INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Data Allocation Strategies for Dense Linear Algebra on two-dimensional Grids with Heterogeneous Communication Links

Olivier Beaumont,
Arnaud Legrand and

Yves Robert

No 4165

April 2001

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4165--FR+ENG

Data Allocation Strategies for Dense Linear Algebra on two-dimensional Grids with Heterogeneous Communication Links

Olivier Beaumont,
Arnaud Legrand and
Yves Robert

Thème 1 — Réseaux et systèmes
Projet ReMaP

Rapport de recherche n ° 4165 — April 2001 — 22 pages

Abstract: In this paper, we study the implementation of dense linear algebra kernels, such as matrix multiplication on 2D grids with homogeneous processors when the communication links between the processors are heterogeneous (i.e. the time to transfer a block of the matrix between two processors depends on these processors). We prove that finding the best allocation of the processors into a grid, with respect to the minimization of the communication overhead, is a NP-complete problem.

Key-words: heterogeneous communications, 2D grids, data distribution, linear algebra kernel

(Résumé : tsvp)

This text is also available as a research report of the Laboratoire de l'Informatique du Parallélisme `http://www.ens-lyon.fr/LIP`.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International : +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International : +33 4 76 61 52 52

Allocation de données pour l'algèbre linéaire dense sur grilles bidimensionnelles avec communications hétérogènes

Résumé : Dans ce rapport, nous nous intéressons à la mise en œuvre de noyeaux d'algèbre linéaire dense, comme le produit de matrices, sur les grilles bidimensionnelles de processeurs identiques reliés par un réseau de communications non homogène (le temps nécessaire au transfert d'un bloc de matrices entre deux processeurs dépend de ces processeurs). Nous montrons que déterminer une organisation des processeurs en grille minimisant les communications est un problème NP-complet.

Mots-clé : Communications hétérogènes, grilles bidimensionnelles, distribution de données, noyeau d'algèbre linéaire.

1 Introduction

We study the implementation of dense linear algebra kernels, such as matrix multiplication on 2D grids with homogeneous processors when the communication links between the processors are heterogeneous (i.e. the time to transfer a block of the matrix between two processors depends on these processors). The case of processors running at different speeds and homogeneous communication links has been studied in [11, 7, 10, 3, 4, 2]. In [2], we have proved that finding the best allocation of heterogeneous processors into a grid so as to balance the load between the processors is NP-Complete. In this paper, we prove the corresponding theorem with homogeneous processors and heterogeneous communications links. Indeed, in this context, finding the best allocation of the processors into a grid so as to minimize the communication overhead turns out to be NP-Complete.
This paper is organized as follows. In Section 2 we briefly recall the algorithm implemented in the ScaLAPACK library [5] on 2D homogeneous grids and derive its theoretical computation time. We discuss in Section 3 the necessary modifications to two-dimensional block-cyclic distributions used in ScaLAPACK to cope with 2D heterogeneous grids. Depending on the data-layout, we obtain a new theoretical computation time and introduce a new optimization problem. We prove in Section 4 that this optimization problem is NP-complete. In Section 5, we give some remarks on other similar problems and finally, we conclude in Section 6.

2 Homogeneous Grids

2.1 Principles

For sake of simplicity we consider the multiplication $C = AB$ of two square $n \times n$ matrices A and B. In that case, ScaLAPACK uses the outer product algorithm described in [1, 9, 12]. Consider a 2D processor grid of size $p \times p$.

Assume first that $n = p$. In that case, the three matrices share the same layout over the 2D grid : processor $P_{i,j}$ stores $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$. Then at each step k,

– each processor $P_{i,k}$ (for all $i \in \{1, .., p\}$) horizontally broadcasts $a_{i,k}$ to processors $P_{i,*}$.

– each processor $P_{k,j}$ (for all $j \in \{1, .., q\}$) vertically broadcasts $b_{k,j}$ to processors $P_{*,j}$.

so that each processor $P_{i,j}$ can independently compute $c_{i,j} += a_{i,k} \times b_{k,j}$.

This algorithm is used in the current version of the ScaLAPACK library because it is scalable, efficient and it does not need any initial permutation (unlike Cannon's algoritm [12]). Moreover, on a homogeneous grid, broadcasts are performed as independent ring broadcasts (along the rows and the columns), hence they can be pipelined.

Of course, ScaLAPACK uses a blocked version of this algorithm to squeeze the most out state-of-the-art processors with pipelined arithmetic units and multilevel memory hierarchy [8, 6]. Each matrix coefficient in the description above is replaced by a $r \times r$ square block, where optimal values of r depend on the communication-to-computation ratio of the target computer.

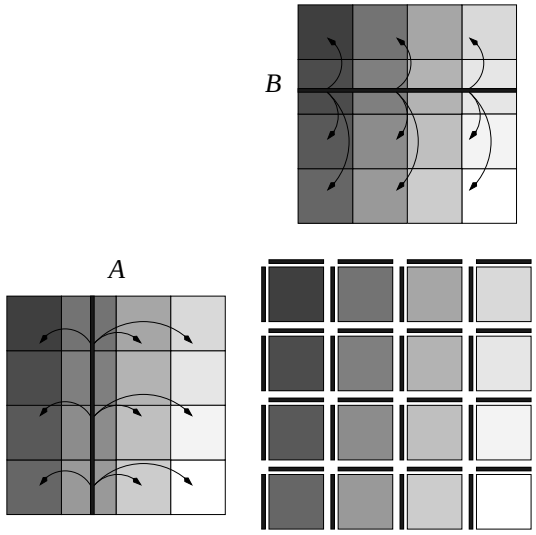


FIG. 1 – SUMMA Algorithm for matrix matrix multiplication on a 4×4 grid.

Finally, a level of virtualization is added : usually, the number of blocks $\lceil \frac{n}{r} \rceil \times \lceil \frac{n}{r} \rceil$ is much larger than the number of processors p^2. Thus, blocks are scattered in a cyclic fashion along both grid dimensions, so that each processor is responsible for updating several blocks at each step of the algorithm. An example is given in Figure 2 with $p = 4$ and $\lceil \frac{n}{r} \rceil = 10$.

1	2	3	4	1	2	3	4	1	2
5	6	7	8	5	6	7	8	5	6
9	10	11	12	9	10	11	12	9	10
13	14	15	16	13	14	15	16	13	14
1	2	3	4	1	2	3	4	1	2
5	6	7	8	5	6	7	8	5	6
9	10	11	12	9	10	11	12	9	10
13	14	15	16	13	14	15	16	13	14
1	2	3	4	1	2	3	4	1	2
5	6	7	8	5	6	7	8	5	6

FIG. 2 – Processors are numbered from 1 to 16. This figure represents the distribution of 10×10 matrix blocks onto 4×4 processors.

