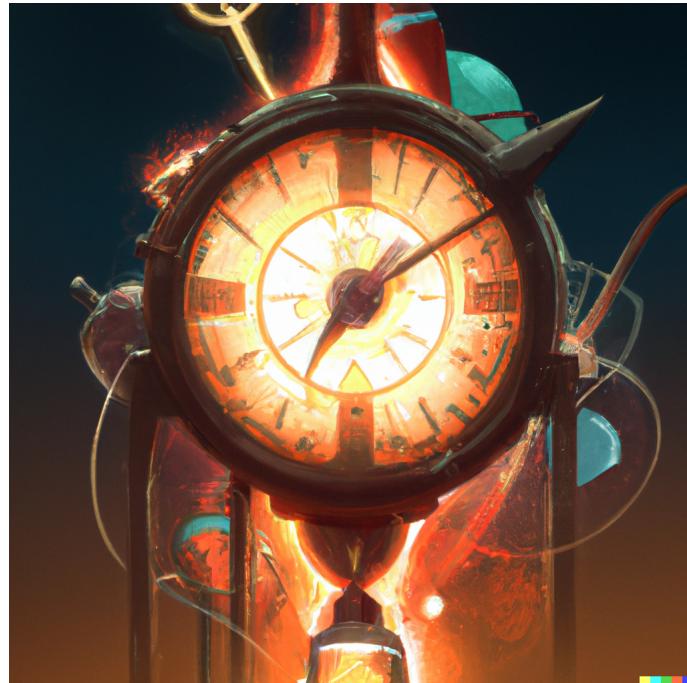


Rapport de Projet

Janvier/Mai 2023

Alicia Houset, Ange Mercoyrol-dol, Nathan Fontaine



Avant-propos

Bonjour, voici le Rapport de projet

Table des matières

1	Introduction	3
2	Calendrier	4
2.1	Structure de données	5
2.2	Fonctionnalités	5
2.3	Fonctionnalités avancées	6
2.3.1	Ajout de plusieurs années	6
2.3.2	Gestion des événements récurrents	6
2.3.3	Calcul de la charge d'entraînement	7
2.3.4	Gestion des conflits d'agenda	7
3	Interface	8
4	Algorithme de génération du planning	9
4.1	Introduction	9
4.2	Mise en place des contraintes	12
4.2.1	Contrainte de charge et de volume d'entraînement	14
4.2.2	Volume horaire hebdomadaire	15
4.2.3	Conclusion des contraintes	16
4.3	Initialisation du planning	16
4.4	Modification du planning	17
5	Bilan	19

1 Introduction

OptiPlan est une application de gestion de planning sportif. Elle permet aux utilisateurs de créer et gérer leur emploi du temps sportif de manière efficace. L'application permet d'ajouter, supprimer et organiser des séances d'entraînement et des activités sportives en fonction du planning de base de l'utilisateur.

L'objectif principal d'OptiPlan est d'optimiser le temps de l'utilisateur en garantissant que les séances d'entraînement et les activités sportives sont planifiées de manière cohérente et respectent les contraintes de disponibilité de l'utilisateur. L'application propose une interface conviviale qui permet de visualiser clairement le planning et de le mettre à jour facilement.

OptiPlan offre également des fonctionnalités de suivi de l'avancement. L'utilisateur peut enregistrer ses progrès et marquer les séances d'entraînement ou les activités sportives comme complétées. Cela permet de suivre les performances et de rester motivé dans l'atteinte des objectifs sportifs.

En outre, l'application propose des fonctionnalités de communication pour améliorer l'efficacité de l'utilisateur. Cela peut inclure des notifications pour rappeler les séances d'entraînement, des rappels d'objectifs ou des conseils personnalisés pour maximiser les résultats.

Il est également important de noter que OptiPlan permet la sauvegarde automatique du planning sportif. Ainsi, les utilisateurs peuvent accéder à leur emploi du temps à tout moment et depuis n'importe quel appareil connecté.

En résumé, OptiPlan est une application de gestion de planning sportif qui aide les utilisateurs à optimiser leur temps en planifiant et en suivant leurs séances d'entraînement et leurs activités sportives de manière efficace.

Dans ce projet Ange s'est occupé de la mise en place du calendrier ainsi que de la gestion de l'algorithme de recuit simulé. Alicia s'est occupé de l'implémentation de l'interface. Nathan quand à lui s'est occupé de la mise en lien des différentes parties.

