

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

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE



MÉMOIRE DE MASTER

Réseaux Informatiques et Systèmes Répartis

THÈME :

IA-PSO-SSW : Implémentation d'un Algorithme PSO pour la Sélection de Services Web

Présenté par :

ATTOU Nadjjet

DAINES Arbia

Encadré par :

Dr. MEKOUR Mansour

Année Universitaire 2020-2021

**IA-PSO-SSW : Implémentation d'un
Algorithme PSO pour la Sélection
de Services Web**

Remerciment

*Avant tout ; nous remercions Allah de nous avoir aidé à faire ce travail
Nous remercions également nos enseignants pour la qualité
de l'enseignement qu'ils nous prodigué au cours de ces 5 années
passées à l'université Moulay Taher SAIDA.*

*Nous remercions tout particulièrement Mr. Mekour lencadreur qui nous
laissé une large part d'autonomie dans ce travail, et qui nous guidés vers des pistes
de réflexions riches et porteuses.*

*Nous souhaiterions aussi remercier tout le corps administratif, ainsi que
toutes les personnes qui souhaiteraient voir un jour notre université au meilleur
rang.*

*Nos profonds remerciements pour les membres de jury qui ont acceptés
d'évaluer ce travail.*

*Nous remercions l'ensemble des nous proches qui nous aidé et motivé
durant ce cursus rempli d'embuches, nous les remercions pour laide quils nous
apporté dans la réalisation de ce travail.*

*Nous remercions enfin les chers parents qui nous ont encouragé et
supporté durant tout ma vie scolaire.*

Merci à tous

...

Dédicace

*Je dédie ce modeste travail a mes très
chers parents qui ont été toujours la pour moi
et qu' «ALLAH » les protéger, pour leur
soutien, leurs encouragement et leur sacrifices
durant toute ma vie.*

*A mes très chers frères et sœurs qui m'ont
soutenu durant la réalisation du mémoire.*

A mes amis avec qui j'ai passé de bons moments.

A tous le département de la technologie.

*A tous ceux qui ont contribué de près
où de loin à la réalisation de ce mémoire.*

*Daines Arbia
Et merci*

....

Dédicace

*Je tien a dédier ce modeste travail à :
tous ceux qui m'ont encouragé durant toute la période de réalisation de ce travail.*

En particulier :

Mon très cher père, ma très chère maman qui ma toujours soutenu

A mes frères Amine, Nasrodin, abderrahim

A mes sœurs et leurs enfants

Hiba, Mounir, khadija , Chahed, Abderezak, Abdelaziz

Ma grand-mère

A Ma chère petite tante

A toutes ma famille (ATTOU)

A tous mes amis

A tous ceux que j'aime et ceux qui m'aiment

*Attou Nadjat
Et merci*

....

Résumé

Les services Web sont des entités logicielles auto descriptives et modulaires exposées sur internet, elles répondent à des besoins fonctionnels et des besoins non fonctionnels, ces derniers s'expriment généralement à travers les valeurs QdS (Qualité de Service), qui permettent aux utilisateurs, de choisir parmi les services Web fonctionnellement équivalents.

Pour une réutilisation de ces applications, et afin de satisfaire aux exigences complexes de l'utilisateur, les concepteurs sont tenus à englober les services Web déjà existants en un seul service dit composé, qui au moment de son exécution peut inclure certains services composants faillibles (devenir indisponible, non fiable ou ne plus offrir la meilleure solution), sur ses entrefaites, et au lieu de changer la composition entière, le remplacement avec un service idoine est une solution cruciale qui garantira la continuité du système.

Notre travail consiste à développer une application basée sur une approche d'optimisation évolutionnaires (PSO, NSGA2). L'objectif est de sélectionner un ensemble de services atomiques pour composer un nouveau service pour satisfaire les contraintes standards (minimiser le temps et le coût, et maximiser la disponibilité, la fiabilité et réputation) en fonction de QoWS.

Mots clés : Services Web, NSGA2, PSO, Composition de services, sélection de services, Qualité de service.

Abstract

Web services are self-descriptive and modular software entities exposed on the internet, they meet functional needs and non-functional needs, the latter generally expressed through QoS (Quality of Service) values, which allow users to choose among web services functionally equivalent.

For reuse of these applications, and in order to meet complex user requirements, designers are required to encompass already existing web services into a single so-called composite service, which at the time of its execution may include some fallible component services (become unavailable, unreliable or no longer offer the best solution), in the meantime, and instead of changing the entire composition, replacement with an appropriate service is a crucial solution that will guarantee the continuity of the system.

Our job is to develop an application based on an evolutionary optimization approach (PSO, NSGA2). The objective is to select a set of atomic services to compose a new service to meet standard constraints (minimize time and cost, and maximize availability, reliability and reputation) according to QoWS.

Keywords : Web services, NSGA2, PSO, Composition of services, selection of services, Quality of service.

ملخص

خدمات الويب عبارة عن كيانات برمجية ذاتية الوصف ووحدات معيارية معروضة على الإنترنت ، وهي تلبية الاحتياجات الوظيفية والاحتياجات غير الوظيفية ، ويتم التعبير عن الأخيرة بشكل عام من خلال قيم جودة الخدمة ، التي تتيح للمستخدمين الاختيار من بين خدمات الويب المكافئة وظيفياً.

لإعادة استخدام هذه التطبيقات ، ومن أجل تلبية متطلبات المستخدم المعقدة ، يتعين على المصممين تضمين خدمات الويب الموجودة بالفعل في ما يسمى بالخدمة المركبة ، والتي قد تتضمن في وقت تنفيذها بعض خدمات المكونات غير المعصومة (تصبح غير متاحة ، غير موثوق بها أو لم تعد تقدم الحل الأفضل) ، في غضون ذلك ، وبدلاً من تغيير التكوين بالكامل ، يعد الاستبدال بخدمة مناسبة حلاً مهماً يضمن استمرارية النظام.

مهمتنا هي تطوير تطبيق يعتمد على نهج التحسين التطوري (PSO,NSGA2) . الهدف هو تحديد مجموعة من الخدمات الذرية لإنشاء خدمة جديدة لتلبية القيود القياسية (تقليل الوقت والتكلفة ، وزيادة التوافر والموثوقية والسمعة) وفقاً لـQoWS.

الكلمات الرئيسية: خدمات الويب ، NSGA2 ، PSO ، تكوين الخدمات ، اختيار الخدمات ، جودة الخدمة.

Table des matières	12
Table des figures	14
Liste des tableaux	15
Listes des abréviations	17
Introduction générale	18
1 Les Services Web	20
1.1 Introduction	21
1.2 Service Web	21
1.2.1 Définition	21
1.2.2 Les caractéristiques des services Web	22
1.3 Architecture Orientée Service (SOA)	22
1.3.1 Définition SOA	23
1.3.2 Les avantages de SOA	23
1.4 Différents types d'architecture des services web	24
1.4.1 Architecture de référence	24
1.4.2 Architecture étendue	25
1.5 Les principales technologies de développement de service Web	26
1.5.1 XML	26
1.5.2 Le Protocol SOAP	27

1.5.2.1	La structure de message SOAP	27
1.5.3	WSDL	28
1.5.3.1	Structure d'un document WSDL	28
1.5.4	UDDI	29
1.6	Les avantages et les inconvénients des services web	30
1.6.1	Les Avantages	30
1.6.2	Les Inconvénients	31
1.7	Conclusion	31
2	Composition des Services Web	32
2.1	Introduction	33
2.2	La Composition des services web	33
2.2.1	Définition	33
2.3	Cycle de vie d'une composition de services	34
2.4	Les propriétés fonctionnelles	35
2.5	Les propriétés non-fonctionnelles	35
2.6	Classification de composition des services web	35
2.6.1	Orchestration de services web	36
2.6.2	Chorégraphie des services web	37
2.7	Le langage BPEL (Business Process Execution Language)	38
2.8	Sélection de service web	39
2.8.1	Exemple de motivation	39
2.9	Optimisation combinatoire	41
2.9.1	Définition	41
2.9.2	Approche de sélection de services web	41
2.9.2.1	La sélection mono-objective	42
2.9.2.2	-La sélection multi-objective	43
2.9.2.3	La sélection mono et multi-objective	44
2.10	Les modèles de base de composition de service web	45
2.10.1	Séquentielle	45
2.10.2	Parallèle	46
2.10.3	Synchrone	46
2.10.4	Choix exclusif	47
2.10.5	Choix différé	47

2.10.6	Choix Multiple	48
2.10.7	Itération	48
2.11	Conclusion	49
3	Les Algorithmes d'optimisation	50
3.1	Introduction	51
3.2	Historique et Définition	51
3.3	Principe de PSO	51
3.3.1	Formalisation	52
3.4	L'algorithme	55
3.4.1	Le coefficient d'inertie	56
3.5	L'algorithme génétique	56
3.5.1	Principe des algorithmes génétiques	57
3.6	L'algorithme génétique de tri non dominé 2	58
3.6.1	Historique et Définition	58
3.6.2	Le principe de l'algorithme NSGA2	58
3.7	Qualité de Service	64
3.7.1	Définitions	64
3.7.2	Critères de QoS considérés	64
3.7.3	Calcul de la qualité de composition de SW	65
3.7.3.1	Fonctions d'agrégation	65
3.7.3.2	La fonction objectif « Score ou fitness »	66
3.8	Conclusion	68
4	Conception et Prototypage	69
4.1	Introduction	70
4.2	Présentation des outils utilisés	70
4.2.1	Le langage JAVA	70
4.2.2	NetBeans	70
4.2.3	JavaFX	70
4.3	Description d'application	71
4.3.1	Les structures de données	71
4.3.2	Structuration de l'algorithme de composition	71
4.4	Présentation d'application	72

Table des matières	12
4.5 Analyse et évaluation	75
4.6 Conclusion	77
Conclusion générale	78
Bibliographie	79

TABLE DES FIGURES

1.1	Architecture de référence des services web	24
1.2	Architecture en pile des services Web[10].	25
1.3	Structure de message SOAP.	27
1.4	Structure d'un document WSDL.	29
1.5	les trois types de l'annuaire UDDI [18].	30
2.1	Cycle de vie d'une composition de services web[22].	34
2.2	Classification des méthodes de composition des services[24]	35
2.3	Vue synthétique de l'orchestration[27].	37
2.4	Exemple de chorégraphie des services[27].	37
2.5	Exemple de composition des SW de l'agence de voyage.[30]	40
2.6	Représentation des composition possibles.	40
2.7	Les approches de sélection.[32]	41
2.8	Modèle Séquentielle.	45
2.9	Modèle Parallèle.	46
2.10	Modèle Synchronisation.	46
2.11	Modèle Choix Exclusif.	47
2.12	Modèle Choix Différé	47
2.13	Choix Multiple.	48
2.14	Modèle d'itération.	48
3.1	Déplacement d'une particule.	52
3.2	L'organigramme de principe de PSO[18].	54
3.3	Un croisement simple.	57

3.4	Une mutation pour un codage binaire.	58
3.5	Principe de NSGA II [55].	58
3.6	Crowding distance.	61
3.7	L'application de NSGA au problème de composition [56].	63
4.1	fenêtre principal de l'application.	72
4.2	affichage les services et les taches ,les contraintes et la resultat de la normalisation.	73
4.3	spécification des contraintes par l'utilisateur.	73
4.4	La fenêtre de l'algorithme de PSO.	74
4.5	La résultat de l'algorithme PSO.	74
4.6	La fenêtre de l'algorithme NSGA2.	75
4.7	resultat de meilleure composition.	75
4.8	Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.	76
4.9	Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.	76
4.10	Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.	77

LISTE DES TABLEAUX

1.1	Les couches technologiques des web services	26
3.1	Les fonctions d'agrégation des critères de QoS	66
3.2	Les intervalles des critères de QoS .[59]	66

LISTES DES ABRÉVIATIONS

B2B Busniss to Busniss

BPEL4WS Business Process Execution Language for Web Services

BPEL Business Process Execution Language

CORBA Common Object Request Broker Architecture

DCOM Distributed Component Object Model

DTD Document Type Définition

FTP File Transfert Protocol

HTML Hypertext Markup Language

HTTP Hyper text Transfer Protocol

IBM International Business Machines

MOEAs Multi-Objective Evolutionary Algorithms

NSGA-2 Non dominated sorting genetic algorithm2

OASIS Organization for the Advancement of Structured Information Standards

OMG Object Management Groupe

PSO Particle Swarm Optimization

QoS-QdS Quality of Service

RMI Remote Method Invocation

SMTP Simple Mail Transfer Protocol

SOAP Simple Object Access Protocol

SOA Service-oriented architecture

UDDI Universal Description Discovery and Integration

URI Uniform Resource Identifier

W3C World Wide Web Consortium

WSDL Web Service Definition Language

XLANG Xml Language

XML eXtensible Markup Language

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Les services Web sont basés sur le modèle Service-oriented architecture (SOA) .

Un service Web, fruit de toute une révolution, concoctée pour concrétiser une architecture SOA voit le jour, ce sont des applications auto descriptives et modulaires fournissant un modèle simple de programmation et de déploiement d'applications. L'intégration de services Web dans une application d'entreprise rend de la QdS un choix de prédilection pour les prestataires de service, voire même des consommateurs. Un fournisseur doit tenir la dragée haute pour spécifier le service ainsi fourni. Par conséquent, faire pérenniser le service Web afin de faire face aux conquérants potentiels qui garantiraient de tels prestations, Vu l'augmentation galopante de nombre de prestataire de service qui se jettent dans le même besoin fonctionnel mais avec un QdS différent, d'une part, et d'assurer une rentabilité pécuniaire, d'autre part.

Plusieurs travaux ont été utiliser dans la littérature, pour traiter les problèmes d'interaction et de composition des services web, en effet Le défi majeur dans ce domaine est que les services web sont faiblement couplées i.e., ces services sont développés par des entreprises qui sont totalement indépendantes les unes des autres. Par conséquent les utilisateurs ont besoin d'approches qui permettent la sélection et la personnalisation de leurs compositions en utilisant les critères de QOS offerts par les fournisseurs de services.

Cette sélection doit optimiser au mieux les différents attributs de QOS, en plus

elle doit prendre en compte les contraintes globales exigées par les utilisateurs.

Dans le cadre de ce mémoire, nous étudions une approche qui se base sur les Méta-heuristiques . nous nous penchons sur la remplaçabilité des services Web défaillants, engagés dans un processus métier, au lieu de modifier la composition entière, on implémenté une approche pour sélection de services web composé , tout en tenant compte des contraintes des QoS qui répondent aux exigences impérieuses et aux caprices des utilisateurs.

Notre projet a pour le but d'assurer la sélection des services web a base de Quality of Service.

On utiliser dans ce travail deux approche de sélection de services Web composites à base de l'Optimisation des essaims de particules et L'algorithme génétique de tri non dominé 2.

Le mémoire est composée en quatre chapitres :

1. **Chapitre 1** : nous entamons ce travail par une introduction et une présentation des concepts de base du domaine de recherche, notamment l'Architecture orientée service (SOA), les services web et leurs technologies principales .
2. **Chapitre 2** : parle de les différents modèles de composition des services web, ainsi que un exemple de selection et les approches de selection de service web.
3. **Chapitre 3** : on a concerne la présentation de notre approche : l'architecture globale du système et les paramètre de qos en adaptant les algorithmes d'optimisation PSO/NSGA2.
4. **Chapitre 4** : qui est dédié à l'implémentation de l'application développée; en commençons d'abord par les différentes outiles utilisées, ensuite les étapes d'implémentation et la resultat de prototypage.

Nous achèverons ce travail par une conclusion générale.

CHAPITRE 1

LES SERVICES WEB

1.1 Introduction

De nos jours, les entreprises et les systèmes d'information expriment un grand besoin pour échanger des informations et des services. Ceci nécessite des langages communs de communication. Les efforts de standardisation de ces langages ont donné lieu à un nouveau domaine de recherche connu sous le nom de « protocoles Business to Business (B2B) ». Une technologie émergente dans ce domaine a permis de tracer quelques pistes intéressantes pour la communication entre entreprises. Cette technologie est celle de services Web.

Les services Web sont un paradigme naissant qui vise à la transposition des architectures par composant dans le cadre du Web. Un service Web est un composant logiciel qui offre des services à travers une interface standardisé. La particularité des services Web est l'utilisation de la technologie Internet comme infrastructure pour la communication entre eux.[1]

1.2 Service Web

1.2.1 Définition

- Plusieurs définitions des services Web ont été proposées par des différents auteurs :
- Le consortium World Wide Web Consortium (W3C) définit un service Web comme étant « Un service Web est un composant logiciel identifié par une URI (Uniform Resource Identifier), dont les interfaces publiques sont définies et appelées en eXtensible Markup Language (XML). Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet. »
 - « Un service Web est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer, et utilise un langage standard pour décrire son interface. »[2]
 - « Les services Web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le Web. Les services Web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois

qu'un service Web est déployé, d'autres applications (y compris des services Web) peuvent le découvrir et l'invoquer. » [3]

- Selon International Business Machines (IBM) « les services Web sont des applications modulaires, auto contenus qui peuvent être publiés, localisés et invoqués à travers le web » [4]

1.2.2 Les caractéristiques des services Web

Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web qui possède les caractéristiques suivantes : [5]

- Il est accessible via le réseau.
- Il dispose d'une interface publique (ensemble d'opérations) décrite en XML .
- **Web based** : les Web services sont basés sur les protocoles et les langages du Web, en particulier HTTP (Hyper text Transfer Protocol) et XML .
- **Self-described, self-contained** : le cadre des Web services contient en lui-même toutes les Informations nécessaires à l'utilisation des applications, sous la forme de trois fonctions : trouver, Décrire et exécuter.
- **Modular** : les Web services fonctionnent de manière modulaire et non pas intégrée. Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée (ou on récupère) plusieurs applications spécifiques qu'on fait inter-opérer entre elles, et qui remplissent chacune une de ces fonctionnalités. Une fonctionnalité développée sous forme de Web services peut dorénavant être réutilisée et recombinaée à une suite d'autres fonctionnalités pour composer une Nouvelle application.
- Ses descriptions (fonctionnalités, comment l'invoquer et où le trouver?) sont stockées dans un annuaire.

