

# Une Approche itérative pour l'allocation des tâches sur réseaux réguliers

Clémentin Tayou Djamegni\* — Patrice Quinton\*\* — Sanjay Rajopadhye\*\*\*  
— Tanguy Risset\*\*\*\* — Maurice Tchuenté\*\*\*\*\*

\* Faculté des Sciences, B.P 069 Dschang, Cameroun, dtayou@yahoo.com

\*\* IRISA, Campus de Beaulieu, 35042 Rennes-Cedex, France

\*\*\* Colorado State University, Fort Collins CO, USA 80523-1873

\*\*\*\* LIP, ENS-Lyon, 69364 Lyon Cedex 07, France

\*\*\*\*\* Faculté des Sciences, Yaoundé, Cameroun

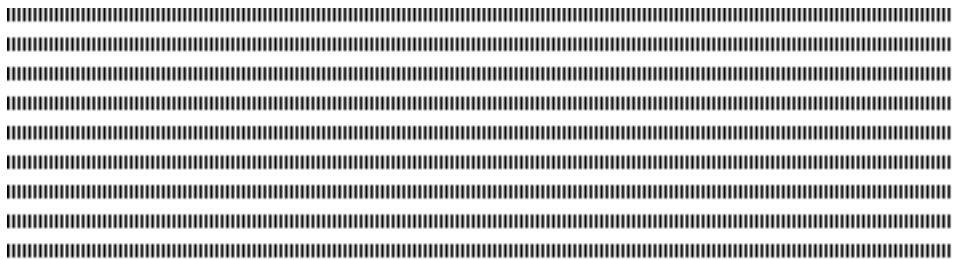


**RÉSUMÉ.** Ce papier introduit formellement une nouvelle approche d'allocation des tâches d'un graphe de dépendences associé à une fonction de temps affine aux processeurs d'une architecture régulière. L'approche proposée ici est basée sur des transformations par re-indexation qui accroissent le parallélisme potentiel de la plus faible des méthodes d'allocation, la méthode de projection. L'approche proposée est automatisable. De plus, elle conduit à des résultats de complexité plus intéressants que ceux donnés par d'autres approches.

**ABSTRACT.** This paper address the issue of mapping DAGs associated with affine timing functions onto regular arrays. This transformation is used in the design (or compilation) of highly parallel embedded systems such as VLSI systolic arrays. In this paper, we formally introduce an allocation method based on a pre-processing by re-indexation that transforms the initial DAG into a new one that enables the projection method to minimize the number of processors along a number of directions. Compared to previous allocation methods this new allocation method provides better results, in term of the overall number of processors. Furthermore, for two-dimensional DAGs it systematically leads to space-optimal designs. For DAGs of upper dimension it systematically leads to designs for which the space complexities are bounded by the best space complexity that both the projection method and the so-called grouping method can give for the initial DAG.

**MOTS-CLÉS :** Système embarqué et parallèle; réseau régulier; fonction de temps; fonction d'allocation; complexité en temps et en surface; re-indexation; graphe de dépendences

**KEYWORDS :** parallel embedded system; regular array; timing function; allocation function; space-time complexity; re-indexation; dependence graph



## 1. Introduction

Ce papier étudie le problème d'allocation des tâches d'un graphe de dépendances associé à une fonction de temps affine sur réseaux réguliers, avec un accent particulier sur l'optimisation en nombre de processeurs (PEs). C'est ainsi que pour situer notre contribution par rapport à l'état de l'art, nous présentons les méthodes d'allocation précédemment proposées :

1) **Méthode de Projection** [13, 15, 18]. Ceci correspond aux allocations linéaires. Le principal défaut de cette méthode provient de la sous-utilisation des PEs du réseau [1, 3]. Dans [18, 15, 11] les auteurs utilisent des techniques de programmation en nombres entiers pour trouver une direction de projection optimale. Cependant, l'allocation linéaire correspondante peut ne pas être optimale.

2) **Méthode de Groupement Linéaire(ou Clustering)** [3]. L'objectif de cette méthode est de réduire le nombre de PEs d'un réseau obtenu par la méthode de projection en fusionnant des groupes de  $|\lambda^t \xi|$  PEs voisins dont les dates d'occupation sont distinctes. Ce qui permet de réduire le nombre de PEs d'un facteur de  $|\lambda^t \xi|$ . où  $\lambda$  est le vecteur d'ordonnancement et  $\xi$  la direction de projection. Le nouveau réseau ainsi obtenu est localement connecté et chaque processeur est actif à chaque instant entre ses dates de début et de fin d'exécution des tâches. Cependant, bien que cette propriété soit intéressante, elle n'implique pas l'optimalité en nombre de PEs. Un autre défaut de cette méthode provient de ce qu'elle s'applique uniquement dans le cas où  $|\lambda^t \xi| \geq 2$ .

