

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab Blida



Faculté des sciences

Département informatique

Mémoire de fin d'études

En vue d'obtenir le diplôme de master

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie des logiciels

THEME

Vers un système auto-adaptatif pour le E-learning

Présenté par :

Benkhedda Acil

Arbouche Nabil

Soutenu le : 24 Novembre 2020

Devant le jury composé de :

Président : M^{me}. TOUBALINE Nesrine

Examineur : M^{me}. LAHIANI Nesrine

Promotrice : M^{me}. GUESSOUM Dalila

Année universitaire : 2019/2020

RESUME

L'auto adaptation est devenue une nécessité dans les systèmes modernes, qui permettent au système de s'adapter à chaque utilisateur en fonction d'une ou plusieurs critères.

Dans ce projet, nous nous sommes intéressés sur l'application de l'auto adaptation dans le domaine de l'apprentissage en ligne (E-learning), en raison de son importance et son avantage surtout après l'émergence de COVID-19. Nous avons proposé une approche basée sur les méta heuristiques (algorithme génétique) qui permet d'adapter le comportement de la plateforme E-learning en variant les fonctionnalités de cette dernière (cours, design, notification, etc.) selon la vitesse de connexion de l'utilisateur et en tenant compte le temps de réponse ainsi que la consommation de données de connexion de la plateforme. Différents tests ont été effectués sur une plateforme E-learning que nous avons développé afin de tester l'efficacité de l'approche proposée.

Mots clés : Auto adaptation, E-learning, Apprentissage en ligne, Algorithme génétique.

Abstract

Self-adaptation has become a necessity in modern systems, which allow the system to adapt to each user based on one or more criteria.

In this project, we focused on the application of self-adaptation based on user connection speeds in e-learning platforms. Due to their importance and their advantage especially after the emergence of COVID-19. We have proposed a method based on meta heuristics (genetic algorithm) which makes it possible to adapt the behavior of the E-learning platform by varying the functionality of it (course, design, notification, etc.) depending on the user's connection speed and taking into account response time as well as the platform's connection data consumption. Various tests were carried out on an E-learning platform that we developed in order to test the effectiveness of the proposed approach.

Key words: Self-adaptation, E-learning, Adaptation, genetic algorithm.

الملخص

أصبح التكيف الذاتي ضرورة في الأنظمة الحديثة، مما يسمح للنظام بالتكيف مع كل مستخدم بناءً على معيار واحد أو أكثر. في هذا المشروع، ركزنا على تطبيق التكيف الذاتي على سرعات اتصال انترنت المستخدم في منصات التعلم الإلكتروني نظرًا لأهميتها ومزاياها خاصة بعد ظهور فيروس كورونا. لقد اقترحنا طريقة تعتمد على الخوارزمية الجينية والتي تمكن المنصة التعليمية على التأقلم من خلال تغيير جودة خدمات المنصة (الدروس، التصميم، التنبيهات، الخ) اعتمادًا على سرعة اتصال المستخدم مع مراعاة وقت الاستجابة وكذلك استهلاك بيانات الانترنت. تم إجراء العديد من الاختبارات على منصة التعلم الإلكتروني التي طورناها من أجل اختبار فعالية الطريقة المقترحة.

الكلمات المفتاحية: التكيف الذاتي، التعلم الإلكتروني، التعلم عبر الإنترنت، الخوارزمية الجينية.

Remerciement

*Nous remercions Dieu de nous avoir donnée patience et courage
pour accomplir ce travail*

*Nous remercions nos parents pour leurs sacrifices et leur
encouragement durant toute notre scolarité*

*Nous remercions surtout notre promotrice Mme D. Guessoum
pour son aide et soutien durant toute la période de travail,
et nous sommes très reconnaissant pour tout ce qu'il nous a fait*

*Nous tenons aussi à exprimer notre gratitude à nos profs
qui nous ont formés durant notre cycle universitaire,*

*ainsi que tout le staff du département
d'informatique*

*Sans oublier tous nos amis et camarades qui font toujours notre
bonheur*

*Aux membres de jury qui nous ont fait l'honneur d'accepter de
juger ce modeste travail*

*Enfin, nous espérons que ce mémoire servira d'exemple et de
support pour les années à venir*

Table des matières

Introduction générale	7
Problématique	1
Objectif	1
CHAPITRE 1 : Généralités sur l’auto-adaptation des systèmes logiciels	3
1. Introduction.....	3
2. Définition de l’auto adaptation	3
1. Systèmes adaptatifs vs Systèmes adaptables	4
2. L’intérêt des systèmes auto adaptatifs	5
3. Propriétés des systèmes auto-adaptatifs.....	5
4. Types des système auto adaptatif.....	8
6. Modèle conceptuel	10
7. Boucle d'adaptation.....	12
8. Méthodologies existantes.....	13
9. Défis des systèmes auto-adaptatifs	15
10. Conclusion	17
11. CHAPITRE 2 : Algorithme Génétique	18
1. Introduction.....	18
2. Algorithme de résolution	18
3. Algorithmes génétiques	19
4. Conclusion	26
5. CHAPITRE 3 : E-learning	27
1. Introduction.....	27
2. Histoire.....	27
3. Définition	27
4. Technologie utilisée dans le E-learning.....	28
5. Les avantages et les inconvénients	30

	Modalités d'adaptation dans le E-learning	31
	Quelques plateformes E-learning	32
	Quelques plateformes E-learning adaptatives.....	33
6.	Discussion	34
7.	Conclusion	35
8.	CHAPITRE 4 : Conception et modélisation de l'approche	36
9.		
10.	Introduction.....	36
	Cas d'étude.....	36
1.	Description de l'approche	40
2.		
3.	Tableau de variations	43
4.	Codage du chromosome.....	45
5.	La population initiale	47
6.		
7.	Fonction de fitness	48
8.	Les opérateurs génétiques	50
9.	Représentation de l'approche.....	51
10.	Conclusion	53
1.	CHAPITRE 5 : Implémentation de l'application et Test.....	54
2.	Introduction.....	54
3.		
4.	Les outils et langages de programmation utilisés	54
5.	Présentation de notre plateforme	56
	Expérimentations et résultats	67
	Conclusion	75
	Conclusion générale	76
	Bibliographie	77

Liste des figures

Figure 1 Propriétés des systèmes auto-adaptatifs [7].....	6
Figure 2 Modèle conceptuel pour les systèmes auto adaptative [9]	10
Figure 3 Feedback Loop MAP-K [10].....	12
Figure 4 Les algorithmes de résolution [13].	19
Figure 5 Les niveaux d'organisation de l'information dans les Gas [13].....	20
Figure 6 l'organigramme de fonctionnement d'un algorithme génétique [14].....	21
Figure 7 Illustration de la roulette de sélection [8]	24
Figure 8 Représentation schématique du croisement à un point [16].....	25
Figure 9 Représentation schématique d'une mutation dans un chromosome [13]	26
Figure 10 Modélisation des variations	37
Figure 11 Diagramme de cas d'utilisateur.....	38
Figure 12: Diagramme de classe.....	39
Figure 13 Représentation de latence	41
Figure 14 Operations génétique	52
Figure 15 Application web mono page	54
Figure 16 Page d'authentification.....	56
Figure 17 Page d'accueil.....	57
Figure 18 Page d'Explore	58
Figure 19 page de Lesson	59
Figure 20 les avis de cours.....	59
Figure 21 Page de Mes Cours	60
Figure 22 Paramètre général	61
Figure 23 Préférences	61
Figure 24 Navigation service	62
Figure 25 Les variations de design	63
Figure 26 Création de cours.....	64
Figure 27 Paramétrés de simulation.....	65
Figure 28 Simulation de GA	66
Figure 29 Journal de simulation.....	66
Figure 30 Interface des courbes graphiques.....	67
Figure 34 Evolution de fonction de fitness	70
Figure 31 Test de Temps de réponse	71

Figure 32 Test de Consommation de connexion.....	72
Figure 33 Test de l'Utilité.....	73
Figure 35 Temps d'exécution	74

Liste des tables

Tableau 1 : Comparaison entre les plateformes populaire, adaptatif et notre plateforme	34
<i>Tableau 2 Tableau de variations</i>	44
Tableau 3 Codage de service de cours	45
Tableau 4 Codage de service de design	46
Tableau 5 Codage de service de navigation des liens.....	46
Tableau 6 Codage de service de chat	46
Tableau 7 Codage de service de notification	46
Tableau 8 Codage de service de font	46
Tableau 9 Codage de service d'avis de cours.....	46
Tableau 10 Codage de service de statistique	47
Tableau 11 Codage de service de recommandation.....	47
Tableau 12 Codage de service Streaming vidéo	47
Tableau 13 Exemple d'une population.....	47
Tableau 14 les poids des critères.	49
Tableau 15 Paramètres de l'algorithme génétique.....	68
Tableau 16 Test de nombre de population.....	69

Introduction générale

Après les récents événements qui ont vu l'émergence du virus Covid 19, la plupart des universités et écoles ont fermé leurs portes pour appliquer la distanciation sociale, donc soudainement l'exigence d'utilisation des méthodes d'apprentissage à distance comme E-learning est devenue indispensable.

Lorsque on parle d'apprentissage à distance on a deux problèmes essentiels, le premier est de développer une plateforme E-learning, et ce problème a été traité et réglé par plusieurs personnes, le deuxième problème n'est pas lié aux plateformes elles-mêmes, mais plutôt à la vitesse Internet des utilisateurs de ces plateformes, car le problème de la vitesse Internet est l'un des problèmes très connue dans les pays du tiers monde comme l'Algérie par exemple.

Problématique

1. Les plateformes E-learning contient plusieurs services et technologie pour facilite le processus d'apprendre et essayer de remplacer les écoles, mais cela résulte le besoin d'une bonne vitesse de connexion pour bien marché, cependant dans certains pays les utilisateurs n'ont pas tous un bon débit de connexion, c'est pour ça il faut trouver une méthode ou approche qui permet au plateformes E-learning d'adapter aux faibles vitesses de connexion des utilisateurs et marché avec un bon temps de réponse.

D'autre part, les plateformes doivent adapter leurs fonctionnalités aux variations constantes des ressources et aux besoins de l'utilisateur. Idéalement, cette optimisation devrait être gérée de manière autonome par l'application elle-même, qui devrait pouvoir optimiser elle-même son fonctionnement.

- Cependant, pour un service d'adaptation, le temps est critique car, pour être utiles, les
2. plateformes doivent être adaptées sans que le temps supplémentaire employé pour le processus d'adaptation ne soit noté.

Objectif

Il est question dans ce PFE d'implémenter une approche alternative pour la génération automatique de plans d'adaptation au moment de l'exécution. L'approche proposée devra générer des plans d'adaptation optimales en ce qui concerne différents critères, tels que le temps de réponse et la fonctionnalité et la consommation des données de connexion. Ceci devra être réalisé à l'aide de métaheuristiques (Génétiq ue algorithm e) qui tentent de trouver des configurations presque optimales à l'exécution. L'approche proposée devra être validée par une

plateforme E-learning pour montrer que l'approche peut être efficace et adaptée aux faibles connexions.

La suite du mémoire est organisée en cinq chapitres :

Le premier chapitre : *Etat de l'art sur l'auto adaptation* : nous présentons dans ce chapitre les éléments et les concepts de base des systèmes auto adaptatifs.

Le deuxième chapitre : *Etat de l'art sur les Algorithmes Génétiques* : nous focaliserons sur la notion des métaheuristiques ainsi que leurs caractéristiques et leurs classifications. Ensuite, nous présentons la métaheuristique « algorithme génétique » utilisée au sein de notre projet.

Le troisième chapitre : *Etat de l'art sur les plateformes E-learning* : dans ce chapitre on va essayer de comprendre l'importance des plateformes E-learning et les Technologies populaires utilisées dans ce domaine.

Le quatrième chapitre : *Conception et modélisation de l'approche* : dans ce chapitre nous présentons la modélisation de notre plateforme E-learning ainsi que la conception de notre solution basée sur l'algorithme génétique.

Le cinquième chapitre : *Implémentation de l'application et Test* : Nous présentons les éléments d'implémentation et les résultats des tests obtenus de notre approche.

Nous terminons la mémoire par une conclusion générale.

CHAPITRE 1 : Généralités sur l'auto-adaptation des systèmes logiciels

Introduction

- Au cours de la dernière décennie, l'importance de l'auto-adaptabilité des systèmes logiciels a été de plus en plus reconnue dans ses divers domaines d'application et technologies. En effet,
1. différentes interprétations et concepts ont été produits par la communauté des chercheurs, le sujet faisant toujours l'objet d'intenses recherches et développements. L'objectif central de ce chapitre est présenté les concepts généraux sur les systèmes auto adaptatifs.

Définition de l'auto adaptation

2. L'auto adaptation a beaucoup de définitions, nous énumérons que quelques-uns de ces définitions qui sont des plus communément acceptées.

Définition 1. L'auto-adaptation est la capacité du système à ajuster son comportement en fonction de l'environnement. Le « auto- » préfixe indique que les systèmes décident de manière autonome (c'est-à-dire avec peu ou pas d'interférence) comment les adapter ou les organiser eux-mêmes afin qu'ils puissent s'adapter aux changements dans leurs contextes et environnements. Bien que certains systèmes auto-adaptatifs puissent fonctionner sans aucune intervention humaine, accompagnement sous forme des objectifs de niveau supérieur (par exemple, par le biais de politiques) sont utiles et réalisables dans de nombreux systèmes [1].

Définition 2. Dans le contexte des systèmes multi-modèles, l'adaptation est une procédure ou une méthode pour passer d'un modèle à l'autre. L'adaptivité peut donc être définie comme la capacité d'un système à atteindre ses objectifs dans un environnement changeant, en exécutant et basculer entre les modèles. Cette capacité contraste avec la con- utilisation intentionnelle du terme adaptation utilisé dans les systèmes mono-modèles, où les paramètres de conception ou les relations sont ajustés pour s'adapter à l'observation de comportement [2].

Définition 3. Un système auto-adaptatif consiste en un système en boucle fermée (c'est-à-dire, modifier pendant l'exécution lui-même en utilisant le feedback due aux changements continus du système), ses exigences et les tendances existantes dans le développement et le déploiement d'un système complexe, réduisant ainsi les efforts humains dans l'interaction avec l'ordinateur. La conception d'un système auto-adaptatif dépend des exigences de l'utilisateur, des propriétés

du système et des caractéristiques environnementales. Un logiciel auto-adaptatif requiert une fiabilité, une robustesse, une adaptabilité et une disponibilité élevées [3].

Définition 4. Un système auto-adaptatif évalue son propre comportement et modifie ses propres performances lorsque l'évaluation indique qu'il n'accomplit pas ce que le logiciel est censé faire, ou lorsqu'une meilleure fonctionnalité ou performance est possible [4].

Systèmes adaptatifs vs Systèmes adaptables

Avant d'apprendre la différence entre ces deux systèmes on doit d'abord connaître la différence

3. entre la personnalisation et la customisation, La personnalisation est souvent rencontrée dans la littérature et se réfère généralement à la fourniture d'informations spécifiquement adaptées aux besoins d'un individu ou d'un groupe d'individus [5]. En d'autres termes, il y a personnalisation dès qu'un système est capable de modifier le contenu ou la forme des informations fournies en fonction de l'utilisateur.

La manière de personnaliser l'information :

- **Spécifiée explicitement par l'utilisateur** : cette façon de personnaliser l'information, désignée sous le terme de « **customisation** » est proposée par la plupart des systèmes interactifs, à des degrés divers d'importance. Généralement ces réglages sont proposés à l'utilisateur via une boîte de dialogue accessible via un menu appelé « préférences » ou « options » [5].
- **Inférée automatiquement par le système** : dans ce cas-là, le système observe l'utilisateur et/ou lui pose un certain nombre de questions dans le but de déterminer la façon d'effectuer la personnalisation. Ce principe est généralement désigné dans la littérature par « personnalisation automatique » [5].

La différence entre la customisation et la personnalisation se situe donc au niveau de la participation de l'utilisateur, dans la customisation l'adaptation est spécifiée explicitement par l'utilisateur mais dans la personnalisation l'adaptation inférée automatiquement par le système.

On peut dire que la distinction entre **Systèmes adaptatifs** et **Systèmes adaptables** découle directement de l'opposition entre personnalisation et customisation. En d'autres termes, un **système adaptable** ne propose que de la customisation alors qu'un **système adaptatif** offre avant tout de la personnalisation automatique.

Donc nous avons intéressé à la personnalisation (systèmes adaptatifs) dans notre projet, pas sur la customisation (systèmes adaptables), car notre plateforme doit observer l'utilisateur et

effectuer une personnalisation ou on peut dire adaptation automatique sans l'intervention d'utilisateur.

L'intérêt des systèmes auto adaptatifs

Les systèmes d'auto-adaptation permettent de suivre les changements fréquents qui se produisent dans leur environnement afin d'assurer une qualité de service optimal pour 4. l'utilisateur final. Ces systèmes sont gérés de manière autonome. L'objectif inhérent aux systèmes adaptatifs est de maintenir la qualité du service rendu quel que soit l'état de l'environnement. Il s'agit de stabiliser la qualité de service [6].

Trois capacités essentielles caractérisent un système adaptatif : l'observation, la décision et l'intro-action : [6]

- **Observation** : Un système adaptatif est un système qui observe son environnement et en détecte les changements. D'un point de vue technique, cette capacité se traduit par l'existence de sondes (logicielles ou matérielles) permettant de mesurer les propriétés pertinentes de l'environnement [6].
- **Décision** : En fonction de l'état de l'environnement, mais également fonction de sa configuration actuelle (capacité d'introspection), un système adaptatif décide par lui-même de la nouvelle configuration à adopter pour fonctionner de manière optimale [6].
- **Intro-action** : Pour modifier sa propre configuration, le système adaptatif manipule et modifie les éléments qui le constituent. Ces « intro-actions » sont des actions de 5. l'intérieur sur l'intérieur. Il s'agit d'une capacité d'introspection active [6].

Propriétés des systèmes auto-adaptatifs

Les propriétés qu'un système devrait avoir pour constituer l'auto adaptation sont décrites dans la figure 1. Ces propriétés peuvent être résumées en quatre objectifs : auto-configurant, auto-réparateur, auto-optimisant et auto-protecteur. Un système auto adaptatif peut s'auto configurer au moment de l'exécution pour répondre à des environnements d'exploitation changeants, s'auto-régler pour optimiser ses performances, s'auto réparer lorsqu'il rencontre des obstacles inattendus pendant son fonctionnement et - d'un intérêt actuel particulier - se protéger des attaques malveillantes. Les équipes de recherche et développement se concentrent sur le développement de théories, de méthodes, d'outils et de technologies pour la construction de systèmes d'auto-configuration, d'auto-réparation, d'auto optimisation et d'auto protection [7].

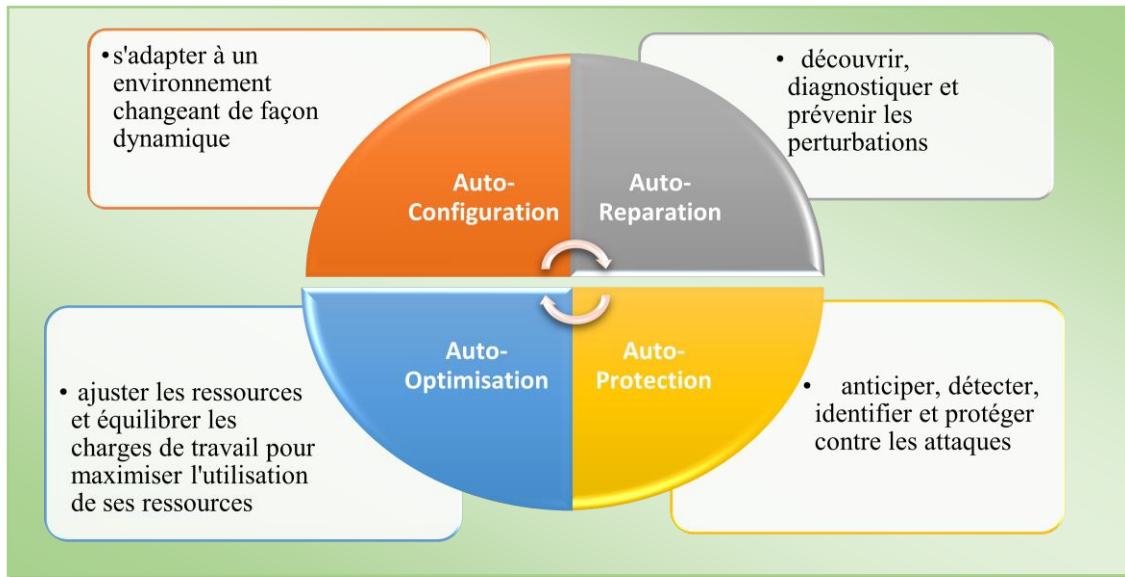


Figure 1 Propriétés des systèmes auto-adaptatifs [7]

Auto configuration

5.1.

Les systèmes auto-configurables offrent une réactivité accrue en s'adaptant à un environnement en évolution dynamique. Un système auto-configurable doit être capable de se configurer et de se reconfigurer dans des conditions variables et imprévisibles. Des degrés variables d'implication de l'utilisateur final devraient être autorisés, de la reconfiguration basée sur l'utilisateur à la reconfiguration automatique basée sur la surveillance et les boucles de rétroaction. Par exemple, l'utilisateur peut avoir la possibilité de reconfigurer le système au moment de l'exécution ; alternativement, les algorithmes adaptatifs pourraient apprendre les meilleures configurations pour atteindre les performances requises ou pour répondre à toute autre exigence fonctionnelle ou non fonctionnelle souhaitée.

La variabilité peut être prise en compte au moment de la conception (par exemple en implémentant des graphiques d'objectifs) ou lors de l'exécution (par exemple, en ajustant les paramètres). Les systèmes doivent être conçus pour offrir une configurabilité au niveau des fonctionnalités avec des capacités telles que la séparation des préoccupations, les niveaux d'indirection, les mécanismes d'intégration (données et contrôle), les couches de script, le plug and Play et les assistants de configuration. Les algorithmes adaptatifs doivent détecter et répondre aux tendances à court et à long terme [7].

Auto optimisation

L'auto optimisation est la capacité de maximiser efficacement l'allocation et l'utilisation des ressources pour satisfaire les exigences des différents utilisateurs. Utilisation des ressources et charge de travail la gestion sont deux aspects importants nécessaires pour une telle caractéristique. Un des modèles couramment utilisés dans l'utilisation des ressources sont la fonction d'utilité. Technologies existantes dans l'aspect de la gestion de la charge de travail, comme le partitionnement logique et le serveur dynamique clustering, devrait être extensible aux systèmes hétérogènes [2].

De cette manière, une seule collection de ressources informatiques sera fournie qui est gérable par un gestionnaire de charge de travail « logique » dans l'ensemble de l'entreprise. Terme, l'auto optimisation peut résoudre la complexité de la gestion des performances du système, dans un à long terme, ses composants chercheront automatiquement et de manière proactive des moyens de régler leur fonctionnement, et se rendre plus efficace en termes de coût [7].

Auto réparation

Les systèmes d'auto réparation fournissent la résilience en découvrant et en empêchant les perturbations ainsi que se remettre de dysfonctionnements. Un tel système pourra récupérer - sans perte de données ou retards notables dans le traitement - dus à des événements courants et extraordinaires qui pourraient certaines de ses pièces ne fonctionnent pas correctement. L'auto-récupération signifie que le système sélectionnera, éventuellement avec l'entrée utilisateur, une configuration alternative à celle qu'il utilise actuellement et passera à cette configuration avec un minimum de perte d'informations ou de retard. L'objectif principal de l'auto-réparation est de maximiser la disponibilité, la survivabilité, la maintenabilité et la fiabilité du système [7].

Auto protection

Les systèmes d'auto protection sécurisent les informations et les ressources en anticipant, détectant et protection contre les attaques. Un tel système sera capable de se protéger en détectant et contrecarrer les menaces grâce à l'utilisation de la reconnaissance de formes et d'autres techniques [7].

Cette capacité signifie que la conception du système comprendra une analyse du vulnérabilités et l'inclusion de mécanismes de protection qui pourraient être utilisés lorsqu'une menace est détectée. La conception doit prévoir des capacités pour reconnaître et gérer différents types de menaces dans divers contextes plus facilement, réduisant ainsi la charge des administrateurs.

Outre les caractéristiques susmentionnées, deux sous-caractéristiques supplémentaires peuvent être énumérées pour un système auto adaptatif, à savoir :

1. **Réflexivité** : Un système auto adaptatif doit avoir une connaissance détaillée de ses composants, état actuel, capacités, limites, frontières, interdépendances avec d'autres systèmes, et les ressources disponibles. De plus, le système doit être conscient de ses possibles configurations et comment elles affectent des exigences non fonctionnelles particulières [7].
2. **Adaptation** : au cœur du problème de complexité abordé par le L'initiative informatique est le problème de l'évaluation des compromis complexes pour rendre les décisions. La plupart des caractéristiques énumérées ci-dessus sont fondées sur la capacité d'un système auto adaptatif pour surveiller ses performances et son environnement et répondre change en passant à un comportement différent. Au cœur de cette capacité se trouve une boucle de contrôle. Les capteurs observent une activité d'un processus contrôlé, un composant du contrôleur décide ce qui doit être fait, puis le composant contrôleur exécute les opérations via un ensemble d'actionneurs. Les mécanismes adaptatifs à explorer seront inspiré par des travaux sur l'apprentissage automatique, les systèmes multi-agents et la théorie du contrôle [7].

