Table des matières

Introduction

But du projet

Le but du projet est de réaliser un emplois du temps au regard de certaines contraintes en utilisant une méthode d'évolution artificielle. Le résultat produit devra évaluer la « fitness » du meilleur élément et respecter l'ensemble des contraintes obligatoires.

Présentation du langage

Le langage considéré se nomme EASEA (EAsy Specification of Evolutionary Algorithms) (voir https://lsiit.u-strasbg.fr/easea/index.php/EASEA_platform) et permet de manipuler le parallélisme des architectures multi-cœur (GPU notamment) de manière basique. Le code produit passe par une compilation en C++. Le formalisme du langage engage quelques paramètres et méthodes de fonctionnement qu'il n'est pas nécessaire de maîtriser pour les employer. Enfin il est à noter que la documentation relative au langage reste très sommaire.

Analyse du sujet

Le sujet nous invite à développer un algorithme d'évolution artificielle permettant de générer le meilleur emplois du temps possible, au regard de contraintes établies.

Il apparaît que l'axe de procédure, bien que prédéfinie par les sections du code EASEA est un élément très important dans le développement de cet algorithme. En effet, une démarche logique engendrera naturellement une évolution logique. A contrario, une définition floue ou ambiguë des éléments de l'emploi du temps peut conduire à une perte de sens dans l'évolution.

Bien que succinctes, les instructions du projet mettent en avant les contraintes obligatoires, fortes et faibles. Ces contraintes doivent permettre de calculer la fitness des génomes, de manière à obtenir celui qui respectera au mieux ces contraintes, et ce, encore une fois, de manière logique et naturelle.

On peut estimer que les contraintes posées sont relativement théoriques, simplifiées par rapport à la réalité d'un vrai établissement. Le model développé devrait donc prendre en compte de potentiels ajustements permettant d'intégrer plus de contraintes, et surtout un nombre plus important d'éléments (classes, profs, matières…)

Lien vers le sujet

Voir sujet de TP envoyé par mail.

Développement et suivi du projet

Etapes de développement

1° Etablissement d'un modèle naturel et cohérent

- Modélisation d'un emplois du temps
- Réflexion relative aux éléments mobiles et fixes
- Détail des contraintes obligatoires implicites

2° Analyse des sections de l'évolution artificielle

- Représentation des sections (Initialisation, Cross-Over, Mutation, Evaluation)
- Schématisation du fonctionnement de l'algorithme
- Orientation de chaque section

3° Mise en place des variables globales et moyens d'affichage

- Permettre la visualisation des génomes
- Faciliter le calcul de la fitness
- Donner du sens aux composants du génome.

4° Implémentation des sections

- Avec l'initialisation, description des classes utiles
- Axer le développement sur les orientations définies
- Affiner l'évaluation pour permettre une correction plus aisée et plus juste

5° Optimisation du fonctionnement

- Analyse des différentes sections individuellement et conjointement
- Lecture de l'affichage final et des scores intermédiaires
- Equilibrer l'exploration et la prospection

Etudes du fonctionnement de l'algorithme

La phase d'initialisation est complètement dépendante des variables globales définies en préambule dans le code. Ces données sont celles émises dans le sujet. Au moyen des classes établies, on génère des génomes de manière semi aléatoire. Cela signifie, que dès l'initialisation, on associe à une classe le bon nombre de cours, et à chaque cours, en fonction de la matière, un enseignant et une salle adéquate.
Cette démarche vise principalement à détecter une incapacité physique à respecter les contraintes fortes. Elle est mise en place essentiellement car on connaît l'attitude du génome que l'on recherche, et ce par rapport aux contraintes implicites étudiées.

L'algorithme se repose ensuite sur le fonctionnement du code EASEA, et selon le paramétrage, détermine au cours des générations, les meilleurs individus. On notera toute l'importance de la fonction d'évaluation, et encore plus celle de l'affichage final, permettant

de détecter d'éventuelles erreurs dans la « boite noire » de l'algorithme.

