Available online at www.sciencedirect.com
SCIENCE DIRECT®
Decision Support Systems 40 (2005) 329–337
Decision Support Systems
www.elsevier.com/locate/dsw

# The e-OCEA project: towards an Internet decision system for scheduling problems

V. T'kindt*, J.-C. Billaut, J.-L. Bouquard, C. Lenté, P. Martineau, E. Néron, C. Proust, C. Tacquard

*Laboratory of Computer Sciences/University of Tours, Ecole Polytechnique de l'Université de Tours, 64 avenue Jean Portalis, 37200 Tours, France*

Received 4 February 2003; received in revised form 23 March 2004; accepted 3 April 2004
Available online 18 May 2004

**Abstract**

This paper deals with an Internet decision support system for scheduling problems. This system, called e-OCEA, is being developed at the Laboratory of Computer Sciences of the University of Tours. It provides a user with tools to help create an effective algorithm to solve a scheduling problem. From the modelisation of the problem to the visualization of a computed schedule, the e-OCEA system offers software that can be used either by operations researchers or industrial engineers. In this paper, we present the current state of this system and provide future directions.


*Keywords:* Scheduling; Decision system; Internet

## 1. Introduction

Scheduling problems are of interest from both practical and theoretical point of views. A key point for their resolution is the information system, a firm's backbone, that provides all the data to the scheduling algorithms. Such a system facilitates the connectivity between various parts of the firm, such as manufacturing, finance, and human resources, since it allows managing most of its information flows. The scheduling level may interact with mid-term and long-term plannings to get preventive maintenance schedules, release dates of jobs, etc. Hence, getting all the necessary data from the information system, the scheduling system is not only able to compute a solution but also offers graphical tools to monitor the implementation of this solution. Scheduling systems are decision systems that can be decomposed into three modules [12]:

(i) The *database* module which is on the borderline with the information system since it provides the user with information about jobs, resources, and the state of the workshop (idle times of the resources, stock levels between resources, laps when a resource is undergoing setups, etc.).
(ii) After all the necessary data have been chosen, the *schedule generation* module is used to compute a schedule. This module generally uses priority

* Corresponding author. Tel.: +33-2-47-36-14-14; fax: +33-2-47-36-14-22.
*E-mail address:* tkindt@univ-tours.fr (V. T'kindt).

0167-9236/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2004.04.001

rules and improvement procedures based on neighborhood searches. The hardness of scheduling problems justifies the use of such heuristic resolution. Despite the fact that the computed solutions may not be of a high quality for the considered criteria, the interest of these methods lies in their efficiency. Therefore, we allow a tradeoff between the duration of the search and the quality of the solution.

(iii) The computed schedule has to be implemented and the user must accurately monitor the state of the workshop. Hence, the *user interface* module has to provide graphical features that allow the user to interact with the workshop. For instance, after a resource breakdown has occurred, he may choose to change some assignments of operations to overcome it. Besides, it might sometimes be interesting to reschedule remaining operations as soon as such an unexpected event occurs. This module is linked to the workshop-floor management system and eventually to the scheduling module.

The e-OCEA project—which stands for Objects for Comparing and Elaborating Algorithms—aims at developing a decision system for solving scheduling problems and acting as a scheduling system. As any scheduling system, the e-OCEA platform offers the tools to generate and visualize schedules. Moreover, it offers the tools that will help the user solve his (or her) problem. The e-OCEA platform is in fact an Internet platform, whose web address is www.ocea.li.univ-tours.fr.

