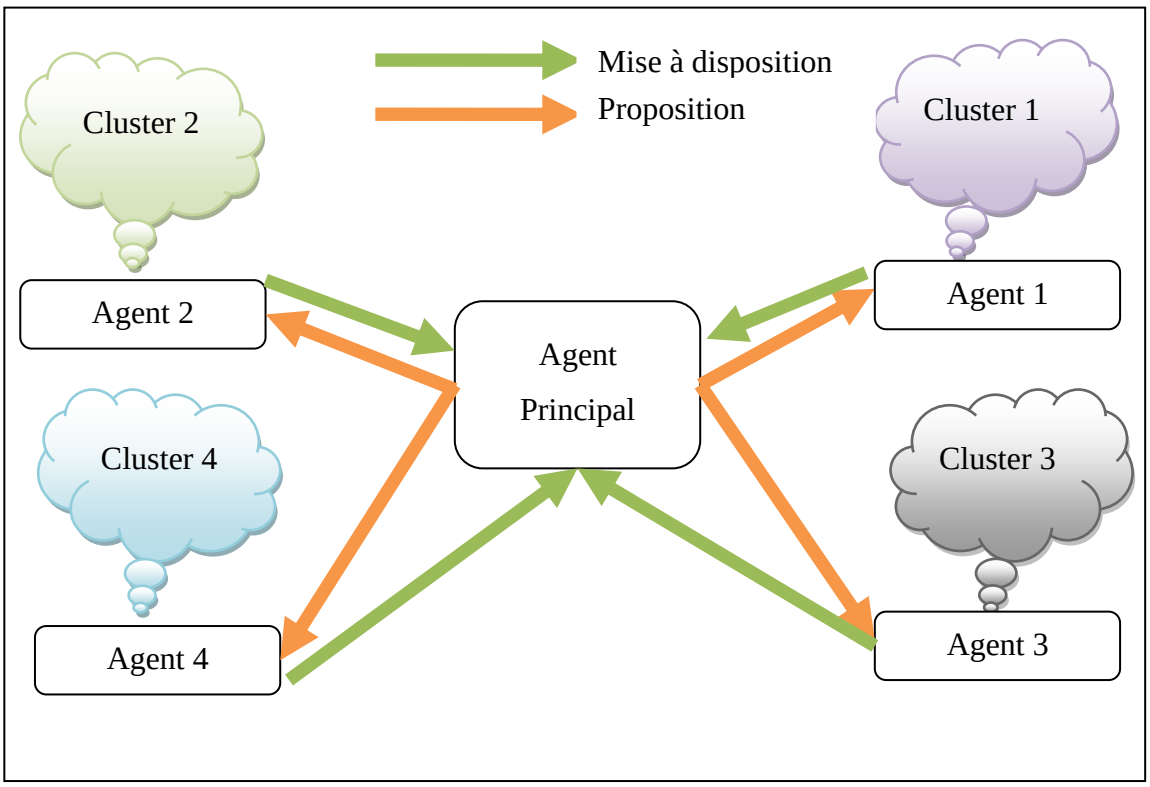


Figure 15: Architecture SMA proposée

6.2 Description du Système Multi Agent proposé

Le but de l'architecture proposée est de raffiner le clustering lors de l'utilisation du PSO, elle est constituée de :

- L'agent principal (modérateur),
- Les agents cluster,

6.2.1 L'agent principal (modérateur)

Examine en permanence la couche d'agents cluster. Il tient à jour une liste d'informations sur tout l'environnement. A l'initialisation, pour chaque cluster il doit créer un agent cluster et lui transfère les données qui lui appartiennent.

Il organise l'opération de transition des nœuds entre les clusters, après qu'un agent met un nœud à la disposition des autres agents, il le communique à l'agent modérateur qui à son tour le (nœud) propose uniquement aux agents clusters qui ont des nœuds voisins.

L'agent modérateur maintient une liste des nœuds mis à disposition.

6.2.2 Agent cluster

Chaque cluster est représenté par un agent cluster. Il maintient une structure de cluster, et pour chaque nœud de ce cluster il maintient une structure de voisinage. L'agent cluster ne peut communiquer avec l'agent modérateur pour lui envoyer un nœud ou pour recevoir un autre. Il doit communiquer avec l'agent modérateur pour lui communiquer les résultats de calculs à chaque fois qu'il est nécessaire.