2.2 Complexity

Suppose we have a $p \times p$ grid and that communications cannot be performed in parallel. In what follows, we denote by $t_c r^3$ the time needed to compute sequentially the product of two $r \times r$ matrices and by τr^2 the time needed to transfer a $r \times r$ matrix between two

processors. The overall time needed to compute the product of two matrices of order n is given by

$$\begin{aligned} C_\Sigma &= \sum_{k=1}^{p} \left[\underbrace{t_c \left(\frac{n}{p}\right)^3}_{\text{computations}} + \underbrace{\sum_{i=1}^{p} \sum_{j \neq k} \tau \left(\frac{n}{p}\right)^2}_{\text{horizontal communications}} + \underbrace{\sum_{j=1}^{p} \sum_{i \neq k} \tau \left(\frac{n}{p}\right)^2}_{\text{vertical communications}} \right] , \\ &= \sum_{k=1}^{p} \left[t_c \left(\frac{n}{p}\right)^3 + \tau(p-1)\frac{n^2}{p} + \tau(p-1)\frac{n^2}{p} \right] , \\ &= t_c \frac{n^3}{p^2} + 2(p-1)n^2\tau \, , . \end{aligned}$$

If all necessary communications can be performed in parallel, then the overall time needed to compute the product of matrices of order n is given by

$$\begin{aligned} C_{\max} &= \sum_{k=1}^{p} \left[\underbrace{t_c \left(\frac{n}{p}\right)^3}_{\text{computations}} + \underbrace{\max_{i=1}^{p} \max_{j \neq k} \tau \left(\frac{n}{p}\right)^2}_{\text{horizontal communications}} + \underbrace{\max_{j=1}^{p} \max_{i \neq k} \tau \left(\frac{n}{p}\right)^2}_{\text{vertical communications}} \right] , \\ &= \sum_{k=1}^{p} \left[t_c \left(\frac{n}{p}\right)^3 + \tau\frac{n^2}{p^2} + \tau\frac{n^2}{p^2} \right] , \\ &= t_c \frac{n^3}{p^2} + 2n^2\tau \, . \end{aligned}$$

3 Heterogeneous Communications

Let us suppose the cost of a unitary communication is not uniform anymore and let us denote by $d(P_i, P_j)$ (as opposed to τ in the homogeneous case) the cost of the transfer of one block between processors P_i and P_j. Suppose that the p^2 processors are arranged as a $p \times p$ grid thanks to a bijective mapping π from $[\![1,p]\!] \times [\![1,p]\!]$ to $[\![1,p^2]\!]$.

– The time needed to compute the product of two matrices of order n if the communications cannot be handled in parallel is given by

$$C_\Sigma = \sum_{k=1}^{p} \left[\underbrace{t_c \left(\frac{n}{p}\right)^3}_{\text{computations}} + \underbrace{\sum_{i=1}^{p} \sum_{j \neq k} d(P_{\pi(i,k)}, P_{\pi(i,j)}) \left(\frac{n}{p}\right)^2}_{\text{horizontal communications}} + \underbrace{\sum_{j=1}^{p} \sum_{i \neq k} d(P_{\pi(k,j)}, P_{\pi(i,j)}) \left(\frac{n}{p}\right)^2}_{\text{vertical communications}} \right] . \tag{3.1}$$

The cost of the computations does not depend on the mapping π but minimizing the overall computational time needed to compute the product ot two matrices turns out to be equivalent to minimizing

$$\sum_{k=1}^{p} \left(\underbrace{\sum_{i=1}^{p} \sum_{j \neq k} d(P_{\pi(i,k)}, P_{\pi(i,j)})}_{\text{horizontal communications}} + \underbrace{\sum_{j=1}^{p} \sum_{i \neq k} d(P_{\pi(k,j)}, P_{\pi(i,j)})}_{\text{vertical communications}} \right) .$$

and therefore to solve the following optimization problem :

Definition 1 (Hete-Comm-Sum) *Given a metric d on Processors $P_1, \ldots, P_{p^2}$, find a bijective mapping π from $[\![1,p]\!] \times [\![1,p]\!]$ to $[\![1,p^2]\!]$ minimizing*

$$\sum_{k=1}^{p} \sum_{i,j} d(P_{\pi(k,j)}, P_{\pi(i,j)}) + d(P_{\pi(i,k)}, P_{\pi(i,j)})$$

– If all necessary communications can be handled in parallel, then the overall time needed to compute the product of two matrices of order n is given by

$$C_{\max} = \sum_{k=1}^{p} \left[\underbrace{t_c \left(\frac{n}{p}\right)^3}_{\text{computations}} + \underbrace{\max_{i=1}^{p} \max_{j \neq k} d(P_{\pi(i,k)}, P_{\pi(i,j)}) \left(\frac{n}{p}\right)^2}_{\text{horizontal communications}} + \underbrace{\max_{j=1}^{p} \max_{i \neq k} d(P_{\pi(k,j)}, P_{\pi(i,j)}) \left(\frac{n}{p}\right)^2}_{\text{vertical communications}} \right] . \tag{3.2}$$

As previously, the cost of computations does not depend on the mapping π and in this case, minimizing the time needed to compute the product of two matrices of order n turns out to be equivalent to the following optimization problem :

Definition 2 (Hete-Comm-Max) *Given a metric d on processors $P_1, \ldots, P_{p^2}$, find a bijective mapping π from $[\![1,p]\!] \times [\![1,p]\!]$ to $[\![1,p^2]\!]$ minimizing*

$$\sum_{k=1}^{p} \left(\max_{i,j} d(P_{\pi(k,j)}, P_{\pi(i,j)}) + \max_{i,j} d(P_{\pi(i,k)}, P_{\pi(i,j)}) \right)$$

In the next section, we show that (Hete-Comm-Sum) is an NP-Complete optimization problem.

4 Complexity

Theorem 1 Hete-Comm-Sum *is NP-Complete.*

In this section, we give the sketch of the proof of the NP-Completeness of Hete-Comm-Sum. The entire proof, which is long and technical, can be found in Appendix.

The decision problem associated to Hete-Comm-Sum is the following :

Definition 3 (Hete-Comm-Sum-Dec) *Given a metric d on processors $P_1, \ldots, P_{p^2}$, and a bound B, is there a bijective mapping π from $[\![1,p]\!] \times [\![1,p]\!]$ to $[\![1,p^2]\!]$ such that*

$$\sum_{k=1}^{p} \sum_{i,j} d(P_{\pi(k,j)}, P_{\pi(i,j)}) + d(P_{\pi(i,k)}, P_{\pi(i,j)}) \leq B$$

We select the following NP-complete problem for the reduction :

Definition 4 *2-Partition-Equal* (2P-eq-Opt)
Given a set of p non-negative integers $\mathcal{A} = \{a_1, \ldots, a_p\}$, is there a partition of $\{1, \ldots, p\}$ into two subsets $\mathcal{A}_1$ and $\mathcal{A}_2$ such that

$$\sum_{i \in \mathcal{A}_1} a_i = \sum_{i \in \mathcal{A}_2} a_i \text{ and } card(\mathcal{A}_1) = card(\mathcal{A}_2)$$

We consider an arbitrary instance of the 2-Partition-Equal problem, i.e. a set $\mathcal{A} = \{a_1, \ldots, a_{2n}\}$ of $2n$ integers. We have to polynomially transform this instance into an instance of the Hete-Comm-Sum-Dec which has a solution if and only if the original instance of 2P-eq-Opt($\mathcal{A}$) has a solution.

