Interactive Scheduling Decision Support System a case study for fertilizer production on supply chain

Ahlam Azzamouri[1], Imane Essaadi[1], Selwa Elfirdoussi[1], Vincent Giard[1,2]
[1] EMINES, University Mohammed VI Polytechnic, 43140 Ben Guerir, Morroco
[2] Paris-Dauphine, PSL Research University, 75016, Paris, France
{Ahlam.Azzamouri, Imane.Essaadi, Selwa.Elfirdoussi, Vincent.Giard}@emines.um6p.ma

Abstract. This paper presents the architecture of an interactive scheduling decision support system (ISDSS) allowing users to find the optimal solution for fertilizer production on parallel heterogeneous processors. The proposed approach takes into account different production process constraints such as launch time, delivery date, preventive maintenance and the impact of scheduling on supply chain management. The ISDSS implemented is run by a relational database used to customize the structural data and the problem parameters. A user interface is available for ISDSS users to define the scheduling problem and design the solution based on tables and graphs for detecting possible issues. The ISDSS architecture was implemented on java using independent modules.

Keywords: Interactive Scheduling Decision Support System (ISDSS), Scheduling, Production plant, Modular architecture.

1 Introduction

After a brief presentation of the decisional problem (§1.1) we present (§1.2) a functional view of the Interactive Decision Support System (ISDSS) we have designed. In section 2, we examine the technical bases of the ISDSS used to schedule fertilizer orders, before illustrating ISDSS implementation and use in Section 3 and concluding in Section 4.

1.1 Analysis of the decisional problem

After clarifying the characteristics of this specific scheduling problem (§a), one will briefly review two possible models, one of which has been implemented in the DSS (§b).

a) Characteristics of the scheduling problem in a fertilizer production plant

OCP SA (http://www.ocpgroup.ma/fr) accounts for one third of world phosphate exports across product segments. The Jorf fertilizer production plant, located near the sea at the end of the Northern axis of OCP's supply chain (SC), is the Moroccan group's largest plant. It produces some twenty three fertilizer references at 7 parallel lines (3 "107" identical lines and 4 "07" identical lines); 8 out of 33 references may only be produced at a single type of line, which yields 58 routings. Fertilizer demand is seasonal and produced to order, with shipping performed by boat at the harbor located near the fertilizer plant.

Each order is characterized by a fertilizer reference to be manufactured, a tonnage and a time window during which production must be completed. The schedule must factor in production stoppage required by preventive maintenance operations, and such stoppage has no impact on ongoing production time. Order fulfillment time varies from line to line, in particular due to their different technical features and due to production rate modulation options. The schedule must also factor in substantial cleaning operation time between two batches of production of different quality on the same line. Such line preparation lead time depends on the line itself and on the previously produced reference. Such a complexity is not considered in the scheduling literature [1, 2]. Besides, the issue dealt with is always local and ignores the consequences of the proposed schedule both downstream and upstream in the SC. However, in our context, this solution has immediate consequences that may make it unfeasible due to pump priming issues (upstream) or finished product inventory saturation (downstream).

Fertilizer manufacturing requires, among other inputs, sulfuric acid and phosphoric acid, the latter using sulfuric acid as a component. The Jorf site SC comprises production units for both acids. Moreover, the phosphoric acid that it produces is also exported and shipped and both acids may be used by the Joint Ventures (JV) also operating at Jorf. An analysis of the fertilizer BOM's reveals a strong dispersion on the consumption rate for these inputs. This means that the adopted scheduling has a strong impact on consumption rate for these acids and that the schedule is only workable if it does not deplete inventory for these raw materials.

The manufactured fertilizers are carried by conveyer belts out of the six storage areas. The storage areas are dedicated to the fertilizer references belonging to a particular family. Area capacity varies depending on the number of stored references, as any mixing of fertilizers is highly hazardous. The allocation of such areas to product families' changes over time and several areas may be dedicated to a single family. The chosen schedule determines fertilizer inventory buildup and will not be workable if inventory buildup is not matched by storage capacity.

Production stoppage risk due to pump priming issues (upstream) and/or saturated storage (downstream) serves to place the scheduling problem in the wider framework of SC management for the relevant fertilizer plant. We have built two MILP models [1, 2] , using the Algebraic Modelling Languages (AML) of Xpress-IVE, that takes simultaneously into account all the characteristics of the local scheduling problem described above. Moreover, one of them considers temporal constraints on input supplies and on output storage to prevent priming and saturation issues. Unfortunately, the model only works with small instances due to the size of the generated problem. The second model is simpler though it solves a very complex local problem but it can be used for reasonably large instances. To guarantee the feasibility of the solution, an optimization module based on this model is embedded in a Decision Support System (DSS) to help managers define a feasible optimal problem.

In an integrated SC, this approach is the best one as it is inefficient to monitor an integrated SC locally when ignoring the upstream and downstream consequences of local decisions. What is more, this approach promotes solutions of constraint negotiation between inter-dependent SC entities, which could work as a substitute for global optimization, as this is impossible to achieve.

1.2 ISDSS Functional Architecture

The bases for DSS were laid by Gorry and Scott Morton as early as 1971 [3] and transformed into a structured approach by Keen and Scott Morton in 1978 [4]. IS/IT development, of course, immensely increased the potential of DSS [5, 6, 7] without altering its rationale. Basically DSSs comprise: i) an interface to formally express a complex issue that is partially structured to define a structured problem; ii) one or multiple modules to solve the structured problem, usually based on optimization or simulation models [8]; iii) an interface capable of exploring all the consequences of a solution obtained and of either adjusting it marginally or of validating it; iv) if none of these solutions is acceptable, the DSS is then used to define a new structured problem using the feedback from prior formulations that failed to deliver a satisfactory solution.