6. **Types des système auto adaptatif**

6.1. Il existe quatre types de systèmes auto-adaptatifs :

Type I

Est la mise en œuvre la plus simple de systèmes intelligents. Il surgit d'une analyse approfondie du domaine pour analyser tous les changements possibles que le système observera et réagira. L'activité de conception comprend l'étude de toutes les réactions possibles que le système déclenchera. La conception conduit à la définition d'un modèle

6.2. comportemental qui contient des points de décision : un point de décision est analogue à une instruction « if ... then ... else », alors que les conditions dépendent généralement des perceptions de l'environnement d'exécution [8].

Type II

Se compose de systèmes complexes que d'autres stratégies alternatives pour atteindre le même objectif. Différentes stratégies englobent différentes situations opérationnelles et ont un impact différent sur les exigences non fonctionnelles. La sélection de la stratégie

contextuelle à appliquer est une décision d'exécution qui prend en compte des compromis complexes entre l'état de l'environnement et la qualité de service souhaitée. Cette décision doit inclure la prise de conscience des exigences fonctionnelles / non fonctionnelles et de la manière différente dont chaque opération les affecte [8].

Type III

- 6.3. L'adaptation de Type III représente une implémentation avancée d'un système intelligent qui est instruit avec un ensemble de fonctionnalités de base. Le système peut être utilisé pour assembler des comportements ad hoc qui ne figurent dans aucune des stratégies de solution prédéfinies. Ce type de système est particulièrement adapté pour travailler avec des connaissances incertaines sur l'environnement et les exigences [8].

Type IV

- 6.4. L'adaptation de type IV est le niveau supérieur d'un système intelligent capable de s'auto-inspecter, d'apprendre de l'expérience et d'auto-modifier ses spécifications. Ils sont conçus pour permettre les pires cas d'adaptation : lorsque le système ne possède pas les actions / stratégies appropriées à utiliser et qu'il n'est capable de générer personne. Dans ce cas, le système est en mesure de réviser son modèle d'exécution, donc de produire une nouvelle version du logiciel. Dans cette catégorie d'adaptation, il est plus approprié de se référer à l'évolution. En effet, ces systèmes sont inspirés des systèmes biologiques qui possèdent la capacité de faire face à la variance de l'environnement par des changements génétiques [8].

Donc on peut dire que :

1. L'adaptation est de **Type I** Si le système possède un ensemble de stratégies prédéfinies (chaque stratégie est une agrégation d'actions).
2. De plus de **Type I**, si la stratégie est sélectionnée et / ou configurée au moment de l'exécution selon des aspects de qualité, alors l'adaptation est de **Type II**.
3. De plus de **Type II**, si le système est capable d'assembler une nouvelle stratégie au moment de l'exécution, alors l'adaptation est de **type III**.
4. De plus de **type III**, si le système peut modifier ses modèles d'exécution pour générer de nouvelles fonctions alors l'adaptation est **type IV**.

Dans notre projet le type d'adaptation est le **type IV**, qui est inspirés de l'évolution génétique.

Modèle conceptuel

Le modèle conceptuel décrit les éléments abstraits composant un système auto-adaptatif et la relation entre eux. En d'autres termes, il présente la base principes d'un système auto-adaptatif

7. Le modèle conceptuel est composé de trois éléments : le système géré (SG), le moteur d'adaptation (MA) et l'environnement (E). Par conséquent, un système auto-adaptatif peut être vu comme un tuple $SAS = (SG, MA, E)$. Figure 2 Modèle conceptuel pour les systèmes auto adaptative [9] illustre l'anatomie du modèle conceptuel et une description de chaque entité est présentée [9].

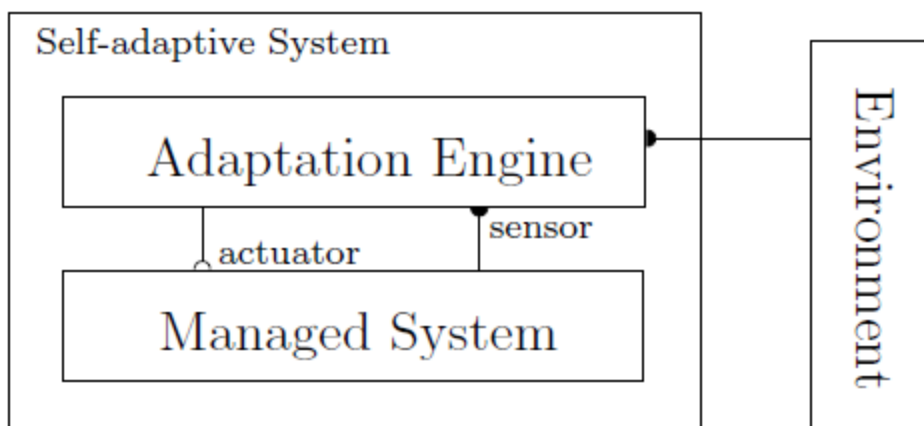


Figure 2 Modèle conceptuel pour les systèmes auto adaptative [9]

7.1.

Système géré

Comprend le code d'application qui réalise la fonctionnalité du système. Dans le cas des systèmes adaptatifs collaboratifs, le système géré peut être considéré comme une série de ressources telles que des robots, des véhicules, etc. Pour prendre en charge l'adaptation, le système géré est équipé d'actionneurs.

L'actionneurs permettre l'exécution des demandes d'adaptation sélectionnées par le moteur d'adaptation. Par exemple, étant donné que plusieurs robots collaborent pour transporter un élément du point A au point B, le système géré est responsable de la navigation des robots et du transfert des éléments. Les actionneurs peuvent restreindre la participation de cinq robots au transfert de l'élément en fonction de son poids. Différents termes sont utilisés dans la littérature pour désigner le concept de système géré. Par exemple, il est également appelé élément géré, couche système, logiciel adaptable [9].

Moteur d'adaptation

Supervise et administre le système géré. Il contient la logique d'adaptation nécessaire pour atteindre les exigences ou les objectifs du système. Le moteur d'adaptation est équipé de capteurs qui surveillent à la fois le système géré et l'environnement et adapte le précédent si nécessaire. Le moteur d'adaptation analyse les données suivies et construit un plan d'adaptation. Par exemple, considérons un robot qui adapte sa stratégie de navigation en fonction de la présence d'obstacles (détectés depuis l'environnement) et de son niveau d'énergie (détecté depuis le système géré). Différents termes sont utilisés dans la littérature pour désigner le concept de moteur d'adaptation. Par exemple, il est également appelé gestionnaire autonome, couche d'architecture, logique d'adaptation [9].

Environnement

7.3. Fait référence au monde extérieur avec lequel le système interagit et par lequel il est effectué. Il peut comprendre des entités physiques telles que des obstacles sur le chemin d'un robot [9].

La représentation de ce modèle conceptuel dans notre projet peut être comme suit :

- **Système géré** : c'est notre plateforme E-learning.
- **Moteur d'adaptation** : c'est notre approche qui est basée sur les algorithmes génétiques.
- **Environnement** : représente la vitesse de connexion d'utilisateur.

Boucle d'adaptation

8.

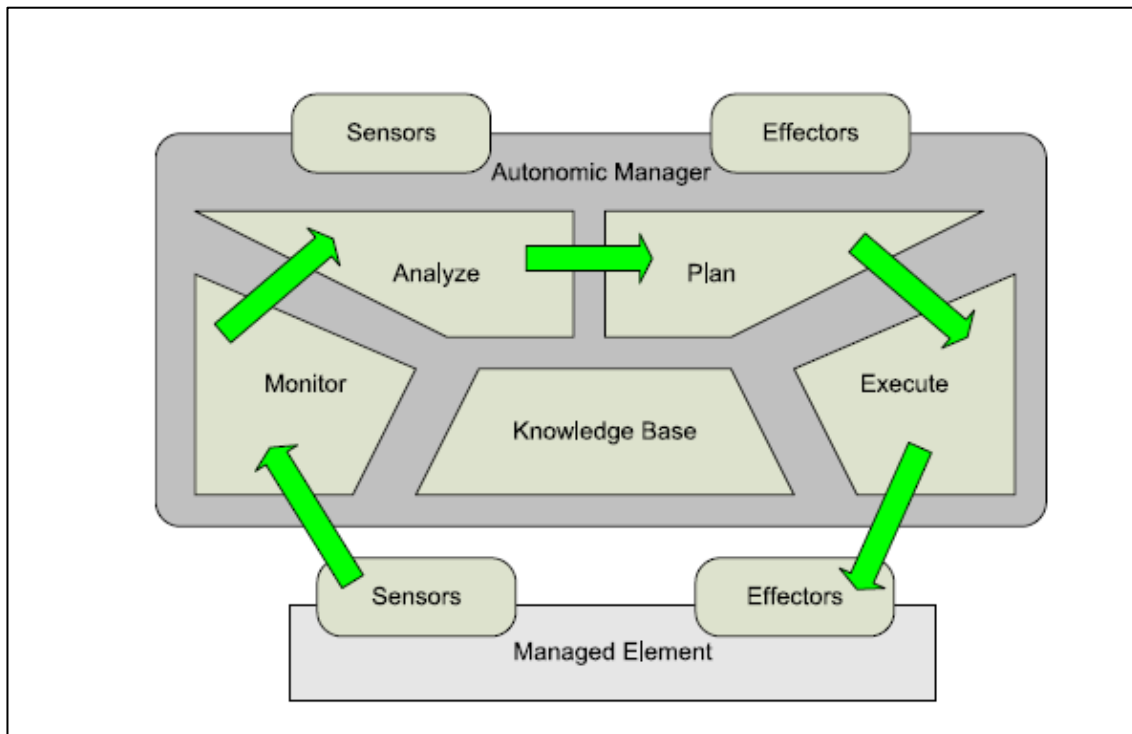


Figure 3 Feedback Loop MAP-K [10]

Les systèmes auto-adaptatifs déploient un mécanisme en boucle fermée, également connu sous le nom de boucle d'adaptation. La boucle d'adaptation comprend le processus utilisé par le moteur d'adaptation pour atteindre l'adaptabilité. Il s'inspire de la boucle de contrôle MAPE-K en informatique autonome et englobe quatre étapes, la surveillance, l'analyse, la planification et l'exécution [7].

1. **Surveillance (Monitor)** : la phase de surveillance concerne l'extraction d'informations - propriétés ou états - hors de l'élément géré. Les mécanismes vont de l'instrumentation du code source à l'interception de communication non intrusive.
2. **Analyser** : il s'agit de déterminer si quelque chose a mal tourné dans le système, généralement parce qu'une propriété système présente une valeur en dehors des limites attendues, ou à une tendance à la dégradation.
3. **Plan** : il s'agit de déterminer un plan d'action pour adapter l'élément géré une fois qu'un problème est détecté.
4. **Exécuter** : concerne l'exécution d'un plan d'action choisi et la réalisation des changements dans le système.

L'efficacité du MAPE-K vient de sa structure intuitive dans la gestion des différentes fonctions requises pour une boucle de rétroaction. La Figure 3 illustre le processus de la boucle d'adaptation. La première étape consiste à surveiller et à collecter les données de l'environnement et du système géré via des capteurs. Les données collectées sont traitées et les connaissances sont mises à jour. Ensuite, les connaissances à jour sont analysées pour déterminer si une adaptation est nécessaire pour atteindre les exigences ou les objectifs du système. Ensuite, si l'adaptation est obligatoire, un Plan est construit comprenant une ou plusieurs actions d'adaptation. Enfin, le plan est exécuté par le système géré à l'aide d'actionneurs. Pour résumer, un système auto-adaptatif nécessite des capteurs pour surveiller l'environnement et le système géré. Cela nécessite également la présence d'un moteur d'adaptation qui a la capacité de surveiller et de collecter des données, d'analyser les informations collectées et d'élaborer un plan d'adaptation optimal. Enfin, les systèmes auto-adaptatifs exigent la présence d'actionneurs qui aideront le système géré dans l'exécution du plan d'adaptation [10].

Dans notre projet la boucle d'adaptation est appliquée comme suit :

La première étape consiste à surveiller la vitesse de connexion d'utilisateur, Ensuite, la vitesse de connexion et le temps de réponse de plateforme sont analysées pour déterminer si une adaptation est nécessaire ou non. Ensuite, si l'adaptation est obligatoire, un Plan est construit pour diminuer le temps de réponse. Enfin, le plan est exécuté par le système géré.

9.

Méthodologies existantes

Des efforts considérables ont été déployés par des ingénieurs et des chercheurs de différentes disciplines pour réaliser des systèmes auto-adaptatifs. Cette section discute de diverses approches qui ont été développées au fil du temps. Chaque approche est inspirée par une

9.1. discipline spécifique et, à ce titre, met en évidence les aspects complémentaires de la réalisation de systèmes auto-adaptatifs.

Technique de contrôle

L'ingénierie de contrôle est une discipline qui se concentre sur la conception de systèmes qui se comportent comme prévu à l'aide de contrôleurs de système. Traditionnellement, l'ingénierie de contrôle s'est intéressée aux systèmes qui sont régis par les lois de la physique, comme les installations de contrôle physique. Ces dernières années, l'application de la théorie du contrôle au contexte informatique a été étudiée dans divers travaux. En outre, les similitudes entre les installations de contrôle physique et les systèmes auto-adaptatifs sont évidentes. Les plantes

physiques réagissent constamment à leur environnement pour atteindre un certain objectif, tout comme les systèmes auto-adaptatifs. Dans cette approche, il existe deux concepts fondamentaux, à savoir le système cible et le contrôleur. Le contrôleur met en œuvre une stratégie de commande qui dicte le signal de commande correct qui adapte le système cible afin de maintenir la sortie du système cible suffisamment proche de l'objectif souhaité. Le signal de commande est généralement basé sur la différence entre la sortie du système cible précédent et l'objectif du système. Le système cible est un modèle analytique basé sur des relations mathématiques qui relient l'effet du signal de commande sur le comportement du système.

Un système basé sur le contrôle repose sur une boucle de contrôle pour incorporer la sortie du système cible et les perturbations extérieures. Une technique de premier plan pour organiser une boucle de contrôle dans des systèmes auto-adaptatifs est la boucle MAPE-K qui a été appelée boucle d'adaptation. En fait, de nombreux travaux soulignent l'importance et l'application de la boucle MAPE-K dans les systèmes auto-adaptatifs basés sur le contrôle [9].

Programmation logicielle

9.2.

Dans cette approche, des langages de programmation à usage général sont utilisés pour réaliser des systèmes auto-adaptatifs. L'une des principales techniques est connue sous le nom de réflexion qui a été introduite dans la communauté des langages de programmation dans le but d'augmenter la flexibilité de programmation et de permettre le développement de systèmes logiciels fermés, qui ne nécessitent pas d'interférences externes. Un système logiciel réfléchissant est un système qui a la capacité d'examiner et de modifier à la fois son comportement et sa structure. Un langage de programmation prenant en charge la réflexion fournit un certain nombre de fonctionnalités disponibles au moment de l'exécution qui facilitent la réflexion, telles que la création de nouveaux types de classes au moment de l'exécution et l'instanciation d'objets de classes qui n'étaient pas définis au moment de la compilation. De

9.3.

nombreux langages de programmation à usage général possèdent déjà des capacités de réflexion telles que JAVA et C#. La capacité d'un logiciel à s'adapter est une caractéristique intrinsèque des systèmes auto-adaptatifs [9]

Intelligence artificielle

L'IA offre aux systèmes la possibilité d'apprendre, de s'améliorer et de prendre des décisions afin d'exécuter des tâches complexes. Le domaine de l'IA est vaste et va du traitement du langage naturel, des systèmes multi-agents, de l'apprentissage automatique, de la théorie de l'utilité, entre autres. Les systèmes auto-adaptatifs ont des points communs avec l'intelligence

artificielle, à savoir la gestion de scénarios inattendus. En faisant référence à la boucle d'adaptation, qui est une exigence de base des systèmes auto-adaptatifs, l'intelligence artificielle peut être trouvée utile dans deux éléments principaux de la boucle d'adaptation, à savoir l'élément d'analyse et de plan. Les techniques d'IA peuvent jouer un rôle central dans l'auto-adaptation en traitant de grandes quantités de données et en effectuant des analyses et des prises de décision. Les techniques d'apprentissage de l'intelligence artificielle peuvent être utilisées pour mieux analyser et identifier les modèles dans les données détectées de l'environnement. En outre, il peut être utilisé pour prendre de meilleures décisions sur le plan d'adaptation à exécuter en tirant les leçons des expériences précédentes [9].

On trouve plusieurs algorithmes d'apprentissage en intelligence artificielle tels que l'apprentissage par renforcement et l'algorithme génétique [11] peuvent être incorporés dans la phase de planification par le moteur d'adaptation d'un système auto-adaptatif.

Donc on peut dire que :

- **Les techniques de contrôle** : sont utiliser lorsque on a un Environment qui contient plusieurs variables qui déterminent l'adaptation, dans notre projet on a une seule variable qui est la vitesse de connexion donc on n'est pas intéressé à cette méthode.
- **La programmation logicielle** : est utiliser lorsque on a une simple adaptation c'est-à-dire un ensemble de stratégies prédéfinies (chaque stratégie est une agrégation d'actions), mais dans notre projet on a une adaptation qui doit faire une optimisation au temps de réponse par rapport au plusieurs critères donc cette méthode ne nous aide pas.
- **L'intelligence artificielle** : est utiliser lorsque on a un problème complexe qui contient plusieurs critères qui définit l'adaptation, donc l'intelligence artificiel c'est le meilleur choix pour nous.

10. Dans notre projet la méthodologie utilisée est l'intelligence artificielle avec l'algorithme génétique.

Défis des systèmes auto-adaptatifs

Les systèmes auto-adaptatifs posent de nouveaux défis au développement et à la conception de systèmes logiciels. Cette section vise à identifier les différents défis auxquels sont confrontés les ingénieurs logiciels dans la réalisation de systèmes auto-adaptatifs.

Défis de confiance

L'un des principaux défis rencontrés après la sortie d'un système auto-adaptatif dans l'industrie est le manque de confiance des utilisateurs / administrateurs face à de tels systèmes. Cela est principalement dû à trois problèmes. Premièrement, l'auto-dépendance du système le laisse

10.1. introuvable et, par conséquent, l'utilisateur est abandonné en ce qui concerne les actions que le système choisit / exécute. Une façon de résoudre ce problème est de rapporter les activités et les décisions prises par le système auto-adaptatif aux administrateurs [9].

Défis d'analyse

Compte tenu des informations surveillées, le principal objectif de l'analyse est de déterminer

10.2. quand le système est en mauvais état. Un mauvais état fait généralement référence à un comportement indésirable du système qui nécessite une adaptation. Dans quelle mesure il peut détecter un mauvais état et sera-t-il détecté assez tôt pour prendre les mesures appropriées ? Ce sont là de nombreuses enquêtes qui devraient être traitées. La tâche d'analyse à ce jour est considérée comme un défi majeur. En fait, sa complexité a conduit les chercheurs à s'appuyer sur des techniques d'analyse ad hoc et basées sur des règles. Des approches prometteuses sont l'utilisation de l'intelligence artificielle et des techniques d'exploration de données pour adopter l'analyse en ligne [12].

10.3.

Défis de planification

La raboteuse prend une capture d'écran de l'état actuel du système avec les objectifs du système pour décider d'un plan d'adaptation qui satisfait les contraintes et les objectifs du système. Le plan d'adaptation est une séquence d'actions qui doivent faire passer le système d'un état indésirable à un état normal. Malheureusement, cette tâche est difficile en termes de calcul et en tant que telle, la plupart des chercheurs s'appuient sur une planification hors ligne. Dans la planification hors ligne, un ensemble de plans est créé au moment de la conception et peut être affiché soit par construction, soit par un processus de vérification pour satisfaire les contraintes du système. Cependant, le véritable défi de la recherche réside dans la planification en ligne, où de nouveaux plans sont synthétisés à la volée à mesure que les objectifs du système changent. D'autres défis incluent la planification de multiples objectifs et la résolution de conflits, la prise en compte des informations système incomplètes dans les systèmes décentralisés et la garantie que le comportement transitoire planifié est sûr [12].

Défis d'exécution

À ce stade, le système géré exécute le plan d'adaptation à l'aide d'actionneurs. Les points à considérer à ce stade sont les suivants : comment gérer l'échec de l'achèvement de l'exécution du plan d'adaptation et l'interférence entre l'exécution de plusieurs plans d'adaptation. De plus, 10.4. une étape importante consiste à valider que l'exécution du plan d'adaptation est en fait correcte et aboutit au comportement souhaité. La plupart des approches existantes reposent sur des exemples limités pour montrer la validité de leur approche, cependant la vérification est une étape essentielle. Le comportement adaptatif de ces systèmes dimensionne la nécessité d'une vérification statique et renforce la nécessité d'une vérification d'exécution. S'appuyer sur la vérification au moment de l'exécution pour un système adaptatif est complexe en raison des nombreuses alternatives et chemins d'exécution qui sont inhérents à la nature des systèmes auto-adaptatifs. Une combinaison de vérification hors ligne et en ligne semble être une solution possible, mais difficile [12].

Conclusion

11. Ce chapitre a présenté, d'un point de vue informatique, les concepts, les méthodes et les défis dans le domaine des systèmes logiciels auto-adaptatifs, de manière simple et systématique. Une compréhension de ce sujet nouveau et stimulant aidera certainement à bien construire notre approche.

Depuis les choses importants qu'on a appris dans ce chapitre c'est qu'on va utiliser comme méthodologie l'intelligent artificiel et plus spécifiquement les algorithmes génétiques pour construire notre approche, et ça ce qu'on va discuter dans le chapitre suivant.

CHAPITRE 2 : Algorithme Génétique

Introduction

Durant ces dernières années, plusieurs métaheuristiques ont prouvé leur efficacité pour la résolution de divers problèmes d'optimisation dans un temps limité, Surtout dans le cas des

1. problèmes difficile où les algorithmes exacts ne possèdent pas un temps polynomial.

Ce chapitre est consacré à l'étude des métaheuristiques notamment les algorithmes génétiques que nous avons utilisé dans notre approche et on va présenterons leur caractéristique et fonctionnement d'une manière détaillée.

Algorithme de résolution

2. En général les problèmes peuvent être résolus par un algorithme exact ou un algorithme approximatif (ou heuristiques), dans les algorithmes exacts les solutions sont optimales et leur optimalité est garantie. Dans le cas des problèmes NP-complets, les algorithmes exacts ne possèdent pas un temps polynomial. Les algorithmes approximatifs (ou heuristiques) génèrent des solutions de grande qualité dans un temps raisonnable, mais il n'y a aucune garantie de trouver une solution optimale globale [13].

Dans **Les méthodes exactes** toutes les solutions de l'espace de recherche sont énumérées implicitement en utilisant des mécanismes qui détectent des échecs (calcul de bornes). Grâce à Ces méthodes on peut trouver des solutions optimales. Mais ces méthodes s'avèrent, malgré les progrès réalisés, plutôt inefficaces à mesure que la taille du problème devient importante [14].

Les méthodes heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution [13].

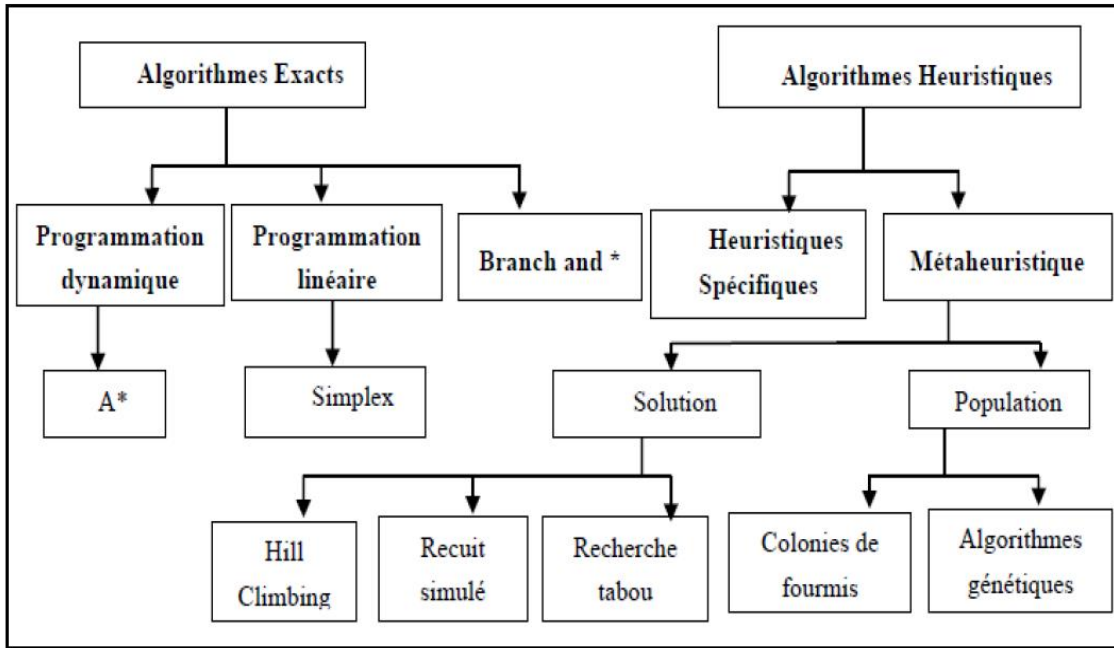


Figure 4 Les algorithmes de résolution [13].

Dans notre projet on s'intéresse aux algorithmes génétiques car elles sont simples à implémenter et plus efficaces.