1.3 Architecture Orientée Service (SOA)

Plusieurs paradigmes de développement de logiciel ont été proposés pour faciliter la création, l'exposition, l'interconnexion et la réutilisation d'applications à base de services. C'est pour cette raison que l'informatique distribuée a dû passer par le modèle client-serveur, l'architecture à plusieurs niveaux (l'orienté objet, l'orienté

composant), les standards basés CORBA (Common Object Request Broker Architecture) de l'OMG (Object Management Groupe), Java RMI (Remote Method Invocation) de Sun, DCOM (Distributed Component Object Model) de Microsoft pour enfin aboutir à l'architecture orientée service (SOA) qui est née pour répondre aux inconvénients des technologies orientées composants tels que : la complexité, le manque de compatibilité, en plus ils sont difficilement interopérables entre eux et la modification dans un composant conduit généralement à une défaillance du système. Comme, ils sont souvent utilisés en Intranet. [6]

1.3.1 Définition SOA

«L'architecture orientée services est un modèle utilisé pour transformer les composants d'un système d'information en services qui peuvent être intégrés pour construire des processus entremétiers. Les services sont fournis à d'autres composants via un protocole de communication, généralement sur un réseau. Les principes d'orientation de service sont indépendants de tout fournisseur, produit ou technologie.»[7].

Une autre définition est celle formulée par le Consortium OASIS(Organization for the Advancement of Structured Information Standards) :

« SOA est un paradigme qui permet d'organiser et d'utiliser des savoir-faire distribués qui peuvent être sous contrôle de domaines variés ».

1.3.2 Les avantages de SOA

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment :

- Une modularité permettant de remplacer facilement un composant (service) par un autre.
- Minimiser le cout de réduction.
- De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service).
- Une plus grande tolérance aux pannes.
- Une maintenance facilitée [8].

1.4 Différents types d'architecture des services web

Deux types d'architecture existent pour les services Web : La première dite architecture de référence, elle contient trois couches principales. La seconde architecture est plus complète, elle utilise les couches standards de la première architecture en ajoutant au-dessus d'autres couches plus spécifiques. Elle est appelé architecture étendue ou encore en Pile.

1.4.1 Architecture de référence

L'architecture classique de « référence » des services Web se fonde sur le modèle SOA (Service-oriented architecture) qui fait intervenir trois acteurs : le client ou l'utilisateur du service, le fournisseur de service, ainsi qu'un intermédiaire jouant le rôle d'annuaire des services.

- **L'annuaire de service (Service Registry) :** Entité logicielle intermédiaire entre les clients et le fournisseur de services. Il correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.
- **Le fournisseur de service (service provider) :** Correspond au propriétaire de service, d'un point de vue technique, il est constitué par la plateforme d'accueil du service.
- **Le client (service requestor) :** Correspond au demandeur de service, d'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer le service, l'application client peut être elle-même un service Web [9].

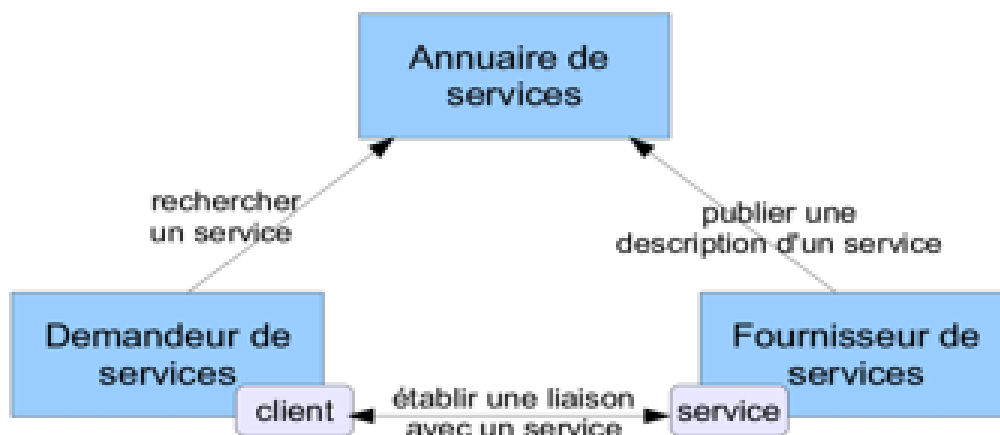


FIGURE 1.1 – Architecture de référence des services web

1.4.2 Architecture étendue

Cette architecture est aussi appelée « pile des services Web », du fait qu'elle est constituée de plusieurs couches se superposant les unes aux autres où chaque couche s'appuyant sur un standard particulier et répond à des préoccupations fonctionnelles différentes telles que la sécurité, la description, la messagerie fiable, le transport et les transactions. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Nous apportons une explication des différents types de couches de l'architecture : Nous apportons une explication de

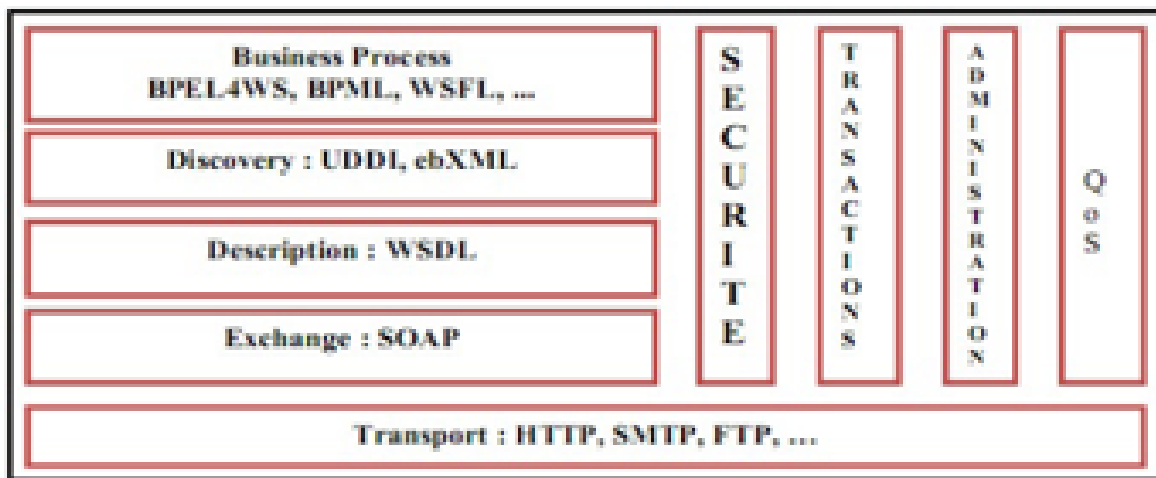


FIGURE 1.2 – Architecture en pile des services Web[10].

la mise en relief des trois types de couches (Figure 1.2) :

- **L'infrastructure de base (Discovery, Description, Exchange) :** Ce sont les fondements techniques établis par l'architecture de référence. Nous distinguons les échanges des messages établis par SOAP (W3C), la description de service par WSDL (W3C) et la recherche de services Web que les organisations souhaitent utiliser via le registre UDDI (OASIS).
- **Couches transversales (Security, Transactions, Administration, QoS) :** Ce sont des couches qui rendent viable l'utilisation effective des services Web dans le monde industriel.
- **La couche Business Processus (BusinessProcess) :** Cette couche supérieure permet l'intégration de services Web, elle établit la représentation d'un « BusinessProcess » comme un ensemble de service Web. De plus, la description de l'utilisation de différents services composant ce service est disponible par

l'intermédiaire de cette couche.

1.5 Les principales technologies de développement de service Web

Les technologies utilisées par les services Web sont XML, SOAP, WSDL, UDDI et HTTP qui sont détaillé comme suit :

- la communication est réalisée par le standard SOAP.
- la description du service est réalisée par le standard WSDL.
- les registres des services utilisent le standard universel UDDI.

UDDI	Découverte de service
WDSL	Description de service
SOAP	Communications
XML	

TABLE 1.1 – Les couches technologiques des web services

A- Communication

1.5.1 XML

XML (Extensible Markup Language, ou Langage Extensible de Balisage) est un standard promulgué par le W3C, l'organisme chargé de standardiser les évolutions du Web et le langage destiné à succéder à HTML. Comme HTML (Hypertext Markup Language) c'est un langage de balisage (markup) il représente de l'information encadrée par des balises. XML est un métalangage, ce qui veut dire que contrairement à HTML qui possède un ensemble de balises de présentation prédéfinies, il va permettre d'inventer de nouvelles balises d'isolement d'informations ou d'agrégats élémentaires que peut contenir une page Web.[11]

XML a été conçu pour des documents arbitrairement complexes. Le contenu d'un document XML est structuré par une suite d'«éléments», qui sont des blocs de texte encadré par des paires de balises ouvrantes et fermante (), qui sont les « unités de contenu». Ces éléments sont liés entre eux par une hiérarchie, certains éléments apparaissant imbriqués dans d'autres (arborescence). XML et DTD (Document Type

Définition) permettent aussi la différenciation réelle du contenu de la structure de document. [12]

1.5.2 Le Protocol SOAP

SOAP (Simple Object Access Protocol) est un protocole de communication défini à l'origine par Microsoft, puis standardisé par le W3C, avec l'élaboration de IBM. Un protocole pour l'échange d'information dans un environnement répartie, basé sur le standard XML.

SOAP permettant de définir les mécanismes d'échanges d'information entre des clients et des fournisseurs de services Web. Il s'appuie sur n'importe quel protocole de communication (HTTP, SMTP (Simple Mail Transfer Protocol), FTP (File Transfert Proctol) ...) pour transmettre les messages.[13]

1.5.2.1 La structure de message SOAP

Un message SOAP est un document XML. composé de deux parties, l'en-tête de protocole de transport et l'enveloppe SOAP.

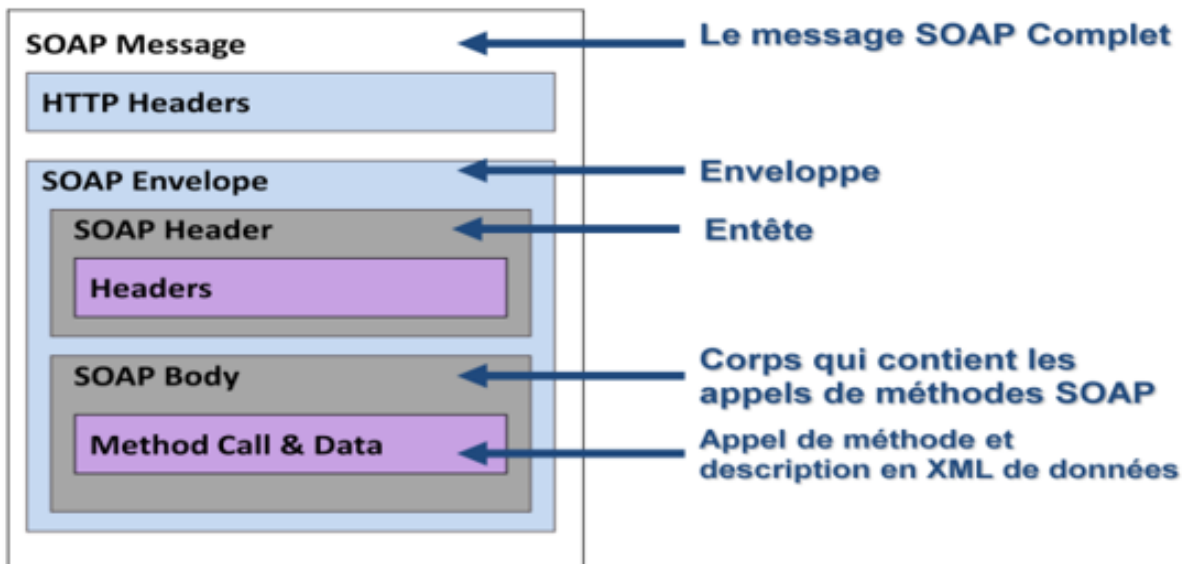


FIGURE 1.3 – Structure de message SOAP.

- **L'élément Enveloppe** : c'est l'en-tête de tous les messages SOAP qui identifie son contenu de message et comment le traiter. L'enveloppe contient la spécification des

espaces de nom définissent la version de SOAP utilisée et un attribut optionnelle « encoding Style » permettant d'identifier les règles d'encodage. Une enveloppe est divisée en 2 parties : Le header et le corps.

- **L'élément Header (entête) :** qui est optionnel et qui contient des informations concernant l'authentification, la gestion des transactions, le payement, etc.
- **L'élément Body :** contient les informations de la requête ou de la réponse, le body de la requête identifier l'objet distant, le nom de la méthode à exécuter et les éventuels paramètres ou des rapports d'erreur 'Fault'.
- **L'élément Fault :** est optionnel et fournit des informations sur d'éventuelles erreurs (le type d'erreur, une description de l'erreur et l'adresse du serveur SOAP).

B- Description de service

1.5.3 WSDL

Web Service Definition Language est un langage qui permet de décrire les services web, et en particulier, les interfaces des services web. Ces descriptions sont documents XML. Le WSDL permettant de fournir les spécifications nécessaires à l'utilisation d'un service Web en décrivant les méthodes, les paramètres et ce qu'il retourne. [14]

1.5.3.1 Structure d'un document WSDL

Défini la structure générale d'un document WSDL (Figure1.4).

- **Types :** fournit la définition de types de données utilisés pour décrire les messages échangés.
- **Messages :** représente une définition abstraite (noms et types) des données en cours de transmission.
- **Opération :** c'est la description d'une action exposée dans le port.
- **Types de Port :** décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
- **Binding :** spécifie une liaison entre un <porttype> et un protocole concret (SOAP, HTTP...).
- **Service :** indique les adresses de port de chaque liaison.

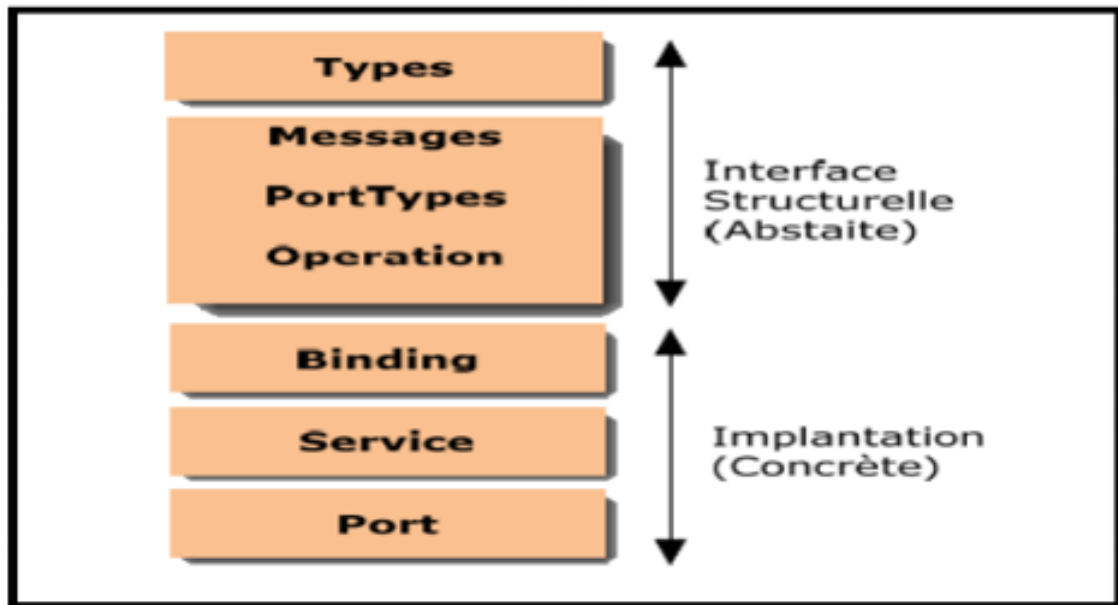


FIGURE 1.4 – Structure d'un document WSDL.

- **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.

Le document WSDL peut être divisé en deux parties. Une partie pour les définitions abstraites, tandis que la deuxième contient les descriptions concrètes.[15]

C- Découverte de service

1.5.4 UDDI

UDDI (Universal Description Discovery and Integration) c'est une norme standardisée par OASIS. C'est comme une base de registre des services web sur un format XML.

UDDI a pour objectif d'offrir aux fournisseurs la possibilité de stocker et publier leurs services et même modifier, permettre au client de découvrir, consulter les informations sur les services et sélectionner pour répondre à ces besoins.[16]

Les informations qu'il contient peuvent être séparées en trois types :

- **Les pages blanches** : elles contiennent toutes les informations pertinentes pour

identifier l'organisation (nom, l'adresse, identifiants).

- **Les pages jaunes** : elles détaillent la description de l'organisation faite dans les pages blanches (la catégorie de l'entreprise, le secteur d'activité dans lequel exerce l'entreprise, les services offerts par cette organisation, le type de service, prix, qualité de service... etc.).

- **Les pages vertes** : donnent les informations techniques sur les services web. Une fois ceci est fait, le service Web peut alors être connu de tous ceux qui le recherchent. Le modèle UDDI comporte quatre types de structures de données décrites sous forme de schéma XML : (Business entity, Business service, Binding Tmodel, Publisher Assertion).[17]

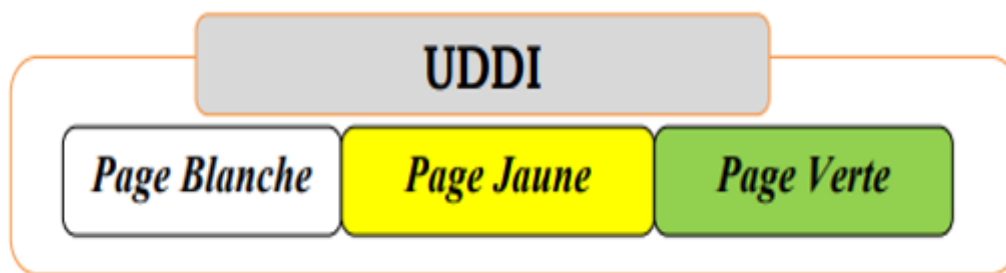


FIGURE 1.5 – les trois types de l'annuaire UDDI [18].

