# OptiView – A Powerful and Flexible Decision Tool Optimising Space Allocation in Shipyard Workshops

**Jean-David Caprace**, University of Liege (aspirant FNRS), Liège/Belgium, jd.caprace@ulg.ac.be
**Frederic Bair**, University of Liege, Liège/Belgium, f.bair@ulg.ac.be
**Nicolas Losseau**, University of Liege, Liège/Belgium, n.losseau@ulg.ac.be
**Renaud Warnotte**, University of Liege, Liège/Belgium, renaud.warnotte@ulg.ac.be
**Philippe Rigo**, University of Liege (FNRS), Liège/Belgium, ph.rigo@ulg.ac.be

**Abstract**

*This paper presents new developments to maximize the number of ship blocks and ship sections produced in various workshops of shipyards during a certain time window. The visualization tool OptiView® was developed to support users to improve the space utilization and workshop productivity. The software is coupled with a heuristic optimisation solver. The paper describes the approach to the space allocation problem and gives three application examples.*

## 1 Introduction

The high complexity of ship production, due to the interaction of many different disciplines (hull construction, electricity, fluids, interior fitting, propulsion, etc.) requires an intensive design and a detailed production planning where most of the tasks are carried out in parallel. In order to obtain the best quality, the lowest price, and the shortest manufacturing lead time, it is necessary to increase the number of simultaneous tasks, *Hughes et al. (1994).*

Today, shipyards adapt their design method in order to increase the number of these simultaneous tasks with the use of more and more structural blocks. Traditionally, the majority of the design decisions are taken based on experience and opinion of the designers. These decisions have a strong influence on production costs, but also subsequent performance of the ship during its life.

Fig.1: Flensburger Schiffbau Gesellschaft shipyard (Germany)

Fig.2: Uljanik shipyard in Pula (Croatia)

The assembly of big elements involves necessarily available space area within the fabrication workshop in order to perform the production. The limited space available in a shipyard, Fig.1 and Fig.2, and the relatively large size of blocks and sections force the planners to optimise the utilization of the available surface within the workshops and storage parks. To reach this objective, the idea was to develop a decision tool that helps the user to improve the space utilization and thus to improve the productivity of the workshop. The visualization tool OptiView® has been developed to satisfy these requests. Coupled with a heuristic optimisation solver, the software becomes a very helpful and powerful tool. One of the most important features of the software is its user friendliness and its easy adaptation to any workshop with space allocation problems.

## 2 Space allocation issue

The goal of this study was to establish a methodology and a tool to solve a common problem in shipyards: the space allocation problem. The space allocation difficulty looks like a nesting problem – cutting different shape in a metal sheet and try to minimize the material loss. However we have here a scheduling issue: we want to build the maximum number of pieces in a workshop with a space constraint, but also with a time constraint on each item. Block height is also important because, sometimes, blocks have to be evacuated by a crane bridge above others blocks. Our problem is thus much more complex than a simple 2D scantling problem: we have three geometric dimensions and an additional time dimension.

A complete tool has been developed in order to solve that problem. It contains a visualization tool and an optimisation tool. Target workshops are mainly assembly halls where huge blocks and sections of ships are assembled just before they are due to be welded in the dry dock. Nevertheless, the tool can be easily adapted to other workshops. In shipyard workshops, space is the most critical factor; all blocks have to be built at a fixed date and this imposes a good scheduling to respect all constraints. Each block requires a certain surface of the hall ground during a certain time; the time required to build a block depends on the number of workers dedicated to this particular block.

The objective of this study is to offer a decision tool to the person in charge of the planning (called "the user" here) to assist him in utilizing efficiently the surface available in a workshop thanks to:

- The automatic allocation of the activities (blocks, section, panels, etc.) in the workshops;
- The minimization of the surface lost on the ground;
- Long-term and day-to-day simulations of how a delay impacts the global planning;
- The generation and processing of the data (generation of allocation plans, display of labour graphics, management of the industrial calendar, etc.).

This tool should thus provide planning proposals, i.e. a location and a starting day for each block. Unfortunately, it may happen that the available surface in the assembly hall is not sufficient to produce the entire set of blocks. The tool should then try to help the user to take the most efficient decision.