2 Calendrier

Le calendrier est au coeur de notre projet. Il permet de stocker toutes les informations relatives aux entraînements et compétitions des athlètes. Dans cette partie, nous allons détailler la structure de données utilisée pour le calendrier ainsi que les différentes fonctionnalités que nous avons implémentées. Voici les différentes structures que l'on utilise. Celles-ci sont détaillées ci-dessous.

```
struct Date {
    int day;
    int month;
    int year;
};

struct Day {
    struct Date date;
    int numActivities;
    int dayTss;
    float CTL;
    float fatigue;
    float stress_balance;

    struct Activity **activities;
};

struct Week {
    struct Day days[7];
    int totalTss;
    int isRestWeek;
};

struct Month {
    char name[20];
    int numDays;
    int firstDayOfWeek;
    struct Week weeks[6];
};

struct Year {
    int year;
    struct Month months[12];
};

struct Calendar {
    int numYears;
    struct Year *years;
};

struct Activity {
    int isSport;
    char name[50];
    int startHour;
    int startMinute;
    int endHour;
    int endMinute;
    int duration; // en minutes
    int intensity;
    int tss;
};
```

2.1 Structure de données

Le calendrier est représenté par une structure de type **Calendar** qui contient un tableau dynamique d'années. Chaque année est représentée par une structure de type **Year** qui contient un tableau de mois. Enfin, chaque mois est représenté par une structure de type **Month** qui contient un tableau de jours.

La structure **Day** contient le jour du mois, une valeur de charge pour l'entraînement du jour, un type d'événement (compétition, entraînement, jour de repos...) et une note qui peut être utilisée pour stocker des informations supplémentaires (ex : conditions météorologiques).

La structure **Month** contient le mois de l'année, le nombre de jours dans le mois, ainsi qu'un tableau de **Day**.

La structure **Year** contient l'année, ainsi qu'un tableau de 12 **Month**, représentant les mois de l'année.

La structure **Calendar** contient le nombre d'années dans le calendrier, ainsi qu'un tableau dynamique de pointeurs vers des structures **Year**.

2.2 Fonctionnalités

Nous avons implémenté plusieurs fonctionnalités pour le calendrier :

- Ajout d'une année : une fonction permet d'ajouter une nouvelle année au calendrier. Le tableau de pointeurs vers les années est alors agrandi de manière dynamique.
- Ajout d'un jour : une fonction permet d'ajouter un jour au calendrier, à une date précise. Cette fonction prend en compte les années bissextiles pour le calcul du nombre de jours dans le mois.
- Modification d'un jour : une fonction permet de modifier les informations d'un jour existant dans le calendrier (charge, type d'événement, note).
- Suppression d'un jour : une fonction permet de supprimer un jour du calendrier.
- Récupération des entraînements : une fonction permet de récupérer tous les jours d'entraînement d'une année donnée, ainsi que leur charge totale.

- Récupération des compétitions : une fonction permet de récupérer toutes les compétitions d'une année donnée, ainsi que leur date et leur type.
- Affichage du calendrier : une fonction permet d'afficher le calendrier complet d'une année donnée, avec les informations de chaque jour.
- Gestion des activités : le calendrier permet l'ajout et la suppression d'activités pour chaque jour. Chaque activité est caractérisée par son nom, son heure de début, sa durée et sa note éventuelle.
- Calcul de la charge de travail : le calendrier permet de calculer la charge de travail totale pour une semaine ou un mois donné en fonction des activités ajoutées.

2.3 Fonctionnalités avancées

En plus des fonctionnalités de base du calendrier, nous avons ajouté des fonctionnalités plus avancées pour améliorer sa flexibilité et son utilité. Voici quelques-unes de ces fonctionnalités :

2.3.1 Ajout de plusieurs années

La première fonctionnalité avancée que nous avons implémentée est la possibilité d'ajouter plusieurs années au calendrier. Au lieu d'avoir une seule année comme c'était le cas dans la version de base du calendrier, notre version permet d'ajouter autant d'années que nécessaire. Cela permet aux utilisateurs de planifier à long terme et de garder une trace de leurs activités sur une période prolongée.