Scheduling systems have been extensively studied and developed by both universities and companies (see Ref. [11] for instance). Some examples of such systems are *Ariane* [15] for scheduling in glass bottles production, *Roman* [7] for project scheduling in the nuclear power industry, *Ordo* [2], *Leikin* [1], and *Parsifal* [9] for workshop scheduling. As these systems are more or less general, meaning that they are able to tackle several workshop configurations, the algorithms used may not be effective on some configurations. Indeed, the purpose being to provide a fast algorithm, both priority rules and improvement procedures reach this goal. Yet, other fast algorithms could perform well since the design of effective algorithms results from an accurate knowledge of the tackled problem and of the available methods to solve it. It reinforces the willingness to design a system mixing the features of a scheduling system and the possibility of building an effective and efficient schedule generation module dedicated to the scheduling problem under consideration. Hence, this system would provide both the modules to develop a *scheduling generation* module (from the study of the literature to the realization of benchmarks) and the tools in a scheduling system such as the user interface module. Parts of such a development system have already been developed. Among others, the *LOCHO* software [13] allows to conduct benchmarks under the MS DOS environment, by using the advantages of existing software for data generation, for result visualization, etc. The *OCEHO* software [14] is based on the same ideas as the *LOCHO* software since it interacts with commercial software to provide a graphical interface for the testing of algorithms under the MS Windows environment. Also notice the existence of the LiSA project [4] (lisa.math.uni-magdeburg.de), which has been initiated in parallel of the e-OCEA project. The LiSA project aims at developing an integrated software, under the MS Windows operating system, for scheduling problems. Basically, it enables the user to define his scheduling problem and to run some heuristics or exact algorithms to solve it. The resulting schedules are shown to the user. All these functionalities are accessible via a single graphical interface. At last, we can quote the Ilog Scheduler package (www.ilog.com), which enables the user to develop scheduling algorithms based on constraint programming.

There also exist general software, which can be used to solve combinatorial optimization problems. Among the most classic ones that can be applied to scheduling, we can find the *BOB* library [8], which helps in implementing sequential or parallel branch-and-bound algorithms.

All along this paper, the use of the e-OCEA decision system is illustrated using a scheduling problem with two resources where each job needs to be processed on the first resource and next on the second one. The job processing orders on the two resources are the same. The aim is to compute a schedule with a minimal value of the maximum job completion time. This problem is known as a two-

machine flowshop scheduling problem with makespan minimization.

The following is organized as thus. Section 2 contains the foundations of the project such as the modular architecture, the information exchanged, etc. Section 3 presents the e-DePI module that allows the user to model his problem and get more information on it. Section 4 is devoted to the e-LEA module that helps build a scheduling algorithm (schedule generation module). Section 5 deals with the e-LCA module used to conduct computational experiments on scheduling algorithms and to retain the most performing one. Section 6 presents the e-Gantt module that is a user interface module allowing to visualize and modify a schedule. Section 7 is devoted to a presentation of the status of the project and to drawing future directions.

## 2. The main concepts

This section presents an overview of the foundations of the e-OCEA tool. Roughly speaking, it is decomposed into two main parts. The first part is related to the database management and enables the user to manipulate algorithms, references, and data sets. The database is managed through a server MySQL (www.mysql.com). The second part contains several modules, each one offering to the user functionalities to identify, solve, or visualize scheduling problems. These two parts are closely connected as shown in Section 2.1. The architecture of the e-OCEA tool as well as the way modules communicate are explained in Section 2.2.

### 2.1. From the problem modeling to the visualization

Scheduling problems have been extensively studied since the mid 50's. Numerous scheduling models dealt with in the literature, have emphasized on the utility of a scheduling problems classification (see Refs. [3,5,10]). The same conclusion applies to resolution methods. Browsing the e-OCEA platform is user-friendly thanks to problems and methods classifications. For instance, the user's access to data is simplified as he only has to select from the graphical interface the entry in the scheduling problem classification matching his problem. Several modules compose the e-OCEA software and each of them allows to manage a part of the decision process related to the resolution of a scheduling problem.

Fig. 1 shows possible interactions between the modules if the user wants to use them all. From a practical point of view, it is clear that these modules can be used in an interactive, multiple passes, approach: At any stage of the process described in Fig. 1, the user can decide to start it again. This can be the case if additional information or idea have come to him during the previous iteration of the process.

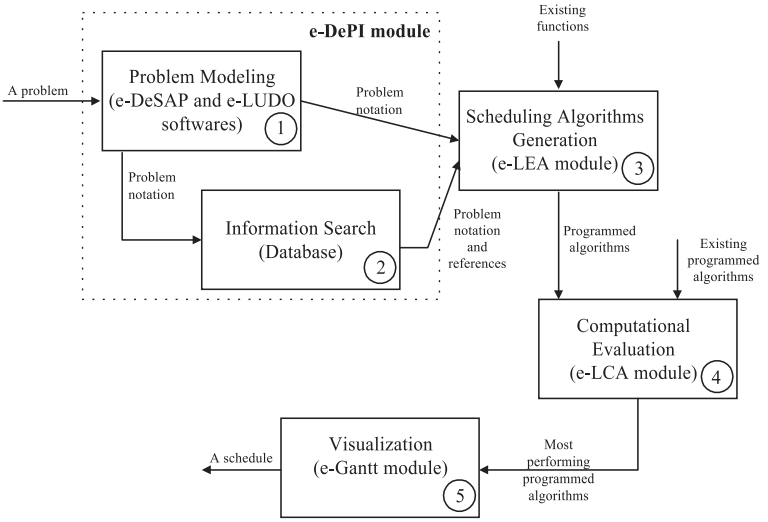


Fig. 1. Functional diagram of e-OCEA modules.