4.1 Hete-Comm-Sum instance

Our instance will be made of a set $(n+1)^2$ processors split into three areas : P^{I}, $(P_1^{\mathrm{II}}, \ldots, P_{2n}^{\mathrm{II}})$, and $(P_1^{\mathrm{III}}, \ldots, P_{n^2}^{\mathrm{III}})$.

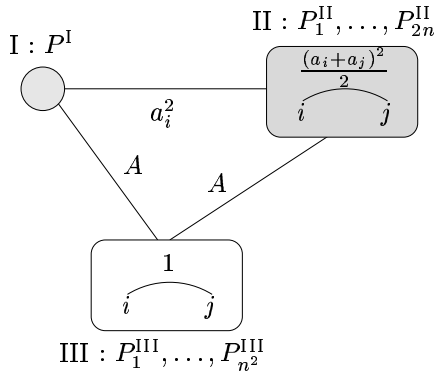
Let $M = \max_i a_i$, $\forall i, \quad b_i = a_i + 2nM$, $A = 32n^3 \max_i b_i^2$, $A' = \sum_i b_i^2$ and $S = \sum b_i$. . Thus, the following inequalities hold true :

$$\forall i,j : A \geq \frac{(b_i + b_j)^2}{2}, \tag{4.1}$$

$$\forall i,j,k : b_k^2 \leq \frac{(b_i + b_j)^2}{2}. \tag{4.2}$$

Intuitively, the b_i^2's are large and close together and are small compared to A.

The metric between the processors is defined as follows :



$$\begin{aligned}
P^{\mathrm{I}} &\longleftrightarrow P_i^{\mathrm{II}} : b_i^2 \\
P^{\mathrm{I}} &\longleftrightarrow P_i^{\mathrm{III}} : A \\
P_i^{\mathrm{II}} &\longleftrightarrow P_j^{\mathrm{II}} : \frac{(b_i + b_j)^2}{2} \\
P_i^{\mathrm{II}} &\longleftrightarrow P_j^{\mathrm{III}} : A \\
P_i^{\mathrm{III}} &\longleftrightarrow P_i^{\mathrm{III}} : 1
\end{aligned}$$

Finally, the bound B is set to

$$4n^2A + n^2(n+1) + (2n-1)A' + \frac{S^2}{2}.$$

4.2 Sketch of the proof.

In this section, we briefly describe the sketch of the proof of the NP-Completeness of (Hete-Comm-Sum-Dec).

First, we prove that the point to point distance we have just defined is a true metric, i.e. that triangular inequality holds true between any triplet of processors.

The second step of the proof is to study the possible positions of processors P^{I} and P_i^{II} into the grid. We prove that the overall communication overhead will be smaller than B if and only if all the processors P^{I} and P_i^{II} belong to the first row or to the first column of processors, as depicted in Figure 3.

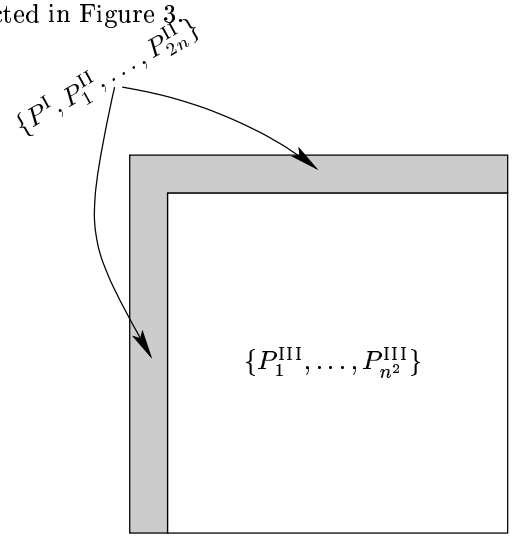


FIG. 3 – Optimal solution organization

The third step of the proof is to study the possible position of processor P^{I} into the first row and the first column of processors. We prove that the overall communication overhead will be smaller than B if and only if the processor P^{I} belongs both to the first row and to the first column of processors, as depicted in Figure 4.

The last step consists in proving that the overall communication overhead is smaller than B if and only if the communication overhead involving the processors of the first column is well balanced with the communication overhead involving the processors of the first row. Then, we prove that tha above condition can be fulfilled if and only if the original instance of 2P-eq-Opt($\mathcal{A}$) has a solution.

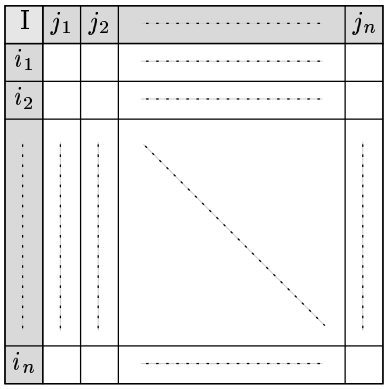


FIG. 4 – Optimal solution organization

This achieves the proof of the NP-Completeness of (Hete-Comm-Sum-Dec).

5 Related Problems

In this section, we recall some results on related problems with homogeneous communication links but processors running at different speeds. In this case, it is not always possible to reach perfect load balancing with 2D grids, and we have to consider different distribution schemes. Figure 5 depicts some other possible data-allocation schemes that are well-suited to heterogeneous platforms.

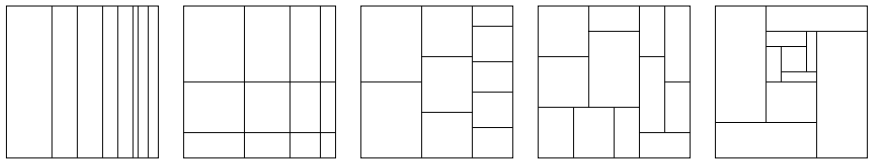


FIG. 5 – Tiling taxonomy

As in Section 2.2, we have to consider two different situations, according to the possibility of handling communications in parallel. For each data allocation scheme depicted in Figure 5, and for each hypothesis concerning the communication network, we give the complexity of the resulting optimization problem in table of Figure 6.

