www.czasopisma.pan.pl PAN www.journals.pan.pl
POLSKA AKADEMIA NAUK

BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 66, No. 2, 2018
DOI: 10.24425/122099

A note on the complexity of scheduling of communication-aware directed acyclic graph

J. MUSIAL[1]*, M. GUZEK[2], P. BOUVRY[2], and J. BLAZEWICZ[1,3]

[1] Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland
[2] Comp. Sci. and Commun. Res. Unit, University of Luxembourg, 6 Avenue de la Fonte, L-4364 Esch-sur-Alzette, Luxembourg
[3] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Z. Noskowskiego 12/14, 61-704 Poznan, Poland

Abstract. The most recent incarnation of distributed paradigm is cloud computing. It can be seen as the first widely accepted business model of mass consumption of the distributed computing resources. Despite the differences in business models and technical details regarding cloud platforms, the distributed computing underlies cloud. Communications in cloud systems include transmissions of the results of cloud applications, users interactions, and exchange of data between different services that compose applications. The latter becomes more critical as applications become richer as well as more complex, and may consist of services operated by various providers. The effective communication between components of cloud systems is thus critical to the end user satisfaction and to the success of cloud services. We will discuss different cloud computing models (communication aware and unaware). Main focus will be placed on communication-aware directed acyclic graph (CA-DAG), which extends the classical DAG model by explicitly modeling communication tasks. Moreover, we will analyze and consult computational complexity of this innovative distributed computation model inspired by the characteristics of cloud computing. Providing a proof of strong NP-hardness of the problem allows for a future implementation and evolution of the communication-aware DAG models.

Key words: computational complexity, models in cloud computing, communication-aware cloud computing, directed acyclic graphs, NP-hardness in cloud computing.

1. Introduction

Scheduling is undoubtedly one of the most important fields of operations research. The scheduling problems can be generally described as the allocation of resources over time to execute a group of tasks, which are a part of the processes [1]. Each task needs a particular resource to be completed. The nature of resources, tasks, connections and relations between them can be defined in many different ways. Predominantly, the scheduling is motivated by applications from industry and service operations management like: transportation planning, factory production lines, events planning, timetabling for general transportation purposes, vehicle routing, duty roster, and many more. Actually, if we perceive scheduling problems in general, they can be observed almost everywhere in real world situations.

If one runs many computers in parallel and the goal is to solve a problem computationally (which is very often complicated and intensive), it can be described as a parallel processing/computing [2]. The main idea is to provide the results in a time that no single computer could deliver. The idea is not new, however, its nature is experimental and therefore is being still strongly developed and studied. There are many new concepts that constantly try to increase speed of computations. Nevertheless, it has to be noted that the parallel processing is strongly connected with a networking and hardware issues.

If we consider a parallel application as a set of communicating processes then it can be easily modeled using a directed acyclic graph (DAG) [1, 3]. Most of the general scheduling theory assumptions can be applied. However, one should notice that the task execution in parallel distributed systems is connected with some communication delays. This communication delay cost occurs when two consecutive tasks are performed (on different machines, on the same machine, on the same processor, or even on the same processor's core).

The most recent incarnation of distributed paradigm is cloud computing [4]. It can be seen as the first widely accepted business model of mass consumption of distributed computing resources. The resources are offered as services (or their components) which means that they are no longer addressed to the IT professionals, but can be also directly offered to the end users. The users can use the resources seamlessly, as the necessary knowledge about a service set up and the underlying hardware [5].

The exact predictions of cloud market growth vary [6], but all of them recognize its large potential, justified by the strong growth in the past. Despite differences in business models and technical details regarding cloud platforms, the distributed computing underlies cloud. Communications in cloud systems include transmissions of the results of cloud applications, users interactions, and exchange of data between different services that compose applications. The latter becomes more critical as applications become richer and therefore, more complex. They may consist of services operated by various providers. The ef-

*e-mail: Jedrzej.Musial@cs.put.poznan.pl

Manuscript submitted 2017-06-20, revised 2017-10-14, initially accepted for publication 2017-10-17, published in April 2018.

fective communication between components of cloud systems is thus critical to the end user satisfaction and to the success of cloud services.