3) **Méthode de ré-allocation** [1]. Comme dans la méthode de groupement, l'objectif de cette méthode est de réduire le nombre de PEs d'un réseau obtenu par la méthode de projection. Pour ce faire, le réseau initial est partitionné en un ensemble de segments parallèles à une direction appelée *direction de partitionnement*. Les tâches de chaque segment sont à nouveau affectées aux PEs de ce segment de manière à minimiser leur nombre. Il va s'en dire que cette méthode cherche à minimiser le nombre de PEs sur chaque  $\langle \xi, p \rangle$ -tranche (Section 2) du graphe de dépendances, où  $\xi$  est la direction de projection et  $p$  la direction de partitionnement. Cette approche garantit l'optimalité en nombre de PEs sur chaque segment. Cependant, ceci n'implique pas que le réseau est optimal en surface.

4) **Méthode de Groupement Généralisé** [17, 3]. Partant d'un graphe de dépendances associé à une fonction de temps affine, on détermine une tranche d'ordonnancement  $M$  de cardinalité maximale et on affecte les tâches aux PEs. Ensuite, on fusionne des PEs qui ne sont pas actifs au même moment et qui sont tels que l'un d'eux a reçu une tâche de  $M$ . Ceci permet de minimiser le nombre de PEs. Malheureusement les regroupements des PEs distants introduisent souvent des communications distantes ou des irrégularités dans le réseau, ce que nous cherchons à éviter dans ce papier. Un autre défaut de cette méthode provient de ce qu'elle est difficile à automatiser. Ce qui constitue un frein pour son intégration dans les outils de synthèse (semi)-automatique.

5) **Méthode de Partitionnement** [14, 16]. Le point de départ de cette méthode est de partitionner le graphe de dépendance initial  $G$  en un ensemble de sous-graphes qui sont construits de deux manières : 1. Soit directement en considérant l'intersection de  $G$  avec un ensemble d'hyperplans parallèles. Dans le cas où les hyperplans sont de dimension deux, l'approche correspond à la méthode de ré-allocation. 2. Soit de manière récurrente en considérant  $G$  comme un empilement de couches successives. La  $i$ -ème couche  $C_i$  étant constituée de points de la frontière du graphe obtenu en supprimant tous les points des  $i - 1$  premières couches, i.e. les couches  $C_j, j < i$ . Ensuite les tâches de chaque sous-graphe sont affectées aux processeurs de manière à minimiser le nombre de processeurs dans chaque sous-graphe.

Toutes ces méthodes d'allocation ne garantissent pas l'optimalité en surface. La plus faible des méthodes d'allocation, en terme de capacité de réduction du nombre de PEs, est la méthode de projection, et les plus intéressantes sont les méthodes de groupement linéaire, de re-allocation et de partitionnement.

Ici, nous introduisons formellement une nouvelle approche d'allocation basée sur un pré-traitement par re-indexation qui transforme le domaine d'itération de départ en un nouveau domaine qui se prête mieux à la méthode de projection en termes de réduction du nombre de PEs. Il convient de noter que cette approche d'allocation a été introduite pour la première fois dans [7, 8], mais d'une manière intuitive. Notons également que les transformations par ré-indexation ont déjà été utilisées pour atteindre d'autres objectifs d'optimisation, notamment en parallélisation automatique [4, 5, 12]. Ici, l'objectif d'optimisation visé est l'augmentation du parallélisme de la méthode de projection.

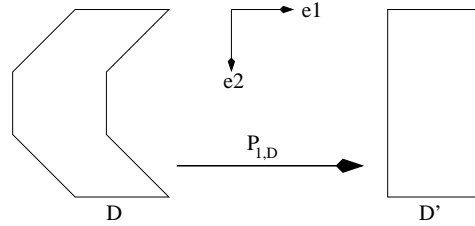
L'approche présentée ici est automatisable. Elle donne des résultats meilleurs, en termes de nombre de PEs, par rapport à d'autres méthodes d'allocation. C'est ainsi que nous établissons un *résultat de comparaison fort* entre cette approche et les méthodes de projection et de groupement linéaire. Cette nouvelle approche d'allocation constitue en réalité une réponse positive à la question de l'automatisation de la méthode de partitionnement bien que connue pour sa puissance dans la synthèse des réseaux réguliers [9, 10, 14, 16].