In this paper, we have proved that Hete-Comm-Sum is NP-Complete but we still do not know the complexity of Hete-Comm-Max. Note that our proof of the NP-Completeness of Hete-Comm-Sum strongly relies on the allocation of the processors into a grid. If we relax this

<table>
<tr><th></th><th>1D</th><th>2D</th><th>Column-based</th><th>Recursive</th><th>Unconstrained</th></tr>
<tr><td>$\sum$</td><td rowspan="2">Polynomial</td><td rowspan="2">NP-complete [2]</td><td>Polynomial [3]</td><td>No known results</td><td>NP-complete. Guaranteed heuristic with 7/4 bound. [4]</td></tr>
<tr><td>max</td><td>NP-complete [2] Guaranteed heuristic with $2/\sqrt{3}$ bound.</td><td>No known results</td><td>NP-complete. Guaranteed heuristic with $2/\sqrt{3}$ bound. [4]</td></tr>
</table>

FIG. 6 – Various complexity results on heterogeneous platforms

topological constraint, the communication overhead is much more difficult and the instance we considered in Section 4 is not optimal anymore.

6 Conclusion

In this paper, we have shown that deriving efficient strategies for data-allocation turns out to be surprisingly difficult (NP-Complete) as soon as the communication network is heterogeneous. This complexity result is not of practical interest since the case of heterogeneous communication links and homogeneous processors may seem surprising, but it states that the algorithmic complexity in the context of heterogeneous resources may come either from the heterogeneity of the processors or from the heterogeneity of the communication links.

Références

[1] R. Agarwal, F. Gustavson, and M. Zubair. A high performance matrix multiplication algorithm on a distributed-memory parallel computer, using overlapped communication. *IBM J. Research and Development*, 38(6) :673–681, 1994.

[2] O. Beaumont, V. Boudet, A. Legrand, F. Rastello, and Y. Robert. Heterogeneity considered harmful to algorithm designers. Technical Report RR-2000-24, LIP, ENS Lyon, June 2000. Available at `www.ens-lyon.fr/LIP/`.

[3] O. Beaumont, V. Boudet, F. Rastello, and Y. Robert. Matrix-matrix multiplication on heterogeneous platforms. Technical Report RR-2000-02, LIP, ENS Lyon, January 2000. Short version appears in the proceedings of ICPP'2000.

[4] O. Beaumont, V. Boudet, F. Rastello, and Y. Robert. Partitioning a square into rectangles : NP-completeness and approximation algorithms. Technical Report RR-2000-10, LIP, ENS Lyon, February 2000.

[5] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. SIAM, 1997.

[6] J. Choi, J. Demmel, I. Dhillon, J. Dongarra, S. Ostrouchov, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. ScaLAPACK : A portable linear algebra library for distributed memory computers - design issues and performance. *Computer Physics Communications*, 97 :1–15, 1996. (also LAPACK Working Note #95).

[7] P.E. Crandall and M.J. Quinn. Block data decomposition for data-parallel programming on a heterogeneous workstation network. In *2nd International Symposium on High Performance Distributed Computing*, pages 42–49. IEEE Computer Society Press, 1993.

[8] J. J. Dongarra and D. W. Walker. Software libraries for linear algebra computations on high performance computers. *SIAM Review*, 37(2) :151–180, 1995.

[9] G. Fox, S. Otto, and A. Hey. Matrix algorithms on a hypercube i : matrix multiplication. *Parallel Computing*, 3 :17–31, 1987.

[10] M. Kaddoura, S. Ranka, and A. Wang. Array decomposition for nonuniform computational environments. *Journal of Parallel and Distributed Computing*, 36 :91–105, 1996.

[11] A. Kalinov and A. Lastovetsky. Heterogeneous distribution of computations while solving linear algebra problems on networks of heterogeneous computers. In P. Sloot, M. Bubak, A. Hoekstra, and B. Hertzberger, editors, *HPCN Europe 1999*, LNCS 1593, pages 191–200. Springer Verlag, 1999.

[12] V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to Parallel Computing*. The Benjamin/Cummings Publishing Company, Inc., 1994.

A Proof of the NP-Completeness of Hete-Comm-Sum-Dec

We detail the proof whose sketch is given in Section 4.

A.1 The distance is a true metric

In order to prove that the distance we have defined is a true metric, we need to prove that triangular inequality holds true for any triplet of processors.

i) Communications between P^{I} and P_i^{II} : We have to check that

$$\forall P,\ d(P^{\mathrm{I}}, P_i^{\mathrm{II}}) \leq d(P^{\mathrm{I}}, P) + d(P, P_i^{\mathrm{II}})$$

– If $P = P_i^{\mathrm{II}}$

$$d(P^{\mathrm{I}}, P_i^{\mathrm{II}}) \leq d(P^{\mathrm{I}}, P_j^{\mathrm{II}}) + d(P_j^{\mathrm{II}}, P_i^{\mathrm{II}}) \Leftrightarrow b_i^2 \leq b_j^2 + \frac{(b_i + b_j)^2}{2}$$

This holds true since $b_i^2 \leq \frac{(b_i+b_j)^2}{2}$ (equation 4.2).

– If $P = P_i^{\mathrm{III}}$

$$d(P^{\mathrm{I}}, P_i^{\mathrm{II}}) \leq d(P^{\mathrm{I}}, P_j^{\mathrm{III}}) + d(P_j^{\mathrm{III}}, P_i^{\mathrm{II}}) \Leftrightarrow b_i^2 \leq A + A,$$

This holds true because $A \geq b_i^2$ (using equations 4.1 and 4.2).

ii) Communications between P^{I} and P_i^{III} : We have to check that

$$\forall P,\ d(P^{\mathrm{I}}, P_i^{\mathrm{III}}) \leq d(P^{\mathrm{I}}, P) + d(P, P_i^{\mathrm{III}})$$

– If $P = P_i^{\mathrm{II}}$

$$d(P^{\mathrm{I}}, P_i^{\mathrm{III}}) \leq d(P^{\mathrm{I}}, P_j^{\mathrm{II}}) + d(P_j^{\mathrm{II}}, P_i^{\mathrm{III}}) \Leftrightarrow A \leq b_j^2 + A$$

– If $P = P_i^{\mathrm{III}}$

$$d(P^{\mathrm{I}}, P_i^{\mathrm{III}}) \leq d(P^{\mathrm{I}}, P_j^{\mathrm{III}}) + d(P_j^{\mathrm{III}}, P_i^{\mathrm{III}}) \Leftrightarrow A \leq A + A$$

iii) Communications between P_i^{II} and P_j^{II} : We have to check that

$$\forall P,\ d(P_i^{\mathrm{II}}, P_j^{\mathrm{II}}) \leq d(P_i^{\mathrm{II}}, P) + d(P, P_j^{\mathrm{II}})$$

– If $P = P^{\mathrm{I}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{II}}) \leq d(P_i^{\mathrm{II}}, P^{\mathrm{I}}) + d(P^{\mathrm{I}}, P_j^{\mathrm{II}}) \Leftrightarrow \frac{(b_i + b_j)^2}{2} \leq b_i^2 + b_j^2 \Leftrightarrow 2b_ib_j \leq b_i^2 + b_j^2$$
$$\Leftrightarrow 0 \leq (b_i - b_j)^2$$

– If $P = P_k^{\mathrm{II}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{II}}) \leq d(P_i^{\mathrm{II}}, P_k^{\mathrm{II}}) + d(P_k^{\mathrm{II}}, P_j^{\mathrm{II}}) \Leftrightarrow (b_i + b_j)^2 \leq (b_i + b_k)^2 + (b_k + b_j)^2,$$

This holds true because $(b_i + b_j)^2 \leq 2b_i^2 + 2b_j^2$ is smaller than $(b_i + b_k)^2 + (b_k + b_j)^2$ (using equation 4.2).