At the first step, the user knows his problem and therefore is able to model it, using the e-DeSAP software (see Section 3). From the resulting model, we can build a notation of the underlying scheduling problem; and using this notation, we can search for existing references of the literature on scheduling problems in the database. Knowledge on the scheduling problem under consideration may help in designing scheduling algorithms and generating corresponding programs, for instance, thanks to the e-LEA module (see Section 4). To evaluate the effectiveness of the available scheduling programs the e-LCA module can be used (see Section 5). It enables conducting computational experiments and comparing the obtained results. At last, once one or more scheduling programs have been selected, the user is able to see the schedules computed on different data sets, realistic or not (thanks to the e-Gantt module). This module also allows to modify data or schedules.

### *2.2. Architecture and communication between modules*

The architecture of the e-OCEA platform has been conceived as a logical bus in which modules can be plugged (Fig. 2).

The master part of this bus contains a database of existing scheduling algorithms (classified according to their related problem and method), of data and schedules sets already generated (classified according to their related problem), and of problems notations (classified according to the related problem). This master part is managed via a graphical interface. It allows the user to manage the classifications of problems and methods, the modules plugged in the bus, to edit and create problems as well as schedules and to import/export and run algorithms.

All the information exchanged between modules using the logical bus are normalized. This mainly implies that file formats, based on the XML language, have been introduced to store data sets and schedules. One of the feature of the e-OCEA environment is to freely provide the necessary code to develop either a compatible e-OCEA module or a compatible e-OCEA scheduling program. Thus, to develop a module that can be used from the e-OCEA platform, we can use existing code to manage data without being concerned with the corresponding e-OCEA file formats. This is also valid when developing a schedule algorithm since the programmer is interested in reading data sets and writing schedules.

A major advantage to the e-OCEA platform is that the user can easily navigate in the database of data sets, schedules, and algorithms. Besides, the normalization of the information exchanged between e-OCEA modules leads to a unified interface to tackle and solve the user's scheduling problems. The architecture of the environment is such that it is an open environment.

## 3. e-DePI: preparing the resolution

One of the major steps before beginning the design of a scheduling algorithm consists in identifying the scheduling problem and searching for information on existing results. To do so, a graphical tool joined with an up-to-date database can be very helpful. e-DePI is a tools-package containing two software, namely e-

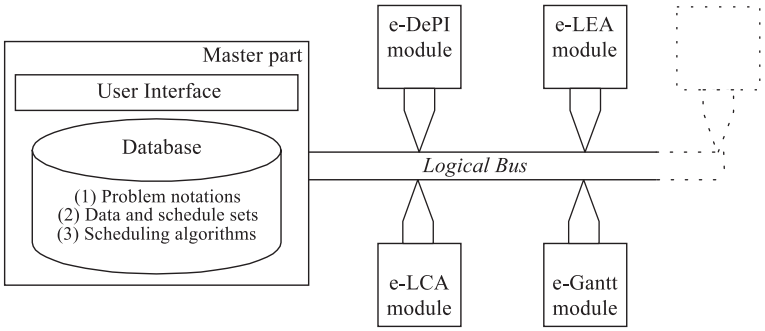


Fig. 2. The architecture of the e-OCEA software.

DeSAP [16] and e-LUDO [6], for describing the workshop configuration, the routing of the parts, and for deriving the scheduling problem. Once the corresponding scheduling problem notation has been identified, queries can be sent to the database to eventually get information on this problem and on existing algorithms. The user interface of the e-DeSAP module showing the two-machine flowshop problem, introduced in Section 1, is presented in Fig. 3.

