Les Réseaux Associatifs,

un modèle de programmation à parallélisme de données, pour algorithmes et données irréguliers, à primitives de calcul asynchrones

Bertrand Ducourthial
Laboratoire de Recherche en Informatique
Institut d'Électronique Fondamentale
Université Paris-Sud FRANCE
ducourth@lri.fr

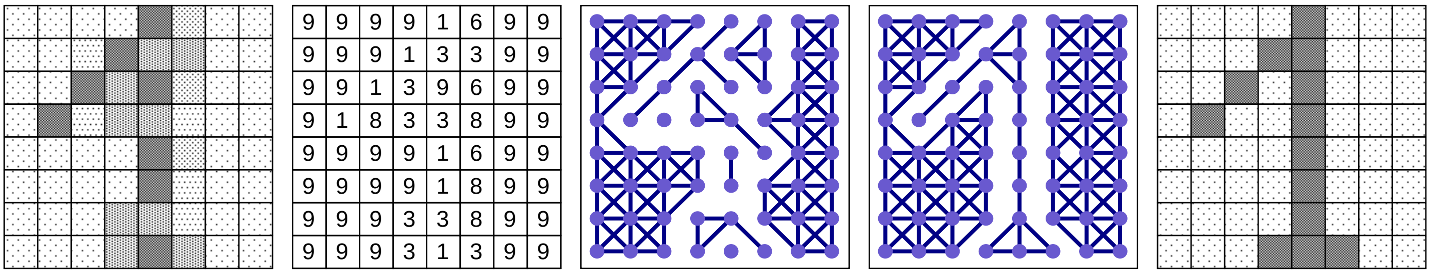
20 Janvier 1998

Soutenance de Thèse
Doctorat de Sciences, spécialité Informatique

Origine de l'étude

■ Modèle des réseaux associatifs [Mérigot 1992]

- traitement d'images et architectures spécialisées
- image = pixels + relations inter-pixels
  - data-parallélisme ⇝ image : variable parallèle (*pvar*)
  - représentation des "objets" d'une image par un sous-réseau, dans un réseau de base fixé (*graphe maille*)
- primitives de calcul global sur les sous-réseaux
- Maille Associative : multi-processeurs SIMD



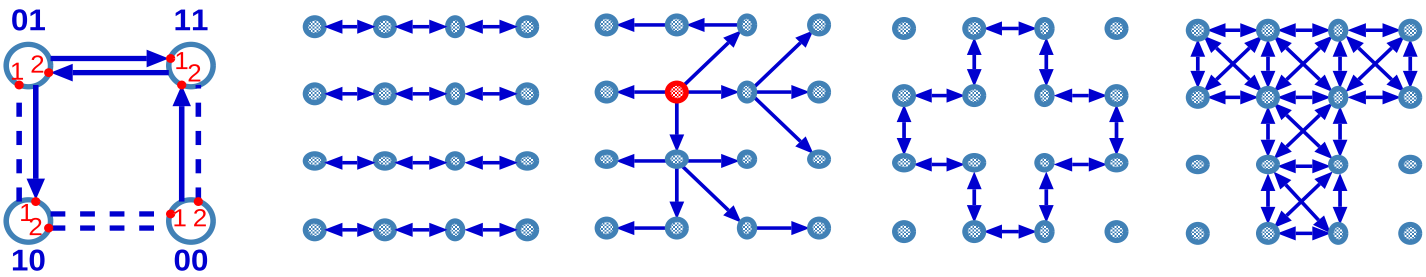
■ Objectif de la thèse

- modèle des réseaux associatifs :
  ⇝ modèle de programmation data-parallèle général

Modèle des Réseaux Associatifs

Codage des objets irréguliers

- masques sur les arcs entrants dans un réseau de base



Nouvelle primitive

- différences entre modèle abstrait et Maille Associative
  ⇝ `direct-association`
- primitive data-parallèle originale :
  calcul global sur un sous-réseau = calculs locaux asynchrones répétés jusqu'à stabilisation du réseau

Démarche suivie

- sous-réseaux d'un réseau de base
  ⇝ quels réseaux ? niveau d'abstraction ?
- primitive `direct-association`
  ⇝ conditions d'existence ? propriétés ?
- calculs dans un réseau ⇝ contrôle du réseau ?
- application ⇝ nouvelle algorithmique
- validation ⇝ mise en œuvre

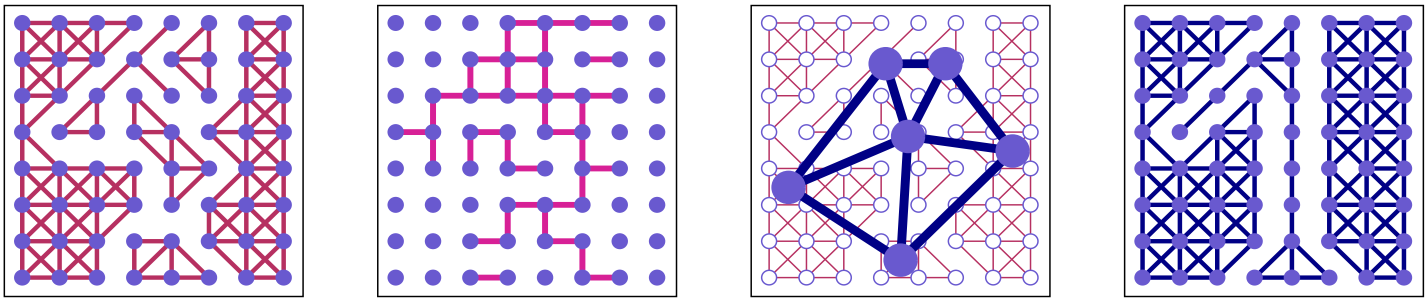
⇝ *modèle de programmation des réseaux associatifs*

Plan

- Étude de la topologie
  - Utilisation des masques
- Étude de la primitive de calcul `direct-association`
  - définition
  - exemple : composantes fortement connexes
  - conditions assurant la définition
  - r-opérateurs
  - applications aux parcours de graphes
- Étude du contrôle du réseau
  - messages, terminaison, auto-stabilisation
- Algorithmique
  - programmation dynamique : calcul sur un DAG
  - image : distance
  - géométrie : domaines de Voronoï
  - segmentation d'images en régions de Voronoï
- Mise en œuvre : ANET
  - objectifs
  - structure
  - parallélisation des primitives
  - test : r-opérateurs et grain du calcul
  - test : asynchronisme
- Conclusion

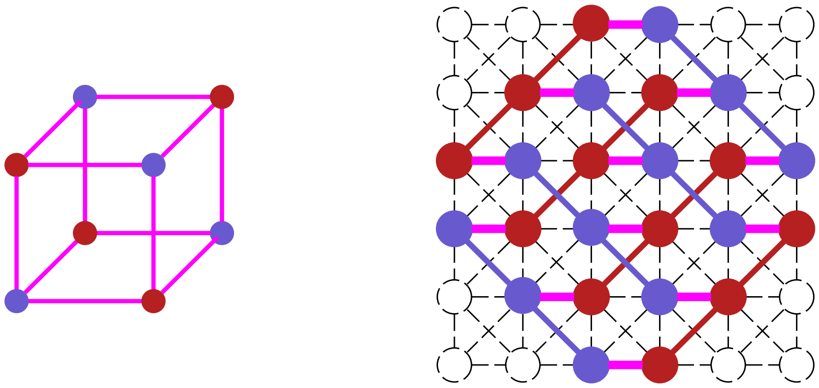
Utilisation des masques

■ Niveau d'abstraction supplémentaire : graphe virtuel



- graphe des pixels $\rightsquigarrow$ graphe des régions
- traitements de haut niveau, sur les objets de l'image
- codage par deux variables parallèles (*pvars de masques*)

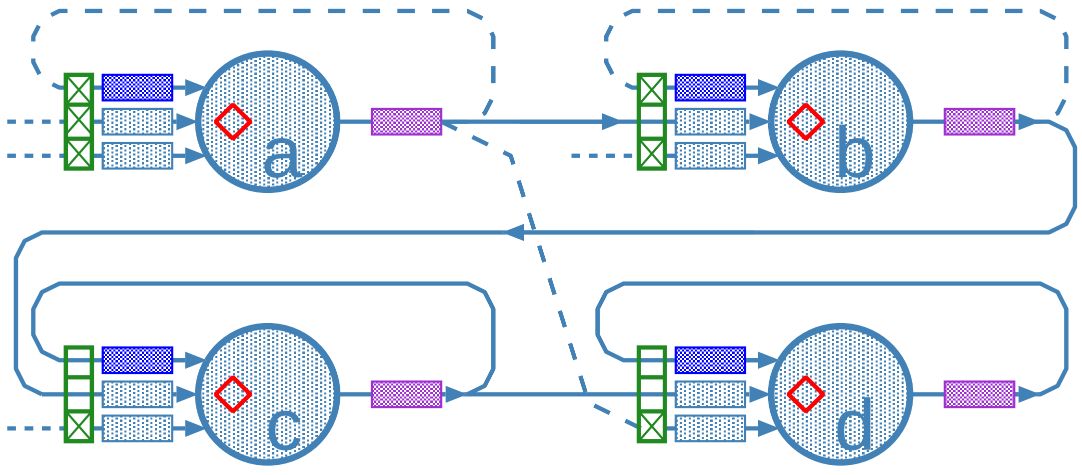
■ Plongements de graphes [RENPAR 9][soumis IEEE TC]



- graphe logique $\rightsquigarrow$ graphe de base
- diffusions asynchrones efficaces
  $\rightsquigarrow$ plongement un-vers-ensemble-connexe
- résultat : pour tout graphe logique, il existe une maille acceptant son plongement un-vers-ensemble-connexe

Primitive `direct-association`

- Réseau codé par les masques ; entités calculantes



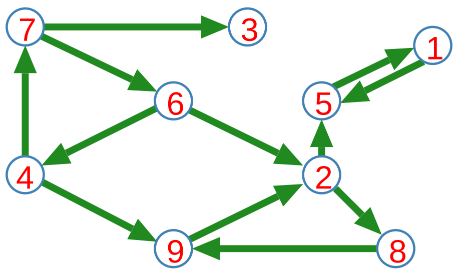
- Calcul local avec un opérateur $\diamond$ binaire ou n-aire

$x_{\text{out}} \leftarrow x_0 \diamond \cdots \diamond x_{\delta_v^-}$ $\qquad x_{\text{out}} \leftarrow \diamond\left(x_0, \ldots, x_{\delta_v^-}\right)$

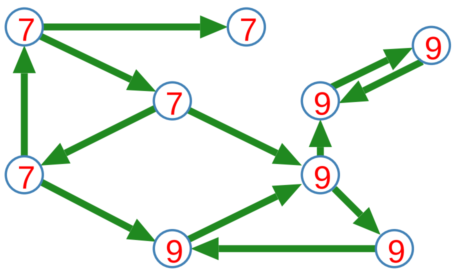
- Calcul global par coopération de calculs locaux

  - $q \leftarrow \diamond$-*direct-association*(G, p)
  - calculs locaux non coordonnés
  - fin du calcul : aucune nouvelle valeur n'est produite
    $\rightsquigarrow$ stabilisation
  - défini si terminaison et déterminisme

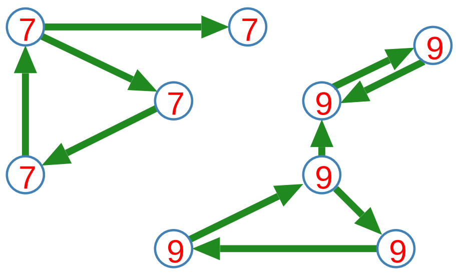
Exemple : composantes fortement connexes



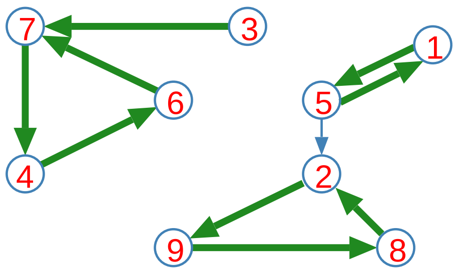
a. Graphe initial, pvar index.



