Messages Scheduling for Data Redistribution between Clusters*

Johanne Cohen[1], Emmanuel Jeannot[2], and Nicolas Padoy[3]

[1] CNRS LORIA, Vandœuvre les Nancy, France jcohen@loria.fr
[2] LORIA, Université H. Poincaré, Vandœuvre les Nancy, France ejeannot@loria.fr
[3] École Normale Supérieure de Lyon, France npadoy@ens-lyon.fr

Abstract. In this paper we study the general problem of parallel data redistribution over a network. Given a set of communications between two parallel machines interconnected by a backbone, we wish to minimize the total time required for the completion of all communications assuming that communications can be preempted and that preemption comes with an extra cost. Our problem, called *k-Preemptive bipartite scheduling (KPBS)* is proven to be NP-Complete. Moreover we prove that approximating KPBS problem within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = \text{NP}$. In spite of this negative result, we study a lower bound on the cost of KPBS problem in terms of its parameters, and we propose an approximation algorithm with ratio 2 and fast heuristics.

1 Introduction

With the emergence of grid computing many scientific applications use code coupling technologies to achieve their computations where parts of the code are distributed among parallel resources interconnected by a network. Code coupling requires data to be redistributed from one parallel machine to another. For instance the NxM ORNL project [1] has for objective to specify a parallel data redistribution interface and CUMULVS [2] (which uses MxN) supports interactive and remote visualization of images generated by a parallel computer. In this paper we concentrate on the scheduling of the messages when a parallel data redistribution has to be realized on a network, called a backbone. Two parallel machines are involved in the redistribution : the one that holds the data and the one that will receive the data. If the parallel redistribution pattern involves a lot of data transfers, the backbone can become a bottleneck. Thus, in order to minimize the parallel data redistribution time and to avoid the overloading of the backbone it is required to schedule each data transfer.

In this paper, we revisit the problem of packet switching (in wavelength-division multiplexed (WDM) optical network [3–7] or in satellite-switched time division multiple access (SS/TDMA) [8–10]) in the context data redistribution.

Data redistribution has mainly been studied in the context of high performance parallel computing [11–13]. In this paper we study a generalization of the parallel data redistribution. Indeed, contrary to some previous works that only deal with block-cyclic redistribution [14, 13], here, no assumption is made on the redistribution pattern. Moreover, contrary to other works which assume that there is no bottleneck [11, 12], we suppose that the ratio between the throughput of the backbone and the throughput of each of the n nodes of the parallel machines is k. Hence, no more than k communications can take place at the same time. We study the problem for all values of k. We focus on the case $k < n$ (the backbone is a bottleneck) whereas the case $k \geq n$ has been tackled in [11, 12].

The contribution of this paper is the following. We prove that the problem of scheduling any parallel data redistribution pattern is NP-Complete for any value of k ($< n$) and that approximating our problem (called KPBS) within a factor smaller that $\frac{4}{3}$ is impossible unless $P = \text{NP}$. We exhibit a lower bound for the number of steps of the redistribution as well as a lower bound for the sum of the duration of each step and prove that both lower bounds are tight. Next, we propose two algorithms: a pseudo-polynomial approximation algorithm with ratio 2, and polynomial approximation algorithm with ratio 2. Finally, we study simple and fast heuristics that achieve a good average performance.

* This work is partially supported by the ARC INRIA redGRID

2 The Problem

2.1 Modelization of the Problem

We consider the following heterogeneous architecture made of two clusters of workstations $\mathcal{G}_1$ and $\mathcal{G}_2$ connected together by a backbone of throughput D. Let n_1 be the number of nodes of $\mathcal{G}_1$ and n_2 be the number of nodes of $\mathcal{G}_2$. All the nodes of the first cluster have a throughput d_1 and the nodes of the second have a throughput d_2.

Let us consider a parallel application that must execute the first part of its computation on $\mathcal{G}_1$ and the second part on $\mathcal{G}_2$. This is the case where an application is made of two parallel components such that each code is only available (for security/license reason) on one cluster.

During the execution of the application parallel data must be redistributed from the first cluster to the second one.

