Multi objective support system for the decisions to plan scheduling of operations

Sistema multiobjetivo de soporte a decisiones para planificación de programación de operaciones

Iván Saavedra Antolínez[1]
Carlos D. Paternina Arboleda[2]
Elías D. Niño Ruiz[3]
Libardo S. Gómez Vizcaíno[4]

Abstract

This research shows the design and construction of a multi objective optimization support system for decision making, DSS-SCHEDULER, in order to effectively perform naval shipbuilding operations, by managing to have a rational usage, under a limited capacity, of the installed capacity. The system creates the operational plan for the scheduling of operations as well as the allocation of all the resources related to the processes carried out at the plant, and, taking into account all the restrictions this type of service entails, generate a set of optimal solutions in order to comply with the objective functions as planned, such as costs and the promise for delivery dates to support the decision making process in scheduling Cotecmar's industrial operations.

Key words: DSS decision support system, multi-objective optimization, operation scheduling, operational planning, and evolutionary algorithms.

Resumen

La presente investigación muestra el diseño y construcción de un sistema de optimización Multiobjetivo de soporte a toma de decisiones, DSS-SCHEDULER, para realizar de manera efectiva la actividad de programación de operaciones en planta de astilleros navales, logrando una utilización racional, bajo capacidad finita, de la capacidad instalada, el cual realiza la creación de la planificación operativa de programación de operaciones y la asignación de todos los recursos asociados a los procesos que se realizan en la planta, y teniendo en cuenta todas las restricciones que conlleva estos servicios, generar un conjunto de soluciones óptimas cumpliendo con las funciones objetivo propuestas, como costos y promesa de fechas de entrega para apoyar la toma de decisiones de la programación de las operaciones industriales de Cotecmar.

Palabras claves: DSS Sistema de apoyo para la toma de decisiones, Optimización multi-objetivo, programación de operaciones, planificación operativa, Algoritmos evolutivos.

Introduction

Planning the operations at the plants of the companies nowadays requires to be carried out through efficient and strong optimization techniques, which not only affect the development as well as the behavior of the company itself, but also, the tactical and strategic decisions this type of solutions entail such as cost estimation, policies, levels of service, priorities, machinery, among others.

Besides, the competition level for the company is increased since by having systems as the one presented by this research, a high technological development is obtained at the Information System (SI) because it is about systems that are at the top of the organizational pyramid due to the type of solutions the system is able to generate.

The theory of complexity shows us that for problems such as the planning and controlling of operation scheduling with a large number of restrictions and various objective functions, there are no exact methods through which it is possible to obtain feasible – optimal solutions. The use of heuristic techniques (EDD, LPT, SPT, etc.) allows us to find sub optimal methods in reasonable calculating times. Notwithstanding, it is the meta heuristic techniques the ones that take the biggest portion of the literature devoted to solving strong combinatory problems as well as the ones that have suggested the Best solutions.

Included here and among the so called meta heuristic types, are all the procedures that, within an iterative process, lead an ordinary heuristic by intelligently combining certain concepts that have been taken in analogy with nature, and explore the solution space by utilizing the learning strategies to structure the information to efficiently find solutions that are very close to the global optimal of a problem as an objective.

It is due to the reasons mentioned above, that it was necessary to implement an evolutionary meta-heuristic that would help, from the seed generating heuristic, to obtain optimal multi- objective plans for the solution of the problem.

Description of the problem and solution

Operation programming in industrial companies is a fairly complex job, due to the great number of variables, restrictions and objectives the entrepreneurs, operators and customers of the company have. The following items show the specific problem to be solved as well as the solution that has been suggested in this research.

Description of the Problem

Within the naval production environments, the ways in which we are able to make a product, be it a repair, empowerment or manufacture, the processes involved face a great deal of restrictions, such as, a product may have more than one production phase, it requires different processing times, etc. The way to assign: materials, human resources, dates, amounts to be produced by each one of the material resources or machines in order to comply with a delivery plan, has become a multiplicity of options from which choosing the best option is a difficult task. The result of this process is better known as the making of a sequencing plan or programming the operations. In various productive systems, the operation programming process takes on great significance, due to the incidence it has on both, the competitiveness and productivity of the company.

Operation programming is the process involving the allocation of resources that are limited to the tasks within a certain deadline, in order to optimize one or more objectives. These resources may very well be machines or people. Some examples of the said tasks include machine operation, moving, transportation, loading, unloading, etc.

Using heuristics and meta-heuristics in the operation programming process, helps deliver programs that are very close to the optimal objective. These mathematical models, used to schedule operations, depend on the way the company has been organized, as far as the physical structure and product flow the company has is concerned.

This company organization goes from the easiest one form a simple machine where just one type of product enters, to the problems of the type of a Flexible Job Shop Flexible that have a high level of complexity to be solved up to the programming of projects, where it is possible to find assembling activities and both machines and workers can be found working in one common activity.

Several different references can be found to use productive configuration. If the business environment were to be studied, it would be easy to find a variety of production systems at manufacturing and service rendering companies.

All tasks may have a deadline and some of the tasks may have priority over others. Among the operation programming objectives it is relevant to take into account the one that aims to reducing to the minimum level possible the time it takes to finalize all tasks (MAKESPAN) by maximizing the number of finalized tasks in a given period of time and reducing the inventory in process.

An Approach to the Solution

In order to find a solution to the problem that has been stated it could be approached from three specific fields, among which we can find:

Good information management: This solution is suggested due to the deficiencies encountered within the productive system when the time comes to store all production related information. Most of the companies only handle the information related to large amounts in long periods of time (days or weeks), but they do not have the information so it can be known in an accurate fashion what time the good enters the company, when it is being processed and an order is issued for its further study.

Good Planning: Companies do not have a system that helps them to plan production, that is, they do not have systems that by using logic works generate the best way to produce an item, which makes organizing the production a more difficult task upon receiving the orders. For this specific problem

it would be a good idea if models were generated to help in the decision making process at the time of organizing the production schedule.

Adequate Technology: In order to give a solution to the production scheduling problems it would be useful to develop a systematic model (software) that helps in the operation programming activities at the company.

A combination of the three solutions would be the most appropriate option to solve the problem that has been stated, because with the aid of a tool that facilitates the task scheduling process, information management and besides uses the correct logic to program, a great contribution would be made in decreasing delayed deliveries, eliminating inventory and minimizing costs.

Decision Support System (DSS)

Description of DSS

Architecture of DSS
The decision support system (DSS) is supported on a two- level- architecture Client – Server, this supplies for the general sustainability of the system, among other aspects. The general architecture of the system can be seen in Fig. 1.