3. Algorithmes génétiques

Les algorithmes génétiques (AGs) ont été mis au point par John Holland de l'université du Michigan aux États-Unis dans les années 60. Dans ces GAs, ils ne s'agissent pas de trouver une solution analytique exacte, ou une bonne approximation numérique, mais de trouver des solutions satisfaisant au mieux à différents critères, souvent contradictoires. Ils manipulent un ensemble de plusieurs solutions simultanément et cherchent des solutions à des problèmes d'optimisation. La recherche de la solution se fait d'une manière itérative, elle commence par une population initiale et se déroule pendant certain nombre d'itérations pour terminer par le retour de la solution optimale si un certain critère d'arrêt est satisfait. L'aspect itératif de ces algorithmes permet d'explorer l'espace de recherche et améliorer la solution à chaque itération afin d'obtenir la solution optimale [11].

Mais avant, il est nécessaire de définir quelques termes de base rencontrés dans la littérature :

- **Population** : ensemble fini d'individus (de solution).
- **Individu** : solution potentielle du problème.
- **Chromosome** : solution potentielle du problème sous une forme codée (forme de chaîne de caractères).

- **Gène** : partie élémentaire (caractère) non divisible d'un chromosome.
- **Fonction de fitness** : Terme anglo-saxon qui désigne la fonction d'évaluation d'un individu. Cette fonction est liée à la fonction à optimiser et permet de définir le degré de performance d'un individu (donc d'une solution) [14].

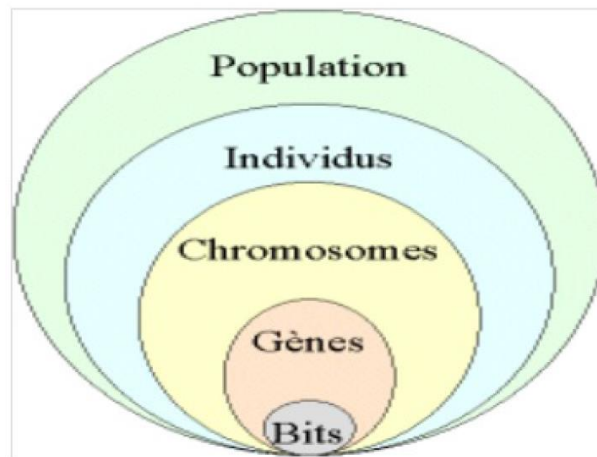


Figure 5 Les niveaux d'organisation de l'information dans les Gas [13]

Principe de base

3.1.

Les algorithmes génétiques sont des approches d'optimisation qui utilisent des techniques dérivées de la science génétique et de l'évolution naturelle : la sélection, la mutation et le croisement. Pour utiliser ces approches, on doit disposer des éléments suivants :

1. Une représentation génétique du problème c'est-à-dire un codage d'un élément de population : une fonction qui permet de modéliser les données du problème réel dans des données utilisables par l'algorithme génétique.
2. Une fonction pour générer la population initiale : la génération de la population initiale est importante puisque cette génération représente le point de départ de l'algorithme et son choix influe sur la rapidité et l'optimalité de la solution finale.
3. Une fonction qui permet d'évaluer l'adaptation d'un chromosome à son environnement, ce qui offre la possibilité de comparer des individus. Cette fonction est construite à partir du critère que l'on désire optimiser. L'application de cette fonction à un élément de la population donne son fitness.
4. Des opérateurs qui permettent d'évoluer d'une population à une autre tout en améliorant la fonction objective. L'opérateur de croisement recompose les gènes d'individus existant dans la population, alors que l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.

- Des paramètres de dimensionnement : taille de la population, nombre total de générations (critère d'arrêt), probabilités d'application des opérateurs de croisement et de mutation, etc. [15].

La figure suivante montre l'organigramme de fonctionnement d'un algorithme génétique.

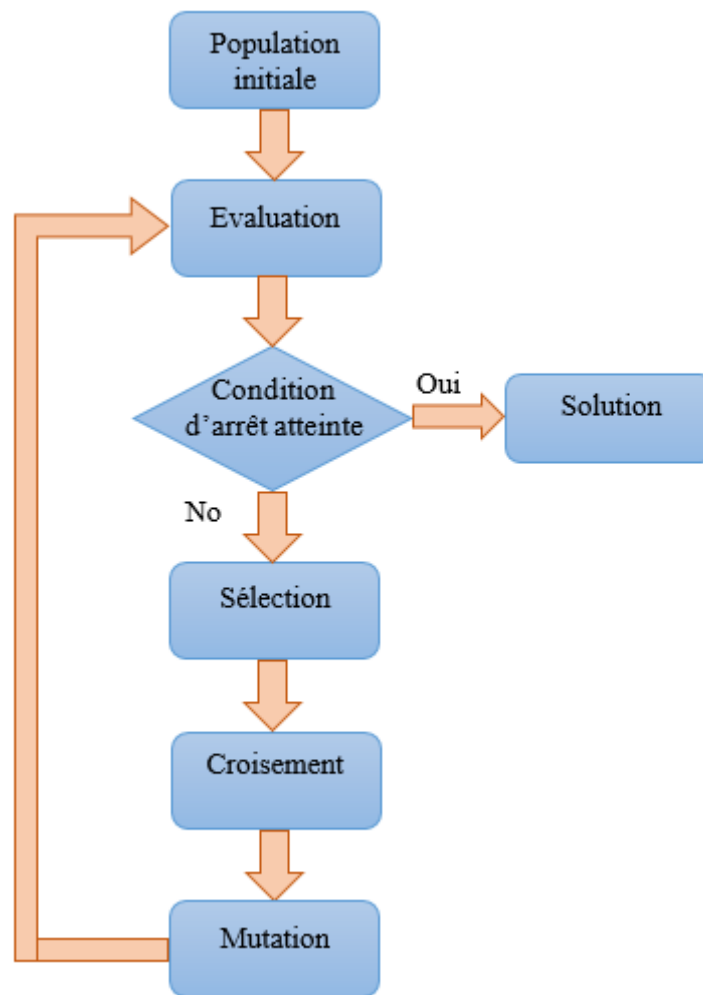


Figure 6 l'organigramme de fonctionnement d'un algorithme génétique [14]

L'algorithme GA commence par une population aléatoire d'individus. Jusqu'à la fin du critère de fin, cet algorithme améliore la population en utilisant les trois opérateurs sélection, croisement, mutation. La meilleure solution dans la dernière population est renvoyée comme la meilleure approximation de l'optimum global pour un problème donné. Le taux de sélection, de croisement et de mutation peut être modifié ou défini pour fixer des nombres pendant l'optimisation.

Représentation des individus (encodage)

La littérature définit deux types de codage : **binaire et réel**.

3.2.1. Le codage binaire :

3.2. Le codage binaire consiste à représenter chaque vecteur des paramètres X par une chaîne de bits dont chaque instance prend la valeur 0 ou 1. Cette chaîne peut être une concaténation des différentes variables du problème. Chaque paramètre est transformé en sa valeur binaire avec un nombre de bits fixé a priori [16].

3.2.2. Codage réel :

La représentation des solutions dans le cadre des AG n'est pas nécessairement réduite à un alphabet de faible cardinalité (0,1), il existe toute une école pour laquelle la représentation la plus efficace est celle qui s'appuie sur des nombres réels. Cette représentation est à la base de l'approche évolutionnaire « Evolution stratégie ». Ce type de codage présente certains avantages par rapport au codage binaire :

- Le codage réel est robuste pour les problèmes considérés comme difficile pour le codage binaire.
- Ce codage nécessite une adaptation des opérateurs de croisement et mutation [16].

3.3. **Operateurs génétiques**

3.3.1. Population initiale

L'algorithme GA commence par une population aléatoire. Cette population peut être générée à partir d'une distribution aléatoire gaussienne pour augmenter la diversité. Cette population comprend de multiples solutions, qui représentent les chromosomes d'individus. Chaque chromosome a un ensemble de variables, qui simule les gènes. L'objectif principal de l'étape d'initialisation est de répartir le plus uniformément possible les solutions autour de l'espace de recherche pour augmenter la diversité de la population et avoir de meilleures chances de trouver des régions prometteuses. Les sections suivantes discutent des étapes pour améliorer les chromosomes dans la première population [13].

3.3.2. Evaluation

L'évaluation consiste à mesurer la performance (ou on peut dire l'importance) de chaque individu de la population. Il utilise pour cela la fonction fitness qui consiste à éliminer les individus moins performants et conserver les individus les plus performants. C'est une fonction réelle positive qui reflète la force de l'individu. Un individu ayant une grande valeur fitness

représente une bonne solution au problème, alors qu'un individu ayant une faible valeur fitness représente une solution médiocre [15].

3.3.3. Sélection

La sélection naturelle est la principale source d'inspiration de ce composant pour l'algorithme GA. Dans la nature, les individus les plus en forme ont plus de chances d'obtenir de la nourriture et de s'accoupler. Cela amène leurs gènes à contribuer davantage à la production de la prochaine génération de la même espèce.

Il existe deux types de sélection :

1. **Sélection pour la reproduction** : On l'appelle tout simplement l'opération de sélection, et elle permet de choisir les individus qui participent à une reproduction (croisement ou mutation). Cette opération choisit généralement les individus les plus forts (meilleurs scores de la fonction de fitness) pour produire les enfants les plus performants [15].

2. **Sélection pour le remplacement** : Elle est simplement appelée opération de remplacement, et elle choisit les individus les plus faibles pour être remplacés par les nouveaux [15].

On trouve plusieurs techniques de sélection :

a) La sélection par roulette [11]

Dans la sélection à la roulette, la probabilité de choisir un individu pour l'élevage de la prochaine génération est proportionnelle à sa valeur de fitness, plus la valeur de fitness est grande, plus il y a de chances pour que cet individu soit choisi. Le choix d'individus peut être décrit comme une sorte de roulette de casino sur laquelle sont placés tous les chromosomes de la population, les places accordées à chacun des chromosomes étant en relation avec sa valeur d'adaptation. Ensuite, la bille est lancée et s'arrête sur un chromosome. Les meilleurs chromosomes peuvent ainsi être tirés plusieurs fois et les plus mauvais ne jamais être sélectionnés.

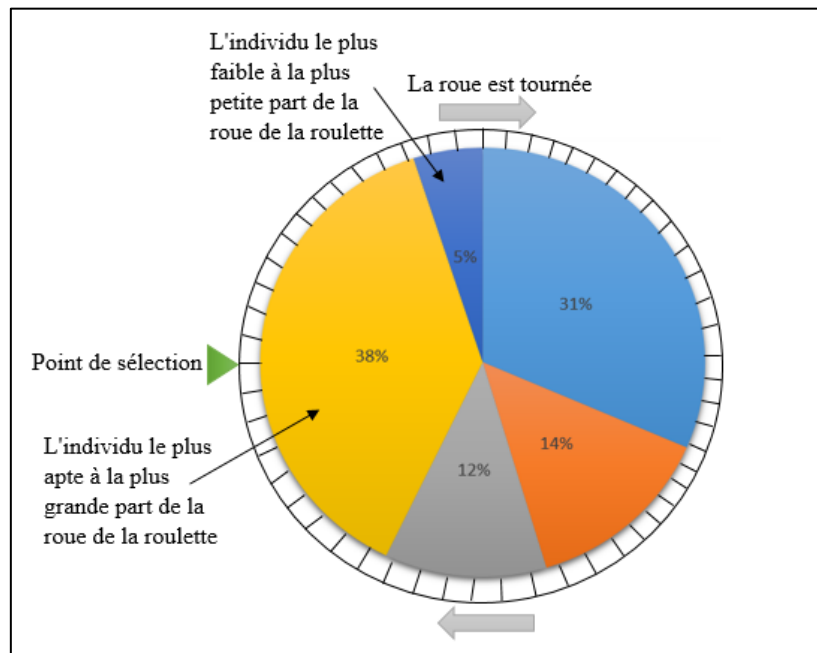


Figure 7 Illustration de la roulette de sélection [8]

b) Sélection par rang [17]

La sélection par rang trie d'abord la population par aptitude. Chaque chromosome se voit associé un rang en fonction de sa position. Le plus mauvais chromosome aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur chromosome qui aura le rang N (pour une population de N chromosomes). La sélection par rang d'un chromosome est la même que par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation. Avec cette méthode de sélection, tous les chromosomes ont une chance d'être sélectionnés. Cependant, elle conduit à une convergence plus lente vers la bonne solution. Ceci est dû au fait que les meilleurs chromosomes ne diffèrent pas énormément des plus mauvais.

c) Sélection par tournoi [16]

La sélection de tournoi est une méthode pour choisir l'individu parmi l'ensemble d'individus. Le gagnant de chaque tournoi est sélectionné pour effectuer un crossover.

d) Sélection d'élitisme [13]

Souvent, pour obtenir de meilleurs paramètres, des stratégies avec reproduction partielle sont utilisées. L'un d'eux est l'**élitisme**, dans lequel une petite partie des meilleurs individus de la dernière génération est transférée (sans aucun changement) à la suivante.

3.3.4. Croisement

Dans les algorithmes génétiques et le calcul évolutif, le croisement, également appelé recombinaison, est un opérateur génétique utilisé pour combiner l'information génétique de deux parents afin de générer un nouveau chromosome. C'est une façon de générer stochastiquement de nouvelles solutions à partir d'une population existante, et analogue au croisement qui se produit lors de la reproduction sexuée en biologie. Des solutions peuvent également être générées en clonant une solution existante, ce qui est analogue à la reproduction asexuée. Les solutions nouvellement générées sont généralement mutées avant d'être ajoutées à la population [14].

Il existe différentes techniques pour l'opérateur de croisement dans la littérature dont deux : un point, un double point

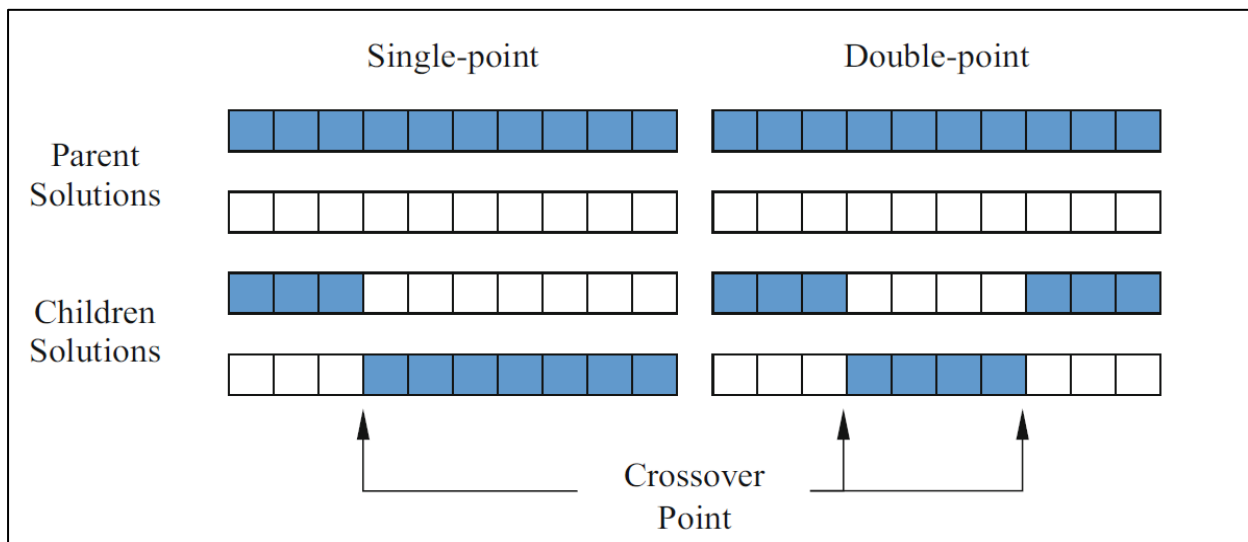


Figure 8 Représentation schématique du croisement à un point [16]

Dans le croisement en un seul point, les chromosomes de deux solutions parentes sont échangés avant et après un seul point. Dans le croisement à double point, cependant, il y a deux points de croisement et les chromosomes entre les points sont échangés uniquement.

3.3.5. Mutation

La mutation est un opérateur génétique utilisé pour maintenir la diversité génétique d'une génération à l'autre donc conserver la diversité de la population et évite en particulier à l'AG de converger vers un optimum local.

L'exemple classique d'un opérateur de mutation implique une probabilité qu'un bit arbitraire dans une séquence génétique soit retourné de son état d'origine, Pour les AGs avec un codage binaire, elle consiste à changer la valeur d'un ou plusieurs bits de l'individu de la population

parent d'une valeur 1 à la valeur 0, ou réciproquement, avec une probabilité de mutation fixée. [11]

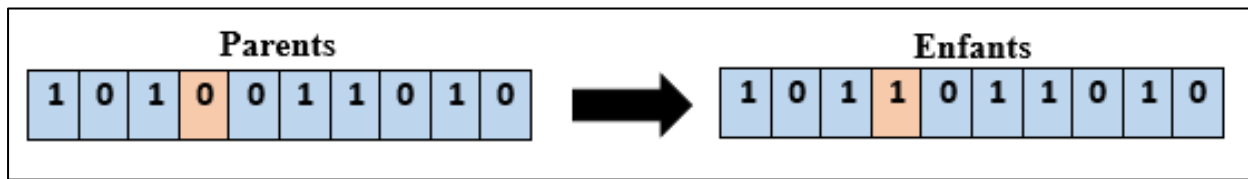


Figure 9 Représentation schématique d'une mutation dans un chromosome [13]

La mutation est un phénomène rare mais permet d'explorer de nouveaux domaines dans l'espace de recherche et aide l'algorithme génétique à évoluer éventuellement vers une solution optimale globale, sans rester pris dans une solution optimale locale [15].

Conclusion

4. Ce chapitre présente L'algorithmes génétiques comme l'un des algorithmes évolutionnaires le plus populaires dans la littérature et plus répandues pour résoudre des problèmes difficiles d'optimisations et de recherche. Donc on a acquis les connaissances de bases sur les métaheuristiques notamment les algorithmes génétiques avec leur opérateurs génétiques comme sélection, crossover, mutation et leur fonction d'évaluation.

On a compris dans ce chapitre le fonctionnement des algorithmes génétiques qu'on va utiliser dans notre approche et qu'on va implémenter dans une plateforme E-learning, mais c'est quoi une plateforme E-learning ? Et pourquoi utiliser une plateforme E-learning et quelle sont leur avantage ? C'est ça ce qu'on va discuter dans le chapitre suivant.

CHAPITRE 3 : E-learning

Introduction

L'apprentissage à distance ou E-learning a toujours été l'un des sujets intéressants pour lesquels un avenir radieux est attendu, mais avec l'émergence du virus Corona dans le monde et la

1. nécessité de parvenir à la séparation, l'apprentissage à distance est devenu une nécessité indispensable.

Dans ce chapitre, on va reconnaître les plateformes E-learning, leurs technologies, leurs avantages et inconvénients et nous terminerons avec des plateformes actuellement célèbre.

Histoire

2. Le terme « E-learning » n'existe que depuis 1999 Cependant, les principes de l'e-learning ont été bien documentés tout au long de l'histoire, et il existe même des preuves qui suggèrent que les premières formes d'e-learning existaient dès le 19ème siècle.

Avec l'introduction de l'ordinateur et d'Internet à la fin du 20e siècle, les outils d'apprentissage en ligne et les méthodes de diffusion se sont développés. Puis, au cours de la décennie suivante, les environnements d'apprentissage virtuels ont commencé à vraiment prospérer, les gens ayant accès à une mine d'informations en ligne et à des possibilités d'apprentissage en ligne.

Dans les années 2000, les entreprises ont commencé à utiliser l'e-learning pour former leurs employés. Les travailleurs nouveaux et expérimentés ont maintenant la possibilité d'améliorer leur base de connaissances de l'industrie et d'élargir leurs compétences. À la maison, les

3. individus ont eu accès à des programmes qui leur offraient la possibilité d'obtenir des diplômes en ligne et d'enrichir leur vie grâce à des connaissances élargies [18].

Définition

L'apprentissage en ligne est toutes les formes d'apprentissage et d'enseignement assistés par l'électronique (Ordinateur, Tablet, Téléphone, ...), qui ont un caractère procédural et visent à réaliser la construction de connaissances en référence à l'expérience, à la pratique et aux connaissances individuelles de l'apprenant. Les systèmes d'information et de communication, en réseau ou non, servent de supports spécifiques (spécifiques au sens développé précédemment) pour mettre en œuvre le processus d'apprentissage [19].

L'apprentissage en ligne peut être défini aussi comme l'utilisation de la technologie des réseaux informatiques, principalement sur un intranet ou via Internet, pour fournir des informations et des instructions aux individus [20].

Donc est un outil ou un système pédagogique informatisé qui vous permet d'apprendre n'importe où et à tout moment. Aujourd'hui, l'apprentissage en ligne est principalement dispensé via Internet, bien que dans le passé, il était dispensé à l'aide d'un mélange de méthodes informatiques comme le CD-ROM [18].

Le E-learning offre la possibilité de partager du matériel dans toutes sortes de formats tels que des vidéos, des diaporamas, des documents Word et des PDF. Organiser des webinaires (cours en ligne en direct) et communiquer avec les professeurs via des forums de discussion et de messagerie est également une option disponible pour les utilisateurs [18].

Technologie utilisée dans le E-learning

4. Aujourd'hui Les plateformes E-learning intègrent plusieurs technologies pour faciliter et assister l'utilisateur à mieux apprendre et aider l'enseignant à enseigner et créer leur cours, les technologies les plus communément utilisées sont :

- **Cours avec plusieurs format (vidéo, image, texte)**

Le processus d'apprentissage est un peu compliqué, et chacun a un mécanisme et l'outil d'apprentissage préféré, en général on trouve trois formats essentiels pour apprendre texte, image, ou vidéo et pour plaire toutes les parties, les plateformes e-learning implémente le contenu avec ses 3 formats et l'utilisateur a le choix entre eux.

- **Communication entre les utilisateurs et l'enseignant (chat)**

Pour répondre aux questions des élèves et en même temps diminuer le sentiment de solitude pour les étudiants.

- **Les avis pour le cours**

Chaque utilisateur a le droit de donner leur avis sur le cours qu'il a inscrit, et sa est très important car ils aideront l'enseignant à améliorer le contenu de leur cours et en même temps pour aider d'autre utilisateur à choisir le meilleur cours disponible.

- **Recommandation**

Consister à recommander des cours à l'utilisateur relative à leur intérêt.

- **Statistique**

Consister à donner les statistiques d'avancement de l'utilisateur dans leurs cours et leur courbe d'apprentissage.

- **Streaming Vidéo**

Cette technologie est utilisée généralement par les enseignants pour donner des cours en direct ou pour répondre aux questions d'utilisateur.

- **Personnalisation de la plateforme**

La chose que tout le monde accepte est que nous sommes différents et on a des préférences différentes, chaque personne est différente de l'autre, on veut dans la plateforme E-learning que l'utilisateur se sente à l'aise avec la plateforme c'est pour ça on lui donne la possibilité d'adapter la plateforme à ses préférences, donc par exemple choisir les couleurs, le volume de police, la mise en page, ...

- **Planification de l'utilisation de la plateforme avec calendrier**

Donc l'utilisateur peut définir son plan d'étude dans la plateforme, et cela l'aidera à clarifier les choses et rendre l'apprentissage un peu plus facile.

- **Gamification**

La gamification est l'utilisation de mécanismes, d'esthétique et de pensée basés sur les jeux pour engager et motiver les gens à agir, améliorer l'apprentissage et la résolution de problèmes [18].

- **Les tests et les questionnaires**

Les tests et les questionnaires jouent un rôle important dans l'apprentissage en ligne et offrent un éventail d'avantages pour l'apprenant et l'instructeur, par exemple fournir à l'utilisateur les résultats instantanément, garder les apprenants engagés car l'étudiant sait que sa progression sera jugée lors d'un examen, faciliter la correction des tests et plusieurs autres choses on peut trouver dans [18].

- **Paiement par internet**

Lorsque le cours n'est pas gratuit, cette technologie rend le processus de paiement beaucoup plus facile pour l'utilisateur et pour l'enseignant, car le paiement sera par internet et instantanément et automatiquement.

Les avantages et les inconvénients

Les avantages [18] :

Le e-learning offre un punch en termes d'avantages qui facilitent et simplifient les processus de création et de livraison.

5.1. • **Pas de limites, pas de restrictions :**

Outre les restrictions géographiques, le temps est l'un des problèmes auxquels les apprenants et les enseignants doivent faire face dans leur apprentissage. Le e-learning, facilite l'apprentissage sans avoir à organiser le moment et l'endroit où tous ceux qui s'intéressent à un cours peuvent être présents.

• **Plus amusant**

Concevoir un cours d'une manière qui le rend interactif et amusant grâce à l'utilisation du multimédia ou des méthodes de gamification plus récemment développées améliore non seulement votre facteur d'engagement, mais également la durée de vie relative du matériel de cours en question.

• **Rentable**

Cela s'adresse à la fois aux apprenants et aux enseignants, mais il y a de fortes chances que, quel que soit votre rôle, vous deviez payer des sommes exorbitantes à un moment donné pour acquérir des versions mises à jour de manuels scolaires ou universitaires. Alors que les manuels deviennent souvent obsolètes après un certain laps de temps, la nécessité d'acquérir constamment de nouvelles éditions n'est pas présente dans l'apprentissage en ligne.

5.2.

Les inconvénients :

Même compte tenu de tous les avantages de l'apprentissage en ligne, on ne peut nier qu'il existe certains inconvénients.