For simplification reasons, no details will be taken into account regarding the production processes. We will also assume that blocks have their final shape during the assembling process. We do not take into account the successive assembly stages. In addition, a block is considered as parallelepiped. Many blocks are indeed almost parallelepipeds and other shapes could be considered using the same optimisation technique. Dealing with simple data is more convenient, and we believe that a decision tool is only efficient if it keeps things easy to use, even if complex methods are used to solve the problem. Indeed, the software would lose part of its power and efficiency if the time needed to prepare the data becomes excessive.

## 3 Description of the software architecture

### 3.1 Data flow of the software

Block data (dimensions, ready date, due date, etc.) are stored in the shipyard database, supplied automatically by other software. Optiview® is able to synchronize to the shipyard database and to update it. If some block characteristics are changed the software detects directly these changes and gives an alert to the user.

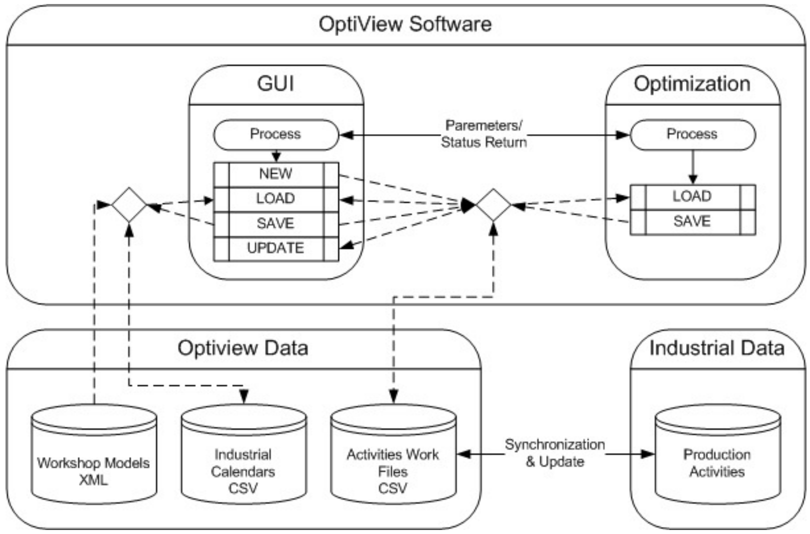


Fig.3: Data flow of the software

### 3.2 Specification and data inputs

#### 3.2.1. Workshop facilities

##### 3.2.1.1 General consideration and data structure

The assembly halls of shipyard contain often more than one working area. Different activities on the block are processed in these areas. Each working area could contain different preferential zones in order to perform the activity in a specific place of the workshop rather than another.

> Workshop
> > Working areas
> > > Preferential zones

We do not care about details in the assembly process, irrelevant for this application, but only about the surface, the time and the workforce needed for the production of the blocks. The major information required about the assembly halls is:

- The available space of the working areas (length, breadth, height) inside workshop;
- The crane capacities (maximum load, height under the hook);
- The definition of preferential zones inside the workshop (length, breadth, height, type of work, etc.);
- The position of the doors;
- The industrial calendar (working days for each ship) – A special tool was developed to manage the days on and the days off (holidays, week-ends, maintenance, etc.);
- The personal availability over time.

It is imperative to know the location of the doors in the assembly hall and the crane bridge height. Indeed, it may happen that a particular block cannot be taken out because there are other high blocks on its way to the door and that the height of the crane bridge may not be sufficient to pass over them. If blocks are too heavy for the crane bridge they need to be driven out by a skid platform. In this case, no block at all should remain on the way and supports for blocks have to be elevated in order to let the skid platform get under the block.

#### 3.2.1.2 Constraints and assumptions

The basic assumptions that we have seen are the following ones:

- Once a block is placed into the workshop, it cannot be moved until all processes on the block are finished.
- The blocks cannot overlap each other.
- All blocks will have their final shape during the whole assembling process.

#### 3.2.1.3 File format

All data related to the workshop facilities are stored in XML file format in order to improve the flexibility of the software. The reading by the XML interpreter of OptiView® file will automatically build the model in memory and allows using it directly in the program.