L'agent cluster doit classer dans l'ordre croissant une liste dans ces nœuds en fonction de la variation de la modularité locale ΔQi, en d'autre termes, avoir les nœuds qui leur suppression du cluster n'influe pas tellement sur la modularité local Qi. La modularité locale Qi est la modularité du cluster i, qui a la même formule que la modularité globale Q en prenant uniquement les nœuds de ce cluster, avec

$$Q = \sum_{i \in C} Qi, \quad o\text{ù } C \; est \; l'ensembledes \; clusters$$

6.3 Récapitulation

La mise en place d'un système multi agents nécessite un temps supplémentaire, nous n'avons pas pu achever l'implantation dans les délais impartis, néanmoins d'après les résultats d'une simple simulation que nous avons faite sur une station en java, nous pouvons conclure qu'une amélioration de la qualité de clustering peut être obtenue en appliquant cette approche.

La programmation de cette simulation nous a permet aussi de ressentir l'utilité de la norme FIPA, utilisée dans le système de communication entre agents.

7 Conclusion

Cette expérimentation nous a présenté l'opportunité d'entamer des divers domaines de recherche d'une part, et d'appliquer autant de connaissances acquises pendant notre cursus.

Nous avons décrit un nouvel algorithme permettant de détecter les structures des communautés à partir de la topologie du réseau et la relation entre ces nœuds, en utilisant la technique PSO, qui fonctionne en optimisant la modularité. Nous n'avons pas testé le temps d'exécution de notre algorithme, et nous n'avons aucune image de comparaison par rapport aux algorithmes généraux précédemment cités.

L'élaboration de cet algorithme exige la maitrise de la technique PSO en plus de quelques notions fondamentales sur les graphes. L'assurance de son fonctionnement nous a obligé de chercher des datasets qui décrivent des graphes dont leurs partitionnements sont déjà connus, et des algorithmes connus pour avoir une sorte de comparaison.

Nous avons eu des difficultés pour appliquer notre algorithme à un corpus d'un réseau social réel qui répond à nos exigences. Le corpus a été trouvé mais nécessite des traitements avant qu'il soit utilisable. Ces traitements nous ont permet à leurs tours de découvrir la puissance de « Python » et d'apprendre à utiliser ses packages spécialisés.

Le Profil modèle ou modèle dans son état pur, est introuvable sur le web, mais il peut être reconstruit à partir des projets de traitement de langage naturel ou à partir d'un thésaurus.

Le SMA « JADE » est très simple à installer sur « Eclipse », mais la mise en œuvre d'une architecture SMA peut paraitre long et épineux.

Conclusion générale

La détection communautaire est d'une grande importance dans les réseaux sociaux où ils sont souvent représentés sous forme de graphes. Beaucoup de recherches dans ce domaine ont été présentées, sans que ce problème est résolu de manière satisfaisante malgré les nombreuses méthodes qui ont proposé.

Dans ce mémoire, une méthode basée sur l'optimisation de l'essaimage de particules (PSO) est proposée pour la détection des communautés complétée par une approche basée sur les systèmes multi agents pour améliorer la qualité de clustering.

Le critère de mesure populaire modularité Q est utilisé comme fonction de fitness. PSO est adopté pour maximiser la modularité du réseau.

Durant l'application de la technique de clustering PSO, nous avons constaté trois inconvénients majeurs entravent son utilisation dans le domaine.

1. Tout d'abord, PSO a été conçu à l'origine pour des problèmes d'optimisation continue, ce qui limite son application dans le domaine d'optimisation discret.
2. PSO souffre de la malédiction de la dimensionnalité, c'est-à-dire que ses performances se détériorent rapidement lorsque la dimensionnalité de l'espace de recherche augmente exponentiellement, ce qui rend son application aux réseaux sociaux, parce que l'échelle d'un réseau social réel est particulièrement grande.
3. PSO est très sensible à l'initialisation des particules, donc une initialisation non adéquate peut mener le swarm à converger vers un optimum local, donc rater l'optimum global.

Nous avons essayé de pallier ces faiblesses, nous avons expérimenté une méthode de PSO discrète conçue pour découvrir les structures communautaires dans les réseaux sociaux, où un algorithme PSO modifié a été proposé, nous avons d'abord redéfini la position et la vitesse des particules sous une forme discrète, puis nous avons modifié les règles de mise à jour des particules.

Des expériences sur des réseaux réels ont démontré que l'algorithme proposé pour le clustering de réseaux est efficace et prometteur. Bien que l'algorithme PSO discret proposé présente une bonne performance, dans nos expériences, nous trouvons que lorsque l'on s'adresse à des réseaux à grande échelle, la performance de l'algorithme se réduit. Donc nous n'avons pas réussi à bien résoudre le problème de la malédiction de la dimensionnalité en modifiant simplement l'algorithme PSO.