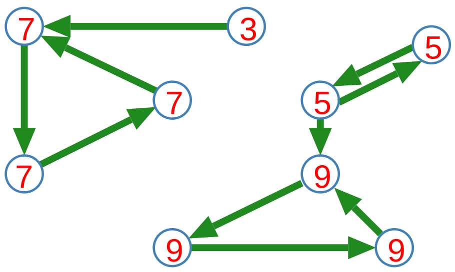
b. max-direct-association



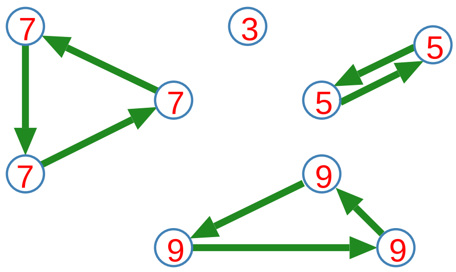
c. link-with-equals



d. Graphe inversé, pvar index.



e. max-direct-association



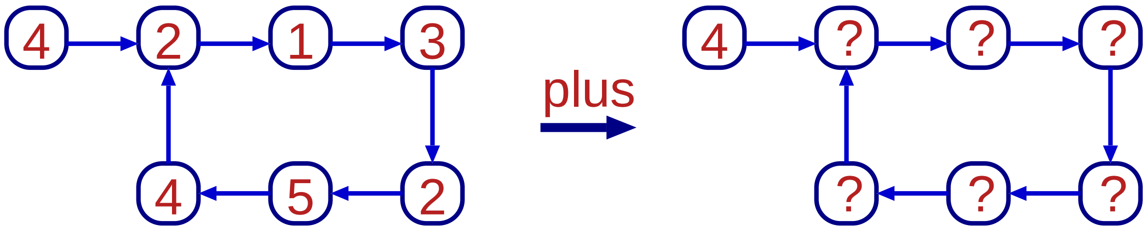
f. link-with-equals

Algorithme 1 CFC(**pvar**⟨ **masque** ⟩ G)

pvar⟨ **masque** ⟩ G_{tmp}
pvar p
répéter
    $G_{tmp} \leftarrow G$
    $p \leftarrow$ `max-direct-association(`G`,`*index*`)`
    $G \leftarrow$ `link-with-equals(`G`, `p`, `p`)`
    $p \leftarrow$ `max-direct-association(`G^{-1}`,`*index*`)`
    $G \leftarrow$ `link-with-equals(`G`, `p`, `p`)`
jusqu'à ce que $G \neq G_{tmp}$
retourner G

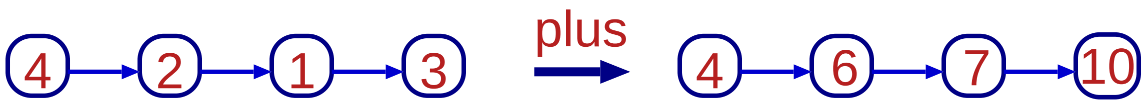
Conditions assurant la définition (1)

- `direct-association`
  - peu de contraintes $\rightsquigarrow$ facilement mise en œuvre
  - mais quels opérateurs sur quels réseaux ?
- plus-`direct-association`

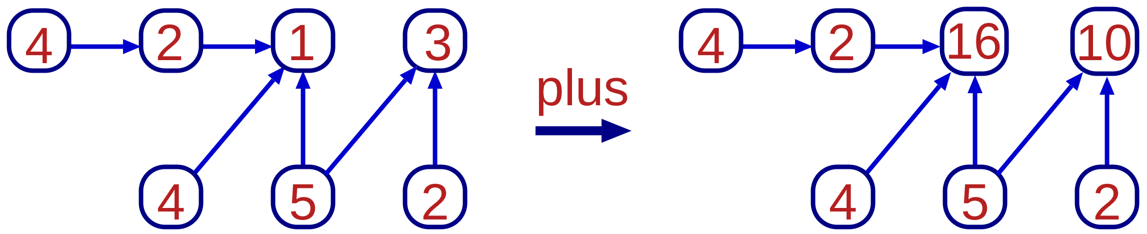


- importance de l'idempotence ($x \diamond x = x$)

  $\rightsquigarrow$ algèbres idempotentes,
  algèbres de chemins [Gondran et Minoux 1978]
- calcul préfixe [Lakshmivarahan et Dhall 1994]
  scan vector model [Blelloch 1990]

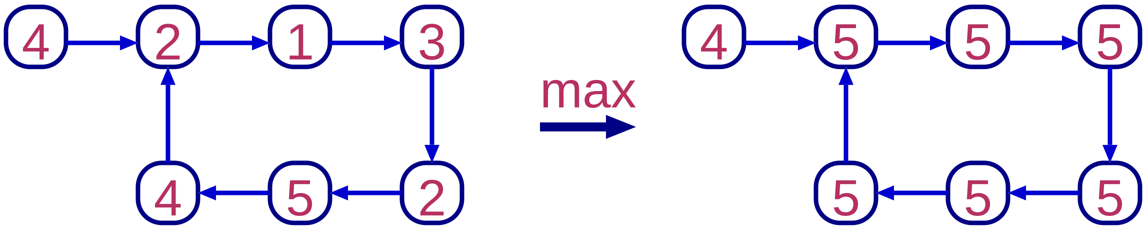


- généralisation aux *graphes préfixes* (connexions d'arborescences)



Conditions assurant la définition (2)

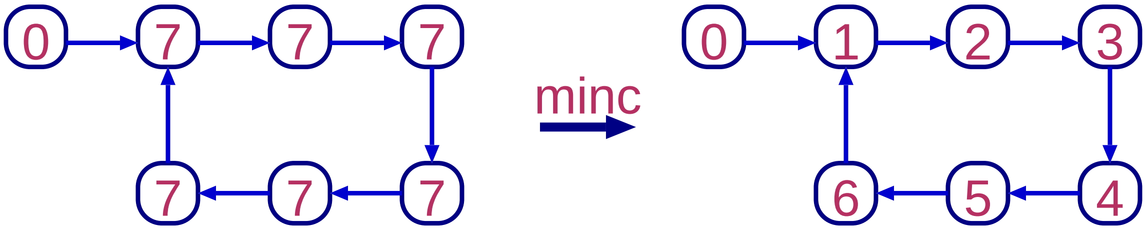
s-opérateur : associatif, commutatif et idempotent



- $\rightsquigarrow$ *total algorithms* (ou *waves*), et infimum [Tel 1991]
- exemples : max, min, or, and, $\cup$, $\cap$, ppcm, pgcd, *etc.*

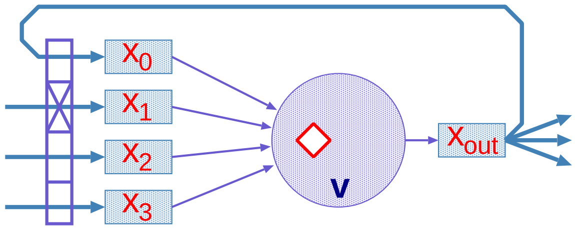
Extension des s-opérateurs

- exemple : minc$(x, y) =$ min$(x, y + 1)$



réseau	opérateur
chaîne $\rightarrow$ graphe préfixe	opérateur quelconque
graphe quelconque	s-opérateur $\rightarrow$ r-opérateur

r-opérateurs (1)



$x_{\mathrm{out}} \leftarrow x_0 \diamond \cdots \diamond x_{\delta_v^-}$ $\qquad$ $x_{\mathrm{out}} \leftarrow \diamond\left(x_0, \ldots, x_{\delta_v^-}\right)$

- s-opérateur $\oplus$ exemple : $\min(x, y)$

associatif	$(x \oplus y) \oplus z = x \oplus (y \oplus z)$
commutatif	$x \oplus y = y \oplus x$
idempotent	$x \oplus x = x$
élément neutre	$x \oplus e_{\oplus} = x$

- r-opérateur $\triangleleft$ exemple : $\mathrm{minc}(x, y) = \min(x, y+1)$

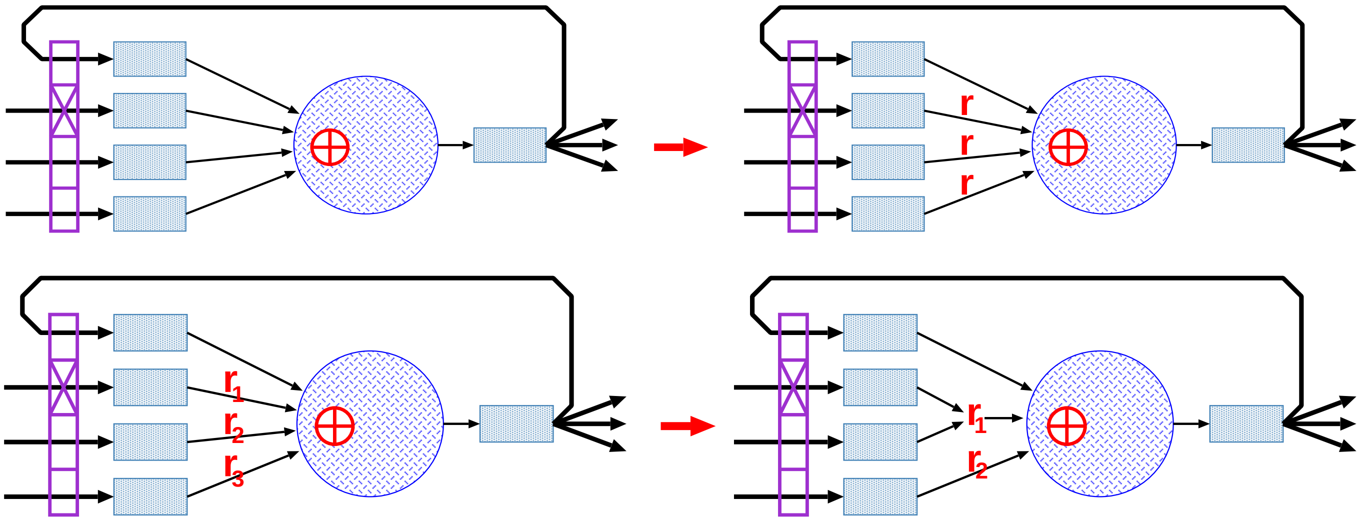
  $\triangleleft$ est un r-opérateur sur $\mathbb{S}$ s'il existe une bijection $\mathsf{r} : \mathbb{S} \rightarrow \mathbb{S}$ telle que $\triangleleft$ est : [SIROCCO'98]

r-associatif	$(x \triangleleft y) \triangleleft \mathsf{r}(z) = x \triangleleft (y \triangleleft z)$
r-commutatif	$\mathsf{r}(x) \triangleleft y = \mathsf{r}(y) \triangleleft x$
r-idempotent	$\mathsf{r}(x) \triangleleft x = \mathsf{r}(x)$
neutre à droite	$x \triangleleft e_{\triangleleft} = x$

- $x \triangleleft y = x \oplus \mathsf{r}(y)$

r-opérateurs (2)

- Interprétation dans le réseau



- $q \leftarrow \diamond$-`direct-association`(net_G, p) :
  - s-opérateur $\oplus$

    $$q[v] \leftarrow \oplus \left\{p[u], u \in \Gamma^-_G(v)\right\}$$

  - r-opérateur idempotent $\triangleleft$

    $$x \triangleleft y = x \oplus \mathsf{r}(y)$$

    $$q[v] = \oplus \left\{\mathsf{r}^{d_G(u,v)}\left(p[u]\right), u \in \Gamma^-_G(v)\right\}$$

  - r-opérateur n-aire $\triangleleft$

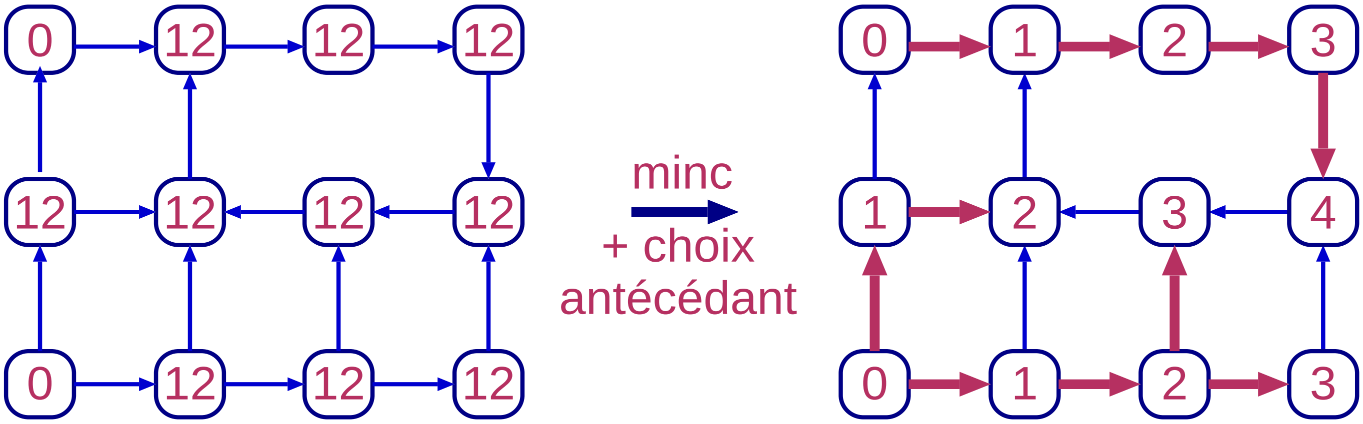
    $$\triangleleft\left(x_0, \ldots, x_n\right) = x_0 \oplus \mathsf{r}_1(x_1) \oplus \cdots \oplus \mathsf{r}_n(x_n)$$

    $$q[v] = \oplus \left\{\mathsf{r}_P\left(p[u]\right), u \in \Gamma^-_G(v), P \text{ chemin de } u \text{ to } v \right\}$$

Applications aux parcours de graphes (1)