Space allocation problems in shipyards are not limited to assembly halls. We therefore created a Graphic User Interface (GUI) able to read the parameters of any workshop.

### 3.2.2. Production activities (blocks, sections, etc.)

#### 3.2.2.1 General consideration and data structure

Basically, the input data of the software may be summarized as a list of "activities". Each activity represents a certain work to be done on a particular block. Hence, the following information is required:

- Description of the block (block identification, ship identification, comments, etc.).
- Prismatic dimensions of each block (length, breadth, height, and related spaces allocated to movements around the blocks). Blocks are considered as parallelepipeds. The major reason for this assumption is that this data is very easily available; it is easier to deal with basic shapes and their representations on a surface are more easily interpretable. This idea does not affect drastically the results since many blocks have indeed a (almost) parallelepiped shape. For accessibility and security reasons, a certain distance may be required between nearby blocks in the assembly hall. Therefore, a margin can be added to the length and to the width. These parameters may also be used to correct manually the dimensions of a block, particularly if the shape of the block is not a parallelepiped.
- Processing time evaluation dealing with two aspects: The total amount of workforce needed for each block and the duration of work. At this stage of the planning, a precise processing time cannot be furnished; therefore the processing time has to be estimated. An estimation of the total amount of man-time needed is available, thus the processing time is computed by dividing this man-time by the available number of workers. The user may need to adjust this time manually (because he is able to change the number of workers during a certain period, or because he knows that the estimates are too high). The workload assessments become more precise over time. In addition to the processing time of an activity, some time may be required to prepare the appropriate surface and build up supports for blocks or to dismantle them. This work has no effect on the start and the end date. Therefore it has to be taken into account separately.
- Dates of production start of each block. In this case, we use the earliest start date (earliest date at which the production can start) and the latest end date (the date at which the activity has to be finished), Fig.4;
- Target date – this option allows the user to give a preferential start date for the optimisation module. If this date cannot be reached by the optimiser. The trend will be to approach it as best as possible.
- Group of blocks – several blocks can be grouped so that the optimiser will position them as if they were welded together. The advantage is that we can simulate the impact of the

production of blocks nearby similar ones. Thus the optimisation module takes into account a group of blocks as a huge block. A snap tool was implemented in order to link several blocks together.

- Subcontractor possibility – this parameter indicates if the block can be subcontracted. During optimisation, these blocks will be preferentially selected to be produced in other workshops if the assembly area is overloaded.
- Preferential zone – This field indicates the zone in which it is preferable to produce the block.
- Ship Slice – This fields indicates the slice of the ship which it belongs.
- Fictitious block – this Boolean parameter indicates if the block is fictitious. This option gives the possibility to introduce zones temporarily reserved for activities different from block mounting operations (e.g. storing the ship engines on the pre-assembly area). Fictitious blocks are fixed for the optimiser, they are only used to reduce the available space during a certain time window.

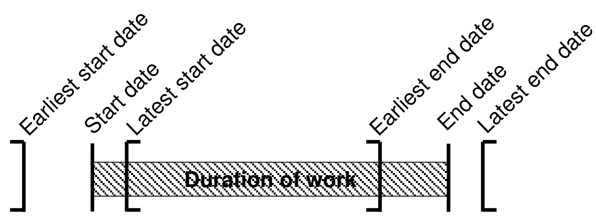

Fig.4: Date and duration of work

#### 3.2.2.2 File format

The data related to the activities (blocks, sections, etc.) are stored in CSV file format. Each line of this file represents an activity. A reading loop allows to load and store all blocks in memory.

### 3.2.3. Optimisation variable and objective function

The goal of this study was to find out automatically the best planning for a workshop. In order to reach this aim, the optimisation module must place the blocks in the best way on the available surface, and gives a correct allocation for the biggest amount of blocks. The optimisation objective is to maximize the number of building blocks produced in the factory during a certain time window.

To achieve this, the optimisation module can adjust the following variables:

- Block position inside the working area (X and Y);
- Block orientation – Four dispositions are available since the blocks have a rectangular shape;
- Starting date of the work, Fig.4;
- Working area – In some cases, the values may be restricted to a subset of the sections, depending on block characteristics.