We have designed an ISDSS that applies this functional breakdown. Starting from a partially structured problem, it allows the definition of a structured problem to be optimized after an initial feasibility check. This structured problem is an instance of a generic MIP problem that uses the AML of Xpress-IVE. The scope of the problem is local as it lifts input availability and output storage capacity constraints. Optimization yields a solution whose results are used by the third functional component that takes into account the lifted constraints. If the solution found, which can be marginally changed, is not feasible, one returns to the structured problem formulation stage. The flowchart below follows the original pattern of the DSS designers, with its four components:

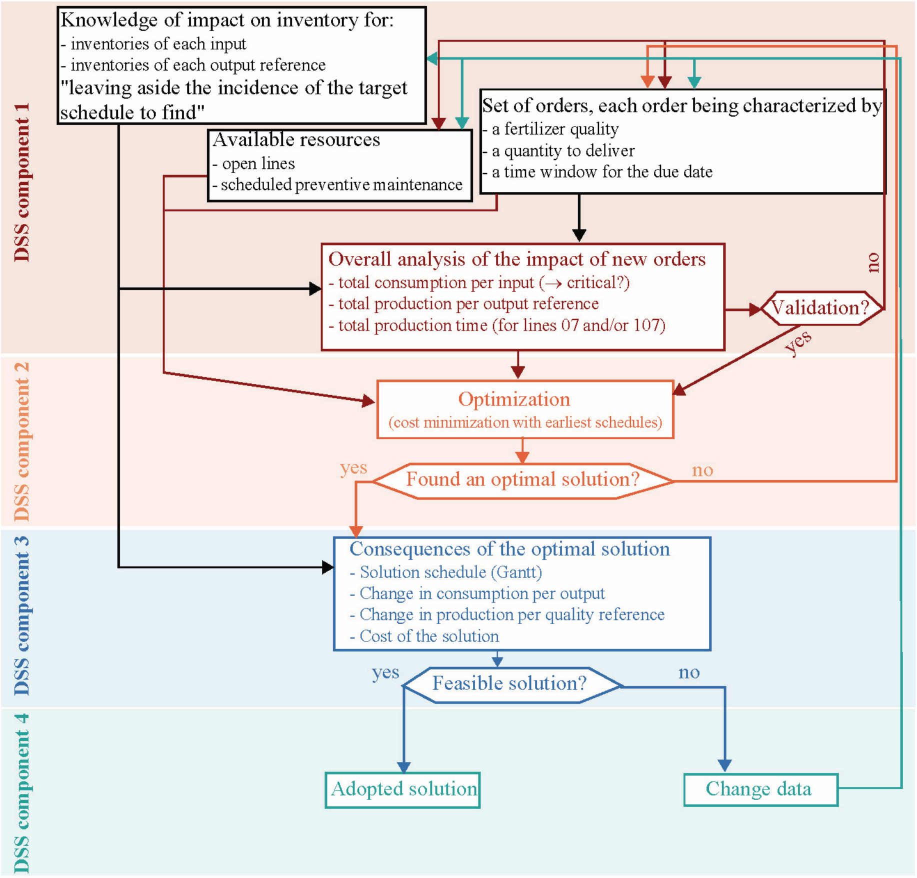


Fig. 1. ISDSS Functional Components

2 Technical bases of ISDSS

We begin with a literature review (§2.1), as this bring out the originality of our ISDSS. We then go on to explain the architecture we implemented (§2.2).

2.1 Literature review

The following analysis reviews a number of papers published in scientific newspaper or proceedings focusing on Decision Support Systems for production scheduling.

An optimization-based Decision Support System (OBDSS) generator, designed by [9] automatically generates a database, a solver and an interface (GUI) using

PowerBuilder. They applied the OBDSS generated in the production plant. The approach, called GESCOPP, uses SML technology to represent the optimization model and generate the database.

Olteanu (2011) addresses in [10] production plant in the biofuel industry. His system optimizes the main activities of the biofuel value chain and supports the decision-making process at management level. It reduces total production cost to the lowest possible level under current technological and market constraints. The shortcoming of the bioPur.OLT.SYS proposed is that is doesn't optimize the production plant process based on the processors and machine constraints.

A simulation-based DSS (GeSIM) is proposed [11] for customer-driven manufacturing. The approach aims to integrate data, generate the simulation model, and display the results and supports interactive changes. The tool requires user interaction for the information to be correctly defined, and the problem correctly expressed before input into the GeSIM. It is therefore too complex for untrained users and too time-consuming.

In [12], Krishnaiyer and Chen propose a Web-based Visual DSS to schedule the process of letter delivery to customers. The approach takes process production planning into account. The information from different systems is collected and analyzed to yield the optimal solution and ensure a high level of customer service quality. The strength of this project is its reliance on advanced technology such as web-based applications to model a number of services but the scheduling process is inefficient.

The power industry is a highly complex and critical field, where financial risk is high. Production Scheduling is an important area of research in this sector. The power generation department of the National Institute of Technology proposes [13] a DSS for operation scheduling and optimization of Sewa Hydro Electric Plan (SHEP). The system was implemented to assist SHEP to make decisions for the optimal use of available water resources at State level combined with the minimization of environmental impact.