Choix opérés sur la représentation des structures de données

Utilisation de variables globales

Comme déjà avancé, les variables globales permettent une bonne initialisation de l'algorithme. Faire tourner l'algorithme avec des valeurs aberrantes n'aurait aucun sens, puisque le résultat que l'on souhaite obtenir doit être valide, naturel et cohérent. La mise en place de ces variables globales permet donc un accès rapide et simple vers ces données. De plus, il est alors plus facile de contrôler la pertinence des valeurs inscrites, ou de les faire évoluer. Des intervalles de valeurs ont été donnés de manière à restreindre les risques de valeurs aberrantes.

Choix des classes définies par l'utilisateur

Le choix a été fait de déclarer 6 plus 1 classes. La classe génome étant naturellement constituée des cinq autres éléments, implémentés par les cinq classes établies.

- Classe « Créneaux » : *CreneauxClass*

Un créneau correspond à une plage horaire un certain jour de la semaine. Ils sont numérotés de 0 à NBHORAIRES * NBJOURS. Le potentiel maximal de cours par créneaux dépend essentiellement du nombre de salles (minimum entre profs, salles, classes).

- Classe « Cours » : *CoursClass*

Un cours a lieu sur un créneau, il relie un prof, une salle et une classe. Il est également rattaché à une matière scolaire. Il peut y avoir plusieurs cours simultanément sur un créneau.

- Classe « Classes » : *ClassesClass*

Une classe représente un ensemble d'élève. Il existe plusieurs types de classes qui ont chacune un certain nombre d'horaires par matières. Une classe ne peut être que dans un seul cours à la fois.

- Classe « Profs » : *ProfsClass*

Un prof représente un enseignant. Chaque enseignant dispose d'un certain nombre d'horaires de cours (au maximum) et ne peut être que dans un cours à la fois. Il ne peut donc faire cours qu'à une classe à la fois. Un enseignant n'enseigne qu'une seule matière prédéfinie.

- Classe « Salles » : *SallesClass*

Une salle représente un lieu d'apprentissage. Il en existe de plusieurs types, relativement aux différentes matières. Un cours ne peut avoir lieu que dans une salle correspondant à la matière associée. Une salle ne peut accueillir qu'un seul cours à la fois.

La définition de ces classes est le résultat du model établi au cours de la première phase.

Un génome se compose donc d'un ensemble de créneaux, remplis chacun par un ou plusieurs cours, eux-mêmes composés obligatoirement d'un prof, d'une salle et d'une classe, respectant la relation à la matière associée au cours.

Choix relatifs aux sections du protocole EASEA

Section initialisation

Comme énoncé dans l'étude du fonctionnement, la phase d'initialisation se fait par rapport aux variables globales. Le choix a été fait de faire une initialisation intelligente, rendant aléatoire les éléments mobiles, et déterminées les contraintes obligatoires fixes. Cette démarche va à l'encontre du « tout aléatoire » et permet de limiter les erreurs lors de l'évaluation.

Remarque : il aurait été possible de rajouter des méthodes interdisant tout décalage et toute faute. Cependant, cette logique ne tirerait aucun bénéfice de la démarche évolutionnaire. Aussi, les choix opérés, bien que plus contraignants que l'aléatoire, respectent le protocole EASEA. Qui plus est, l'algorithme développé, sous réserve de petits ajustements, reste complètement réceptif à tout type de contraintes faibles supplémentaires. Les données arbitrairement figées sont des données naturellement fixes.

Section Cross-Over

Cette section ne comporte pas d'attribut majeur. Elle permet par rapport à un seuil (par défaut 50%) de déplacer un cours dans le créneau associé du parent 1 ou du parent 2. Cela permet une exploration, et peut engendrer des erreurs de cohérence. Ces erreurs sont rattrapées dans la section de mutation suivante.

Section Mutation

A l'instar de la section d'initialisation, cette section se veut intelligente. Elle permet d'établir des diagnostiques sur l'utilisation des différents créneaux, ainsi que sur le potentiel et l'occupation des différents éléments des cours. Elle permet de réajuster les incohérences en cas d'erreur par un replacement aléatoire mais valide. Cette démarche s'éloigne donc également un peu du tout aléatoire, en vue de rechercher une efficacité plus forte.