We assume that the communication pattern of the redistribution is computed by the application. This pattern is modeled by a *traffic matrix* $T = (t_{i,j})_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$, where $t_{i,j}$ represents the amount of information that must be exchanged between node i of cluster $\mathcal{G}_1$ and node j of cluster $\mathcal{G}_2$.

For a given traffic pattern and for a particular architecture our goal is to minimize the total transmission time. In order to do this, we need to optimize the scheduling of the messages such that the available bandwidth is used without generating congestion. In this work, we do not rely completely on the network transport layer (i.e. TCP). Indeed, due to the control of the flow, TCP tends to use only a fraction of the total available bandwidth when congestion occurs. Here, thanks to our knowledge of the underlying architecture, a large part of the congestion control is performed at the application level..

Let us consider the constraints relative to the communications. A transmitter (resp. receiver) cannot transmit (resp. receive) more than one message at a time (1-port model). However, we allow several messages between different transmitters and receivers to be transmitted simultaneously as long as the backbone is not saturated. A parallel transmission *step* is a communication phase in which there can be simultaneous transmissions between several transmitters and receivers. We denote by k the maximum number of simultaneous transmissions that can take place during one step. This number depends on the number of nodes (n_1 and n_2) of each cluster as well as on the bandwidth of the network card of each node (d_1 and d_2) and on the bandwidth of the backbone (D). We denote by d the speed of each communication.

For instance let us assume that $n_1 = 200$, $n_2 = 100$, $d_1 = 10$Mbit/s, $d_2 = 100$Mbit/s and $D = 1$GBbit/s($D = 1000$Mbit/s). In that case, $k = 100$ because $\mathcal{G}_1$ can send 100 outgoing communications at 10 Mbit/s generating a total of 1 Gbit/s aggregated bandwidth (which is supported by the backbone) and each network card of $\mathcal{G}_2$ can receive the data at d =10 Mbit/s.

A common approach to minimize the overall transmission time is to allow preemption, i.e. the possibility to interrupt the transmission of a message and complete it later. In practice, this involves a non-negligible cost, called *set-up delay* and denoted here by β, which is the time necessary to start a new *step*.

2.2 Formulation of the Problem

Let T be a traffic matrix, k be the maximum number of communications at each step, β be the startup delay and d be the speed of each communication.

We can normalize the problem by d and β as follows: (1) The traffic matrix T, can be replaced by the matrix $Q = (q_{i,j}) = (\frac{t_{i,j}}{d})_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$ that represents the communication times for each messages . (2) The matrix Q can be replaced by the matrix $M = (m_{i,j}) = (\frac{q_{i,j}}{\beta})_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$ that represents the fraction of setup delay required for sending each messages.

In the following we will always consider the normalized problem ($\beta = 1$).

The matrix M can be represented by a bipartite graph $G = (V_1, V_2, E)$ and a positive edge-weight function $w : E \to \mathbb{Q}$. Each node of cluster $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) is represented by a node of V_1 (resp. V_2). Hence, $|V_1| = n_1$ and $|V_2| = n_2$. The weight of an edge between node i and j is equal to $m_{i,j}$.

We use the 1-port model for the communication and at most k communications can occur during one step. Hence, a communication step is a weighted matching of G with at most k edges. The weights refer to

preemption. We denote the matching corresponding to a communication step by a *valid weighted matching* (for the remainding, a valid weighted matching contains at most k edges).

We call this problem *k-Preemptive bipartite scheduling (KPBS)*, formally defined as follows:

Given a weighted bipartite graph $G = (V_1, V_2, E, w)$ where $w : E \to \mathbb{Q}$, an integer[4] $k \geq 2$ find a collection $\{(M_1, W_1), (M_2, W_2), \ldots, (M_s, W_s)\}$ of valid weighted matchings such that:

1. Let w_i be the edge weight function of each matching M_i. It must respect the following inequalities: for any $e \in E$, $\sum_{i=1}^{s} w_i(e) \geq w(e)$. If $e \notin M_i$ then $w_i(e) = 0$.
2. For any $1 \leq i \leq s$, matching M_i has at most k edges ($|M_i| \leq k$) and its cost is equal to the rational number $W_i = \max_{e \in M_i} w_i(e)$.
3. $(\sum_{i=1}^{s} W_i) + s$ is minimized. In the normalized form of the problem, each step has a cost equal to W_i plus 1 for the setup cost.