IBM research authors describe in [14] a DSS for paper production scheduling (A-Team) which was deployed in IBM environment. A-Team is made up of three types of agents which work asynchronously: constructors, improvers, and destroyers. The 'Constructors' agent only deals with problem definition and creates new solutions. The 'Improvers' agent attempts to improve upon the current set of solutions by modifying or combining existing solutions. The 'Destroyers' agent attempts to limit the number of solutions and focus the efforts of 'Improvers' by eliminating poor solutions. Freed et al. [15] present a scheduling decision- support system implemented on VBA called the Dispatcher.

Galasso et al. [16] suggest a model of the decision-making parameters involved in the production management process. Their DSS framework aims to schedule the supply chain based on flexible demand. These authors deal in [17] with the design of a DSS that integrates machine scheduling with inventory management for a multi-product manufacturing industry. Other algorithms are proposed [18] to determine a model for production scheduling based on constraint parameters. The paper [19] describes an intelligent DSS for real-time control of a flexible manufacturing system.

Based on the above literature review, we have defined four analytical dimensions for the DSS to be designed, all of which may not all be used in a DSS: a) application type implemented to run the DSS (desktop application vs web-based); b) problem solving; c) interface used to define the problem or design the solution; d) DSS proposing generic system.

a) Some authors experiment their DSS using such desktop applications as Excel or Microsoft Access (MA) [11, 13, 15, 16, 19, 18]. Although Excel may be used to manage very simple data base, as it is limited to just 2 dimensions and doesn't support the relational concept. Similarly, MA is limited to the Windows platform and doesn't provide high scalability and security. In the same analytical dimension, other solutions consist in web-based approaches integrating data from different systems [12, 17].

b) Each DSS requires problem-solving capacities. The proposed optimization processes are based on different local approaches (simulation [11, 15, 19], optimization [13], heuristics [20]) or general software (ERP) and their APS (Advanced Planning System, [12]). Some of those DSSs are implemented through such programming languages as VB (Visual Basic), VBA (Visual Basic for Application), Visual C++. Although VBA or VB are user-friendly languages, they are Windows-compatible only and exclude an object-oriented approach.

c) Few DSSs propose interfaces able to transform a semi-structured problem into a structured one. Downstream, the results processing interface is generally quite basic. The DSSs proposed in [9, 11, 12, 13, 15, 17, 19] use Web-based technology, Visual Basic or Visual C++ or DSS generator.

d) Finally, some papers describe approaches using generic system aiming to take the three first dimensions (a, b, c) into account [14, 9] using technology such as IBM or DSS generators. Nowadays IBM products deliver multiple benefits especially in dealing with big data and computing capacity. Nevertheless, the most IBM packages will only work in IBM environments and deployment is very expensive. Turning to the DSS generator, this is a general approach based on structural data medialization and its implementation using a tagging system such as XML (eXtensible Markup Language). In itself, the general nature of such generators is restrictive because, for each request, one needs to formalize the new data and make fresh definitions of the constraints according to the working environment. From the technical point of view, these technologies are very costly and time-consuming to deploy.

2.2 ISDSS Technical Architecture

The structured problem of our ISDSS optimization modules is modeled using an AML (Algebraic Modeling Language), which allows dissociating the formulation of a generic problem from a data set, thereby enabling the instantiation of a specific problem. The cardinalities of the set of parameters and variables are totally customizable. To preserve this thorough customization capacity, it makes sense to use a relational database for the ISDSS, as this is also characterized by no-predefined cardinalities of the entities and associations and guarantees the integrity of the data in the database under a set of given rules. These characteristics lend the ISDSS an ability to adjust immediately to any changes in the number of parallel processors, inputs, outputs, routings, bills of materials (BOM) and costs. They also enable definition of a scheduling problem for any number of orders. The general architecture of the ISDSS is presented on Fig.2.

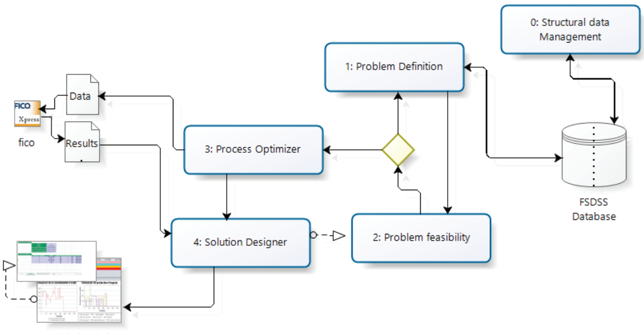


Fig. 2. ISDSS Architecture

a) Data Management module

This module is used to define the structural characteristics of the scheduling problem: production lines, references of fertilizers that these lines can produce, inputs (acids ...) used in producing these fertilizers, fertilizer BOM for a given line (the chemical performance of these lines may vary), launching time of a reference on a line succeeding another fertilizer reference. This data is relatively stable and only evolves in the event of cancellation, transformation or addition of fertilizer lines and / or references.