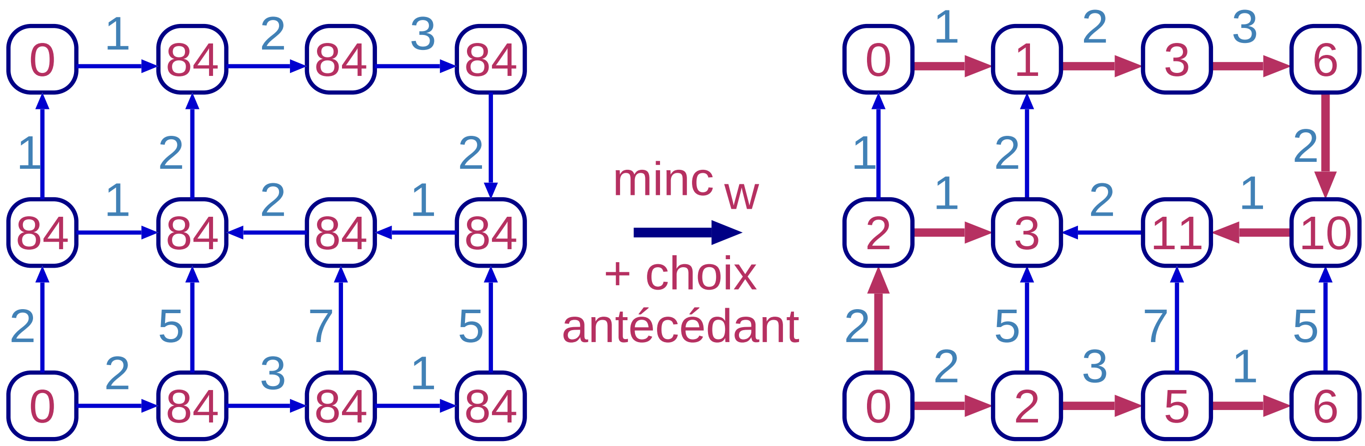
- Distance
  minc$(x, y) = $ min$(x, y + 1)$ dans $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{+\infty\}$

- Forêt des plus courts chemins



Applications aux parcours de graphes (2)

- *Single / Multiple source shortest path*
  - graphe avec pondération des arcs $w\left((u,v)\right)$
  - $\mathsf{minc}_w[x_0,\ldots,x_n] = \min[x_0, r_1(x_1),\ldots,r_n(x_n)]$
    $r_i(x) = x + w\left((u_i,v)\right)$



- Forêt des chemins les plus fiables
  $\mathsf{maxmul}_\pi[x_0,\ldots,x_n] = \max[x_0, r_1(x_1),\ldots,r_n(x_n)]$
  $r_i(x) = x \times \pi\left((u_i,v)\right)$

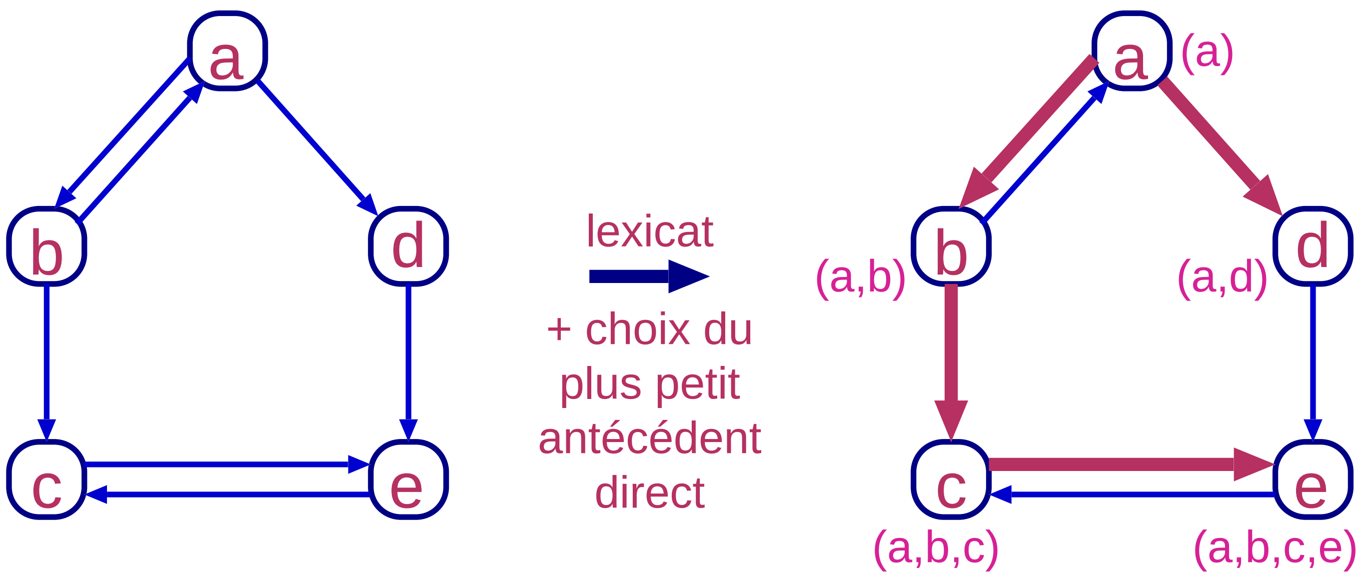
- Forêt des chemins de capacité maximale
  $\mathsf{maxmul}_\kappa[x_0,\ldots,x_n] = \max[x_0, r_1(x_1),\ldots,r_n(x_n)]$
  $r_i(x) = \min\left(x, \kappa\left((u_i,v)\right)\right)$

Applications aux parcours de graphes (3)

Opérateur lexicat $\triangleleft$

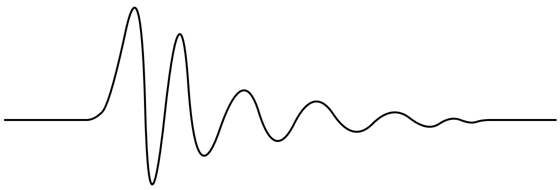
- $\mathbb{S}$ est l'ensemble des listes de sommets
- $\oplus$: ordre lexicographique sur les listes
  exemples: $(a, b, d) \oplus (a, b, c, d) = (a, b, c, d)$
  $(a, a) \oplus (a) = (a)$
- $\mathsf{r} : \mathbb{S} \to \mathbb{S}$, $\mathsf{r}(l) = l \cup (v)$ sur le sommet v
- $\triangleleft$ défini par $l \triangleleft l' = l \oplus \mathsf{r}(l')$ est idempotent

Application : arbre en profondeur



Contrôle du réseau

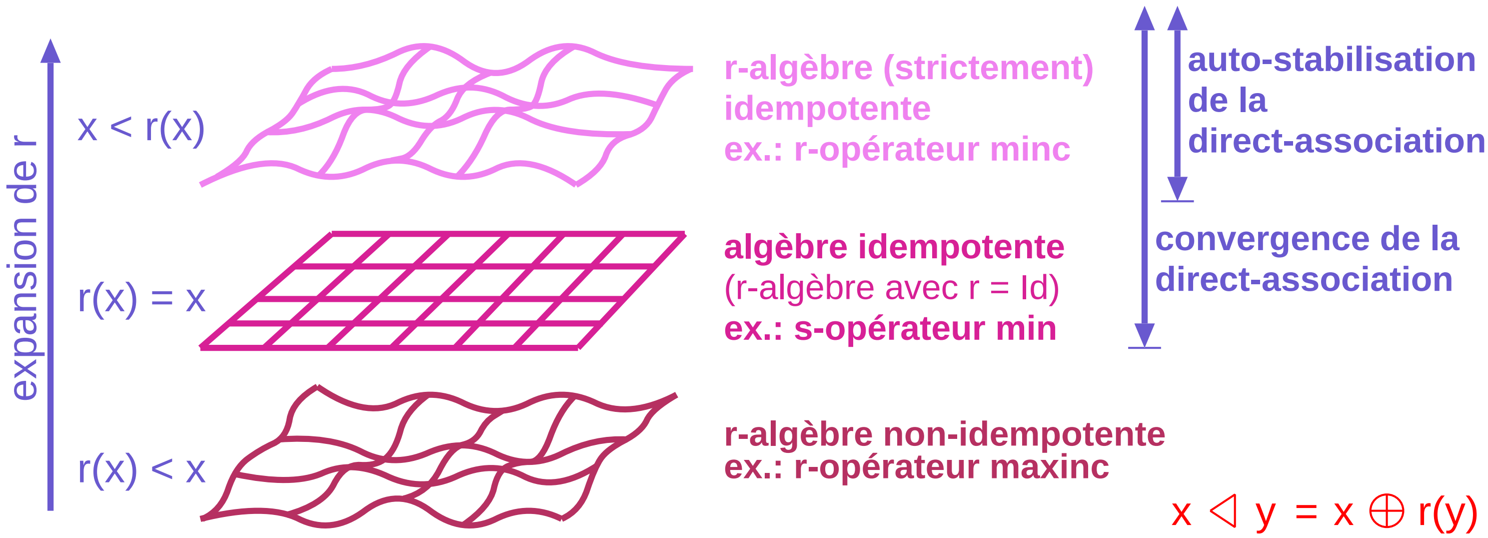
- Passage de messages
  - s- ou r-opérateur $\rightsquigarrow$ $x_0 \oplus \mathsf{r}_1(x_1) \oplus \cdots \oplus \mathsf{r}_l(x_l)$
  - $\oplus$ idempotent $\rightsquigarrow$ duplication
  - $\oplus$ commutatif $\rightsquigarrow$ désordonnancement
  - $\oplus$ $\rightsquigarrow$ ordre partiel $\preceq_\oplus$ $\rightsquigarrow$ si perte, alors remplacement par une "valeur plus petite"
- Détection de terminaison
  - bibliothèque de calcul ou Maille $\rightsquigarrow$ centralisée
  - protocole distribué PIF $\rightsquigarrow$ intégrée [RENPAR 10]
- Résistance aux pannes
  - données erronées dans les registres (ROM et code sûrs)
  - auto-stabilisation : en l'absence de nouvelles fautes, le système retrouve un état correct en temps fini



  - algorithme écrit avec deux règles, de copie et de calcul ; communications par registres ; lecture / ecriture atom.
  - résultat : [OPODIS'98]
    r-opérateur *strictement* idempotent ($x \prec_\oplus \mathsf{r}(x)$) $\rightsquigarrow$ algorithme silencieux auto-stabilisant sur tout réseau

De l'étude à l'algorithmique

- Réseau
  - sous-réseaux dynamiques d'un réseau de base
  - maille : plongements
  - abstraction avec les graphes virtuels
- $\diamond$-`direct-association`(net_G, p)
  - spécification des calculs locaux, et non du calcul global
  - conditions sur le réseau G et/ou l'opérateur $\diamond$



- Méthodologie de programmation
  - circuit $\rightsquigarrow$ opérateur idempotent
  - opérateurs non-idempotent $\rightsquigarrow$ sous-réseau préfixe
  - manipulations du réseau
  - utilisation de l'asynchronisme dans l'algorithmique : relaxation locales, jusqu'à la stabilisation

Calculs sur un DAG

Exemple : programmation dynamique

- $V_{n+1}(l) = \max\left(V_n(l), V_n(l - w_{n+1}) + v_{n+1}\right)$
- r-opérateur binaire $\text{maxinc}_v(x, y) \mapsto \max(x, y + v_i)$
- non idempotent mais absence de circuits

bagages

n°	poids	volume
0	0	0
1	1	2
2	1	1
3	2	3
4	3	2
5	4	3

volume maximal pour un poids inférieur ou égal à 6 : 7
sous-ensemble des bagages solution : 1, 3, 4