In this paper, we discuss the complexity of an innovative distributed computation model inspired by these characteristics of cloud computing. The model, communication-aware directed acyclic graph (CA-DAG), extends the classical DAG model by explicitly modeling communication tasks. It enables to model multicast communications or gather and scatter patterns, and explicitly schedule communications on various network topologies. Because of its relevance, the model has been already studied experimentally [7] however, rapid practical developments left the theoretical analysis behind. This paper is devoted to filling that gap.

2. Resource allocation in cloud computing

Basic requirements of cloud computing (such as managing multiple machines, providing services that work on different systems) have already been addressed and initially solved with some first ideas, analysis and computations with specific algorithms [8]. Further improvements require more holistic approaches, i.e. simultaneous taking into account different aspects and layers of the system to enhance efficiency and performance of cloud computing systems. New research opportunities and challenges reside also in linking the existing cloud computing issues [9] with other known problems, sometimes from different or neighboring fields [10], e.g. problems that tackle shopping optimization of products available over the Internet [11, 12] and the analysis of possible usage of their algorithms [13].

As communications are crucial for cloud services, a network becomes one of the most important resources. The current approach to resource optimization and scheduling is far from optimal. One of the reasons is that cloud computing is a highly dynamic online environment. Therefore, management of resources must react and adapt to a large amount of state changes in communication. Traditional approaches to resource optimization are not sufficient [14].

At this point it is worth mentioning some general information regarding task scheduling and the scheduling area (cf. [1] or [15] for a general treatment of the topic).

Let us remind a basic presentation of the scheduling problem with a set $J = \{J_1, J_2, \ldots, J_n\}$ of jobs/tasks which need to be scheduled on a set $M = \{M_1, M_2, \ldots, M_m\}$ of non-identical, parallel machines. Additional parameters that describe the problem are:

- p_{ij} – processing time of job J_i on the machine M_j,
- t_j – setup (warm-up) time for machine M_j.

The goal is to find a non-preemptive schedule that minimizes total time (including set-ups) the machines used to process the jobs.

Workflows can be modeled by directed acyclic graphs (DAG) $G_j = (V_j, E_j)$. V_j is the set tasks. $E_j = (T_u, T_v)$ is the set of directed arcs between tasks where $T_u, T_v \in V_j$ and $u \neq v$. There should be no cycles $T_u \to T_v \to T_u$. Precedence constraints are represented by arcs (T_u, T_v), where task T_u has to be completed

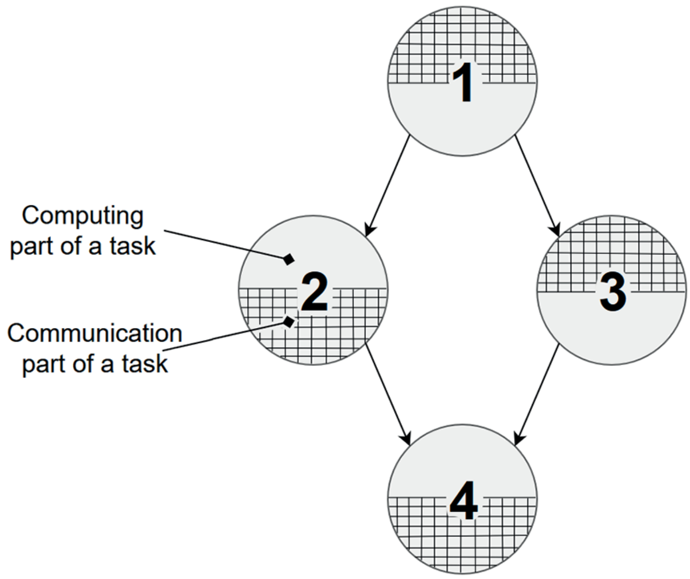


Fig. 1. CU-DAG model

before execution of T_v. It is worth noting that arc can be additionally connected with communication costs. Communication delays are not taken into account if two following tasks are executed by the same processor [16].

Task scheduling with communication costs is a recognized and extensively studied problem in the scheduling community and it was investigated for different systems and applications, e.g. complete and homogeneous network [17], network communication model with shared medium [18], or more detailed way of communication delay descriptions based on LogP model [19]. Classification of scheduling problems with communication delays was widely described by Drozdowski [2].

Recently Kliazovich et al. [7] proposed an innovative cloud computing model where communications between tasks are crucial – the Communication-Aware DAG model (CA-DAG, Fig. 3). They showed many advantages of the model over already known CU-DAG [20] (Communication-Unaware DAG, Fig. 1) and EB-DAG [18, 21] (Edges-Based DAG, Fig. 2), and discussed the positive impact of the CA-DAG model on existing scheduling algorithms.