1.6 Les avantages et les inconvénients des services web

1.6.1 Les Avantages

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Permettent de profiter de différents environnements et langages de développement par une publication, localisation, description et une invocation via XML.
- Les services Web sont très flexibles, indépendants des langages de programmation et des systèmes d'exploitation.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des

échanges.

- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants[19][20]

1.6.2 Les Inconvénients

- La sémantique n'est pas prise en charge de façon efficace car le WSDL décrit les services de manière syntaxique.
- Les services Web ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- En l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place à travers les firewalls.
- Ils ne sont pas sécurisés à 100 % [21].

1.7 Conclusion

Dans ce chapitre, nous avons vu les services Web, ainsi que les bases de l'architecture SOA (Service-oriented architecture) qui sont l'architecture de référence et l'architecture d'étendue. Aussi bien que nous avons présenter Les principales technologies de développement de service Web (SOAP, WSDL, XML, UDDI), et introduis brièvement Les avantages et les inconvénients des services web. la composition de service web seront décrites dans le chapitre suivant.

CHAPITRE 2

COMPOSITION DES SERVICES WEB

2.1 Introduction

Quand la requête de l'utilisateur est non satisfaite par un seul service, il est inévitable de composer les services atomiques de l'environnement pour pouvoir satisfaire la demande du client. Pour cette raison, plusieurs services peuvent interagir et échanger dynamiquement des informations. L'objectif de cette interaction est l'accomplissement d'un but complexe non concevable par un seul service. Cette agrégation des compétences pour réaliser un but commun est connue par "composition de services". En parlant d'une composition de services, le problème qui se pose est "comment composer ces services?".

Le problème de composition peut être étendu, pour prendre en compte la qualité de service, et les contraintes métiers, en plus du besoin fonctionnel.

2.2 La Composition des services web

2.2.1 Définition

"La composition des services Web est le processus de construction de nouveaux services Web valeur ajoutée, à partir de deux ou plusieurs services Web déjà présents et publiés sur le Web. "

Une autre définition plus raffinée est celle proposée par Georges Gardarin, qui définit la composition des services web comme étant : « une technique permettant d'assembler des services Web afin d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôle (boucle, test, traitement d'exception, etc.) et d'échange (envoi et réception de messages). Les services composants existent au préalable et peuvent ne pas être fournis par la même organisation »[22]

L'objectif principal de la composition de Web service est la possibilité de créer de nouvelles fonctionnalités d'un nouveau Web service.

2.3 Cycle de vie d'une composition de services

La composition de services peut être vue comme étant un moyen efficace pour créer, exécuter, et maintenir des services qui dépendent d'autres.

- **L'encapsulation de services natifs** : Cette tâche garantit que tout service peut être appelé lors d'une composition, (quel que soit son modèle de données, son protocole d'interaction, etc.).
- **L'établissement d'accord d'externalisation** : Cette tâche consiste à négocier et établir les obligations contractuelles entre les services partenaires.
- **L'assemblage de services composants** : Cette tâche consiste en l'identification des services réalisant une composition donnée tout en spécifiant leurs interactions à un haut niveau d'abstraction.
- **L'exécution de services composants** : Cette tâche permet d'exécuter les spécifications de la composition satisfaisant certaines contraintes pratiques (l'efficacité, la disponibilité, etc.).
- **Le contrôle de l'exécution de services composites** : Cette étape permet la vérification de plusieurs aspects et propriétés, par exemple : elle contrôle l'accès aux services, vérifie les changements de statut et les échanges de messages.
- **L'évolutivité des services** : Cette étape assure l'évolution de la composition, (c.à.d. la modification des services invoqués, utilisation de nouveaux services, prise en compte des retours de l'activité de contrôle).

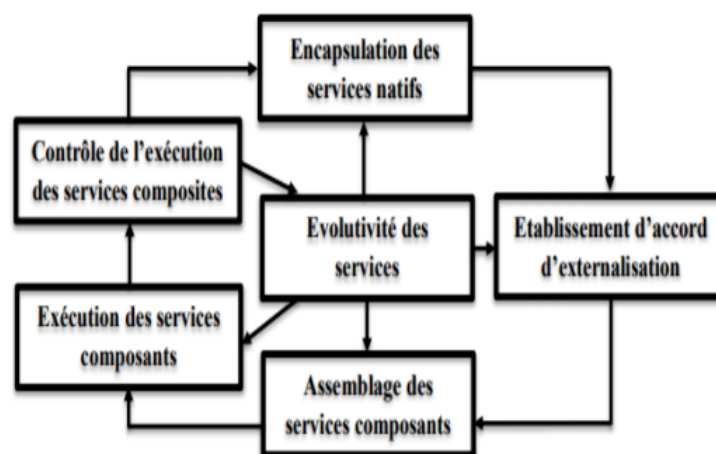


FIGURE 2.1 – Cycle de vie d'une composition de services web[22].

2.4 Les propriétés fonctionnelles

Les propriétés fonctionnelles[23] d'un service désignent les fonctionnalités que ce service peut fournir. Les propriétés fonctionnelles sont décrites dans la description de service en termes d'opérations, et reflètent le fonctionnement du service .

2.5 Les propriétés non-fonctionnelles

Les propriétés non fonctionnelles[23] de service appelé aussi : qualités de service définissent les capacités de service à fonctionner dans de bonnes conditions en termes de disponibilité, fiabilité, coût d'invocation, réputation, temps de réponse.

2.6 Classification de composition des services web

Dans la littérature nous pouvons distinguer deux axes de classification de la composition des services web :

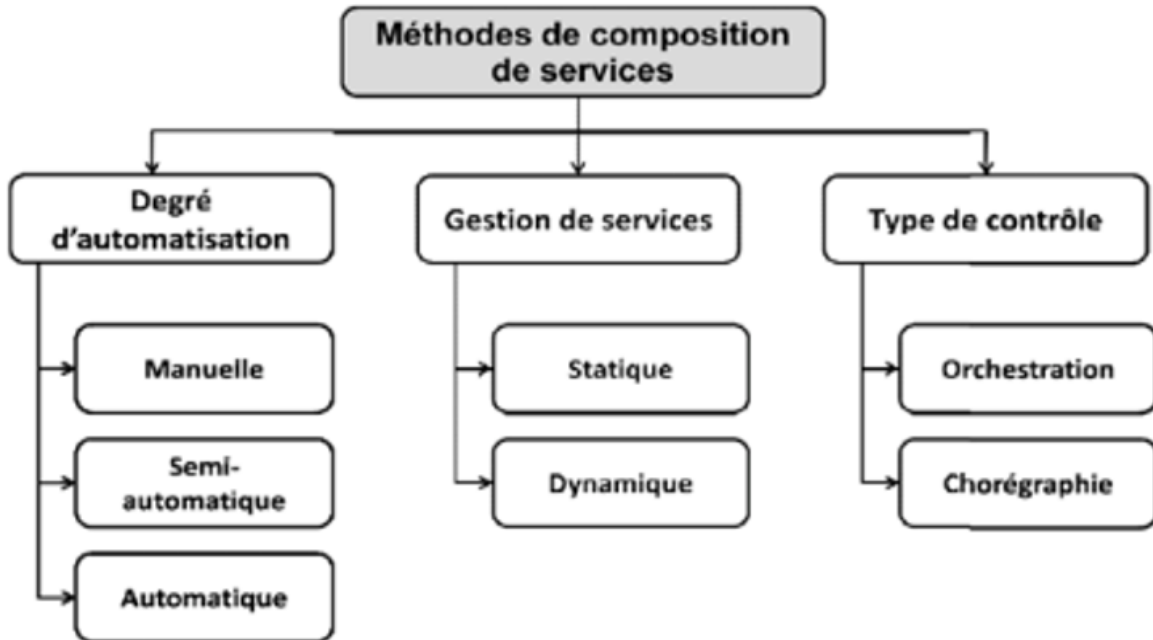


FIGURE 2.2 – Classification des méthodes de composition des services[24]

- **Premier axe :** En fonction du degré de participation de l'utilisateur dans la définition du schéma de composition (Composition manuelle, semi-automatique ou automatique) [25].

- **Composition manuelle** : La composition manuelle est basée sur l'intervention de l'utilisateur qui gère la composition sans l'aide d'outils dédiés. Donc c'est à l'utilisateur de programmer et d'implémenter la composition.
- **Composition semi-automatique** : La composition semi-automatique est constituée comme un pas en avant en comparaison avec la composition manuelle. Elle fournit des suggestions sémantiques pour aider à la sélection des services web dans le processus de composition.
- **Composition automatique** : La composition automatique prend en charge tout le processus de composition sans qu'aucune intervention de l'utilisateur ne soit requise. Cependant, la composition automatique a été et sera toujours une tâche difficile à réaliser.
- **Deuxième axe** : Selon la nature du processus de la sélection des services web (sélection à priori ou non des services composants). Dans ce cas, nous distinguons deux catégories (Composition statique ou dynamique).
- **Composition statique** : La composition statique est élaborée à la phase de la conception de l'application à base de service Web. Ainsi, les services web impliqués sont choisis, liés et agrégés avant d'être compilés et déployés. Une telle composition est plus appropriée aux environnements stables où les composants ne sont pas susceptibles de changer à court terme.

La composition statique regroupe les méthodes d'orchestration et les méthodes de chorégraphie des services web, définis dans ce qui suit :

2.6.1 Orchestration de services web

Dans une orchestration de services web, un seul processus (appelé moteur d'orchestration) est responsable de la composition et du contrôle des interactions entre les différents services. Il est le seul à connaître les services partenaires (composants), faisant coordonner de manière centralisée les différentes opérations de ces services qui n'ont aucune connaissance de cette composition [26].

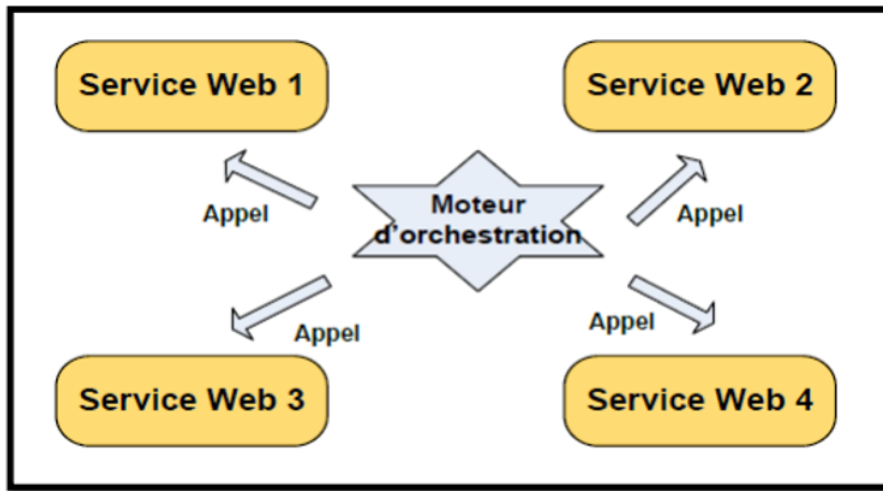


FIGURE 2.3 – Vue synthétique de l'orchestration[27].

2.6.2 Chorégraphie des services web

La chorégraphie, représente une collaboration et une coordination décentralisée des services web, pour accomplir un certain objectif où chaque service composant responsable d'une partie du workflow¹, connaît les autres services qui interagissent de manière autonome

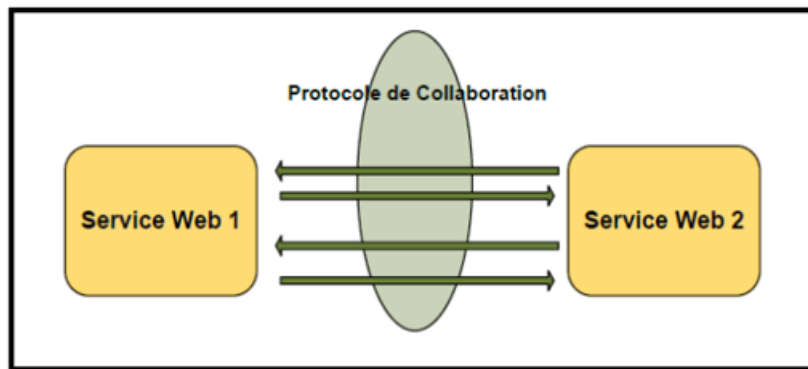


FIGURE 2.4 – Exemple de chorégraphie des services[27].

On conclut que l'orchestration de service est une approche de composition centralisée, où les services invoqués par le moteur d'orchestration ne savent pas qu'ils font partie d'un processus métier². Tandis que la chorégraphie est une approche de composition décentralisée, résultante des interactions coordonnées, des différentes parties de la composition, où chaque participant connaît le rôle qui lui appartient.

1. Workflow : Processus d'automatisation des tâches permettant un enchaînement automatisé des opérations
 2. Une transaction nécessitant des informations ou le changement d'une donnée dans une base de données.

Composition dynamique : Contrairement à la composition statique où le nombre de services fournis est limité et les services à composer sont spécifiés au préalable, la composition dynamique, elle, est initiée par une requête de l'utilisateur. Elle permet de découvrir, sélectionner et combiner dynamiquement les services à partir des annuaires de services Web, au moment de l'exécution de ladite requête, tout en tenant compte des contraintes de l'utilisateur. L'approche dynamique répond à des besoins de flexibilité et d'adaptation. Elle permet en effet de générer des plans de compositions adaptés à chaque requête à partir des services disponibles au moment de la composition. L'approche dynamique présente plusieurs avantages par rapport à l'approche statique. Outre le fait qu'elle est plus flexible et tient compte des changements qui peuvent survenir aux moments de l'exécution, elle demeure moins restrictive puisqu'elle permet de s'adapter à de nouvelles exigences de services [27].

2.7 Le langage BPEL (Business Process Execution Language)

Dans le but de créer une fonctionnalité complexe qui réutilise les services déjà existants, plusieurs langages de composition de services web ont été créés, mais un seul est devenu le formalisme le plus utilisé, le standard BPEL ou BPEL4WS (Business Process Execution Language for Web Services), standardisé par OASIS, s'est imposé depuis 2005 comme le langage standard d'orchestration de services web, il combine et remplace le WSDL d'IBM et le XLANG (Xml Language) du Microsoft.

Le BPEL est un langage basé sur XML et sur les Workflows, utilisé pour définir les processus métiers d'entreprise au sein de services Web, en décrivant le flux de données (les variables) et le flux de contrôle (les activités simples et composées, les partenaires...).

Le BPEL [11][28] est largement utilisé dans le cadre de la mise en œuvre d'une architecture orientée services, il permet la modélisation du comportement des services Web à travers les interactions d'un processus métier, La grammaire BPEL4SW peut ensuite être interprétée et exécutée par un moteur d'orchestration, contrôlé par l'un des participants du processus. Ce moteur coordonne les activités du processus et fournit des mécanismes de compensation en cas d'erreur.

- Il support deux types différents de processus :
- **Les processus exécutables** qui permettent de spécifier les détails internes du processus métier, notamment la nature et l'ordre des échanges entre partenaires. Ils s'exécutent par le biais d'un engin d'orchestration.
- **Les processus abstraits** qui représentent une interface comportementale du processus, spécifiant les échanges de messages entre les différents partenaires sans révéler leur comportement interne.

Un processus BPEL [29] spécifie l'ordre exact de l'invocation des services web participants dans la composition. L'invocation peut se faire soit en parallèle soit séquentiellement. Nous pouvons conditionner le comportement, par exemple, lorsque l'invocation d'un service web peut être dépendante du résultat d'une invocation antérieur.

2.8 Sélection de service web

Les organisations sont de plus en plus nombreuses à se tourner vers des architectures à base de services Web pour le développement et l'intégration d'applications ou de systèmes d'information. Il arrive très fréquemment que de nombreux services répondent à un même ensemble de besoins fonctionnels. Ces services se distinguent les uns des autres par leurs propriétés non fonctionnelles. On s'intéresse ici à les comparer selon des critères de QoS (cout, réputation, disponibilité, etc.). Avec la sélection des services web, on cherche à choisir le meilleur fournisseur d'un service web, étant donné un ensemble de fournisseurs de ce service.

2.8.1 Exemple de motivation

La sélection des compositions de services « QoS-aware service composition », est l'une des problématiques les plus importantes de l'architecture orientée service. Pour la présenter on considère la situation suivante : On suppose qu'il y a un utilisateur qui veut planifier un voyage, pour cela il a besoin de consommer 3 types de services au minimum, une réservation de billet d'avion, une réservation d'hôtel et une location de voiture comme montre la figure (2.5), on note aussi qu'on doit sélectionner un seul service (ou entreprise, fournisseur, ...) de chaque catégorie (ou classes) en utilisant les critères de QoS (réputation, fiabilité, couts, temps d'exécution...), en

plus l'utilisateur exige des contraintes globales sur chaque critère de QoS tel que une contrainte globale s'applique sur les 3 services sélectionnés, par exemple le cout total des 3 services ne doit pas excéder une certaine limite.

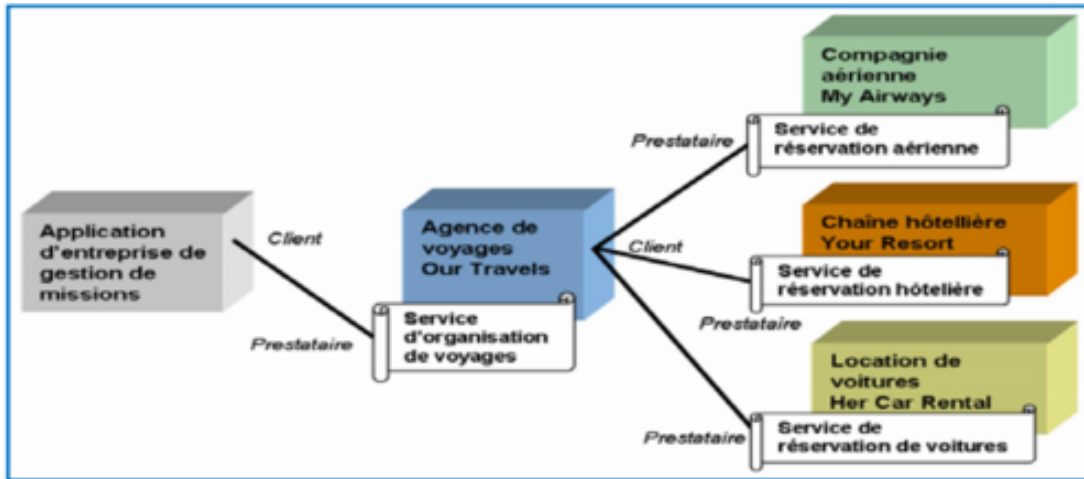


FIGURE 2.5 – Exemple de composition des SW de l'agence de voyage.[30]

Génération aléatoire de la composition de service web :

(S2 et S1 et S2) est un composition .

