



UNIVERSITÉ
DE LORRAINE



Saint-Dié

IUT SAINT-DIÉ-DES-VOSGES

INFORMATIQUE

PORTFOLIO UE2

COMPÉTENCE 2 : Optimiser

Enseignants : BOURJII Abdellatif

CASPARY Olivier

Étudiante : LIGEIRO, Eunice Eugenio

Saint-Dié-Des-Vosges, Juin 2025

Table of Contents

Introduction.....	3
Analyse des ressources et SAÉ	4
Semestre 1.....	4
R1.03 : Introduction à l'architecture des ordinateurs.....	4
Mon apprentissage :	4
R1.07 : Outils mathématiques fondamentaux.....	5
Mon apprentissage :	5
R1.04 : Introduction aux SE et à leur fonctionnement Objectifs :	6
Mon apprentissage :	6
SAÉ 1.02: Comparaison d'approches algorithmiques.....	6
Réalisations:	6
Semestre 2.....	8
R2.04 : Communication et fonctionnement à bas niveau	8
Mon apprentissage :	8
R2.07: Graphes.....	9
Mon apprentissage :	9
R2.09: Méthodes numériques.....	10
Mon apprentissage :	10
SAÉ 2.02: Exploration algorithmique d'un problème.....	10
Réalisations:	10
Analyse réflexive	11
Progression et acquis	11
Difficultés rencontrées	11
Perspectives d'amélioration	12
Conclusion	13

Introduction

Dans ce portfolio, je présente mon parcours d'apprentissage dans la compétence 2 « Optimiser des applications informatiques » au cours des deux premiers semestres du BUT informatique. Pour développer cette compétence, les ressources et SAÉ suivantes ont été étudiées et réalisées :

Semestre 1:

- R1.03 : Introduction à l'architecture des ordinateurs
- R1.06: Mathématiques discrètes
- R1.07: Outils mathématiques fondamentaux
- R1.04 Introduction aux SE et à leur fonctionnement
- SAÉ 1.02: Comparaison d'approches algorithmiques

Semestre 2:

- R2.07: Graphes
- R2.09: Méthodes numériques
- R2.04: Communication et fonctionnement bas niveau
- SAÉ 2.02: Exploration algorithmique d'un problème

Analyse des ressources et SAÉ

Semestre 1

R1.03 : Introduction à l'architecture des ordinateurs

Objectifs :

- Comprendre la structure et les composants d'un ordinateur.
- Appréhender le fonctionnement et les limites des systèmes informatiques.

Mon apprentissage :

Ce cours a été une véritable immersion dans les fondements du système informatique. J'ai appris à aller au-delà du code de haut niveau pour comprendre comment les algorithmes sont traduits en langage machine (MIPS). Cela m'a permis de manipuler les registres et la mémoire pour des opérations basiques mais essentielles à l'optimisation.

```
Algorithme

entier x,a=30 ;
Début ; x←a ; Fin

Code Source .asm (MIPS)

        .data 0x10000000
a :      .word 30
x :      .space 4
        .text
        .globl __start

__start:
lw $2,a
sw $2,x
```

Img1 : Opérations mémoire en MIPS

Explication : Ce code montre l'utilisation des instructions lw et sw pour charger et stocker des

données entre la mémoire et les registres. Cette manipulation directe est cruciale pour comprendre le lien entre code et matériel.

R1.07 : Outils mathématiques fondamentaux

Objectifs :

- Proposer des applications informatiques optimisées.
- Maîtriser les outils d'analyse descriptive de données.
- Représenter graphiquement des données.

Mon apprentissage :

Ce cours a consolidé mes bases en mathématiques, un atout essentiel pour tout développeur. J'ai travaillé sur le calcul numérique et algébrique : dérivation de fonctions, résolution de systèmes linéaires à l'aide de matrices (méthode de Gauss), etc. Cela m'a appris à modéliser des problèmes et à vérifier la cohérence des résultats.

$$\begin{cases} a + b + c + d = 7 \\ 8a + 4b + 2c + d = 25 \\ 27a + 9b + 3c + d = 61 \\ 64a + 16b + 4c + d = 121 \end{cases}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 7 \\ 8 & 4 & 2 & 1 & 25 \\ 27 & 9 & 3 & 1 & 61 \\ 64 & 16 & 4 & 1 & 121 \end{pmatrix}$$

3.

$$\begin{cases} a & & & & = 1 \\ & b & & & = 3 \\ & & c & & = 2 \\ & & & d & = 1 \end{cases}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Img2 : Résolution d'un système linéaire

Explication : Cette image montre le résultat d'un système résolu par la méthode de Gauss. Comprendre cette méthode est fondamental pour optimiser des algorithmes mathématiques complexes.

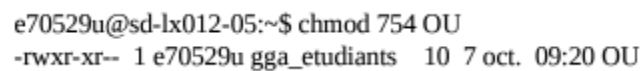
R1.04 : Introduction aux SE et à leur fonctionnement

Objectifs :

- Implémenter des fonctions de base d'un OS.
- Optimiser l'usage des ressources matérielles et logicielles.