The interfaces used (e.g., Fig. 4 and 5) ensure the integrity of this structural information, in the conventional sense of being retained in the relational database but also while integrating physical constraints. For example, in BOM definition, the sum of the inputs needed to produce a tone of fertilizer in a given line (and hence with a certain technology) must be equal to that of its outputs (tone of fertilizer, slag and gaseous emissions).

b) Problem Definition Module

The definition of a scheduling problem requires the structural data referred to above and supplements them with the following relative data:
- Available production resources: open lines, running mode (production flow), planned maintenance program, orders in production;
- the orders to be fulfilled: an order is characterized by a quantity of fertilizer to be produced, this production having to be completed in a time window, and a coefficient allowing to set its scheduling at the earliest or at the latest opportunity when a slack exists;
- The projected change in stocks of inputs and fertilizers, excluding the impact of the scheduling to be defined; this information is necessary to ensure feasibility of the optimal solution found.

c) Problem Feasibility module

This module allows the user to test the possibility of finding a solution, to avoid launching the optimization unnecessarily. The test, described by the algorithm described below, leads to three possible diagnostics: a solution exists, a solution seems possible and no solution exists. In the first two cases, the optimization process must be launched. In the last one, we must return to the structured problem definition.

This module checks the problem feasibility at local level (i.e. without checking the upstream and downstream impacts of the solution) – see algorithm 1 - and avoids trying to solve a problem for which no solution exists.

Algorithm 1: Problem Feasibility

```
1. Count the period concerning production and verify that this number is smaller than the production
   time required for each order. If not go to 6.
2. Place each order using a backward scheduling with the latest due dates and including the
   preventive maintenance allowance. If possible go to 4, if not go to 3.
3. Place each order using a forward scheduling with the earliest due dates and including the
   preventive maintenance allowance. If possible go to 4, if not go to 5.
4. End: the problem is feasible.
5. End: the problem posed seems feasible.
6. End: the problem is not feasible.
```

If the problem is feasible, a file containing all problem parameters is generated. Its structure is defined by the generic optimization model integrated into the ISDSS. The combination of this generic model and this dataset generates an instance of the fertilizer scheduling problem.

d) Process Optimizer module

A generic problem [1, 2] based on the AML of FICO Xpress [20] is activated by a dataset including all the information defining the specific problem to generate a specific problem to be solved. The file of the proposed solution is then available for the solution analysis module. This model, created with the Fico Xpress-IVE modeller [20], imports the dataset created by the previous module. The optimization results are then exported to a file that is used by the next module.

e) Interactive Solution Analysis

The solution contains a proposed schedule for each production line and for each order, using tables and colored Gantt diagrams. The Gantt take into account the preventive maintenance allowance and initial occupation of lines. After this, different graphs are proposed to illustrate the progress of each fertilizer produced and the input consumptions implied by the schedule. This result presentation is in the form of a user-friendly graphical interface for efficient use. It helps users make informed decisions or reformulate the problem according to the supply chain manager's need. This flexibility is one of the DSS's crucial benefits.

The optimal solution being that of a local problem that relaxes the constraints of availability of the acids consumed and the storage capacity of the fertilizers produced. The analysis of the stock levels makes it possible to check immediately whether the proposed solution is feasible. If this is not the case, a first possibility is offered to the user who can point to the task corresponding to the programming of a command to move it, while respecting the constraints of non-overlap of the productions and those of the windows of the production end dates for all the orders programmed on the line. The solution thus obtained remains optimal (the same production costs) and only uses the possible slacks of the obtained solution which yields the earliest possible schedules of orders (driven by the parameterization of the objective function). Of course, scheduling changes are immediately reflected in the of acid consumption and fertilizer production graphs.

If this approach does not yield a feasible optimal solution, we have to return to the definition of the structured problem to be solved; for this, several non-exclusive tracks are available. The first is a change in the coefficients applied to orders to push their scheduling at the earliest or at the latest available opportunity. The second is to change problem parameters of the: splitting orders into two to allow parallel production, changing the date interval of end of production for the orders or the maintenance programs or increasing the number of production lines (if possible).

3 ISDSS: Implementation & Experiment

In this section, we address the technologies implemented in our ISDSS (§3.1) and we illustrate the use of ISDSS using a case study (§3.2).

3.1 ISDSS Implementation

Obviously, the success of any information system depends on the technologies deployed and on the integrity of the data managed by the database. Our work differs from that of the papers discussed in section 2.1. Indeed, we propose a framework relative to an ISDSS using generic and portable implementation combined with data management which ensures integrity and uses an object-oriented concept enabling proper flexibility of the cardinalities. All the ISDSS modules are grouped into a package that can run separately or following the ISDSS workflow.

For the deployment environment, the ISDSS is implemented on java which is easy to learn, object-oriented, robust, safe and platform-independent. The ISDSS API implements different libraries, such as mosel.jar to interact with external tools or jfreechart.jar to use graphical presentation. It can be integrated into any type of system and supports all database server providers. We have chosen MySQL for our prototype based on our conceptual database presented on Fig.3.