The e-DeSAP module makes it possible to draw a Flexible Manufacturing System by specifying the resources, the available tools, and the routes available to the automated guided vehicles. Such information are not sufficient to define the scheduling problem. Hence, complementary information such as jobs routings, the use of different job release dates, precedence constraints, etc., can be specified by the user. Once these information have been given, we can get a translation of the graphical workshop structure into a full notation (*Analyse function*). Another possibility is to automatically identify the underlying scheduling problem (*Detection function*, see Fig. 4). Later on, the

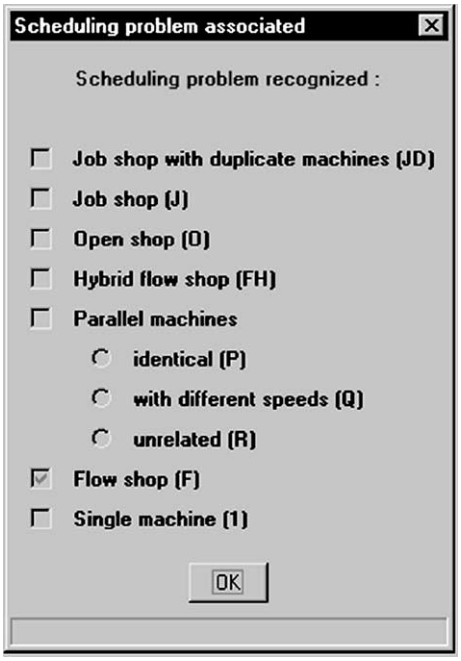


Fig. 4. Retrieving of the scheduling problem.

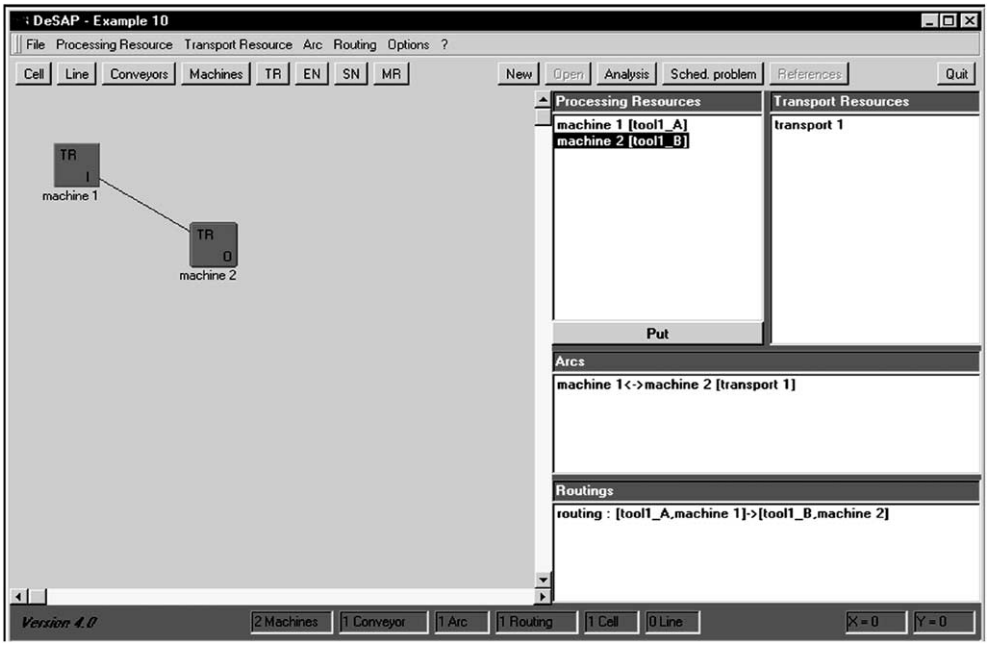


Fig. 3. The e-DeSAP module.

user is able to get information on existing algorithms related to his problem by exploring the database.

The e-LUDO software offers other possibilities. A macro-language can be used to describe the scheduling problem. The result can be used to automatically generate an exact scheduling algorithm based on constraint programming and Ilog Solver. Links between e-DeSAP, e-LUDO, and the database remain to be established.

## 4. e-LEA: generating scheduling programs

One of the most challenging issues is the automatic generation of a scheduling module dedicated to the problem under consideration. Despite the fact that such an intelligent generation software may not exist, there exists a means to help the user create his scheduling module. The key point is that when developing a resolution algorithm (such as a branch-and-bound algorithm, a greedy algorithm, etc.), most of the program can be taken from already implemented algorithms. Notice that Software Engineering emphasizes such an approach to software development since it facilitates the reuse of already tested objects. The e-LEA module aims at offering a *semi-automatic mode*, which allows the user to only specify the part of his algorithm that is problem dependent, in an existing programming language.