– If $P = P_k^{\mathrm{III}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{II}}) \leq d(P_i^{\mathrm{II}}, P_k^{\mathrm{III}}) + d(P_k^{\mathrm{III}}, P_j^{\mathrm{II}}) \Leftrightarrow \frac{1}{2}(b_i + b_j)^2 \leq A + A,$$

which is true (see equation 4.1).

iv) Communications between P_i^{II} and P_j^{III} : We have to check that

$$\forall P,\ d(P_i^{\mathrm{II}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{II}}, P) + d(P, P_j^{\mathrm{III}})$$

– If $P = P^{\mathrm{I}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{II}}, P^{\mathrm{I}}) + d(P^{\mathrm{I}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; A \leq b_i^2 + A$$

– If $P = P_k^{\mathrm{II}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{II}}, P_k^{\mathrm{II}}) + d(P_k^{\mathrm{II}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; A \leq \frac{1}{2}(b_i + b_k)^2 + A$$

– If $P = P_k^{\mathrm{III}}$

$$d(P_i^{\mathrm{II}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{II}}, P_k^{\mathrm{III}}) + d(P_k^{\mathrm{III}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; A \leq A + 1$$

v) Communications between P_i^{III} and P_j^{III} : We have to check that

$$\forall P,\; d(P_i^{\mathrm{III}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{III}}, P) + d(P, P_j^{\mathrm{III}})$$

– If $P = P^{\mathrm{I}}$

$$d(P_i^{\mathrm{III}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{III}}, P^{\mathrm{I}}) + d(P^{\mathrm{I}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; 1 \leq A + A$$

– If $P = P_k^{\mathrm{II}}$

$$d(P_i^{\mathrm{III}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{III}}, P_k^{\mathrm{II}}) + d(P_k^{\mathrm{II}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; 1 \leq A + A$$

– If $P = P_k^{\mathrm{III}}$

$$d(P_i^{\mathrm{III}}, P_j^{\mathrm{III}}) \leq d(P_i^{\mathrm{III}}, P_k^{\mathrm{III}}) + d(P_k^{\mathrm{III}}, P_j^{\mathrm{III}}) \;\Leftrightarrow\; 1 \leq 1 + 1$$

Thus, triangular inequality holds true and our instance of Hete-Comm-Sum-Dec is valid.

A.2 Position of processors P^{I} and P_i^{II}

Let us first suppose that there is a solution to the instance of Hete-Comm-Sum-Dec with the b_i's, i.e. that the overall communication cost is smaller than B.

In this section, we prove that if the overall communication cost is smaller than B, then the processors P^{I} and P_i^{II} have to be arranged as depicted in Figure 3. This proof consists in two main steps. First, we prove that the processors P^{I} and P_i^{II} have to belong to the first row or the first column, and then, we proove that processor P^{I} has to be in the left-upper corner.

We look for a bijective mapping of $(n+1)^2$ processors onto a $(n+1) \times (n+1)$ grid so that the overall amount of communications is smaller than B.

Let π a processor allocation function (a bijective mapping from $[\![1, n+1]\!] \times [\![1, n+1]\!]$ to $[\![1, (n+1)^2]\!]$). Let r_i (resp. c_j) denote the number of processors belonging to Area I or Area II in the i^{th} row (resp. the j^{th} column). Is is always possible to reindex rows and columns s.t. $r_1 \geq r_2 \geq \cdots \geq r_{n+1}$ and $c_1 \geq c_2 \geq \cdots \geq c_{n+1}$. We suppose in the following that these

conditions are fulfilled. Let us define ψ by $\psi(i,j) = 0$ if $P_{\pi(i.j)} \in \{P^{\mathrm{I}}, P_1^{\mathrm{II}}, \ldots, P_{2n}^{\mathrm{II}}\}$ and $\psi(i,j) = 1$ otherwise. The overall volume of communication is given by

$$C(\pi) = \sum_k \sum_i \sum_j \left(d(P_{\pi(k,j)}, P_{\pi(i,j)}) + d(P_{\pi(i,k)}, P_{\pi(i,j)}) \right)$$

that can also be written :

$$C(\pi) = \sum_i C_i^v(\pi) + \sum_j C_j^h(\pi) \text{ with } \begin{cases} C_i^v(\pi) = \sum_{j,k} d(P_{\pi(i,k)}, P_{\pi(i,j)}) \text{ and} \\ C_j^h(\pi) = \sum_{i,k} d(P_{\pi(k,j)}, P_{\pi(i,j)}) \end{cases} .$$

Thus,

$$\begin{aligned} C_i^v(\pi) &= \sum_{\substack{\psi(i,k)=0 \\ \psi(i,j)=0}} d(P_{\pi(i,k)}, P_{\pi(i,j)}) + \sum_{\substack{\psi(i,k)=1 \\ \psi(i,j)=1}} d(P_{\pi(i,k)}, P_{\pi(i,j)}) + \sum_{\psi(i,k)\neq\psi(i,j)} d(P_{\pi(i,k)}, P_{\pi(i,j)}) \\ &= \sum_{\substack{\psi(i,k)=0 \\ \psi(i,j)=0}} d(P_{\pi(i,k)}, P_{\pi(i,j)}) + (n-r_i)(n+1-r_i) + 2A \cdot \left(r_i(n+1-r_i) \right) \end{aligned}$$

Since $\forall i,\ b_i^2$ is small compared to A we can bound some terms in previous expression.

$$\begin{aligned} 0 \leq \sum_{\substack{\psi(i,k)=0 \\ \psi(i,j)=0}} d(P_{\pi(i,k)}, P_{\pi(i,j)}) + (n-r_i)(n+1-r_i) \ , &\leq \sum_{\substack{\psi(i,k)=0 \\ \psi(i,j)=0}} 2 \max_l b_l^2 + (n-r_i)(n+1-r_i) \ , \\ &\leq r_i(r_i-1) \cdot 2 \max_l b_l^2 + (n+1)^2 \\ &\leq (n+1)^2 \cdot 2 \max_l b_l^2 + (n+1)^2 \\ &\leq 4n^2 \max_l b_l^2 \text{ (when } n \text{ is big enough)} \ , \\ &\leq \frac{A}{8n} \ . \end{aligned}$$