The Stand –Alone application which supports the system GUI can be found at the client's side. On the other side it is possible to find the Data Base server, which allows for persistence and the search for information.

Structural Components for the Programming Algorithm
The system information is supported on data structures (see Fig. 2) which facilitate the execution of the programming. These structures are linked and match their behavior to that of the portrayed reality, defining a programming algorithm that is as real and natural as possible.
As a whole, the programming model being proposed upon loading the information on these
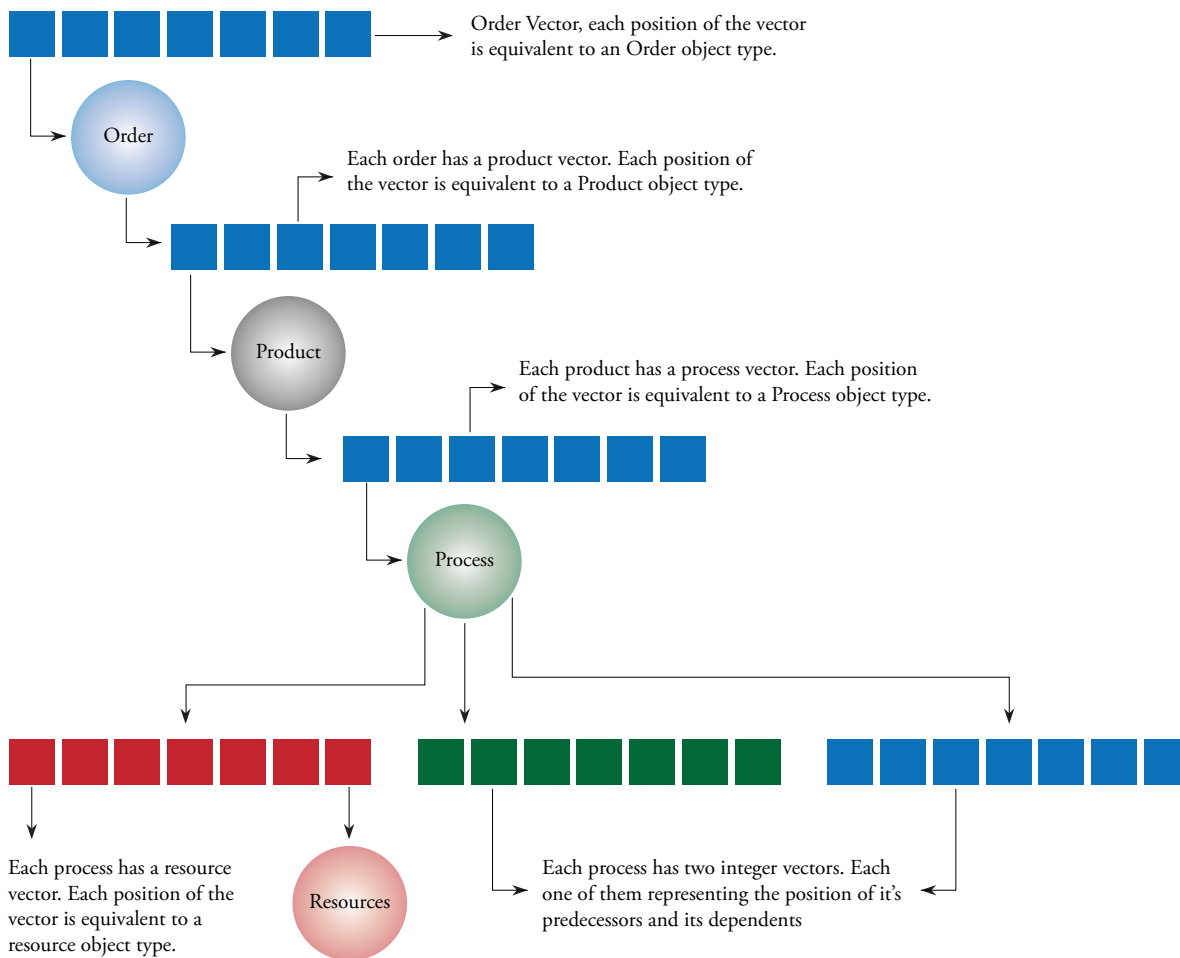
Fig. 1. General architecture of the DSS.
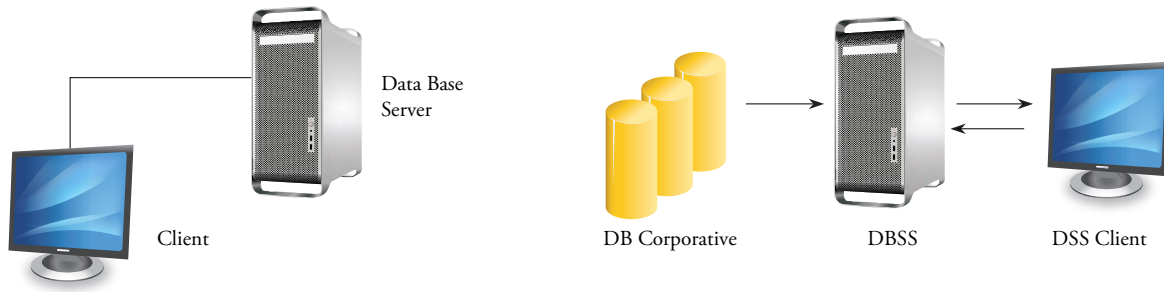
Source: Personal creation

structures has the following Multi-objective function:

- Minimizing costs and the total processing time. Subject to these restrictions:

1. Validate the working day for (Processes)$_k$
2. Validate the execution time (Processes)$_k$
3. Validate the availability of (Resources)$_j$

Fig. 2. Data structures as implemented on a logical and DSS programming level.

Order Vector, each position of the vector is equivalent to an Order object type.

Order

Each order has a product vector. Each position of the vector is equivalent to a Product object type.

Product

Each product has a process vector. Each position of the vector is equivalent to a Process object type.

Process

Each process has a resource vector. Each position of the vector is equivalent to a resource object type.

Resources

Each process has two integer vectors. Each one of them representing the position of it's predecessors and its dependents

Source: Personal creation

Preliminary Heuristics

The DSS operates by means of a programming algorithm which in order to issue schedules that are as optimized as possible regarding cost vs. time suggests four mechanisms before the execution of the programming algorithm, as follows:

• Organize all production order by applying heuristics for EDD, SPT, SMPT, and LPT.

• Apply the heuristics that were mentioned in the previous item to the products in each one of the orders.