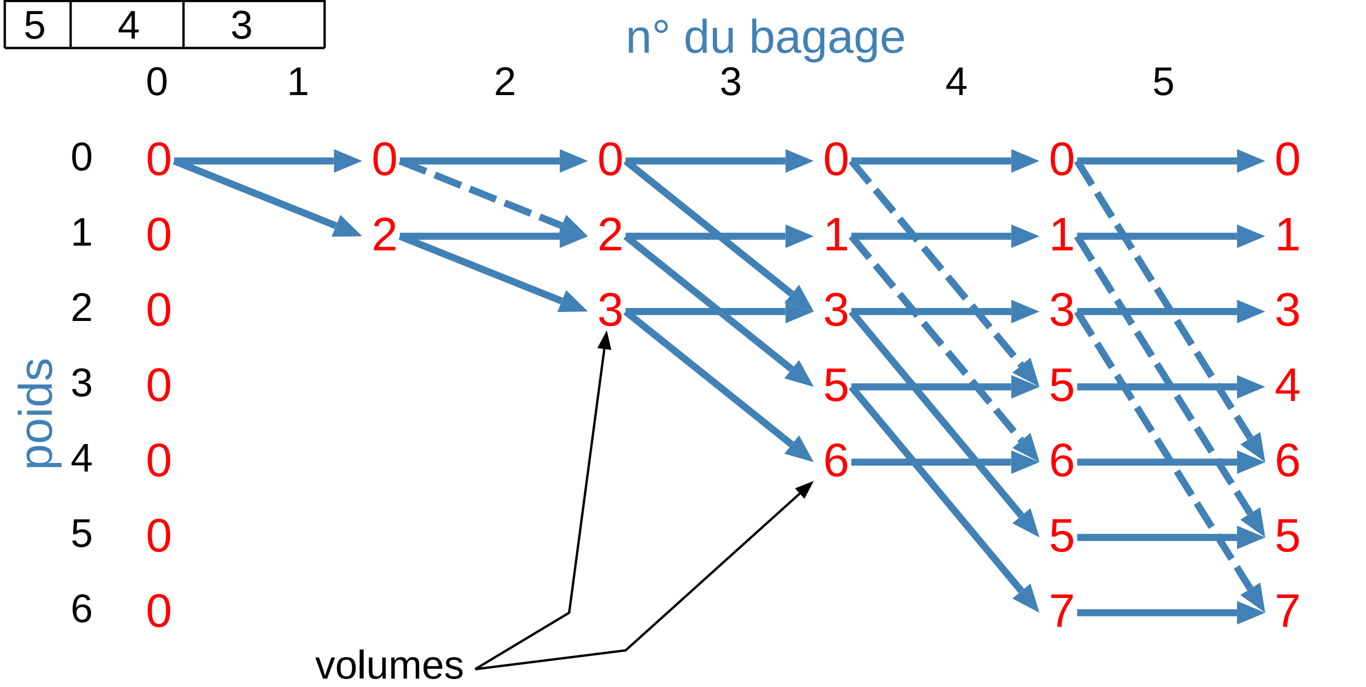
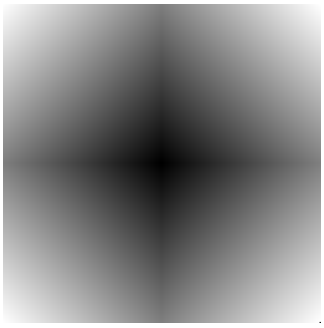


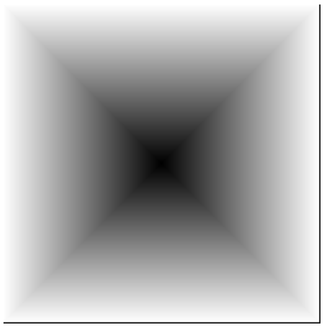
Image : distance

- Transformée en distance [soumis Computer Journal]
  - $d_4(p,q) = |x_p - x_q| + |y_p - y_q|$
  - $d_8(p,q) = \max[|x_p - x_q|, |y_p - y_q|]$
  - $d_v(p,q) = \quad a \times \min[|x_p - x_q|, |y_p - y_q|] + b \times (\max[|x_p - x_q|, |y_p - y_q|] - \min[|x_p - x_q|, |y_p - y_q|])$

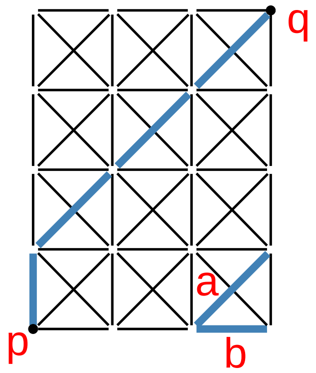
  $(a,b) = (3,2)$, $(a,b) = (7,5)$ [Borgefors 1986]



d_4



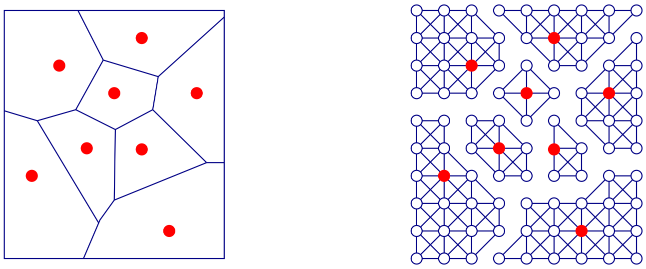
d_8



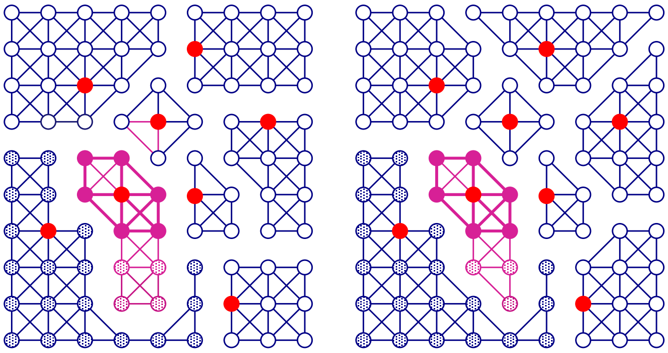
d_v

- $\rightsquigarrow$ $\mathrm{minc}(x,y) = \min(x, y+1)$ ou minc_w
  - un calcul asynchrone plus gros au lieu de plusieurs petits calculs synchrones
  - $\rightsquigarrow$ expressivité
    grossissement du grain du calcul

Géométrie : domaines de Voronoï

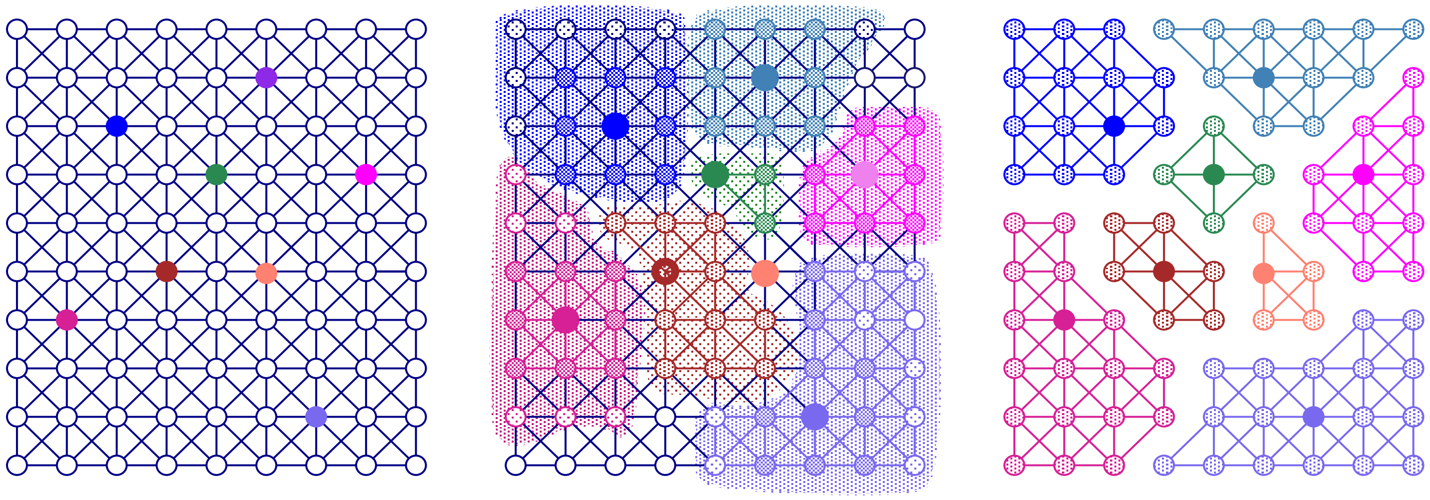


Calcul data-parallèle synchrone [Computing Supp.]



régions initiales quelconques
tant qu'une région évolue, faire
  tant qu'une région voisine est à visiter, faire
    calcul de la distance à la graine voisine
    rendre les sommets de la graine voisine
  fin tant que
fin tant que

Calcul data-parallèle asynchrone

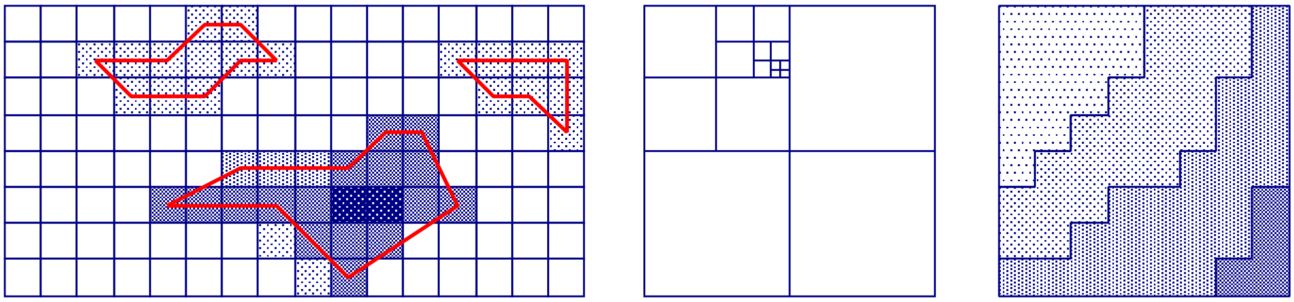


- une seule `minc-direct-association`(net_{maille}, p)
- $\leadsto$ suppression d'itérations

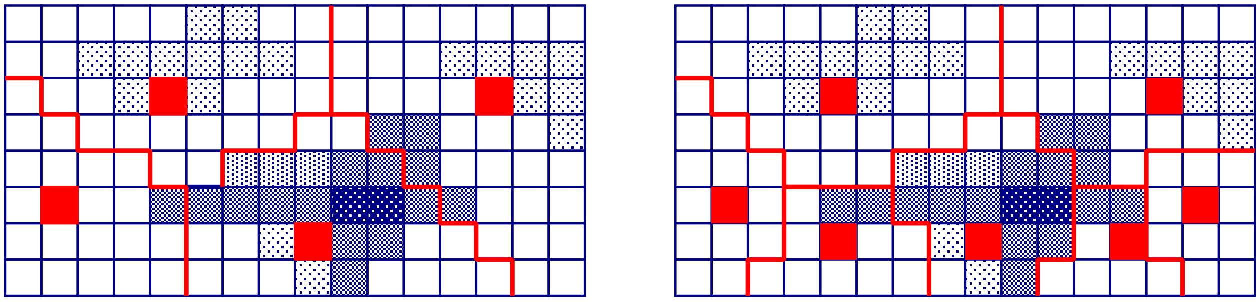
Segmentation d'images

Segmentation d'images

- déterminer les objets
- critères d'homogénéité, de texture, de forme...
- découpe et fusion [Horowitz et Pavlidis 1976]



Segmentation de Voronoï [Ahuja et Shachte, 1985]



- critères d'homogénéité et de forme $\rightsquigarrow$ complexe
- parallélisme de données difficile à exploiter
- codage à base de graphe adapté
- $\rightsquigarrow$ découpe avec des minc-`direct-associations`, fusion à l'aide des graphes virtuels

ANET (1). Objectifs

Réseau associatif

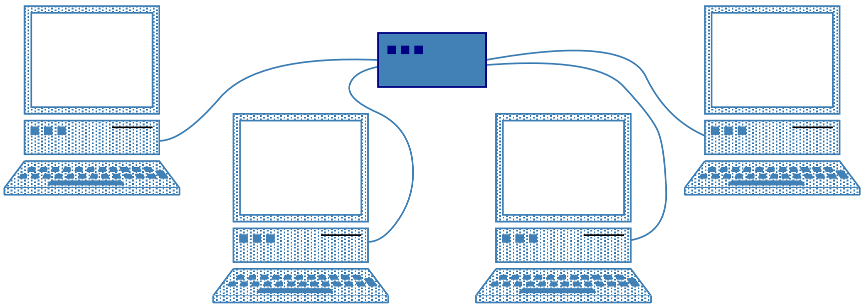
- réseau de base + sous-réseaux $\leadsto$ objets irréguliers
  exemple : régions de Voronoï
- primitives `associations`, calculs asynchrones
- irrégularité des traitements
  exemple : segmentation d'images

Paralléliser les primitives de calcul en respectant :

- leurs spécificités (ex : asynchronisme)
- l'irrégularité de la forme des données
- l'évolution des formes très dynamique
- l'irrégularité des traitements