- Les compétences pratiques sont un peu plus difficiles à acquérir à partir des ressources en ligne. Par exemple, bien que la construction d'une table en bois soit quelque chose sur lequel vous pouvez facilement partager des informations, enregistrer des vidéos et expliquer, l'expérience pratique est essentielle. La poterie et l'ingénierie automobile sont des exemples de compétences qui nécessitent une expérience pratique. [18]

- **Isolement** [18]

L'apprentissage en ligne est pour la plupart un acte isolé, ce qui peut donner à l'apprenant le sentiment d'agir complètement seul. Les apprenants peuvent ressentir un sentiment d'isolement.

- **Préoccupations liées à la santé** [18]

L'apprentissage en ligne nécessite l'utilisation d'un ordinateur et d'autres appareils similaires, cela signifie que la fatigue oculaire, une mauvaise posture et d'autres problèmes physiques peuvent affecter l'apprenant.

- **Problème d'accès à Internet**

Les plateformes d'apprentissage en ligne contiennent plusieurs services et technologie, cela implique que l'utilisateur doit avoir une bonne vitesse de connexion pour pouvoir l'utiliser, mais malheureusement ce critère n'est pas disponible à tout le monde.

6. **Modalités d'adaptation dans le E-learning**

Dans cette section on va discuter comment l'adaptabilité peut-elle améliorer l'apprentissage en ligne ou en d'autres termes comment on peut appliquer l'adaptation dans les plateformes E-learning.

6.1. On peut divisée l'adaptation dans E-learning en trois catégories :

Adaptation du contenu

Le contenu des pages doit s'adapter aux caractéristiques de l'utilisateur. Par conséquent, n'est pas la même information qui sera présentée à tous les utilisateurs. L'information peut être transmise en utilisant de différents termes, de phrases courtes ou lentes et plus ou moins de détails. L'adaptation du contenu consiste à sélectionner l'information qui dépend de l'état courant du modèle utilisateur. Pour un débutant, le contenu des pages sera différent à celui d'un expert pour un sujet déterminé. L'utilisateur peut se trouver à une situation où il ne comprend pas un concept ou ne peut distinguer la différence entre deux concepts reliés. Donc, l'adaptation du contenu vise soit à cacher une portion du contenu qui semble inappropriée au contexte d'un utilisateur particulier, soit à fournir une information supplémentaire à l'égard d'un contenu destiné à un ensemble d'utilisateurs, ou à choisir une des alternatives prédéfinies qui apparaît

la plus appropriée à un utilisateur donné. L'explication est une méthode qui permet d'enrichir la compréhension du contenu. On distingue quatre méthodes : l'explication additionnelle, l'explication comparative, l'explication pré requise et l'explication variante [21].

Adaptation de la présentation

L'adaptation de la présentation consiste à ajuster la disposition et l'aspect visuel, des éléments (des fragments) qui seront présentés, aux attentes ou aux préférences de l'utilisateur. Le principe est que la même information peut être présentée de différentes façons. L'adaptation de la présentation offre à l'utilisateur plusieurs présentations alternatives en lui permettant de choisir une ou plus selon son choix. La présentation adaptative a l'intérêt d'améliorer l'attention de l'utilisateur. L'adaptation de la présentation peut être aussi appelé Adaptation des IHM [5].

Adaptation de la navigation

6.3. L'utilisateur peut se perdre dans un hyperespace lorsque le système lui fournit une quantité très importante d'informations. Pour que l'utilisateur puisse s'orienter facilement aux informations dont il a besoin, le système doit apporter une bonne navigation. Donc, une mise en place des mécanismes d'aide à la navigation est devenue nécessaire. Le principe est de fournir à un utilisateur des conseils pouvant être lui fournis par un expert connaissant ses préférences et ses compétences [5].

7.

Quelques plateformes E-learning

L'utilisation des plateformes E-learning a fait de grands progrès ces dernières années et c'est ce qui a conduit à l'émergence de nombreuses plateformes E-learning. Ci-dessous nous présenterons quelques plateformes E-learning populaires.

Udemy ¹

Udemy, Inc. est une plateforme e-learning destinée aux adultes et aux étudiants professionnels. Il a été fondé en mai 2010 par Eren Bali, Gagan Biyani et Oktay Caglar, En janvier 2020, la plateforme comptait plus de 35 millions d'étudiants et 57000 instructeurs dispensant des cours dans plus de 65 langues. Il y a eu plus de 400 millions d'inscriptions aux cours. Les étudiants et les instructeurs viennent de plus de 180 pays et 2/3 des étudiants sont situés en dehors des États-Unis [22].

¹ Accessible sur www.udemy.com

Coursera ²

Est une plateforme e-learning fondé en 2012 par les professeurs d'informatique de l'Université de Stanford, Andrew Ng et Daphne Koller, qui propose des cours en ligne ouverts massifs, des spécialisations, des diplômes, des cours professionnels et des cours de master.

7.2.

Coursera travaille avec des universités et d'autres organisations pour proposer des cours en ligne, des certifications et des diplômes dans une variété de sujets, tels que l'ingénierie, la science des données, l'apprentissage automatique, les mathématiques, les affaires, le financement, l'informatique, le marketing numérique, les sciences humaines, la médecine, la biologie, sciences sociales, 3000 plus une variété de cours donnant aux étudiants un très large éventail d'informations et d'expériences dans différents domaines [23].

Moodle ³

7.3. Moodle est un système de gestion de l'apprentissage (LMS) gratuit et open-source écrit en PHP et distribué sous la licence publique générale GNU. Développé sur des principes pédagogiques, Moodle est utilisé pour l'apprentissage mixte, l'enseignement à distance, la classe inversée et d'autres projets d'apprentissage en ligne dans les écoles, les universités, les lieux de travail et d'autres secteurs.

Avec des fonctionnalités de gestion personnalisables, il est utilisé pour créer des sites Web privés avec des cours en ligne pour les éducateurs et les formateurs afin d'atteindre les objectifs d'apprentissage. Moodle permet d'étendre et de personnaliser les environnements

8. d'apprentissage à l'aide de plugins provenant de la communauté [24].

Quelques plateformes E-learning adaptatives

Malheureusement, la majorité des plateformes E-learning populaires qu'on a présenté dans la section précédente n'offrent pas ou peu de caractéristiques d'adaptabilité. Bien que l'adaptabilité soit un sujet de recherche depuis une quinzaine d'années, elle est encore utilisée principalement dans les projets de recherche plutôt que dans les plateformes E-learning les plus fréquemment utilisés, on va présenter quelque projet de recherche populaire qui ont un soutien étendu pour l'adaptabilité, la plupart de ces systèmes fournissent à la fois une prise en charge de l'adaptation de contenu et de navigation :

² Accessible sur www.coursera.org

³ Accessible sur <https://moodle.org/>

8.1.1. AHA

Est une architecture hypermédia adaptative ouverte fournissant une présentation de contenu adaptative basée sur des fragments ainsi que des annotations de liens et un masquage de liens [25].

8.1.2. ANATOM-TUTOR

Est un système adaptatif pour l'enseignement de l'anatomie. Il peut être utilisé dans trois modes différents : mode navigation (sans aucune adaptabilité), mode question (utilisation intensive du modèle utilisateur pour trouver des questions et évaluer les réponses) et hypermode (présentation adaptative et support de navigation) [25].

8.1.3. iWeaver

Est un projet de doctorat conçu pour fournir un environnement d'apprentissage adaptatif et flexible pour le langage de programmation Java. Le système crée un profil de l'apprenant en évaluant les styles d'apprentissage à l'aide d'une gamme de questions à choix multiples lorsque l'utilisateur entre pour la première fois. Les utilisateurs ultérieurs reçoivent des recommandations personnalisées et une vue individuelle des outils d'apprentissage disponibles. iWeaver combine des techniques de navigation adaptative et de présentation de contenu adaptative [25].

9. Discussion

Le tableau suivant contient la comparaison entre les plateformes E-learning populaire, adaptatif et notre plateforme.

Tableau 1 : Comparaison entre les plateformes populaire, adaptatif et notre plateforme

Plateforme	Populaire	Adaptatif	Notre plateforme
Technologies	Beaucoup	Peu	Beaucoup
Adaptation	Adaptable	Adaptatif (contenu, présentation, navigation)	Adaptatif (présentation, navigation)
Expérience d'utilisateur	Bonne	Mauvaise	Bonne

Dans **les plateformes populaires** on trouve :

- Beaucoup de technologies qui facilite l'apprentissage à l'utilisateur comme : statistiques, recommandation, communication, ...etc.
- Les plateformes populaires sont des plateformes adaptables c'est-à-dire l'adaptation spécifique explicitement par utilisateur, et cela ne nous aide pas car dans notre projet nous

avons besoin d'une plateforme adaptatif c'est-à-dire l'adaptation est inférée automatiquement par le système.

- L'utilisateur aura une bonne expérience d'utilisation de ces plateformes.

Et dans **les plateformes adaptatives** qui sont des plateformes de recherche on trouve :

- Peu de technologie utiliser dans ces plateformes car leur but principal est de clarifie l'adaptation, donc on trouve presque que les technologies qui sont en relation avec le cours comme : cours avec plusieurs formats et les liens de navigation de cours.
- Par rapport à l'adaptation ces plateformes sont des plateformes adaptatives c'est-à-dire l'adaptation est inférée automatiquement par le système, et avec l'utilisation presque de toutes les modalités d'adaptation (contenue, présentation, navigation).
- L'utilisateur n'aura pas une bonne expérience d'utilisation de ces plateformes car elles sont principalement destinées à la recherche.

Dans **notre plateforme** on va essayer de tirer le meilleur de deux donc on trouve :

- Beaucoup de technologies qui facilite l'apprentissage à l'utilisateur dans la plateforme.
- Notre plateforme est adaptative c'est-à-dire l'adaptation est inférée automatiquement par le système, et dans les modalités d'adaptation on va utiliser l'adaptation de présentation et de navigation.
- L'utilisateur aura une bonne expérience d'utilisation de notre plateforme.

Donc dans notre projet de fin d'étude on va créer une plateforme similaire au plateforme populaire qui contient plusieurs technologies et au même temps avec une adaptation automatique par rapport aux vitesses de connexion d'utilisateur, avec l'utilisation d'une
10. approche basé sur les algorithmes génétiques pour assurer la vitesse de réponse de notre plateforme dans les faibles connexions d'utilisateur.

Conclusion

Dans ce chapitre, nous avons d'abord expliqué le E-learning, son histoire, son importance et les avantages et inconvénients offerts par lui, puis nous avons appris les différentes technologies utiliser dans le E-learning, enfin nous avons vu des exemples des plateformes E-learning actuellement célèbre et explique la fonctionnalité qui distingue notre plateforme.

CHAPITRE 4 : Conception et modélisation de l'approche

Introduction

Parmi les grands challenges des plateformes E-learning dans les pays du tiers-monde c'est d'assurer la performance et le temps de réponse de la plateforme dans des vitesses de

1. connexions faibles.

L'objectif de notre projet est de trouver une solution qui permet d'adapter l'utilisation de différentes fonctionnalités de plateforme à un instant donné avec la vitesse de connexion d'utilisateur, Cette solution doit être auto adaptative sans aucune intervention manuelle.

Nous nous intéressons dans ce chapitre à la conception de notre étude de cas (plateforme E-learning) et de notre approche qui permet la planification de l'auto adaptation, en se basant sur la métaheuristique précédemment étudiées (Algorithme génétique).

Cas d'étude

2.

Nous avons choisi comme étude de cas une plateforme E-learning qui intégré plusieurs services (technologies) pour faciliter et assisté l'utilisateur à mieux apprendre leur cours, L'application fournit l'ensemble des services variables suivants :

- Accès aux cours avec plusieurs format : vidéo, image, texte.
- Echange des messages entre enseignants et étudiants avec plusieurs formats (texte, image, vidéo).
- L'utilisateur a le droit de donner son avis dans le cours où il est inscrit.
- Recommander des cours à l'utilisateur relative à leur intérêt.
- Statistique de l'avancement de l'utilisateur dans leurs cours.
- Streaming Vidéo fait par les enseignants pour donner des cours en direct ou pour réponde aux questions d'utilisateur.
- Notification pour l'utilisateur avec plusieurs formats : vidéo, image, texte.
- Personnalisation de la plateforme selon les préférences de l'utilisateur, comme les couleurs de la plateforme, le volume de police, la langue utiliser, ...

Cette application peut être adaptée au :

- ✓ Vitesse de connexion (par exemple, utiliser cours avec texte par ce que la vitesse de connexion est faible).
- ✓ Préférences de l'utilisateur (par exemple, un cours avec vidéo est préférable qu'avec image).
- ✓ Consommation des données de connexion.

Modélisation des variations dans la plateforme proposée

On a représenté toutes les variations des services qu'on a dans notre plateforme avec le modèle de fonctionnalité.

Dans le développement logiciel, un modèle de fonctionnalité « feature modèle » est une représentation compacte de tous les produits de la gamme de produits logiciels en termes de « fonctionnalités ». Une fonctionnalité est une propriété de système qui est pertinente pour certaines parties prenantes et qui est utilisée pour capturer des points communs ou faire une distinction entre les systèmes [26].

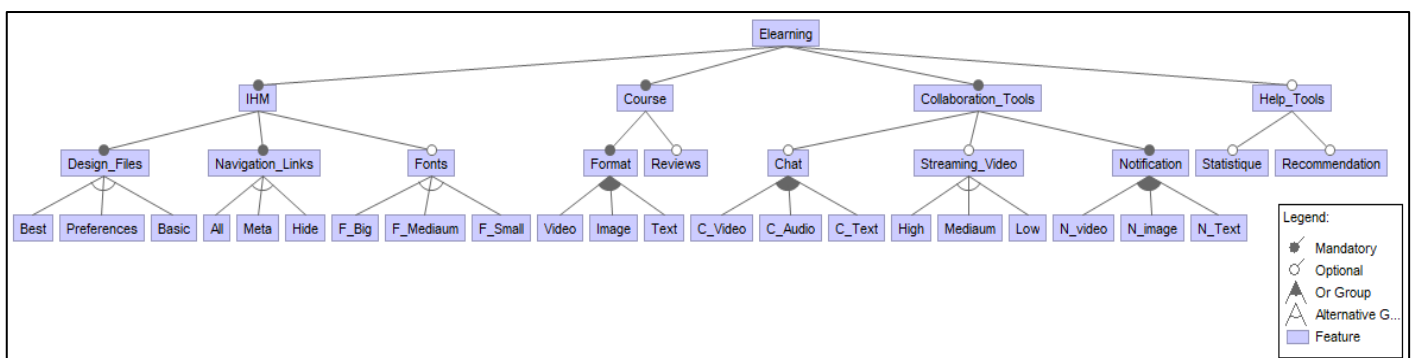


Figure 10 Modélisation des variations

Modélisation de notre système

Pour concevoir notre système, nous avons choisi d'utiliser le langage de modélisation unifié UML (Unified Modeling Language). Ce dernier est un langage de modélisation graphique structuré sur un méta modèle définissant les éléments de modélisation (concept manipulé par le langage) et la sémantique de ces éléments (définitions et sens de leurs utilisations). C'est un langage formel organisé autour des diagrammes qui seront développés dans la suite du présent chapitre.

2.2.1. Diagramme des cas d'utilisation

Les cas d'utilisation permettant de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Dans un diagramme des cas d'utilisation, Un acteur représente

un rôle joué par une personne qui interagit avec le système. La figure ci-dessous illustre le diagramme de cas d'utilisation.

Dans notre plateforme un utilisateur peut jouer le rôle d'un apprenant et d'un enseignant au même temps dans des différents cours.

Par exemple l'utilisateur 1 est un apprenant dans le cours A, et il a créé le cours B donc il est au même temps un enseignant dans le cours B.

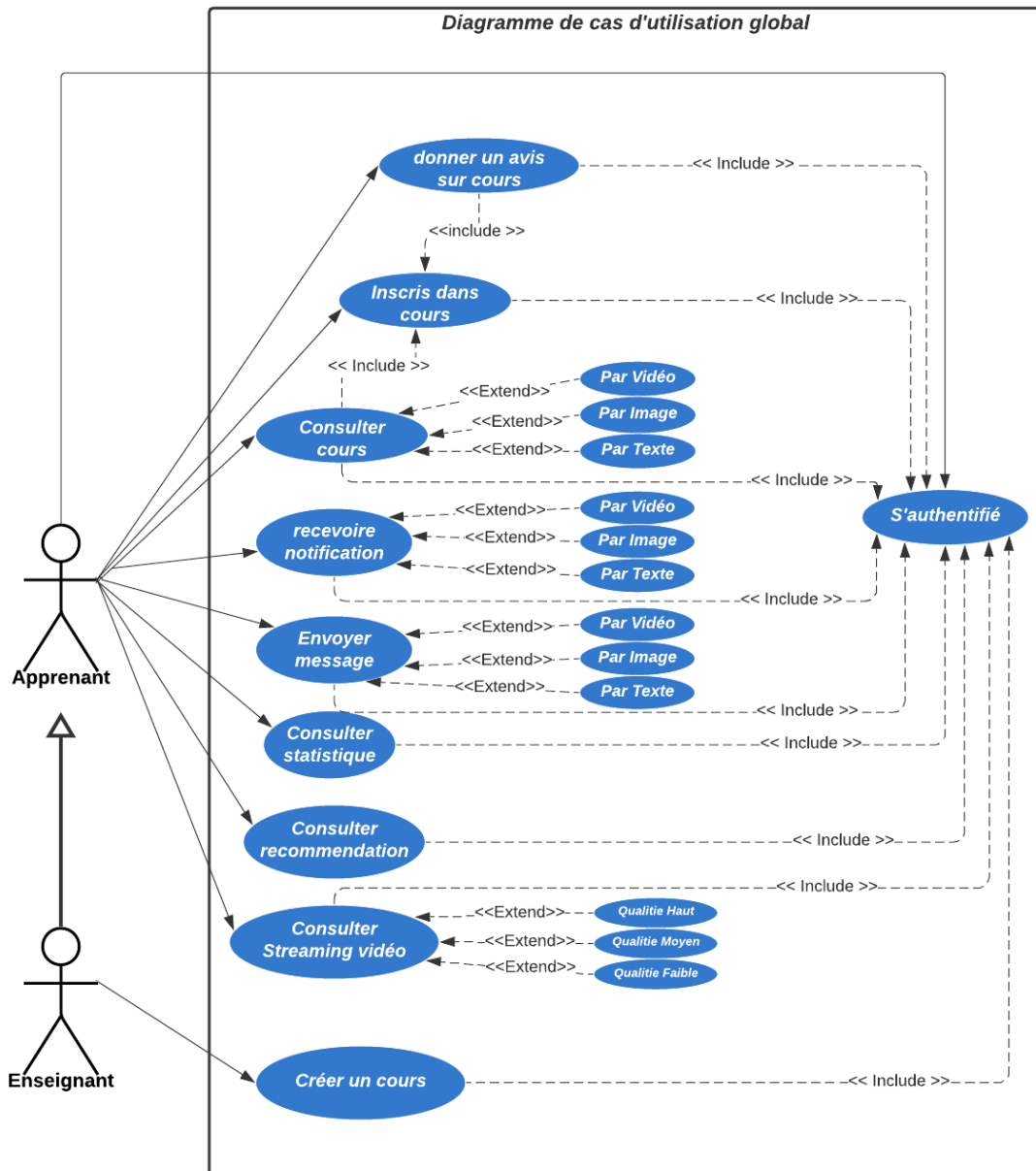


Figure 11 Diagramme de cas d'utilisateur

2.2.2. Diagramme de classe

Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles sont regroupées dans des classes. Le diagramme met en évidence d'éventuelles relations entre ces classes. Le diagramme de classe comporte 6 concepts : classe, Attribut, identifiant, relation, opération et généralisation / spécialisation.

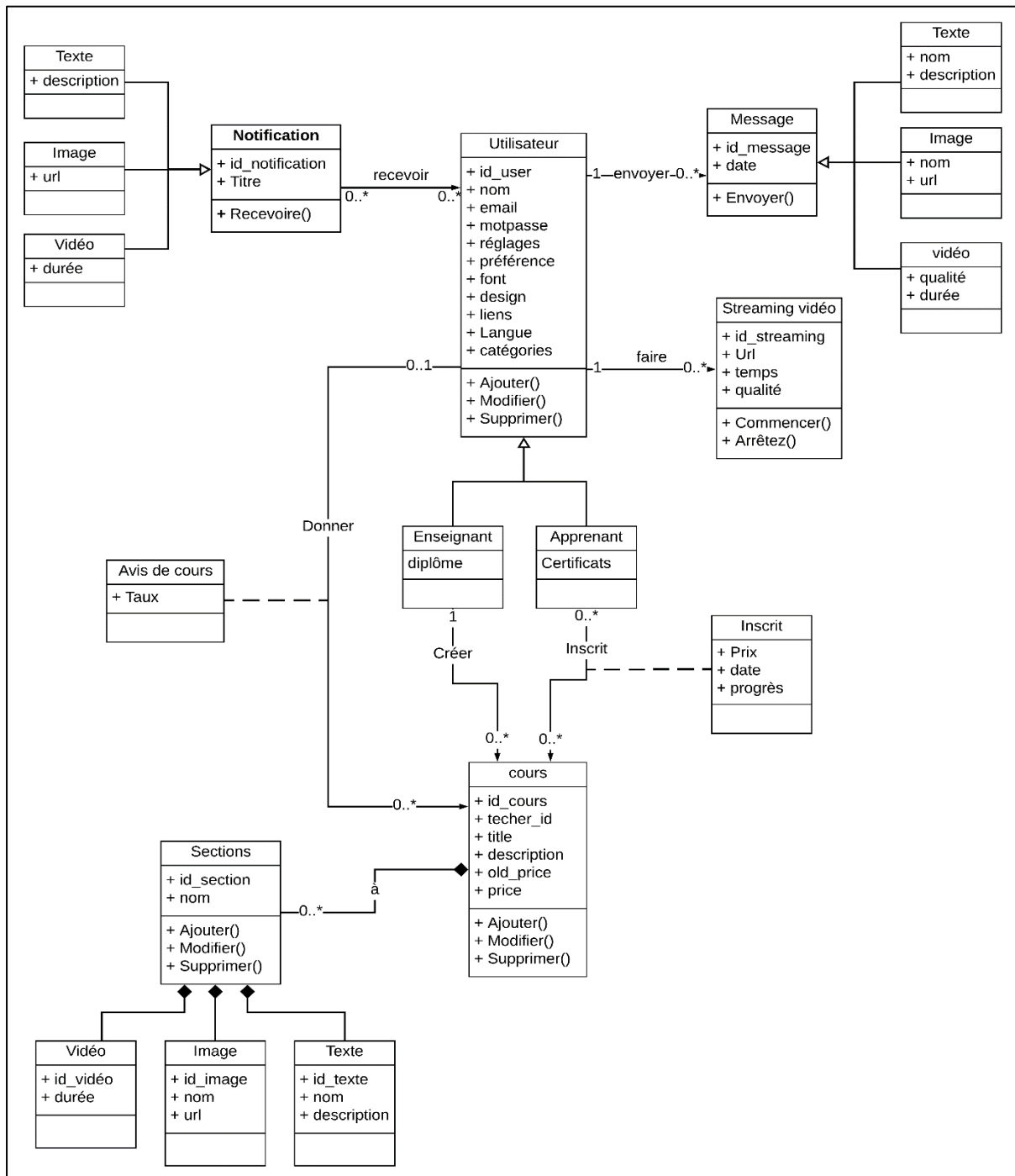


Figure 12: Diagramme de classe

Description de l'approche

Tous d'abord dans notre plateforme E-learning on a :

3.
 - Plusieurs **services** ou on peut dire fonctionnalités (ex cours, streaming, recommandation, ...etc.)
 - Chaque service a plusieurs **variations** (ex cours : vidéo, image, texte).
 - Chaque variation a une **utilité** et une **consommation de données** (Kbps) et un **temps de réponse**(ms).

L'utilité c'est l'efficacité de l'utilisation de ce service. Par exemple l'utilité de cours avec vidéo est meilleure que l'utilité de cours avec texte à cause de son efficacité.

La consommation des données (Kbps) c'est combien le service consomme les données de connexion.

Par rapport au **temps de réponse (ms)** les chose sont un peu compliqué, car pour nous on veut diminuer le temps de réponse d'utilisateur dans la plateforme et pour avoir ce temps on doit connait :

- Premièrement le temps de réponse de serveur de notre plateforme pour l'obtention du service (l'achèvement de la requête).
- Deuxièmement le temps de réponse de connexion d'utilisateur qui est connu sous le nom **Latence** « **en anglais Latency** » de l'utilisateur qu'on peut connait avec un test de connexion⁴.

Donc comme le montre la Figure 13 ci-dessous, le temps de réponse d'un utilisateur dans notre plateforme (400ms) **égale** leur temps de réponse de connexion ou **latence** (300ms) **plus** le temps de réponse de serveur de notre plateforme (100ms).