The computational experiments showed that the usage of CA-DAG model enables more efficient scheduling in distributed systems.

Since the great part of cloud computing applications are highly dependent on communication requirements (cost, deadlines, stability, etc.) the CA-DAG is a complex model that focuses on sophisticated communication-aware issues. A new type of task is introduced: the computing tasks V_c correspond to tasks in the classical DAG model, while all communications are modeled as a new type of task: communication tasks V_{comm}. Each communication task is additionally described by parameters S and D_{comm}. D_{comm} determines deadline for transmitting S bits of information. Moreover with each node v_i there is associated cost of communication v_i^{comm}.

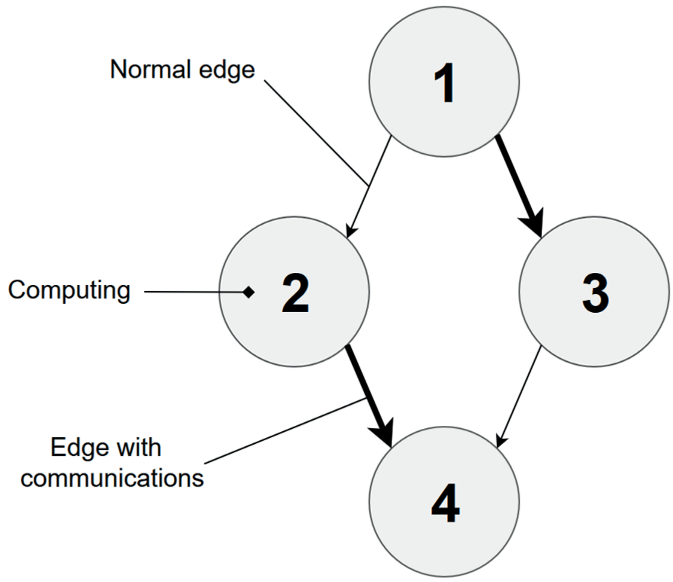


Fig. 2. EB-DAG model

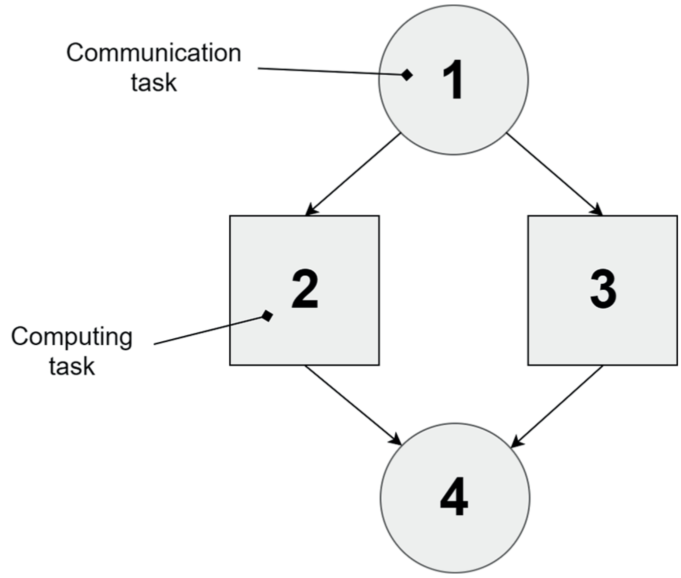


Fig. 3. CA-DAG model

The CA-DAG model for cloud computing applications is thus a novel proposition, which overcomes shortcomings of existing approaches using communication awareness. However, the CA-DAG model has a large weakness, which is the lack of adequate mathematical presentation of formal model with computational complexity analysis. The model and the complexity analysis is necessary for the proper implementation and evolution of the CA-DAG model and its derivatives.

3. Computational complexity of CA-DAG

As discussed in [7], the problem of cloud computing can be defined by a set of tasks $\mathcal{T}$ requiring an access to the communication network $\mathcal{N}$ and processor set $\mathcal{P}$.
Two subsets are distinguished:

- communication tasks – $\mathcal{T}^{\mathcal{N}}$, each requiring an access to the network, ○ – T_j^N,
- processing tasks – $\mathcal{T}^{\mathcal{P}}$, each requiring a processor to be processed, □ – T_j^P.