La suite est organisée comme suit. La section 2 introduit quelques définitions et notations. La section 3 présente formellement la nouvelle approche d'allocation. La section 4 compare cette approche à d'autres méthodes. La section 5 conclut ce papier.

---

## 2. Définitions et notations

**Définition 1** *Un vecteur est unimodulaire si ses composantes sont premiers entre eux. Étant donné un vecteur unimodulaire  $s$ , un domaine  $D$  est dit convexe suivant la direction  $s$  si son intersection avec tout segment parallèle à la direction  $s$  correspond soit à un ensemble vide, soit à un segment. Le domaine  $D$  est dit convexe s'il est convexe suivant toutes les directions. Si  $D$  est convexe suivant une direction  $s$ , nous appelons frontière*



**Figure 1.** Illustration de la fonction de re-indexation  $P_{1,D}$

de  $D$  suivant la direction  $s$ , tout sous ensemble de  $D$  défini par  $Front(D, s) = \{z \in D \mid z + s \notin D\}$ .

**Définition 2** Étant donné deux vecteurs linéairement indépendants  $v_1$  et  $v_2$ , nous appelons tranche de direction  $\langle v_1, v_2 \rangle$ , notée  $\langle v_1, v_2 \rangle$ -tranche, d'un domaine  $D$  toute intersection de  $D$  avec un plan de direction  $\langle v_1, v_2 \rangle$ , i.e parallèle aux vecteurs  $v_1$  et  $v_2$ . Un tel plan est appelé plan de support de la tranche.

**Proposition 1** [10] Une fonction de re-indexation de la forme  $z \rightarrow Uz + z_0$ , où  $U$  est une matrice constante d'ordre  $n$  et  $z_0$  un point constant de  $Z^n$ , transforme la fonction de temps affine  $[\lambda, \alpha]$ ,  $t(z) = \lambda^t z + \alpha$ , en une nouvelle fonction de temps affine  $[U^{-t}\lambda, \alpha - \lambda^t U^{-1} z_0]$ .

$seg(z_1, z_2)$  désigne le segment d'extrémités  $z_1$  et  $z_2$ . Étant donné un vecteur unimodulaire  $s$  et un point  $z$  d'un domaine  $D$  de dimension  $n$  et convexe suivant  $s$ , nous notons par  $z_{s,D}$  l'unique point de  $Front(D, s)$  appartenant à la ligne définie par le vecteur  $s$  et le point  $z$  et par  $z_s$  la projection du point  $z$  suivant  $s$  sur l'hyperplan de dimension  $(n - 1)$  qui est orthogonal à  $s$  et qui contient le point nul  $(0, 0, 0, \dots, 0)^t \in Z^n$ .

### 3. Présentation de la méthode

Ici nous montrons comment appliquer la re-indexation pour obtenir des domaines d'itération bornés et de dimension  $n$  à partir desquels on peut obtenir par projection des réseaux qui minimisent le nombre de PEs suivant plusieurs directions. Partant du fait que la méthode de projection suppose que les points ayant une même date d'exécution appartiennent à un même hyperplan de dimension  $(n-1)$  qui est non-parallèle à la direction de projection, nous considérons uniquement les fonctions de re-indexation qui assurent cette propriété.

Notons  $D$ , un domaine qui borné et de dimension  $n$ . Posons  $t = [\lambda, \alpha]$ ,  $t(z) = \lambda^t z + \alpha$ . Pour simplifier la présentation, nous supposons que  $t$  est normalisée, i.e.  $\lambda$  est unimodulaire. Notons  $\langle e_1, e_2, e_3, \dots, e_n \rangle$  la base canonique de  $Z^n$ . Comme dans plusieurs

travaux liés à cette question, nous supposons que les tranches d'ordonnement sont convexes.

### 3.1. Cas des Domaines de Dimension 2

Ici les tranches d'ordonnement sont des segments de  $D$ . Elles seront appelées dans la suite *segments d'ordonnement*. Suivant la direction de  $\lambda$ , nous considérons deux cas.