• Organize the internal processes of each one of the products according to their levels

• Randomly select the resources needed to carry out a process.

Scheduling Algorithm

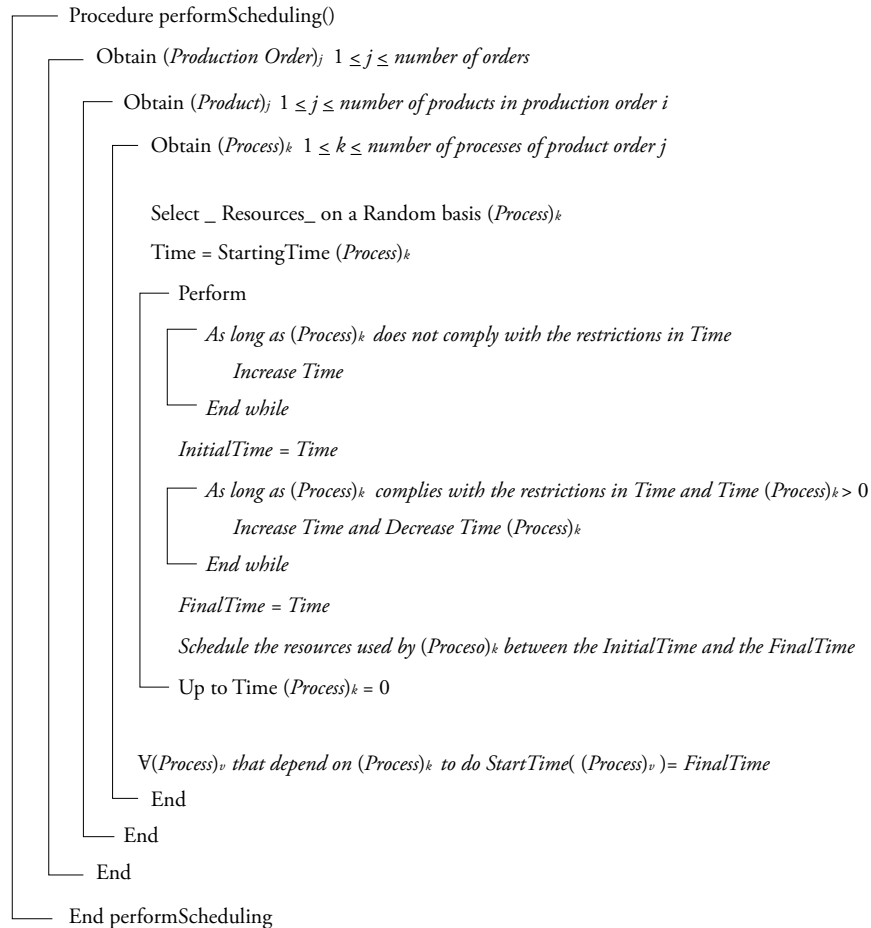There are two main objectives to the algorithm that has been described in Fig. 3, one is to minimize the idle time of the resources which will implicitly result in the optimization of the time needed to schedule the various activities that are part of the production process of the company and minimize costs by randomizing the selection of the resources involved in the operation of the process.

As a whole the DSS operates with heuristic techniques that from the beginning start contributing new solutions to the scheduling process, by determining each one of these and thanks to the randomization of the selection process of the resources as well as the minimization of the time needed to carry out the processes, various solutions can be obtained through one same heuristic such as EDD, SPT, among others, the general work flow of the system can be seen in the Fig. 3.
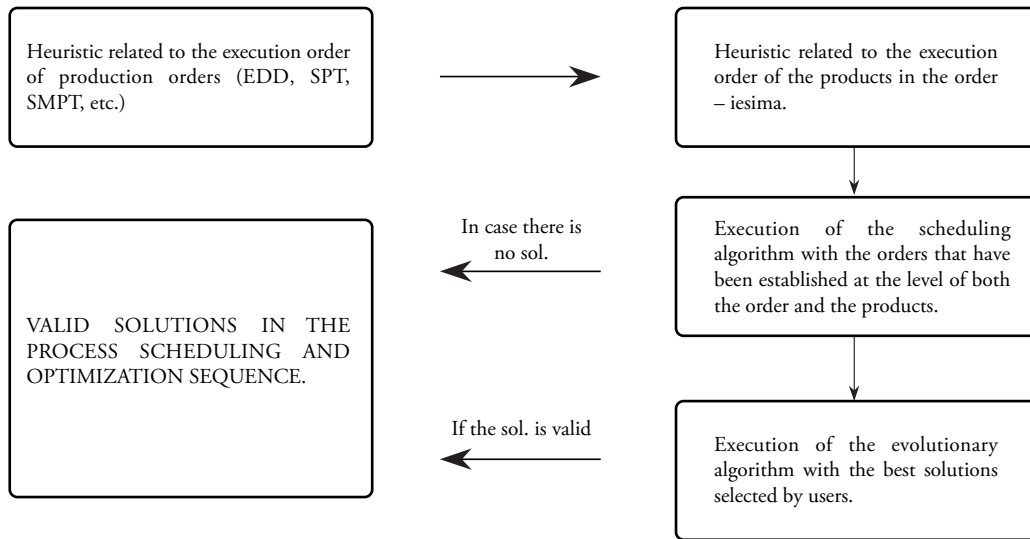
Fig. 3. Design and scheduling algorithm as implemented in the DSS.

Procedure performScheduling()

Obtain (*Production Order*)$_j$ $1 \leq j \leq$ *number of orders*

Obtain (*Product*)$_j$ $1 \leq j \leq$ *number of products in production order i*

Obtain (*Process*)$_k$ $1 \leq k \leq$ *number of processes of product order j*

Select _ Resources_ on a Random basis (*Process*)$_k$

Time = StartingTime (*Process*)$_k$

Perform

As long as (*Process*)$_k$ *does not comply with the restrictions in Time*

Increase Time

End while

InitialTime = Time

As long as (*Process*)$_k$ *complies with the restrictions in Time and Time* (*Process*)$_k$ > 0

Increase Time and Decrease Time (*Process*)$_k$

End while

FinalTime = Time

Schedule the resources used by (*Proceso*)$_k$ *between the InitialTime and the FinalTime*

Up to Time (*Process*)$_k$ = 0

∀(*Process*)$_v$ *that depend on* (*Process*)$_k$ *to do StartTime(* (*Process*)$_v$ *)= FinalTime*