In the remainder of this paper, note that for any solution S of $KPBS$, if the cost of S is $\alpha + s$, the number of steps is s and the useful transmission cost equals α. See Figure 1 for an example.

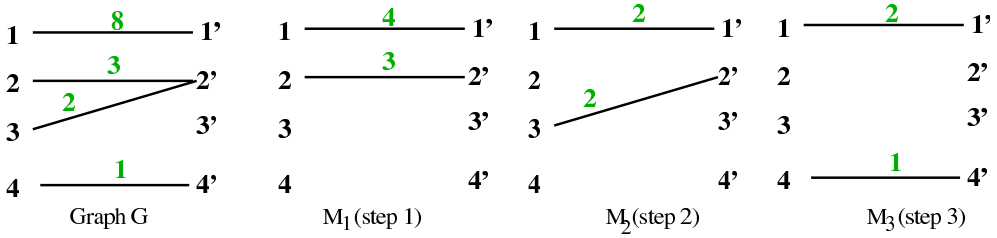

Fig. 1. An example for KPPS problem ($k = 2$). The cost of the solution is $8 + 3 = 11$

3 Complexity Results

This problem has already been proven NP-complete for the particular case where $k \geq \min(n_1, n_2)$ [15, 10]. We prove that it remains NP-complete for any fixed $k \geq 2$ (with a different reduction than in [15, 10]).

Theorem 1 *Let $k \geq 2$ be a fixed integer. KPBS is NP-complete.*

Moreover, we improve the result in [12]. We prove that one cannot approximate the problem KPBS within a factor smaller than 4/3 if $P \neq$ NP.

Theorem 2 *If $P \neq NP$, there is no polynomial time approximation algorithm for the problem KPBS with an approximation ratio smaller than* 4/3.

Proof. Theorem 1 and 2 are proven in [16].

4 Lower Bounds

Before giving a lower bound for the optimal solution, we give some graph notations. We define the weight $w(v)$ of a node v of G to be the sum of weights of all edges incident to vertex v. We denote the maximum of $w(v)$ over all vertices by $W(G)$. Let $P(G)$ be the sum of the weights of all edges of graph G. We denote the maximum degree of the bipartite graph G by $\Delta(G)$, its number of edges by $m(G)$ and its number of vertices by $n(G)$.

[4] the case $k = 1$ is not interesting: the backbone is saturated by one communication

Proposition 1 *Let $G = (V_1, V_2, E, w)$ be a weighted bipartite graph. Let k be an integer. The cost of the optimal solution for the instance $\langle G, k, \beta \rangle$ of KPBS is at least $\eta(G) = \eta_d(G) + \eta_s(G)$ where*

$$\eta_d(G) = max\left(W(G), \left\lceil \frac{P(G)}{k} \right\rceil\right) \quad and \quad \eta_s(G) = max\left(\Delta(G), \left\lceil \frac{m(G)}{k} \right\rceil\right)$$

Proof. $\eta_s(G)$ is a lower bound for the number of steps. The first term of the maximum accounts for the fact that two edges incident to the same node cannot appear in the same step and the second term for the fact that a step contains at most k edges. $\eta_d(G)$ is a lower bound for the useful transmission cost and is obtained similarly. The total cost is therefore minimized by $\eta_d(G) + \eta_s(G)$. □

Next, we study the quality of these lower bounds. The remainder of this section is to prove that there are polynomial time algorithms to optimize the number of steps (see Proposition 3) or the useful transmission cost (see Proposition 2).