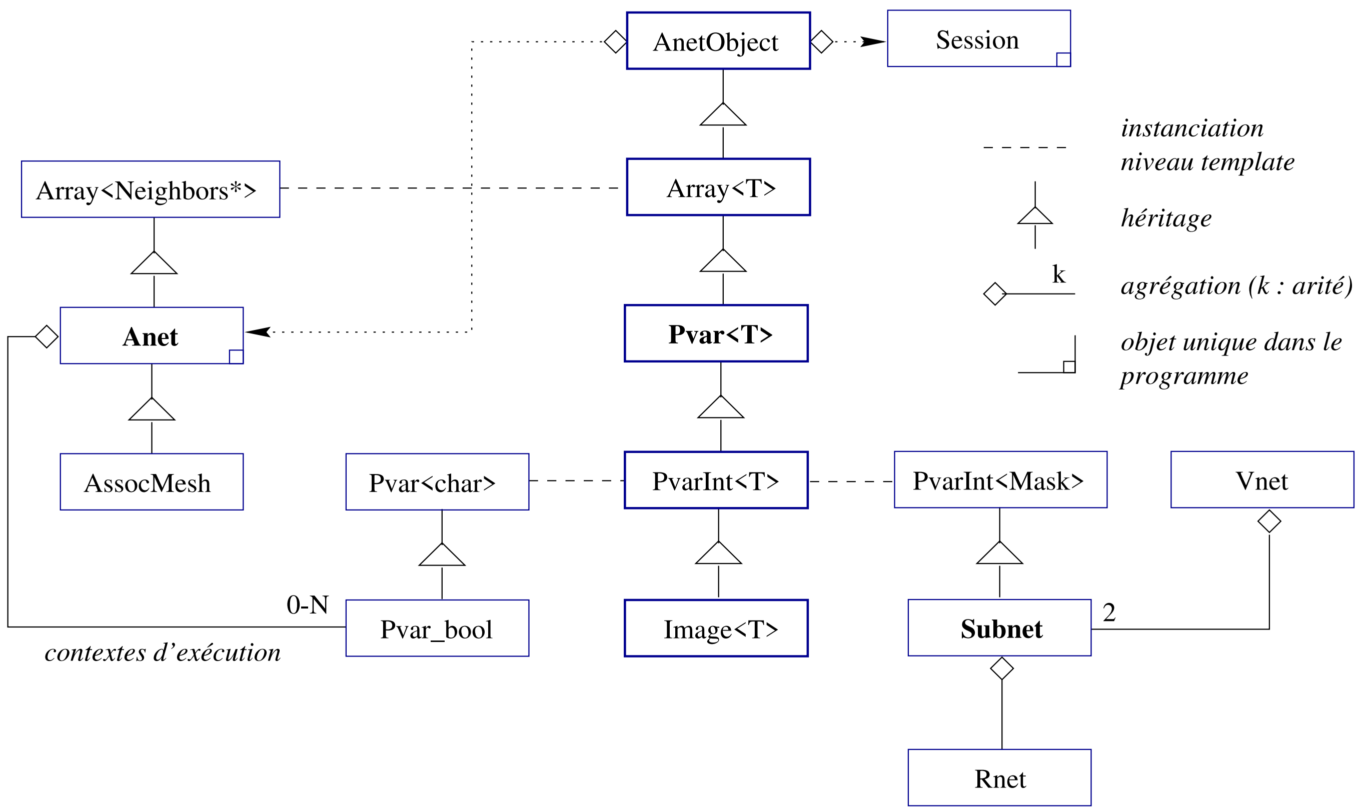
Plateformes visées

- PCs multi-processeurs
- grappes de PCs



ANET (2). Structure

Bibliothèque C++

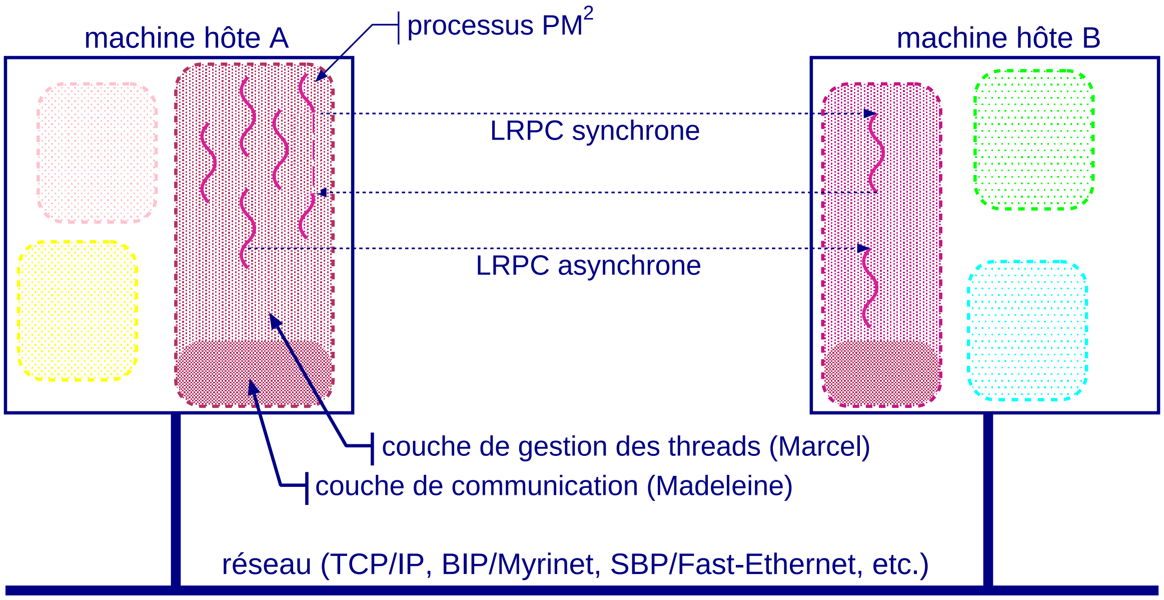


Intérêt

définition d'objets par l'utilisateur,
et d'opérateurs pour les manipuler

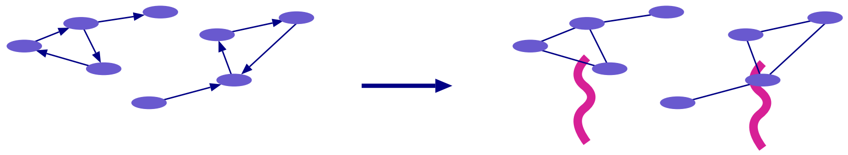
ANET (3). PM^2

- Besoins
  - placement souple sur les ressources de calcul
  - gestion de l'irrégularité préventive
  - plusieurs calculs, calculs et communications
  - irrégularité $\rightsquigarrow$ équilibrage de charge (à terme)
  - grain fin $\rightsquigarrow$ mécanismes peu chers
- PM^2 : *Parallel Multithreaded Machine*
  - processus “légers” $\rightsquigarrow$ dynamisme
  - exécutions multi-fils
  - recouvrement calcul / communication
  - exécution à distance / migration
    $\rightsquigarrow$ équilibrage de charge

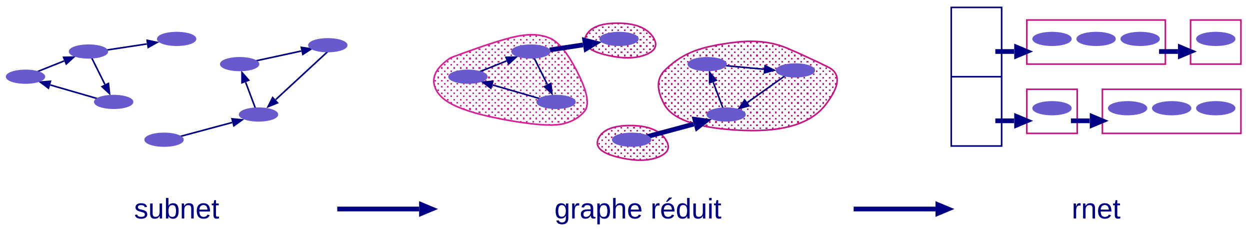


ANET (4). Parallélisation

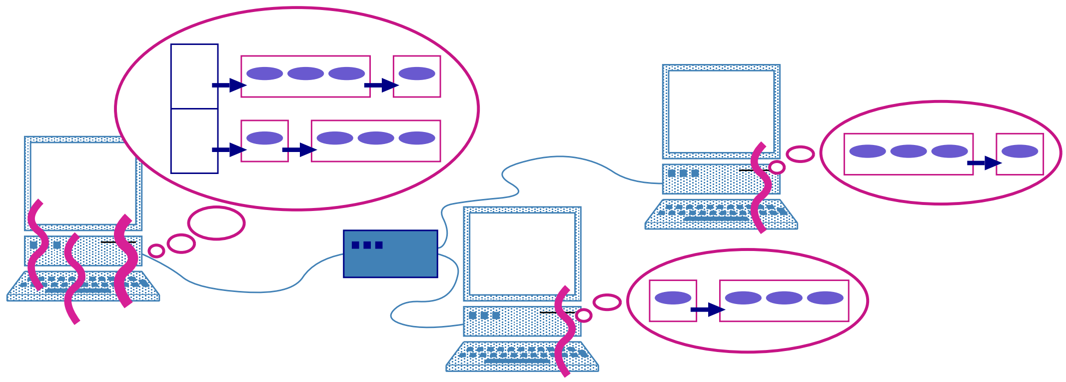
- $q \leftarrow \diamond$-`direct-association`(net_G, p)



- Gestion de l'irrégularité des données



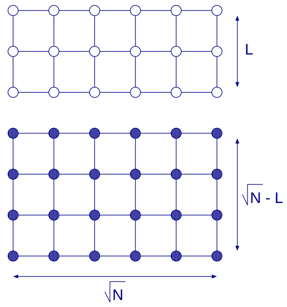
- Lancement des *threads*



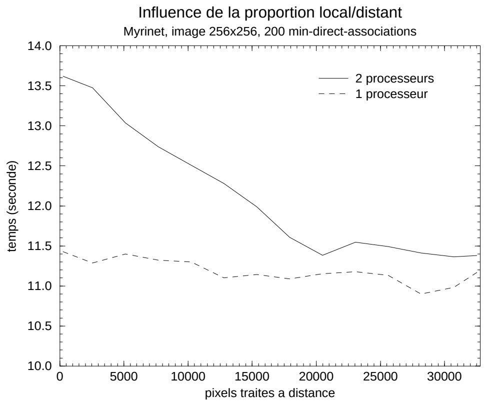
- Bilan
  - primitives des réseaux associatifs
  - gestion de l'irrégularité, structure objet
  - utilisable UNIX, Myrinet + BIP, Ethernet, SMP
  - réseaux associatifs $\rightsquigarrow$ placement de *threads*
    $\rightsquigarrow$ équilibrage de charge (à terme)
  - exécutions à distance $\rightsquigarrow$ ok si grain pas trop fin

Intérêt des r-opérateurs : grain du calcul

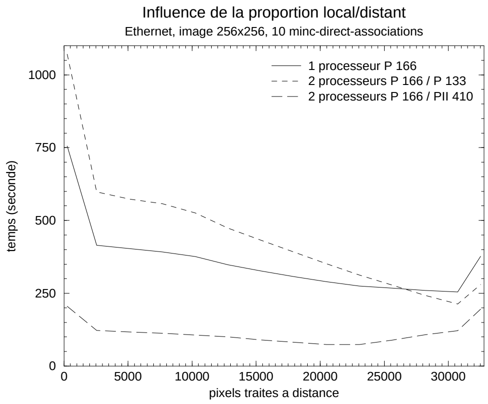
Comparaisons d'opérateurs



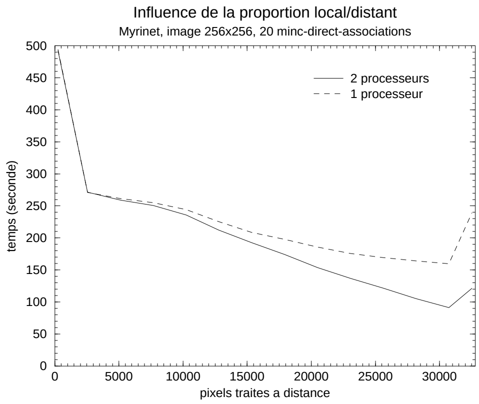
- deux processeurs
- un seul calcul global
- comparaison de min, or et minc
- influence réseau et machines



Myrinet min

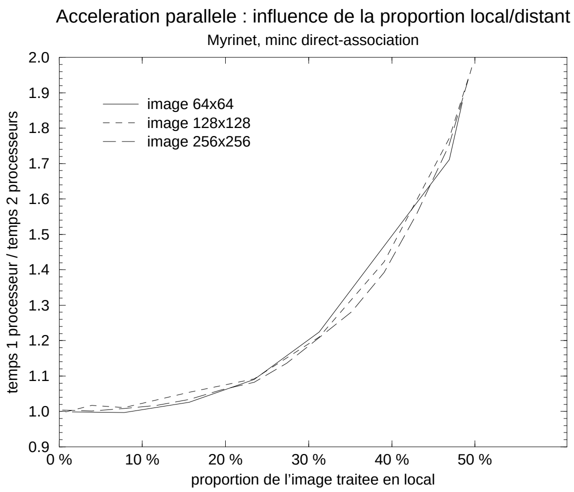


Ethernet minc



Myrinet minc

Conclusion



- r-opérateurs ⇝ grain plus gros ⇝ facilite la parallélisation
- intérêt des réseaux rapides
- influence de l'équilibrage de charge

Intérêt des r-opérateurs : asynchronisme

Découpe de Voronoï sur un seul processeur

Taille image	Type image	ANET s-opérateurs	ANET r-opérateurs	rapport s- / r-
Images 64×64 10 racines	forme	10,48	1,55	6,76
	icoron	6,98	1,44	4,84
	imuscle	8,77	1,16	7,56
	roks	9,78	1,23	7,95
	moyenne	*9,04*	*1,29*	*6,77*
Images 128×128 40 racines	forme	88,14	7,20	12,24
	icoron	41,05	5,53	7,42
	imuscle	52,28	5,47	9,55
	roks	57,64	5,92	9,73
	moyenne	*57,75*	*5,9*	*9,73*
Images 256×256 160 racines	forme	454,82	42,65	10,66
	icoron	230,40	28,96	7,95
	imuscle	340,77	31,74	10,73
	roks	301,50	30,25	9,96
	moyenne	*328,89*	*32,14*	*9,82*