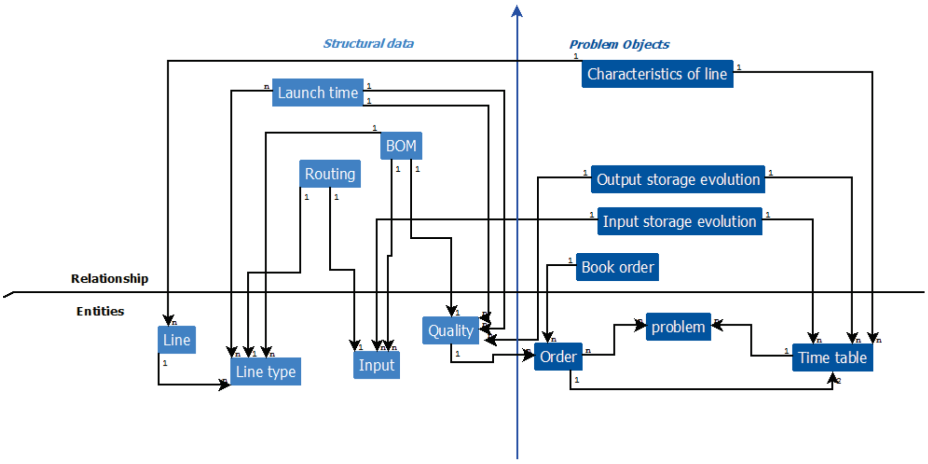


Fig. 3. ISDSS Conceptual Database

ISDSS has a Graphical User Interface (GUI) that allows the users to manipulate the structural data easily, define the problem, test its feasibility, solve it and analyze the solution proposed. It also provides different menus to manage all ISDSS objects.A key feature of the ISDSS framework is data management. In fact, all the information used in the ISDSS have specific properties that we have modeled as objects defined by different attributes and methods. The ISDSS API is defined using different packages.

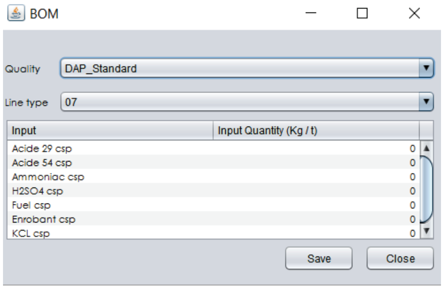
Each of them includes the classes relative to the ISDSS module. Most of the classes extend the “Jframe” and propose a list of drow-up menus to ensure data integrity. The most important class is “Problem Definition” which delivers a sequence of interfaces that display the components for this module, each interface interacting with the database to generate the parameters required by the optimizer module. Subsequently, the ISDSS interacts with Xpress using generated files containing the previous parameters (data and result), these steps are transparent for the user.

The most interesting feature of our framework is that the structural data is modeled to support industrial standards.

3.2 ISDSS: Experiment

A study of different datasets was used to evaluate our proposed approach and architecture. In this section, we present a scenario using the ISDSS to define a fertilizer scheduling problem, the feasibility algorithm and the solution is then generated.

The case study presented in this section concerns fertilizer production at OCP. The initial setup of ISDSS consists in integration of the structural data related to our customer including the lines, the fertilizer, the inputs, BOM (Fig. 4) and launch time required between qualities (Fig.5).

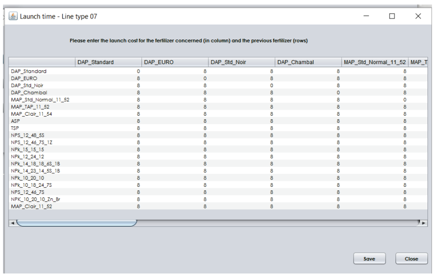


Quality: DAP_Standard

Line type: 07

Input	Input Quantity (Kg / t)
Acide 29 csp	0
Acide 54 csp	0
Ammoniac csp	0
H2SO4 csp	0
Fuel csp	0
Enrobant csp	0
KCL csp	0

Fig. 4. Fertilizer Composition Interface

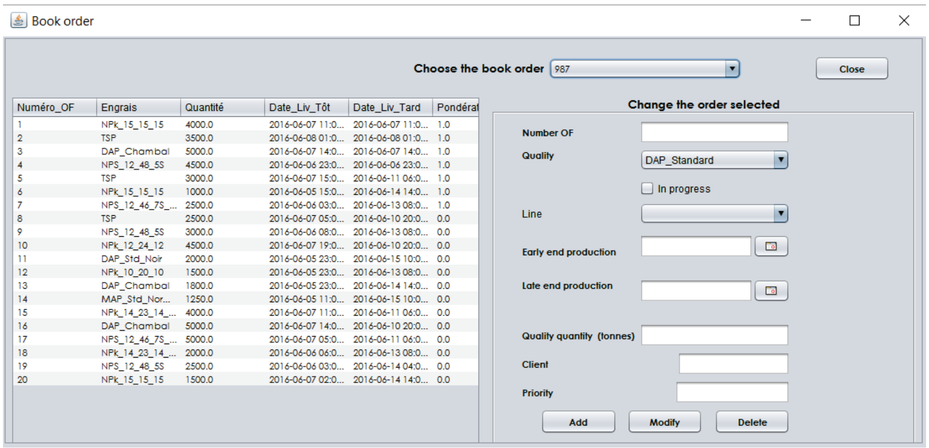


Please enter the launch cost for the fertilizer concerned (in column) and the previous fertilizer (rows)

	DAP_Standard	DAP_EURO	DAP_Std_Noir	DAP_Chambal	MAP_Std_Normal_11_52	MAP_T
DAP_Standard	0	8	8	8	8	
DAP_EURO	8	0	8	8	8	
DAP_Std_Noir	8	8	0	8	8	
DAP_Chambal	8	8	8	0	8	
MAP_Std_Normal_11_52	8	8	8	8	0	
MAP_TAP_11_52	8	8	8	8	8	
MAP_Clair_11_54	8	8	8	8	8	
ASP	8	8	8	8	8	
TSP	8	8	8	8	8	
NPS_12_48_5S	8	8	8	8	8	
NPS_12_46_7S_1Z	8	8	8	8	8	
NPk_15_15_15	8	8	8	8	8	
NPk_12_24_12	8	8	8	8	8	
NPk_14_18_18_6S_1B	8	8	8	8	8	
NPk_14_23_14_5S_1B	8	8	8	8	8	
NPk_10_20_10	8	8	8	8	8	
NPk_10_18_24_7S	8	8	8	8	8	
NPS_12_46_7S	8	8	8	8	8	
NPK_10_20_10_Zn_Br	8	8	8	8	8	
MAP_Clair_11_52	8	8	8	8	8	