#### Cas 1. $\lambda$ est colinéaire à un vecteur $e_l$ de la base canonique

Soit  $s$  un vecteur qui désigne  $e_h$  ou  $-e_h$ , où  $h \neq l$ . Nous présentons ici un pré-traitement par re-indexation qui permet d'obtenir une allocation optimale par la méthode de projection. L'idée de base de la re-indexation est de translater les segments d'ordonnement suivant la direction  $s$  de manière à localiser les points de  $Front(D, s)$  sur l'axe défini par  $e_l$ . Pour ce faire, nous utilisons la fonction de re-indexation  $P_{h,D}$  :

$$P_{h,D}(z) = P_{h,D}(z_{s,D}) + (z - z_{s,D})$$

Ceci est équivalent à :

$$P_{h,D}(z) = z_s + (e_h^t(z - z_{s,D}))e_h \quad [1]$$

la fonction  $P_{h,D}$  correspond à une translation par morceaux. Il est facile de voir que  $P_{h,D}$  correspond à une fonction de re-indexation valide, i.e. que  $P_{h,D}$  est injective. Nous avons  $P_{h,D}(z) = (e_1^t(z - z_{s,D}), e_2^t z)^t$  pour  $h = 1$  et  $P_{h,D} = (e_1^t z, e_2^t(z - z_{s,D}))^t$  pour  $h = 2$ .

La fonction de re-indexation  $P_{1,D}$  est illustrée à la figure 1. Dans cette figure, la projection du domaine  $D' = P_{1,D}(D)$  suivant la direction  $e_2$  conduit à une allocation optimale. Dans cet exemple, la fonction de re-indexation  $P_{1,D}$  améliore le parallélisme de la méthode de projection par rapport au graphe de dépendances initial. En effet, la projection du domaine initial  $D$  suivant n'importe quelle direction ne conduit pas à une allocation optimale.

**Proposition 2** [10] Nous avons soit  $e_h^t(z - z_{s,D}) \geq 0$  pour tout  $z \in D$ , soit  $e_h^t(z - z_{s,D}) \leq 0$  pour tout  $z \in D$ .

**Proposition 3** [10] La projection de  $D'$  suivant  $e_l$  conduit à une allocation optimale.

Dans le réseau correspondant, les opérations d'E/S sont effectuées par un PE localisé à une des extrémités du réseau si les points associés à ces opérations appartiennent à  $Front(D, s)$ .

#### Cas 2. Aucun vecteur de la base canonique n'est colinéaire à $\lambda$ .

Nous ramenons problème au cas précédent. A cet effet, nous appliquons au domaine initial  $D$  une fonction de re-indexation  $q$  qui conduit à un nouveau vecteur d'ordonnement qui est parallèle à  $e_l$ . Pour ce faire, notons d'abord que  $e_l = U^{-t}\lambda$  pour une

certaine matrice unimodulaire  $U$  d'ordre 2 puisque les colonnes de  $e_l^t$  et celles de  $\lambda^t$  g n rent le m me treillis [?]. De la proposition 1 nous posons  $q : z \rightarrow Uz$ . Le nombre de PEs est minimis  en appliquant la strat gie d'allocation d finie pour le cas 1 au nouveau domaine d'indexation  $D' = q(D)$ .

### 3.2. Cas des domaines de dimension $n \geq 3$

 tant donn  deux vecteurs unimodulaires et lin airement ind pendants  $v_1 = x_1e_1 + \dots + x_n e_n$  et  $v_2 = y_1e_1 + \dots + y_n e_n$ , nous montrons comment les t ches des  $\langle v_1, v_2 \rangle$ —tranches peuvent  tre s par ment affect es aux PEs de mani re   minimiser le nombre de PEs sur chaque  $\langle v_1, v_2 \rangle$ —tranche. A cet effet, comme dans la m thode de partitionnement [14, 16], notre point de d part est de partitionner  $D$  en  $\langle v_1, v_2 \rangle$ —tranches not es  $D_1, D_2, D_3, \dots, D_r$ . Cependant, pour chercher une allocation qui minimise le nombre de PEs sur chaque  $D_i$ , nous proc dons d'une diff rente. Nous effectuons un pr -traitement par re-indexation qui fournit une direction de projection qui affecte s par ment les t ches de chaque tranche  $D_i$  aux PEs de mani re   minimiser leur nombre pour chaque tranche. Notons que si  $v_1$  et  $v_2$  sont parall les aux surfaces d'ordonnancement le nombre minimal de PEs requis par chaque  $\langle v_1, v_2 \rangle$ —tranche est  gal   son cardinal. Ceci nous am ne   supposer que les  $\langle v_1, v_2 \rangle$ —tranches ne sont pas parall les aux surfaces d'ordonnancement, i.e  $\lambda^t v_1 \neq 0$  ou  $\lambda^t v_2 \neq 0$ . Dans ce qui suit, nous supposons que  $\lambda^t v_2 \neq 0$ . Suivant la direction du vecteur d'ordonnancement  $\lambda$  nous consid rons deux cas.