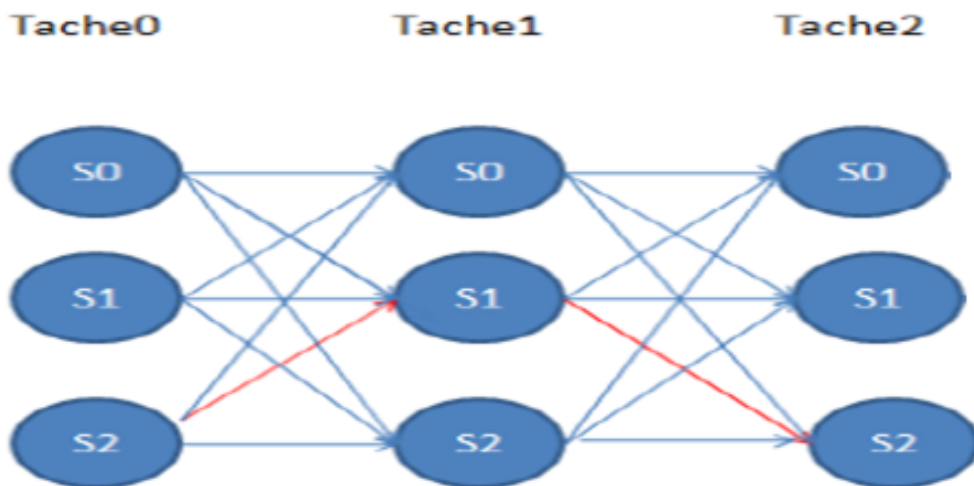


FIGURE 2.6 – Représentation des composition possibles.

2.9 Optimisation combinatoire

2.9.1 Définition

L'optimisation est une branche des mathématiques qui permet de résoudre des problèmes en déterminant le meilleur élément d'un ensemble selon certains critères prédéfinis. De ce fait, l'optimisation est omniprésente dans tous les domaines et évolue sans cesse depuis Euclide. Un problème d'optimisation en général est défini par un espace de recherche S et une fonction objective f . le but est de trouver la solution $s^* \in S$ de meilleure qualité $f(s^*)$. Suivant le problème posé, on cherche soit le minimum, soit le maximum de la fonction f [31].

2.9.2 Approche de sélection de services web

La sélection automatique de services web, a fait l'objet de plusieurs travaux, de façon générale on distingue 03 grandes classes : la sélection mono objective, La sélection multi objective, et la sélection hybride (mono et multi objective), ces classes forment une hiérarchie comme montrée sur la figure 2.7, par la suite en donnant quelques algorithmes utilisés par ces dernières.

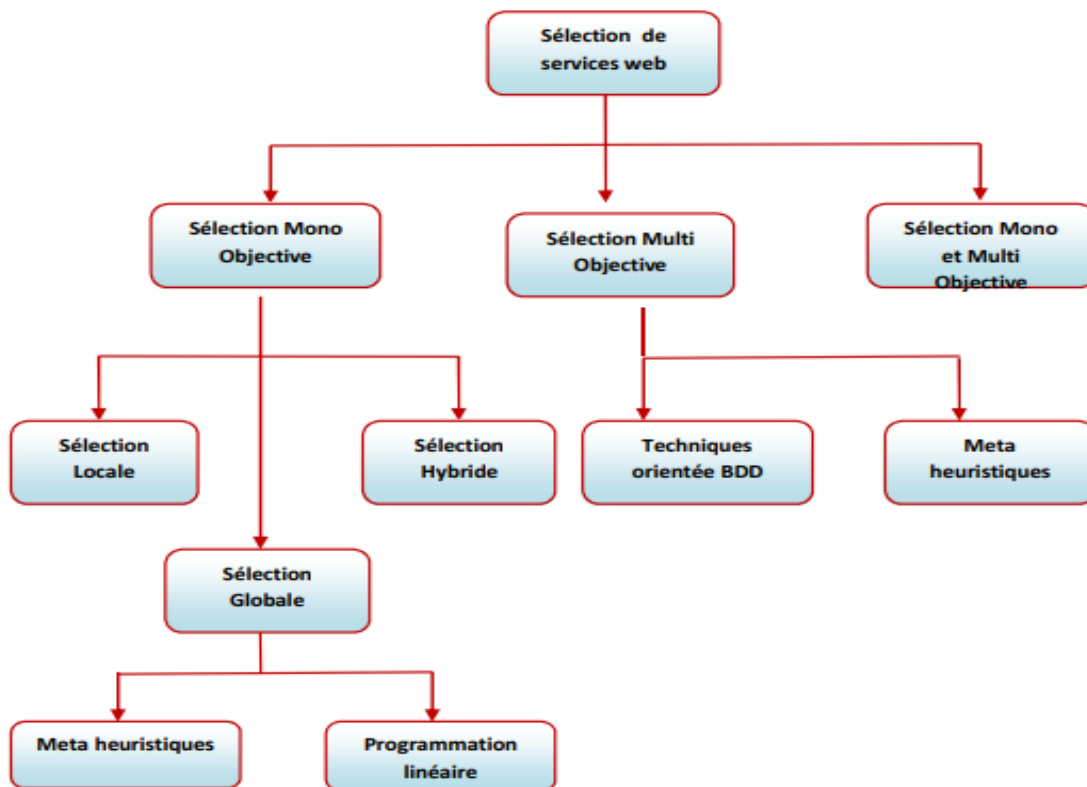


FIGURE 2.7 – Les approches de sélection.[32]

2.9.2.1 La sélection mono-objective

Cette catégorie suppose que les m valeurs de qualité de service soient agrégés en un seul score, c.-à-d. On considère une seule fonction objective qui associe des poids aux différents attributs de QoS. Cette dernière est aussi divisée en 03 sous approches (classes) : la sélection locale, la sélection globale et hybride c.-à-d globale et locale.

a -La sélection globale : Ces approches explorent un espace de recherche dont les nœuds sont des compositions complètes (c.-à-d contenant toutes les tâches), on distingue deux sous classes l'optimisation globale exacte et approximative.

b -La sélection locale : Elle consiste à filtrer un seul service de chaque classe en utilisant une fonction objective et indépendamment des autres classes, ensuite elle compose les n résultats qui correspondent aux n classes. Cette approche possède une complexité linéaire $O(1)$, en plus elle est fortement adaptée aux environnements distribuée, en effet la gestion de la QoS (mesures, mise à jours ...) est faite par des facilitateurs (brokers) distribués. [33][34] Mais nous notons en contrepartie la non prise en charge de cette classe des contraintes globales (ex : le cout global de la composition), ce qui la rend obsolète pour les problèmes réels.

[35] utilisent une approche locale qui consiste à diviser les contraintes globales, en contraintes locales en se basant sur la distribution statistiques des valeurs de QoS. Les auteurs gèrent uniquement la séquence.

L'optimisation par essaim particulaire (SPO) est utilisée dans une approche de sélection de services web pour faire aussi une optimisation mono-objective et globale. C'est une nouvelle classe des méta-heuristiques proposée en 1995 par Kennedy et Eberhart.[36], TRIBES est un algorithme SPO créé par M. Clerc dans [37]. Cet algorithme permet d'avoir un paramétrage automatique de l'optimisation par essaim particulaire.

La sélection clonale un autre algorithme utilisé pour faire une optimisation mono objective et globale sur l'espace de recherche dans une approche de sélection de services web. Dans le même but de résoudre le problème de sélection de services

web, et toujours des idées inspirées des mécanismes naturels qui ont été exploitées pour développer des heuristiques inspirées de la nature.

c -La sélection hybride : Cette approche est un compromis des deux précédentes approches, en commençant la recherche par une optimisation globale, puis continuant le travail avec une optimisation local, sa complexité temporelle est inférieure à celle de l'optimisation globale, l'approche peut également manipuler des contraintes globales.

Alrifai en [38] montre que plusieurs chercheurs ont proposé l'utilisation de la programmation de nombre entier mixte[39] pour résoudre le problème de composition de service web à base QoS. Des variables binaires de décision sont employées dans le modèle pour représenter les candidats de service. Un service candidat s_{ij} est choisi dans la composition optimale si son variable x_{ij} correspondant est placé à 1 dans la solution du modèle et jeté autrement. Pour inclure les variables de décision, le problème de résoudre le modèle peut être formulé comme problème de maximisation de valeur de service globale, sujet aux contraintes globales de QoS, tout en satisfaisant les contraintes d'attribution sur les variables de décision.

Pour faire face aux limitations des approches précédentes, nous divisons le problème averti de composition de service web à base de QoS en deux sous-problèmes qui peuvent être résolus plus efficacement dans deux phases suivantes :

Dans la première phase , le compositeur de service décompose chaque contrainte globale de QoS en contraintes locales au niveau de services de composant et envoie ces contraintes aux courtiers de service.

Dans la deuxième phase , chaque courtier de service effectue le choix local pour trouver le meilleur service composant qui satisfait les contraintes locales. Les résultats indiquent que l'approche hybride est plus performante que l'WS-UHE en termes d'optimalité.

2.9.2.2 -La sélection multi-objective

Un grand nombre d'approches existent pour résoudre les problèmes d'optimisation multi-objective. Certains d'entre eux utilisent la connaissance qu'ils ont au sujet du problème pour donner des préférences à quelques objectifs, de ce fait déviant l'as-

pect multi-objective.

D'autres donnent à tous les objectifs le même niveau d'importance,... Parmi ces approches, nous devrions distinguer deux catégories : approches non-Pareto et Pareto. Les approches de Non-Pareto ne traitent pas réellement le problème comme problème multi-objective. Elles essayent de le convertir en problème mono-objectif. D'autre part, les approches de Pareto ne transforment pas les objectifs du problème, mais essayent de les optimiser simultanément.

L'optimisation à base de skyline « the skyline based optimization » peut être manipulé en employant les techniques de base de données multiples [40], par exemple nous pouvons employer l'algorithme de clivage et conquérir, l'algorithme Bitmap, l'algorithme basé sur index (B tree, Hash table), et l'algorithme de plus proche voisin (R tree). Il existe plusieurs travaux qui tient compte des préférences d'utilisateur pour sélectionner les k top skylines dominants [41] certains d'entre eux utilisent la théorie des ensembles flous pour modeler les préférences et le rapport de dominance, les autres emploie le concept de parteo-dominance pour ranger les services web.

L'utilisation des algorithmes évolutionnaires pour résoudre des problèmes Multi-objective a été motivée principalement en raison de la nature d'AEs basée sur la population qui permet la génération de plusieurs éléments du Pareto ensemble optimal dans une seule course. Les Algorithmes Évolutionnaires multi-objective (MOEA)(Multi-Objective Evolutionary Algorithms) être parmi les méthodes de résolution les plus puissantes pour l'optimisation multi-objective [42]. MOEA tiennent compte des objectifs contradictoires et laissent trouver un ensemble des solutions non dominées.

2.9.2.3 La sélection mono et multi-objective

C'est une approche hybride qui combine les deux sélections mono et multi-objective, telle que elle effectue une recherche (ou filtrage) multi objective pour chaque classe, après elle groupe hiérarchiquement les services de chaque classe en choisissant un représentant de ces groupes, ensuite elle continue avec une recherche mono-objective (par niveau) sur les représentants des groupes (ex. la sélection clonale).

En[41] utilisent une approche de sélection hybride, telle qu'ils effectuent une re-

cherche multi objective pour chaque classe, en groupant hiérarchiquement les services de chaque classe en choisissant les services skylines car seulement ces services ne sont pas dominés par n'importe quel autre service et sont valides comme candidats pour la composition en utilisant l'algorithme intelligent Kmeans, ensuite elle continue avec une recherche mono-objective sur les représentants des groupes en utilisant le standard MIP pour décomposer les contraintes de QoS en contraintes locales, qui sont alors employées pour choisir efficacement le meilleur service de chaque classe. Les variables dans le modèle MIP de l'approche hybride représentent les niveaux locaux de QoS de chaque classe de service plutôt que les candidats réels de service, en le rendant scalable au nombre de service candidats que l'approche d'optimisation globale.

Les résultats de l'évaluation expérimentale indiquent une performance significative par rapport aux approches existantes, qui se fondent sur l'optimisation globale. Ces expériences ont prouvé que l'exécution des méthodes à base des skylines est affectée par la difficulté du problème de composition, en termes de nombre de contraintes de QoS.

2.10 Les modèles de base de composition de service web

2.10.1 Séquentielle

Les services web dans ce modèle s'exécutent séquentiellement, l'un après l'autre, dans cette séquence le service S2 s'exécute une fois que le service S1 ait terminé son exécution.

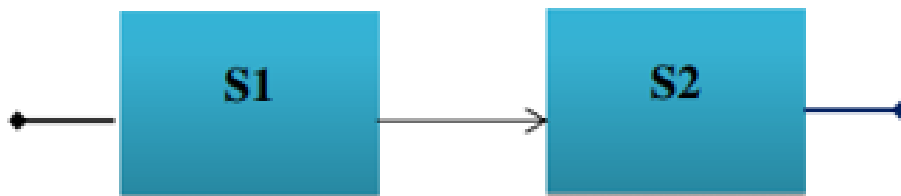


FIGURE 2.8 – Modèle Séquentielle.

2.10.2 Parallèle

Vu ces indépendances, les services dans ce modèle s'exécute simultanément.

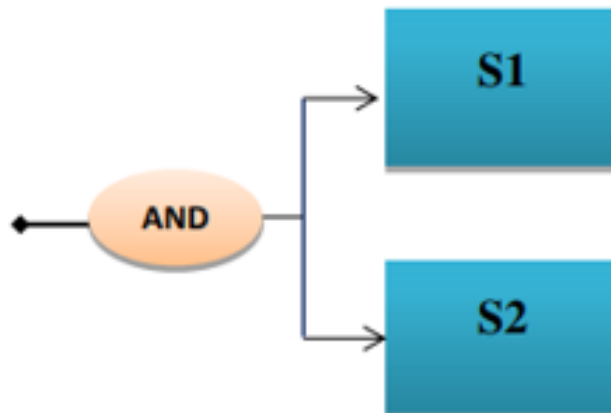


FIGURE 2.9 – Modèle Parallèle.

2.10.3 Synchrone

La synchronisation est un point dans le procédé où plusieurs flux de contrôle convergent et fusionnent dans un seul flux de contrôle synchronisant tous ces liens [43].

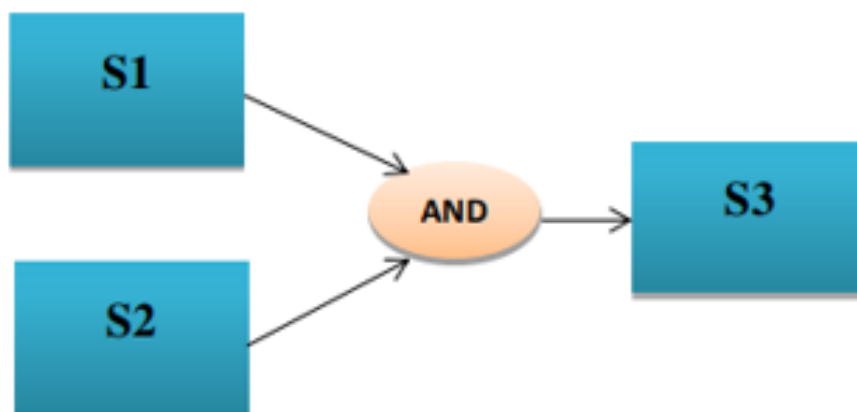


FIGURE 2.10 – Modèle Synchronisation.

2.10.4 Choix exclusif

Le choix exclusif est un point dans le procédé où un chemin est choisi parmi plusieurs, ce choix est fait à l'aide d'une décision prise au moment de l'exécution, cette décision est basée sur une information ou une donnée du procédé[43].

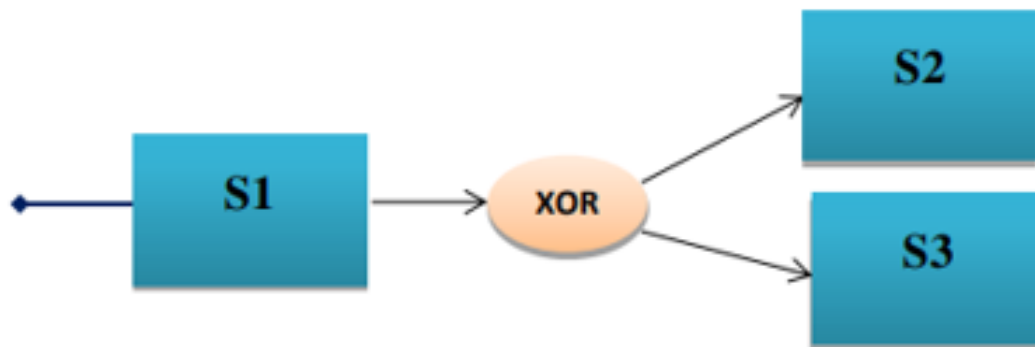


FIGURE 2.11 – Modèle Choix Exclusif.

2.10.5 Choix différé

Le choix différé est un point dans le procédé nécessite de choisir une parmi plusieurs branches alternatives. Ce choix est fait à l'aide d'une décision prise au moment de l'exécution, cette décision est basée sur une information ou une donnée du procédé.