For instance, while creating a branch-and-bound algorithm for the flowshop scheduling problem involving the total completion time criterion (see, for instance, Ref. [3] on this problem), we only need to describe the branching process, the bounding process (upper and lower bounding), the search strategy, and the dominance conditions. The whole code related to the management of the search tree does not need to be programmed. It implies that the e-LEA module must have data structures able to tackle a number of scheduling problems as high as possible. Maybe, the tradeoff is in obtaining an algorithm less efficient than a specifically developed one. Works on a generic branch-and-bound algorithm are in progress [17].

Most of the development of the e-LEA module is currently under scrutiny for feasibility. However, the generation of branch-and-bound algorithms and recovering beam search heuristics is operational. Several prototypes, implementing other kind of methods, have already been developed on different operating systems. The generation of greedy heuristics based on simple scheduling rules has already been experimented (under the MS DOS environment). In this prototype, the user can associate a scheduling rule to each resource (such as the *Shortest Processing Time First* rule or the *Earliest Due Date First* rule), which describes how to schedule the jobs available for processing on this resource. Such a prototype could be adapted to handle problems with stages, such as hybrid flowshop problems or general jobshop problems. The user can also select the criteria that must be computed after the schedule is computed. Such a reasoning can also be applied to other methods, be they exact or heuristic, such as genetic algorithms, tabu search, or mathematical programming.

## 5. e-LCA: obtaining the best scheduling programs

After several scheduling algorithms have been programmed, either using the e-LEA module or not, the purpose of a user is to evaluate their effectiveness and efficiency in order to keep the best ones. The aim of the e-LCA module is to help realize benchmarks. One of the highlights in this software is to provide a simple interface to describe the variables to be generated in the benchmarks as well as the computational process. This consists in defining the variables to randomly generate and how to generate them. Notice that instead of using random data sets we can use pre-existing data sets contained in the database. At the end the list of scheduling problems to compare must be specified. Next, the experiments are automatically conducted by the e-LCA module. Fig. 5 shows the definition of computational experiments.

But at this point, the main work remains to be done, namely to deduce the "best" scheduling programs from the computational experiments. Notice that the quality of such a program is usually measured in terms of resolution time and/or objective function value. The user can freely decide. Basically, the e-LCA module is able to compute statistics such as average, maximum, and minimum deviations of scheduling programs from either the best one or a reference program. This classic way to proceed enables the user to decide of the "best" algorithms over the set of considered instances.

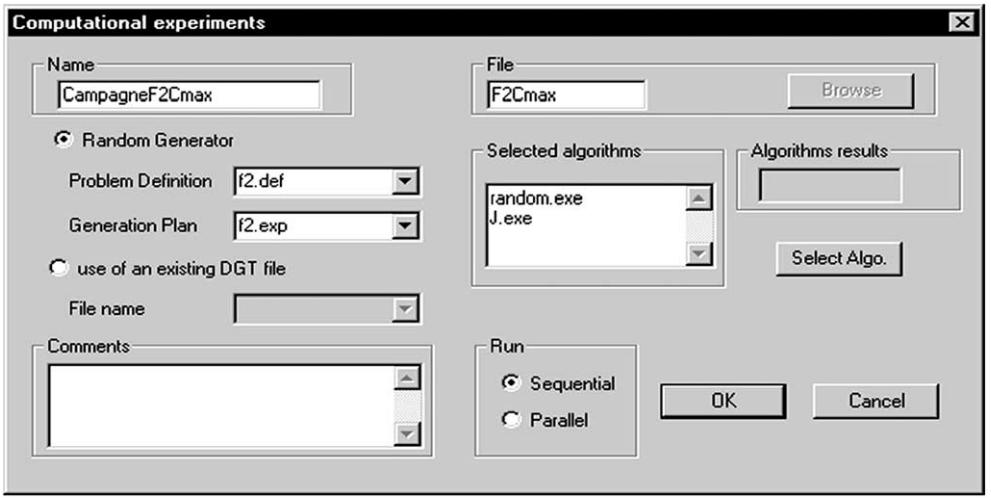


Fig. 5. Specifying the benchmark.