Remplacement d'itérations

- s-opérateurs : $p \times q$ itérations $O(N)$
- r-opérateurs : p itérations en $O(N^2)$
- expériences : q, N, et α, $1 < \alpha \leq 2$, tels que
$$N^\alpha < q \times N$$
- N borné (images)

Conclusion (1). Résumé

- Origine : Maille $\leftrightarrow$ modèle abstrait
- Étude de la topologie
  - graphes virtuels
  - plongements
- Étude de la `direct-association`
  - conditions sur le réseau : graphes préfixes
  - conditions sur l'opérateur : r-algèbre
  - contrôle et auto-stabilisation
- Algorithmique
  - graphe
  - géométrie
  - image
- Mise en œuvre : ANET
  - objets, irrégularité
  - parallélisation des primitives
  - tests probants

Conclusion (2). Perspectives

- Bilan
  - modèle de programmation data-parallèle original
  - masque : puissance du calcul sur réseau dynamique
  - asynchronisme : algorithmique $\rightsquigarrow$ expressivité
    parallélisation $\rightsquigarrow$ efficacité
- Étude du modèle : perspectives
  - extension pour opérateurs non-idempotents
  - hypothèses pour l'auto-stabilisation
    - $\rightsquigarrow$ *étude d'autres primitives asynchrones*
    - $\rightsquigarrow$ *optimisations des calculs sur les graphes*
- Étude de l'algorithmique : perspectives
  - $\rightsquigarrow$ *étude d'une toute nouvelle algorithmique*
- Étude de la mise en œuvre : perspectives
  - équilibrage de charge, optimisations diverses
  - nouveaux tests
    - $\rightsquigarrow$ *d'un démonstrateur à un "produit fini"*
    - $\rightsquigarrow$ *étude d'un coprocesseur image*
- Applications concrètes de ce travail
  - $\rightsquigarrow$ programmation parallèle
  - $\rightsquigarrow$ serveur de traitements multimédia MPEG 4 et 7

Mémo

- réseau de base :
  réseau contenant tous les sous-réseaux utilisés
- pvar :
  variable parallèle, $i.e.$, une variable sur chaque nœud
- pvar de masques :
  code le sous-réseau, en masquant des arcs
- s-opérateur $\oplus$:
  associatif, commutatif, idempotent
  exemple : min, max, or, and, $\cup$, $\cap$, $\ldots$
- r-opérateur $\triangleleft$:
  r-associatif, r-commutatif, r-idempotent
  exemple : minc$(x, y) =$ min(x, y)
- `direct-association` :
  primitive de calcul data-parallèle ;
  calcul global sur un sous-réseau, constitué de calculs locaux répétés jusqu'à stabilisation du réseau

Applications aux parcours de graphes (4)

- Opérateur ant $\triangleleft$
  - $\mathbb{S}$ est l'ensemble des listes d'ensembles de sommets
  - $(S_1, \ldots, S_k) = (S_1, \ldots, S_k, \emptyset, \ldots, \emptyset)$.
  - $\oplus$: fusion terme à terme + suppression des doubles
    exemple : $(\{d\}, \{b\}, \{a, c\}) \oplus (\{c\}, \{a, e\}, \{b\}) =$
    $(\{d, c\}, \{b, a, e\}, \{\cancel{a}, \cancel{c}, \cancel{b}\}) =$
    $(\{d, c\}, \{b, a, e\})$
  - $\mathrm{r}(\{S_1, \ldots, S_k\}) = \{\emptyset, S_1, \ldots, S_k\}$
    exemple : $\mathrm{r}\left((\{d\}, \{b\}, \{a, c\}) \right) = \left(\{\emptyset, \{d\}, \{b\}, \{a, c\}\} \right)$
  - $\triangleleft$ défini par $\quad l \triangleleft l' = l \oplus \mathrm{r}(l')$ est idempotent
- Application :

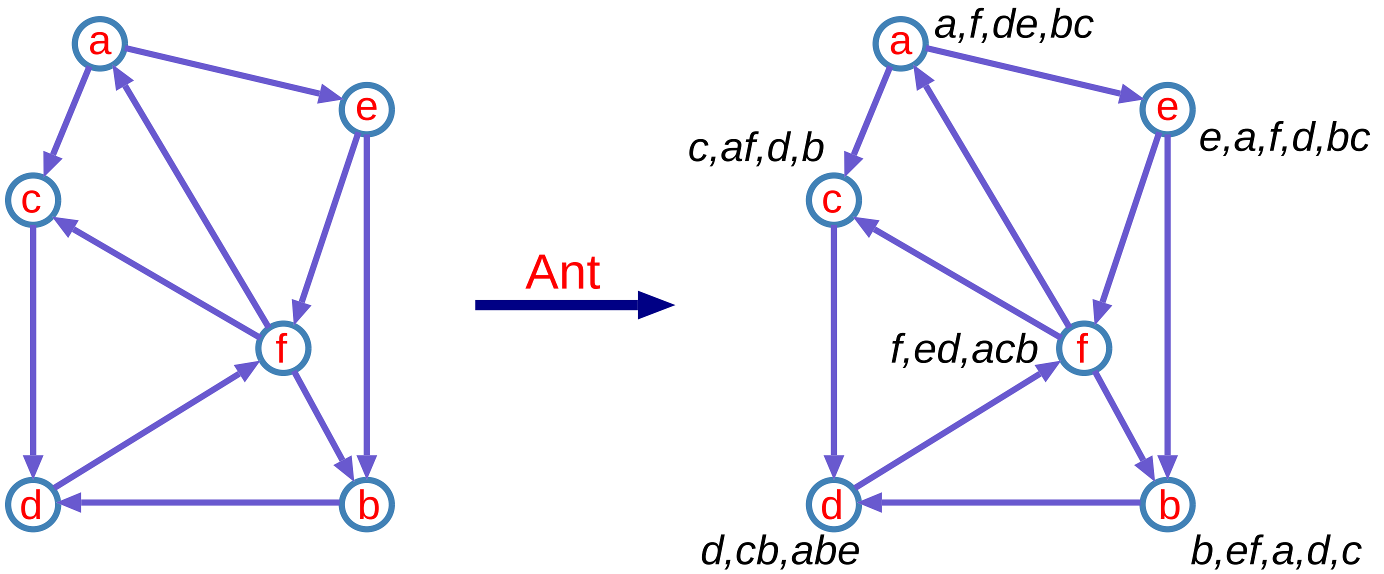
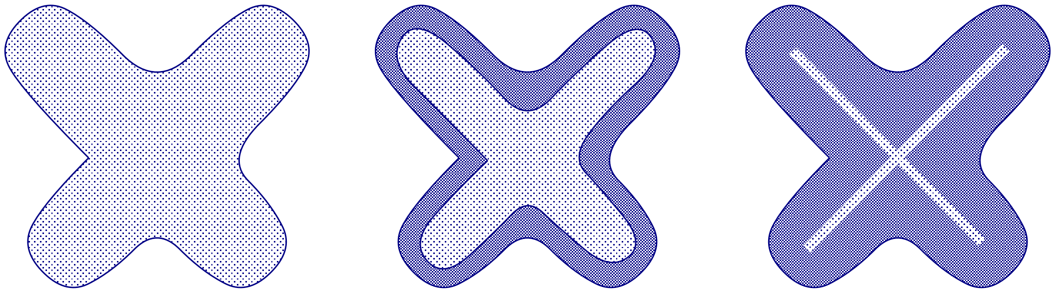


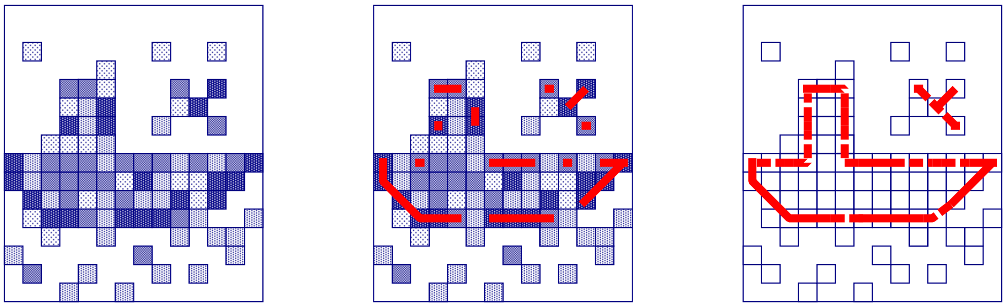
Image : squelette, fermeture de contours

Squelette



- feu de prairie
- $\rightsquigarrow$ r-opérateur minc$(x, y) =$ min$(x, y + 1)$

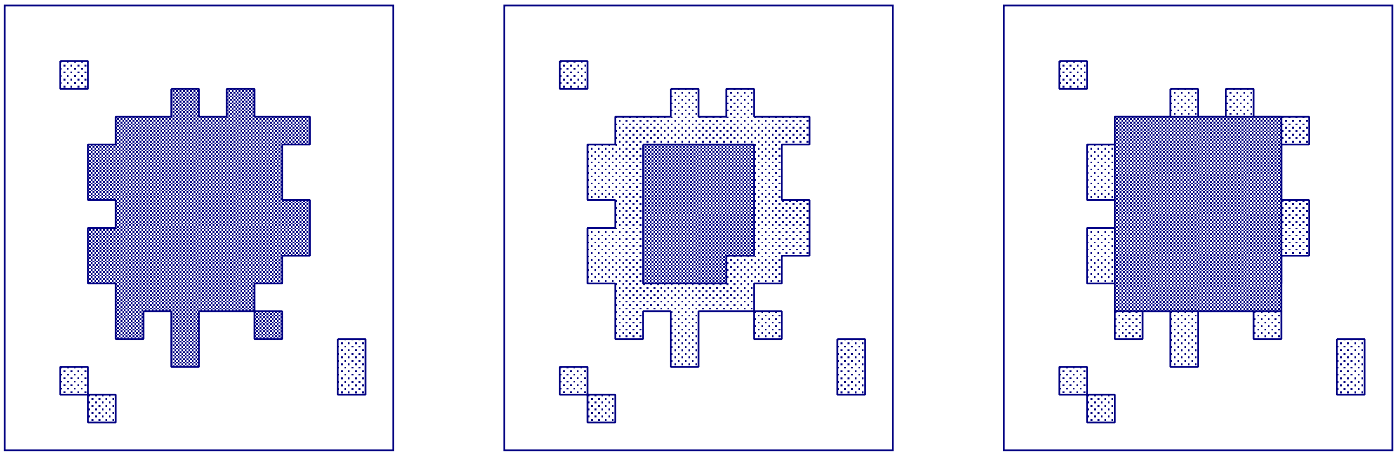
Fermeture de contours



- détermination de morceaux sûrs
- recherche d'un meilleur chemin pour fermer le contour
- $\rightsquigarrow$ r-opérateur minc$_w$
  minc$_w[x_0, \ldots, x_n] =$ min$[x_0, x_1 + w_1, \ldots, x_n + w_n]$
- w : distance, ou orientation

Image : morphologie mathématique

- Simplification de l'image

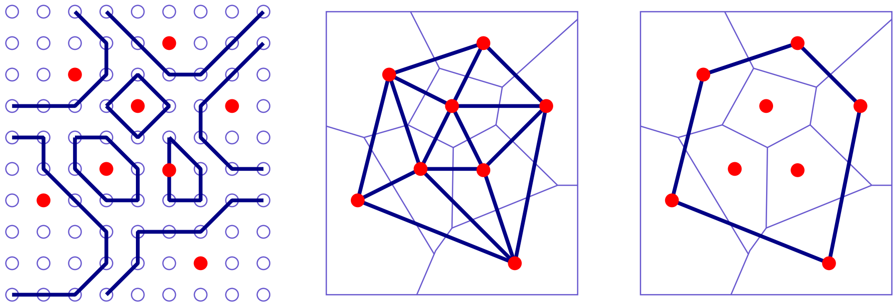


$$\begin{aligned} \textit{FerRec}\,(x_0, \ldots, x_8) &= \mathsf{min}\,(x_0, \mathsf{max}\,(x_0, \ldots, x_8)) \\ \textit{OuvRec}\,(x_0, \ldots, x_8) &= \mathsf{max}\,(x_0, \mathsf{min}\,(x_0, \ldots, x_8)) \end{aligned}$$

- r-opérateurs généralisés
  - r-fonction expansive $\rightsquigarrow$ idempotent $\rightsquigarrow$ convergence
  - reconnaissance du résultat [Robin 1997]
  - $\rightsquigarrow$ une seule `direct-association`