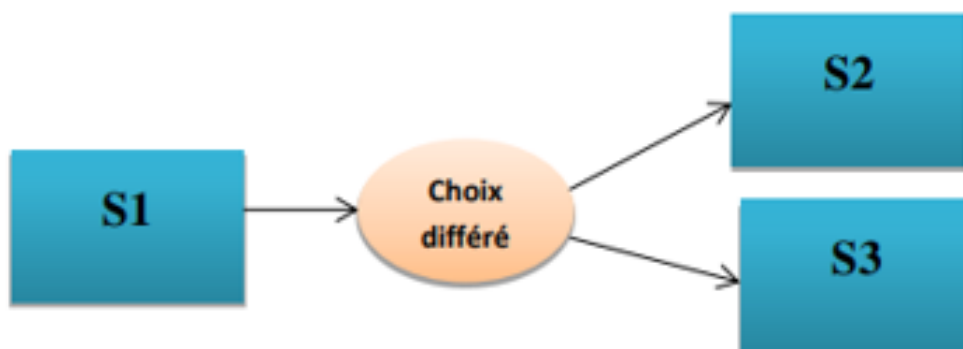


FIGURE 2.12 – Modèle Choix Différé

2.10.6 Choix Multiple

Le choix multiple est un point dans le procédé où plusieurs branches sont sélectionnées et exécutées en parallèle. Ce choix est fait à l'aide d'une décision prise au moment de l'exécution, cette décision est basée sur une information ou une donnée du procédé [43].

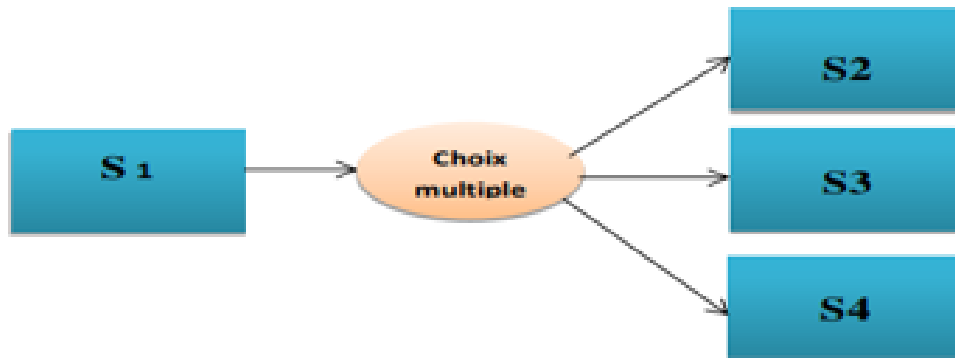


FIGURE 2.13 – Choix Multiple.

2.10.7 Itération

Ce modèle désigne l'exécution itératif, jusqu'à la satisfaction des conditions définis.



FIGURE 2.14 – Modèle d'itération.

2.11 Conclusion

Dans ce chapitre on a parlé de la composition de service web et leurs Cycle de vie et on a présenté une formalisation du problème de sélection automatique de services web avec un exemple de motivation, ensuite on a montré une partie d'état de l'art des différentes approches de sélection de services Web composites proposées dans la littérature pour résoudre ce problème. Le chapitre suivant est consacré à étudier les algorithmes de sélection et son méthodes de fonctionnement.

CHAPITRE 3

LES ALGORITHMES D'OPTIMISATION

3.1 Introduction

Les méta-heuristiques, sont des méthodes d'optimisation permettant d'obtenir une valeur approchée de la solution optimale en un temps raisonnable. Elles ont pour objectif la résolution d'un ensemble de problèmes dans différents domaines sans avoir à modifier le principe de base de l'algorithme de la méthode. Un intérêt particulier est apporté à la méthode d'optimisation approchée PSO. Elle est basée sur les « interactions sociales » entre des « agents » appelés « particules », dans le but d'atteindre un objectif donné dans un espace de recherche commun où chaque particule a une certaine capacité de mémorisation et de traitement de l'information [44].

3.2 Historique et Définition

L'optimisation par essaim particulaire (OEP), ou Particle Swarm Optimization Particle Swarm Optimization (PSO) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale au problème. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [45]. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim.

3.3 Principe de PSO

L'essaim de particules correspond à une population d'agents simples, appelés particules. Chaque particule est considérée comme une solution du problème, où elle possède une position (le vecteur solution) et une vitesse. De plus, chaque particule possède une mémoire lui permettant de se souvenir de sa meilleure performance (en position et en valeur) et de la meilleure performance atteinte par les particules « voisines » (informatrices) : chaque particule dispose en effet d'un groupe d'informatrices, historiquement appelé son voisinage.

Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, « survole » l'espace de recherche, à la recherche de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes [46] :

1. **La composante d'inertie** : la particule tend à suivre sa direction courante de déplacement .
2. **La composante cognitive** : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée .
3. **La composante sociale** : la particule tend à se diriger vers le meilleur site atteint par ses voisines.

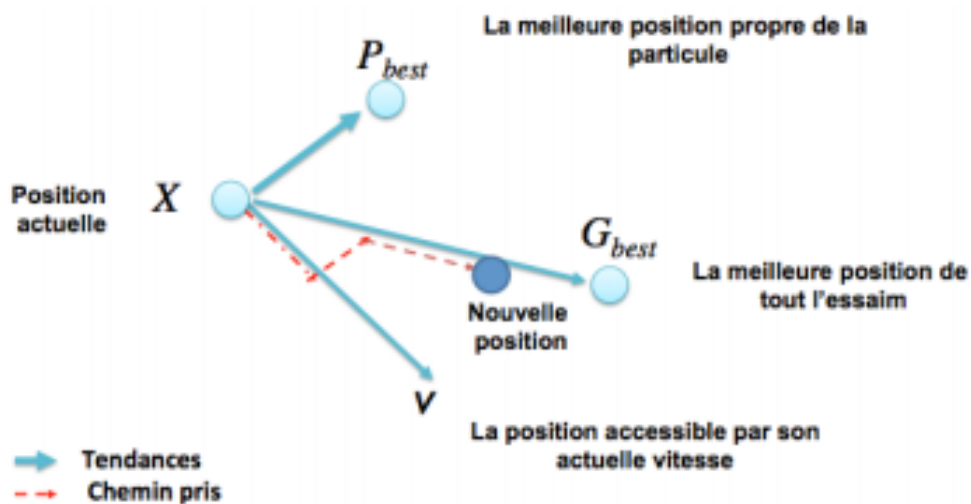


FIGURE 3.1 – Déplacement d'une particule.

3.3.1 Formalisation

Une particule i de l'essaim dans un espace de dimension D est caractérisée, à l'instant t , par :

- \mathbf{X} : sa position dans l'espace de recherche .
- \mathbf{V} : sa vitesse
- \mathbf{Pb} : la position de la meilleure solution par laquelle elle est passée
- \mathbf{Pg} : la position de la meilleure solution connue de tout l'essaim
- $f(\mathbf{Pb})$: la valeur de fitness de sa meilleure solution
- $f(\mathbf{Pg})$: la valeur de fitness de la meilleure solution connue de tout l'essaim.

Dans l'espace de recherche de dimension \mathbf{D} , une particule i de l'essaim est représentée par son vecteur position et par son vecteur vitesse ; formulés ainsi :[47]

$$\begin{aligned}\vec{X} &= (X_1, X_2, X_3, \dots, X_D) \\ \vec{V} &= (V_1, V_2, X_3, \dots, V_D)\end{aligned}$$

L'appréciation de la qualité de sa position est arrêtée par la valeur de la fonction "objectif" en ce point. Il est indispensable que cette particule puisse mémoriser la meilleure position par laquelle elle est déjà passée, formulée comme suit :

$$\begin{aligned}\vec{Pb} &= (Pb_1, Pb_2, Pb_3, \dots, Pb_D) \\ \vec{Pg} &= (Pg_1, Pg_2, Pg_3, \dots, Pg_D)\end{aligned}$$

La notion de Pg (global best), est calquée sur la version PSO Globale (PSOG) où toutes les particules de l'essaim sont issues de la particule i . Au commencement de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire/régulière dans l'espace de recherche. Par la suite, à chaque itération, les particules se déplacent, en fusionnant les trois composantes citées ci-dessus

- $w\mathbf{V}(t)$: représente la composante d'inertie du déplacement, où le paramètre w gère l'influence de la direction de déplacement sur le déplacement futur ;
- $\mathbf{C1r1}(Pb(t)-X(t))$: représente la composante cognitive du déplacement, où le paramètre $\mathbf{C1}$ gère le comportement cognitif de la particule ;
- $\mathbf{C2r2}(Pg(t)-X(t))$: représente la composante sociale du déplacement où le paramètre $\mathbf{C2}$ gère l'aptitude sociale de la particule.

nous utilisons l'algorithme de base de Kennedy & Eberhart (1995) avec un facteur de constriction (Clerc & Kennedy, 2002). Le déplacement de la particule $iV(t+1)$ entre les itérations t et $t+1$ se fait selon les deux équations (3.1) et (3.2) :[46]

$$V(t + 1) = V(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t)) \quad (3.1)$$

$$X(t + 1) = X(t) + V(t + 1) \quad (3.2)$$

avec les méta-paramètres $w = 0.729844$; $c1 = c2 = 1.496180$ [48].

- $\mathbf{C1}$ et $\mathbf{C2}$: deux constantes qui représentent les coefficients d'accélération, elles peuvent être non constantes dans certains cas selon le problème d'optimisation posé.
- $\mathbf{r1}$ et $\mathbf{r2}$: deux nombres aléatoires tirés de l'intervalle $[0,1]$.

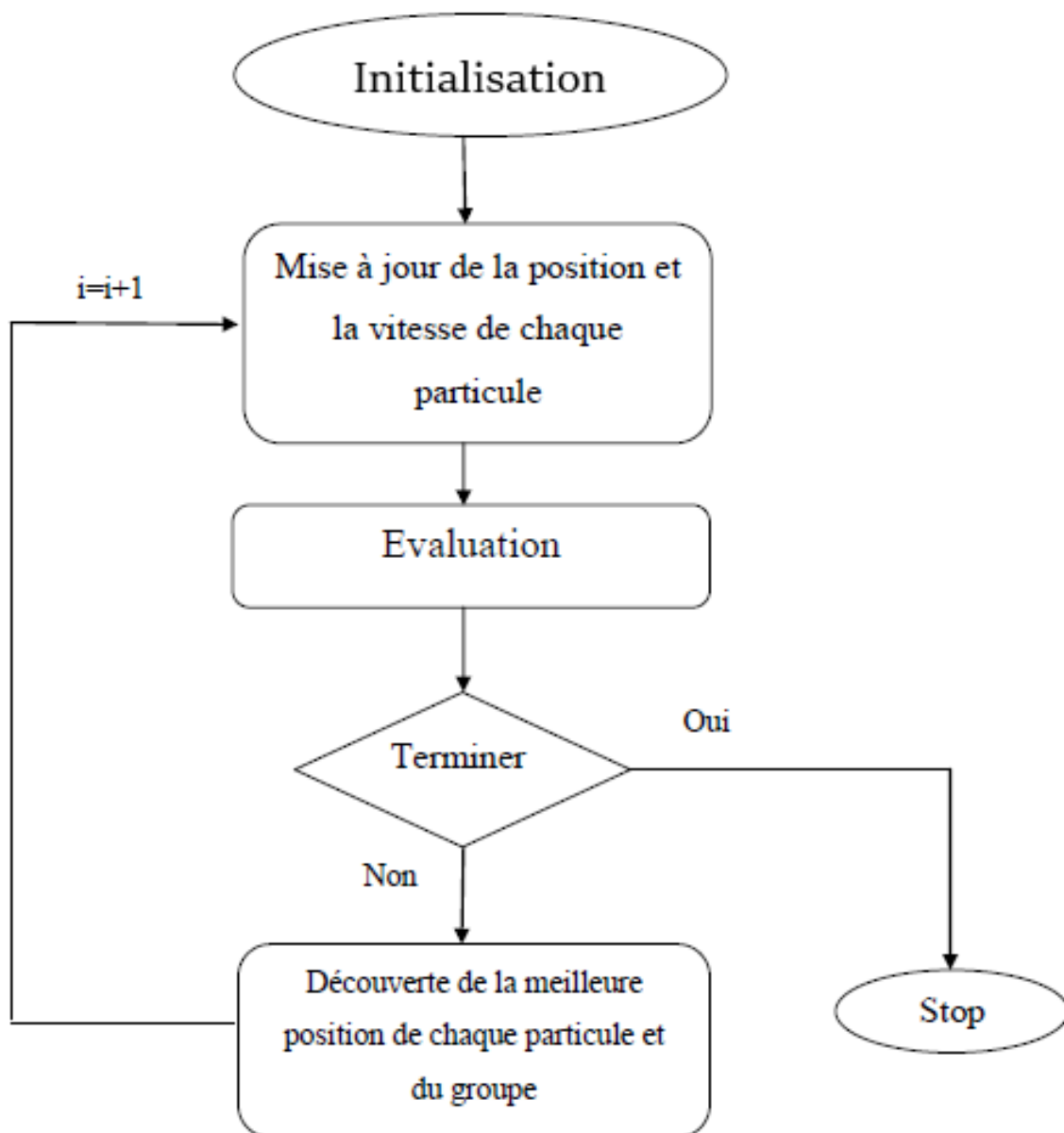


FIGURE 3.2 – L'organigramme de principe de PSO[18].

3.4 L'algorithme

L'algorithme de base de la méthode PSO proposé par (kenn) [46], commence par une initialisation aléatoire des particules dans leur espace de recherche, en leurs attribuant une position et une vitesse initiales. À chaque itération de l'algorithme les particules se déplacent selon les équations (3.1) et (3.2) et les fonctions objectif (fitness) des particules sont calculées afin de pouvoir calculer la meilleure position de toutes P_g . La mise à jour des P_b et P_g est faite à chaque itération suivant l'algorithme cité en figure ?? [49]. Le processus est répété jusqu'à satisfaction du critère d'arrêt.

L'algorithme de PSO.

Début

Initialiser les paramtres et la taille S de l'essaim ;
 Initialiser les vitesses et les positions aléatoires des particules dans chaque dimension de l'espace de recherche ;
 Pour chaque particule, $P_b = X$;
 Calculer $f(X)$ de chaque particule ;
 Calculer P_g ; // la meilleure P_b

Tant que(la condition d'arrêt n'est pas vérifiée)**faire**

Pour(i allant de 1 à S)**faire**

 Calculer la nouvelle vitesse à l'aide de l'équation (3.1)

 Trouver la nouvelle position à l'aide de l'équation (3.2)

 Calculer $f(X)$ de chaque particule ;

Si $f(X)$ est meilleure que $f(P_b)$ **alors**

$P_b = X$;

Si $f(P_b)$ est meilleure que $f(P_g)$ **alors**

$P_g = P_b$;

Fin pour

Fin tant que

Fin

3.4.1 Le coefficient d'inertie

Le coefficient d'inertie w a été introduit par (SHI,98)[50] pour contrôler l'influence de la direction de la particule sur le déplacement futur. Le but de l'introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). La formule (3.1) de calcul de la vitesse devient :

$$V(t + 1) = wV(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t))$$

La valeur de w est généralement constante, mais peut être variable dans certains cas, une grande valeur de w est synonyme d'une grande amplitude de mouvement et donc d'exploration globale de l'espace de recherche. Les études menées par (SHI,98) [50] indiquent une meilleure convergence pour w entre 0.2 et 0.8. La détermination de la meilleure valeur de ce paramètre pour chaque algorithme se fait à travers des expérimentations numériques. Dans (SHI,99) [51], les auteurs ont proposé un coefficient d'inertie dynamique qui varie au cours du temps, et diminue linéairement au cours du processus de l'optimisation. Il commence par une valeur proche de 0.9 et descend linéairement pour arriver à 0.4.

3.5 L'algorithme génétique

Les algorithmes génétiques constituent une technique de recherche et d'optimisation basée sur les principes génétiques et la théorie de la sélection naturelle développée par Charles Darwin. Cette technique a été développée pour la première fois par John Holland (Holland 1975) et popularisée par David Goldberg (David 1989).

Les algorithmes génétiques sont un des algorithmes à base de population évoluant d'une génération à une autre tout en cherchant des solutions au problème donné. Ce principe permet d'atteindre plusieurs solutions optimales ou proches des solutions optimales grâce à l'évolution de tout un ensemble d'individus. pour plus détails voir [52].

3.5.1 Principe des algorithmes génétiques

1. **La sélection** : pour faire évoluer un algorithme génétique, les individus d'une population, sur lesquels des opérations génétiques (le croisement et la mutation) seront effectuées, doivent être sélectionnés. Il existe plusieurs techniques de sélection, les principales utilisées sont Deb 1999[53] : **la sélection par rang** qui consiste à toujours choisir les individus possédant les meilleurs scores d'adaptation (fitness), le hasard n'entre donc pas dans ce mode de sélection. La technique de la roulette de fortune, où la probabilité de chaque individu à être sélectionné est proportionnelle à sa fitness. **La sélection par tournoi** utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation. La **Quelques concepts** théoriques sélection **uniforme** se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité d'être sélectionné, où est le nombre total d'individus de la population.
2. **Le croisement** : Lors de cette opération, deux chromosomes s'échangent des parties de leurs chaînes, pour donner de nouveaux chromosomes (Figure 3.3). Ces croisements peuvent être simples (sur un seul point de croisement) ou multiples (sur plusieurs points de croisement).



FIGURE 3.3 – Un croisement simple.

3. **La mutation** : il s'agit de substituer un gène avec un autre au sein d'un chromosome donné, et cela d'une manière aléatoire (Figure 3.4). De la même façon que pour le croisement, on définit ici un taux de mutation lors des changements de population. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à conserver le principe de sélection et d'évolution naturelle et ne pas transformer l'algorithme en une recherche aléatoire. La mutation sert à éviter une convergence prématurée de l'algorithme.

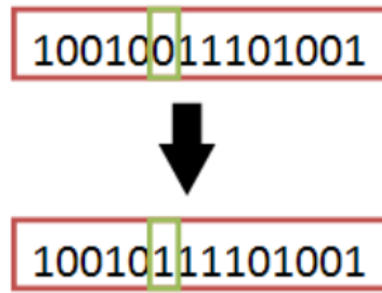


FIGURE 3.4 – Une mutation pour un codage binaire.