Every task T_j is characterized by its processing time p_j. Tasks are ordered by the precedence relation *prec* ($\prec$).

The capacity of the communication network (a number of tasks transmitted in parallel through the network) will be denoted by c, while the cardinality of the processor set by m. Thus, by using a slightly modified three-field notation of scheduling problems [22], we can denote the problem of minimizing C_{max} for the cloud computing as:

$$P(m), N(c)|prec, p_j|C_{max}. \quad (1)$$

Using some ideas from [23], we can prove strong NP-hardness of problem $P(1), N(1)|chain, p_j = 1|C_{max}$.

Theorem 1. Problem $P(1), N(1)|chain, p_j = 1|C_{max}$ is strongly NP-hard.

Proof. We will prove this theorem by using a pseudo-polynomial transformation from strongly NP-complete problem 3-PARTITION [24] to the decision version of problem $P(1)$, $N(1)|chain, p_j = 1|C_{max}$, denoted by Π_1 hereafter. The first problem is defined as follows.
3-PARTITION: Given a set $S = \{a_1, a_2, \ldots, a_{3t}\}$ of positive integers $a_1, a_2, \ldots, a_{3t}$ and integer b with $\sum_{a_j \in S} a_j = tb$, can S be partitioned into t disjoint 3-element subsets S_i such that $\sum_{a_j \in S_i} a_j = b$, for $i = 1, \ldots, t$?

In this problem we assume without loss of generality that $\frac{1}{4}b < a_j < \frac{1}{2}b$ for all $a_j \in S$. Given any instance of this problem, we construct an instance of Π_1 in the following way.

- There is a single chain X of $2tb$ tasks:

$$\begin{array}{llllll}
X = T_1^P & \prec T_2^P & \prec \ldots & \prec T_b^P & T_1^N & \prec T_2^N \\
 & & & & \prec \ldots & \prec T_b^N \prec \\
T_{b+1}^P & \prec T_{b+2}^P & \prec \ldots & \prec T_{2b}^P & \prec T_{b+1}^N & \prec T_{b+2}^N \\
 & & & & \prec \ldots & \prec T_{2b}^N \prec \\
\ldots & & & & & \\
T_{(t-1)b+1}^P & \prec T_{(t-1)b+2}^P & \prec \ldots & \prec T_{tb}^P & \prec T_{(t-1)b+1}^N & \prec T_{(t-1)b+2}^N \\
 & & & & \prec \ldots & \prec T_{tb}^N .
\end{array}$$

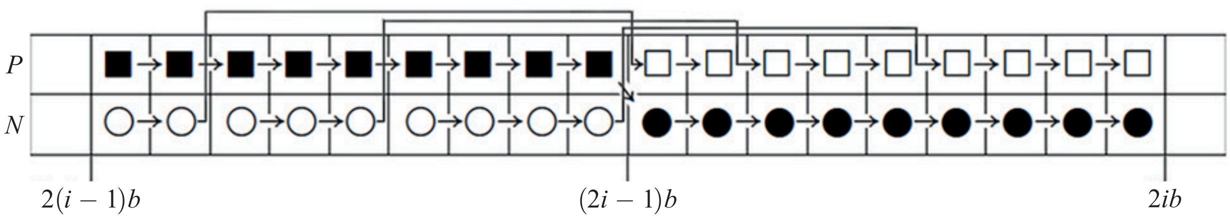
- For each $a_j \in S$, there are two chains Y_j^N and Y_j^P of a_j tasks:

$$Y_j^N = T_{j_1}^N \prec T_{j_2}^N \prec \ldots \prec T_{j_{a_j}}^N$$
$$Y_j^P = T_{j_1}^P \prec T_{j_2}^P \prec \ldots \prec T_{j_{a_j}}^P$$

where $Y_j^N \prec Y_j^P$.

We will prove that 3-PARTITION has a solution if and only if there exists a feasible schedule with value $C_{max} \le 2tb$.