In addition, each variable can be fixed so that the optimiser does not have the opportunity to modify the value. For example, this feature is used in order to define the daily production situation of the workshop (real block position inside the workshop).

## 3.3 Graphical User Interface (GUI)

The intuitive and ergonomic GUI, developed in Java ("platform independent"), supports an easy adaptation of the software to any workshop with space allocation problems. The main frame of the GUI is divided into two windows. One is the spatial view of the workshop (top view of the workshop on a given date; see on left side of Fig.5), the other is the timeline view (top view of the workshop with a dimension in space and a dimension in time; see on right side of Fig.5). These two frames interact in order to display the situation of the workshop at different dates by the dragging of the daily line in the timeline view.

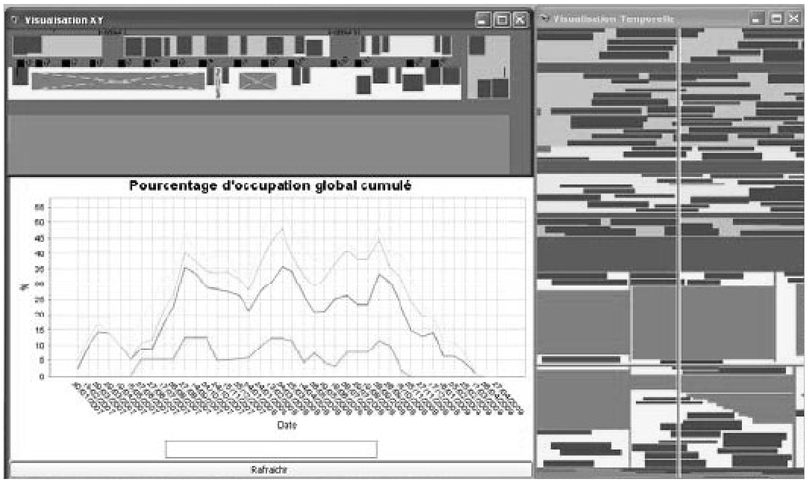


Fig.5: Main frame of OptiView®

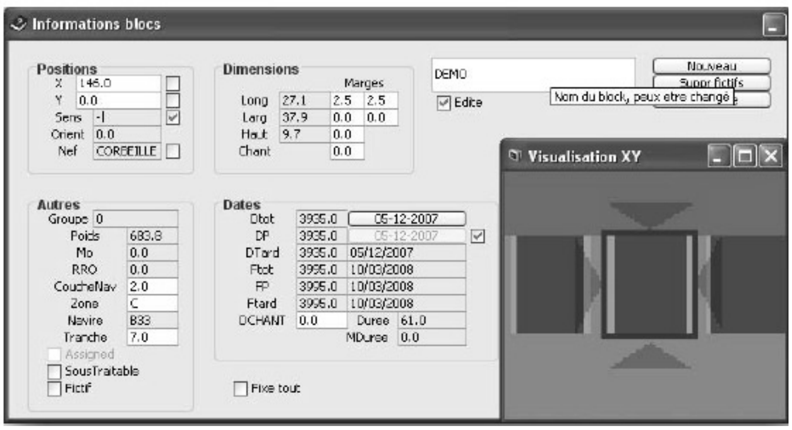


Fig.6: General data frame

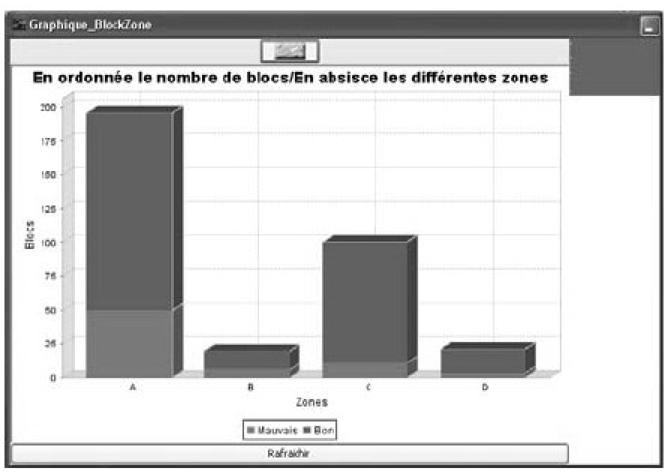


Fig.7: Graphics assignment of preferential areas

### 3.3.1. Spatial view of the workshop

This frame simply shows a top overview of the workshop at a selected date. The user can move blocks (drag and drop) inside space (X and Y) for the day selected. Also, double clicking on a block will select the entire block; display and edit its information into the general data frame (see Fig.6). A hierarchical tree structure with some filtering facilities is also available within the software.