L'ajout de plusieurs années a été réalisé en utilisant un tableau de pointeurs de structures d'année. Lorsqu'un utilisateur ajoute une nouvelle année, nous avons réalloué le tableau de pointeurs pour inclure une case supplémentaire et alloué dynamiquement une nouvelle structure d'année pour stocker les données de cette année.

2.3.2 Gestion des événements récurrents

Une autre fonctionnalité utile que nous avons ajoutée est la gestion des événements récurrents. Cette fonctionnalité permet aux utilisateurs de

créer des événements qui se répètent à intervalles réguliers, tels que des entraînements hebdomadaires ou des rendez-vous mensuels.

Pour implémenter cette fonctionnalité, nous avons ajouté un champ à la structure d'événement pour stocker la fréquence de répétition. Les utilisateurs peuvent choisir de répéter un événement quotidiennement, hebdomadairement, mensuellement ou annuellement. Lorsqu'un utilisateur crée un événement récurrent, le calendrier génère automatiquement des occurrences de cet événement aux intervalles spécifiés.

2.3.3 Calcul de la charge d'entraînement

Enfin, nous avons ajouté une fonctionnalité pour calculer la charge d'entraînement pour chaque semaine du calendrier. La charge d'entraînement est une mesure de la quantité de travail que le corps effectue lors d'une séance d'entraînement et est généralement utilisée pour aider les athlètes à planifier leur entraînement et à éviter les blessures.

Nous avons utilisé la méthode de calcul de la charge d'entraînement de Coggan, qui utilise la puissance (mesurée en watts) et la durée de chaque séance d'entraînement pour calculer la charge d'entraînement. Nous avons ajouté un champ à la structure d'événement pour stocker la puissance moyenne de chaque séance d'entraînement, et nous avons utilisé cette information pour calculer la charge d'entraînement pour chaque semaine du calendrier.

2.3.4 Gestion des conflits d'agenda

Une autre fonctionnalité que nous avons ajoutée est la gestion des conflits d'agenda. Lorsqu'un utilisateur ajoute un nouvel événement à une date donnée, le système vérifie s'il y a déjà un événement prévu à cette date et à cette heure. Si c'est le cas, le système affiche un message d'erreur indiquant que l'utilisateur doit choisir une autre date ou heure.

Nous avons également ajouté une fonctionnalité de fusion pour les événements qui se chevauchent partiellement. Si deux événements ont des périodes de chevauchement, nous fusionnons les deux événements en un seul événement de plus longue durée. Cette fonctionnalité permet d'éviter les conflits d'agenda et de regrouper les événements similaires.

3 Interface

Pour l'interface graphique de notre application, nous avons utilisé Glade. Glade est un outil interactif de conception graphique GTK+. Il prend en charge la partie gestion/génération de l'interface pour permettre au développeur de se concentrer sur la partie développement et sur le code dit " utile ". GtkBuilder permet quant à lui de lire ces fichiers dynamiquement.

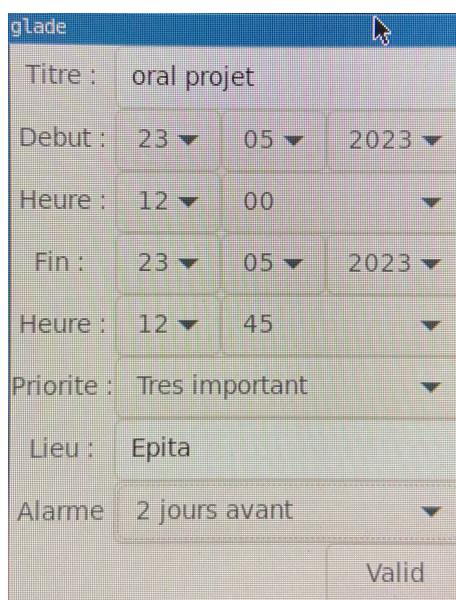
Dans un premier temps, pour la première soutenance, il a fallu se familiariser avec Glade, connaître et comprendre ses différentes fonctionnalités pour ensuite les initialiser avec GTK dans le code. Nous avons alors commencé l'interface en créant la base pour notre application. C'est-à-dire la page principale où sera affiché l'emploi du temps, avec différents boutons. Nous avons donc mis plusieurs boutons à l'aide de GtkButton, soit un bouton " + " pour lequel lorsqu'on appuie dessus nous pouvons ajouter un événement au calendrier, des boutons " > " et " < " afin de voir le planning des semaines précédentes ou des semaines suivantes. Un bouton " Paramètres " pour changer les différents paramètres de l'interface, une barre de recherche " Rechercher " qui permettra à l'utilisateur de rechercher la date d'un événement particulier et pour finir un bouton " Fermer " pour fermer l'interface et donc quitter l'application.