End

End

End

End performScheduling

Source: Personal creation

Fig. 4. DSS performance sequence.


```
┌─────────────────────────┐              ┌─────────────────────────┐
│ Heuristic related to    │              │ Heuristic related to    │
│ the execution order     │  ──────────▶ │ the execution order of  │
│ of production orders    │              │ the products in the     │
│ (EDD, SPT, SMPT, etc.)  │              │ order – iesima.         │
└─────────────────────────┘              └─────────────────────────┘
                                                      │
                         In case there is             ▼
                            no sol.        ┌─────────────────────────┐
┌─────────────────────────┐ ◀──────────── │ Execution of the        │
│                         │               │ scheduling algorithm    │
│ VALID SOLUTIONS IN THE  │               │ with the orders that    │
│ PROCESS SCHEDULING AND  │               │ have been established    │
│ OPTIMIZATION SEQUENCE.  │               │ at the level of both    │
│                         │               │ the order and the       │
│                         │               │ products.               │
│                         │ If the sol.   └─────────────────────────┘
│                         │ is valid                 │
│                         │ ◀──────────── ┌─────────────────────────┐
└─────────────────────────┘               │ Execution of the        │
                                          │ evolutionary algorithm  │
                                          │ with the best solutions │
                                          │ selected by users.      │
                                          └─────────────────────────┘
```

Source: Personal creation

Evolutionary Algorithm

Once having finalized the scheduling process, by getting the end user to select the best solutions the evolutionary algorithm is carried out. This algorithm has been specified in Fig. 5. The main objective of the algorithm is to obtain new solutions that somehow get to excel the ones suggested by the said algorithm through cross operators with Solutions that are generated from the scheduling algorithm.

DSS Interfaces
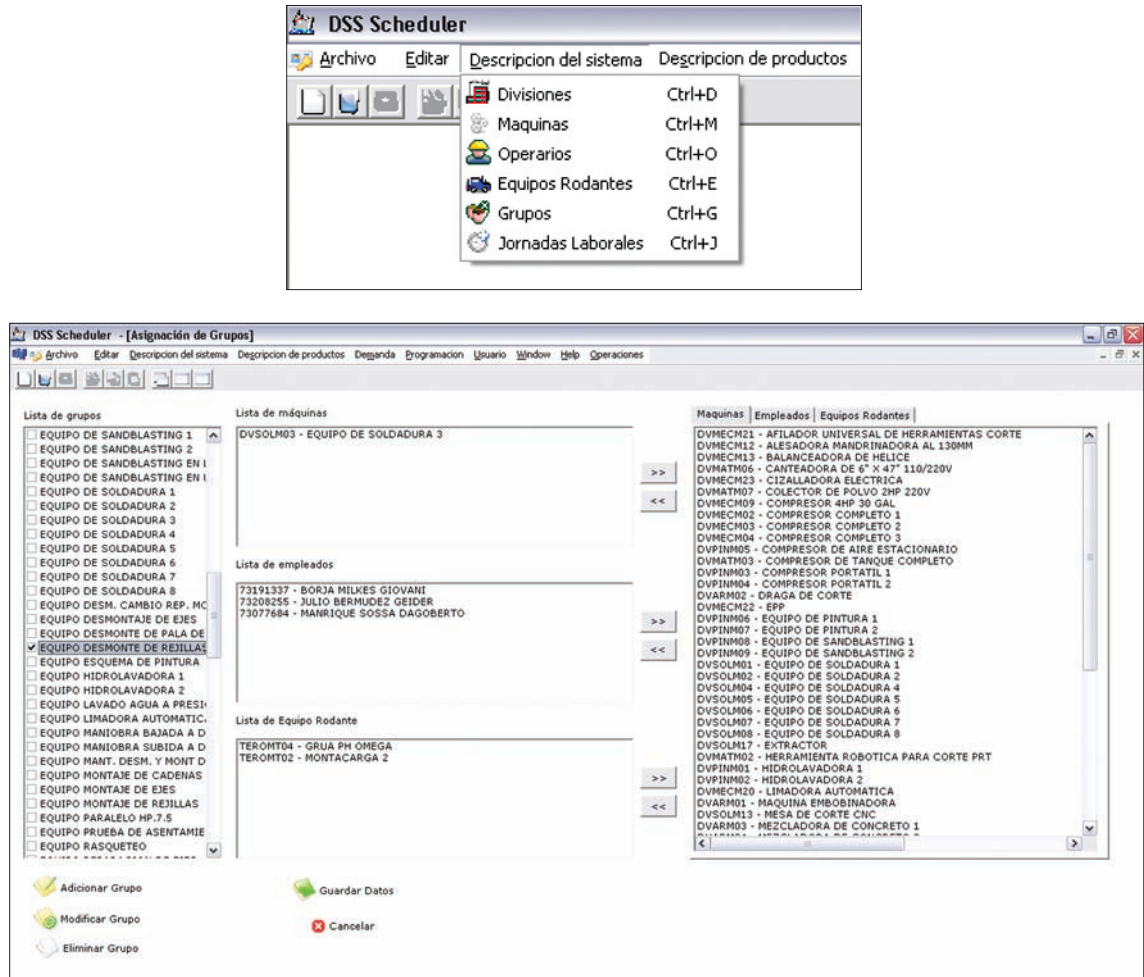
Productive system configuration

In this module you will find the definition of all the topics related to how the plant is doing, where the processes are going to be performed, the company's divisions or areas are also defined here, and so are the resources and resource associations for the creation of work teams and work days.

Fig. 5. The structure of the optimization evolutionary algorithm.

Scheduling optimization Procedure()

As long as the solution does not improve in θ ggenerations and the maximum iteration number is not over passed do Mi

Obtain $(Solution)_k$ $1 \leq j \leq$ *number of solution*

Obtain $(Solution)_j$ $1 \leq j \leq$ *number of solution*

Load solution $(Solution)_k$

Load solution $(Solution)_j$

New Solution = Perform Crossing $((Solution)_k,(Solution)_j)((Solution)_k,(Solution)_j)$
If it is valid New Solution then Save $(Solution)_{new\ solution}$