Then, we have

$$2A \cdot c_j(n+1-c_j) \leq C_j^v(\pi) \leq 2A \cdot c_j(n+1-c_j) + \frac{A}{8n}.$$

Similarly,

$$2A \cdot r_i(n+1-r_i) \leq C_i^h(\pi) \leq 2A \cdot r_i(n+1-r_i) + \frac{A}{8n} \ ,$$

and finally

$$C(\pi) = 2A \cdot \left(\sum_i r_i(n+1-r_i) + \sum_j c_j(n+1-c_j) \right) + R \text{ with } R \in [0, A/4].$$

Since $B = 4n^2A + n^2(n+1) + (2n-1)A' + \frac{S^2}{2}$, and $n^2(n+1) + (2n-1)A' + \frac{S^2}{2} < \frac{A}{4}$, then

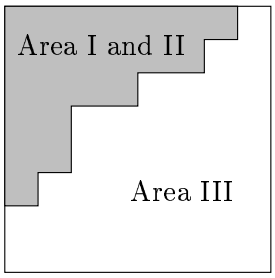
$$C(\pi) \leq B \Longrightarrow \left(\sum_i r_i(n+1-r_i) + \sum_j c_j(n+1-c_j) \right) \leq 2n^2.$$

Moreover, $\left(\sum_i r_i(n+1-r_i) + \sum_j c_j(n+1-c_j) \right) = 2(n+1)(2n+1) - \sum_i r_i^2 - \sum_j c_j^2$, and therefore,

$$C(\pi) \leq B \Longrightarrow \sum_i r_i^2 + \sum_j c_j^2 \geq 2(n+1)(2n+1) - 2n^2 = 2((n+1)^2 + n).$$

Let us first note that if $r_1 = c_1 = n+1$ and $\forall i > 1, \quad r_i = c_i = 1$, then $\sum_i r_i^2 + \sum_j c_j^2 = 2((n+1)^2+n)$. In what follows, we prove that with any other values for the r_i's and the $c_j's$, $\sum_i r_i^2 + \sum_j c_j^2 < 2((n+1)^2 + n)$, so that the only possible configuration for the processors P^{I} and P_i^{II} consists in putting them in the first column and the first row.

Lemma 1 *An optimal allocation has the following shape :*



Any processor belonging to Area I or Area II has for left and upper neighbour a processor belonging to Area I or Area II : processors belonging to Area I and Area II are grouped together on left and upper sides.

Proof Let us suppose that there exists an optimal allocation scheme s.t. there exists a processor belonging to Area I or Area II whose left neighboor belongs to Area III (the upper neighbour can be treated in a similar way). Then, it is possible to build a better allocation scheme (see Figure 7), by moving this processor to the left.

Let k be the index of the column of this processor. Let us define r' and c', the indices obtained ater the permutation. r' and c' are given by $r'_i = r_i \ \forall i$, $c'_j = c_j \ \forall j \notin k, k-1$, $c'_k = c_k - 1$ and $c'_{k-1} = c_k + 1$.

Then,

$$\begin{aligned} \sum_i {r'_i}^2 + \sum_j {c'_j}^2 - \sum_i r_i^2 - \sum_j c_j^2 &= {c'_k}^2 + {c'_{k-1}}^2 - c_k^2 - c_{k-1}^2 \\ &= (c_k - 1)^2 + (c_{k-1} + 1)^2 - c_k^2 - c_{k-1}^2 \\ &= 2 + 2(c_{k-1} - c_k) > 0, \end{aligned}$$

which is absurd since the cost of the solution after permutation is smaller. □

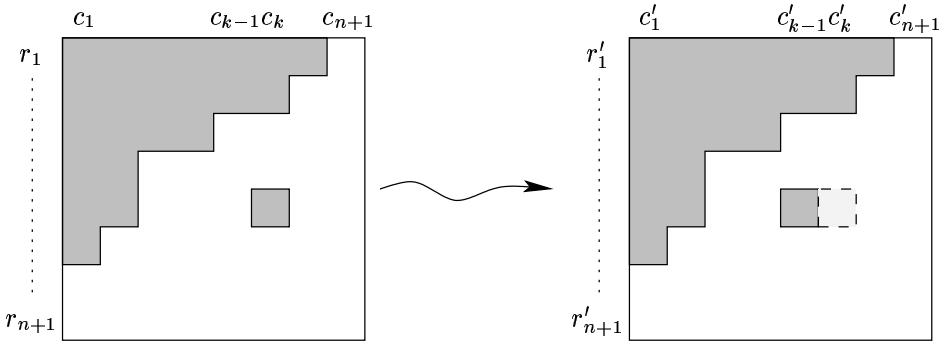


FIG. 7 – Minimizing communications between $\{P^{\mathrm{I}}, P_1^{\mathrm{II}}, \ldots, P_{2n}^{\mathrm{II}}\}$ and $\{P_1^{\mathrm{III}}, \ldots, P_{n^2}^{\mathrm{III}}\}$: grouping on the top and on the left

Lemma 2 *Optimal allocations satisfy :*

$$\begin{cases} \forall i \in [\![2, n+1]\!] : r_i \geq 2 \Rightarrow c_1 + r_i - 1 > n+1 \\ \forall j \in [\![2, n+1]\!] : c_j \geq 2 \Rightarrow c_j + r_1 - 1 > n+1 \end{cases}$$

Proof Let us suppose for example that there exists k s.t. $r_k \geq 2$ and $c_1 + r_k - 1 \leq n+1$.

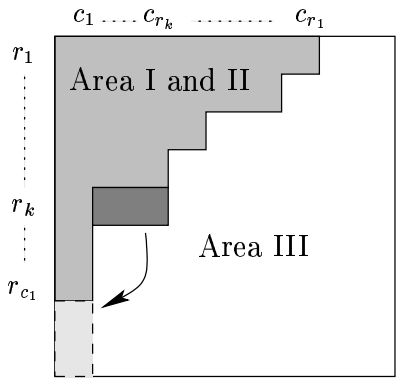


FIG. 8 – Minimizing communications between $\{P^{\mathrm{I}}, P_1^{\mathrm{II}}, \ldots, P_{2n}^{\mathrm{II}}\}$ and $\{P_1^{\mathrm{III}}, \ldots, P_{n^2}^{\mathrm{III}}\}$: grouping on the upper and leftmost sides