⁴ <https://testmy.net/latency>

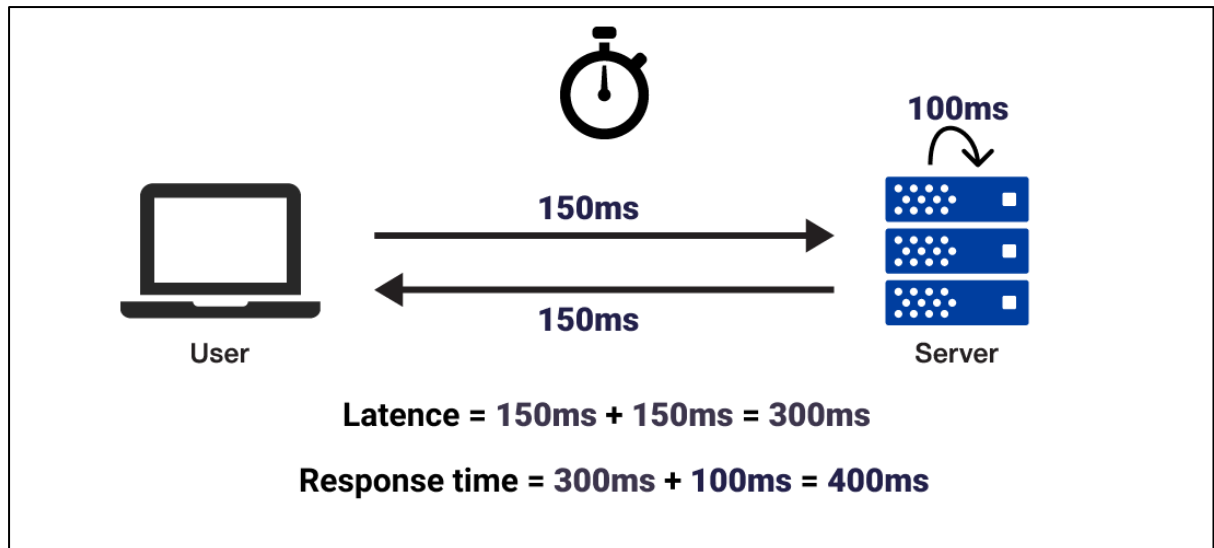


Figure 13 Représentation de latence

Déclenchement de notre approche

3.1. Les critères principaux qui décident le démarrage de notre approche c'est la vitesse de connexion d'utilisateur plus spécifiquement leur temps de réponse dans la plateforme avec cette vitesse, donc si l'utilisateur à un bon temps de réponse (qui implique une bonne vitesse de connexion) il peut utiliser tous les services avec les meilleures variations, car notre approche est concernée avec le cas contraire (faible temps de réponse qui implique une faible connexion).

La vitesse de connexion constitue de 3 facteurs principale :

- **Le débit descendant** également appeler débit en réception (ou download en anglais) correspond à la vitesse de téléchargement.
- **Le débit montant** aussi appeler débit en envoi (ou upload) se rapporte à la vitesse de transfert de vos appareils vers le réseau internet.
- **Latence** c'est le délai qui s'écoule entre le départ d'une information depuis d'un ordinateur et son arrivée sur un serveur web, il s'agit donc de la vitesse de réaction de votre connexion internet.

Dans le déclenchement de notre approche on est intéressé au temps de réponse et le facteur qui représente sa dans la vitesse de connexion est **la latence** (le délai) de connexion d'utilisateur, plus la connexion est vite plus la latence est petite, et plus la **latence** et petite plus le **temps de réponse** est petit, donc si l'utilisateur à une grande latence c'est-à-dire leur temps de réponse est grand alors notre approche doit démarrer (limite de temps de réponse d'utilisateur dans la plateforme qui ne doit pas dépasser est 3000ms dans notre cas).

Pour faire sa on a créé une stratégie qui surveiller la latence de connexion d'utilisateur avec le temps de réponse des services activé dans notre plateforme et décide si on active notre approche ou non.

Donc on a un **observateur** qui exécute une fonction lorsque la valeur de latence d'utilisateur change, cette fonction va calculer la somme de temps de réponse (TR) de tous les services activés dans la plateforme **plus** la latence de l'utilisateur pour chaque service active, si le résultat est supérieur à un seuil alors notre approche va démarrer pour donner un plan optimal qui diminue le temps de réponse, sinon rien à faire.

On peut représenter la stratégie comme suit :

Si $(\sum E(i) (TR(i) + \text{latence}) > \text{seuil})$

Alors Approche Activé

Sinon rien à faire

- **TR(i)** : Temps de réponse de serveur de plateforme pour le service i.
- **E(i)** : L'état de service i, prendre les valeurs 0 si désactiver et 1 si activé.
- **Latence** : Temps de réponse de connexion d'utilisateur.
- **Seuil** : Le niveau de temps de réponse d'un utilisateur dans notre plateforme et qui ne doit pas dépasser 3000ms dans notre cas.

Notre stratégie exécute et fait ce teste lorsque l'utilisateur connecté pour la première fois dans 3.2. notre plateforme et lorsque la vitesse de connexion d'utilisateur change.

Fonctionnement de l'approche

Notre approche va utiliser les algorithmes génétiques pour générer un plan d'adaptation optimal (la meilleure stratégie d'adaptation) dans les cas de connexion faible, notre problématique peut être présentée comme un problème d'optimisation (une minimisation de temps de réponse et de consommation de donnés avec une maximisation d'utilité des fonctionnalités).

Donc si la connexion est faible notre approche générer un plan d'adaptation contient une variation depuis chaque service.

Donc par exemple un plan peut être :

- Cours : Image.

- Design files : Préférable.
- Lien de navigation : Meta.
- Chat : Texte.
- Notification : Image.
- Police : Petit.
- Avis de cours : Désactiver.
- Statistique : Activer.
- Streaming vidéo : qualité faible.

But de notre approche

Notre but est de **Diminuer** le temp de réponse de la plateforme, et la consommation de données 3.3. de connexion et **maximiser** l'utilities des services par rapport aux vitesses de connexion d'utilisateur par l'augmentation au la diminution de qualité des services.

Tableau de variations

4. Cette table contient tous les services et leurs variations dans notre plateforme avec leur temps de réponse(ms), utilité et consommations de données de connexion (Kbps).

Tableau 2 Tableau de variations

Variations Table ☰ Default view ▾				
☰ Services	Aa Variation	# Response Time (ms)	# Utility	# Consumption (kbits)
Course	Video	400	100	500
Course	Picture	200	70	200
Course	Text	100	30	100
Design	Best	450	80	300
Design	Preferable	200	60	150
Design	Basic	100	30	100
Navigation Link	All	200	80	200
Navigation Link	Meta	100	60	120
Navigation Link	Hide	0	20	10
Chat	Disabled	0	0	0
Chat	Video	400	70	500
Chat	Audio	200	50	200
Chat	Text	100	20	150
Notification	Disabled	0	0	0
Notification	Picture	200	30	120
Notification	Text	150	20	80
Notification	Video	50	60	500
Font Size	Disabled	0	0	0
Font Size	Big	150	40	40
Font Size	Medium	100	20	30
Font Size	Small	50	10	20
Course Reviews	Disabled	0	0	0
Course Reviews	Activate	200	80	200
Recommendation	Activate	250	80	200
Statistic	Disabled	0	0	0
Statistic	Activate	200	80	200
Streaming Video	Disabled	0	0	0
Streaming Video	High Quality	600	70	600
Streaming Video	Medium Quality	400	60	400
Streaming Video	Low Quality	200	30	200

Codage du chromosome

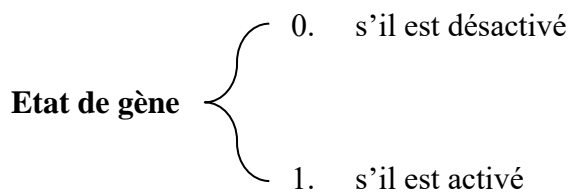
Le chromosome (individu) est un plan de solution possible pour notre approche, l'AG utilise plusieurs chromosomes dans leur fonctionnement, c'est pour ça le codage est l'une des étapes importantes pour l'implémentation de l'AG.

5.

Un chromosome constitué de gènes, chaque gène représente un service de plateforme (cours avec vidéo, recommandation activé, ...).

Le gène se caractérise par :

1. **Un état** : si le service est activé ou désactivé



2. **Une utilité** : qui le spécialise, l'utilité c'est-à-dire l'efficacité de l'utilisation de ce service. Par exemple l'utilité de cours avec vidéo est meilleure que l'utilité de cours avec texte à cause de son efficacité.
3. **Une consommation des données** : c'est à dire combien le service consomme les données de connexion.
4. **Un temps de réponse** : Temps de réponse de service.

5.1. Le chromosome réel est formé en collectant simplement tous les gènes (tous les services).

Description des services existants

Dans notre approche un service peut avoir plusieurs variations, par exemple le cours peut être : vidéo, image et texte, Pour garder la structure correcte d'un individu et éviter de tomber dans une configuration qui n'a aucun sens (ex : l'approche donne une solution de cours avec vidéo et avec image), nous avons effectué un classement des services et un codage d'état des services de la manière suivante :

Les variations des services sont choisies d'une manière aléatoire

Service 1 : Course

Tableau 3 Codage de service de cours

State	Video	Picture	Text
-------	-------	---------	------

Coding	0	1	2
---------------	---	---	---

Service 2 : Design

Tableau 4 Codage de service de design

State	Best	Preferable	Basic
Coding	0	1	2

Service 3 : Navigation Links

Tableau 5 Codage de service de navigation des liens

State	All	Meta	Hide
Coding	0	1	2

Service 4 : Chat (communication)

Tableau 6 Codage de service de chat

State	Disabled	Video	Picture	Text
Coding	0	1	2	3

Service 5 : Notification

Tableau 7 Codage de service de notification

State	Disabled	Video	Picture	Text
Coding	0	1	2	3

Service 6 : Font Size

Tableau 8 Codage de service de font

State	Disabled	Big	Medium	Small
Coding	0	1	2	3

Service 7 : Course Reviews

Tableau 9 Codage de service d'avis de cours

State	Activate	Disabled
Coding	1	0

Service 8 : Statistic

Tableau 10 Codage de service de statistique

State	Activate	Disabled
Coding	1	0

Service 9 : Recommendation

Tableau 11 Codage de service de recommandation

State	Activate	Disabled
Coding	1	0

Service 10 : Streaming video (Quality)

Tableau 12 Codage de service Streaming vidéo

State	Disabled	High	Medium	Low
Coding	0	1	2	3

La population initiale

6.

La population initiale est considérée comme un point de départ pour l'AG et on l'appelle la première génération. Les générations futures se développent en fonction d'elle pour atteindre la meilleure solution avec l'utilisation d'opérateurs génétiques (sélection, crossover, mutation).

Une bonne initialisation aide d'une manière considérable l'évolution du processus. On a plusieurs mécanismes pour générer une population initiale. Les plus connues sont : le tirage aléatoire, les heuristiques ou une combinaison de solutions heuristique et l'aléatoire. Dans notre approche, on génère la population initiale de façon aléatoire, l'état de service est fait par la fonction « Random » qui donne des variations aléatoires parmi les variations existantes d'un

Services : 000, 001, 010... et 111, on applique cette fonction pour tous les services.

Chromosome 0 1 2 0 1 2 2 0 0 1 1 (10 services)

Chromosome 1 0 2 1 2 1 4 1 2 0 2

Chromosome 2	1	1	2	3	3	1	1	2	2	3

Tableau 13 Exemple d'une population

Donc par exemple la description de chromosome 0 est :

- Service1 (cours) : 1 => Image.
- Service 2 (Fichiers de design) : 2 => De Base.
- Service 3 (Liens de navigation) : 0 => Tous.
- Service 4 (Chat) : 1 => Vidéo.
- Service 5 (Notification) : 2 => Texte.
- Service 6 (Police) : 2 => Moyen.
- Service 7 (Avis de cours) : 0 => Désactiver.
- Service 8 (Statistique) : 0 => Désactiver.
- Service 9 (Recommandation) : 1 => Activer.
- Service 10 (Streaming vidéo) : 1 => Qualité haut.

Fonction de fitness

7. Pour déterminer la meilleure solution dans l'espace de solution (trouver l'individu le plus proche au l'optimum) on a besoin de fonction qui s'appelle fonction de fitness. Cette dernière utilise des critères qui spécialise notre problème d'optimisation pour calculer l'optimalité de chaque individu.

Dans notre système nous sommes concentrés sur les trois critères suivants :

- ✓ **Critère 1** : la minimisation de temps de réponse.
- ✓ **Critère 2** : la maximisation d'utilité
- ✓ **Critère 3** : la minimisation de la consommation de données.

La fonction de fitness $f(X)$ est défini comme étant la somme pondérée de ces trois objectifs (critères), caractérise par :

- **Temps de réponse TR (i)** : temps de réponse d'un service.
- **L'utilité U (i)** : l'utilité de service ou l'efficacité de service.
- **Consommation C(i)** : consommation de données d'un service.
- **Les poids Wi** : sont présentés sous forme de nombres de types réels associés à chaque critère, et qui traduisent leurs importances relativement à l'exigence de l'utilisateur. La somme de ces poids égal à 1.

On n'a choisi que critères 1 à un poids (W_1) de 50 % et critères 2 et 3 ont un poids de 25%. Parce que l'importance est de minimiser de temps de réponse et plus important pour nous que de maximisé l'utilité et minimiser la consommation de données

Tableau 14 les poids des critères.

Critère	Poids
Temps de réponse moyen (T)	50%
Consommation moyen (C)	25%
Utility moyen (U)	25%

- **Etat E(i) :** si le gène est actif (1) sinon (0)

La fonction fitness $f(X)$ pour le chromosome X de N gènes peut être exprimée comme suit :

Si on minimise un critère on multiplie par (-1) Sinon (maximiser) on multiplie par (1)

$$f(X) = \sum_{i=1}^N E(i) (-W1 * TRi - w2 * Ci + W3 * Ui)$$

- Dans cette fonction on peut trouver des valeur fitness chromosome négative qui cause des problèmes dans le prochain état de l’algorithme génétique (sélection). Donc on va normaliser avec une formule pour éviter les possibilités d’avoir une valeur négative de fitness.

La formule est expliquée en détail dans la section suivante.

7.1.

Matrice de limites

En utilisant des critères de qualités, nous définirons une matrice de limites ML. ML donne une plage de valeurs acceptable pour chaque critère et son poids. Chaque colonne représente un critère, représentant respectivement la valeur minimale acceptable, puis la valeur maximale acceptée et enfin son poids.

7.2.

$$ML = \begin{pmatrix} Q_1^{min} & Q_2^{min} & Q_3^{min} \\ Q_1^{max} & Q_2^{max} & Q_3^{max} \\ w_1 & w_2 & w_3 \end{pmatrix}$$

La fonction de fitness normalisée

On a trois critères pour normalisation : temps de réponse Q1, consommation Q2, utilité Q3. Puisque Les trois critères de la qualité ayant des normes différentes, nous devons les normaliser sur la même échelle dans [0,1].

Supposons que $Q(S) = X(S) \cup Y(S)$ où $X(S) = \{Q_3(S)\}$ un ensemble de critères de qualité positifs et $Y(S) = \{Q_1(S), Q_2(S)\}$ ensemble de critères de qualité négatifs. On note une meilleure qualité de X par des valeurs plus élevées calculées par l'équation (6) et une meilleure qualité de Y par des valeurs plus faibles. Les valeurs des critères de qualité de X (respectivement Y) sont normalisées à X' (respectivement Y') comme :

$$X'_i(S) = \begin{cases} \frac{X_i(S) - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}} & \text{Si Critère à maximiser} \quad i=3 \end{cases} \quad (6)$$

$$Y'_j(S) = \begin{cases} \frac{Q_j^{\max} - Y_j(S)}{Q_j^{\max} - Q_j^{\min}} & \text{Si Critère à minimiser} \quad j=1,2 \end{cases} \quad (7)$$

La fonction fitness f est calculée par une simple somme pondérée de valeurs des critères normalisés :

$$f(S) = \sum_{\forall i|Q_i \in X} W_i * X'_i(S) + \sum_{\forall j|Q_j \in Y} W_j * Y'_j(S) \quad (8)$$

8. Les opérateurs génétiques

8.

8.1. La sélection

Avant l'opération de sélection nous avons déjà la population initiale, donc la sélection consiste à choisir les solutions potentiellement utiles pour la recombinaison et qui influencera la prochaine génération.

Il existe plusieurs méthodes de sélection, telles que la sélection par rang ou la sélection par la méthode de l'élitisme ou par roulette etc.

8.2. Dans notre approche on va utiliser la sélection par élitisme qui consiste à choisir les meilleurs chromosomes par rapport à la fonction de fitness.

Le croisement uniforme

Après la sélection qu'on a citée ci-dessus plusieurs parents ont été choisis, pour obtenir des enfants on doit faire un croisement (ou on peut dire un couplage) entre ces parents.

La méthode la plus connue de croisement est le croisement par pourcentage, c'est-à-dire lorsque on a deux parents C1 et C2, le chromosome enfant sera 75% de C1 et le reste de C2, et on va faire ce croisement avec plusieurs parents pour obtenir plusieurs enfants.

La mutation uniforme

Après l'étape du croisement, la prochaine génération est construite à partir de la mutation de tous les chromosomes de l'ancienne génération.

Pour chaque service (gène) on fait un test avec l'aide de fonction Random qui donne un nombre

8.3. aléatoire entre 0 et 1 et si le nombre est petit au poids de mutation (0.25) on fait la mutation sinon on ne fait pas.

La mutation consiste à changer la variation de service aléatoirement par une autre variation.

Par exemple :

Aléatoirement le choix de service qui va faire la mutation dans un chromosome est venu dans le service cours, qui a comme variation cours avec vidéo. Avec la mutation on fait un choix aléatoire entre les variations de service cours (vidéo, image, texte) pour obtenir la même ou une autre variation.

Le critère d'arrêt

8.4. Le critère d'arrêt peut être différent d'un problème à un autre, peut être fixé en imposant un nombre maximal de générations ou lorsque on arrive à une valeur de fonction de fitness prédéfinie, dans notre approche on va utiliser la première qui est fixé un nombre maximal de générations.

On estime alors que l'algorithme a convergé et que l'individu de plus forte performance en une 9. des générations correspond à la solution au problème.

Représentation de l'approche

Dans cette section on va essayer de représenter l'approche et plus spécifiquement les opérations de l'algorithme génétique, et pour simplifier les choses on va représenter que 4 services (cours, Avis de cours, design, statistique) donc chaque service représente une valeur dans le chromosome, et on va expliquer comment on a fait le crossover et la mutation.

Tous d'abord on fait la génération de la population initiale avec l'aide de fonction de random, donc on obtient plusieurs chromosomes (solutions) qui sont générés au hasard, après on fait une trie de cette population à l'aide de fonction de fitness, après on choisit les meilleurs et on fait un couplage entre eux avec le crossover enfin une mutation pour l'enfant obtenue la Figure 14 montre ça plus clairement, les services avec couleur bleu signifient que le service est activé et les services avec couleur orange signifient que le service est désactivé, le tableau vert c'est le chromosome.

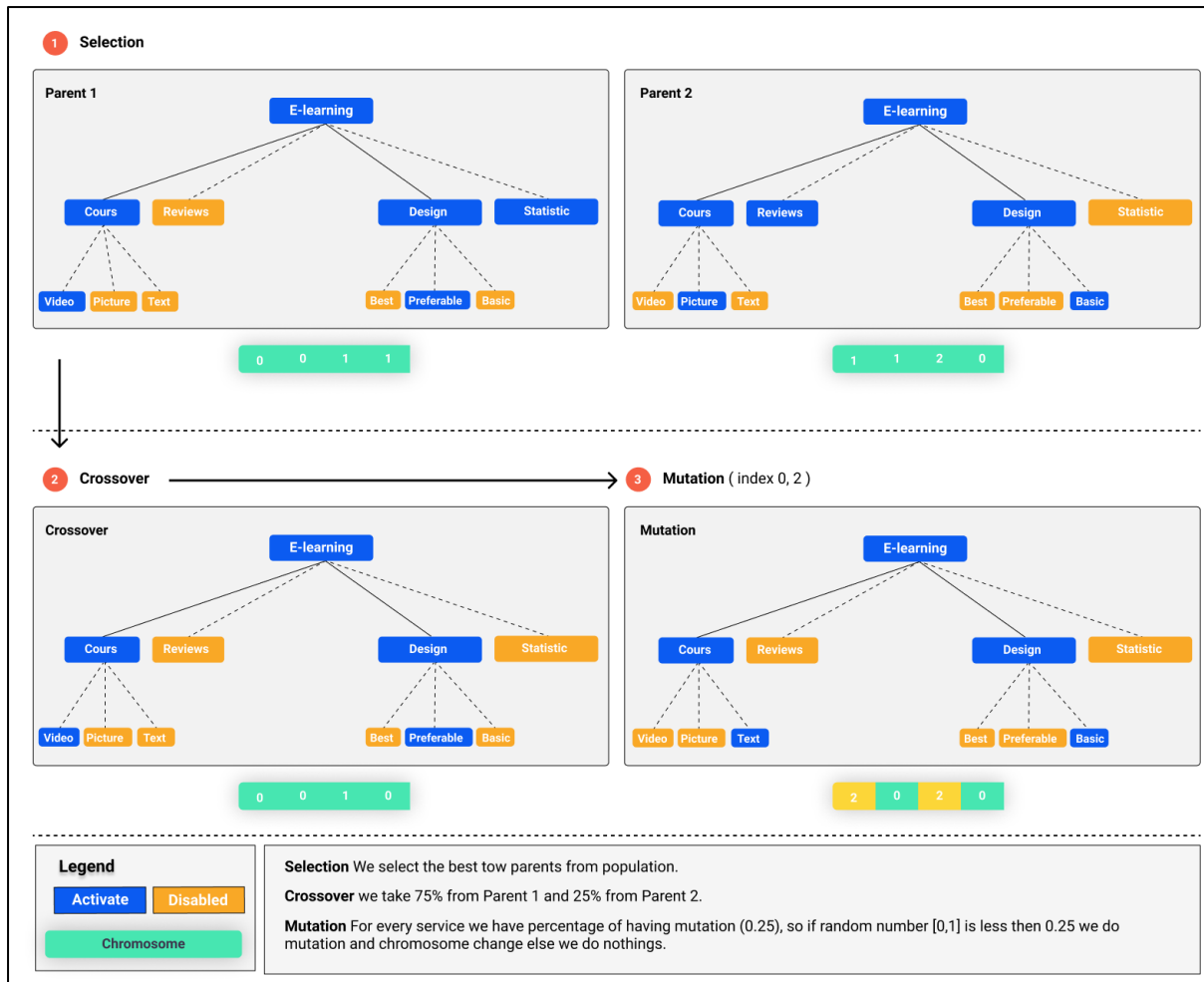


Figure 14 Operations génétique

On remarque dans la Figure 14 que dans le crossover il prendre 75% de parent 1 c'est-à-dire les 3 premier services de parent 1 et le quatrième service de parent 2 (25%), l'enfant résultat est comme suite :

- Cours avec Vidéo.
- Avis de cours désactivé.
- Design préférable.
- Statistique désactivé.

Après les services qui ils ont choisi au hasard pour la mutation sont Cours et Design, dans le chromosome sont les index 0 et 2 qui sont transformer comme suit :

- Cours de vidéo à texte.
- Design de Préférable à basic.

On fait ces opérations pour plusieurs chromosomes qu'on a sélectionné précédent, après on prend les chromosomes résultats pour la prochaine génération, et on répète ça jusqu'à l'arriver à la dernière génération (100). Et enfin on prend le meilleur depuis la dernière génération.

Conclusion

A travers ce chapitre nous avons présenté tout d'abord le cas d'étude sur lequel nous avons testé notre approche. Nous avons également détaillé la conception des deux parties de 10. conception de l'AG proposé c.à.d. le codage des chromosomes et la fonction de fitness auxquels nous avons accordé beaucoup d'importance afin d'assurer la convergence vers les solutions rapidement. Nous avons également présenté dans la troisième partie la conception de l'AG qui consiste à pratiquer l'évolution des générations, cette partie est la moins importante dans la phase de conception car elle peut changer après les tests. Le chapitre suivant sera donc consacré à exposer les tests ainsi que les résultats obtenus de nos expérimentations.

CHAPITRE 5 : Implémentation de l'application et Test

Introduction

Précédemment nous avons présenté la conception de notre plateforme E-learning ainsi que notre solution pour l'optimisation de la planification de l'auto adaptation et les mécanismes

1. utilisés pour avoir un bon résultat.

Dans ce chapitre nous allons présenter les outils et Language de programmation utilisée pour implémenter notre plateforme et notre approche, par la suite on va décrire les tests et les résultats obtenus.

Les outils et langages de programmation utilisés

2. Le choix des outils de travail pour notre projet de fin d'étude n'été pas facile vue la variété des technologies et outils existants, notre choix était fait par rapport aux nos objectifs et au même temps on a essayé d'utiliser les technologies les plus récente.

2.1. Application web mono page (single-page-application ou SPA)

Est une technique de création des application web accessible via une page web unique. Le but est d'éviter le chargement d'une nouvelle page à chaque action demandée, et de fluidifier ainsi l'expérience utilisateur. Donc l'ensemble des éléments de l'application est chargé (contenu, images, CSS et JavaScript) dans un unique fichier HTML.