De plus, la fonction objective (fitness) est liée à tous les nœuds du graphe, donc nous trouvons qu'il est coûteux de l'évaluer quand l'échelle du réseau est particulièrement grande.

La solution d'hybridation par un système multi-agent n'a pu être testé par défaut de temps, malgré que nous avons imaginé la conception du système et que nous avons bien décrit les composantes de ce système. Sa simulation nous a révélé des perspectives prometteuses

Pour une future reprise du sujet nous proposons de trouver une fonction objective qui est moins liée aux nœuds ou d'optimiser ses calculs. Nous suggérons aussi de trouver une autre astuce afin d'initialiser les particules, car les résultats sont fortement liés à cette initialisation.

De même, nous proposons d'achever l'implantation du système SMA précédemment décrit.

Références et bibliographie

[1] Charles Hélou «*Plateforme pour se protéger tant de soi-même que de ses amis sur Facebook*»2012.

[2] Russell Eberhart, James Kennedy: «*A New Optimizer Using Particle Swarm Theory*. »1995.

[3] Jaco F. Schutte «,*The Particle Swarm Optimization Algorithm*»2005

[4]. Maxime BOMBRUN «*L'optimisation par essaim particulaire pour des problèmes d'ordonnancement*» Abdoulaye SENE 2011.

[5]. Benoit Dupont, Pierre-Eric Lavoie & Francis Fortin « *Les crimes sur le web 2.0 Une recherche exploratoire* » 2010 Université de Montréal

[6] S. Russel, P. Norvig, «*Artificial intelligence – A modern approach*», Prentice Hall, 1995.

[7]. P. Maess, «*Artificial life meets entertainment: Life like autonomous agents*», 1995.

[8] Raheel Ahmad ,Yung-Chuan Lee, Shahram Rahimi «*A Multi-Agent Based Approach for Particle Swarm Optimization*»

[9) Ricco RacotoMalala «*Détection des communautés dans les réseaux sociaux*»

[10] JayshreeGhorpade-Aher ,Vishakha Arun Metre. « *PSO based Multidimensional Data Clustering: A Survey* ». 2014

[11] Duquesnoy Maxime « *Pour un usage éducatif des réseaux Sociaux* » 19 mars 2013

[12] Elie Raad, Richard Chbeir «*Privacy in Online Social Networks*»

[13] Abbas El Dor «*Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en électronique*»15 Feb 2013

[14] Lior Rokach et Oded Maimon «*CLUSTERING METHODS*»

[15] Strehl, A., Ghosh, J., Mooney, R.: «*Impact of similarity measures on web-page*

clustering. In Proc. AAAIWorkshop on AI forWeb Search» 2000.

[16] Alboukadel Kassambara «*Ractical Guide To Cluster Analysis in R Unsupervised Machine Learning*»2017

[17] L.V. Bijuraj «*Clustering and its Applications*»2013

[18] Jain, A. K ; Murty, M. N. Flynn, P. J «*Dataclustering: a review ACM Computing Survey* » 1999.

[19] Steinbach, M; Karypis, G; Kumar, V. «*A Comparison of Document Clustering Techniques*» 2000.

[20] Zhao. Y; Karypis G «*Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning*» 2004

[21] Sunita Sarkar et Arindam Roy, Bipul Shyam Purkayasth «*Application of Particle Swarm Optimization in Data Clustering: A Survey*»(2013)

[22] Vincent Levorato ,Thanh Van Le , Michel, Lamure, and Marc Bui *« Distance de compression et classification prétopologique*»

[23] KAUFMAN L., ROUSSEEUW P. J., «*Finding groups in data: An introduction to cluster Analysis* », 1990.

[24] Peter Janssen ,M. K. B. Lüdeke, Carsten Walther«*Cluster Analysis to Understand » Socio-Ecological Systems: A Guideline* »September 2012

[25] David Combe — Christine Largeron — Elöd Egyed-Zsigmond ,Mathias Géry « *Détection de communautés dans des réseaux scientifiques à partir de données relationnelles et textuelles* »

[26] E. Zahara, S. Fan, D. Tsai. *« Optimal multi-thresholding using a hybrid optimization approach. Pattern Recognition Letters*» . Vol. 26, N° 8, pp. 1082- 1095, 2005.
[27] Charl es-Edmond Bichot « *Introduction au partitionnement de graphe*» Novembre 2007

[28] Palla, G., Der_enyi, I., Farkas, I., & Vicsek, T.. « *Uncovering the overlapping community structure of complex networks in nature and society*. » 2005

[29] X. Pu, Z. Yi, Z. Fang. *« Holistic and partial facial features fusion by binary particle swarm optimization. Neural Computing and Applications*. » Vol. 17, N° 5-6, pp. 481-488, 2008.
[30] Jean-Daniel KANT «*Modélisationet Simulation Multi-Agents Cours1 – Prolégomènes*»

[31] DW van der Merwe, AP Engelhrecht « *Data Clustering using Particle Swarm Optimization*» 2003.