Section Evaluation

La phase d'évaluation détermine la fitness de chaque individu, et est donc centrale dans le protocole d'évolution artificielle. Bien que permettant une finesse et une facilité d'estimation de la fitness de chaque individu, cette section reste relativement rapide. A un très gros volume de données, il se peut qu'elle cause des ralentissements dans le processus de recherche du meilleur individu, mais ce très grand nombre ne serait pas cohérent par rapport au sujet. (exemple : aucun établissement ne possède plus d'un million d'enseignants).

Section Affichage

Bien que cette section ne fasse pas parti du protocole EASEA, une attention vraiment particulière y a été apportée pour les raisons énoncées au cours de ce rapport. Cette section reprend par ailleurs l'algorithme de la section précédente pour enrichir l'affichage.
(Ralentit potentiellement le processus ; à ne pas mettre à chaque nouvelle génération.)

Etude partielle du code implémenté

Extrait de code de la section initialisation

```
/*          Initialisation des disponibilités des Profs & Salles par Matiere                    */

for (int x=0; x<NBMATIERES; x++)
{
        for (int k=0; k<nbProfsTx[x]; k++)              { nbPoPTx[x] += Genome.profs[k+seuilP]->potentiel; }
        for (int k=0; k<nbSallesTx[x]; k++)             { nbPoSTx[x] += Genome.salles[k+seuilS]->potentiel; }
        po_par_profsTx[x]   = new int [nbPoPTx[x]]; nbP = 0;
        po_par_sallesTx[x]  = new int [nbPoSTx[x]]; nbS = 0;

        for (int k=0; k<nbProfsTx[x]; k++)
        {
                for (int d=0; d<Genome.profs[k+seuilP]->potentiel; d++)
                { po_par_profsTx[x][nbP] = k+seuilP; nbP++; }
        }
        for (int k=0; k<nbSallesTx[x]; k++)
        {
                for (int d=0; d<Genome.salles[k+seuilS]->potentiel; d++)
                { po_par_sallesTx[x][nbS] = k+seuilS; nbS++; }
        }
        seuilP += nbProfsTx[x];
        if (x > 0)
        { seuilS += nbSallesTx[x]; }
}
```

Cet extrait présente l'initialisation des disponibilités des enseignants et des salles par matière. Il est a noté principalement, le sens de nommage des variables ainsi que l'optimisation substantielle du code. La rédaction du code se veut au maximum générique, de manière à réaliser le moins possible de changements lors d'ajout de contraintes, ou lors de modifications de paramètres.

Comme expliqué précédemment, l'imbrication des boucles n'engendre pas de ralentissement conséquent puisque la valeur des données est minime, et le calcul de complexité n'a pas d'importance pour cet ordre de grandeur. (dans la configuration par défaut : 4*[1 à 3]*10).

Afin de comprendre l'extrait, il faut considérer que `Genome.profs` représente l'ensemble des enseignants, que `nbProfsTx` représente le nombre de profs enseignant la matière x. De plus, le seuilP (et réciproquement pour S pour les salles) permet de considérer uniquement les enseignant rattachés à cette matière. Les enseignants sont par ailleurs triés par matière.

Extrait de code de la section mutation

Le code suivant représente la correction avancée lors de la présentation de la section. En cas d'erreur, ou de mutation volontaire, le cours est repositionné dans l'emplois du temps. Si aucune solution valide n'est possible, le cours n'est pas déplacé et l'individu sera donc sanctionné lors de l'évaluation.

```
/* en cas d'erreur, on "corrige" le cours en le réinitialisant de manière valide.		*/
	if (err > 0)
	{
		nb++;
		r		= j;		t = 0;
		do
		{
			r = (r + 1)%NBCRENEAUX;
			if (t++ > NBCRENEAUX)
			{ break; }
		}
		while
		(
			(Genome.creneaux[r]->potentiel <= 0) ||
			(
				(Genome.classes	[c]	->occupation[r] > 0) ||
				(Genome.profs	[p]	->occupation[r] > 0) ||
				(Genome.salles	[s]	->occupation[r] > 0)
			)
		);

		Genome.classes	[c]	->occupation[j]--;
		Genome.profs	[p]	->occupation[j]--;
		Genome.salles	[s]	->occupation[j]--;
		Genome.creneaux	[j]	->potentiel++;

		Genome.classes	[c]	->occupation[r]++;
		Genome.profs	[p]	->occupation[r]++;
		Genome.salles	[s]	->occupation[r]++;
		Genome.creneaux	[r]	->potentiel--;
	}

	Genome.cours[i]->creneau = r;
```