#### Cas 1. $\lambda$ est colin aire   un certain $e_l$

**Sous-Cas 1.1.** Les  $\langle v_1, v_2 \rangle$ —tranches sont parall les   un certain  $e_h$  avec  $h \neq l$ .

Supposons que  $s$  d signe soit  $e_h$ , soit  $-e_h$ . Ici nous g n ralisons la fonction de re-indexation d finie par l' quation (1). L'id e qui sous-tend cette g n ralisation est de localiser les points de  $Front(D, s)$  sur l'hyperplan de dimension  $n - 1$  et d' quation cart sienne  $e_h^t z = 0$  de mani re   maintenir chaque point de  $D$  sur sa surface d'ordonnancement et sur son plan de support de sa  $\langle v_1, v_2 \rangle$ —tranche. Pour ce faire, nous utilisons la fonction de re-indexation  $P_{h,D}$  :

$$P_{h,D}(z) = z_s + (e_h^t(z - z_{s,D}))e_h = (e_1^t z, \dots, e_{h-1}^t z, e_h^t(z - z_{s,D}), e_{h+1}^t z, \dots, e_n^t z)^t \quad [2]$$

Il est facile de voir que  $P_{h,D}$  est une fonction de re-indexation valide. Elle agit de mani re similaire sur chaque  $\langle v_1, v_2 \rangle$ —tranche. Sur  $D_i$  elle ram ne tous les points de  $Front(D_i, s)$    l'intersection du plan de support de  $D_i$  et de l'hyperplan de dimension  $n - 1$  et d' quation cart sienne  $e_h^t z = 0$ . Ceci entra ne que  $P_{h,D}(Front(D_i, s))$  est parall le au vecteur  $v_2 - y_h e_h$ .

**Proposition 4** [10] *La projection de  $D' = P_{h,D}(D)$  suivant la direction  $v_2 - y_h e_h$  affecte s par ment les points de chaque  $\langle v_1, v_2 \rangle$ —tranche =  $\langle e_h, v_2 - y_h e_h \rangle$ —tranche aux PEs de mani re   minimiser leur nombre sur chaque  $\langle e_h, v_2 - y_h e_h \rangle$ —tranche.*

La proposition 4 montre que la projection de  $D' = P_{h,D}(D)$  suivant la direction  $v_2 - y_h e_h$  conduit à un réseau dont le nombre de PEs est minimisé suivant la direction  $e_h$ , i.e. pour chaque ligne de PEs de direction  $e_h$  il existe une date pour laquelle tous les PEs sont simultanément actifs. Comme mentionné plus haut dans le cas particulier où  $D$  est de dimension 2, le réseau obtenu est tel que les opérations d'E/S sont effectuées par des PEs situés à la frontière du réseau si toutes les tâches d'E/S appartiennent à  $Front(D, s)$ . Une telle propriété est chère aux implémentations VLSI.

**Sous-Cas 1.2.** Les  $\langle v_1, v_2 \rangle$ —tranches ne sont pas parallèles à un certain  $e_h$  où  $h \neq l$

Ici nous effectuons une re-indexation qui ramène le problème au sous cas 1.1. L'idée qui sous-tend cette re-indexation est de transformer une tranche  $D_i$  en une nouvelle tranche parallèle à un certain  $e_h$ , avec  $h \neq l$ , et ceci de manière à laisser inchangé le vecteur d'ordonnement  $\lambda$  [10].

**Cas 2. Aucun vecteur de la base canonique n'est colinéaire à  $\lambda$ .**

Nous effectuons une re-indexation pour ramener le problème au cas 1. Pour ce faire, nous considérons une fonction de re-indexation  $q$  qui conduit à un nouveau vecteur d'ordonnement qui est parallèle à un certain  $e_l$  [10].