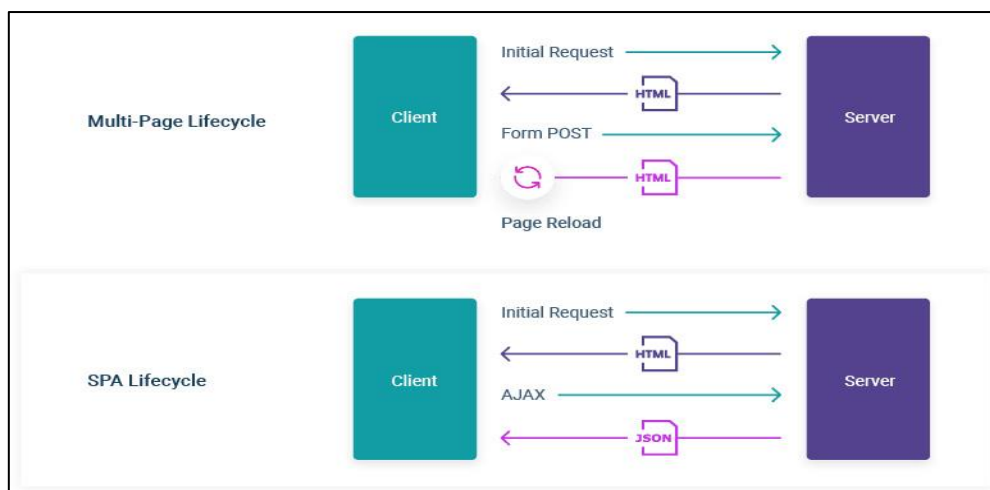


Figure 15 Application web mono page

Sublime Text

Est un éditeur de texte générique codé en C ++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités. Depuis la version 2.0, sortie le 26 juin 2012, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares, on a utilisé la version 3.

Laravel

Est un Framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

Html

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.

CSS

Les feuilles de style en cascade¹, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.

2.6.

Bootstrap

Est une collection d'outils utilisée à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web.

Javascript

Est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

Vue.js

Aussi appelé simplement Vue, est un Framework JavaScript open-source utilisé pour construire des interfaces utilisateur et des applications web mono page (SPA).

Présentation de notre plateforme

Dans notre projet de fin d'étude on a une plateforme E-learning qui implémente une approche d'adaptation, on va présenter premièrement les interfaces et les services (fonctionnalités) de notre plateforme, après on va présenter les interfaces de la simulation qu'on a fait pour tester

3. notre approche.

Présentation de plateforme

Dans notre plateforme on a plusieurs pages qui contient plusieurs services, on va essayer de présenter les pages essentiels et au même temp les services qu'on a implémentés.

3.1.

3.1.1. Interface d'authentification

La première chose que l'utilisateur de notre plateforme doit faire et l'authentification pour accéder aux services de plateforme (Figure 16).

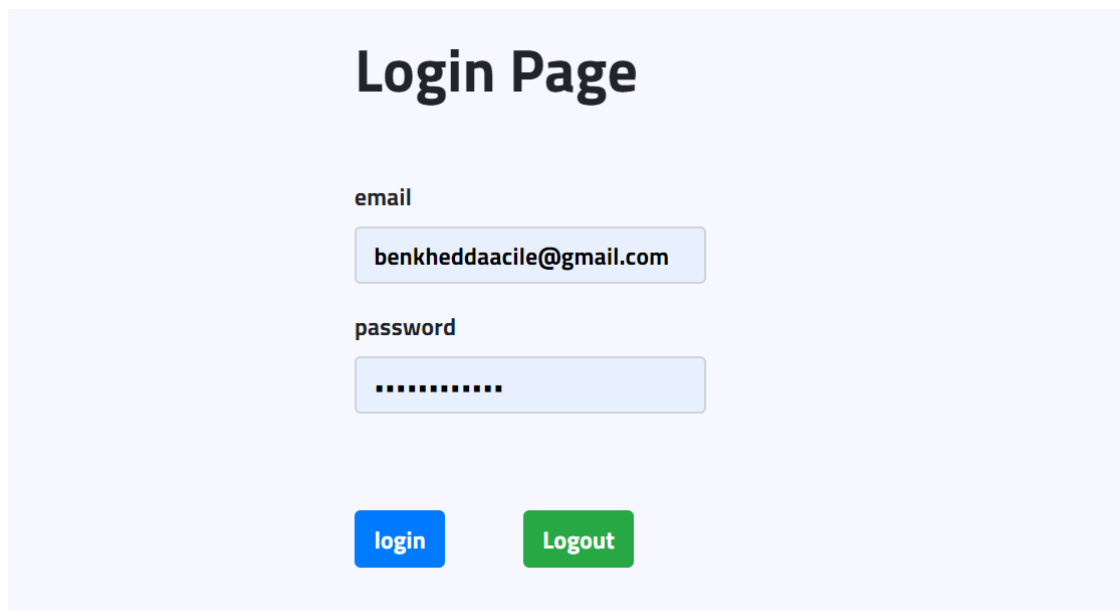


Figure 16 Page d'authentification

3.1.2. Interface d'accueil

Après si les données entrées dans l'authentification sont correctes, l'utilisateur accède au **Dashboard** qui contient les deux services **Statistique (My Classes)** et **Recommandation (Suggestion)**.

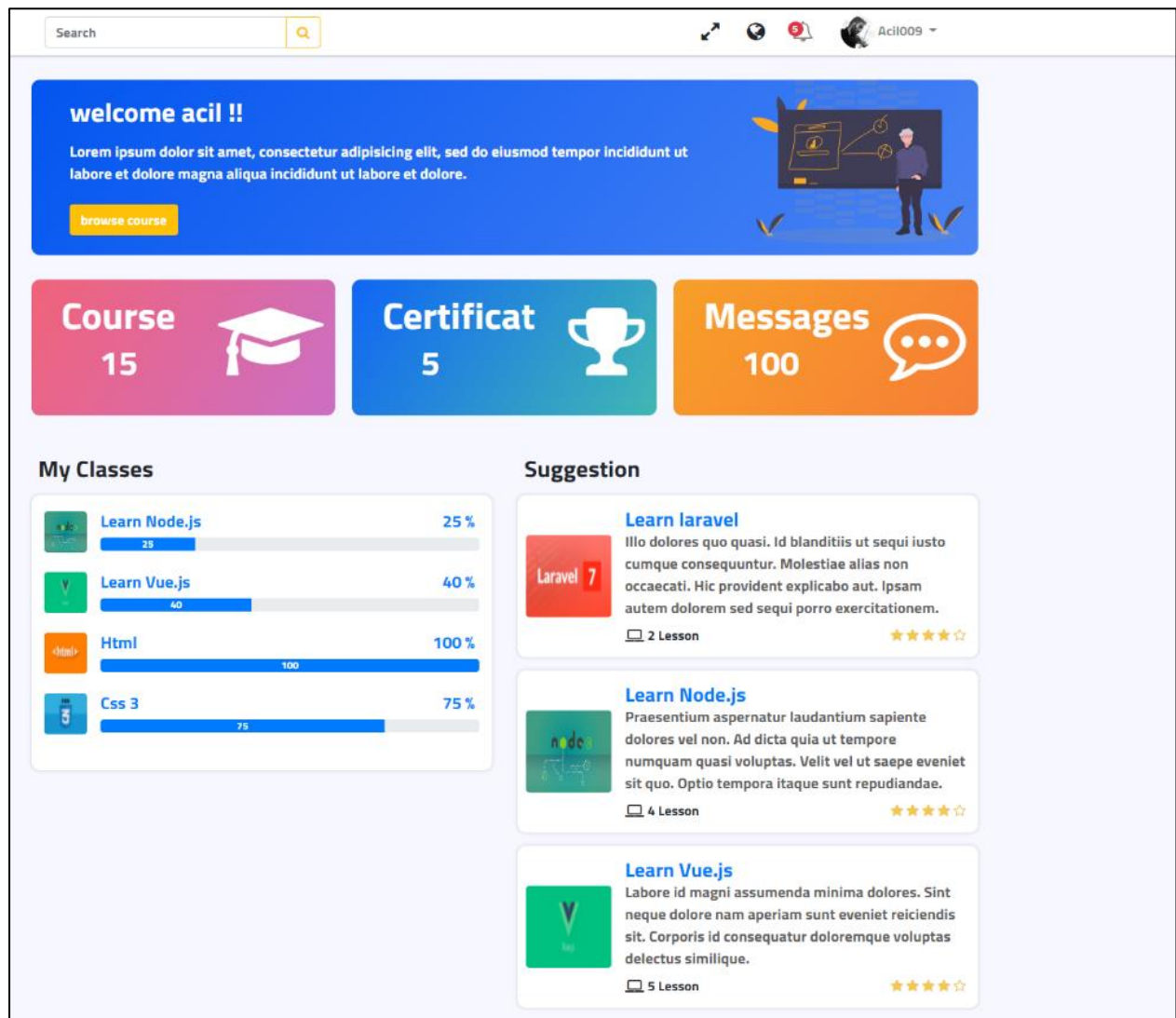


Figure 17 Page d'accueil

3.1.3. Interface d'explorer

L'interface **d'explorer** dans Figure 18, contient tous les cours existents dans notre plateforme, donc depuis cette interface l'utilisateur peut choisir un cours.

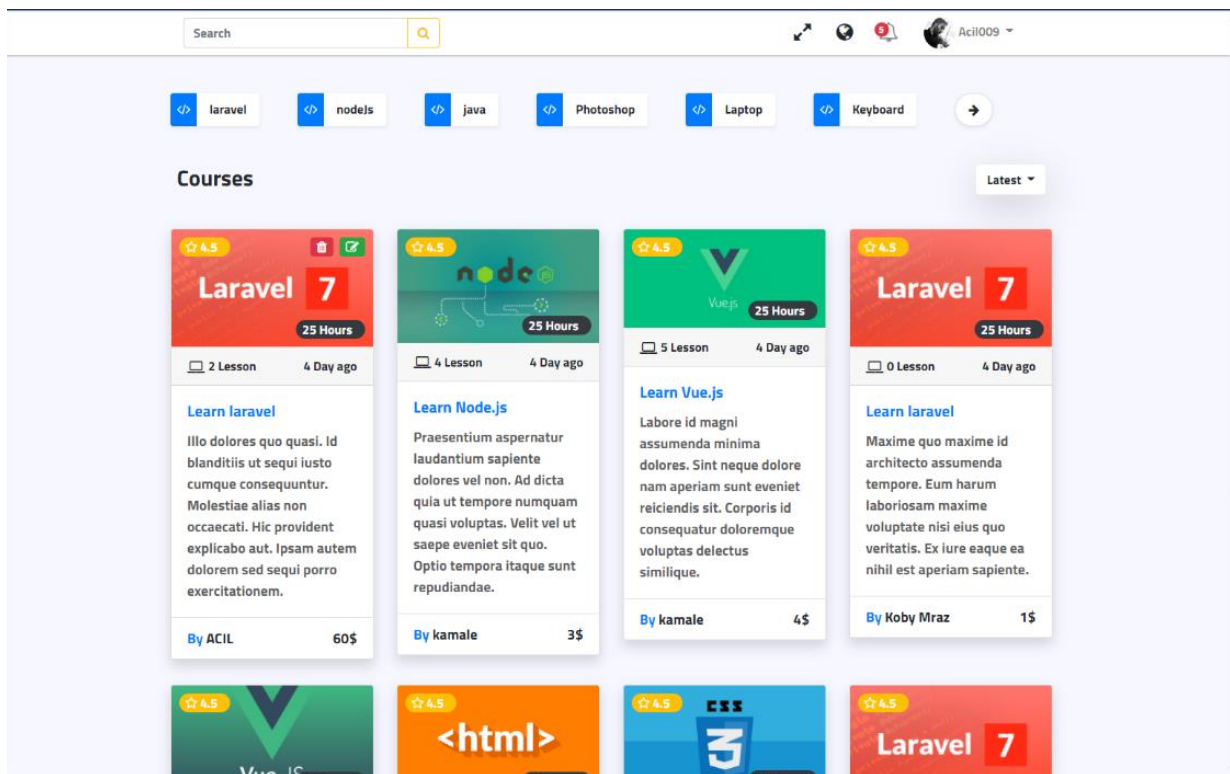


Figure 18 Page d'Explore

3.1.4. Interface de cours


Depuis l'interface d'**Explore** on peut accéder au page de cours, dans cette page on peut avoir les informations générales sur le cours comme : titre, nombre d'étudiant inscrits dans ce cours, langue, prix, nombre de Lesson et l'enseignant.

L'étudiant doit s'inscrire dans le cours (Enrolle course), pour accéder au contenu et donner un avis sur ce cours.

Dans cette interface on a le **service de cours (Vidéo, Image, Texte)** où l'apprenant peut étudier ce cours avec Vidéo ou Image ou Texte.

On peut voir tous sa dans Figure 19.


Et on a aussi dans cette page le **service d'avis de cours (Reviews)** (Figure 20).





Learn Node.js

Praesentium aspernatur laudantium sapiente dolores vel non. Ad dicta quia ut tempore numquam quasi voluptas. Velit vel ut saepe eveniet sit quo. Optio tempora itaque sunt repudiandae.


4.5



4 Lessons


25 Hours


0 Students

Price: 3 \$


English


7/2020 Last update

By kamale
Enrolle course

Text
Image
Video
Reviews

First

Html basic
✓

Html Advance
⊘

Html 5
⊘

outro
⊘

Figure 19 page de Lesson

Reviews

★
★
★
★
☆

4,0

★
★
★
★
★

100%

★
★
★
★
★

★
★
★
★
★

★
★
★
★
★

Your Review


★
★
★
★
☆

Good course

✓

Submit Review

Students Feedback



ACIL

4 days ago

★
★
★
★
★
☆

Good course

Figure 20 les avis de cours

3.1.5. Interface de « Mes cours »

Dans cette interface l'utilisateur peut :

- Créer un nouveau cours avec le bouton « créer new course ».
- Voir tous les cours qu'il a inscrits.
- Voir tous les cours qu'il a créés.

On peut voir cette interface dans la Figure 21.

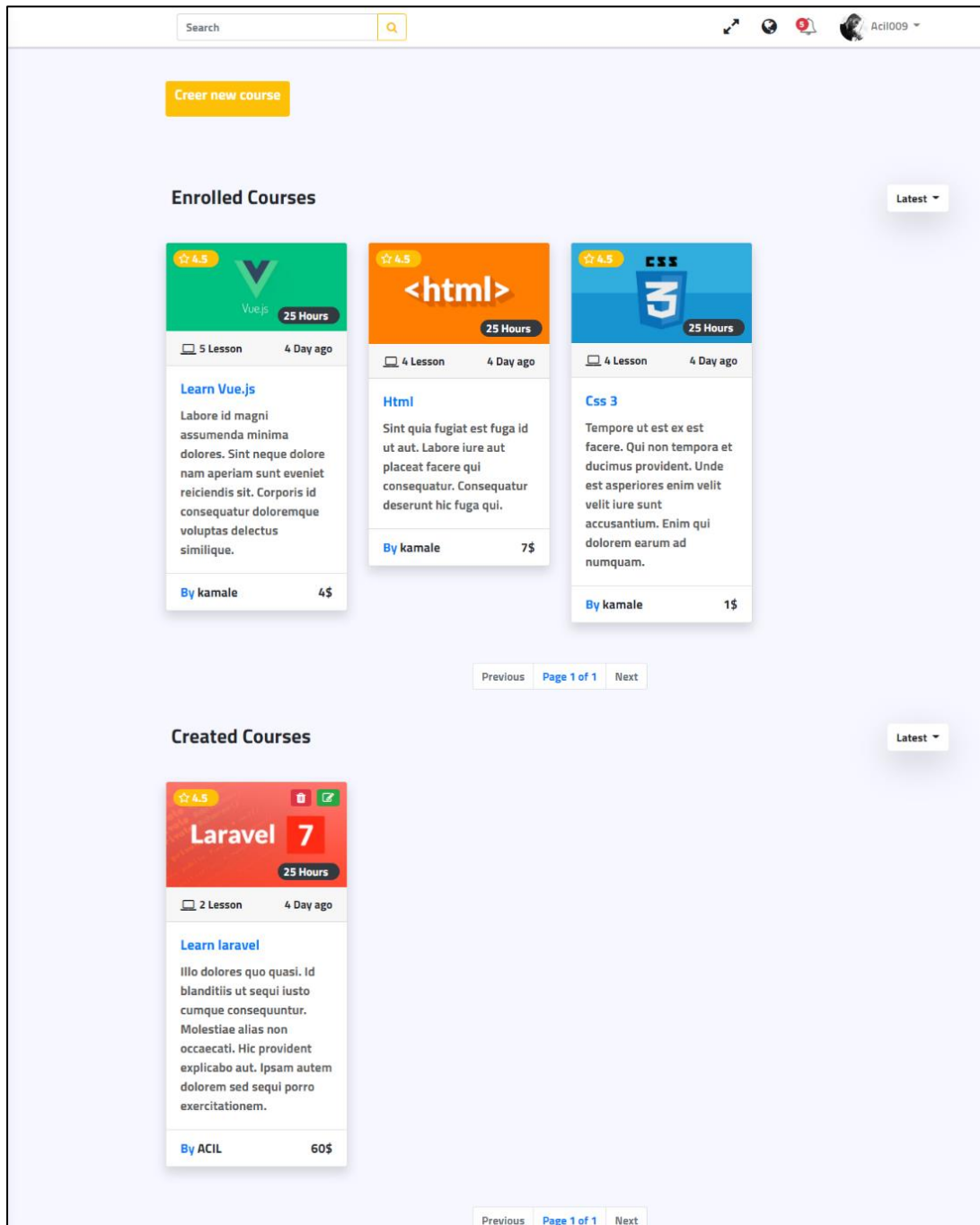


Figure 21 Page de Mes Cours

3.1.6. Interface de Réglage

Dans cette interface l'utilisateur peut changer leur nom, prénom, mot passe et leurs liens des médias sociaux (Figure 22)

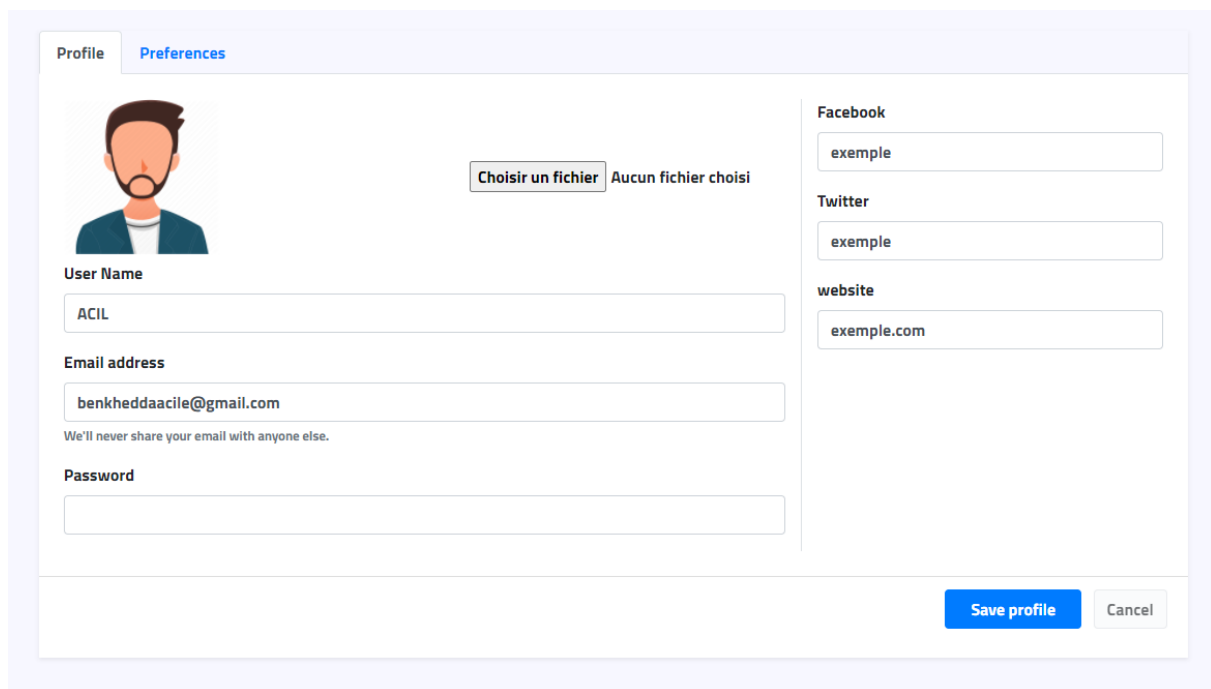


Figure 22 Paramètre général

Il peut aussi dans cette interface de changer les préférences de la plateforme (Figure 23).

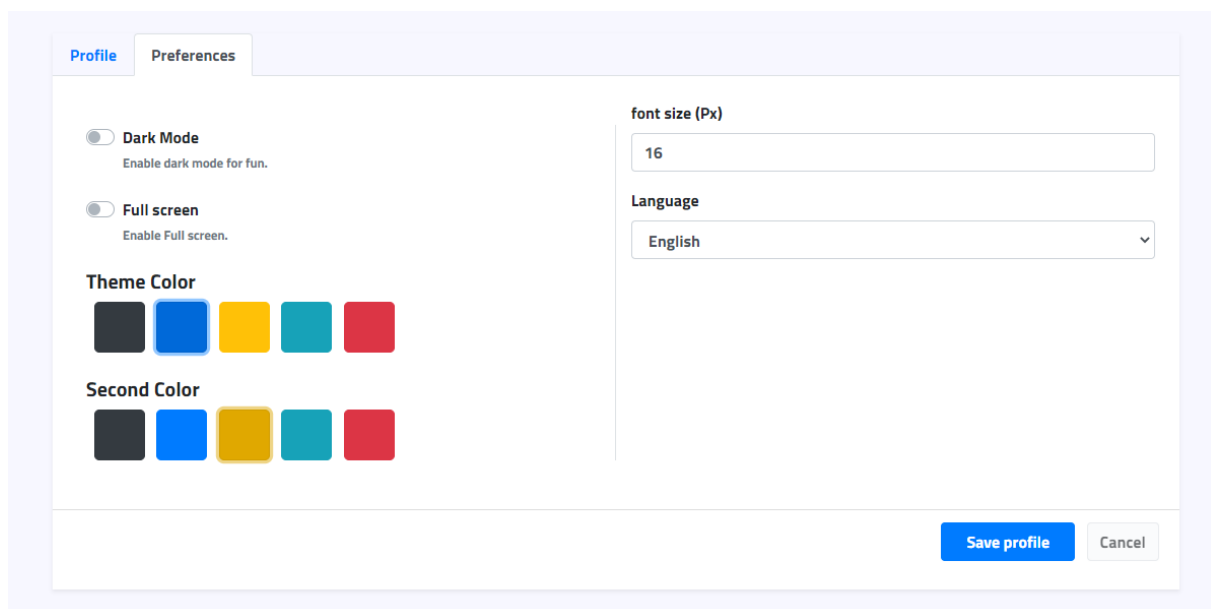


Figure 23 Préférences

3.1.7. Interface de service de navigation des liens

On a trois qualités ou on peut dire variations de service **navigation des liens** et on peut le représenter comme dans la Figure 24, les variations sont :

- Afficher tous les liens.
- Meta affichage des liens.
- Caché les liens.

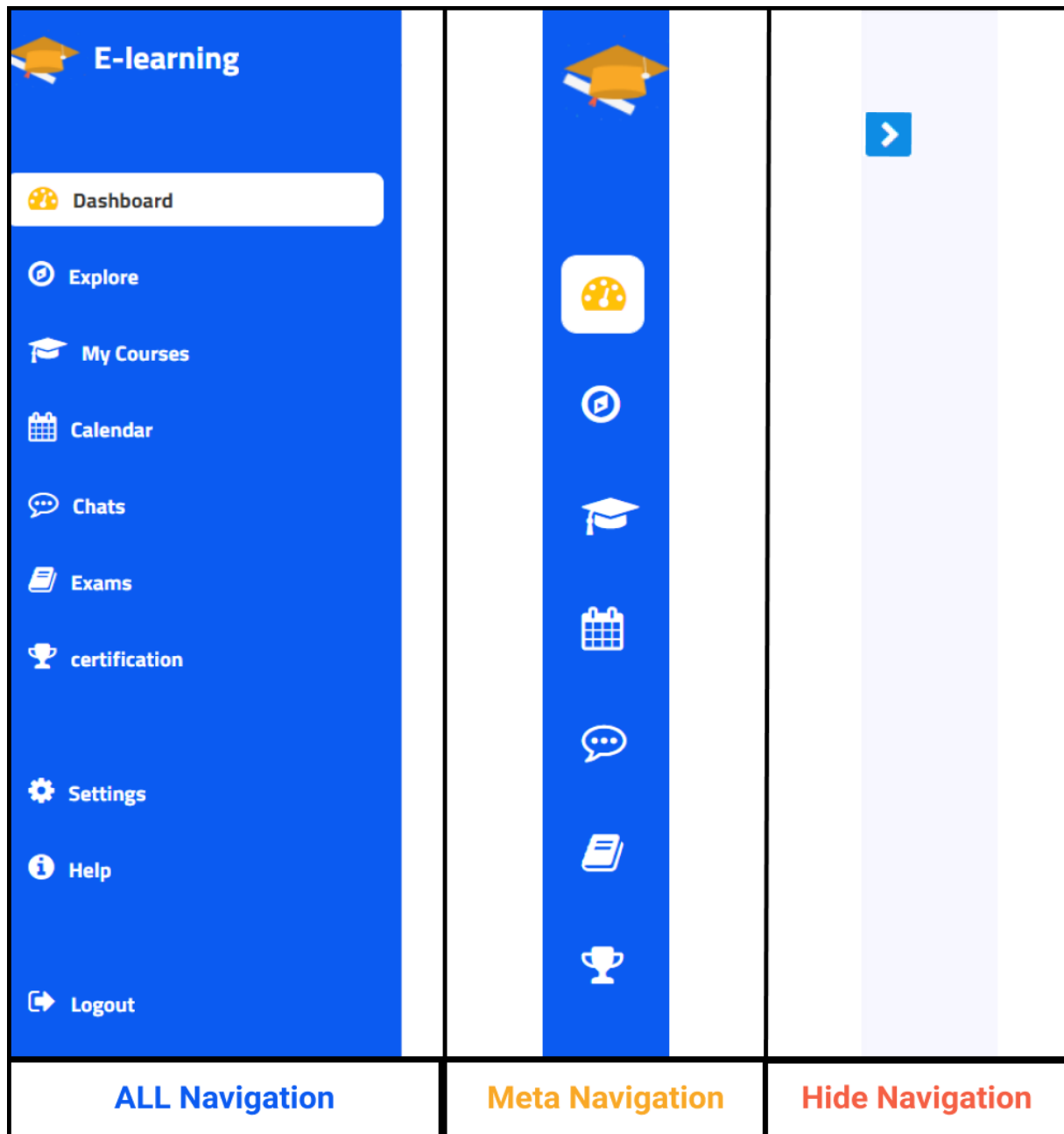


Figure 24 Navigation service

3.1.8. Interface de service de Design

On a trois qualités de service Design et on peut représenter comme dans la Figure 25, les trois variations sont :

- Meilleur design avec la couleur bleu.
- Préférence design avec la couleur vert.
- Basic design avec la couleur noir.