Proposition 2 *Let G be a weighted bipartite multigraph. Then G can be decomposed such that the total transmission cost is $\eta_d(G)$.*

Proposition 3 *Let G be a weighted bipartite multigraph. Then G can be decomposed in $\eta_s(G)$ valid weighted matchings in polynomial time.*

Propositions 3 and 2 are equivalent. Indeed by setting all the weights to 1, Proposition 2 minimizes the number of steps because, in that case it is equal to the total transmission cost. On the contrary, by splitting all the edges into edges of weight 1, Proposition 3 gives a solution that minimizes the total transmission cost. We present a similar polynomial-time algorithm for Proposition 3 that will be used later.

The previous propositions can be seen as a consequence (see [17]) of a coloration theorem (given in [18] pages 132–133). Moreover, a proof of proposition 3 can be found in [9]. However, an other proof can be found in [16].

The decomposition is achieved in $\mathcal{O}(n(G)^{3/2} \times m(G)^3)$. The authors of article [8] provide a polynomial time algorithm that proves Proposition 2 for matrices, and shows that the number of steps is bounded by a polynomial in $n(G)$. We use it in section 5.

We separately studied η_s and η_d, what about η? There are quite simple graphs [17] (with all the edges having the same weight) such that η is not reached, and we can exhibit class of graphs (for instance graphs with edges having the same weight and with $k|m(G)$) for which it is.

5 Algorithms

The following algorithm approximates KPBS with a constant ratio.

Algorithm 1

Input: A weighted bipartite graph $G = (V_1, V_2, E, w)$ and an integer k
a rational number α
Output: A set of valid weighted matchings.

1. Split every edge e of G into $\lceil \frac{w(e)}{\alpha} \rceil$ edges having each a weight equal to α, which leads to a multigraph H.
2. Find $\eta_s(H)$ valid weighted matchings whose union is H.
3. Every matching represents a communication step of length α.

In each matching of the solution the edges have the same weight, and in order to evaluate the solution, we decide that all steps have the same length α, where α is a constant that will be fixed to 1. The algorithm splits each edge in edges of weight α (it is an idea used in [11]) to make a multigraph H, then we find a solution such that the number of matchings is minimum (thanks to Proposition 3).

Its complexity is $\mathcal{O}\left(n(H)^{3/2} \times m(H)^3\right) = \mathcal{O}\left(n(G)^{3/2} \times m(G)^3 \times W(G)^3\right)$ and therefore pseudo-polynomial since the running time of Algorithm 1 depends linearly on the weights of G.

Proposition 4 *Let $cost(G, \alpha)$ be the cost of the solution given by Algorithm 1. $cost(G, 1) \leq 2 \times \eta(G)$. Therefore, Algorithm 1 is a 2-approximation algorithm.*

Let us first consider a particular class of graphs such that the parameter η_s is equal to 1. Let G be a graph such that $\eta_s(G) = 1$. By definition, we have $\Delta(G) = 1$ and $m(G) \leq k$. Thus, the scheduling is composed of 1 step and the cost of this scheduling corresponds to the lower bound. For the remainder of the proof, we only consider graphs G such that $\eta_s(G) \geq 2$.

Proof of Proposition 4: Assume first, that the weights of the edges of G are multiple of α. The definitions of η_s and η_d imply $\alpha \times \eta_s(H) \leq \eta_d(G) + \alpha$ and therefore:

$$cost(G, \alpha) = \alpha \times \eta_s(H) + \eta_s(H) \leq \eta_d(G) + \frac{1}{\alpha} \times \eta_d(G) + \alpha + 1 \tag{1}$$

Since only graphs G such that $\eta_s(G) \geq 2$ are considered, we have $\eta(G) \geq \eta_d(G) + 2$. From equation 1, we get

$$cost(G, 1) \leq 2\eta_d(G) + 2 \leq 2\eta(G) - 2 \tag{2}$$

Therefore, the approximation ratio is 2 with $\alpha = 1$.