Pour la seconde soutenance, nous avons initialisé les deux boutons : " Paramètres " et " + ". C'est-à-dire que lorsque l'on appuie sur le bouton " + ", une nouvelle interface apparaît, où il est possible de rentrer les différentes informations du nouvel événement que l'on souhaite ajouter. Comme l'intitulé, l'importance de l'événement ou de la tâche (si elle doit être faite en priorité ou non), la date à laquelle nous voulons que commence l'évènement et à laquelle nous voulons qu'elle se termine. Pour le bouton " Paramètres ", une nouvelle interface apparaît aussi. Elle permet de gérer les différents réglages de l'application. L'affichage du calendrier et la synchronisation des événements.

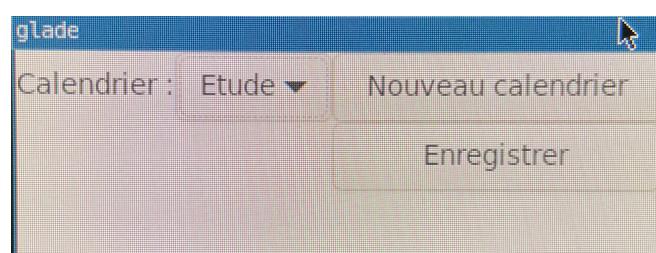
Pour cette soutenance finale, nous avons pour l'interface principale créé un calendrier et initialisé les boutons " > " et " < " afin de pouvoir accéder aux semaines suivantes et précédentes. Nous avons donc fait en sorte que le nombre jours de chaque mois correspondent. Nous avons ensuite initialiser l'interface des paramètres afin de pouvoir ajouter un nouveau calendrier à la liste puis le sélectionner. Lorsqu'on

sélectionne un calendrier dans la liste, le nouveau calendrier s'affiche avec son titre en haut de page. Pour l'interface qui permet d'ajouter un événement au calendrier, nous avons modifier quelques paramètres comme ajouter une alarme pour les jours, les heures ou les minutes qui précèdent cet événement. Nous avons aussi changer la façon de sélectionner le début et la fin d'un événement en faisant en sorte qu'il ne soit pas possible de valider si le jour ou l'horaire de fin ne soit pas cohérente avec celle de debut. Il est aussi impossible de valider si toutes les données ne sont pas rentrées et valides. Pour finir l'interface est bien fonctionnelle, il est donc possible de parcourir son calendrier, ajouter des évènement et créer différents calendriers.

Exemple d'un ajout d'un événement :



Ajouter et selectionner un calendrier :



4 Algorithme de génération du planning

4.1 Introduction

L'algorithme de recuit simulé est un outil puissant utilisé dans divers domaines pour résoudre des problèmes d'optimisation. Dans le

contexte de la génération de planning sportif automatisé, cet algorithme joue un rôle essentiel pour organiser de manière efficace les séances d'entraînement et les activités sportives.

Le recuit simulé tire son inspiration de l'analogie avec le processus de recuit en métallurgie, où un matériau est chauffé puis progressivement refroidi pour atteindre une structure cristalline optimale. De manière similaire, l'algorithme de recuit simulé explore un espace de solutions possibles en acceptant occasionnellement des solutions sub-optimales afin de s'échapper des minima locaux et de converger vers une solution globale optimale.

L'objectif de l'algorithme de recuit simulé dans la génération du planning sportif est de trouver une répartition optimale des séances d'entraînement et des activités sportives en tenant compte des contraintes et des préférences de l'utilisateur. Cela inclut la maximisation du temps d'entraînement, la prise en compte des disponibilités de l'utilisateur, l'équilibrage des types d'activités, etc.

En utilisant l'algorithme de recuit simulé, il est possible de générer un planning sportif de manière automatisée, en explorant différentes combinaisons de séances d'entraînement et en évaluant leur qualité en fonction des critères définis. L'algorithme ajuste progressivement le planning en effectuant des transitions entre les solutions pour rechercher des améliorations. Ce processus itératif permet d'obtenir un planning de haute qualité, qui répond aux besoins et aux objectifs de l'utilisateur.