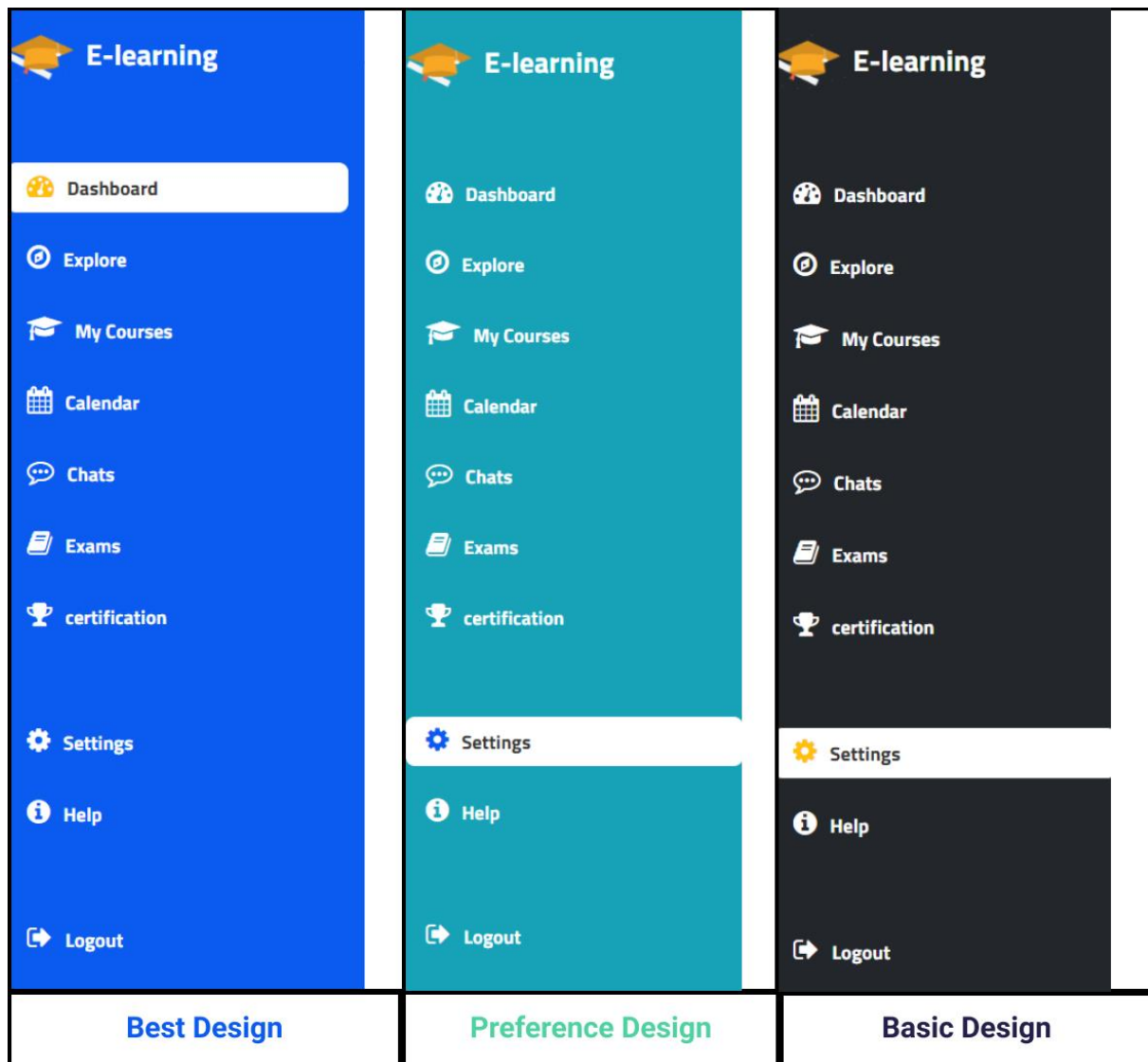


Figure 25 Les variations de design

3.1.9. Interface de création de cours

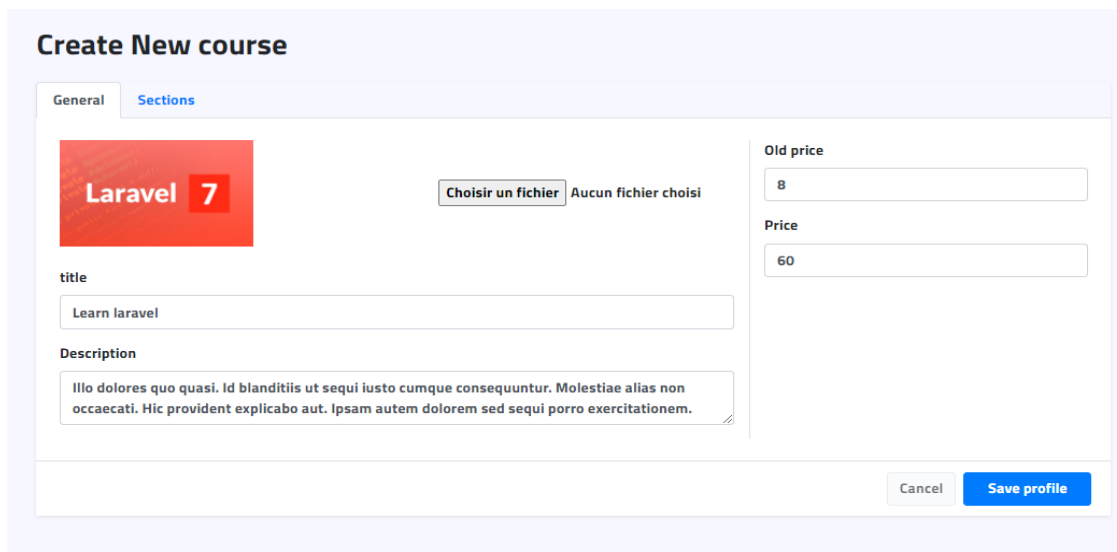


Figure 26 Création de cours

On a présenté les interfaces essentielles de notre plateforme, et les services qu'on a implémenter dans notre plateforme :

- Cours (Vidéo, Image, Texte).
- Avis de cours.
- Statistique.
- Recommandation.
- Police.
- Navigations.
- Design files

Donc lorsque l'approche est activée elle :

- 3.2.
 - Soit changer la qualité de certain services (Cours, Navigation, Design files).
 - Soit désactiver ou activer autres services (Avis de cours, Statistique, Recommandation).

Présentation de la simulation du l'approche

La simulation est un test qui fait par l'utilisateur pour savoir la validité de l'application par l'obtention des graphes comme résultats, Dans la simulation de notre approche on a trois parties essentielles, les paramétrés d'algorithmme, la simulation et les graphes.

3.2.1. L'interface de paramétrés d'algorithmme

Afin d'examiner le comportement de notre algorithmme, et examiner la convergence de la fonction de fitness, nous devons passer par l'étape de la configuration pour choisir les

paramètres de simulation de l'AG. Notre application permet la configuration à partir de l'interface d'AG présenté dans la Figure 27.

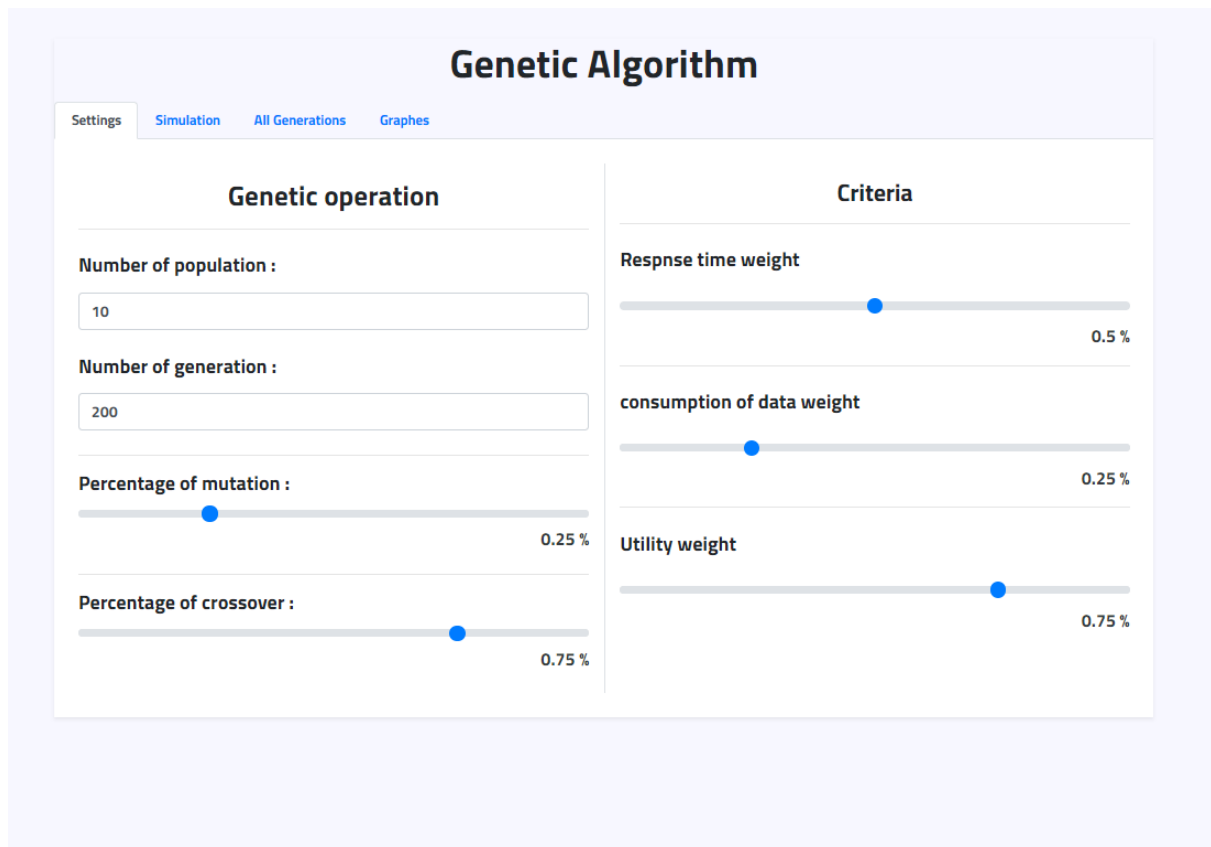


Figure 27 Paramétrés de simulation

3.2.2. L'interface de simulation

Dans cette interface on peut choisi la limite de temps de réponse de notre plateforme, et la latence (Temps de réponse de connexion d'utilisateur), et on fait notre simulation, Les résultats aussi sont exposer dans cette interface « Adaptation plan », avec des informations sur le plan d'adaptation optimal comme :

- Temps de réponse de plateforme avec le plan optimal.
- Consommation de connexion avec le plan optimal.
- Utilité de plan optimal.
- Le score de fonction de fitness de ce plan.
- Représentation de chromosome.

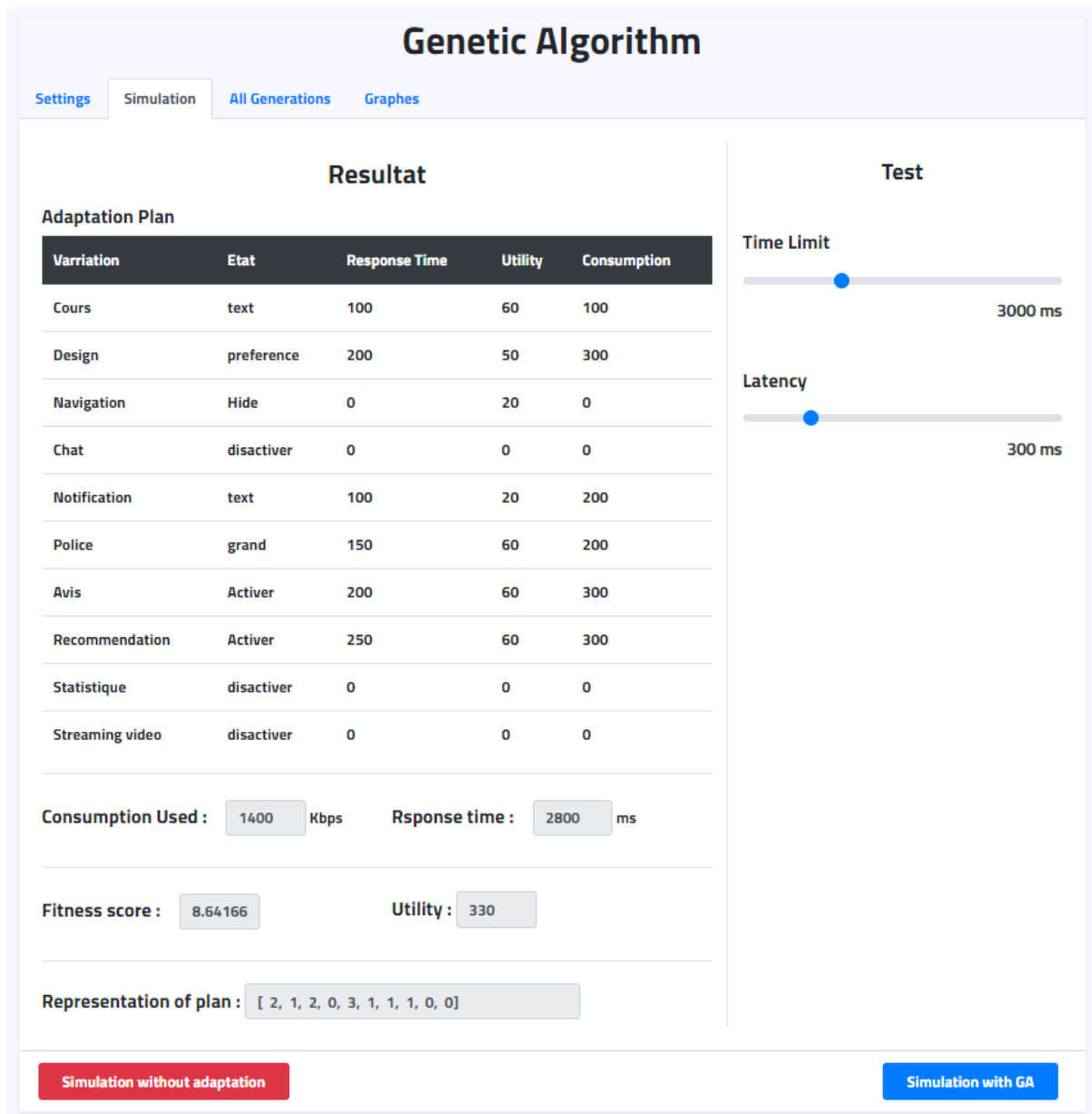


Figure 28 Simulation de GA

On peut aussi voir le meilleur plan pour chaque génération dans notre simulation (Figure 29)

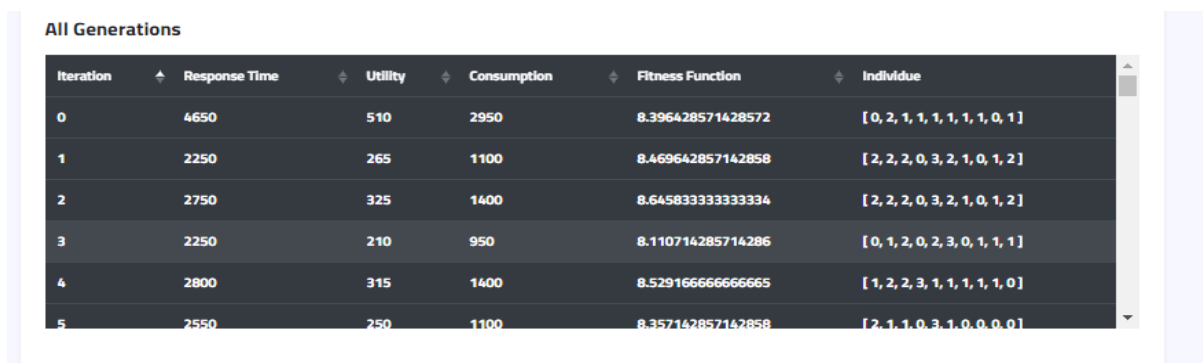
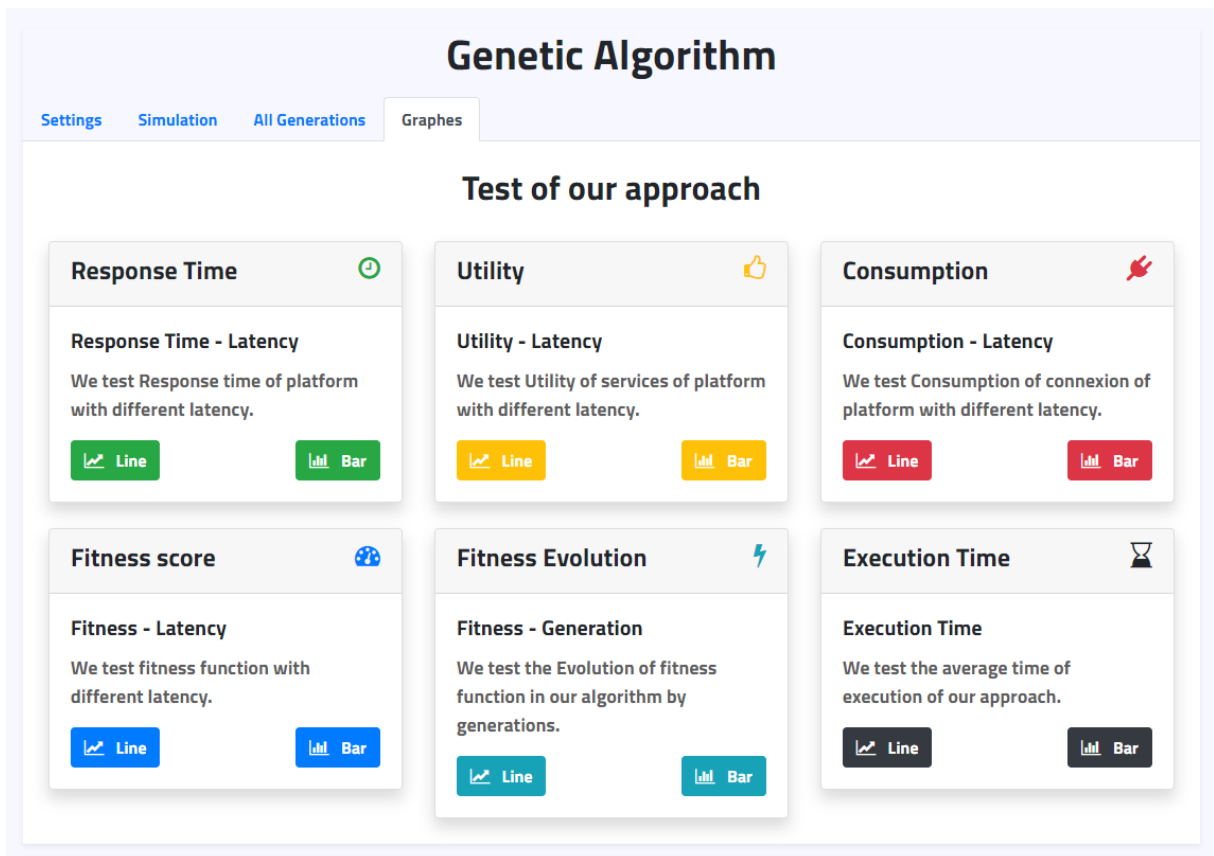


Figure 29 Journal de simulation

3.2.3. Interface des courbes graphiques

Notre Approche permet de générer des graphes depuis six interfaces des courbes graphiques à l'aide de la bibliothèque **Chart.js**, ces courbes permettent de voir les caractéristiques et les résultats de notre approche comme il est montré dans la Figure 30.

Pour chaque test on peut voir les graphes en deux formes : **ligne** et **barre**.



4.

Figure 30 Interface des courbes graphiques

Expérimentations et résultats

Cette section présente les résultats des essais expérimentaux que nous avons effectués sur le cas d'étude E-learning, en appliquant deux approches différentes qui sont : approche sans adaptation ; approche basée sur l'algorithme génétique. Les graphes sont effectués à l'aide de la bibliothèque **Chart.js**. Ces courbes permettent de voir les caractéristiques et les résultats d'une ou plusieurs simulations.

Les tests ont été effectués sur un ordinateur portable Toshiba sous Windows 7 professionnel, avec un processeur Intel(R) Core (TM) i3 CPU M380 @2.53GHz et une RAM de 6.00 Go

Dans les tests on va essayer de tester notre approche par rapport au critère qu'on a fixé précédemment (Temps de réponse, Utilité et consommation de connexion) et voire l'efficacité de l'utilisation de plateforme avec et sans notre approche, Nous avons fait les expériences suivantes :

- Paramètres d'algorithme génétique.
- Temps de réponse de notre de plateforme avec et sans notre approche.
- Consommation des données de connexion avec et sans notre approche.
- L'utilité Globales des services activés dans la plateforme avec et sans notre approche.
- Score de fonction de fitness avec et sans notre approche.
- L'évolution de fonction de fitness dans l'algorithme génétique.
- Le temps moyen d'exécution de notre approche.

Ce tableau illustre les paramètres que nous avons utilisés dans les simulations Pour l'algorithme génétique :

Tableau 15 Paramètres de l'algorithme génétique

<i>Paramètre</i>	<i>Valeur</i>
Taille population	10
Nombre de génération	100
Pourcentage de croisement	0.75
Probabilité de mutation	0.25
Temps de réponse maximal	3000
Poids de temps de réponse	0.5
Poids d'utilité	0.25
Poids de consommation	0.25

4.1.

Paramètres d'algorithme génétique

Dans cette partie on va essayer de justifier le choix des valeurs des paramètres d'AG.

Par rapport au :

- Poids des critères (temps de réponse, utilité, consommation) on a expliqué précédemment pourquoi on a donné ces valeurs (dans la section de Fonction de fitness).

- La mutation on a choisie 0.25 pour ne pas rester dans les mêmes résultats, et au même pour ne pas éliminer le principe « d'évolution depuis les générations précédentes pour atteindre à la solution finale » car avec une valeur proche de 1 on va toujours faire une mutation qui implique toujours des nouveaux individus et puis nous ne prendrons rien depuis la génération précédente.
- Temps de réponse maximal est 3000ms car il est recommandé par Google ⁵.

On va justifier la taille de population et le nombre de génération dans les sections suivantes.

4.1.1. Taille de population

Dans ce test on va vérifier que le paramètre la taille de population (10) d'algorithme génétique est bien définie, et pour faire ça on va calculer le temps d'exécution de notre algorithme et la fonction de fitness de plan résultat d'AG avec plusieurs nombres de population.

Tableau 16 Test de nombre de population

Taille de population	Fonction de fitness	Temps moyen d'exécution(ms)
5	5.66	30
10	8.60	70
30	8.64	100
60	8.58	150
100	8.67	250
200	8.66	500

On peut voir que la taille de population n'affecte pas beaucoup le résultat d'algorithme génétique après la taille 10, car les valeurs de fonction de fitness des résultats d'AG sont très proche après la taille 10 (10, 30, 60, 100, 200), mais par rapport au temps de réponse d'AG on peut voir clairement la différence, par exemple lorsque on a une population avec une taille de 10 le temps d'exécution est 70ms qui est moins d'une seconde, mais avec une taille de 200 le temps d'exécution est 500ms qui est 5 second, et pour nous le temps d'exécution de notre approche est très important et on veut le minimiser le maximum possible.

Donc puisque le nombre de population n'affecte pas beaucoup les résultats d'AG après la taille 10 et au même temps affecte beaucoup le temps d'exécution de notre approche, on peut dire qu'une population avec une taille de 10 est très approprié.

⁵ <https://www.stanventures.com/blog/google-recommended-page-load-time/>

4.1.2. Nombre de génération

Dans ce test on va vérifier que le paramètre de nombre de génération d'algorithme génétique (100) est bien définie, et pour faire ça on va calculer la fonction de fitness pour chaque 10 génération dans 100 génération.

La Figure 31 au-dessous représente l'évolution de fonction de fitness par rapport au nombre de génération.