---

## 4. Comparaison avec d'autres méthodes

Soit  $\lambda$  un vecteur d'ordonnement. Soient  $v_1 = x_1 e_1 + \dots + x_n e_n$  et  $v_2 = y_1 e_1 + \dots + y_n e_n$ , deux vecteurs unimodulaires et linéairement indépendants tels que les  $\langle v_1, v_2 \rangle$ —tranches ne sont pas parallèles aux surfaces d'ordonnement. Dans toute la suite,  $NPE(\xi, D)$  désigne le nombre de PEs obtenu en projetant le domaine  $D$  suivant la direction  $\xi$  et  $NPE(\xi, D, \gamma)$  (resp.  $NB(\xi, D, \gamma)$ ) désigne le nombre de PEs obtenu en appliquant la méthode de groupement linéaire (resp. la méthode de re-allocation) suivant la direction  $\gamma$  au réseau obtenu en projetant le domaine  $D$  suivant la direction  $\xi$ .

**Proposition 5** [10] *Si  $\lambda$  est colinéaire à un certain  $e_l$  et les  $\langle v_1, v_2 \rangle$ —tranches sont parallèles à un certain  $e_h$  avec  $l \neq h$ , alors nous avons  $NPE(v_2 - y_h e_h, P_{h,D}(D)) \leq NPE(v_2, D, e_h) \leq NPE(v_2, D)$ .*

### 4.1. Deux résultats de comparaison forts

Ici, nous établissons deux résultats de comparaison forts entre la nouvelle approche d'allocation et les méthodes de projection et de groupement linéaire. Pour simplifier la présentation, nous supposons sans perte de généralité que le vecteur d'ordonnement satisfait  $\lambda = e_n$ . Il vient que  $t = [e_n, \alpha]$ . Notons  $D^{(0)} = D$ ,  $D^{(h)} = P_{h,D^{(h-1)}}(D^{(h-1)})$  avec  $h \in \{1, 2, \dots, n-1\}$  et supposons que  $D^{(h-1)}$  est convexe suivant la direction  $e_h$ . Considérons un vecteur unimodulaire  $\xi = \xi_1 e_1 + \xi_2 e_2 + \dots + \xi_n e_n$  qui définit

une direction de projection valide. Puisque  $\lambda = e_n$ , nous avons  $\xi_n \neq 0$ . Notons  $\xi^{(h)}$ ,  $h \in \{0, 1, 2, \dots, n-1\}$ , un vecteur unimodulaire qui est colinéaire au vecteur  $\xi_{h+1}e_{h+1} + \xi_{h+2}e_{h+2} + \dots + \xi_n e_n$ .

**Corollaire 1** [10]  $NPE(D^{(n-1)}, \lambda) \leq NPE(D^{(0)}, \xi)$  pour tout  $\xi$  tel que  $\lambda^t \xi \neq 0$ .

Ce corollaire exprime donc un résultat de comparaison fort entre la nouvelle méthode d'allocation et la méthode de projection.

**Théorème 1** [10]

$$NPE(D^{(n-1)}, \lambda) \leq \frac{1}{|\lambda^t \xi|} NPE(D^{(0)}, \xi) + \sum_{h=0}^{h=n-2} \Theta(nl_h)$$

pour tout  $\xi$  tel que  $\lambda^t \xi \neq 0$  où  $nl_h$  est le nombre de  $\langle e_{h+1}, \xi^{(h)} \rangle$ -tranches de  $D^{(h)}$ .

Le théorème 1 exprime un résultat de comparaison fort entre la nouvelle méthode d'allocation et la méthode de groupement linéaire. Comparé à d'autres méthodes d'allocation, il n'existe pas de résultat de comparaison fort et connu entre la méthode de re-allocation et la méthode de groupement linéaire, ni entre la méthode de partitionnement et la méthode de groupement linéaire. De même, il n'existe pas de résultat de comparaison fort et connu entre la méthode de re-allocation et la méthode de projection, ni entre la méthode de partitionnement et la méthode de projection.