When the weights are not multiple of α, they are rounded up to the first multiple of α, to make a graph G', then the previous algorithm is applied to G'. So, from equation 1, we get

$$cost(G, \alpha) = cost(G', \alpha) \leq \eta_d(G') + \frac{1}{\alpha} \times \eta_d(G') + \alpha + 1 \tag{3}$$

We compare $\eta(G)$ to $\eta(G')$. We have $\eta_s(G') = \eta_s(G)$, but $\eta_d(G')$ differs:

$$\eta_d(G') = max\left(W(G'), \left\lceil \frac{P(G')}{k} \right\rceil\right) \tag{4}$$

$$\leq max\left(W(G) + (\alpha - 1)\Delta(G), \left\lceil \frac{P(G) + (\alpha - 1)m(G)}{k} \right\rceil\right)$$

$$\leq \eta_d(G) + (\alpha - 1) \times \eta_s(G) \tag{5}$$

Hence, from in-equations 3 and 5 we get:

$$cost(G, 1) \leq \eta_d(G')(1 + \frac{1}{\alpha}) + \alpha + 1 \leq 2\eta(G) + 2(1 - \eta_s(G)) \tag{6}$$

Since we only consider graphs G such that $\eta_s(G) \geq 2$, Algorithm 1 is a pseudo-polynomial time algorithm for KPBS with an approximation ratio 2. □

We use now this algorithm to describe a polynomial-time algorithm for KPBS with an approximation ratio 2. Given a graph G, we evaluate an expression depending on $P(G)$ that represents roughly the average cost of a step (expressed in the number of set-up delays), then depending on the result of its comparison with the number of set-up delays, we branch on the previous algorithm or on another one.

When $\gamma \leq 1$ all the weights of G are bounded, therefore Algorithm 1 is polynomial. Indeed $W(G) \leq P(G) \leq k(n^2(G) + n(G) + 1)$. This yields to a complexity of $\mathcal{O}(kn^{15/2}(G) \times m^3(G))$

We need to determine the approximation ratio in the second case (when executing line 3). The paper [8] gives (with a matrix formulation) a polynomial algorithm for optimizing the useful transmission cost with in the worst

Algorithm 2

Input: A bipartite graph G.
Output: A set of valid weighted matchings.

1. Calculate $\gamma = \frac{P(G)}{k \times (n(G)^2 + n(G) + 1)}$
2. If $\gamma \leq 1$, branch on Algorithm 1 with G and $\alpha = 1$ as input
3. Otherwise, branch on the algorithm which find the valid weighted matchings such that the useful transmission cost is minimized

case a number of steps lower than $(n(G)^2 + n(G) + 1)$. For this algorithm, we have: ($cost(G)$ being the cost of the solution given by Algorithm 2, when executing line 3).

$$cost(G) \leq \eta_d(G) + (n(G)^2 + n(G) + 1) \leq \eta_d(G) + \frac{P(G)}{k} \leq 2 \times \eta(G)$$

Therefore, we can deduce that:

Theorem 3 *There is a polynomial-time 2-approximation algorithm for KPBS.*

6 Heuristics

Algorithm 2 has a large complexity. In this section, we concentrate on fast heuristics that we have studied in practice.

Heuristic on weights

Input: A bipartite graph G.
Output: A set of valid weighted matchings.

1. Find a maximal matching.
2. Keep only the k (or less if there are less than k edges) edges whose weights are the biggest.
3. Set all the weights of the matching equal to the lowest one.
4. Subtract the matching from G.
5. Loop until there is no more edge left in G.

Here are two heuristics that appear to work well in practice (a heuristic on weighs and a heuristic on degrees). The heuristic on degrees is the same as the heuristic on weights except that line 2. is changed into "*2. Keep only the k (or less if there are less than k edges) edges with highest degrees.*".

Complexity: We use the Hungarian method of complexity $\mathcal{O}(m(G) \times n(G)^{1/2})$ for finding a maximum cardinality matching in a bipartite graph. For both heuristics, at each step, at least one edge is removed from G. Therefore, the complexity of both heuristics is $\mathcal{O}(m(G)^2 \times n(G)^{1/2})$ which is better than the complexity of algorithm 2.

Experiments: We have tested each heuristic (with k fixed) on a sample of 100 000 random graphs (the number of edges, the edges, and finally the weights were chosen randomly with a uniform distribution). We made a difference between lightly and heavily weighted graphs. Small weights were taken between 1 and 20, whereas large weights were

taken between 1 and 100 000. The result of a heuristic is calculated as the solution cost divided by the lower bound η. The plots show the average and the maximum calculated over the samples.