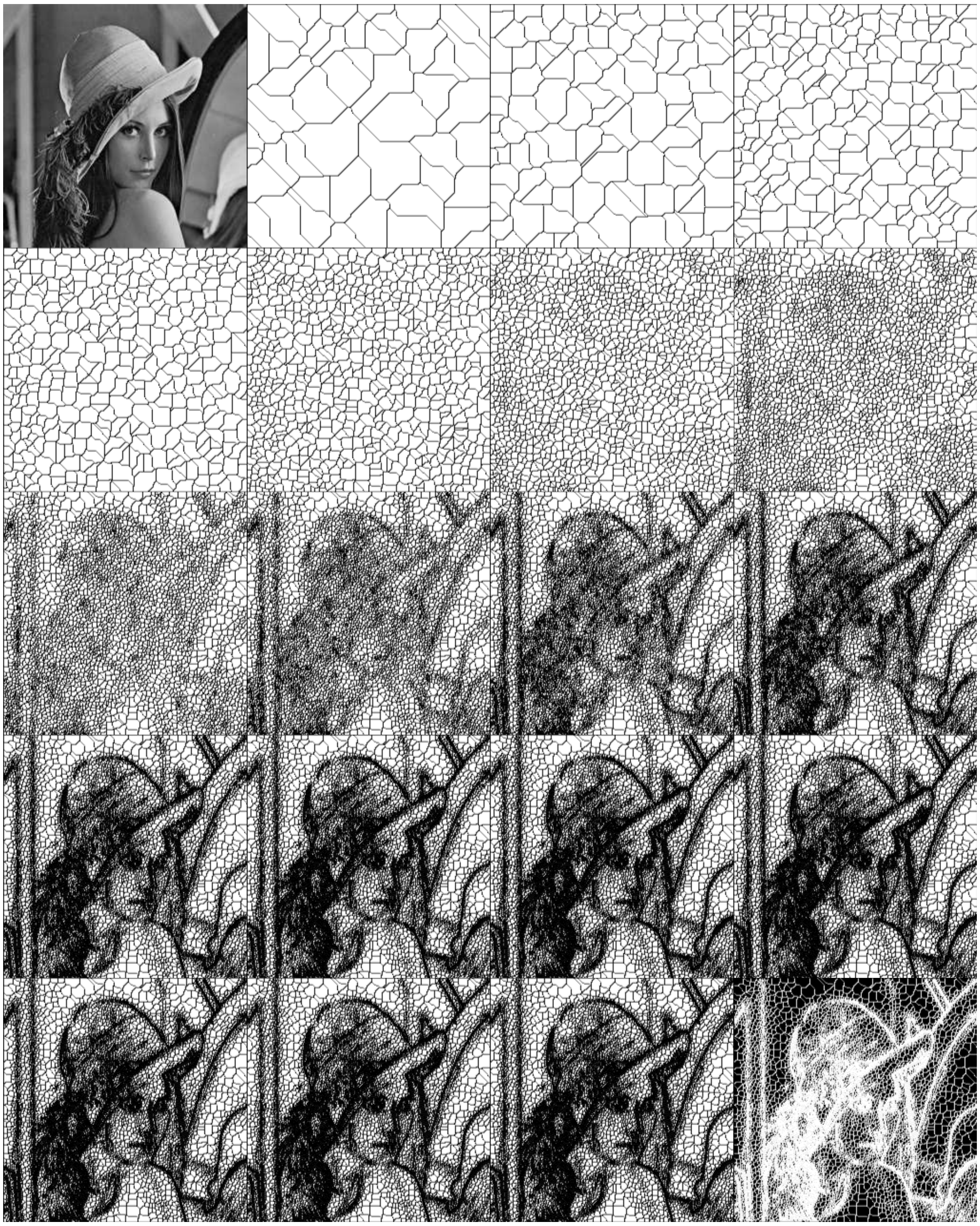
Géométrie (2)