We want to maximize $\sum_{j=1}^{r_1} c_j^2 + \sum_{i=1}^{c_1} r_i^2$, which is related to the overall amount of communications. that has to be the largest possible. Moving $r_k - 1$ processors from the k^{th} row to the first column, we get the allocation depicted in Figure 8. In this case, the indices r' and

c' defining the new allocation are given by :

$$r'_i = \begin{cases} r_i & \text{if } i \in [\![1, k-1]\!] \\ 1 & \text{if } i = k \\ r_i & \text{if } i \in [\![k+1, c_1]\!] \\ 1 & \text{if } i > c_1 \end{cases} \text{ and } c'_j = \begin{cases} c_1 + r_k - 1 & \text{if } j = 1 \\ c_j - 1 & \text{if } j \in [\![2, r_k]\!] \\ c_j & \text{if } j > r_k \end{cases},$$

what leads to

$$\begin{aligned} \sum_j {c'_j}^2 + \sum_i {r'_i}^2 &= \left((c_1 + r_k - 1)^2 + \sum_{j=2}^{r_k} (c_j - 1)^2 + \sum_{j=r_k+1}^{r_1} c_j^2 \right) \\ &\quad + \left(\sum_{i=1}^{k-1} r_i^2 + 1 + \sum_{i=k+1}^{r_1} r_i^2 + (r_k - 1) \right), \\ &= \left(c_1^2 + 2c_1(r_k - 1) + r_k^2 - 2r_k + 1 + \sum_{j=2}^{r_k} c_j^2 - 2\sum_{j=2}^{r_k} c_j + (r_k - 1) + \sum_{j=r_k+1}^{r_1} c_j^2 \right) \\ &\quad + \left(\sum_{i=1}^{k-1} r_i^2 + 1 + \sum_{i=k+1}^{r_1} r_i^2 + (r_k - 1) \right), \\ &= \left(\sum_{j=1}^{r_1} c_j^2 \right) + \left(\sum_{i=1}^{c_1} r_i^2 \right) + 2\sum_{j=2}^{r_k} (c_1 - c_j) \\ &> \sum_{j=1}^{r_1} c_j^2 + \sum_{i=1}^{c_1} r_i^2, \text{iff the } c_j\text{'s are not all equal} \end{aligned}$$

If the c_j's are not all equal, it leads to an absurdity. If the c_j's are equal then the r_i's have also to be equal and $r_1 c_1 = 2n + 1$ (the r_i's are all equal and processors of Area I and Area II are then grouped together in a rectangle).

- If $r_1 = 1$ then $c_1 = 2n + 1$, which is impossible since c_1 belongs to $[\![1, n + 1]\!]$ and if $r_1 = 2$ then $2n + 1 = r_1 c_1$ is even. Thus, $r_1 \geq 3$ and similarly $c_1 \geq 3$.
- Since $r_1 c_1 = 2n + 1$, at least one of r_1 and c_1 is smaller than $\sqrt{2n+1}$. Let us suppose (by symmetry) that $r_1 \leq \sqrt{2n+1}$.

In this case, $r_1 \in [\![3, \sqrt{2n+1}]\!]$ and

$$\sum_{j=1}^{r_1} c_j^2 + \sum_{i=1}^{c_1} r_i^2 = r_1 \left(\frac{2n+1}{r_1} \right)^2 + r_1^2 \left(\frac{2n+1}{r_1} \right).$$

Since previous expression is a decreasing function in r_1 on, we get

$$\sum_{j=1}^{r_1} c_j^2 + \sum_{i=1}^{c_1} r_i^2 \leq \frac{(2n+1)^2}{3} + 3(2n+1).$$

Nevertheless, allocating processors of Area I and Area II only in the first row and the first column leads to $\sum_{j=1}^{r_1} c_j^2 + \sum_{i=1}^{c_1} r_i^2 = 2((n+1)^2 + n)$, which is better than previous allocation if n is big enough.

Thus, optimal allocations satisfy

$$\forall i \in [\![2, n+1]\!] : r_i \geq 2 \Rightarrow c_1 + r_i - 1 > n + 1$$

The other case ($c_j \geq 2$ and $c_j + r_1 - 1 \leq n + 1$) is similar. □

Lemma 3 *The optimal allocation satisfies* $r_1 = c_1 = n + 1$ *and* $r_i = c_j = 1, \ \forall i, j \geq 2$.

Proof Suppose there exists $i \in [\![2, n+1]\!]$ s.t. $r_i \geq 2$. Then, $r_2 > 1$ and $\exists j \in [\![2, n+1]\!]$ s.t. $c_j \geq 2$.

Since we know that $r_i \geq 2 \Rightarrow c_1 + r_i - 1 > n + 1$ and $c_j \geq 2 \Rightarrow c_j + r_1 - 1 > n + 1$, we have $c_1 + r_i \geq n + 3$ and $c_j + r_1 \geq n + 3$. Moreover, since the number of processors belonging to Area I and Area II is larger than $r_1 + c_1 + r_i + c_j - 4$, then $2n + 1 \geq r_1 + c_1 + r_i + c_j - 4 \geq 2n + 6 - 4 = 2n + 2$, which is absurd. Optimal solutions are then organized as depicted in Figure 3.

□

We have shown that, in an optimal allocation, processors belonging to Area I and Area II are grouped together on the first row and the first column.

Lemma 4 *In an optimal allocation,* P^I *is in the leftmost upper corner.*

Proof

- Suppose we have an optimal allocation where the processor I is not in the leftmost upper corner. Let $I = \{i_1, \ldots, i_n\}$ (resp. $J = \{j_1, \ldots, j_n\}$) be the indices of Area II's processors located on the first column (resp. the first row). We will suppose in the following that processor I is in the first column but the other case can be treated in a very similar way.
  Since the leading term of the communications overhead has already been minimized (the communications between Area III and Area I/II), we need to minimize internal communications in Area I/II.
  Horizontal communications are then given by

$$\begin{aligned} C_h &= \sum_l \frac{(b_{j_l} + b_{i_1})^2}{2} + \sum_k \sum_{l \neq k} \frac{(b_{j_l} + b_{j_k})^2}{2} + \sum_k \frac{(b_{j_k} + b_{i_1})^2}{2}, \\ &= \sum_k \sum_{l \neq k} \frac{(b_{j_l} + b_{j_k})^2}{2} + \sum_k (b_{j_k} + b_{i_1})^2 \end{aligned}$$

and vertical communications are given by

$$C_v = \sum_k \sum_{l \neq k} \left(\frac{(b_{i_l} + b_{i_k})^2}{2} + b_{i_k}^2 \right) + \sum_l b_{i_l}^2 ,$$
$$= \sum_k \sum_{l \neq k} \frac{(b_{i_l} + b_{i_k})^2}{2} + 2 \sum_l b_{i_l}^2 .$$

The overall cost of communications between processors of Area I/II and Area III is therefore given by

$$C = \sum_k \sum_{l \neq k} \frac{(b_{i_l} + b_{i_k})^2}{2} + \frac{(b_{j_l} + b_{j_k})^2}{2} + \sum_l (b_{j_l} + b_{i_1})^2 + 2 \sum_l b_{i_l}^2$$