3.6 L’algorithme génétique de tri non dominé 2

3.6.1 Historique et Définition

NSGA-II (Non dominated sorting genetic algorithm2) est un algorithme multi-objectifs solide, proposé par Deb et al[54], En 2002. Il s’agit d’une extension et d’une amélioration de NSGA, qui a été proposée plus tôt par Srinivas et Deb, en 1995 . largement utilisé dans de nombreuses applications du monde réel. Bien qu’aujourd’hui, elle puisse être considérée comme une approche obsolète, nsga2 a toujours une grande valeur, sinon comme une référence solide à comparer. La NSGA-II génère des descendants à l’aide d’un type spécifique de croisement et de mutation, puis sélectionne la génération suivante en fonction d’un tri non dominé et d’une comparaison de distance d’encombrement.

3.6.2 Le principe de l’algorithme NSGA2

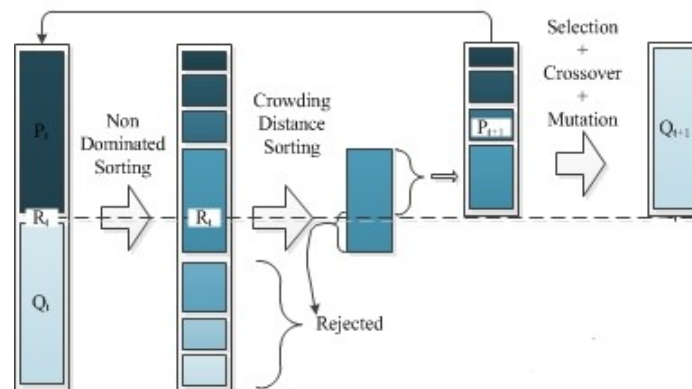


FIGURE 3.5 – Principe de NSGA II [55].

L'Algorithme suivant présente le fonctionnement du NSGA-II, qui peut être expliqué comme suit :

Initialement, une population parente $parentPop_0$ de taille N est aléatoirement créée. La population est triée en fonction de la non-dominance. Chaque solution se voit attribuée une fitness égale à son niveau de non-dominance (1 étant le meilleur niveau). Des opérateurs de sélection, croisement et mutation sont utilisés pour créer une population enfant $childPop_0$ de taille N. Ensuite, l'algorithme 1 passe à sa boucle principale

Algorithme 1 : Algorithme du NSGA-II

```

Générer (parentPop)
enfantPop = créer_nouvelle_population(parentPop)
tant que (t < nombreIteration)
    /*combiner population parent et enfant */
    newPop =  $parentPop_t \cup enfantPop_t$ 
    T = fast_nondominated_sort (newPop)
     $parentPop_{t+1} = \emptyset$ 
    /*préparer la population pour la prochaine itération*/
    tant que (  $|parentPop_{t+1}| < N$ )
        crowding_distance_assignment(T)
         $parentPop_{t+1} = parentPop_{t+1} + T$ 
    fin tant que
    trier (parentpop)
     $parentPop_{t+1} = premiersNlementsdeparentPop_{t+1}$ 
    /*utiliser la sélection, croisement, et mutation pour créer enfantPop */
     $enfantPop_{t+1} = créer_nouvelle_population(parentPop_{t+1})$ 
    t = t + 1
fin tant que

```

- Une population combinée $newPop = parentPop_t \cup childPop_t$ est formée. La population newPop sera de taille $2N$.
- La population newPop est triée en fonction de la non-domination en utilisant la fonction `fast_nondominated_sort()`. Cette fonction est définie dans Algo 2 .
- Une nouvelle population parente $parentPop_{t+1}$ est formée en ajoutant des solutions du premier front jusqu'à ce que sa taille dépasse N .
- les solutions du dernier front accepté sont triées en fonction de l'opérateur de densité \geq_n (expliqué dans les paragraphes suivants), et les N premiers points sont pris. Ainsi est construite la population $parentPop_{t+1}$ de taille N . Cette population est ensuite utilisée pour la sélection, croisement et mutation pour créer une nouvelle population $childPop_{t+1}$ de taille N également.

Algorithme 2 :la fonction `fast_nondominated_sort()`

```

Pour chaque  $p, q \in Population$ 
  Si ( $p$  domine  $q$ ) Alors       $S_p = S_p \cup q$ 
  Sinon
    Si ( $q$  domine  $p$ )      Alors  $n_p = n_p + 1$ 
  FinSi
FinPour
Si  $n_p = 0$  Alors  $F_i = F_i \cup p$ 
FinPour
 $i = 1$ 
tant que  $F_i \neq \emptyset$        $H = \emptyset$ 
  Pour chaque  $p \in F_i$ 
    Pour chaque  $h, q \in S_p$ 
       $n_q = n_q - 1$ 
      Si  $n_q = 0$  Alors  $H = H \cup q$ 
    FinPour
  FinPour
   $i = i + 1$        $F_i = H$ 
FinTantQue

```

L'opérateur de densité (crowded comparison operator) (\geq_n) guide le processus de sélection dans l'algorithme vers une répartition plus ou moins uniforme des solutions sur le front Pareto optimal. Pour cette raison, chaque individu de la population possède deux attributs :

- 1) Classement de non-domination (i_{rank})
- 2) Crowding-distance locale ($i_{distance}$)

L'opérateur \geq_n est défini comme suit :

$$i \geq_n j \text{ si et seulement si } (i_{rank} < j_{rank}) \vee ((i_{rank} = j_{rank}) \wedge (i_{distance} < j_{distance}))$$

Cela signifie que, entre deux solutions avec des niveaux de non-domination, la solution avec un i_{rank} bas est préférée. Dans le cas où les deux solutions sont du même rang, alors la solution qui possède le voisinage le moins dense est préférée. Pour obtenir une estimation de la densité des solutions entourant un point particulier i de la population, on calcule la distance moyenne entre deux points de chaque côté de i , avec chacun des objectifs. Cette quantité $i_{distance}$ représente une estimation de la taille du plus grand rectangle enfermant le point i sans y inclure d'autres points de la population (Figure 3.6).

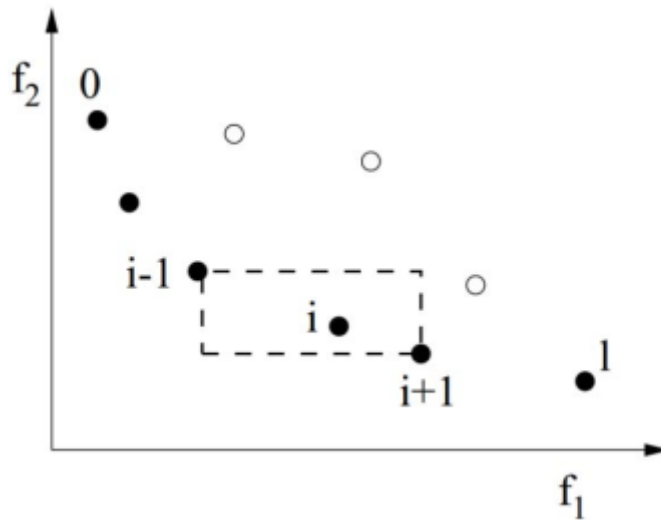


FIGURE 3.6 – Crowding distance.

Le Crowding-Distance de la i^{ime} solution sur le front (indiqué par des cercles pleins sur la Figure 3.6) est le côté moyen du cuboïde (représenté par un cadre en pointillés). L'algorithme 3 est utilisé pour calculer le Crowding-Distance de chaque point de front :

Algorithme 3 :la fonction crowding_distance_assignment(I)

nb = |I| /*nombre de solutions sur le front I */ **Pour chaque** $i \in I$

$$I[i]_{distance} = 0$$

FinPour

Pour chaque objectif m

I = trier(I,m) /*trie selon les valeurs m*/

$$I[1]_{distance} = \infty$$

$$I[nb]_{distance} = \infty$$

Si /*forcer la prise des extrémités du front*/

Pour $i = 2$ to $(nb-1)$

$$I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m)$$

FinPour

FinPour

Ici, $I[i].m$ réfère à la m^{ime} valeur de la fonction objectif du i^{ime} individu de l'ensemble

I.

$I[i+1].m$ et $I[i-1].m$ sont les voisin de $I[i].m$.

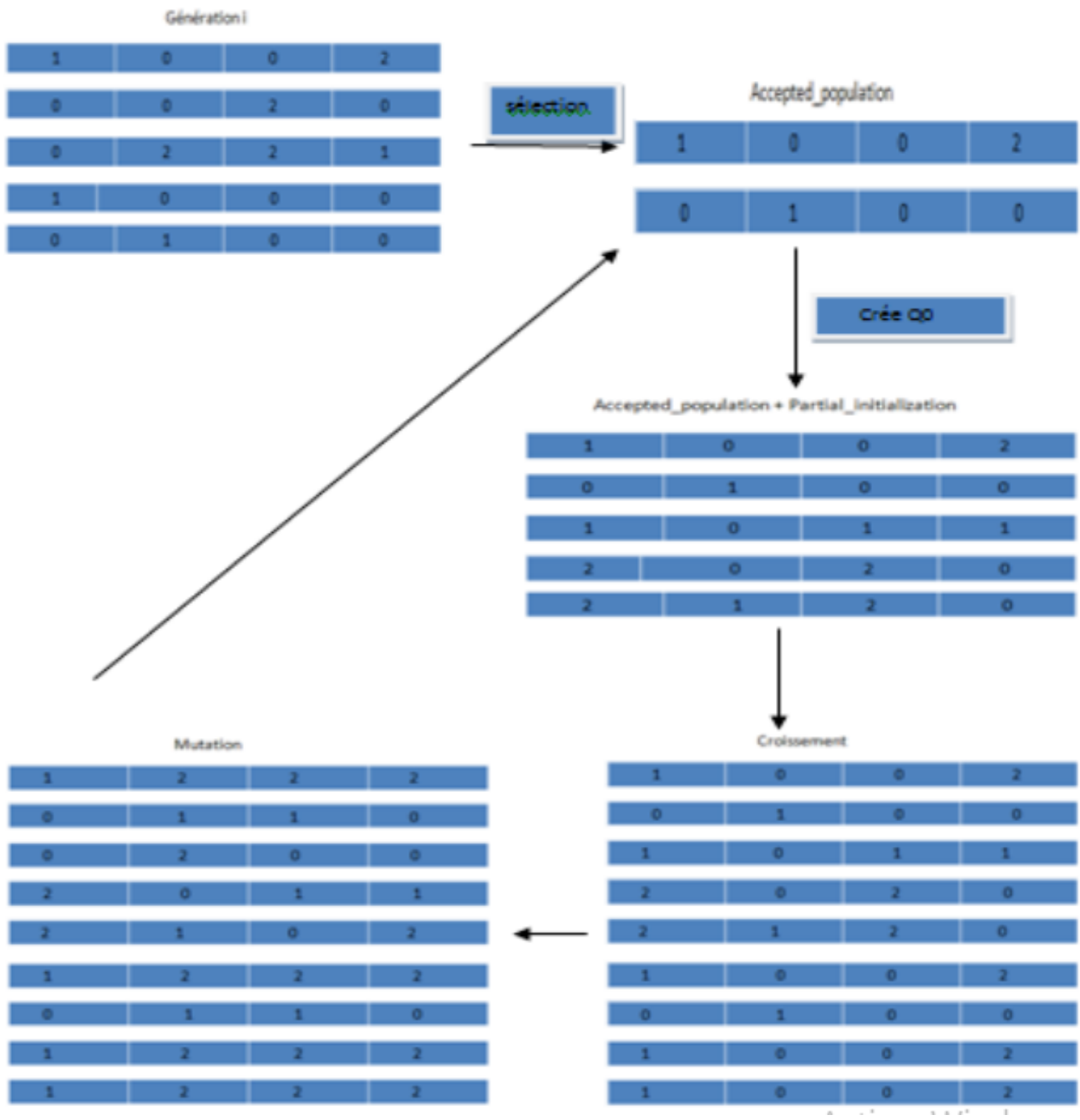


FIGURE 3.7 – L'application de NSGA au problème de composition [56].

3.7 Qualité de Service

Etant donné qu'il y a plusieurs services Web publiés sur Internet, beaucoup d'entre eux, ne peuvent satisfaire les différents besoins d'un utilisateur. Il sera donc nécessaire, d'ordonner et de classer plusieurs services Web en se basant sur différents critères de qualité. En effet, en disposant d'une bibliothèque de services Web similaires, nous pouvons faire une sélection à la main, en faisant l'écoute sur les services les plus populaire, ou guidée par un outil d'aide à la décision. La sélection des services Web a pour objectif de déterminer le service le plus adéquat.

3.7.1 Définitions

Dans le contexte des technologies de l'information et multimédia, la QoS a été Définie par Vogel et al, 1995 [57] comme "l'ensemble des caractéristiques quantitatives et qualitatives d'un système multimédia, nécessaires pour atteindre la fonctionnalité Requise par l'application".

3.7.2 Critères de QoS considérés

Dans ce qui suit nous nous focalisons sur les cinq critères de QoS suivants pour Calculer la qualité de la composition de services Web sélectionnée par notre système [58] :

1. **Coût** : est défini comme le montant (selon une certaine devise) pour exécuter l'opération.
2. **Le Temps de latence** : c'est le temps nécessaire pour traiter une requête dès l'instant de son envoi jusqu'au moment de la réception de la réponse.
 - **Formule latence** = Le temps de réception de la réponse par le client –Le temps d'envoi de la requête par le client.
 - **Quantification** : seconde.
3. **La Disponibilité** : représente le pourcentage de requêtes réussites par le fournisseur. Les réponses échouées correspondent aux exceptions reçues du côté client.
 - **Formule Disponibilité** = Nombre de requêtes réussites / Nombre total de requêtes
 - **Quantification** : Pourcentage.

4. **La Fiabilité (sûreté)** : la fiabilité des opérations est la capacité que l'opération doit être exécutée dans le délai prévu au maximum.
 - **Formule fiabilité** = $N_{\text{success}}(\text{op}) / N_{\text{invoked}}(\text{op})$,
Où $N_{\text{success}}(\text{op})$ est le nombre de fois que « op » a été exécutée avec succès et $N_{\text{invoked}}(\text{op})$ est le nombre total d'appels.
 - **Quantification** : Pourcentage.
5. **La Réputation** : de l'opération est une mesure de la fiabilité de l'opération. Elle dépend principalement du rapport à laquelle la fourniture effective du service est conforme à sa promesse.

3.7.3 Calcul de la qualité de composition de SW

3.7.3.1 Fonctions d'agrégation

Certains des critères de QoS peuvent être évalués de manière qualitative, mais la plupart d'entre eux sont des attributs quantitatifs.

Due à la présence inévitable des erreurs et des défaillances, les systèmes ne sont jamais totalement disponibles, fiable et sur. Pour cela, ces attributs sont généralement exprimés relativement en probabilité. Pourtant, généralement tous les systèmes nécessitent de définir la disponibilité, l'intégrité et la maintenabilité.

L'ensemble des critères de QoS peut être divisé en deux : attributs négatifs et attributs positifs, telle que les valeurs des attributs négatifs ont besoin d'être minimisées (temps de réponse, prix, ...), et les valeurs des attributs positifs ont besoin d'être maximisées (disponibilité, fiabilité, ...). Pour simplifier, nous considérons seulement les attributs positifs et pour les attributs négatifs il suffit de multiplier leurs valeurs par (-1).

Maintenant, nous avons besoin de calculer les paramètres de QoS pour la composition des services Web (CSW), En se basant sur la signification de chaque critère de QoS, nous définissons un ensemble de fonctions d'agrégation permettant de calculer les valeurs agrégées des critères de QoS pour la CSW.

Les fonctions d'agrégation de chaque critère sont données par le tableau 3.1 :

Alors la qualité du web service composite peut être décrite par un vecteur de QoS comme suit :

$$QoS(CSW) = (sur(CSW), dis(CSW), rep(CSW), lat(CSW), cou(CSW)).$$

Critères de QoS	Fonctions d'agrégation
Sûreté	$\prod_{i=1}^n sur(opi)$
Disponibilité	$\prod_{i=1}^n dis(opi)$
Réputation	$1/n \sum_{i=1}^n rep(opi)$
latence	$\sum_{i=1}^n lat(opi)$
Coût	$\sum_{i=1}^n cou(opi)$

TABLE 3.1 – Les fonctions d'agrégation des critères de QoS

La latence et le coût prennent des valeurs scalaires dans \mathbb{R}^+ , la disponibilité et la fiabilité (sûreté) représentent des valeurs de probabilité (une valeur réelle entre 0 et 1), et la réputation varie sur l'intervalle $[1,5]$, voir le tableau suivant :

Critères de QoS	Valeurs
Sûreté (sur)	0.5 - 1
Disponibilité (dis)	0.7 - 1
Réputation (rep)	1 - 5
latence (lat)	0 - 300 s
Coût (cou)	0- 30 \$

TABLE 3.2 – Les intervalles des critères de QoS .[59]

3.7.3.2 La fonction objectif « Score ou fitness »

Ces critères doivent optimiser une fonction globale qui maximise les critères positifs (Réputation, Disponibilité et Fiabilité) et minimise les critères négatifs (cout et temps d'exécution).

La qualité d'un service web peut être définie comme un vecteur de QOS :

$$Qualité(S_i) = (latence(S_i), fiabilité(S_i), disponibilité(S_i), coût(S_i), réputation(S_i)).$$

$$Q_i(C) = \sum_{j=1}^n Q_i(S_j)$$

Le but de fonction de score c'est pour calculer une valeur hors du vecteur de QOS des services. Cela peut faciliter la comparaison de la qualité des services. Comme les utilisateurs peuvent avoir des préférences sur la façon dont leurs requêtes sont traitées. [60]

$$\begin{aligned}
 Fctobj(sol) &= (Cout(sol), Rep(sol), Disp(sol), Fiab(Sol), Temp(sol)) \\
 &= \left(\sum_{Q_i \in Neg} W_i \frac{Q_{i,max} - Q_i}{Q_{i,max} - Q_{i,min}} + \sum_{Q_i \in Pos} W_i \frac{Q_i - Q_{i,min}}{Q_{i,max} - Q_{i,min}} \right) + penalit(sol)
 \end{aligned}$$

Où :

$$Penalit(sol) = - \sum_{i=1}^R -(D_i^2)$$

avec :

$$D_i = \begin{cases} 0 & \text{si } Q_i \geq B_i \\ |Q_i - B_i| & \text{sinon} \end{cases} \quad \text{pour les criteres negatifs}$$

$$D_i = \begin{cases} 0 & \text{si } Q_i \leq B_i \\ |Q_i - B_i| & \text{sinon} \end{cases} \quad \text{pour les criteres positifs}$$

Où Q_i est l'ième qualité de service, Q_i est calculé par la fonction d'agrégation, Neg et Pos sont les ensembles de QoWS négatifs et positifs respectivement. En négative (resp. positive) des paramètres, le plus élevé (resp. inférieure) de la valeur, le pire est la qualité. Q_i max est la valeur maximale pour l'ième paramètre QoWS pour tous les plans possibles d'exécution du service et Q_i min est le minimum.

la fonction fitness est calculée . Dans cet algorithme, M est un grand nombre (environ 10000000) et à la fin, si la valeur d'ajustement était inférieure ou égal à 0, ce plan de composition sera un plan approprié [54].