Fig. 5. Launch Time Interface

The problem discussed in this model was defined to find the optimal scheduling solution for an order book. The order book concerned contains twenty orders for different customers (Fig. 6) with production in progress on four operational lines and over a 360-hour production horizon. Each order concerns a fertilizer and a customer, specifying the constraints such as the earliest and the latest delivery dates, the weighting directing the orders launches at their earliest or at their latest possibilities given by their time windows. The module also takes into account the order being produced at each production line.



Choose the book order: 987

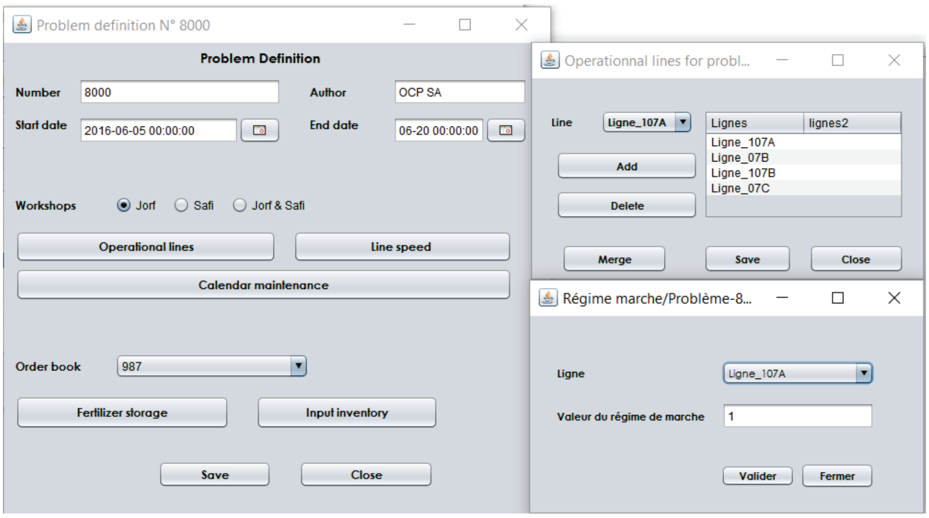
Quality: DAP_Standard

☐ In progress

Numéro_OF	Engrais	Quantité	Date_Liv_Tôt	Date_Liv_Tard	Pondérat
1	NPk_15_15_15	4000.0	2016-06-07 11:0...	2016-06-07 11:0...	1.0
2	TSP	3500.0	2016-06-08 01:0...	2016-06-08 01:0...	1.0
3	DAP_Chambal	5000.0	2016-06-07 14:0...	2016-06-07 14:0...	1.0
4	NPS_12_48_5S	4500.0	2016-06-06 23:0...	2016-06-06 23:0...	1.0
5	TSP	3000.0	2016-06-07 15:0...	2016-06-11 06:0...	1.0
6	NPk_15_15_15	1000.0	2016-06-05 15:0...	2016-06-14 14:0...	1.0
7	NPS_12_46_7S_...	2500.0	2016-06-06 03:0...	2016-06-13 08:0...	1.0
8	TSP	2500.0	2016-06-07 05:0...	2016-06-10 20:0...	0.0
9	NPS_12_48_5S	3000.0	2016-06-06 08:0...	2016-06-13 08:0...	0.0
10	NPk_12_24_12	4500.0	2016-06-07 19:0...	2016-06-10 20:0...	0.0
11	DAP_Std_Noir	2000.0	2016-06-05 23:0...	2016-06-15 10:0...	0.0
12	NPk_10_20_10	1500.0	2016-06-05 23:0...	2016-06-13 08:0...	0.0
13	DAP_Chambal	1800.0	2016-06-05 23:0...	2016-06-14 14:0...	0.0
14	MAP_Std_Nor...	1250.0	2016-06-05 11:0...	2016-06-15 10:0...	0.0
15	NPk_14_23_14_...	4000.0	2016-06-07 11:0...	2016-06-11 06:0...	0.0
16	DAP_Chambal	5000.0	2016-06-07 14:0...	2016-06-10 20:0...	0.0
17	NPS_12_46_7S_...	5000.0	2016-06-07 05:0...	2016-06-11 06:0...	0.0
18	NPk_14_23_14_...	2000.0	2016-06-06 06:0...	2016-06-13 08:0...	0.0
19	NPS_12_48_5S	2500.0	2016-06-06 03:0...	2016-06-14 04:0...	0.0
20	NPk_15_15_15	1500.0	2016-06-07 02:0...	2016-06-14 14:0...	0.0

Fig. 6 : Order Book Management Interface

Once this has been done, the user can generate the problem depending on the available resources (Fig.7). Each problem is identified by a unique number and assigned to an author and can be assigned to one or more workshops. The production time is an important constraint and it depends on delivery time for each order. In the same user form, the author can also change the line characteristics (Fig.7), their speed (Fig.7), specify their preventive maintenance schedules (Fig. 8) and include the storage information for the input or output.