Part of feasible instance for 3-PARTITION:
$b = 9,\ a_1 = 2,\ a_2 = 3,\ a_3 = 4;\ S_i = \{a_1, a_2, \ldots, a_3\}$
and the corresponding schedule for the cloud computing problem:



$X: \blacksquare - T_k^P,\ \bullet - T_k^N,\ ((i-1)b + 1 \le k \le ib);\quad Y: \bigcirc - T_{jk}^N,\ \square - T_{jk}^P,\ \{a_j \in S_i, 1 \le k \le a_j\}$

Fig. 4. An example slot of the constructed schedule for $P(1), N(1)|chain, p_j = 1|C_{max}$

Supposing that 3-PARTITION has a solution $\{S_i, S_2, \ldots, S_t\}$. A feasible schedule with value $C_{max} \le 2tb$ is then obtained as follows (cf. Fig. 4). First, chain X is assigned to processor P and after b time units its communication tasks use network N for b units. Then this pattern is repeated t times. As a result one gets free slots for the communication network in the intervals $[2(i-1)b, (2i-1)b]$, $i = 1, 2, \ldots, t$, and free slots for processor P in the intervals $[(2i-1)b, 2ib]$, $i = 1, 2, \ldots, t$. For each $i \in 1, 2, \ldots, t$, it is now possible to assign the three chains $Y_j^N(a_j \in S_i)$ to network N in the interval $[2(i-1)b, (2i-1)b]$ and to schedule chains $Y_j^P(a_j \in S_i)$ on processor P in interval $[(2i-1)b, 2ib]$. The obtained schedule is feasible with respect to processor, network and precedence constraints and its length is equal to $2tb$.

On the other hand, supposing that there exists a feasible schedule with value $C_{max} \le 2tb$. It is clear that in this schedule both the processor and then network are saturated until time $2tb$. Moreover, chains $Y_j^N(a_j \in S)$ are executed in intervals $[2(i-1)b, (2i-1)b]$, $(i = 1, 2, \ldots, t)$ and chains $Y_j^P(a_j \in S)$ in the remaining intervals. Let S_i be the index set of chains Y_j^N completed in interval $[2(i-1)b, (2i-1)b]$, for $i = 1, 2, \ldots, t$. Consider set S_i. It is impossible that $\sum_{a_j \in S_i} a_j > b$, due to the definition of S_i; the case $\sum_{a_j \in S_i} a_j < b$ cannot occur either, since this would lead to processor idle time in $[b, 2b]$. It follows that $\sum_{a_j \in S_i} a_j = b$, and our assumption about the size of $a_j(a_j \in S)$ implies that $|S_i| = 3$. This argument is easily extended to an inductive proof that $S_i, S_2, \ldots, S_t$ constitutes a solution to 3-PARTITION. □

4. Conclusions

Cloud computing and its applications are gaining enormous popularity in recent years, what resulted in building a vast market of services. Communication-aware directed acyclic graph is a novel distributed computation model inspired by the characteristics of cloud computing.

In the paper we widely discussed the CA-DAG computational complexity and prove strong NP-hardness of the problem. This step enables future implementation and evolution of the communication-aware DAG models including optimization aspects [25, 26] driven from known problems that are partially similar.

A great number of applications is highly dependent on bandwidth and a low latency communication (e.g. cloud gaming, a video conferencing, a collaborative editing, an online office, a remote desktop, and so on). The CA-DAG model overcomes shortcomings of existing communication aware approaches. With current adequate mathematical modeling and computational complexity analysis future paths of research are open wider.

References

[1] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, Handbook on Scheduling – From Theory to Applications, Springer Verlag, Berlin, New York (2007).

[2] M. Drozdowski, Scheduling for Parallel Processing, Springer London (2009).

[3] N. Christofides, Graph Theory: An Algorithmic Approach (Computer Science and Applied Mathematics), Academic Press, Inc., Orlando, FL, USA (1975).

[4] P. Mell and T. Grance, "The NIST definition of cloud computing", *National Institute of Standards and Technology* 53 (6), 50 (2009).

[5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing", *Communications of the ACM* 53 (4), 50–58 (2010).

[6] L. Columbus, "Roundup Of Cloud Computing Forecasts And Market Estimates, 2016", www.forbes.com/sites/louiscolumbus/2016/03/13/roundupof-cloud-computing-forecasts-and-market-estimates-2016, accessed: 2016–06–16.

[7] D. Kliazovich, J.E. Pecero, A. Tchernykh, P. Bouvry, S.U. Khan, and A. Y. Zomaya, "CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing", *Journal of Grid Computing* 14 (1), 23–39 (2016).

[8] M. Guzek, P. Bouvry, and E.-G. Talbi, "A Survey of Evolutionary Computation for Resource Management of Processing in Cloud Computing [Review Article]", *IEEE Computational Intelligence Magazine* 10 (2), 53–67 (2015).