Function FitnessFunc()

```
Function FitnessFunc()  
    M=Big number ;  
    Fit=0 ;  
    For all web services in composition plan do  
        If  $Agg_d$  better than  $Con_d$   
            Fit=Fit- $|Agg_d-Con_d|$  ;  
        Else  
            Fit=Fit+M* $|Agg_d-Con_d|$  ;  
        End  
    End  
    Return Fit ;  
End
```

3.8 Conclusion

Dans ce chapitre nous avons commencé par une représentation sur l'algorithme PSO(définition,principe), Puis nous allons présenter le fonctionnement de l'algorithme NSGA2.Ensuite nous allons voir les critères de Qos et la fonction objectif.Le chapitre qui ce suit présente notre travail implémenté sous forme d'une application, l'implémentation et l'environnement de travail.

CHAPITRE 4

CONCEPTION ET PROTOTYPAGE

4.1 Introduction

Dans le présent chapitre on s'intéresse à implémenter et à valider l'approches proposés pour le problème de composition de services au chapitre précédent, nous allons commencer par présenter les différents outils techniques liés à l'implémentation, puis nous évaluons les performances de chaque approche en comparant ses résultats entre- eux.

4.2 Présentation des outils utilisés

4.2.1 Le langage JAVA

Pour le langage de programmation notre choix s'est porté sur le langage JAVA, car il est un langage orienté objet et simple, ce qui réduit les risques d'incohérence. Il est portable, c.à.d., il peut être utilisé sous Windows, Linux, Macintosh et d'autres plateformes sans aucune modification. En fin, il possède une riche bibliothèque de classes comprenant de diverses fonctions.

4.2.2 NetBeans

NetBeans est un environnement de développement intégré pour développer principalement avec Java, mais aussi avec d'autres langages, en particulier PHP, C / C ++ et HTML5. Il s'agit également d'un cadre de plate-forme d'application pour les applications de bureau Java et autres. L'EDI NetBeans est écrit en Java et peut s'exécuter sur Windows, OS X, Linux, Solaris et d'autres plates-formes prenant en charge une JVM compatible. La plate-forme NetBeans permet de développer des applications à partir d'un ensemble de composants logiciels modulaires appelés modules. Les applications basées sur la plate-forme NetBeans peuvent être étendues par des développeurs tiers.

4.2.3 JavaFX

Avec l'apparition de Java 8 en mars 2014, JavaFX devient la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web),

le développement de son prédécesseur Swing étant abandonné (sauf pour les corrections de bogues). JavaFX est désormais une pure API Java (le langage Descript spécifique qui a été un temps associé à JavaFX est maintenant abandonné). JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc. Le SDK de JavaFX étant désormais intégré au JDK standard JavaSE, il n'y a pas besoin de réaliser d'installation spécifique pour JavaFX.

4.3 Description d'application

4.3.1 Les structures de données

Dans notre exemple nous avons générer un nombre aléatoire de services Web dont chaque service est caractérisé par cinq critères de QOS et chaque critère est généré aléatoirement comme suit :

- $0 < \text{cout} < 30$
- $0 < \text{temps de réponse} < 300$
- $0,7 < \text{disponibilité} < 1$
- $0,5 < \text{fiabilité} < 1$
- $1 < \text{réputation} < 5$

4.3.2 Structuration de l'algorithme de composition

L'implémentation en JAVA de l'approche pso pour la composition nécessite la mise en œuvre de trois types de classes (un ensemble de classes pour la normalisation, génération, tache, service et un ensemble de classes pour les opérations de l'algorithme pso et un ensemble de classes pour les opérations de l'algorithme de NSGA2 et un ensemble de classes pour les opérations d' IASS).

4.4 Présentation d'application

L'application est présentée en seq Onglets :

- Onglet 1 : représente la phase de Data set (la base , contraintes ,normalisation des services).
- Onglet 2 : représente la phase de l'algorithme de PSO.
- Onglet 3 : représente la phase de l'algorithme de NSGA2.
- Onglet 4 : est fait le rôle de représentation d'algorithme PSO/NSGA2 (afficher le graphe orienté).
- Onglet 5 : est une comparaison (graphe-histogramme) entre les deux algorithmes .

Le premier onglet : pour la génération et le chargement de la base de données (les services web et les tâches) et saisié les contraintes,les poids ,afficher aléatoire les normalisations des services .

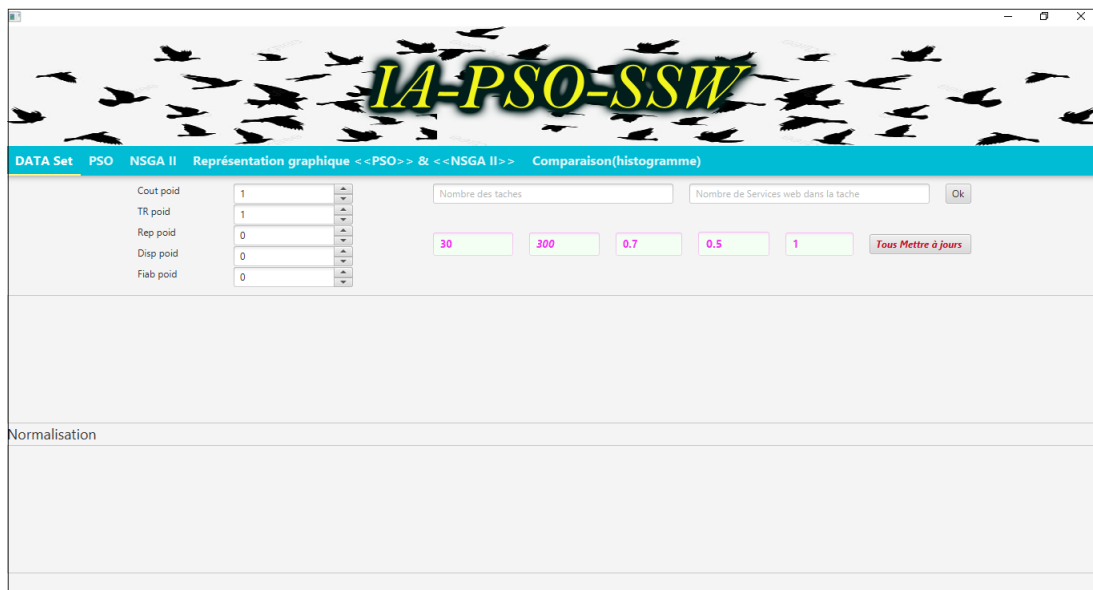


FIGURE 4.1 – fenêtre principal de l'application.

on a choisé pour cette simulation qu'on a 5 services pour chaque tâche et le nombre de tâches a exécutés égale à 7,et appyué sur le bouton ok, voici la figure :

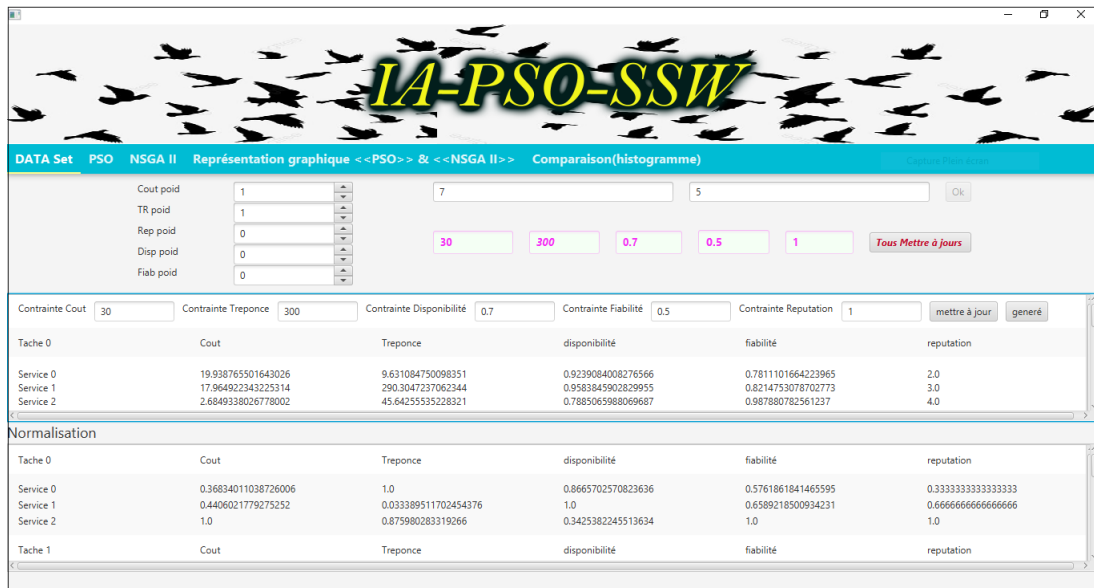


FIGURE 4.2 – affichage les services et les taches ,les contraintes et la resultat de la normalisation.

par exemple l'utilisateur a spécifié que le coût doit être inférieur à 20 et le Tr doit être inférieur à 200,et poids de cout 1 et poids de Tr 1 ,et appyué sur le bouton Tout mettre a jours.

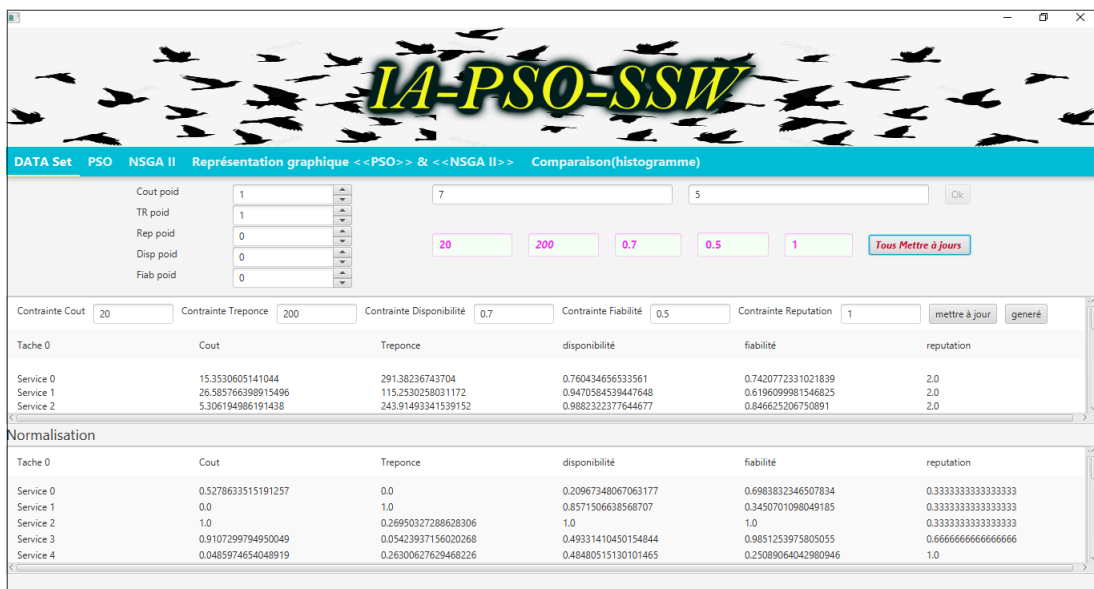


FIGURE 4.3 – spécification des contraintes par l'utilisateur.

Le deuxième onglet : pour la génération de le coefficient d’inertie w et le coefficient d’accélération $C1, C2$.

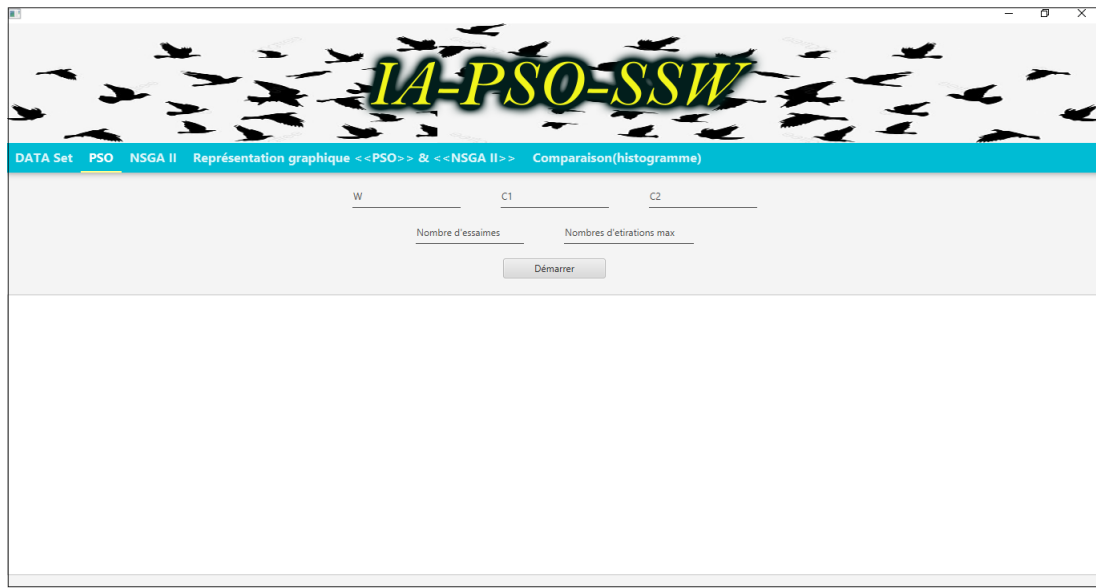


FIGURE 4.4 – La fenêtre de l’algorithme de PSO.

l’utilisateur Remplit $W, C1, C2$ avec les méta-paramètres de [48], et Nombre D’essaims égale 7 et Nombre d’éterations 3, en clic sur le bouton Démarrer.

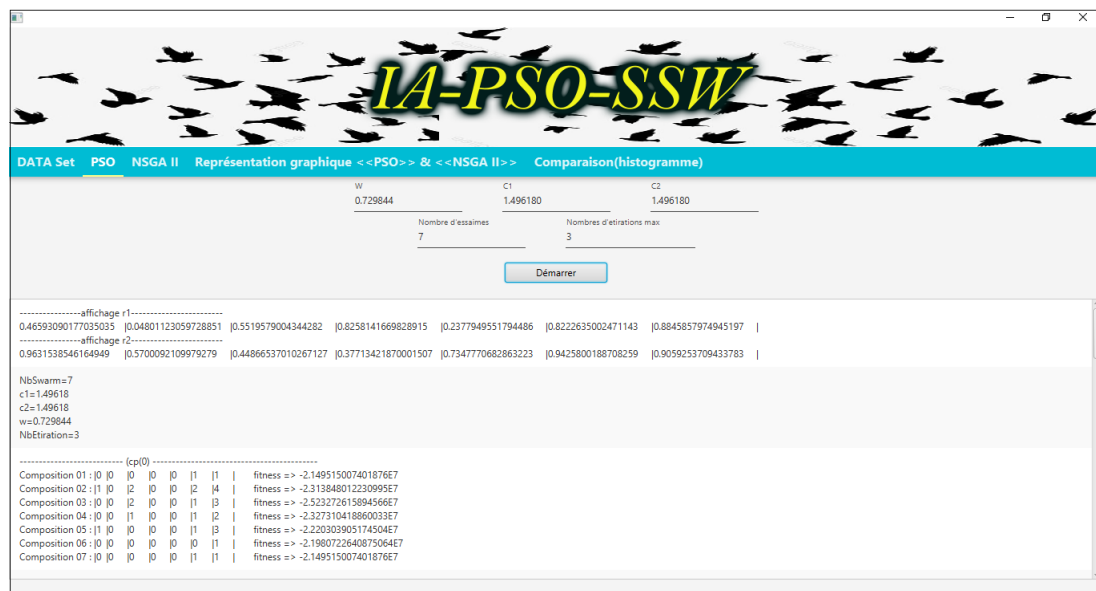


FIGURE 4.5 – La résultat de l’algorithme PSO.

La troisième onglet :Afficher la resultat de l’algorithme de NSGA2.

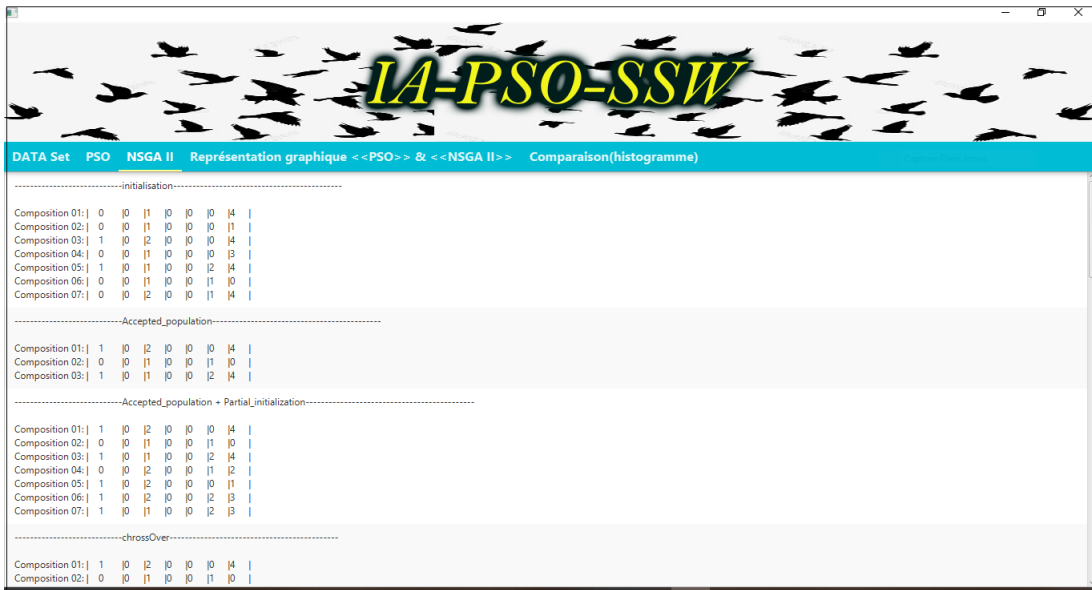


FIGURE 4.6 – La fenêtre de l’algorithme NSGA2.