Besides, we can use a more sophisticated tool, the decision aid module which provides two main functions. The first one calculates a simple statistic sheet that is independent of the scheduling problem, and which can be printed using graphical indicators. The latter show average and maximum information on the waiting time of jobs, the idle time of resources, etc. Despite the fact that it provides absolute information on the computed schedules, it is not sufficient to decide on the best scheduling programs. Therefore, a data analysis must be conducted using the second function of the decision aid module. It performs a classification of all the instances to try and gather those for which the scheduling programs compute similar criteria vectors. For instance, if we consider the minimization of both the maximum completion time and the total completion time of jobs (by five different scheduling programs), we can define a vector composed of the 10 values of the criteria for each data set. The use of a classification algorithm allows to obtain classes of similar data sets and reduces the number of results to consider. For each class, the criteria vector associated to a scheduling program is obtained by computing the average value among the aggregated data sets for each criterion. The problem of deciding what the best scheduling programs are, is a multicriteria problem and the parametric approach is used to compute one strict Pareto optimum (see Ref. [18] for more details on this approach). More precisely, to decide of the best algorithms, we minimize a convex combination of the criteria with nonzero weights subject to bound constraints on the criteria. The user feeds in the value of the weights and of the bounds, thus the best algorithms can be deduced for each class. This way to proceed enables the user to choose which algorithms perform best for each class of instances. This is valid for both exact and heuristic algorithms, where efficiency is usually measured in terms of resolution time and quality, respectively.

## 6. e-Gantt: editing the solution

From now on, the user has the best scheduling programs, among those tested, for his problem. However, when a practical problem has to be solved, he not only wants to get a good schedule but also visualize it. Besides, he may want to modify it in order to take account of non modeled constraints (job families, tools changing, etc.).

The e-Gantt module allows the user to edit solutions of a given scheduling algorithm. The corresponding Gantt chart is displayed either by showing the scheduling of operations for each resource or by showing the job-by-job schedule (Fig. 6). Using the graphical interface, the user is able to modify the schedule in several ways. First, he can either change the assignment and the start date of an operation, or permute two

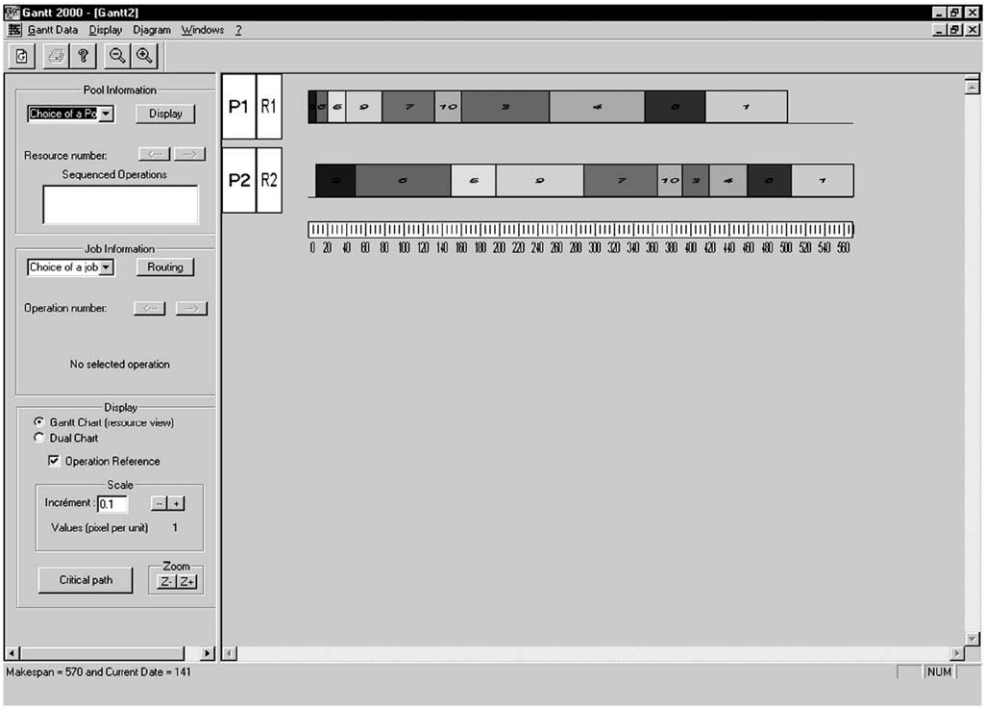


Fig. 6. A Gantt chart.

operations on the same resource. He can also modify the processing time of an operation. These possibilities are of interest for instance to evaluate the robustness of the solution.