– Consider the same allocation as before where P^{I} is moved in the leftmost upper corner. Thus, the communications during the first step are given by :

$$C'_1 = \sum_l b_{i_l}^2 + \sum_l b_{j_l}^2$$

and the communications during the step $k > 1$ are given by :

$$C'_k = \left(\sum_{l \neq k} \frac{(b_{i_l} + b_{i_k})^2}{2} + b_{i_k}^2 \right) + \left(\sum_{l \neq k} \frac{(b_{j_l} + b_{j_k})^2}{2} + b_{j_k}^2 \right)$$

The overall cost of communications between processors of Area I/II and Area III is therefore given by

$$C' = \sum_k \sum_{l \neq k} \frac{(b_{i_l} + b_{i_k})^2}{2} + \frac{(b_{j_l} + b_{j_k})^2}{2} + 2 \sum_l b_{j_l}^2 + 2 \sum_l b_{i_l}^2 . \qquad \text{(A.1)}$$

Then,

$$C - C' = \sum_l (b_{j_l} + b_{i_1})^2 - 2 \sum_l b_{j_l}^2 > 0 \text{ (using equation 4.2).}$$

Processor I is then in the leftmost upper corner. □

We have therefore proved that if the overall communication cost is smaller than B, the the processors have to be arranged as depicted in Figure 4. In what follows, we slightly refine previous result. Indeed, we prove that if the overall communication cost is smaller than B then there must be a solution to 2P-eq-Opt with the a_i's.

A.3 Equivalence with 2P-eq-Opt

We know that optimal allocations are organized as depicted in Figure 4

Let $I = \{i_1, \ldots, i_n\}$ (resp. $J = \{j_1, \ldots, j_n\}$) denote the set of indices of processors belonging to Area II that are located on the first column (resp. the first row).

Thus, the overall communication cost is given by

$$C(\pi) = 4n^2A + n^2(n+1) + 2\sum_l b_{i_l}^2 + 2\sum_k b_{j_k}^2 + \sum_{k=1}^{n}\sum_{l\neq k}\left((b_{i_l}+b_{i_k})^2 + (b_{j_l}+b_{j_k})^2\right),$$

and therefore, the overall communication cost is smaller than B if and only if

$$2\sum_l b_{i_l}^2 + 2\sum_k b_{j_k}^2 + \sum_{k=1}^{n}\sum_{l\neq k}\left((b_{i_l}+b_{i_k})^2 + (b_{j_l}+b_{j_k})^2\right) \leq (2n-1)A' + \frac{S^2}{2}.$$

Let us note that

$$\sum_{k=1}^{n}\sum_{l\neq k}\left((b_{i_l}+b_{i_k})^2 + (b_{j_l}+b_{j_k})^2\right) = (2n-2)A' + 2\sum_{k=1}^{n}\sum_{l\neq k}(b_{i_l}b_{i_k} + b_{j_l}b_{j_k}),$$

and

$$2\sum_{k=1}^{n}\sum_{l\neq k}(b_{i_l}b_{i_k} + b_{j_l}b_{j_k}) = \left(\sum_k b_{i_k}\right)^2 + \left(\sum_k b_{j_k}\right)^2 - \sum_i b_i^2.$$

Therefore,

$$2\sum_l b_{i_l}^2 + 2\sum_k b_{j_k}^2 + \sum_{k=1}^{n}\sum_{l\neq k}\left((b_{i_l}+b_{i_k})^2 + (b_{j_l}+b_{j_k})^2\right) = 2A' + (2n-1)A' + \left(\sum_k b_{i_k}\right)^2 + \left(\sum_k b_{j_k}\right)^2$$

and if the overall cost of communications is smaller than B, than

$$\left(\sum_k b_{i_k}\right)^2 + \left(\sum_k b_{j_k}\right)^2 \leq \frac{S^2}{2}.$$

Since $\sum_k b_{i_k} + \sum_k b_{j_k} = S$, then

$$\left(\sum_k b_{i_k}\right)^2 + \left(\sum_k b_{j_k}\right)^2 \leq \frac{S^2}{2} \Longleftrightarrow \sum_k b_{i_k} = \sum_k b_{j_k} = \frac{S}{2}$$

and the partition of the processors of Area II into the first row and the first column solves the 2P-eq-Opt problem for the b_i's. Let us prove that this partition solves the 2P-eq-Opt problem with the a_i's. Since

$$\sum_k b_{i_k} = \sum_k b_{j_k} = \frac{S}{2},$$

then

$$\sum_k a_{i_k} + 2nM.n = \sum_k a_{j_k} + 2nM.n$$

and finally

$$\sum_k a_{i_k} = \sum_k a_{j_k}.$$

Therefore, if there is a solution to Hete-Comm-Sum-Dec with the b_i's, there is also a solution to 2P-eq-Opt with the a_i's.

Reciprocally, if there is solution to the 2P-eq-Opt with the a_i's, we can arrange the processors as depicted in Figure 4, and solve Hete-Comm-Sum-Dec with the b_i's.

This achieves the proof of the reduction of Hete-Comm-Sum-Dec to 2P-eq-Opt.

A.4 Conciseness of the transformation

In order to prove the NP-Completeness of Hete-Comm-Sum-Dec, we need to check the conciseness of the transformation of the instance of 2P-eq-Opt with the a_i's to the instance of Hete-Comm-Sum-Dec with the b_i's.

The size of the encoding of our initial instance $\mathcal{A} = \{a_1, \ldots, a_{2n}\}$ for 2P-eq-Opt is

$$c(\mathcal{A}) = \Omega\left(\sum_i \log a_i\right)$$

The size of the coding of our new instance $\mathcal{B} = \{b_1, \ldots, a_{2n}, A, \ldots, A\}$ for Hete-Comm-Sum is

$$c(\mathcal{B}) = O\left(\sum_i \log b_i + (n^3 - (2n+1)) \log A\right)$$

If we set $MAX = \max_i a_i$ then we have

$$\sum_i \log a_i \geq \log(MAX) + (n-1)\log(\min_i a_i) \geq \log(MAX) + (n-1)\log 2$$

and thus

$$c(\mathcal{A}) = \Omega(\sum_i \log a_i) = \Omega(n + \log MAX)$$

Since $b_i = a_i + 2n \max a_j \leq (2n+1) \max a_j$, we have

$$\log b_i \leq \log(2n+1) + \log(MAX) = O(n + \log MAX).$$

We have $A = 32n^3 \max_l b_l^2$ and thus,

$$\log A \leq \log 32 + 3\log n + 2(\log(2n+1) + \log(MAX)) = O(n + \log MAX).$$

Frow the previous equation, we deduce :

$$\begin{aligned} c(\mathcal{B}) =& O(\sum_i \log b_i + \log A) \\ =& O((2n+1)(n+\log MAX) + n^3(n+\log MAX)) \\ =& O((n+\log MAX)^4) \\ =& O(c(\mathcal{A})^4) \end{aligned}$$

This achieves the proof of the NP-Completeness of Hete-Comm-Sum-Dec.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399