[9] M. Guzek, A. Gniewek, P. Bouvry, J. Musial, and J. Blazewicz, "Cloud Brokering: Current Practices and Upcoming Challenges", *IEEE Cloud Computing* 2 (2), 40–47 (2015).

[10] J. Blazewicz, N. Cheriere, P.-F. Dutot, J. Musial, and D. Trystram, "Novel dual discounting functions for the Internet shopping optimization problem: new algorithms", *Journal of Scheduling* 19 (3), 245–255 (2016).

[11] J. Blazewicz, P. Bouvry, M.Y. Kovalyov, and J. Musial, "Internet shopping with price sensitive discounts", 4OR – *A Quarterly Journal of Operations Research* 12 (1), 35–48 (2014).

[12] J. Blazewicz, P. Bouvry, M.Y. Kovalyov, and J. Musial, "Erratum to: Internet shopping with price-sensitive discounts", *4ORA Quarterly Journal of Operations Research* 12 (4), 403–406 (2014).

[13] J. Blazewicz and J. Musial, "E-Commerce Evaluation – Multi-Item Internet Shopping. Optimization and Heuristic Algorithms", in Operations Research Proceedings 2010 (edited by B. Hu, K. Morasch, S. Pickl, and M. Siegle), Springer-Verlag, Berlin, 149–154 (2011).

[14] M. Zotkiewicz, M. Guzek, D. Kliazovich, and P. Bouvry, "Minimum Dependencies Energy-Efficient Scheduling in Data Centers", *IEEE Transactions on Parallel and Distributed Systems* PP (99) (2016).

[15] J. Leung, L. Kelly, and J. H. Anderson, Handbook of Scheduling: Algorithms, Models, and Performance Analysis, CRC Press, Inc., Boca Raton, FL, USA (2004).

[16] H. El-Rewini and T.G. Lewis, "Scheduling Parallel Program Tasks Onto Arbitrary Target Machines", *J. Parallel Distrib. Comput*. 9 (2), 138–153 (1990).

[17] C.H. Papadimitriou and M. Yannakakis, "Towards an Architecture-independent Analysis of Parallel Algorithms", *SIAM J. Comput*. 19 (2), 322–328 (1990).

[18] O. Sinnen and L.A. Sousa, "Communication contention in task scheduling", *IEEE Transactions on Parallel and Distributed Systems* 16 (6), 503–515 (2005).

[19] D.E. Culler, R.M. Karp, D. Patterson, A. Sahay, E.E. Santos, K.E. Schauser, R. Subramonian, and T. von Eicken, "LogP: A Practical Model of Parallel Computation", *Commun. ACM* 39 (11), 78–85 (1996).

[20] G. Srikanth, A. Shanthi, V. Maheswari, and A. Siromoney, "A Survey on Real Time Task Scheduling", *Eur. J. Sci. Res*. 69 (1), 33–41 (2012).

[21] P. Choudhury, P.P. Chakrabarti, and R. Kumar, "Online Scheduling of Dynamic Task Graphs with Communication and Contention for Multiprocessors", *IEEE Transactions on Parallel and Distributed Systems* 23 (1), 126–133 (2012).

[22] R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey", in Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver (edited by E. J.P.L. Hammer and B. Korte), volume 5 of *Annals of Discrete Mathematics*, Elsevier, 287–326 (1979).

[23] J. Blazewicz, J. Lenstra, and A. Rinnooy Kan, "Scheduling subject to resource constraints: classification and complexity", *Discrete Applied Mathematics* 5 (1), 11 – 24 (1983).

[24] M. Garey and D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, New York, Freeman (1979).

[25] M.C. Lopez-Loces, J. Musial, J. E. Pecero, H.J. Fraire-Huacuja, J. Blazewicz, and P. Bouvry, "Exact and heuristic approaches to solve the Internet shopping optimization problem with delivery costs", *International Journal of Applied Mathematics and Computer Science* 26 (2), 391–406 (2016).

[26] J. Musial, J.E. Pecero, M.C. Lopez-Loces, H.J. Fraire-Huacuja, P. Bouvry, and J. Blazewicz, "Algorithms solving the Internet shopping optimization problem with price discounts", *Bull. Pol. Ac.: Tech.* 64 (3), 505–516 (2016).