[32] Diego INGARAMOa, Marcelo ERRECALDE a and Leticia CAGNINAa, Paolo ROSSO « *Particle Swarm Optimization for clustering short-text corpora* »2006

[33] Qing Cai, Maoguo Gong , Lijia Ma, Shasha Ruan, Fuyan Yuan, Licheng Jiao « *Greedy discrete particle swarm optimization for large-scale social network clustering* »2014

[34] Wasserman S., Faust K. (1994). "*Social network analysis: Methods and applications"* no 8. Cambridge University Press.

[35] David COMBE. (2014). "*Détection de communautés dans les réseaux d'information utilisant liens et attributs"* : thèse de doctorat, université Jean Monnet ; Paris

[36] M. E. J. Newman and M. Girvan*, "Finding and evaluating community structure in networks"*. Phys. Rev. E 69, 026113 (2004).

[37] Amira Gherboudj *« Méthodes de résolution de problèmes difficiles académiques"2013*

[38] C. Pizzuti, GA-Net: "*a genetic algorithm for community detection in social networks",* in: Parallel Problem Solving from Nature (PPSN), vol. 5199, 2008,pp. 1081–1090.

[39] C. Pizzuti, *"A multiobjective genetic algorithm to find communities in complex networks"*, IEEE Trans. Evolut. Comput. 16 (3) (2012) 418–430.

[40] M. Rosvall, C.T. Bergstrom, "*Maps of random walks on complex networks reveal community structure*", Proc. Nat. Acad. Sci. USA 105 (4) (2008) 1118–1123.

[41] J.P. Bagrow, E.M. Bollt, "*Local method for detecting communities*", Phys. Rev. E 72 (October) (2005) 046108.

[42] A. Clauset, M.E.J. Newman, C. Moore, *"Finding community structure in very large networks"*, Phys. Rev. E 70 (6) (2004) 066111.

[43] S. T. Hsieh, T. Y. Sun, C. C. Liu, and S. J. Tsai. *"Efficient population utilization strategy for particle swarm optimizer*". pp. 444–456, IEEE Press, Piscataway, NJ, USA, 2009.

[44] Stuti Karol_, Veenu Mangat « *Evaluation of text document clustering approach based on particle swarm optimization* » 2012

[45] Van der Merwe ,DW; AP Engelhrecht, « *Data clustering using particle swarm optimization. In: Conference of evolutionary computation* »2003 CEC' 03, vol 1. pp 215–220.

[46] Premalatha, K and Natarajan, AM. «*Hybrid PSO and GA Models for Document Clustering*. » Int. J. Advance. Soft Comput. Appl., Vol. 2, No. 3,(2010) .

[47] Khaled Ziane « Analyse, Évaluation et Réduction des Risques d'un Parc Éolien » 2017

[48] Henri Sanson « *L'Intelligence Artificielle expliquée aux humains* » publication web sur Orange.com Publié le jeudi 05 janvier 2017

[49]Ahmady fard, A; Modares, H « *Combining PSO and k-means to enhance data clustring* ». In: International symposium on telecommunications. pp 688–691 (2008)

[50] Niknam,T; Nayeripour, M; Firouzi, BB(2008). « *Application of a New Hybrid optimization Algorithm on Cluster Analysis Data clustring* ». World Academy of Science, Engineering and Technology 46

[51] Marinakis, Y; Marinaki, M; and Matsatsinis, N (2007). « *A Hybrid Particle Swarm Optimization Algorithm for Clustering Analysis* » .DaWaK 2007, Lecture notes in computer science, LNCS 4654, pp. 241–250

[52] Hwang, J.-I.G.; Huang, C-J.(2010) « *Evolutionary dynamic particle swarm optimization for data clustering. In: International Conference on Machine Learning and Cybernetics*» (ICMLC)

[53] H. Liu, A .Abraham. An Hybrid Fuzzy « *Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems* ». Journal of Universal Computer Science. Vol. 13, N° 9, pp. 1309-1331. 2007.

[54] Q. Shen, W-M Shi, W. Kong. « *Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data*. » Computational Biology and Chemistry. Vol. 32, N° 1, pp. 53-60, 2007.

[55] R-M. Chen, D-F. Shiau ,S-T. Lo. « *Combined Discrete Particle Swarm Optimization and Simulated Annealing for Grid Computing Scheduling Problem. Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence*», pp. 242-251. Springer, 2009.