---

## 5. Conclusion

Dans ce papier, nous avons introduit formellement une nouvelle approche d'allocation des tâches d'un graphe de dépendances aux PEs d'une architecture régulière. Cette nouvelle méthode d'allocation est basée sur un pré-traitement par ré-indexation guidé par des propriétés de convexité du graphe de dépendances.. Cette nouvelle méthode constitue en réalité une réponse positive à la question de l'automatisation de la méthode de partitionnement bien que connue pour sa puissance dans la synthèse des réseaux réguliers [9, 10, 14, 16]. Le pré-traitement améliore le parallélisme de la méthode de projection par rapport au domaine d'indexation initial. Il permet aussi d'affecter les noeuds d'entrée et de sortie aux PEs localisés à la frontière du réseau. Cette possibilité intéressante permet d'éliminer un temps supplémentaire dû au chargement et au déchargement du réseau. A cet effet, il convient de noter que les problèmes de chargement/déchargement du réseau peuvent ralentir significativement l'exécution du réseau s'ils ne sont pas gérés convenablement [6, 7]. Cette nouvelle méthode est simple et peut être intégrée dans les outils de synthèse (semi)-automatique.

---



## 6. Bibliographie

- [1]P. Clauss. Optimal mapping of systolic algorithms by regular instruction shifts. IEEE Int. Conf. on Application Specific Array Processors, (1994) 224-235.
- [2]P. Clauss. Synthèse d'algorithmes systoliques et implantation optimale en place sur réseaux de processeurs synchrones. Ph.D Thesis, Univ. of Franche-Compté, (May 1990).
- [3]P. Clauss, C. Mongenet and G. R. Perrin. Calculus of space-optimal mappings of systolic algorithms on processor arrays. IEEE Int. Conf. on ASAP, pp. 591-602, 1990.
- [4] A. Darte, Robert, Vivien, Scheduling and Automatic Parallelization, *Birkhäuser Boston*, 2000.
- [5] A. Darte, Robert, Vivien, Loop parallelization algorithms, in *volume 1808 of LNCS on Compiler Optimizations for Scalable Parallel Systems : Languages, Compilation Techniques, and Run Time Systems*, Dharma P. Agrawal and Santosh Pande editors, Springer Verlag, (2001) 141-172.
- [6] C.T. Djamegni, Quinton, Rajopadhye, Risset, Derivation of systolic algorithms for the algebraic path problem by recurrence transformations, *Parallel Comput.*, 26 (11) (2000)1429-1445.
- [7]C. T. Djamegni, Synthesis of space-time optimal systolic algorithms for the Cholesky factorization, *Discrete Mathematics and Theoretical Computer Science*, 5 (2002) 109-120.
- [8]C.T. Djamegni, Execution d'un graphe cubique de taches sur un reseau bi-dimensionnel et asymptotiquement optimal, *African Revue in Informatics and Mathematics Applied*, 4, (2006) 53-65.
- [9]C. T. Djamegni, Contribution to the synthesis of optimal algorithms for regular arrays, *Thèse de Doctorat de troisième cycle*, Université de Yaoundé I, (Dec. 1997).
- [10]C. T. Djamegni, Méthode d'optimisation pour la synthèse d'architectures régulières, *Thèse de Doctorat d'Etat*, Département d'Informatique, Université de Yaoundé I, (Dec. 2005).
- [11]K.N. Ganapathy, B.W. Wah, Optimal synthesis of algorithm-specific lower-dimensional processor arrays, *IEEE Trans. on Parallel and Dist. System*, 7 (3) (March 1996) 274-287.
- [12] A.W. Lim, M.S. Lam, Maximizing parallelism and minimizing synchronization with affine partitions, *Parallel Computing*, 24 (3-4) (May 1998) 445-475
- [13] P. Quinton and V. V. Dongen. The mapping of linear equations on regular arrays. *J. VLSI Signal Processing* 1(2) :95-113, 1989.
- [14] I. Sakho and M. Tchunte. Methode de conception d'algorithmes paralleles pour reseaux reguliers. *Technique et Science Informatiques*, 8 (1989), pp. 63-72.
- [15] W. Shang, J.A.B. Fortes, On mapping of uniform dependence algorithms into lower dimensional processors arrays, *IEEE Trans. on Parallel and Distributed Systems*, 3 (1992) 350-363.
- [16] M. Tchuenté. Parallel Computation on regular arrays. Manchester University Press and John Wiley Sons, 1992.
- [17]J.C. Tsay, P.Y. Chang, Design of space-optimal regular arrays for algorithms with linear schedules, *IEEE Trans. on Computers*, 44 (5) (1995) 683-694.
- [18]Y. Wong, J. M. Delosme, Space-optimal linear processor allocation for systolic arrays synthesis, In Proc. *Sixth Int'l Parallel Processing Symp.*, (1992) 275-282.