A la première erreur (contrainte forte violée), l'algorithme rentre dans cette sous-section. On peut clairement identifier le point de succès de la boucle « do-while » qui s'énonce de la manière suivante : « J'arrête d'avancer dans les créneaux quand je trouve un créneau vierge, ou que de placer le cours sélectionné, ne génère aucune autre erreur. » Un point d'arrêt supplémentaire est inséré dans la boucle, qui permet de s'arrêter après avoir testé tous les cas. (dans la configuration par défaut, tous les cas = 20 = nombre de créneaux).

La modification se fait en terme constant puisqu'il s'agit de modifier les valeurs de plusieurs tableaux.

Remarque : dans cette section, une fonction (ou sous-section) a été mise en commentaire. Elle avait pour but de muter selon un seuil, l'enseignant du cours sélectionné, en le remplaçant si possible par un autre prof (enseignant la même matière), cours pour cours. N'apportant aucune amélioration significative en terme d'exploration, elle a été désactivée.

Aperçu du rendu d'exécution

```
                    Current generation 250 Generational limit : 250
                                Stopping criterion reached

+--------------------+-------------------------------------------------------------------------+
|                    |  LUNDI   +  MARDI   + MERCREDI +  JEUDI   + VENDREDI +  SAMEDI  |
| HORAIRE  +  TYPE   |                                                                         |
+--------------------+-------------------------------------------------------------------------+
|          |CLASSE   |          |2 3       |3         |0 3       |3         |          |
|Horaire 0 |MATIERE  |          |0 2       |2         |2 1       |0         |          |
|          |PROF     |          |0 6       |6         |6 3       |1         |          |
|          |SALLE    |          |0 3       |3         |3 0       |0         |          |
+--------------------+-------------------------------------------------------------------------+
|          |CLASSE   |0 2 3     |1 2 3     |0 1 3     |0 1 3     |0 1 3     |          |
|Horaire 1 |MATIERE  |0 2 1     |0 2 1     |2 0 1     |1 2 3     |3 0 2     |          |
|          |PROF     |2 7 5     |0 6 5     |6 0 4     |3 6 8     |8 1 7     |          |
|          |SALLE    |2 3 0     |0 3 2     |3 2 1     |1 3 4     |4 1 3     |          |
+--------------------+-------------------------------------------------------------------------+
|          |CLASSE   |0 1 2 3   |1 2       |0 1       |0 1 2     |0 1 2 3   |          |
|Horaire 2 |MATIERE  |2 0 1 0   |3 1       |0 1       |0 3 1     |0 2 1 3   |          |
|          |PROF     |7 2 5 0   |9 5       |0 4       |1 8 3     |1 7 4 8   |          |
|          |SALLE    |3 0 1 2   |4 0       |0 2       |2 4 0     |0 3 1 4   |          |
+--------------------+-------------------------------------------------------------------------+
|          |CLASSE   |1 2       |2         |          |0 1 2     |0 2       |          |
|Horaire 3 |MATIERE  |2 0       |3         |          |3 1 2     |1 3       |          |
|          |PROF     |7 0       |9         |          |8 3 7     |4 8       |          |
|          |SALLE    |3 0       |4         |          |4 1 3     |2 4       |          |
+--------------------+-------------------------------------------------------------------------+

+--------------------+-------------------------------------------------------------------------+
|                    |            Affichage par Classe de leurs stats :                        |
+--------------------+-------------------------------------------------------------------------+
|          |  Total  |      0       |       1       |       2       |       3       |
+--------------------+-------------------------------------------------------------------------+
|Nb jours  |       18|             4|              5|              4|              5|
+--------------------+-------------------------------------------------------------------------+
|Nb Horaire|       44|            11|             11|             11|             11|
+--------------------+-------------------------------------------------------------------------+
|Err.   H.|        2|             1|              0|              1|              0|
+--------------------+-------------------------------------------------------------------------+
|Err. Creux|        0|             0|              0|              0|              0|
+--------------------+-------------------------------------------------------------------------+
|Err.  m/j|        0|             0|              0|              0|              0|
+--------------------+-------------------------------------------------------------------------+

+------------------------+-------------------------------------------------------------------------+
|                        |            Affichage par Prof de leurs stats :                          |
+------------------------+-------------------------------------------------------------------------+
|          |    Total    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
+------------------------+-------------------------------------------------------------------------+
|Nb jours  |           20|  3|  2|  1|  1|  2|  2|  3|  3|  2|  1|
+------------------------+-------------------------------------------------------------------------+
|Nb Horaire|           44|  6|  4|  2|  4|  4|  4|  6|  6|  6|  2|
+------------------------+-------------------------------------------------------------------------+
|Err.   H.|            3|  0|  1|  0|  1|  0|  0|  0|  1|  0|  0|
+------------------------+-------------------------------------------------------------------------+
|Err. Creux|            0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
+------------------------+-------------------------------------------------------------------------+
|Err.  c/j|            0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
+------------------------+-------------------------------------------------------------------------+

+------------------------+-------------------------------------------------------------------------+
|                        |            Affichage par Creneau de leurs stats :                       |
+------------------------+-------------------------------------------------------------------------+
|          |    Total    | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|18|19|
+------------------------+-------------------------------------------------------------------------+
|Occup.    |           44| 0| 3| 4| 2| 2| 3| 2| 1| 1| 3| 2| 0| 2| 3| 3| 3| 1| 3| 4| 2|
+------------------------+-------------------------------------------------------------------------+
|Err.  1000|            0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
+------------------------+-------------------------------------------------------------------------+
|Err.     1|           43|                                                                         |
+------------------------+-------------------------------------------------------------------------+


                    OBJECTIF           :        Atteindre 0 pts.
                    MEILLEUR SCORE     :        43 pts
                    TEMPS D'EXECUTION  :        0.6 secondes
```

