

FlopC#

Présentation des fonctionnalités
du langage

Introduction à FlopC#

- On prononce "flop c dièse" en français et "flop c sharp" en anglais.
- FlopC# veut dire: "Formulation of Linear Optimisation Problem" en C#. Il reprend le nom de FlopC++ duquel il s'inspire.
- FlopC# est bâti sur le même principe que FlopC++
 - Un langage de modélisation algébrique dans un langage procédural.
 - Utilisation de la POO
 - Redéfinition d'opérateurs

Elements du langage

- FCS_Set : un ensemble d'éléments indexés de 0 à n associés à une description textuelle.
- FCS_Subset : un ensemble de 5-uplets dont les composante prennent des valeurs (int) dans des FCS_Set ou des FCS_Subset.
- FCS_Data : un ensemble de "double" indexés par des 5-uplets prenant des valeurs dans des FCS_Subset.
- FCS_Variables : un ensemble de variables LP indexées par un FCS_Subset.
- FCS_Constraint : un ensemble d'expressions stockées sous forme d'arbre et indexées par un FCS_Subset

Elements du langage

- FCS_Index : un objet contenant un "int". C'est une façon de passer un int par référence et non par valeur sans utiliser "ref". Il permet de contrôler l'évaluation d'une expression en jouant sur la valeur des index.
- FCS_BaseDomain : un FCS_set associé à un FCS_index. Il permet de contrôler l'itération sur un FCS_Set en se limitant aux valeurs définies.
- FCS_Domain : des FCS_Subset associés à des index et à d'autres FCS_BaseDomain ou FCS_Domain. Une généralisation du contrôle des itérations sur les domaines.

Utilisation de FlopC#

- Pour créer simplement un set et le remplir soi-même.
 - `FCS_Set set1=new FCS_Set();`
 - `set1.Add();`
 - `set1.Add("deuxième élément.");`
- Pour créer automatiquement un set avec un nombre prédéfini d'éléments
 - `FCS_Set set1=new FCS_Set(8); // 8 l'éléments.`

Utilisation de FlopC#: FCS_Data

- Création d'un FCS_Data
 - `FCS_Data data1=new FCS_Data(set1,set2);`
//jusqu'à 5 set possibles.
 - `Data1[1,1]=4.56;` //marche pourvu que qu'il y ait au moins 2 éléments dans chacun des sets (index commence à zero.).

Utilisation de FlopC#: FCS_Variable

- Création de variables LP
- `FCS_Variable var1=new
FCS_Variable(set1,set2); //jusqu'à 5 sets`
- `var1[i,j]` pour faire référence à l'élément `i` dans `set1` et `j` dans `set2`.

Utilisation de FlopC#: FCS_Constraints

- Création des contraintes
 - `FCS_Constraint cons1=new
FCS_Constraint(set1,set2); //jusqu'à 5 sets`
 - `cons1[i,j]=data1[i,j]*var1[i,j] <= data2[i,j];`

Utilisation de FlopC#: fonction objectif

- Création du modèle:
 - Pour créer le modèle il faut ajouter les variables et les constraints à l'objet FCS_Model.
 - `FCS_Model model=new FCS_Model();`
 - `model.AddVariable(var1);`
 - `model.AddConstraint(cons1);`
 - `model.CreateProblem();`
- Fonction objectif:
 - `model.Objective(var1[1]);` //just a definition
 - `model.maximise();` //to launch the optimisation.