For these tests, the maximum is always below 2.5, even 1.8 for small weights, and the average is always below 2, and even 1.3 in case of large weights. Unfortunately, we didn't succeed into giving an approximation ratio for these two heuristics.

We explain the convex shape of the plots as follows:

- when $k = 1$ the two heuristics obtain the optimal solution which consists in one communication per steps;
- when k is greater than 2 and lower than a certain value (close to $n/2$), the quality of the solution degrades (compared to the lower bound); We believe that this is due to the fact that, at each step, the number of valid matchings increases;
- When k is greater than $n/2$ the quality of the solution tends to improve. At each stage of the two heuristics the choice of valid matchings decreases, therefore the heuristics are less likely to select bad valid matchings.

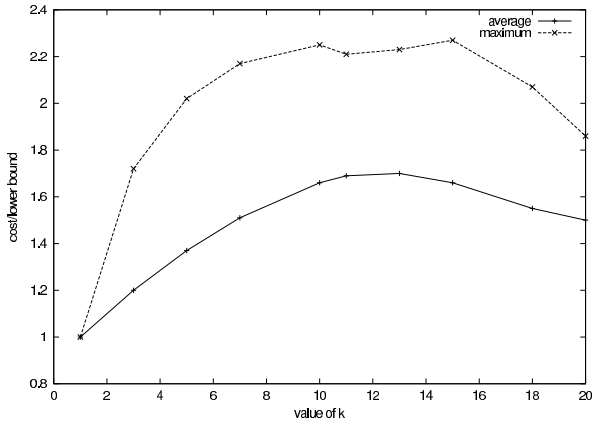

Fig. 2. Heuristic on weights. $n = 20$. Simulation on 100000 graphs with small weights per point.

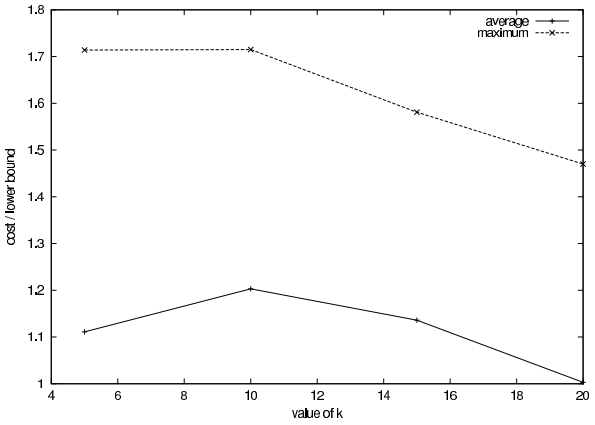

Fig. 3. Heuristic on edges. $n = 20$. Simulation on 100000 graphs with large weights per point.

7 Related Work

Up to our knowledge, there is no work on the KPBS problem in its generality ($n_1 \neq n_2$ and k can have any value, etc.).

This problem partially falls in a field originated by packet switching in communication systems for optical network called wavelength-division multiplexed (WDM) broadcast network [3–7].

The problem of minimizing the number of steps is studied in [9, 4], and the problem of minimizing the total cost is studied in [5].

In [3] and in [6], the author consider a version of the KPBS problem where the number of receivers is equal to the number of messages that can be transmitted at the same time ($k = n_2$) and where the set-up delay can be overlapped by the communication time (In [6] authors also assume that the size of all messages are the same). In that case, a list-scheduling algorithm is proven to be a 2-approximation algorithm in [3].

The case where the backbone is not a constraint ($k \geq \min(n_1, n_2)$) has been studied in [11, 12] and it is known as the *preemptive bipartite scheduling (PBS)*. PBS was proven to be NP-complete in [15, 10]. In [12], two different polynomial time 2-approximation algorithms for PBS have been proposed and in [11], an improvement of this result is given.