End

End

End

End Optimize Scheduling

Source: Personal creation

Fig. 6. System Configuration.
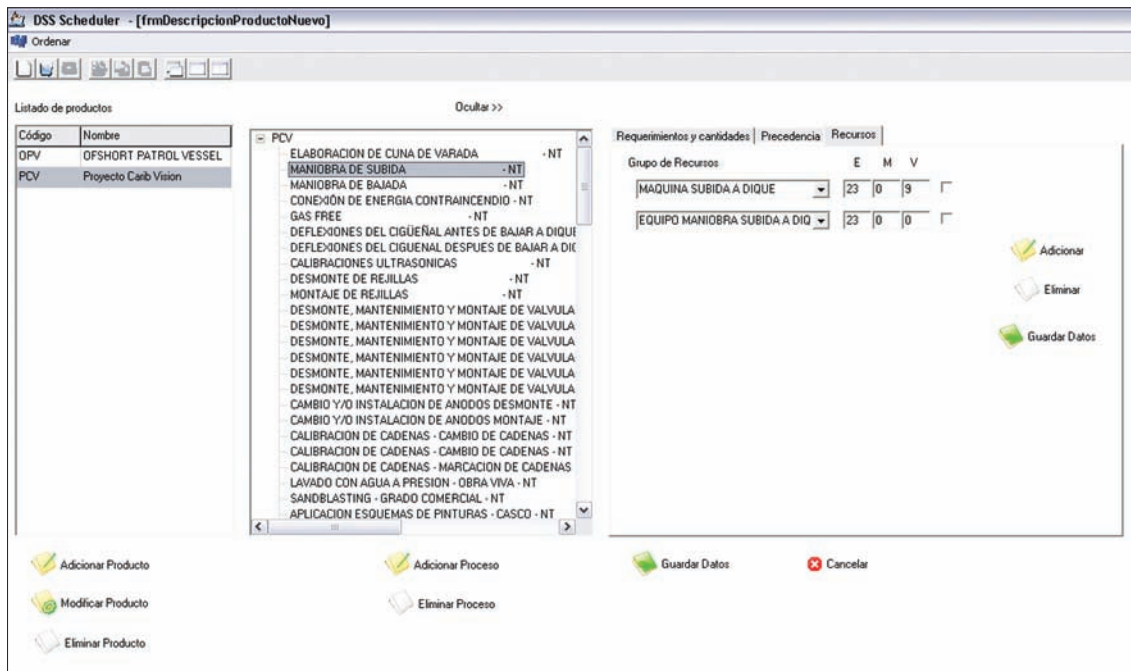

Source: DSS-SCHEDULER Modules.

Product configuration within the system

This module specifies the requirements needed to make a product. The product tree is done representing what is needed to make a product, the processes, and requirements for by- products or raw materials, priorities and resource groups.

In the Fig. 8 we are able to see the configuration of the product tree, where we can find the finished product at the root. This finished product may very well be the unit. The finalized product, naval device or a service to be rendered, and if we go down each one of the levels we can see how each sub level

Fig. 7. Product description.
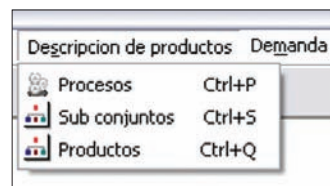
Source: DSS-SCHEDULER Modules.

of the tree is made up that refers to intermediate products until we reach the leaves of the tree where the requirements for raw materials can be found as well as some intermediate products. The processes required to be performed in order to complete a level and keep moving on until we get to the root of the tree, which is the final product to be developed, are associated to these levels.

Fig. 8. Description of the tree of the product.
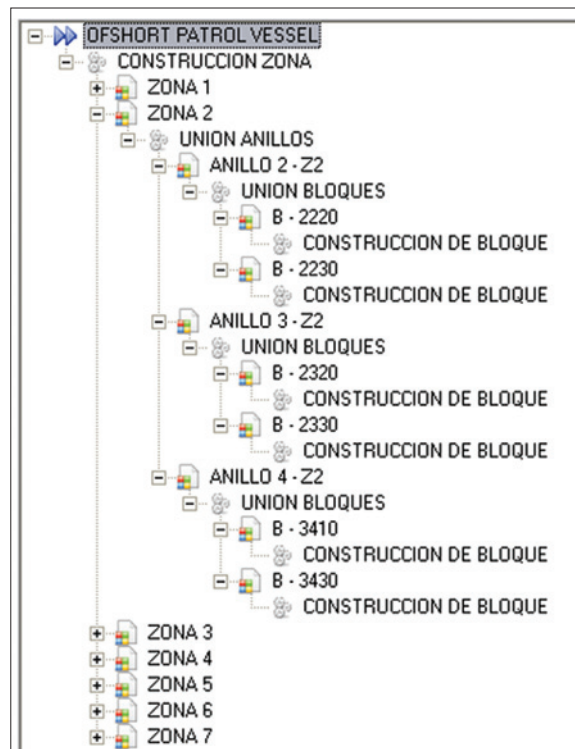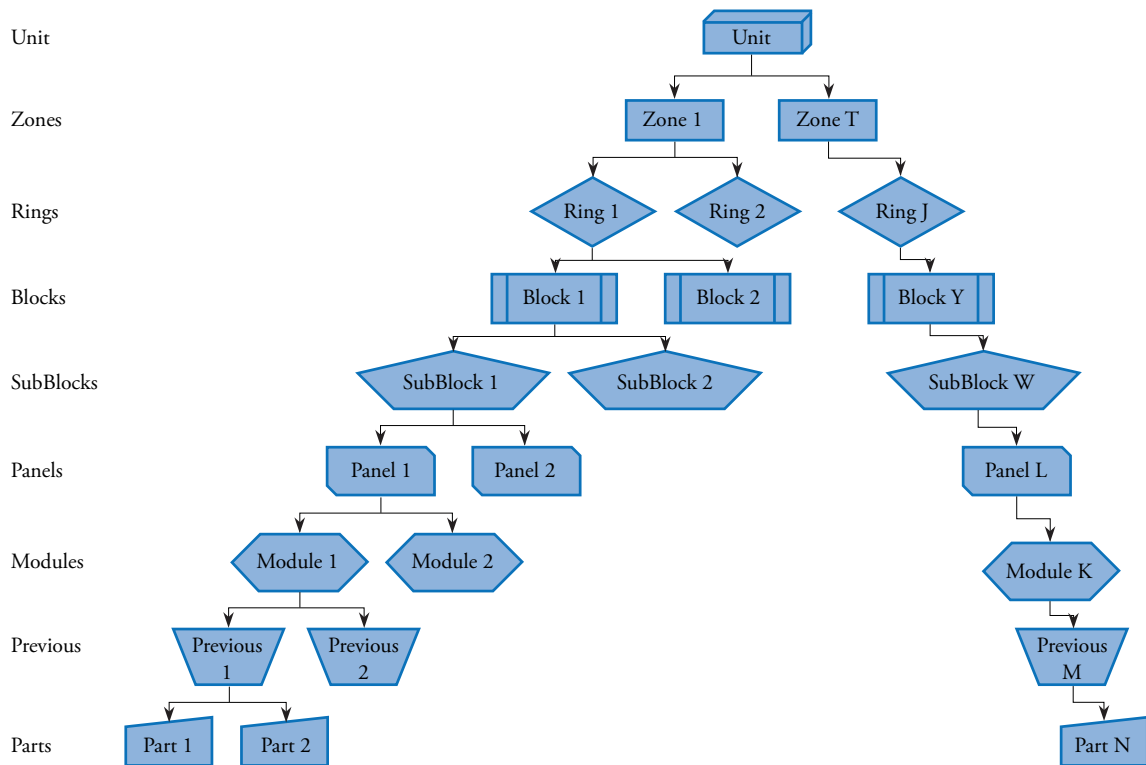
Source: Personal creation

Description of the Orders

This module defines the customers, priorities, orders of purchase and production orders.

