Master SESI [M1]
UE CAO – Examen
Placement de Rectangles «Mous»

Alain GREINER

Juin 2010

Durée : 2 heures – Tous documents autorisés

Le barème est donné à titre indicatif et peut être modifié par le correcteur

Écrivez lisiblement, un texte difficilement déchiffrable sera toujours considéré comme faux

On s'intéresse au problème du placement d'objets déformables de forme rectangulaire dans un rectangle englobant soumis à une contrainte de hauteur maximale.

Le problème du placement des rectangles mous se pose – entre autres – dans le contexte de la conception des circuits intégrés analogiques : un circuit analogique est une interconnexion de composants élémentaires (transistor, capacitance, résistance, inductance, ...). À une fonction analogique $\mathcal{F}$ donnée correspond un certain schéma électronique réalisé avec un nombre fixé N de composants élémentaires (quelques dizaines au maximum). À la différence des composants numériques, les composants analogiques sont déformables : un composant élémentaire peut occuper une surface très variable, qui dépend de ses caractéristiques électriques. En première approximation, lorsque la caractéristique électrique d'un composant est fixée (valeur de la capacité d'un condensateur, valeur du courant d'un transistor), la surface S se trouve fixée, mais sa forme (c'est à dire sa largeur L et sa hauteur H) peuvent varier en respectant la condition :

$$S = L \times B = \texttt{constante}$$

Par ailleurs, suivant le contexte d'utilisation, la même fonction analogique $\mathcal{F}$ (c'est à dire le meme schema de principe) peut avoir des réalisations très différentes correspondant à différentes valeurs électriques (et donc à différentes surfaces) des composants élémentaires.

Dans la suite on considèrera que chaque composant élémentaire est représenté par un «rectangle mou» R_i. Un rectangle mou R_i est doublement déformable :

- La surface S_i du rectangle R_i est variable puisqu'elle dépend des caractéristiques électriques du composant.
- Pour une surface S_i fixée, tous les couples (L_i, H_i) tels que $L_i \times H_i = S_i$ sont des réalisations possibles.

Pour une fonction analogique $\mathcal{F}$, la topologie (c'est à dire le placement relatif des rectangles mous les uns par rapport aux autres) est une caractéristique fixe de $\mathcal{F}$.

La topologie est décrite de façon hiérarchique en utilisant la technique des «conteneurs». Un conteneur est un aboutement de rectangle mous. Il y a deux types de conteneur :

– Un conteneur horizontal est un aboutement d'un nombre quelconque de rectangles mous R_i pour former une rangée. Tous les rectangles R_i doivent avoir la même hauteur H, mais ont des largeurs L_i quelconques.
– Un conteneur vertical est un empilement d'un nombre quelconque de rectangles mous R_i pour former une colonne. Tous les rectangles R_i doivent avoir la même largeur L, mais ont des hauteurs H_i quelconques.

Un conteneur est lui-même un rectangle mou.

Puisqu'un conteneur est soit un composant élémentaire, soit un aboutement d'autres conteneurs, la topologie d'une fonction analogique donnée est donc décrite par un «arbre des conteneurs». Chaque nœud de l'arbre correspond à un conteneur. Les nœuds fils d'un nœud $\mathcal{C}$ quelconque représentent la liste ordonnée des conteneurs contenus dans $\mathcal{C}$ (aboutés de gauche à droite pour un conteneur horizontal, ou de bas en haut pour un conteneur vertical). Les conteneurs feuilles sont les conteneurs représentant les composants élémentaires déformables. Le conteneur racine de l'arbre représente le rectangle englobant la fonction analogique $\mathcal{F}$ complète.