In the context of block cyclic redistribution several works exist [13, 14]. In this case the communication pattern is not arbitrary and and the backbone is not a constraint.

8 Conclusions

In this paper we have formalized and studied the problem (called KPBS) of redistributing parallel data over a backbone. Our contribution is the following. We have shown that KPBS remains NP-Complete when k is constant. We have shown that approximating the KPBS problem within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = \text{NP}$. We have studied lower bounds related to KPBS. We have proposed a polynomial time approximation algorithm with ratio 2. We have studied two fast and simple heuristics that have good properties in practice.

Our future work is directed towards studying the problem when the throughput of the backbone varies dynamically, when the redistribution pattern is not completely known in advance or when the network cards on each cluster are not all identical. We would also like to perform real tests on real architectures in order to compute a realistic value of the startup time and to be able to build a library for parallel redistribution.

References

1. Labs, O.R.N.: Mxn. (`http://www.csm.ornl.gov/cca/mxn`)
2. Geist, G.A., Kohl, J.A., Papadopoulos, P.M.: CUMULVS: Providing Fault-Tolerance, Visualization and Steering of Parallel Applications. International Journal of High Performance Computing Applications **11** (1997) 224 – 236
3. Choi, H., Choi, H.A., Azizoglu, M.: Efficient Scheduling of Transmissions in Optical Broadcast Networks. IEEE/ACM Transaction on Networking **4** (1996) 913–920
4. Ganz, A., Gao, Y.: A Time-Wavelength Assignment Algorithm for WDM Star Network. In: IEEE INFOCOM'92. (1992) 2144–2150
5. Mishra, M., Sivalingam, K.: Scheduling in WDM Networks with Tunable Transmitter and Tunable Receiver Architecture. In: NetWorld+Interop Engineers Conference, Las Vegas, NJ (1999)
6. Pieris, G.R., G.H., S.: Scheduling Transmission in WDM Broadcast-and-Select Networks. IEEE/ACM Transaction on Networking **2** (1994)
7. Rouskas, N., Sivaraman, V.: On the Design of Optimal TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies. In: IEEE INFOCOM'96. (1996) 1217–1224
8. Bongiovanni, G., Coppersmith, D., Wong, C.K.: An Optimum Time Slot Assignment Algorithm for an SS/TDMA System with Variable Number of Transponders. IEEE Transactions on Communications **29** (1981) 721–726
9. Gopal, I.S., Bongiovanni, G., Bonuccelli, M.A., Tang, D.T., Wong, C.K.: An Optimal Switching Algorithm for Multibean Satellite Systems with Variable Bandwidth Beams. IEEE Transactions on Communications **COM-30** (1982) 2475–2481
10. Gopal, I., Wong, C.: Minimizing the Number of Switching in an SS/TDMA System. IEEE Transactions on Communications (1985)
11. Afrati, F., Aslanidis, T., Bampis, E., Milis, I.: Scheduling in switching networks with set-up delays. In: AlgoTel 2002, Mèze, France (2002)
12. Crescenzi, P., Xiaotie, D., Papadimitriou, C.H.: On Approximating a Scheduling Problem. Journal of Combinatorial Optimization **5** (2001) 287–297
13. Desprez, F., Dongarra, J., Petitet, A., Randriamaro, C., Robert, Y.: Scheduling Block-Cyclic Array Redistribution. IEEE Transaction on Parallel and Distributed Systems **9** (1998) 192–205
14. Bhat, P.B., Prasanna, V.K., Raghavendra, C.S.: Block Cyclic Redistribution over Heterogeneous Networks. In: 11^{th} International Conference on Parallel and Distrinuted Computing Systems (PDCS 1998). (1998)
15. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problem. SIAM J. Comput. **5** (1976) 691–703
16. Cohen, J., Jeannot, E., Padoy, N.: Parallel Data Redistribution Over a Backbone. Technical Report RR-4725, INRIA (2003)
17. Padoy, N.: Redistribution de données entre deux grappes d'ordinateurs. Rapport de stage, de l'École Normale Supérieure de Lyon (2002)
18. Berge, C.: Graphs. North-Holland (1985)