Different movements for the blocks exist:

- Normal moves: Allow moves inside a working area for a single block.
- Moves from working area to working area: Allow block to change the working area.
- Group moves: Allow moves for group.
- Rotations: Allow user to rotate the block.

A colour code was implemented in order to represent the different ships in process. For instance, the dark blue colour is assigned for "normal" blocks; light grey colour for accessibility spaces, brown for fixed blocks, orange for fictitious blocks, pink for blocks going out, green for group of blocks, etc. A management tool for the displaying options (colours, legends, font size, filtering by colours, etc.) is also implemented.

### 3.3.2. Timeline view of the workshop

The timeline frame shows an overview of each working area with an axis for the time (vertical axis on Fig.5) and another one for a dimension (X or Y – horizontal axis on Fig.5). The user can move blocks along the temporal and spatial (X or Y) dimension by a simple drag and drop. However the displacement of the blocks is limited between the earliest start date and the latest end date. The vertical line can be placed on a precise day of timeline and shows the state of all areas at this date.

### 3.3.3. Detection of overlaps

The tool detects all the collisions and overlaps between the blocks into the spatial view and timeline view. To help the user to have an overall view of the model, we built a hierarchical display including all the overlaps of the blocks, groups of blocks, blocks outside working area, blocks inside inadequate preferential zone, blocks with incorrect dates (when there is not enough time to produce the block between the start date and the end date), blocks too heavy to be carried by the crane bridge, etc. The tree is sorted by problematic dates so that the user can easily correct the problems.

If no feasible solution is found by the optimiser module, the user can directly see which blocks lead to problems and then choose an adapted solution. The user can identify the production surface utilization problems that may happen for the actual data (basically the fact that not all blocks can be produced in time). He may then for example raise the workforce availability or subcontract some blocks. It is therefore necessary to analyse the problems and to adjust manually some of the input.

### 3.3.4. Post-processing

After an optimisation, all output data can be easily treated and different results can be obtained:

- Generation of space allocation plan and scheduling with PostScript, PDF, HPGL and DXF format.
- Graphics display management (printing and possible export with PDF and CSV format), Fig.7: Space allocation; Activities number per production area (blocks, panels, etc); Workload; Use of preferential areas; etc.

## 3.4 Optimisation module

The optimisation objective is to maximize the number of blocks mounted in the factory during a certain time window. The problem of placing blocks into workshops over time is NP-hard because of the exponential explosion of the solution space, *Garey and Johnson (1979)*. I.e. an “optimal solution” for a large application cannot be found within reasonable computing times. Therefore, the user should accept obtaining instead a “nearly optimum” solution. An efficient tool should make use of modern heuristics to find such results within short computing time.

The space allocation issue is very similar to the three-dimensional bin packing problem (3D-BPP) which consists in packing orthogonally all items into a minimum of bins. Our developments are based on the practical and efficient method of *Faroe et al. (2003)* for that problem. The method is guided on a global-local search sequence, which has been numerically validated tested for several standard test cases. The approach offers large flexibility; it can be adapted to various different constraints and fits perfectly to our application. The algorithm outperformed other approaches and solutions for our industrial problems were found within seconds. More details about the optimiser tool are given in *Langer et al. (2005)*, *Voudouris and Tsang (1999), Scholl et al. (1997).*

## 4 Application to three different workshops

### 4.1 Introduction

We will present applications of OptiView® to three different workshops. Each workshop has its own characteristics. The two first workshops are pre-assembly workshops, where blocks are built before they are due to be welded in the pre-assembly area or directly in the dry dock. The main difference between these both workshops is the number of working areas and the total capacity of the cranes: 120 tons for the first one and 180 tons for the second one. The third workshop is a space-limited outside area where blocks are welded together just before being placed on the ship in the dry dock. Thanks to the flexibility of the GUI and the optimiser, it was easy to adapt the tool for each workshop.

