



## Multi-Start Simulated Annealing for FPGA Floorplanning with Partially-Reconfigurable Regions

---

Francois Galea, Sergiu Carpov and Lilia Zaourar

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 1, 2019

# Recuit simulé multi-start pour le floorplanning FPGA avec régions partiellement reconfigurables

François Galea<sup>1</sup>, Sergiu Carpov<sup>1</sup>, Lilia Zaourar<sup>1</sup>

CEA, LIST, PC 172, F-91191 Gif-sur-Yvette Cedex, France  
{francois.galea,sergiu.carpov,lilia.zaourar}@cea.fr

**Mots-clés :** *recuit simulé, floorplanning, FPGA, multi-start.*

## 1 Introduction

Dans le flot de conception de circuits intégrés sur cible FPGA, le floorplanning peut s'avérer très utile en amont de l'étape du placement/routage, afin de faciliter l'obtention d'un placement initial de bonne qualité. Il consiste à trouver un placement satisfaisant de régions pré-déterminées du circuit sur la matrice de ressources qui compose le matériel FPGA, en minimisant la distance entre les régions qui communiquent entre elles, ainsi qu'entre les régions et les ports d'entrée/sortie qui y sont reliés. À ce problème difficile peuvent s'ajouter des contraintes supplémentaires, telles que la prise en compte de régions partiellement reconfigurables.

Nous présentons la méthode que nous avons proposée au concours RAW Floorplanning Design Contest, organisée à l'occasion du 25ème anniversaire du Reconfigurable Architectures Workshop (RAW), tenu en conjonction de la conférence IPDPS'18. La méthode de résolution est une procédure de recuit simulé multi-start, qui a obtenu la première place au concours.

## 2 Modélisation du problème

Le problème est décrit précisément sur le site web du concours [2]. Nous présentons une reformulation mineure qui correspond mieux aux besoins de notre méthode.

La représentation choisie du FPGA est une matrice rectangulaire constituée de *blocs*, chaque bloc englobant un ensemble de ressources disponibles d'un type donné (CLB, BRAM, DSP), des ressources inutilisées ou des ressources réservées. Tous les blocs sont de même largeur  $B_w$  et hauteur  $B_h$ .

Le circuit est modélisé par un ensemble  $R$  de *régions* à placer sur le FPGA. À chaque région  $r \in R$  est associé un ensemble de demandes en ressources exprimé en nombre de blocs de ressources des différents types possibles. Un placement valide pour la région  $r$  est un rectangle  $Pl(r) = (x_0, y_0, x_1, y_1)$  qui recouvre, pour chaque ressource du FPGA, un nombre supérieur ou égal de blocs de ce type de ressources requis par la région et ne couvre aucun bloc réservé.

Certaines régions peuvent être de type « partiellement reconfigurable » (PR), alors que les autres sont des régions dites « statiques ». Le placement d'une région PR doit respecter, si l'on subdivise la matrice FPGA en *tuiles* dont la largeur est d'un bloc et dont la *hauteur de tuile* est une constante donnée en entrée du problème, la contrainte d'être inscrit dans une bande horizontale de tuiles. De plus, les côtés gauche et droit d'un placement pour une région PR doivent respecter le fait d'être situés sur des sous-ensembles de colonnes prédéterminées.

Un placement valide de l'ensemble des régions doit être tel que le placement de deux régions ne se recouvrent pas (n'ont pas de bloc en commun), et deux régions PR n'ont pas de tuile en commun.

À chaque région  $r \in R$  est associé un ensemble  $P_r$  d'au moins zéro ports d'entrée/sortie, chacun caractérisés par un tuple  $(x, y, nc)$  comprenant les coordonnées du port dans la matrice

du FPGA, ainsi que le nombre de connexions entre ce port à la région. On en déduit un coût lié aux longueurs de fils d'entrée/sortie :

$$IO_{cost} = \sum_{\substack{r \in R \\ (x_0, y_0, x_1, y_1) \\ = Pl(r)}} \sum_{(x, y, nc) \in P_r} \left( \left| \frac{x_0 + x_1}{2} - \left( x + \frac{1}{2} \right) \right| \cdot B_w + \left| \frac{y_0 + y_1}{2} - \left( y + \frac{1}{2} \right) \right| \cdot B_h \right) \cdot nc.$$

À chaque paire de régions  $r_1, r_2 \in R$  est associé un coût d'interconnexion  $IC_{r_1 r_2}$  qui représente un volume de communication entre ces deux régions. Le coût d'interconnexion est le suivant :

$$IC_{cost} = \sum_{\substack{r_1 \in R \\ (x_0, y_0, x_1, y_1) \\ = Pl(r_1)}} \sum_{\substack{r_2 \in R \\ (x'_0, y'_0, x'_1, y'_1) \\ = Pl(r_2)}} \left( \left| \frac{x_0 + x_1}{2} - \frac{x'_0 + x'_1}{2} \right| \cdot B_w + \left| \frac{y_0 + y_1}{2} - \frac{y'_0 + y'_1}{2} \right| \cdot B_h \right) \cdot IC_{r_1 r_2}.$$

À ces coûts s'ajoutent un coût de surface  $A_{cost}$  qui correspond au nombre de total de blocs couverts par le placement des régions.

Dans notre reformulation nous relâchons toutes les contraintes, et les remplaçons par un coût d'infaisabilité  $Inf_{cost}$  qui cumule, pour chaque contrainte non respectée, le carré du nombre de blocs en conflit par rapport à la contrainte. Cette manière d'intégrer un coût de non faisabilité dans la fonction objectif se rapproche de l'étape initiale d'une méthode de pénalité [3].

La fonction objectif à minimiser est une somme pondérée des coûts définis précédemment :

$$Obj = B \cdot Inf_{cost} + AW \cdot A_{cost} + WW \cdot (IO_{cost} + IC_{cost})$$

### 3 Méthode de résolution

Nous résolvons ce problème grâce à une méthode de recuit simulé [1]. La solution initiale est un placement aléatoire des régions de taille de 1 bloc, qui ne se recouvrent pas et ne couvrent pas de blocs réservés. La fonction de voisinage effectuée, au hasard, soit la modification de plus ou moins 1 de l'une des coordonnées du rectangle du placement de l'une des régions, soit l'échange de deux régions.

Comme nous n'avons pas de garantie d'obtenir une solution réalisable (telle que  $Inf_{cost} = 0$ ), nous augmentons les chances d'obtenir en solution réalisable en adoptant une approche *multi-start* qui a en outre l'avantage de diversifier la recherche de solutions. Les exécutions multiples de l'algorithme étant indépendantes, elles peuvent être effectuées en parallèle. La solution retenue parmi les solutions obtenues est la solution réalisable de coût minimum.

Cette méthode a résolu avec succès l'ensemble des instances fournies par les organisateurs du concours RAW Floorplanning Design Contest. Les résultats nous ont permis d'atteindre la première place de ce concours.

### Références

- [1] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598) :671–680, 1983.
- [2] Marco Rabozzi and Marco D. Santambrogio. RAW Floorplanning Design Contest. <http://raw-floorplanning-contest.necst.it/>.
- [3] Alice E. Smith and David W. Coit. Penalty functions. In *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1996.