Reports

DSS-Scheduler has text reports where all the optimal solutions can be visualized as found by the optimizer and it allows for the exploration of each one of them in a detailed fashion. There are also graph reports in Gantt diagram with several filters for the user to be able to generate the reports as needed. Besides these reports, DSS lets us export the schedule as chosen to Microsoft Project where it can have a manipulation that is familiar with the various users the tool has.

Results and implementations

As can be seen in Fig. 11 the DSS return a set of no dominated solutions. Each solution optimizes the overall cost and time of the production. In this test we build a unique ship. The solutions generated by the DSS always try to use equitative the resources for the scheduling step. Due to this, the resources are available for future works. Besides, the DSS uses the resources and take into account the cost. Consequently, the overall cost is minimized.

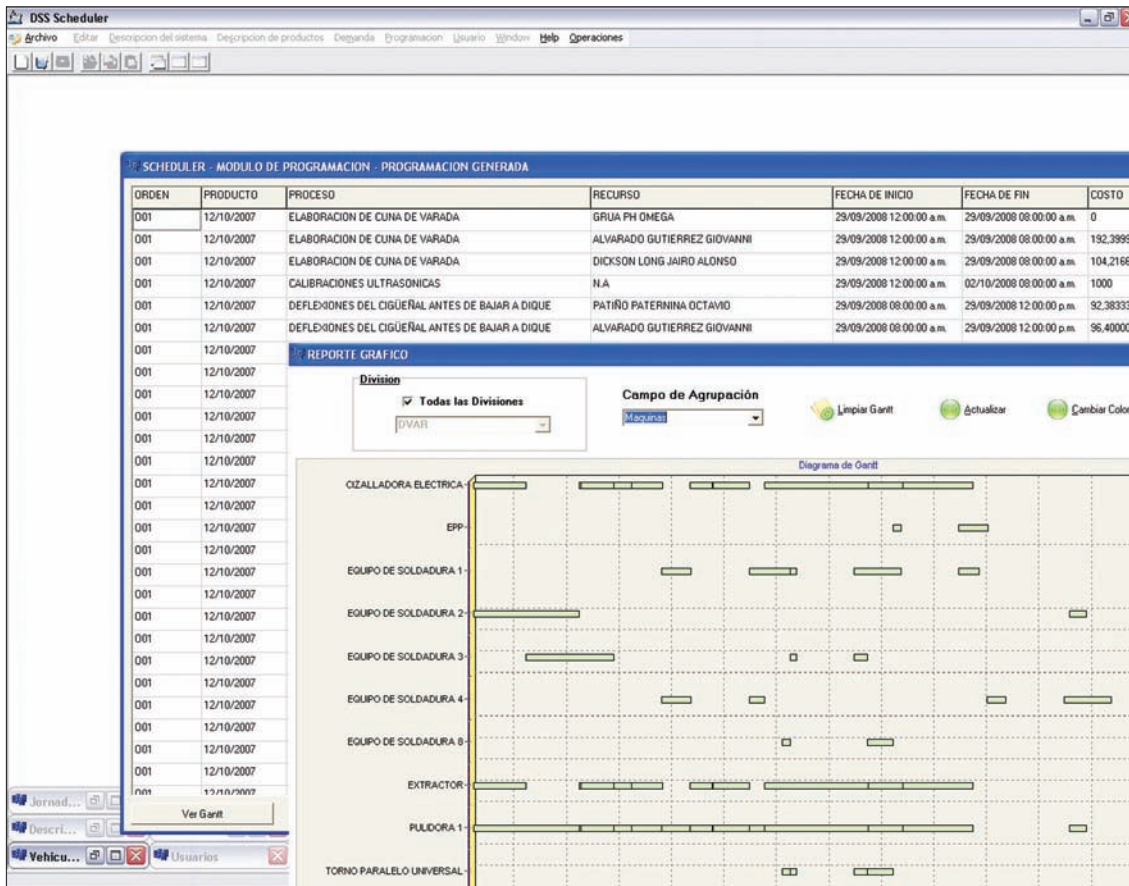As can be seen in Fig. 12 b. The set of solutions are independent.
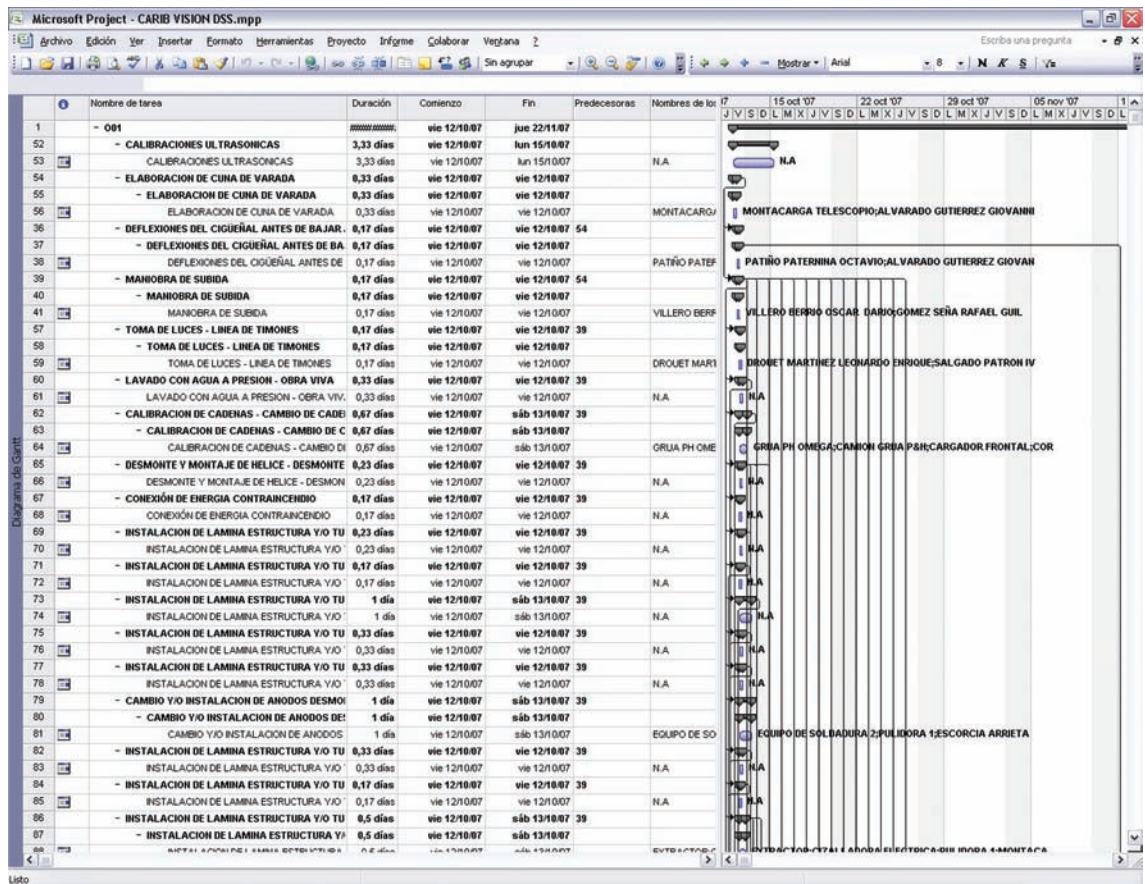
Fig. 9. Demand.
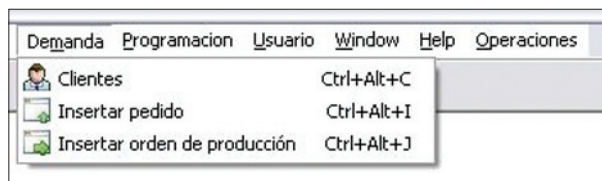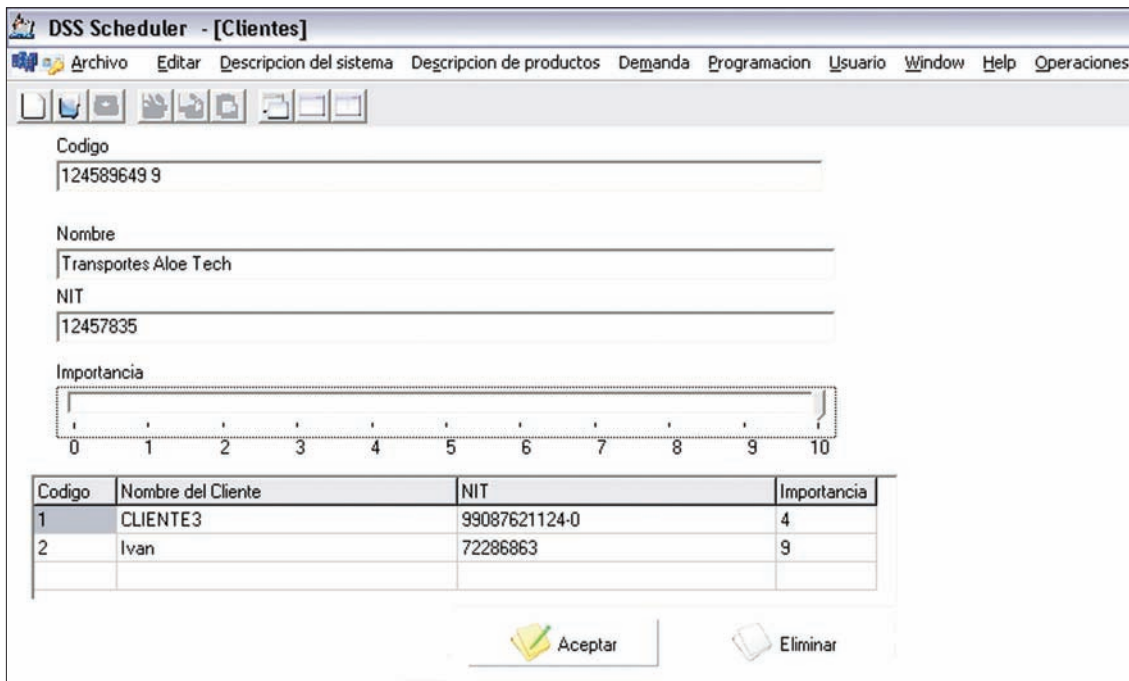
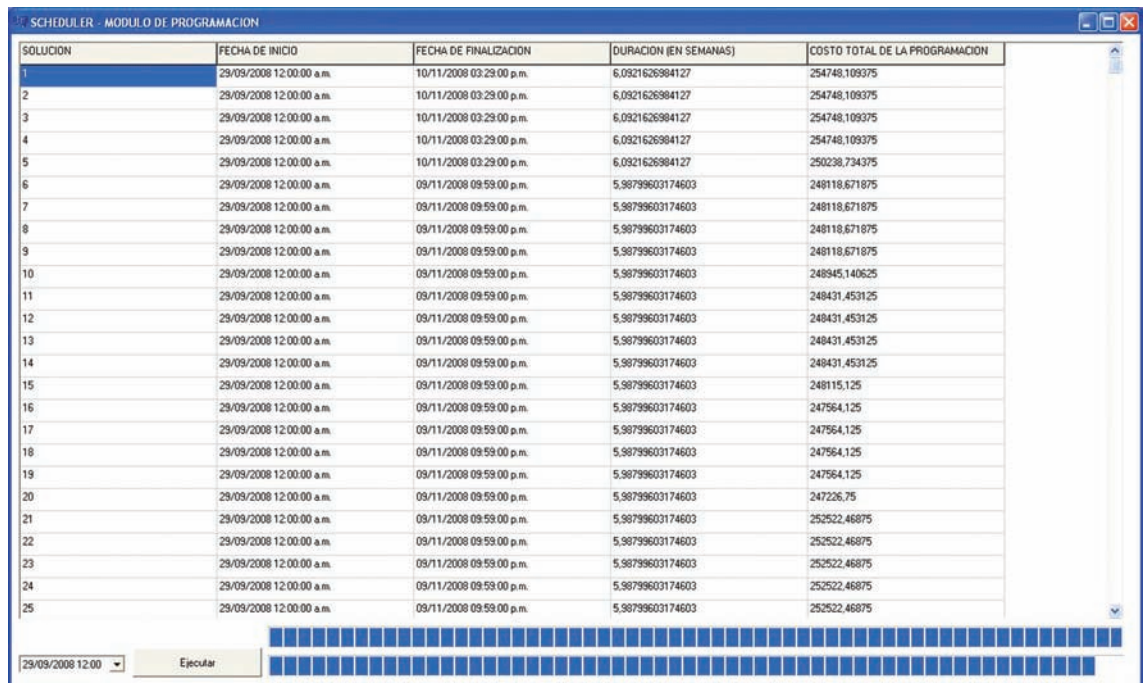Source: DSS-SCHEDULER Modules.