### 4.2 Pre-assembly workshop (120 tons)

This workshop has 8 different working areas, Fig.8, but no preferential area. The only restrictions are thus the space available, the time window to respect, and the exit constraints. Time to prepare the appropriate surface and build up areas of work (positioning of support) or to dismantle them is considered as a parameter of the blocks. The “bin” in Fig.8 is a supplementary area where blocks are placed if they have no allocation. This is a fictive area. Manually the user can drag and drop a non-allocated block into an empty working area. Even if an area seems to be void, another block may have been allocated at this place some days later. The software detects automatically such conflicts: critical blocks will directly change colour. For this workshop we added a tool to see “true” empty space: if we want to put a block at a specific date, the software shows blocks present at this date but also blocks that will appear between the start and the end date of the selected block.

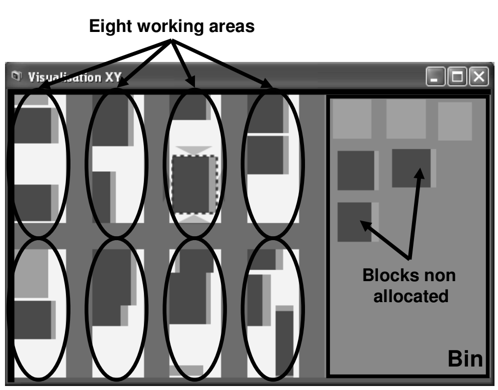

Fig.8: Pre-assembly workshop (120 tons)

A double click on any block gives directly all information about the block, Fig.6. Multi-selection is also available to change the value of any parameters for several blocks. A powerful tool of search and selection is included in the software: we can e.g. select all blocks that have a start date included in a given range of dates, or blocks having a certain width, etc. Other useful functionalities are the possibility to have in real time the list of blocks in each working area and a tool to directly see a list of blocks in collision, Fig.9.

In practice, allocating blocks with the optimisation tool ensures that we will not have any collision. The optimiser tool gives always a feasible solution. Nevertheless the tool of collision detection is very useful for manual planning and when correcting the planning given by the optimisation. When scheduling over several months, seeing directly collisions without the help of such tool is difficult indeed.

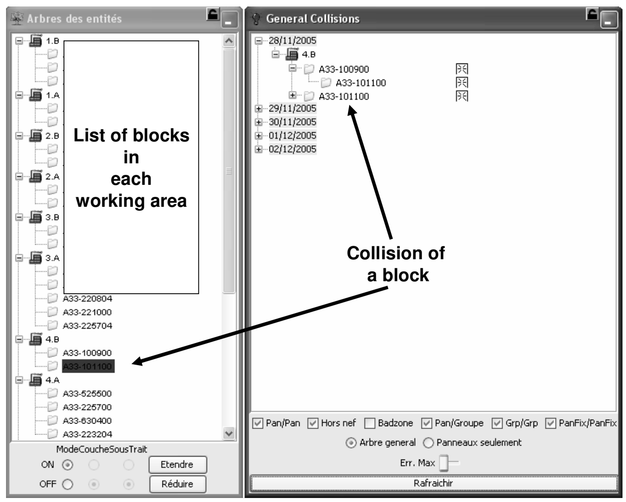


Fig.9: List of blocks in each working area and detection of collisions

### 4.3 Pre-assembly workshop (180 tons)

This workshop is similar to the previous one, but it contains only four working areas, Fig.10. Exit ways are also different and the crane bridge has different height and weight capacity. Huge blocks cannot be evacuated by the crane of the workshop; they have to be evacuated by a skid platform which runs in the centre of the workshop. These blocks should thus be placed close the central path.