Mon apprentissage :

Ce cours a fait le lien entre le matériel et le logiciel. J'ai étudié les systèmes d'exploitation, la gestion des processus, de la mémoire et des E/S. L'utilisation de la ligne de commande m'a permis de comprendre comment interagir efficacement avec le système.



```
e70529u@sd-lx012-05:~$ chmod 754 OU  
-rwxr-xr-- 1 e70529u gga_etudiants 10 7 oct. 09:20 OU
```

Img3 : Gestion des permissions avec chmod

Explication : Le terminal permet de contrôler précisément les droits d'accès aux fichiers. Savoir utiliser chmod est essentiel pour sécuriser et optimiser un système.

SAÉ 1.02: Comparaison d'approches algorithmiques

Réalisations:

- Conception d'un algorithme pour trouver le chemin le plus court entre plusieurs points.
- Calcul de distances euclidiennes.

- Implémentation d'une heuristique pour le problème du voyageur de commerce, avec gestion de batterie.
- Présentation des résultats (coordonnées, matrices de distances, chemin optimisé).

Dans ce projet, j'ai dû relever le défi d'optimiser un problème complexe similaire au TSP, en tenant compte de la batterie (limitée à 15 minutes de recharge). Ma contribution a porté sur la génération de coordonnées aléatoires et le calcul de la matrice de distances. J'ai aussi développé une heuristique prenant en compte la consommation énergétique.

```

18 // generer 8 coordenates aleatoire
19 // @param points: une tableau qui armazenne les coordennes aleatoires
20 // @note Les coordonnées sont choisies parmi un intervalle défini sans chevauchement
21 void coordenates_aleatoires(Coord points[])
22 {
23     int used[50][20] = {0}; // Tableau pour marquer les coordonnées utilisées
24
25     for (int i = 0; i < NUM_POINTS; i++)
26     {
27         int x, y;
28         do
29         {
30             x = rand() % 48 + 1; // Valeurs entre 1 et 48 pour éviter les bords
31             y = rand() % 18 + 1; // Valeurs entre 1 et 18 pour éviter les bords
32         } while (used[x][y]); // Répéter si la coordonnée est déjà utilisée
33
34         points[i].x = x;
35         points[i].y = y;
36         used[x][y] = 1; // Marquer la coordonnée comme utilisée
37     }
38 }

```

Img4 : Génération de coordonnées aléatoires

Explication : Ce code en C illustre la création d'un scénario réaliste, combinant mathématiques, structures de données et optimisation. L'implémentation en équipe via GitLab m'a aussi permis de progresser en collaboration et en gestion de versions.

Semestre 2

R2.04 : Communication et fonctionnement à bas niveau

Objectifs :

- Analyser les caractéristiques d'architectures informatiques.
- Implémenter des protocoles de communication.
- Optimiser les ressources logicielles et matérielles.

Mon apprentissage :

Ce cours m'a permis de mieux comprendre la communication entre logiciel et matériel. J'ai programmé en MIPS, en me concentrant sur les appels de procédure (jal, jr), la gestion de la pile, et la transmission de paramètres via les registres \$a0 à \$a3.

```
jal change

    li $v0 10
    syscall

change:
    sw $t0, 8($sp)
    sw $t1, 4($sp)
    sw $t2, 0($sp)
    mul $t0,$a1,4 #3
    add $t0,$a0,$t0 #4
    lw $t1,0($t0) #5
    lw $t2,4($t0) #6
    sw $t1,4($t0) #7
    sw $t2,0($t0) #8
    lw $t0, 8($sp)
    lw $t1, 4($sp)
    lw $t2, 0($sp)
    addi $sp, $sp, 12
    jr $31
    li $v0 10
    syscall      # syscall 10 (exit)
```

Img5 : Appel de procédure en MIPS

Explication : Ce code montre comment appeler une sous-routine en MIPS tout en sauvegardant l'état avec la pile. C'est une base essentielle pour les programmes efficaces en bas niveau.