Problem definition N° 8000

Problem Definition

Number: 8000
Author: OCP SA
Start date: 2016-06-05 00:00:00
End date: 06-20 00:00:00
Workshops: ● Jorf ○ Safi ○ Jorf & Safi
Order book: 987

Operationnal lines for probl...

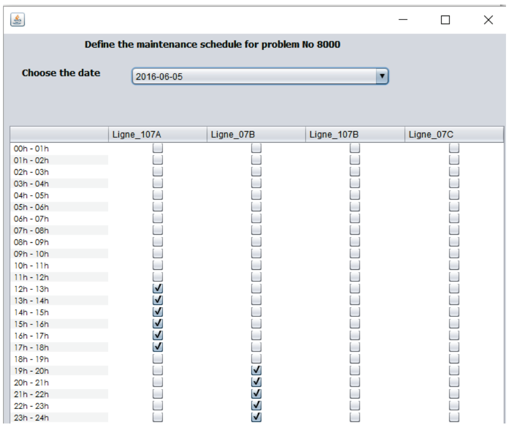
Line: Ligne_107A

Lignes	lignes2
Ligne_107A	
Ligne_07B	
Ligne_107B	
Ligne_07C	

Régime marche/Problème-8...

Ligne: Ligne_107A
Valeur du régime de marche: 1

Fig. 7. Problem Definition Interface


Define the maintenance schedule for problem No 8000

Choose the date: 2016-06-05

	Ligne_107A	Ligne_07B	Ligne_107B	Ligne_07C
00h - 01h	☐	☐	☐	☐
01h - 02h	☐	☐	☐	☐
02h - 03h	☐	☐	☐	☐
03h - 04h	☐	☐	☐	☐
04h - 05h	☐	☐	☐	☐
05h - 06h	☐	☐	☐	☐
06h - 07h	☐	☐	☐	☐
07h - 08h	☐	☐	☐	☐
08h - 09h	☐	☐	☐	☐
09h - 10h	☐	☐	☐	☐
10h - 11h	☐	☐	☐	☐
11h - 12h	☐	☐	☐	☐
12h - 13h	☑	☐	☐	☐
13h - 14h	☑	☐	☐	☐
14h - 15h	☑	☐	☐	☐
15h - 16h	☑	☐	☐	☐
16h - 17h	☑	☐	☐	☐
17h - 18h	☑	☐	☐	☐
18h - 19h	☐	☐	☐	☐
19h - 20h	☐	☑	☐	☐
20h - 21h	☐	☑	☐	☐
21h - 22h	☐	☑	☐	☐
22h - 23h	☐	☑	☐	☐
23h - 24h	☐	☑	☐	☐

Fig. 8. Maintenance Scheduling

Once the problem is defined, ISDSS supports the feasibility study before going ahead with optimization. Fig. 9 presents the optimal solution in a Gantt form; this Gantt can be zoomed to view solution for each order number, with a specific color for each product quality. The maintenance program schedules appear in black in the Gantt. The solution on display also contains the graph of fertilizer production and sulfuric acid 29 consumption, both per period and aggregated (Fig. 10). All this information can be printed or exported to Excel to perform any additional operations as required or to share with different workshop managers.

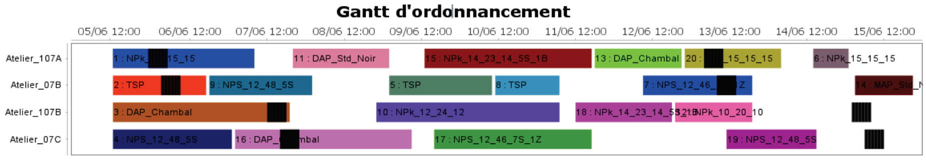


Fig. 9. Gantt diagram proposing the scheduling production

In addition to the optimizer process and to improve the decision support system, ISDSS enables interactive change of the proposed solution using the Gantt diagram. Indeed, the user can drag and drop the different tasks on the date axis or the line axis according the production constraints and proposed planning coherence such as order overlap. Those changes are automatically integrated into the consumption and production graphs for real-time display of the impact of the changes.