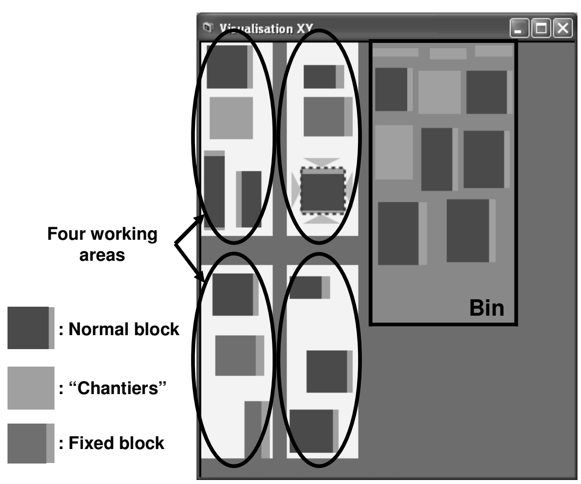


Fig.10: View of the Pre Assembly Workshop (180 tons)

Fixed blocks, shown in Fig.10, cannot be moved by the optimiser. This functionality is useful to represent the starting state of the workshop: Blocks already in the workshop cannot be moved by the optimisation!

## 4.4 Pre-assembly area

The modelling of this area contains the pre-assembly area itself and the dry dock area, Fig.11. If the pre-assembly Area is saturated, some blocks can be put in the dry dock. Before the optimisation, the user has the possibility to select which working area is available, Fig.12. In this case, we first try to maximise the number of blocks placed in the four working areas (the dry dock is prohibited) and then we launch a second optimisation with the dry dock “open”.

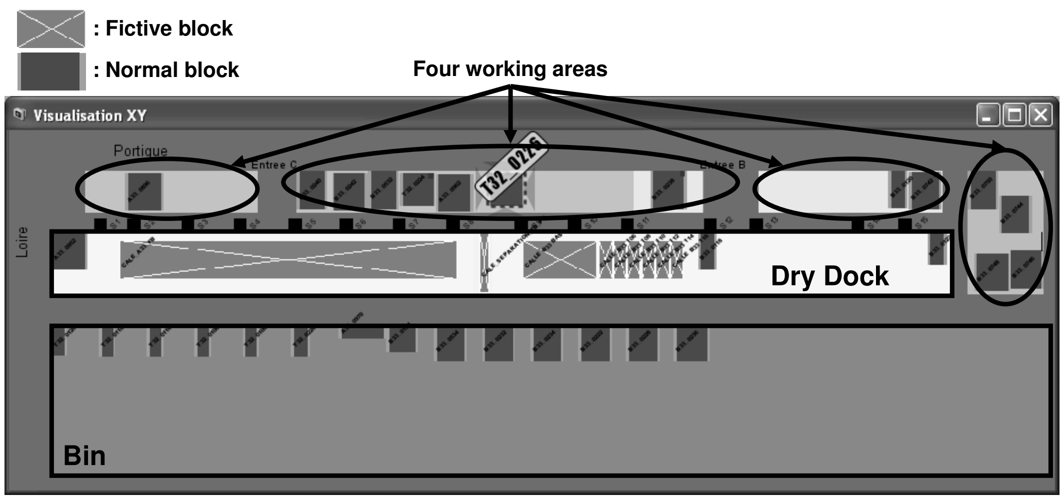


Fig.11: View of the Pre Assembly Area

In Fig.12, we see different parameters for the optimisation. The main parameter is the total calculation time. Internal parameters of the optimisation can also be modified in order to obtain better results. Some advanced parameters are:

- *Zone probability* and *slice probability*: it can be desirable to keep some blocks stay together. For those adjacent on the ship, we economize the use of the crane if they are built close together in the assembly area. In the same way, the movement time of workers in the same portion of the ship decreases. For each block, we can define a *preferential zone* or a *ship slice*. If we activate the parameter *zone probability* or *slice probability* the optimiser tries to put blocks of the same family as close as possible;
- With these parameters we can define the priority for the optimiser to respect this constraint. For value zero, the optimiser does not take into account this allocation constraint. For value one, the optimiser takes into account the restriction when it is possible; the first priority of the optimiser is still to put the maximum number of blocks. Thus this constraint is not always completely respected. Intermediate values are also available.