L'algorithme de recuit simulé est une méthode d'optimisation qui permet d'explorer de manière intelligente un espace de solutions possibles dans le but de trouver une solution optimale, même lorsque la fonction objectif n'est pas convexe et présente des minima locaux. Il est basé sur le concept de thermodynamique et utilise des mécanismes d'acceptation de solutions sub-optimales pour éviter de rester bloqué dans des solutions locales non optimales.

Le fonctionnement de l'algorithme de recuit simulé peut être décrit en plusieurs étapes :

1. **Initialisation** : L'algorithme démarre en générant une solution initiale aléatoire ou en utilisant une solution de départ préexistante. Cette solution représente un premier agencement des séances d'entraînement et des activités sportives dans le planning.
2. **Évaluation de la solution** : La solution initiale est évaluée en

utilisant une fonction objectif qui mesure sa qualité. Dans le contexte de la génération du planning sportif, la fonction objectif peut prendre en compte des critères tels que le temps d'entraînement total, l'équilibre entre les types d'activités, la satisfaction des contraintes de disponibilité, etc.

3. Boucle principale : L'algorithme entre dans une boucle principale qui itère sur un certain nombre d'itérations ou jusqu'à ce qu'une condition de convergence soit atteinte. À chaque itération, l'algorithme effectue les étapes suivantes :

- A) **Génération d'une nouvelle solution voisine :** Une nouvelle solution est générée à partir de la solution courante en effectuant une perturbation ou une modification spécifique. Cela peut inclure des échanges de séances d'entraînement, des modifications d'horaires ou de durées, etc.
- B) **Évaluation de la nouvelle solution :** La nouvelle solution est évaluée en utilisant la fonction objectif pour mesurer sa qualité.
- C) **Acceptation ou rejet de la nouvelle solution :** L'algorithme décide d'accepter ou de rejeter la nouvelle solution en fonction de sa qualité et de la température courante. Plus précisément, l'algorithme utilise la formule de Metropolis pour calculer une probabilité d'acceptation basée sur la différence de qualité entre la solution courante et la nouvelle solution, ainsi que sur la température courante. Cette probabilité permet d'accepter des solutions sub-optimales dans le but d'explorer de nouvelles régions de l'espace de recherche.
- D) **Mise à jour de la température :** Après chaque itération, la température est mise à jour en utilisant un schéma spécifique. La température diminue progressivement au fil des itérations, ce qui réduit la probabilité d'accepter des solutions sub-optimales et permet à l'algorithme de converger vers une solution optimale.

4. Solution finale : Une fois que la boucle principale est terminée, l'algorithme renvoie la meilleure solution trouvée au cours du processus.

L'algorithme de recuit simulé dans la génération du planning sportif utilise donc ces principes pour explorer l'espace des solutions possibles,

accepter des solutions sub-optimales et progressivement converger vers une solution optimale qui respecte les contraintes du planning sportif.

La formule utilisée pour calculer la probabilité d'acceptation dans l'étape de l'acceptation ou du rejet de la nouvelle solution est la formule de Metropolis, donnée par :

$$P(\text{accepter la nouvelle solution}) = \exp\left(-\frac{\text{différence de qualité}}{\text{température courante}}\right)$$

Cette formule indique que plus la différence de qualité entre la solution courante et la nouvelle solution est grande, plus la probabilité d'acceptation de la nouvelle solution est élevée. Cependant, cette probabilité diminue à mesure que la température courante diminue au fil des itérations. Ainsi, au début de l'algorithme, lorsque la température est élevée, des solutions sub-optimales peuvent être acceptées plus facilement pour permettre l'exploration de nouvelles régions de l'espace de recherche. Au fur et à mesure que la température diminue, la probabilité d'acceptation de solutions sub-optimales diminue également, ce qui favorise la convergence vers une solution optimale.

En résumé, l'algorithme de recuit simulé dans la génération du planning sportif utilise une approche itérative pour explorer l'espace des solutions possibles, en acceptant occasionnellement des solutions sub-optimales pour éviter les minima locaux. La probabilité d'acceptation est calculée en utilisant la formule de Metropolis, qui prend en compte la différence de qualité entre la solution courante et la nouvelle solution ainsi que la température courante. Grâce à ce processus, l'algorithme converge progressivement vers une solution optimale qui satisfait les contraintes du planning sportif.