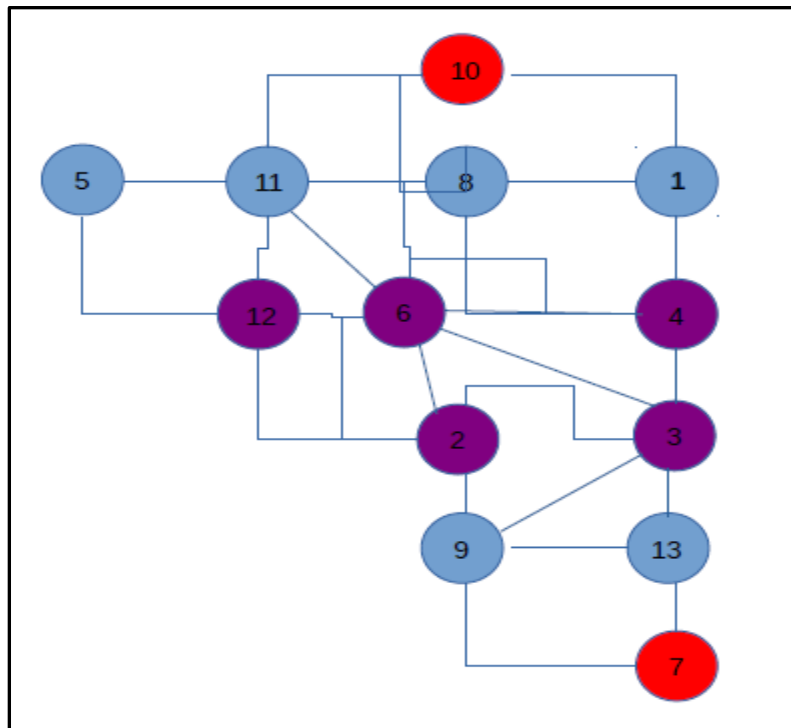
R2.07: Graphes

Objectifs:

- Modéliser des problèmes à l'aide d'outils mathématiques discrets.
- Implémenter des algorithmes de graphes.
- Résoudre des problèmes réels avec des représentations abstraites.

Mon apprentissage :

Les graphes sont devenus un outil puissant de modélisation. J'ai étudié les parcours (BFS, DFS) et les algorithmes de plus court chemin (Floyd-Warshall), ainsi que des techniques de coloration pour la gestion de conflits.



Img6 : Coloration de graphes

Explication : Ce graphe coloré représente un problème de planification. Les sommets liés ne peuvent avoir la même couleur. C'est une solution efficace pour organiser des emplois du temps ou répartir des ressources.

R2.09: Méthodes numériques

Objectifs:

- Modéliser des problèmes à l'aide de méthodes numériques.
- Proposer des solutions informatiques optimisées.
- Implémenter des algorithmes de régression, optimisation et approximation.

Mon apprentissage :

J'ai exploré des méthodes numériques pour résoudre des équations et approcher des solutions avec précision. L'étude du comportement des fonctions, des dérivées et des racines m'a donné une meilleure compréhension de l'optimisation.

SAÉ 2.02: Exploration algorithmique d'un problème

Réalisations:

- Modélisation d'un problème de planification d'examens.
- Application des algorithmes de coloration de graphe (Welsh-Powell et Glouton).
- Comparaison des méthodes selon l'ordre des sommets.
- Présentation collective des résultats.

Dans cette SAÉ, j'ai appliqué deux algorithmes de coloration et analysé l'impact de l'ordre des sommets. Cette expérience m'a permis de voir comment les choix algorithmiques influencent directement les résultats concrets. Le travail d'équipe, la gestion de projet en personne et la réflexion collective ont été très formatrices.

Sommet i	15	16	17	18	19	20	21	22	23	24	25	26	27
Couleur	2	1	4	3	3	2	1	3	1	2	2	1	3
Étudiant -module	E5-E	E6-B	E6-C	E6-D	E7-A	E7-C	E7-E	E8-B	E8-C	E8-D	E9-B	E9-D	E9-E

Im7 : Tableau de planification coloré

Explication : Chaque couleur correspond à une plage horaire. L'optimisation permet de minimiser les conflits entre événements et de réduire le nombre total de créneaux utilisés.

Analyse réflexive

Progression et acquis

Mon parcours suit une progression logique : des bases théoriques (architecture, OS, mathématiques) vers des applications concrètes et des optimisations algorithmiques.

Aujourd'hui, je me sens capable de :

- Comprendre le fonctionnement des systèmes informatiques dans leur globalité.
- Modéliser des problèmes réels avec des structures comme les graphes ou les matrices.
- Appliquer des algorithmes efficaces pour trouver des solutions optimisées.
- Analyser les performances et améliorer l'utilisation des ressources.
- Collaborer efficacement dans des projets en utilisant Git et d'autres outils de versionnage.

Difficultés rencontrées

- Le MIPS et la gestion de la pile ont exigé beaucoup de rigueur.
- Le choix des heuristiques et des stratégies de tri dans les graphes a été complexe à maîtriser.

- Passer du travail individuel à la collaboration en équipe m'a appris à être critique, à m'adapter, et à intégrer différentes approches.

Ces obstacles m'ont appris à persévérer et à raisonner de façon plus analytique et créative.

Perspectives d'amélioration

Je souhaite:

- Approfondir l'optimisation matérielle (caches, parallélisme).
- Explorer l'optimisation continue et les problèmes à grande échelle.
- M'améliorer dans la vulgarisation de concepts techniques.
- Apprendre de nouveaux langages et comprendre leurs mécanismes d'optimisation.

Conclusion

La Compétence 2 - Optimiser des applications informatiques n'a pas été qu'un apprentissage technique. Elle a été une nouvelle manière de penser : optimiser, c'est réfléchir intelligemment, avec méthode et vision.

Grâce aux SAÉs, j'ai pu passer de la théorie à des cas proches de la réalité professionnelle. Ces expériences m'ont appris l'importance du travail en équipe, de la planification et de l'analyse rigoureuse. Je sais que le chemin continue, mais aujourd'hui je me sens solide sur mes bases.