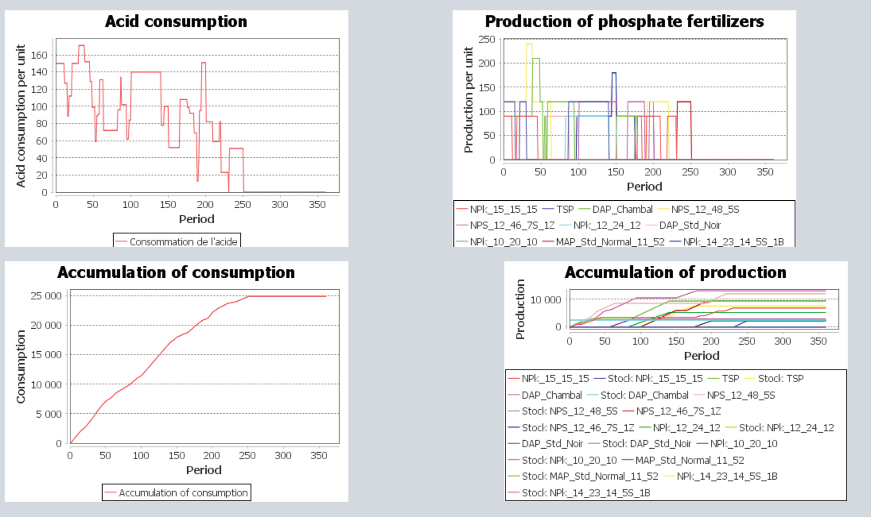


Fig. 10. Consumption and production graphs

4 Conclusions and prospects

Our ISDSS is currently being tested to confirm its relevance for real-life environments. It can be used not only for scheduling purposes over a two to three-week horizon, but also over a longer horizon to test the impact of commercial scenarios with coupling of due dates and prices. This involves starting with formulations using weak constraints that are gradually tightened. The use of the ISDSS approach to improve the overall consistency of the decisions taken locally by different supply chain entities is clearly necessary, especially as it also supports intelligent negotiation of constraints at the interface of these entities.

The characteristics of our ISDSS architecture make it fit for use in other industrial environments confronted with scheduling problems involving parallel heterogeneous processors, taking into account time intervals for the definition of control, preventive maintenance and launch timing depending on the sequence of references produced and on the processor used.

The implementation and test of IDSS are in progress. It can be used within the horizon of a very few weeks (at the operational level) or more (at the tactical level, in the framework of spot market negotiations). Later, we aim to insert it in a system of constraints negotiations between fertilizer plant and the upstream units (acid plants) and downstream units (storage, conveyors, boat loading) in the phosphoric supply chain.

References

1. Giard, V., Ahlam, E., Essadi, I., Fénies, P., Fontane, F.: A DSS approach for heterogeneous parallel machines scheduling with due windows, processor-&-sequence-dependent setup and proximate supply chain constraints. Cahier de recherche 375, Paris Dauphine (2015)
2. Giard, V., Azzamouri, A., &.Essadi, I.: A DSS approach for heterogeneous parallel machines scheduling with due windows, processor-&-sequence-dependent setup and availability constraints. In : Information system Logistics and Supply Chain, Bordeaux, pp.1-8 (2016)
3. Gorry, G., Morton, M.: A framework for management information systems. Sloan Management Review 30(3), 49-61 (1989)
4. Keen, P., Scott Morton, M.: Decision Support Systems: An Organizational Perspective. Addison-Wesley Publishing, Reading (1978)
5. Shim, J., Warkentin, M., Courtney, J., Power, D., Sharda, R., & Carlsson, C.: Past, present, and future of decision support technology. Decision Support Systems 33, 111 –126 (2002)
6. Eom, S., Kim, E. .: A Survey of Decision Support System Applications. Journal of the Operational Research Society 57(11), 1264-1278 (2006)
7. Bhargava, H. K., Power, D. J., Sun, D.: Progress in Web-based decision support technologies. Decision Support Systems 43(3), 1083-1095 (2007)
8. Power, D. J., Sharda, R.: Model-Driven Decision Support Systems: Concepts and Research Directions. Decision Support Systems 43(3), 1044–1061 (2007)
9. Maturana, S., Ferrer, J., Barañao, F.: Design and implementation of an optimization-based decision support system generator. European Journal of Operational Research 145(1), 170-183 (2004)
10. Olteanu, A.: A Decision Support System (DSS) for Project Management in the Biodiesel Industry. Informatica Economică 15(4), 189-202 (2011)
11. Heilala, J., Montonen, J., Järvinen, P., Kivikunnas, S., Maantila, M., Sillanpää, J., & Jokinen, T.: Developing simulation-based decision support systems for customer-driven

manufacturing operation planning. In : Winter Simulation Conference (WSC), Baltimore, pp.3363-3375 (2010)

12. Krishnaiyer, K., Chen, F.: Web-based Visual Decision Support System (WVDSS) for letter shop. Robotics and Computer-Integrated Manufacturing (2015)
13. Sharma, R., Chand, N., Sharma, V., Yadav, D.: Decision support system for operation, scheduling and optimization of hydro power plant in Jammu and Kashmir region. Renewable and Sustainable Energy Reviews 43, 1099-1113 (2015)
14. Keskinocak, P., Wu, F., Goodwin, R., Murthy, S., Akkiraju, R., Kumaran, S., Derebail, A.: Scheduling solutions for the paper industry. Operations Research 50(2), 249-259 (2002)
15. Freed, T., Doerr, K., & Chang, T.: In-house development of scheduling decision support systems: case study for scheduling semiconductor device test operations. International Journal of Production Research 45(21), 5075–5093 (2007)
16. Galasso, F., Mercé, C., & Grabot, B.: Decision support framework for supply chain planning with flexible demand. International Journal of Production Research 47(2), 455-478 (2009)
17. Krishna, L., Janardhan, G., Rao, C.: Web integrated decision support system for machine scheduling and inventory management. IUP Journal of Operations Management 8(1), 35 (2009)
18. Tang, L., Liu, G.: A mathematical programming model and solution for scheduling production orders in Shanghai Baoshan Iron and Steel Complex. European Journal of Operational Research 182(3), 1453-1468 (2007)
19. Shirazi, B., Mahdavi, I., & Solimanpur, M.: Intelligent decision support system for the adaptive control of a flexible manufacturing system with machine and tool flexibility. International Journal of Production Research 50(12), 3288-3314 (2012)
20. FICO: Robust Optimization With Xpress. In: Fico. (Accessed June 22, 2015) Available at: https://community.fico.com/docs/DOC-1818