- Problèmes connexes



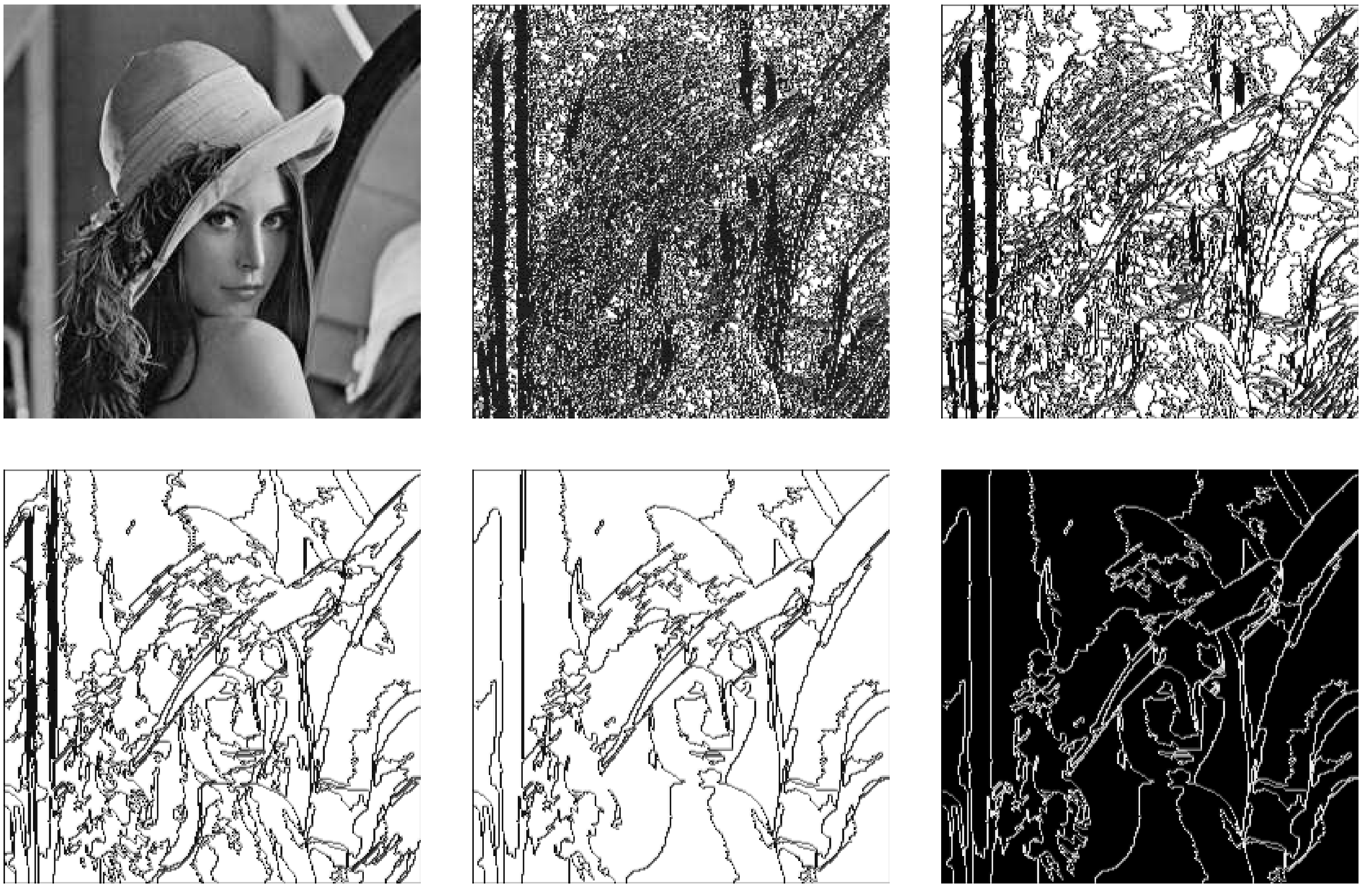
Segmentation d'images

- Découpe de Voronoï



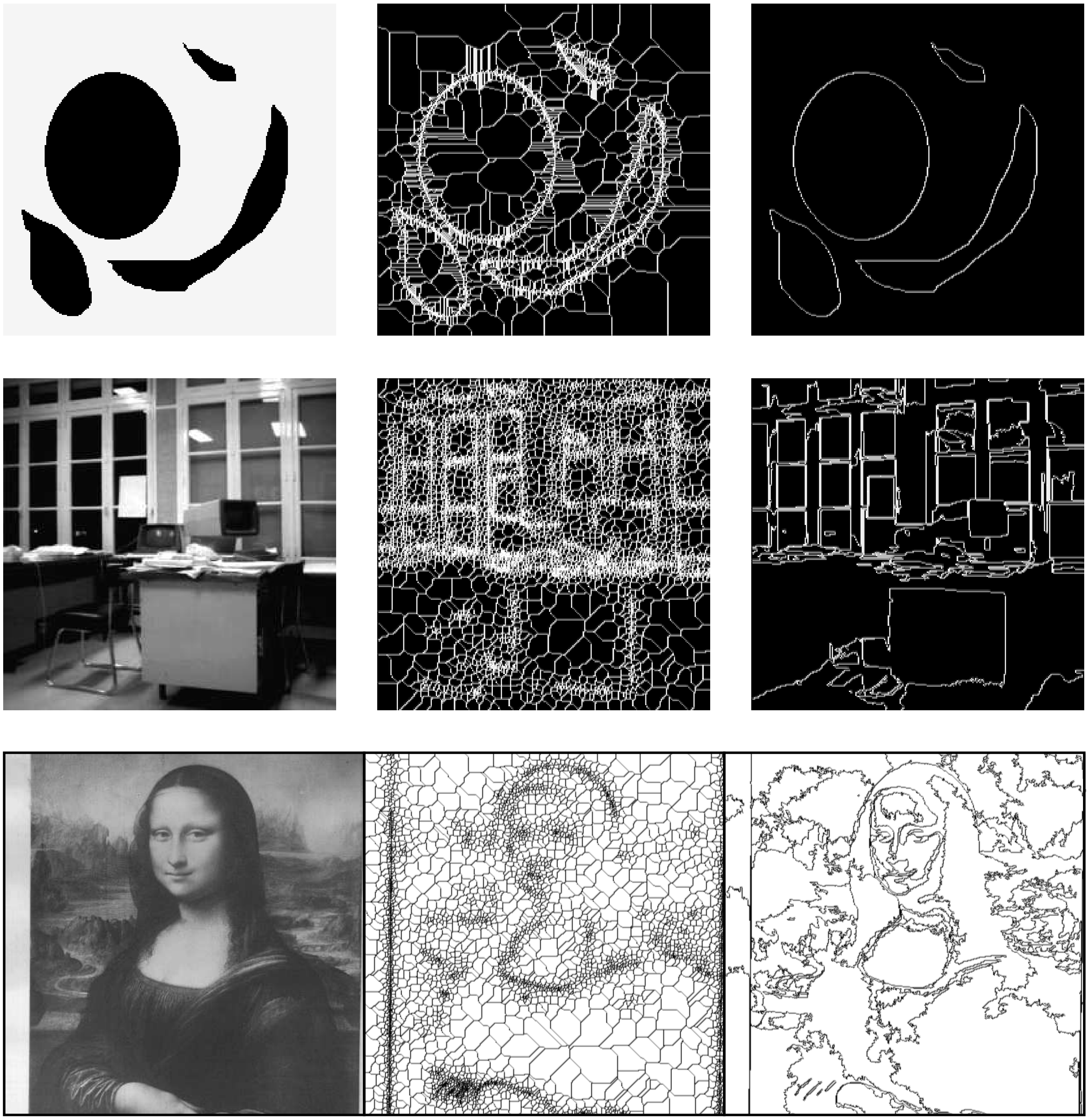
Segmentation d'images

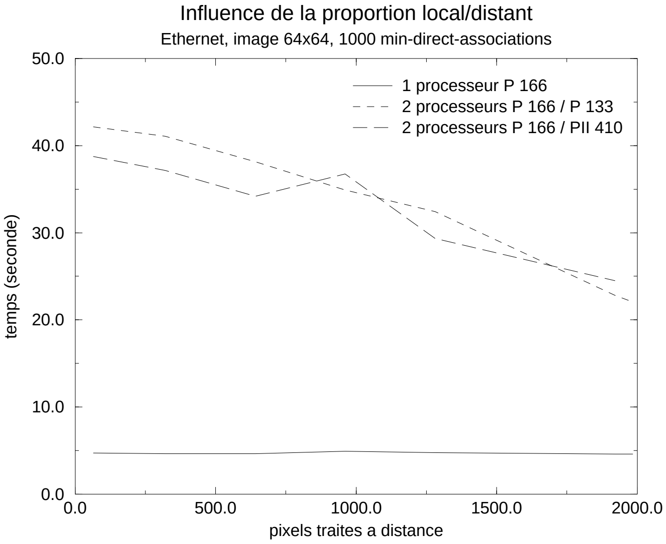
- Fusion de Voronoï

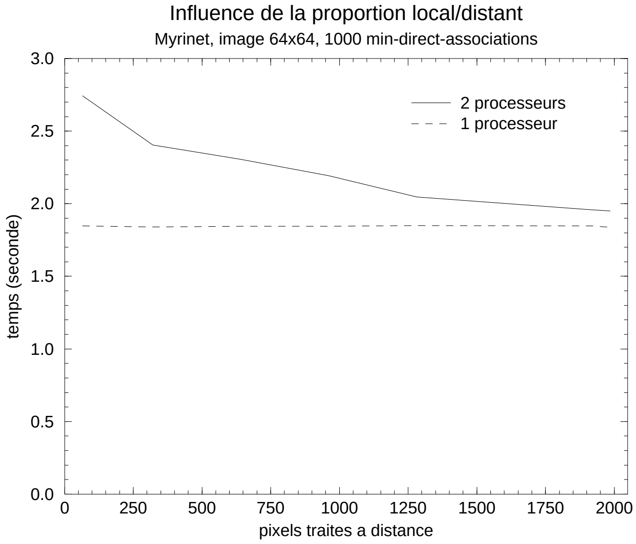


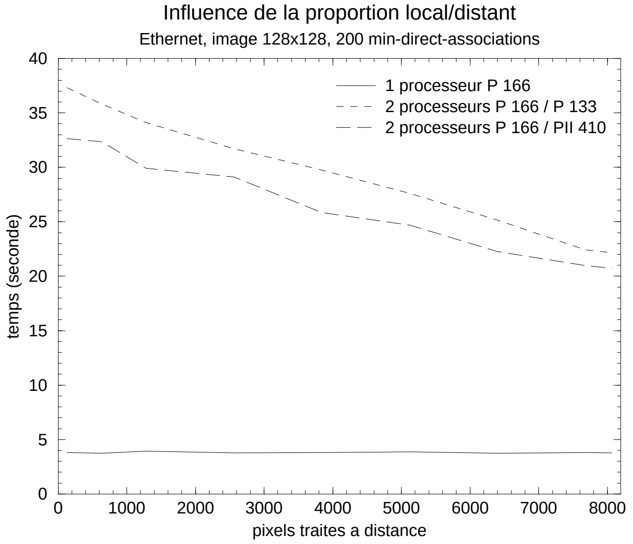
Segmentation d'images

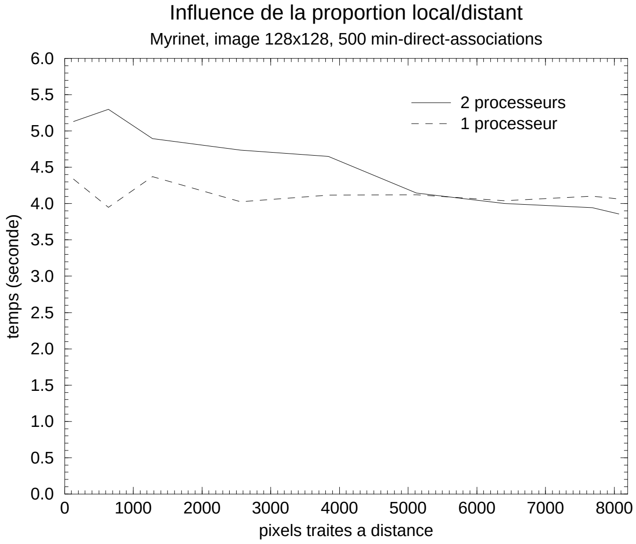
- Découpe et fusion de Voronoï

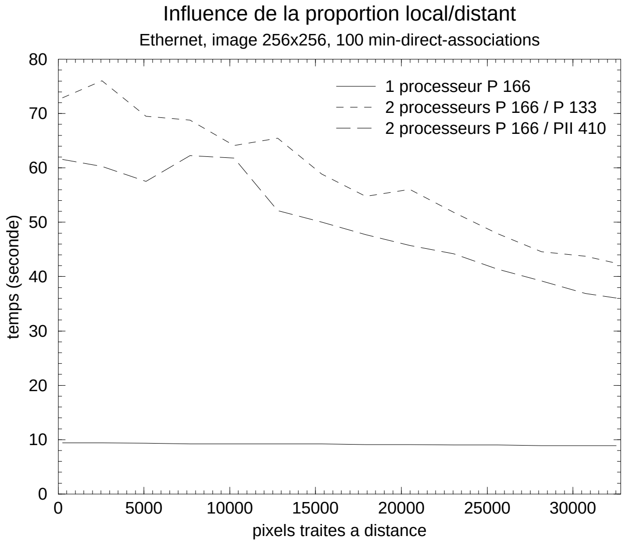


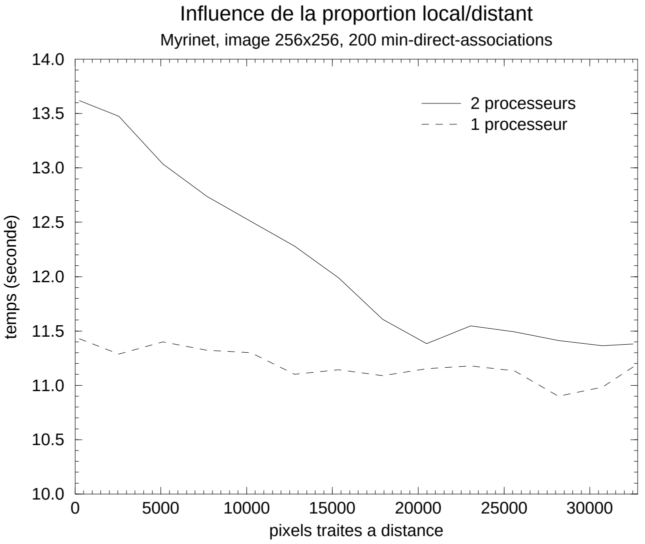

Influence de la proportion local/distant
Ethernet, image 64x64, 1000 min-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance

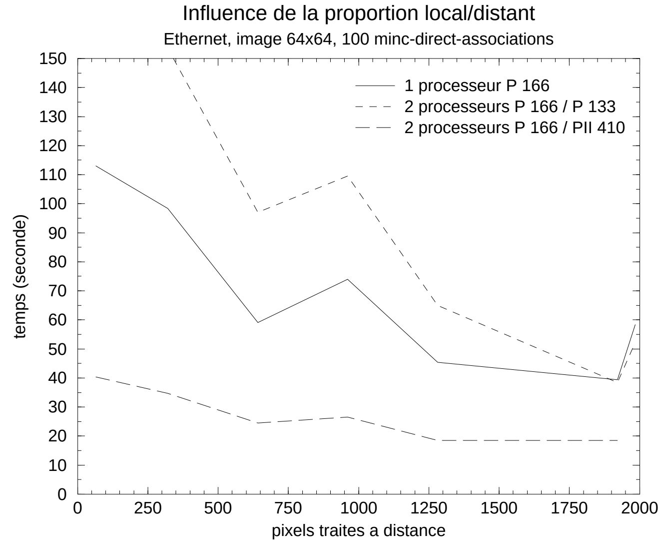

Influence de la proportion local/distant
Myrinet, image 64x64, 1000 min-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance

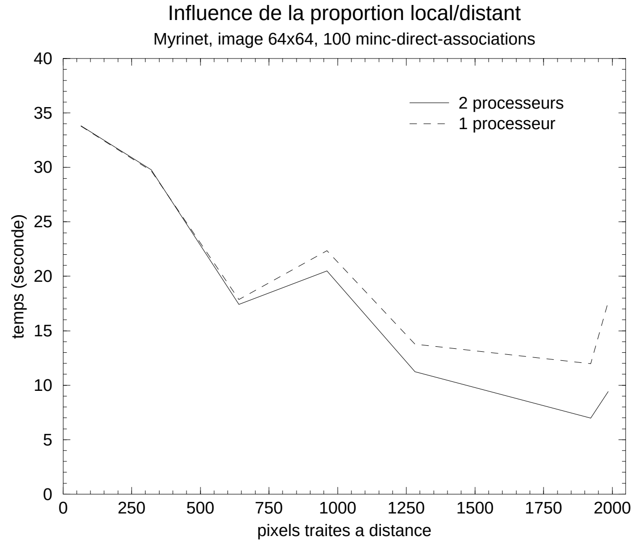

Influence de la proportion local/distant
Ethernet, image 128x128, 200 min-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance

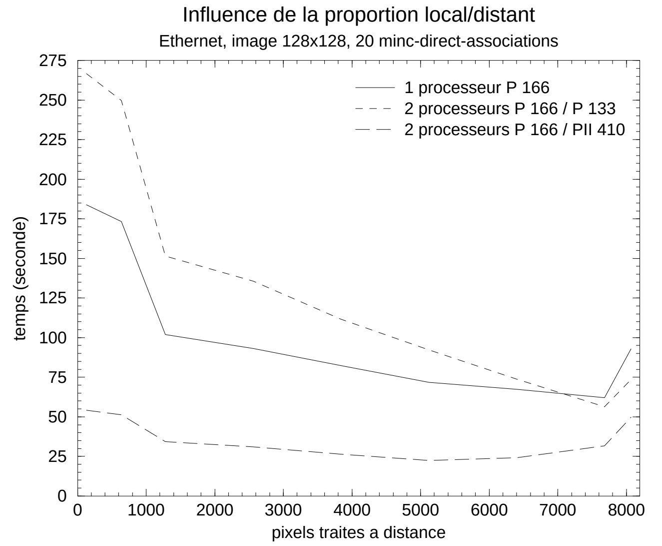

Influence de la proportion local/distant
Myrinet, image 128x128, 500 min-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance

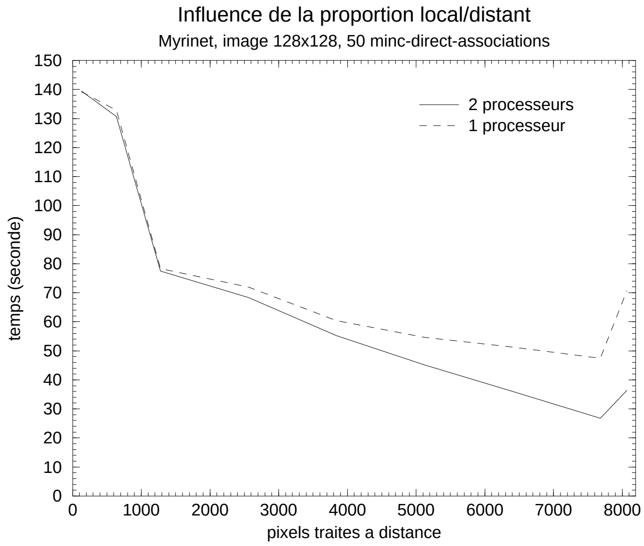

Influence de la proportion local/distant
Ethernet, image 256x256, 100 min-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance

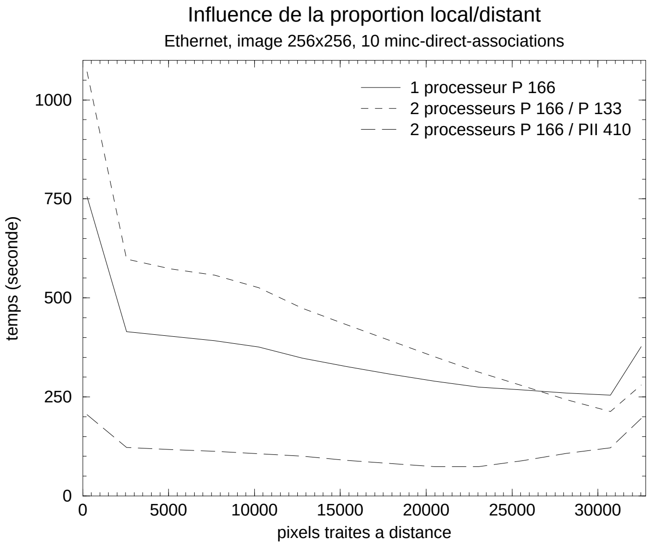

Influence de la proportion local/distant
Myrinet, image 256x256, 200 min-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance

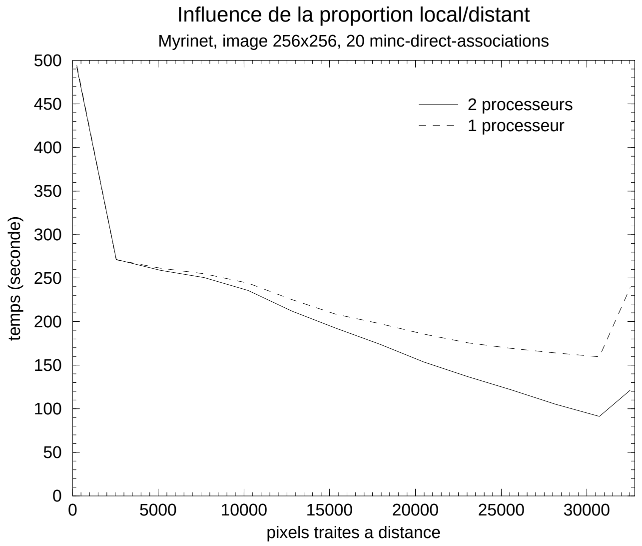

Influence de la proportion local/distant
Ethernet, image 64x64, 100 minc-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance


Influence de la proportion local/distant
Myrinet, image 64x64, 100 minc-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance


Influence de la proportion local/distant
Ethernet, image 128x128, 20 minc-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance


Influence de la proportion local/distant
Myrinet, image 128x128, 50 minc-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance


Influence de la proportion local/distant
Ethernet, image 256x256, 10 minc-direct-associations
1 processeur P 166
2 processeurs P 166 / P 133
2 processeurs P 166 / PII 410
temps (seconde)
pixels traites a distance


Influence de la proportion local/distant
Myrinet, image 256x256, 20 minc-direct-associations
2 processeurs
1 processeur
temps (seconde)
pixels traites a distance