Fig. 10. Reports.

Source: DSS-SCHEDULER Modules.

Fig. 11. Test results promise delivery dates.

Source: DSS-SCHEDULER Modules.

Fig. 12. Resource utilization results.

Using resources for OPV Building

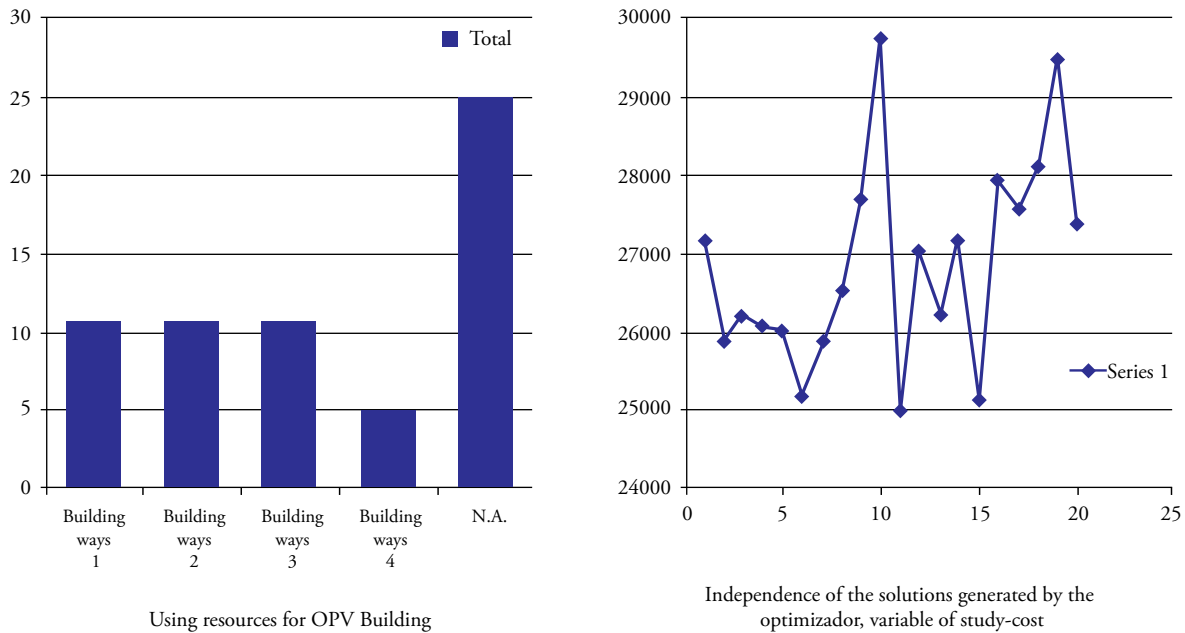Independence of the solutions generated by the optimizador, variable of study-cost
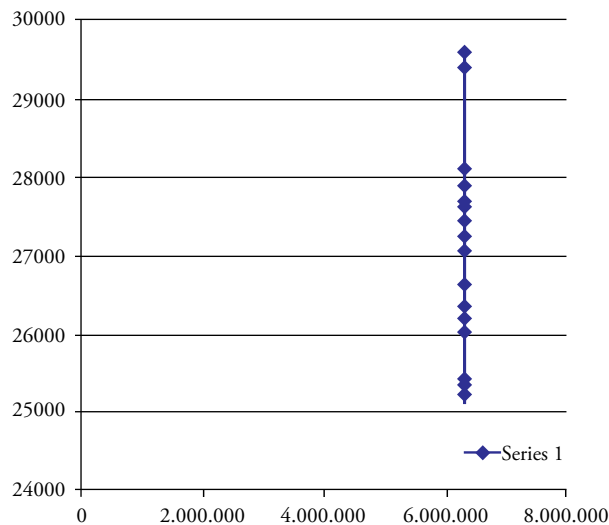
Fig. 13. Solutions generated by the DSS.

The DSS can be applied to companies of the productive sector. Due to this, each company that need a set of process for building a product can use the software for controlling and yielding of the overall production.

Conclusions

DSS is an information system used as a support to make decisions instead of an automatic decision maker. A support to make decisions is regarded as an aid for the people to work alone or in group in such a way that alternatives are generated and the most appropriate decisions can be made based on actual information. This tool has been assessed under the repair works of a ship where the results obtained by the DSS were lower in terms of times of execution as compared to the actual values for the execution of the test project. It is necessary to use the tool to promise delivery dates to the customers and to make an estimate of the costs

involved in the development of the projects to be carried out since it allows for the assessment of the probabilistic scenerios requested by the company's potential clients.

References

CASTILLO, ANA K. y MUŃOS DAVID, *Desarrollo de un sistema para programar recursos y operaciones en la fabricación de calentadores eléctricos bajo pedido*, en revista Facultad de Ingeniería, U.T.A. (Chile), Vol. 12 No. 2, pp. 35-42. 2004.

DELGADO, EFREDY, CORTÉS, CARLOS JULIO y DUARTE, ÓSCAR. 2005. *Aplicación de algoritmos genéticos para la programación de tareas en una celda de manufactura*, en revista de Ingeniería e Investigación (58), Vol. 25, No. 2, pp. 24-31.

KIRAN, ALI S., *Hierarchical Modeling of a Shipyard Integrated with an External Scheduling Application* [conferencia], Simulation Conference. 2001.

LOPEZ DE HARO, SANTIAGO, SANCHEZ MARTÍN, PEDRO y CONDE COLLADO, JAVIER, *Secuenciación de tareas mediante metaheurísticos* [conferencia], VIII Congreso de Ingeniería de Organización. 2004.

MEDEIROS, D. J., *Simulation Based Design for a Shipyard manufacturing Process* [conferencia], Simulation Conference. 2000.

McLEAN, CHARLES, SHAO, GUODONG, *Simulation of Shipbuilding Operations* [conferencia], Simulation Conference. 2001.

SHAFIEI FARZAD y SUNDARAM DAVID, *Multi-Enterprise Collaborative EnterpriseResource Planning and Decision Support Systems* [conferencia], 37th Hawaii International Conference on System Sciences. 2004.