4.2 Mise en place des contraintes

Dans cette partie, nous décrirons les différentes contraintes que nous avons mis en place pour l'algorithme principal de développement. En effet nous avons vu précédemment que l'algorithme de recuit simulé s'appuie sur des données chiffrées afin de déterminer la solution avec le coût le plus bas. S'il n'existe qu'une seule solution au sudoku, la notion du meilleur planning est subjective. C'est pourquoi il est nécessaire de

définir des contraintes et d'ajouter des pénalités au cout total de l'algorithme si ceux-ci ne sont pas respectés.

- Contrainte de charge d'entraînement : nous avons fixé une limite supérieure pour la charge d'entraînement totale de chaque semaine, afin d'éviter les blessures dues à une surcharge d'entraînement. Nous avons également mis en place une contrainte de variation maximale de la charge d'entraînement entre deux semaines consécutives. Cette charge d'entraînement est calculée à partir du modèle de Coggan. Ainsi on compare le score de stress total (TSS) de la semaine en cours avec celui de la semaine précédente. Si le TSS de la semaine en cours dépasse 10
- Contrainte de volume horaire hebdomadaire : nous avons fixé une limite supérieure pour le volume horaire total de chaque semaine, afin de garantir que les athlètes disposent d'un temps suffisant pour récupérer entre les séances d'entraînement. Nous avons aussi mis en place une contrainte de variation maximale de la charge d'entraînement entre deux semaines consécutives. Aussi on vérifie si la semaine en cours est plus difficile que la semaine précédente. Cela peut dépendre de critères spécifiques tels que l'intensité, la durée et le type d'activités. Si la semaine en cours est considérée comme plus difficile, on ajoute une pénalité de +1000 au coût. On vérifie néanmoins si la semaine en cours est plus difficile que la semaine précédente. Cela peut dépendre de critères spécifiques tels que l'intensité, la durée et le type d'activités. Si la semaine en cours est considérée comme plus difficile, on ajoute une pénalité de +1000 au coût.
- Contrainte de temps minimal entre deux entraînements : nous avons défini un temps minimal entre deux entraînements consécutifs pour éviter les blessures et permettre une récupération adéquate. On s'assure qu'il y ait au moins 6 heures de repos entre deux activités. Si le repos est insuffisant, on ajoute une pénalité de +1000 au coût.
- Contrainte de type d'entraînement : nous avons fixé une proportion minimale de chaque type d'entraînement pour garantir une progression équilibrée de l'athlète. Ceci est fait lors de l'initialisation du planning.

On identifie aussi les opportunités d'amélioration, même si la semaine est valide. Par exemple :

Si le TSS de la semaine en cours est inférieur à 5

Si des activités non sportives sont planifiées trop proches dans le temps, on ajoute une pénalité de +10 au coût. Cela encourage à répartir les activités non sportives pour une meilleure récupération et variété.

Plus il y a de temps entre les activités difficiles, mieux c'est. Si une journée d'activités faciles se trouve entre les journées qui contiennent une part importante du TSS total de la semaine, la semaine est mieux organisée.

Si les activités difficiles sont planifiées plus tôt dans la journée, il s'agit d'un meilleur arrangement.

Les valeurs de pénalité (+1000, +10, etc.) peuvent être ajustées selon les besoins et préférences spécifiques.

Ces contraintes ont été implémentées dans l'algorithme principal de développement pour garantir un plan d'entraînement sûr, équilibré et efficace pour les athlètes.

4.2.1 Contrainte de charge et de volume d'entraînement

La charge d'entraînement est une mesure de la quantité de travail que le corps effectue lors d'une séance d'entraînement et est généralement utilisée pour aider les athlètes à planifier leur entraînement et à éviter les blessures. Nous avons choisi d'utiliser la méthode de calcul de la charge d'entraînement de Coggan, car elle est largement utilisée dans le domaine du cyclisme et peut être appliquée à une variété d'autres sports.