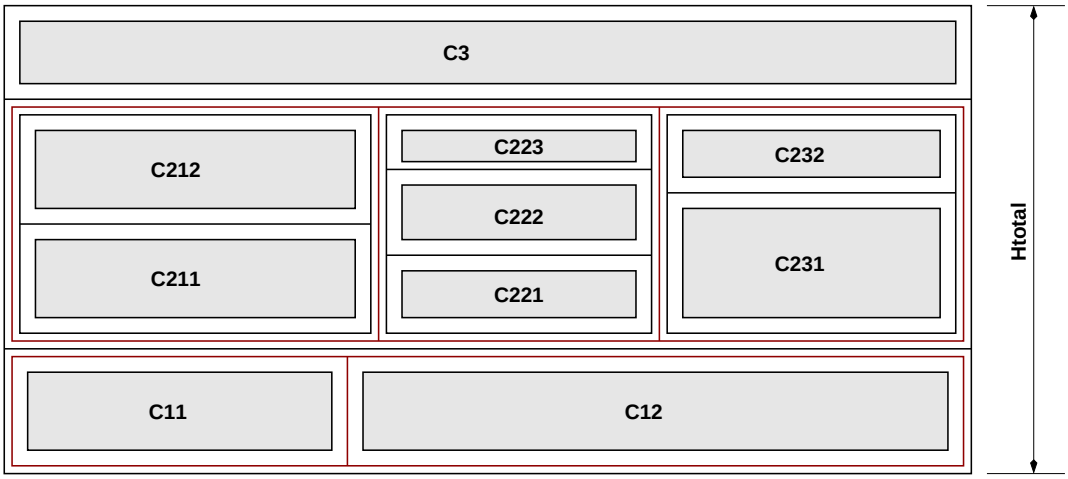

FIG. 1 – Première réalisation de la fonction analogique $\mathcal{F}$

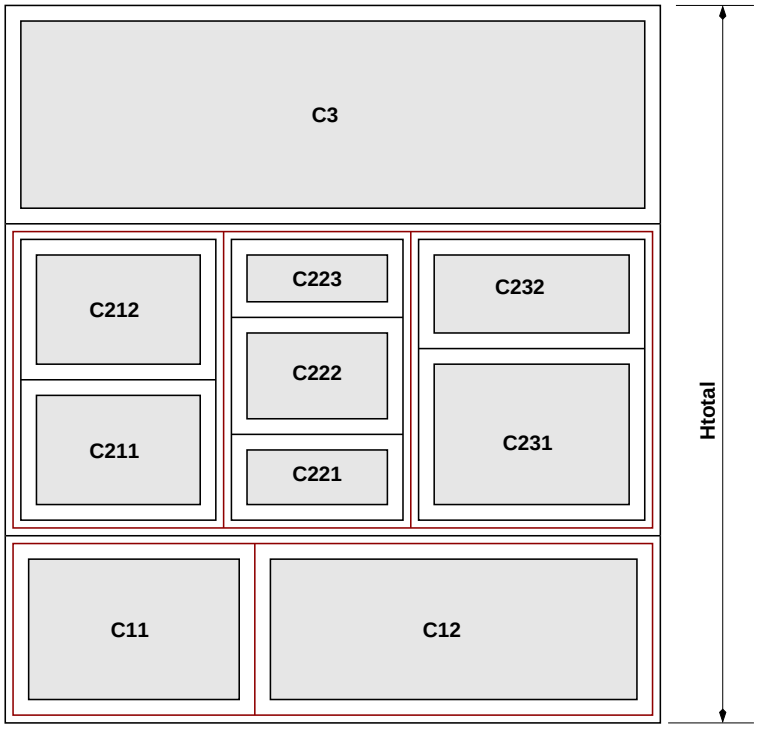

FIG. 2 – Seconde réalisation de la fonction analogique $\mathcal{F}$

Le problème que l'on cherche à résoudre

Pour une réalisation particulière de la fonction $\mathcal{F}$, on attribue à chaque composant élémentaire R_i une surface S_i. Ces surfaces S_i sont imposées par les caractéristiques électriques requises pour les composants élémentaires. On impose une valeur H_{total} sur la hauteur du rectangle englobant la fonction analogique complete, ainsi que les coordonnées (X_0, Y_0) du coin bas gauche. On cherche à calculer la largeur L_i et la hauteur H_i et les coordonnées (X_i, Y_i) de chaque composant élémentaire C_i

Idée Générale

Le problème peut être résolu par trois parcours récursifs de l'arbre des conteneurs :

1. Le premier parcours permet de calculer la surface S de tous les conteneurs, quelle que soit leur position dans l'arbre, sachant qu'initialement seules les surfaces S_i des conteneurs feuilles représentant les composants élémentaires R_i sont définies.
2. Le deuxième parcours récursif permet de calculer les dimensions (L, H) de tous les conteneurs, en propageant de la racine vers les feuilles la contrainte de hauteur maximale H_{total}.
3. Le troisième parcours récursif permet de calculer les coordonnées (X_i, Y_i) du coin bas gauche de tous les conteneurs.