La quatrième Onglet : chaque algorithme afficher la meilleure composition.

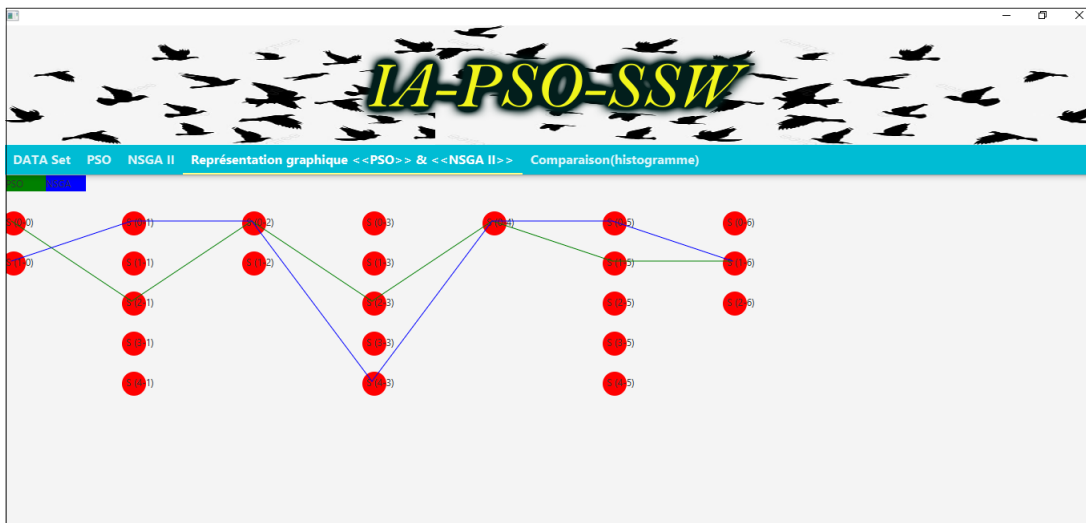


FIGURE 4.7 – resultat de meilleure composition.

4.5 Analyse et évaluation

la cinquième onglet :représente La vue graphique du résultats d’implémentation des deux algorithmes en comparant la meilleure composition comme se suit :

1- pour nombre des taches égale à 7 et nombre des services égale à 5 et nombre d’étiration 3 : On remarque que le temps de reponse de PSO est meilleure que le

temps de réponse de NSGA, on voit que le fitness de PSO meilleure par rapport le fitness de NSGA2.



FIGURE 4.8 – Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.

- nous avons choisé maintenant une autre itération pour un nombre de tache égale 7 et nombre de service a égale 5 ,nous avons fait 150 iteration .Sachant que nous gardons les mêmes paramètres d'entrées : On remarque que l'algorithme Pso et nsga2 nous donne le même temps de réponse qui ne dépasse pas les onze (11)ms.Quant la fitness, l'algorithme pso nous donne solution de bonne qualité par rapport le NSGA2.

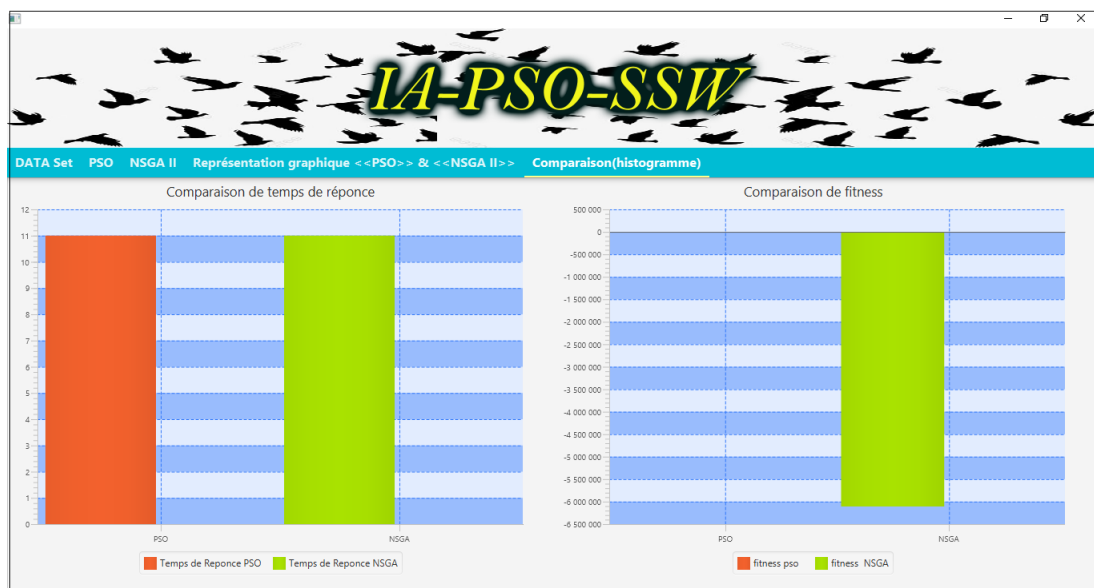


FIGURE 4.9 – Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.

3- pour nombre des taches égale à 7 et nombre des services égale à 5 et nombre d'étiration 1000 : On remarque que l'algorithme PSO est pris un temps de reponse moins que le NSGA2 , et le même pour le fitness de PSO c'est mieux que le fitness de NSGA2.



FIGURE 4.10 – Temps de réponse et le fitness pour trouver la meilleur composition des 2 algorithmes.

4.6 Conclusion

Dans ce chapitre nous avons implémenté deux algorithmes d'optimisation PSO et NSGA2 pour la sélection de service web. Nous avons présenté les résultats de la validation des approches, afin d'évaluer l'efficacité de chaque méthode. D'après les résultats qu'on a obtenus, on a confirmé l'efficacité de NSGA2 et l'algorithme PSO dans le domaine de sélection et la composition des services.

CONCLUSION GÉNÉRALE

Les services web représentent une technologie essentielle pour la concrétisation des applications distribuées, ils permettent aux entreprises et individus de publier des liens vers leurs données et leurs applications de la même manière qu'ils publient des liens vers leurs pages web. La composition des services web, notamment, est considérée comme un point fort car elle apporte une solution au problème des besoins complexes du client.

Dans ce mémoire nous avons étudié un algorithme de sélection qui se base sur l'Optimisation par Essaim Particulaire. sous ce dernier est une métaheuristique destinée à la résolution de problèmes à variables continues, inspirée du comportement en essaim de certains animaux. Ensuite nous avons étudié un notre algorithme NSGA2 de sélection que ce base sur les algorithmes évolutionnaires pour l'optimisation des composition .

Notre prototype sélectionne les compositions de services les plus satisfaisantes, en se basant sur cinq(05) critères de QOS : Latence, fiabilité, disponibilité, coût, et réputation. La composition concrète recherchée doit maximiser un ensemble de ces critères positifs et minimiser un ensemble d'autres critères négatifs, en plus elle doit satisfaire un groupe de contraintes globales.

- [1] MEGHERBI SAMI, Une approche basée agent pour la sélection des Services Web , mémoire de master, Université Mohamed khider Biskra,2018.
- [2] Tarek Melliti, Interopérabilité des Services Web complexes, Thèse de Doctorat, Université Paris IX Dauphine, le 8 Décembre 2004.
- [3] Julien Ponge. Model Based Analysis of Time-aware Web Services Interactions. Thèse de Doctorat de l'Université Blaise Pascal - Clermont-Ferrand II, dans le cadre de l'Ecole Doctorale des Sciences pour l'Ingénieur, France, 2008.
- [4] Visión general de servicios Web - IBM,eclipse.
- [5] Ethan Cerami, Web Services Essentials, édition O'Reilly, Février 2002.
- [6] BACHTARZI, Fayçal. Une Approche de Composition des Services Web Basée Transformation de graphes. université costantine,vol. 2, 2014.
- [7] ZUO, Wei. Managing and modeling web service evolution in SOA architecture. 2016. Thèse de doctorat. Université de Lyon.
- [8] Web-services,SOA-Architecture Orientee Services,commentcamarche, Juin 2013.
- [9] P. Kellert and T. Toumani. Les Web Services Sémantiques. Action spécifique 32 CNRS/STIC, Octobre 2003.
- [10] A. Bertino, M. Keidl, et E. Kemper, A Framework for Contextaware Adaptable Webservices, EDBT, LNCS 2992, Springer- Verlag Berlin Heidelberg, 2004.
- [11] ALILI Sofiane-ZIANI CHERIF Bassim,Réalisation d'un SERVICES WEB Exemple d'un Compte Bancaire, UNIVERSITE ABOU BAKR BELKAÏD – TLEMCEM,le 27 Juin 2013.

- [12] Chauvet J., “Service Web avec SOAP, WSDL, UDDI, ebXML“ ; jouve, Paris ; Mars 2002 .
- [13] Batiste Bieler, Etude de faisabilité d’une application SOAP avec un système embarqué, école HE-ARC ingénierie informatique, 2005.
- [14] Roberto Chinnici, et Martin Gudgin, Web Services Description Language, édition W3C, le 22 Août 2004.
- [15] Chinnici R., Gudgin M., Moreau J. and Weerawarana S. “Web Services Description Language (WSDL) Version 1.2 ”, W3C Working Draft ,Janvier 2003,
- [16] carolgeyer,The UDDI XML,le 18 Septembre 2006.
- [17] Céline Lopez-Velasco, Sélection et composition de services Web pour la génération d’applications adaptées au contexte d’utilisation, pour obtenir le grade de Docteur de l’université JOSEPH FOURIER, 18 Novembre 2008.
- [18] Benamar Abdeladim-Ait Hamouda Mounir,Sélection De Services Web Par L’optimisation D’essaim Particulaire Hybride,UNIVERSITE ABOU BAKR BELKAÏD – TLEMCEN,le 15 Juin 2013,.
- [19] Avantages de Services Web, wikipedia,le 20 Juin 2016.
- [20] David Chappell, et Tyler JEWELL, Java Web Service, édition O’Reilly, Mars 2002.
- [21] XML Schema instance,le 16 Mars 2001.
- [22] G. Gardarin, “Xml, des bases de données aux services web (in french)”, Dunod, Novembre 2002.
- [23] Canfora, G., Penta, M. D., Esposito, R., et Villani, M. L. An approach for qos-aware service composition based on genetic algorithms. In Proceedings of GECCO,pages 1069–1075, Janvier 2019.
- [24] Y. Charif, e. N. Coordination in Introspective MultiAgent Systems. California, USA : Proceeding of the International Conference on Intelligent Agent Technology (IAT07),2007.
- [25] M. M. Akbar, E. G. Manning, G. C. Shoja, and S. Khan. Heuristic solutions for the multiple-choice multi-dimension knapsack problem. In Proceedings of the International Conference on Computational Science-Part II, pages 659-668, London, UK, Juillet 2001.

- [26] Mounir Lallali, Thèse de Doctorat : Modélisation et Test Fonctionnel de 'Orchestration de Services Web, Institut National des Télécommunications, Français, 2009.
- [27] Hajar OMRANA, Thèse de Doctorat : Vers une composition dynamique des Services Web : une approche de décomposabilité offline, université Mohammed v – Agdal Rabat école Mohammedia d'ingénieurs, 2014.
- [28] Meftah Mohammed Charaf Eddine, Thèse de Doctorat : Une approche formelle pour les applications Web 2.0, Université de Biskra, 2016.
- [29] Nerea Arenaza, Projet de Master : Composition semi-automatique de Services Web, école polytechnique fédérale de Lausanne. Février 2006.
- [30] BOUROUIS Meriem et DJERRIRI Sabah, Projet de Master : La sélection de services Web À base de l'algorithme d'abeilles, UNIVERSITE ABOU BAKR BELKAÏD – TLEMCEN, 2012-2013.
- [31] Shuping Ran ; "A model for web services discovery with QoS"; ACM SIGecom Exchanges ; Volume 4 , Issue 1 Spring, ACM Press, 2003.
- [32] BOUROUIS Meriem , DJERRIRI Sabah, Projet de Master : Selection de services web À base de l'algorithme d'abeilles, UNIVERSITE ABOU BAKR BELKAÏD – TLEMCEN, 2012-2013.
- [33] BENATALLAH, B., SHENG, Q. Z., NGU, A. H. H., AND DUMAS, M. Declarative composition and peer-to-peer provisioning of dynamic web services. In Proceedings of the International Conference on Data Engineering. IEEE, Los Alamitos, CA, 297–308, 2002.
- [34] LI, F., YANG, F., SHUANG, K., AND SU, S. Q-peer : A decentralized QoS registry architecture for web services. In Proceedings of the International Conference on Services Computing. 145–156, 2007.
- [35] M. Alrifai and T. Risse. Combining global optimization with local selection for efficient qos-aware service composition. In WWW April 20–24, 2009, Madrid, Spain. ACM 978-1-60558-487-4/09/04., pages 881–890, 2009.
- [36] Kennedy J. , Eberhart R.C. , Particle Swarm Optimization, Proceedings of the IEEE International Conference On Neural Networks, pages 1942-1948, IEEE Press, 1995.

- [37] Clerc M. , TRIBES - Un exemple d'optimisation par essaim particulaire sans paramètres de contrôle, Conférence OEP'03, Paris, France,le 2 Octobre 2003.
- [38] Alrifai, M., Risse, T., and Nejdil, W. 2012. A hybrid approach for efficient Web service composition with endto- end QoS constraints. *ACM Trans. Web* 6, 2, Article, 31 pages. DOI = 10.1145/2180861.2180864,le 7 Mai 2012.
- [39] NEMHAUSER, G. L. AND WOLSEY, L. A. *Integer and Combinatorial Optimization*. Wiley, New York,1988.
- [40] [Bouguettaya et Qi, 2010] : Qi Yu, A Bouguettaya. *Foundations for Efficient Web Service Selection* Springer Science+Business Media, 2010.
- [41] T. Risse Selecting Skyline Services for QoS-based Web Service Composition In *Proceedings of the WWW 2010*, Raleigh, North Carolina, USA,le 26-30 Avril 2010.
- [42] Coello C.C.A., Van Veldhuizen D.A, Lamont G.B. *Evolutionary Algorithms for Solving Multi-objective Problems*. Kluwer Academic Publishers, New York,2002.
- [43] Hernán Darío ROJAS, *Projet de Master : ORCHESTRATION A HAUT NIVEAU ET BPEL*, Université Joseph Fourier, Grenoble (France), 2006.
- [44] Maria Zemzami, *Application d'un modèle parallèle de la méthode PSO au problème de transport d'électricité* , LITIS, Université de Normandie, INSA, Rouen,2016.
- [45] J. Kennedy and R. C. Eberhart. "Particle Swarm Optimization". In : *Proceedings of the IEEE International Conference on Neural Networks IV*, pp. 1942–1948, Perth, Australia, November 1995.
- [46] COOREN Y., *Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique. Thèse de Doctorat*, Université de Paris 12 Val de Marne, France. 2008.
- [47] BESTAOUI Abdallah Anes, *Gestion de spectre dans un réseau de radio cognitive en utilisant l'algorithme d'optimisation par essaim de particules*, Université Abou Bakr Belkaid– Tlemcen,2015.
- [48] Jeremy Fix et Matthieu Geist, *Optimisation de contrôleurs par essaim particulaire*, Supélec, Équipe IMS 2 rue edouard Belin, 57070 Metz (France),2012.

- [49] GHERBOUDJ A., Méthodes de résolution de problèmes difficiles académiques. Thèse de Doctorat, Université de Constantine 2, Algérie, 2013.
- [50] SHI Y., EBERHART R.C., Parameter selection in particle swarm optimization. In : Proceedings of the Seventh Annual Conference on Evolutionary Programming, pp. 591-600. 1998.
- [51] SHI Y., EBERHART R.C., Empirical study of particle swarm optimization. In : Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp.1945-1950. 1999.
- [52] Gabriel Cormier, Systemes Intelligents, Université de Moncton-Moncton, NB, le 27 Février 2011.
- [53] Deb, K. An introduction to genetic algorithms. *Sadhana*, 24, pp.293–315, 1999.
- [54] BENSMAÏNE Abderrahman, Algorithmes évolutionnaires et méthodes approchées multicritères pour la génération des processus de fabrication dans un environnement reconfigurable, l'université de Lorraine, le 27 Novembre 2013.
- [55] LAMAMRA KHEIREDDINE, Optimisation multi-objectifs par les algorithmes génétiques et application à la commande des systèmes, Université Mentouri, Constantine, le 01 Mars 2012.
- [56] Tahri fatima zohra, Louazani Imane Bouchra, Les algorithmes génétiques pour la sélection des services web, Université Dr Moulay Tahar Saida, 2020.
- [57] Andreas Vogel, Brigitte Kerhervé, Gregor von Bochmann, and Jan Gecsei. Distributed multimedia and qos : A survey. *IEEE MultiMedia*, 2(2), le 19 Octobre 1995.
- [58] Melle BOUROUIS Meriem , Melle DJERRIRI Sabah, Projet de Master : Selection de services web a base de l'algorithme d'abeilles, UNIVERSITE ABOU BAKR BELKAÏD – TLEMCEN, 2012-2013.
- [59] Qi Yu, A Bouguettaya. Foundations for Efficient WebService Selection Springer Science+Business Media, 2010.
- [60] Othmane bouguettaya : Foundation dor Efficient web Service Selection, 2009.