Le modèle de Coggan utilise la puissance (mesurée en watts) et la durée de chaque séance d'entraînement pour calculer la charge d'entraînement. Plus précisément, la charge d'entraînement est calculée en multipliant la puissance moyenne de la séance d'entraînement (mesurée en watts) par la durée de la séance (mesurée en heures), puis en multipliant le résultat par un facteur de pondération qui prend en compte l'intensité de l'entraînement.

Le facteur de pondération est calculé à partir de la puissance maximale aérobie (PMA) de l'athlète, qui est mesurée en watts et est utilisée pour déterminer les zones d'entraînement de l'athlète. Les zones d'entraînement sont des plages de puissance qui correspondent à différents niveaux d'intensité d'entraînement, et chaque zone a son

propre facteur de pondération.

Nous avons ajouté un champ à la structure d'événement pour stocker la puissance moyenne de chaque séance d'entraînement, et nous avons utilisé cette information pour calculer la charge d'entraînement pour chaque semaine du calendrier. Nous avons également ajouté des contraintes pour s'assurer que la charge d'entraînement de chaque semaine reste dans une plage acceptable pour l'athlète.

Les contraintes de charge d'entraînement sont définies comme suit :

- La charge d'entraînement totale pour chaque semaine ne doit pas dépasser un certain pourcentage de la charge d'entraînement maximale que l'athlète peut tolérer.
- La charge d'entraînement pour chaque jour ne doit pas dépasser un certain pourcentage de la charge d'entraînement maximale que l'athlète peut tolérer.
- La charge d'entraînement pour chaque entraînement ne doit pas dépasser un certain pourcentage de la charge d'entraînement maximale que l'athlète peut tolérer.

Nous avons également ajouté des contraintes pour limiter le volume horaire hebdomadaire et le temps minimal entre deux entraînements, afin d'éviter la surcharge et de permettre à l'athlète de récupérer entre les entraînements. Ces contraintes sont définies dans les sections suivantes.

4.2.2 Volume horaire hebdomadaire

Le volume horaire hebdomadaire est une autre contrainte importante à prendre en compte lors de la planification de l'entraînement. Le volume horaire représente la quantité de temps que l'athlète passe à s'entraîner chaque semaine. Une augmentation rapide du volume horaire peut entraîner une fatigue excessive et augmenter le risque de blessure, tandis qu'une diminution trop rapide peut entraîner une perte de forme physique.

Nous avons donc ajouté une contrainte pour limiter le volume horaire hebdomadaire de l'athlète. Cette contrainte est définie par une limite supérieure et inférieure de volume horaire pour chaque semaine du calendrier. Si le volume horaire prévu pour une semaine donnée dépasse la limite supérieure ou tombe en dessous de la limite

inférieure, l'algorithme d'optimisation doit trouver un autre horaire qui respecte ces limites.

Pour calculer le volume horaire pour chaque semaine, nous avons simplement ajouté la durée de chaque séance d'entraînement prévue pour cette semaine. La durée de chaque séance d'entraînement est calculée à partir des heures de début et de fin de l'événement.

4.2.3 Conclusion des contraintes

Dans cette partie, nous avons présenté les différentes contraintes que nous avons mises en place pour notre algorithme principal. Nous avons tout d'abord expliqué comment nous avons calculé la charge d'entraînement en utilisant le modèle de Coggan. Nous avons ensuite présenté la contrainte de volume horaire hebdomadaire, qui permet de limiter le nombre d'heures d'entraînement par semaine. Enfin, nous avons introduit la contrainte du temps minimal entre deux entraînements, qui permet d'éviter les surcharges de travail.

En mettant en place ces contraintes, nous avons cherché à créer un algorithme qui permet de planifier un calendrier d'entraînement qui soit à la fois efficace et sûr pour l'athlète. Nous avons utilisé des méthodes éprouvées pour calculer la charge d'entraînement et limiter le volume horaire hebdomadaire, tout en nous assurant que l'athlète dispose d'un temps de récupération suffisant entre les séances d'entraînement.

Nous sommes convaincus que ces contraintes nous permettent de proposer un algorithme efficace et fiable pour la planification de l'entraînement, en prenant en compte les besoins spécifiques de chaque athlète.

4.3 Initialisation du planning

L'initialisation du planning dans l'algorithme de génération du planning sportif se fait en attribuant de manière aléatoire les séances d'entraînement et les activités sportives aux différents créneaux horaires du planning. Cela peut être réalisé en respectant les contraintes de disponibilité des ressources (telles que les équipements ou les lieux d'entraînement) et en équilibrant les types d'activités sur la semaine.