La classe permettant de représenter un nœud de l'arbre des conteneurs en langage C++ est définie comme suit :

```
class Conteneur {
  public:
    enum conteneur_type { H, V, R }
  private:
  // Variables Membres -- Attributs
    std::string            m_name;     // nom du conteneur
    conteneur_type         m_type;     // H / V / R
    std::list<Conteneur*>  m_fils;     // liste des conteneurs fils
    float                  m_surface;  // surface
    float                  m_largeur;  // largeur
    float                  m_hauteur;  // hauteur
    float                  m_x;        // coordonnee X coin bas gauche
    float                  m_y;        // coordonnee X coin bas gauche
  public:
  // Accesseurs
    inline float           getSurface     () { return m_surface; }
    inline float           getLargeur     () { return m_largeur; }
    inline float           getHauteur     () { return m_hauteur; }
    inline float           getX           () { return m_x; }
    inline float           getY           () { return m_y; }
  public:
  // Modifieurs
           float           setSurface     ();
           void            setDimensions  (float largeur, float hauteur);
           void            setCoordinates (float x, float y);
  public:
  // Constructeur
                           Conteneur      (std::string name, conteneur_type type);
}; // Fin de "Conteneur".
```

La liste des conteneurs fils est vide pour les conteneurs terminaux (type `R`).

Tous les calculs de hauteur, largeur, surface, coordonnées sont effectués sur des nombre réels représentés par des `float`. Pour simplifier, on négligera (à tort…) les problèmes d' imprécision dûs aux arrondis de calcul.

Dans le cas d'un conteneur non-terminal, la fonction recursive `setSurface()` calcule la valeur de la surface du conteneur, met à jour la variable `m_surface`, et retourne cette valeur. Dans le cas d'un conteneur terminal, cette fonction retourne simplement la valeur de la variable `m_surface`.

La fonction récursive `setDimensions(float largeur,float hauteur)` prend en entrée les valeurs des deux parametres `largeur` et `hauteur` du conteneur. Elle met à jour les deux variables membres `m_largeur` et `m_hauteur`, et fait en sorte que les variables `m_largeur` et `m_hauteur` de tous les conteneurs fils soient également mis à jour.

La fonction recursive `setCoordinates(float x,float y)` prend en entrée les coordonnées du coin bas gauche du conteneur. Elle met à jour les les deux variables membres `m_x` et `m_y`, du conteneur, et fait en sorte que les coordonnées du coin bas gauche de chacun des conteneur fils soient également mises à jour.

Question 1 **2pt**

Représenter graphiquement l'arbre des conteneurs correspondant à la figure. Les nœuds représenteront les conteneurs. Les arcs représenteront les relations père → fils. Les feuilles de l'arbre représenteront les composants élémentaires et porteront les noms définis dans la figure.

Question 2 **2pt**

Proposer une méthode permettant de calculer la surface d'un conteneur père connaissant la surface de tous ses fils.

Question 3 **3pt**

Ecrire en langage C++ la fonction récursive `setSurface()` qui permet de calculer la surface des conteneurs non-terminaux, lorsque toutes les surfaces des conteneurs terminaux sont définies.

Question 4 **3pt**

Proposer une méthode permettant de calculer les dimensions (L_i, H_i) de chacun des conteneurs fils lorsqu'on a fixé les dimensions (L, H) du conteneur père, et qu'on connait la surface S_i occupée par chacun des fils.

Question 5 **3pt**

Ecrire en langage C++ la fonction récursive `setDimensions()`, qui propage de la racine vers les feuilles la contrainte de dimension imposée sur la racine de l'arbre.

Question 6 **2pt**

Proposer une méthode permettant de calculer ces coordonnées (X_i, Y_i) du coin bas gauche de chacun des conteneurs fils, lorsqu'on a fixé les coordonnées (X, Y) du coin bas-gauche du conteneur père, et qu'on connait les largeurs et hauteurs (L_i, H_i) des fils.

Question 7 **3pt**

Ecrire en langage C++ la fonction récursive `setCoordinates()`, qui propage de la racine vers le calcul des coordonnées du coin bas gauche de chacun des conteneurs, lorsqu'on a fixé les coordonnées du coin bas gauche du rectangle englobant.

Question 8 **2pt**

Est-il possible d'optimiser ce code en modifiant les fonctions de parcours, de façon à effectuer deux parcours de l'arbre des conteneurs au lieu de trois ?