## 7. Future directions

The e-OCEA platform is a free and open platform aiming at becoming an Internet decision system acting as a scheduling system for the design of dedicated effective scheduling algorithms. Moreover, one of its highlights is to facilitate information sharing among the scheduling community via the scheduling problems database and the addition of scheduling programs to the platform. Thus, it allows anyone to retrieve some scheduling programs, or data sets to conduct computational experiments again. In 1996, we initiated this project under the MS Windows operating system, by the way leading to a quite operational prototype. Later on, in 2000, this was abandoned in favour of an Internet platform, the ideal medium for a powerful sharing of information system.

So far, the development of the e-OCEA project has been mainly devoted to the e-DePI, e-LCA, and e-Gantt software. All the database management part is also operational, so that users can freely connect to the web site and add or consult references, data sets, and algorithms. Besides, we make available all the object codes required when developing either a scheduling program or an e-OCEA module. Great research and engineering efforts are being done in order to develop the e-LEA and e-LCA modules. We also always seek to enable the identification and resolution of more and more complex scheduling problems, such as industrial problems. We also believe that it might be interesting to study how this platform can use GRID computing.

## Acknowledgements

The authors would like to very warmly thank Fabrice Tercinet for his essential help in managing the e-OCEA project. He also contributed to make this project a success.

## References


[1] N. Asadathorn, *Scheduling of Assembly Type of Manufacturing Systems: Algorithms and Systems Developments*. PhD thesis, Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology, Newark (USA), 1997.

[2] J.-C. Billaut, F. Roubellat, A new method for workshop real time scheduling, International Journal of Production Research 34 (6) (1996) 1555–1579.

[3] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, J. Weglarz, Scheduling Computer and Manufacturing Processes, Springer Verlag, Heidelberg, 1996.

[4] H. Braesel, N. Shakhlevich, LiSA—Fit for Cooperative Development, Sixth Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP'03), Aussois (France), 2003, pp. 107–108.

[5] P. Brucker, Scheduling Algorithms, Springer Verlag, Heidelberg, 1998.

[6] J. Carneiro, P. Lourenco, J.-L. Bouquard, Conception d'un logiciel d'aide à la planification des tâches, à partir d'une spécification de FMS, en utilisant la programmation par contrainte, End Studies Report, E3I, University of Tours, France, 1998, in French.

[7] C.P. Gomes, D.R. Smith, S. Westfold, Synthesis for schedulers for planned shutdowns of power plants, 11-th Knowledge Based Software Engineering Conference, Los Alamitos, CA (USA), IEEE Computer Society Press, 1996, pp. 12–20.

[8] B. Le Cun, C. Roucairol and PNN Team, Bob: A Unified Platform for Implementing Branch-and-Bound Like Algorithms, Research report 95/16, PRiSM, University of Versailles (France), 1995.

[9] T.E. Morton, D.W. Pentico, Heuristic Scheduling Systems, Wiley-Interscience, 1993.

[10] M. Pinedo, Scheduling: Theory, Algorithms and Systems, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[11] M. Pinedo, X. Chao, Operations Scheduling with Application in Manufacturing and Services, Irwin/McGraw-Hill, Boston, 1999.

[12] M. Pinedo, P.-C. Yen, On the design and development of object-oriented scheduling systems, Annals of Operation Research 70 (1997) 359–378.

[13] C. Proust, A. Ferreira, J. Bijaoui, Le paradoxe de l'interprogrammation: la diversité des logiciels au service d'une intégration efficace, Congrés de Génie Industriel, Tours (France), 1991, pp. 125–132, in French.

[14] F. Riane, C. De Brauwer, A. Artiba, OCEHO: un outil de comparaison et d'evaluation d'heuristiques de résolution de problèmes d'Optimisation, European Journal of Automation (JESA) 32 (7–8) (1998) 853–874 (in French).

[15] P. Richard, C. Proust, Maximizing benefits in short-term planning in bottle-glass industry, International Journal of Production Economics 64 (1–3) (2000) 11–19.

[16] C. Tacquard, P. Martineau, Automatic notation of the physical structure of a flexible manufacturing system, International Journal of Production Economics 74 (1–3) (2001) 279–292.

[17] F. Tercinet, E. Néron, Logiciel pour la génération de procédures par séparation et evaluation, 3rd FRANCORO Conference, Québec (Canada), 2001, pp. 99–100, in French.

[18] V. T'kindt, J.-C. Billaut, Multicriteria Scheduling: Theory, Models and Algorithms, Springer Verlag, Heidelberg, 2002.