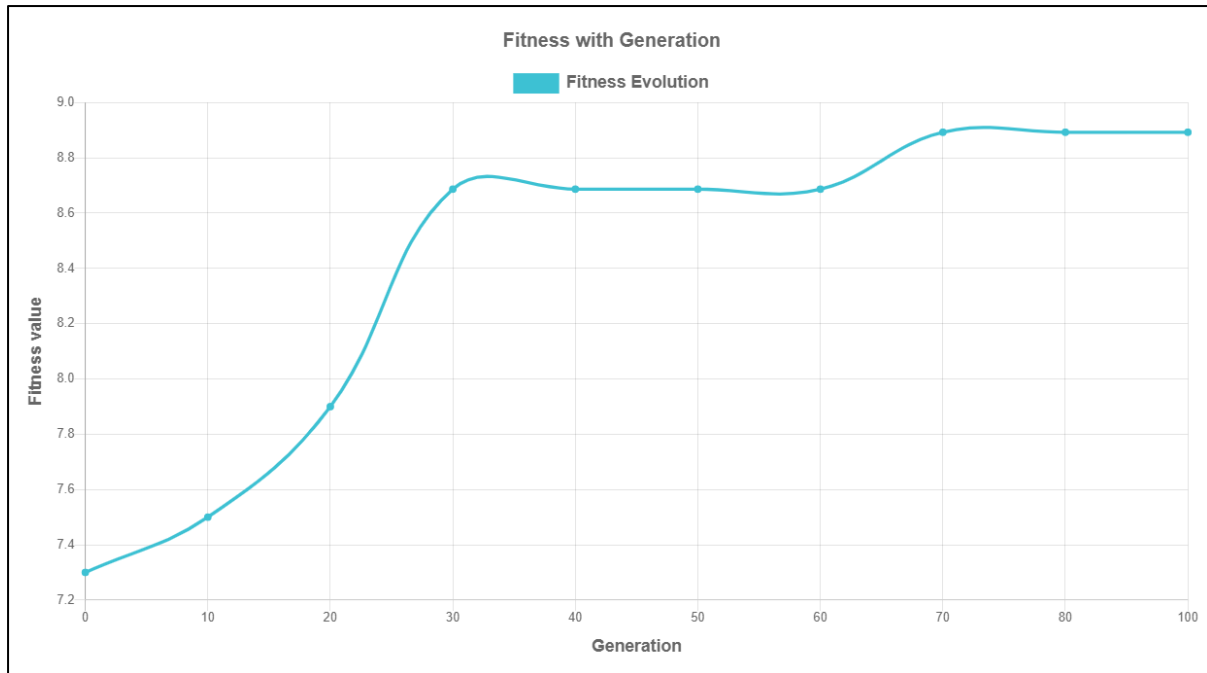


Figure 31 Evolution de fonction de fitness

4.2. Ce test montre que le nombre de génération approprié pour notre approche est 100, car si on met moins de 100 on trouve que jusqu'à 75 il n'est pas encore arrivé au meilleur plan, donc pour assurer que notre approche arrive au meilleur plan 100 génération est très approprié.

Temps de réponse

Le temps de réponse c'est le critère le plus important dans notre approche, on a fixé une limite de temps de réponse (3000 ms) que notre plateforme ne doit pas dépasser avec notre approche, on va tester le temps de réponse dans la plateforme par rapport à la latence d'utilisateurs.

La Figure 32 au-dessous contient les graphes de temps de réponse de notre plateforme avec et sans notre approche par rapport à la latence (ms) de l'utilisateur, le trait avec la couleur verte représente le temps de réponse avec notre approche et le trait noire sans notre approche.

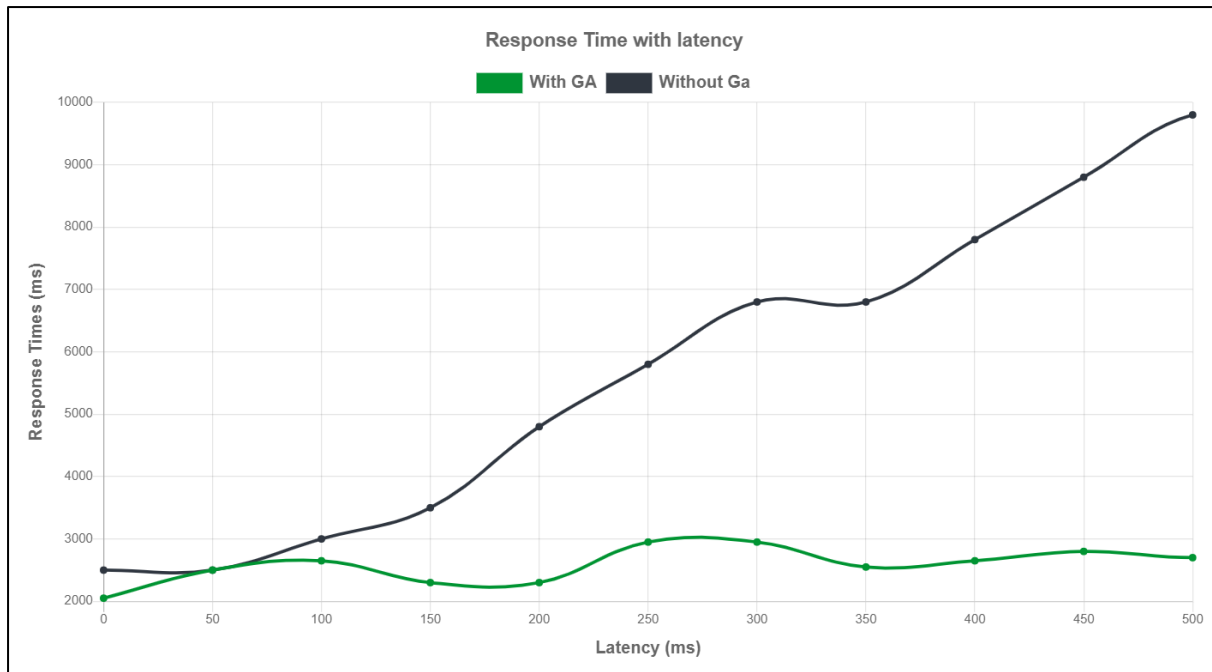


Figure 32 Test de Temps de réponse

On peut voir que sans notre approche (trait noire) dans La Figure 32, montre clairement que sans adaptation il y'a eu un très grand dépassement par rapport au temps de réponse de plateforme qui normalement ne doit pas dépasser le seuil maximal (3000 ms) où il est arrivé presque au 10000ms (10 seconds) ce qui est très mauvais.

Mais avec notre approche (trait vert) dans La Figure 32, on peut voir que le temps de réponse de plateforme ne dépasse pas 3000ms (3 seconds) ce qui est considéré comme un très bon temps de réponse pour un utilisateur.

Par exemple dans latence = 500 ms, le temps de réponse sans AG = 9800ms et avec AG = 2800ms, donc on a diminué le temps de réponse par 7 secondes avec notre approche.

Le résultat obtenu est satisfaisant où le temps de réponse est raisonnable et ne dépasse pas le seuil max (3000 ms) avec notre approche, ce qui répond parfaitement aux exigences du
4.3. plateforme. Nous pouvons conclure que notre approche a bien aidé la plateforme s'adapter aux faibles vitesses de connexion d'utilisateur.

Consommation de connexion

Notre espérance est qu'avec notre approche la plateforme consomme moins de données de connexion, c'est pour ça on va tester la consommation de données de connexions avec et sans notre approche par rapport à la latence d'utilisateur.

La Figure 33 au-dessous contient les graphes de consommation de données de connexion (Kbps) de notre plateforme avec et sans notre approche par rapport à la latence (ms) de l'utilisateur, le trait avec la couleur rose représente la consommation avec notre approche et le trait noire sans notre approche.

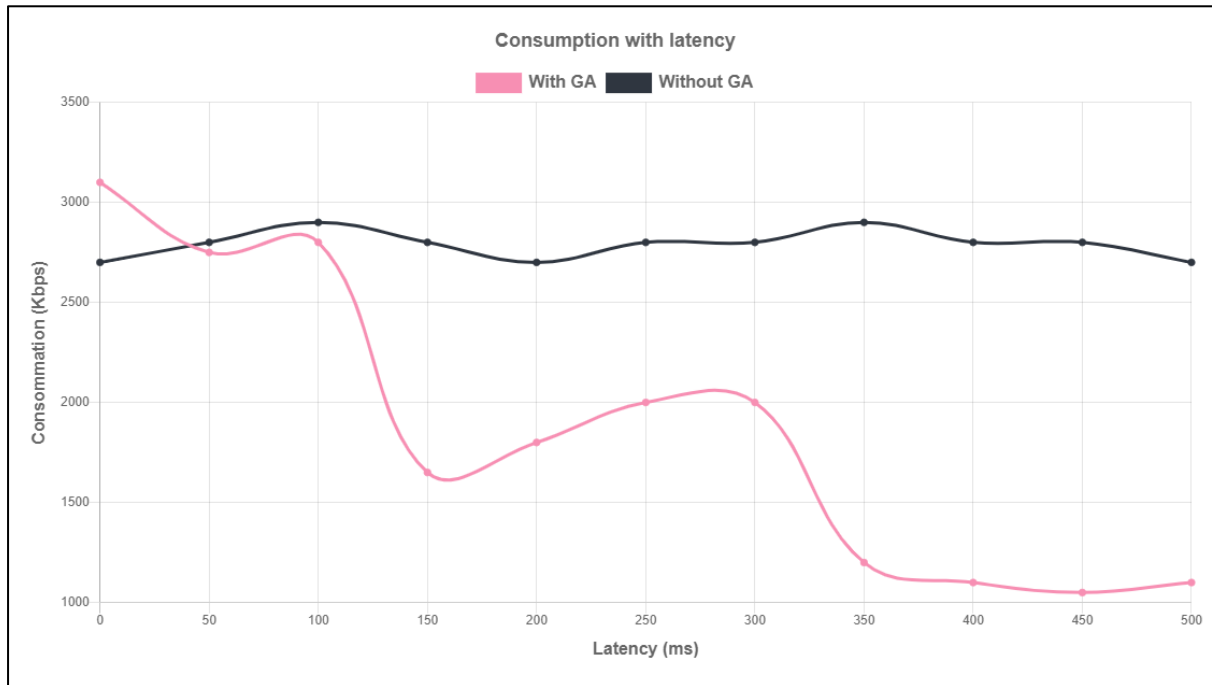


Figure 33 Test de Consommation de connexion

On peut voir que sans notre approche (trait noire) dans La Figure 33, que la consommation de connexion est grande quel que soit la vitesse de connexion d'utilisateur (latence).

Mais avec notre approche (trait rose) dans Figure 33, la consommation de connexion diminue avec l'augmentation de latence, donc plus la connexion est faible plus notre plateforme consomme moins de données de connexion avec notre approche,

4.4. Le résultat obtenu montre le succès de notre approche de diminuer la consommation de données de connexion dans les faibles vitesses de connexion d'utilisateur.

Utilité

Dans notre approche on veut maximiser l'utilité des services, Mais cela se heurte directement avec le temps de réponse car à mesure que l'utilité augmente elle augmente avec lui aussi le temps de réponse de plateforme est c'est ce que nous ne voulons pas car le poids de temps de réponse (0.5) qu'on a fixé précédemment et plus grand que le poids d'utilité (0.25), donc pour

nous le temps de réponse est plus important que l'utilité, On va tester l'utilité des services activé dans la plateforme avec et sans notre approche par rapport à la latence d'utilisateur.

La Figure 34 au-dessous contient les graphes de l'utilité des services de notre plateforme avec et sans notre approche par rapport à la latence (ms) de l'utilisateur, le trait avec la couleur jaune représente l'utilité avec notre approche et le trait noire sans notre approche.

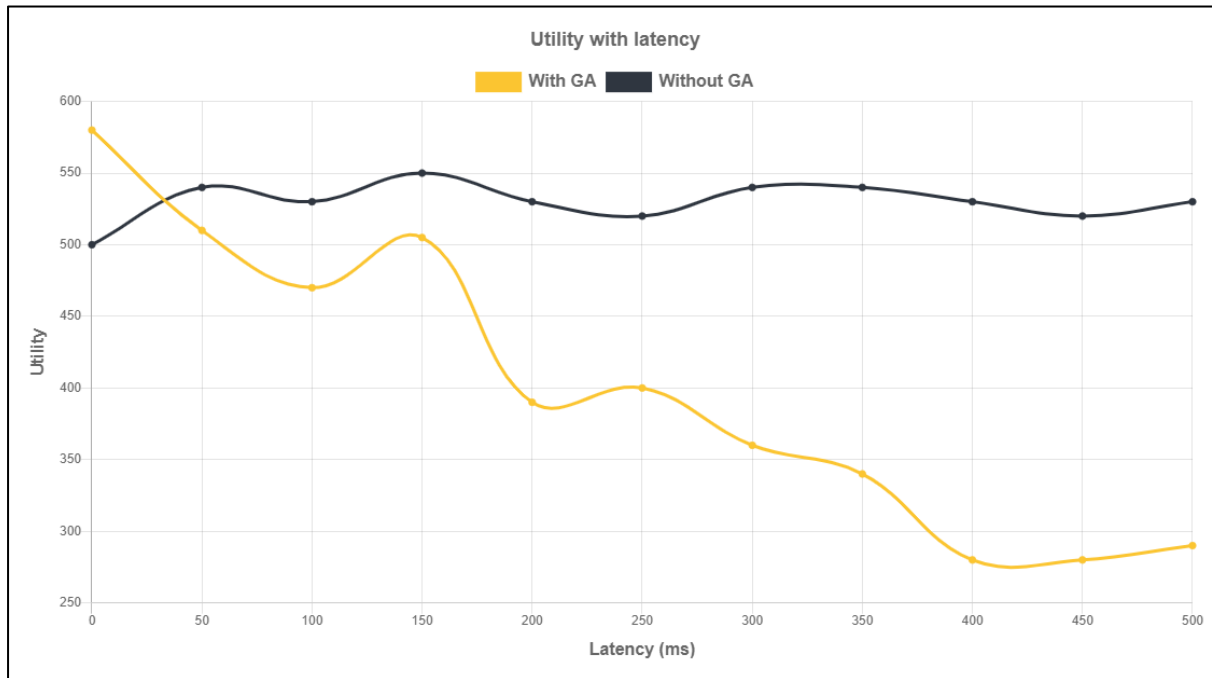


Figure 34 Test de l'Utilité

On peut voir que sans notre approche (trait noire) dans La Figure 34, l'utilité est grande (environ 550), car l'utilisateur peut choisir les services avec la qualité qu'il veut (malgré que ça affecte le temps de réponse).

Mais avec notre approche (ligne jaune) de la figure 31, nous voyons que notre approche essaie de garder l'utilité bien en cas de connexions faibles et elle n'a pas négligée l'utilité et juste se soucier au temps de réponse, par exemple dans latence = 350 on peut voir que l'utilité = 330 est c'est une bonne utilité par rapport aux cette latence.

Nous pouvons conclure après ce test que malgré que le temps de réponse et plus important que l'utilité mais notre approche n'a pas négligé l'utilité et elle a fait un bon travaille de garder l'utilité bien dans des connexions faibles.

Temps d'exécution d'approche

Le temps d'exécution de notre approche est très important, est-on ne veut pas que l'exécution de notre approche prendre beaucoup de temps qui sera ajouter au temps de réponse car fondamentalement notre objectif est de diminuer le temps de réponse.

- 4.5. Dans ce test on va voir le temps moyen qu'on a besoin pour exécuter notre approche, donc on va faire plusieurs exécutions de notre approche est voir le temps d'exécution de chacun d'eux.

On peut voir ça dans la Figure 35 au-dessous.

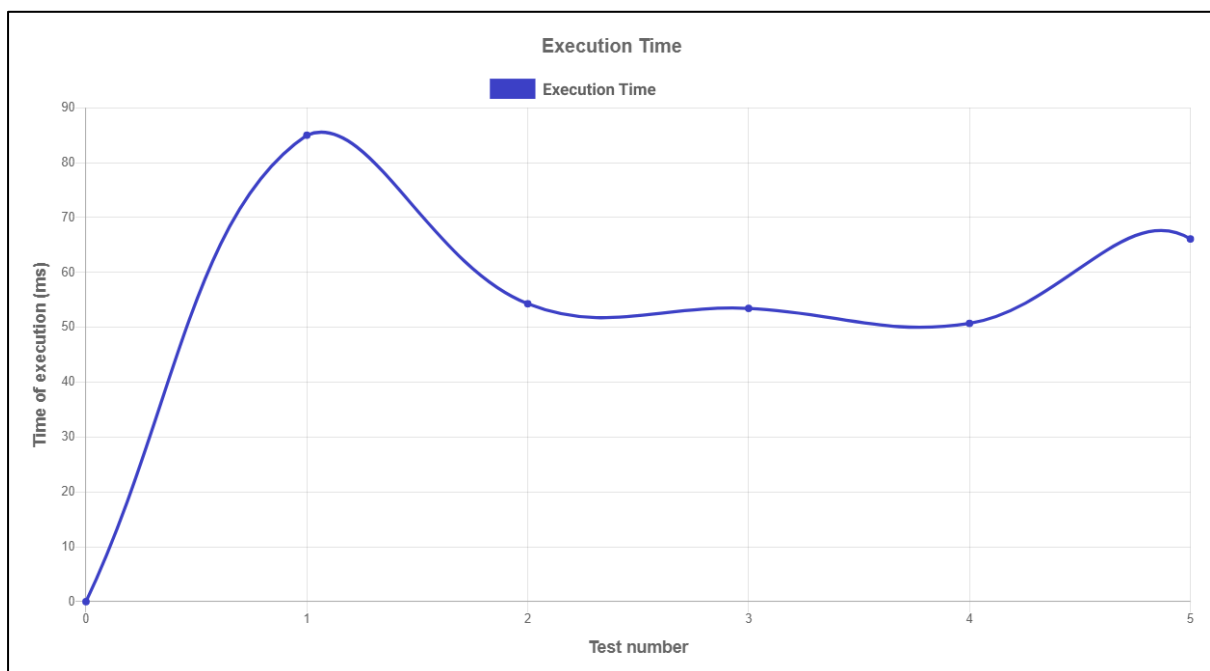


Figure 35 Temps d'exécution

Le test précédent montre que le temps moyen d'exécution de notre approche est 70ms c'est-à-dire moins d'une seconde ce qui est très acceptable.

Les résultats obtenus après plusieurs tests montrent que notre approche donne des bonnes résultats, en effet, le temps de réponse et la consommation de connexion avec notre approche sont optimisés et l'effet de diminution de ces deux critères est très claire et efficace et au même temps sans la négligence de la critère d'utilité, De plus, le temps d'exécution de notre approche càd le temps nécessaire à notre approche pour trouver le bon plan d'adaptation est très

acceptable (moyenne de 70ms). Nous pouvons conclure que notre approche s'apprête bien pour les problèmes de connexion dans les plateformes E-learning.

Conclusion

Le présent chapitre constitue la dernière étape pour présenter notre projet dans ce mémoire, il a été dédié à la présentation de l'application développée ainsi que l'environnement et les outils de travail utilisés. Une grande partie du chapitre a été consacrée aux différents tests effectués ainsi que les résultats obtenus qui montre bien que notre approche basée sur AG s'accommode très bien à l'auto adaptation des plateformes par rapport aux vitesses de connexion d'utilisateur.

Conclusion générale

Le projet que nous avons présenté dans ce mémoire consiste à la conception et le développement d'une plateforme E-learning dotée de la fonctionnalité d'auto-adaptation.

En effet, l'approche proposée permet d'adapter l'utilisation de différentes fonctionnalités de plateforme E-learning à un instant donné avec la vitesse de connexion d'utilisateur, pour se faire l'approche se base sur une technique d'optimisation basée sur les métaheuristiques (algorithme génétique).

Lors de la réalisation de ce projet, nous avons commencé par une étude bibliographique sur les systèmes auto adaptatifs. Par la suite, nous avons mené une étude sur les métaheuristiques en se basant principalement sur AG que nous avons utilisé dans notre approche.

Suite à cela, nous avons présenté la conception et l'implémentation de l'approche adoptée et les solutions proposées de l'auto adaptation basé sur les algorithmes génétiques.

Finalement, nous avons présenté les tests et les validations de l'approche proposée.

Les résultats obtenus sont très satisfaisants en termes de temps de réponse de plateforme et de consommation de données de connexion, en effet, dans les faible connexion notre approche a réussi du diminuer le temps de réponse plus de 7 seconds ce qui est très significatif, et en plus de ça notre approche nécessite seulement 70 ms pour trouver le bon plan d'adaptation ce qui est très acceptable, on a vu aussi que notre approche ne détériorer pas la performance de système car l'utilité reste proche de la valeur optimale, enfin on peut dire que notre approche s'apprête bien pour les problèmes de faible connexion et peut très bien fonctionner dans les utilisateurs ayant des problèmes de connexion.

En perspective, nous recommandons de traiter les thématiques suivantes :

1. Poursuivre le développement de notre approche d'adaptation des systèmes en utilisant d'autres algorithmes d'intelligent artificiel comme réseaux de neurones.
2. Généraliser notre approche à d'autres problèmes réels.

Bibliographie

- [1] Y. Brun *et al.*, « Engineering self-adaptive systems through feedback loops », dans *Software engineering for self-adaptive systems*, Springer, 2009, p. 48-70.
- [2] M. Ravindranathan et R. Leitch, « Heterogeneous intelligent control systems », *IEE Proceedings-Control Theory Appl.*, vol. 145, n° 6, p. 551-558, 1998.
- [3] M. Naqvi, « Claims and supporting evidence for self-adaptive systems—A literature review ». 2012.
- [4] M. Salehie et L. Tahvildari, « Towards a goal-driven approach to action selection in self-adaptive software », *Softw. Pract. Exp.*, vol. 42, n° 2, p. 211-233, 2012.
- [5] S. Lemouzy, « Systèmes interactifs auto-adaptatifs par systèmes multi-agents auto-organiseurs: application à la personnalisation de l'accès à l'information ». Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011.
- [6] F. Chauvel, « Méthodes et outils pour la conception de systèmes logiciels auto-adaptatifs ». 2008.
- [7] J. O. Kephart et D. M. Chess, « The vision of autonomic computing », *Computer (Long Beach, Calif.)*, vol. 36, n° 1, p. 41-50, 2003.
- [8] L. Sabatucci, V. Seidita, et M. Cossentino, « The four types of self-adaptive systems: a metamodel », dans *International Conference on Intelligent Interactive Multimedia Systems and Services*, 2018, p. 440-450.
- [9] R. El Ballouli, « Modeling self-configuration in Architecture-based self-adaptive systems ». 2019.
- [10] E. Rutten, N. Marchand, et D. Simon, « Feedback control as MAPE-K loop in autonomic computing », dans *Software Engineering for Self-Adaptive Systems III. Assurances*, Springer, 2017, p. 349-373.
- [11] S. Mirjalili, « Evolutionary algorithms and neural networks », *Stud. Comput. Intell.*, 2019.
- [12] R. D. Lemos, « Software engineering for self-adaptive systems: A second research roadmap (draft version of may 20, 2011) », dans *Dagstuhl Seminar*, 2011, vol. 10431.

- [13] A. BECHIR, « RESOLUTION DE PROBLEMES D'OPTIMISATION PAR LES SYSTEMES MULTI-AGENTS ET LES APPROCHES EVOLUTIONNAIRES », M.S. thesis, Université Mohamed Khider-Biskra, 2016.
- [14] E. A. Lemamou, *Ordonnancement de projet sous contraintes de ressources à l'aide d'un algorithme génétique à croisement hybride de type OEP*. Université du Québec à Chicoutimi, M.S. thesis, 2009.
- [15] IMED CHOUCHANI, « Utilisation d'un algorithme génétique pour la composition de services Web », M.S. thesis, Université du Saad dahleb de blida, 2010.
- [16] K. BOUMEDIENE et D. ARIBI, « Optimisation de la planification de l'auto-adaptation des systèmes par l'utilisation des métaheuristiques », M.S. thesis, Université du Saad dahleb de blida, 2019.
- [17] M. SERIR et Y. ATIK, « Meta-heuristique pour la gestion d'évolution d'architectures logicielles », M.S. thesis, Université du Saad dahleb de blida, 2018.
- [18] Dabbagh, N., and Bannan-Ritland, B. *Online learning: Concepts, strategies, and application*. Upper Saddle River, NJ: Pearson/Merrill/Prentice Hall, 2005.
- [19] D. Tavangarian, M. E. Leypold, K. Nölting, M. Röser, et D. Voigt, « Is e-Learning the solution for individual learning? Electronic Journal of e-Learnin », *Electron. J. e-Learning*, vol. 2, n° 2, p. 273-280, 2004, [En ligne]. Disponible à : <http://wwwra.informatik.uni-rostock.de>.
- [20] E. T. Welsh, C. R. Wanberg, K. G. Brown, et M. J. Simmering, « E-learning: emerging uses, empirical results and future directions », *Int. J. Train. Dev.*, vol. 7, n° 4, p. 245-258, 2003, doi: 10.1046/j.1360-3736.2003.00184.x.
- [21] P. Brusilovsky, « Methods and techniques of adaptive hypermedia », dans *Adaptive hypertext and hypermedia*, Springer, 1998, p. 1-43.
- [22] I. Cetina, D. Goldbach, et N. Manea, « Udemy: a case study in online education and training », *Rev. Econ.*, vol. 70, n° 3, 2018.
- [23] WILSON, L. et GRUZD, A. *MOOCs—international information and education phenomenon? Bulletin of the Association for Information Science and Technology*, 2014, vol. 40, no 5.

- [24] W. Rice et H. William, *Moodle*. Packt publishing Birmingham, 2006.
- [25] D. Hauger et M. Köck, « State of the Art of Adaptivity in E-Learning Platforms. », dans *LWA*, 2007, p. 355-360.
- [26] K. Czarnecki, S. Helsen, et U. Eisenecker, « Staged configuration using feature models », *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3154, p. 266-283, 2004, doi: 10.1007/978-3-540-28630-1_17.