Lors de cette étape d'initialisation, une approche couramment utilisée est de suivre un modèle polarisé, qui consiste à répartir les activités en fonction de leur objectif principal. Par exemple, on peut attribuer les activités à but d'endurance principalement le matin ou en début de journée, tandis que les activités de renforcement musculaire peuvent être programmées plutôt en fin de journée. Cette répartition est basée sur les principes de l'entraînement sportif, qui recommandent de séquencer les activités en fonction de leurs effets sur le corps et le système d'énergie.

L'objectif de cette répartition est d'optimiser les performances et les bénéfices de l'entraînement en exploitant les avantages des différents moments de la journée pour chaque type d'activité. Par exemple, l'entraînement en endurance le matin peut aider à stimuler le métabolisme et à améliorer la condition cardiovasculaire, tandis que les séances de renforcement musculaire en fin de journée peuvent favoriser la récupération et le développement musculaire.

En résumé, lors de l'initialisation du planning dans l'algorithme de génération du planning sportif, les séances d'entraînement et les activités sportives sont attribuées de manière aléatoire aux créneaux horaires du planning, en respectant les contraintes de disponibilité et en suivant un modèle polarisé qui tient compte des objectifs spécifiques des activités. Cette étape vise à établir un point de départ initial pour le processus d'optimisation et à favoriser une répartition équilibrée des activités tout au long de la semaine.

4.4 Modification du planning

Dans l'algorithme de génération du planning sportif, la modification du planning est effectuée en fonction du coût total. Si le coût total dépasse un certain seuil, par exemple 1000, cela signifie qu'une ou plusieurs des contraintes principales du planning ne sont pas respectées. Dans ce cas, des modifications doivent être apportées afin de rétablir le respect de ces contraintes. Après avoir effectué les modifications nécessaires pour respecter les contraintes principales, l'algorithme peut se concentrer sur l'optimisation du planning. Cela peut être réalisé en ajustant certains aspects du planning, tels que la difficulté des activités ou la répartition des activités en fonction des objectifs de l'utilisateur. Par exemple, si l'utilisateur souhaite augmenter le niveau de difficulté de son programme d'entraînement, des activités plus intenses peuvent être

programmées ou la durée des séances peut être augmentée.

L'objectif de la modification et de l'optimisation du planning est d'assurer que les contraintes principales du planning sportif sont respectées, tout en offrant une planification qui correspond aux préférences et aux objectifs de l'utilisateur.

En résumé, la modification du planning dans l'algorithme de génération du planning sportif est réalisée en fonction du coût total du planning. Si le coût total dépasse un seuil prédéfini, des modifications sont effectuées pour rétablir le respect des contraintes principales.

Ensuite, des ajustements peuvent être réalisés pour optimiser le planning en fonction des objectifs de l'utilisateur, tels que l'augmentation de la difficulté ou la répartition des activités.

5 Bilan

Durant ce projet, nous avons développé une application de planning sportif en C avec une interface utilisateur graphique conviviale. Notre objectif était d'aider les athlètes à planifier leurs entraînements de manière efficace et à atteindre leurs objectifs sportifs.

Nous avons réussi à implémenter plusieurs fonctionnalités clés, telles que l'ajout d'années, de jours et la possibilité de modifier ou supprimer des événements existants. De plus, pour répondre à la demande de recherche algorithmique, nous avons créé un algorithme de recuit simulé qui permet de générer automatiquement des programmes d'entraînement.

Notre application prend en compte diverses contraintes, telles que la charge d'entraînement optimale et la prévention des blessures. Ainsi, les athlètes peuvent s'entraîner de manière équilibrée et éviter les risques de surcharge.

Cependant, nous avons également rencontré des difficultés en termes d'organisation de groupe, en particulier après le départ de notre chef de groupe. Malgré ces défis, ce projet nous a permis de gagner en autonomie dans la programmation en langage C et de mieux comprendre les avantages et les limitations de ce langage.

En conclusion, notre projet offre une solution complète de planification d'entraînements sportifs grâce à notre application de calendrier interactif et à ses fonctionnalités spécifiquement conçues pour répondre aux besoins des athlètes.