Le minimum absolu théorique fixé avec les consignes du sujet est de 16+18 = 34 points.
Une liberté a été prise par rapport au sujet, de mettre les points de manière positive.
Le but, comme indiqué, est donc de minimiser la fitness, et non plus de maximiser.

Conclusion

La réalisation de ce projet a été intéressante dans l'optique de comprendre et appréhender les techniques et approches d'évolution artificielle. Bien que de nombreux paramètres soient figés, la modélisation, et l'axe initial jouent un rôle majeur dans la cohérence et l'efficacité de l'algorithme. Des compromis sont à faire, et des choix à opérer. La difficulté ne réside ainsi pas sur l'implémentation à proprement parler, mais sur le modèle que l'on cherche à établir. La conception d'un tel modèle doit rester naturel, et surtout, respecter une validité certaine. Aucun résultat ne pourra être entériné si les composants de base ne permettent pas d'atteindre un objectif valide.

L'observation du comportement d'un tel algorithme a été pour moi la manière la plus simple et la plus adéquate de savoir dans quel sens l'orienter. En effet, l'intégration de composants aléatoires rend plus qu'incertain le résultat. De plus, les mécanismes sous-jacent du protocole EASEA restent des boites noires dont on ne ressort que des résultats à posteriori. Influencer les données à l'entrée, de manière intelligente mais non restrictive semble être la meilleure manière d'obtenir des résultats probants.

L'idée directrice de ce mini projet a été pour moi de me rendre compte qu'il est inutile de générer des erreurs en quantité, quand on connaît le résultat attendu. Pour autant, l'opération du protocole EASEA nécessite une certaine exploration, et l'indéterminisme des éléments mobiles (notamment le placement du cours dans un créneau) est crucial. La difficulté a donc résidé dans le fait de concilier orientation et exploration.

Liste des qualités estimées du projet

- Un pourcentage de résultat valide et cohérent très intéressant.
- Une diversité relative des « bons individus » générés à chaque essai.
- Une capacité de trouver un « bon individu » en peu de générations et génomes.
- Une rapidité notable d'exécution du processus.
- Un affichage fonctionnel, détaillé et lisible.
- Une généricité pour une évolution possible du code en vue d'intégrer facilement d'autres contraintes et/ou modifier les valeurs d'initialisation.