In this workshop we do not use the earliest start date and the latest end date. The start date is fixed directly by a database fulfilled by the planner and/or software. As for other workshops, the assembling duration is also fixed. The only degree of freedom is the spatial position of the blocks and the choice of the working area.

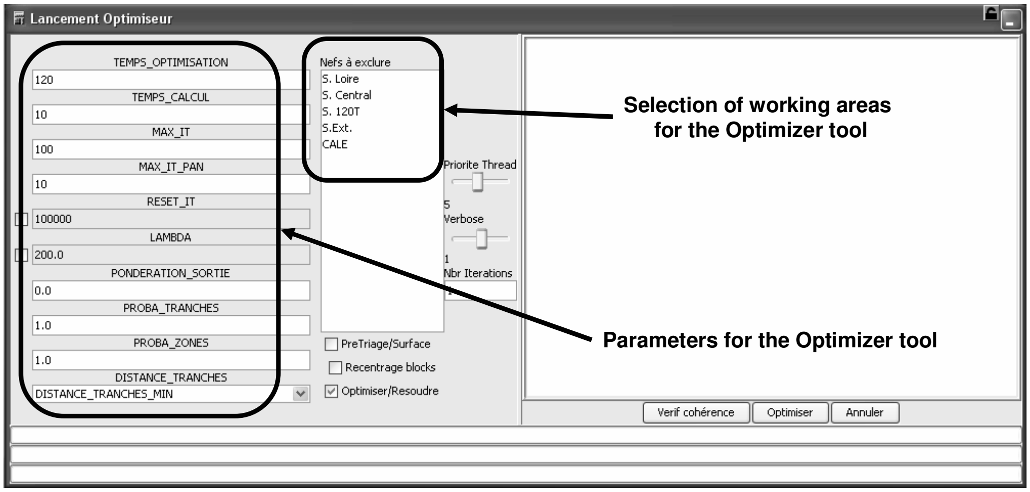


Fig.12: Main parameters of the optimiser tool

It is irrelevant to consider processing times over normal calendar days (as some days may be off). For this purpose, we developed functions to convert "normal days" to "open days". But the pre-assembly area has another important particularity. For the previous workshops, we only have one industrial calendar. Here, for practical reasons, we could have one industrial calendar for each ship: each block belonging to the same ship has the same calendar, but it is not necessarily the case for blocks of different ships. The software can manage different industrial calendars. However, there is an important constraint: if the start date is a variable – as for the two previous workshops – we can only have one industrial calendar.

Finally, a 3D interface (OpenGL) allows a better view of the state of the workshop to the user, Fig.13.

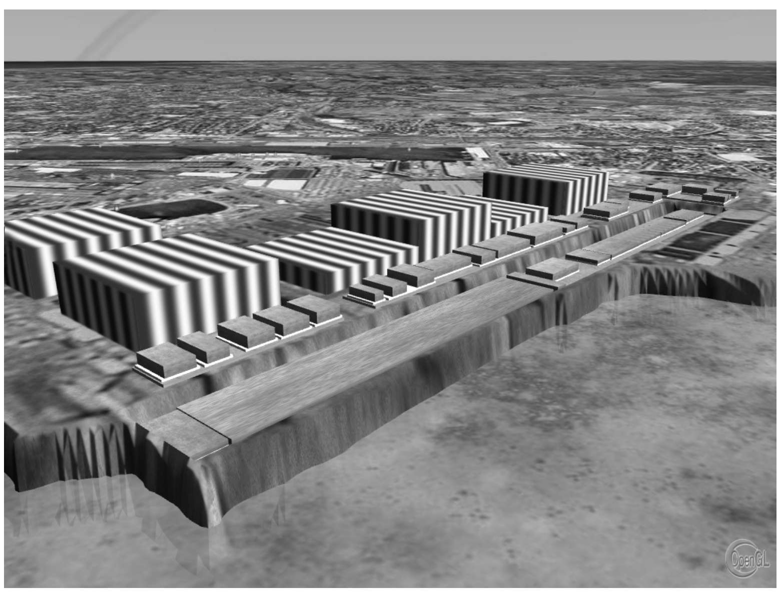

Fig.13: 3D view of the Pre Assembly Area

## 5 Conclusion

The software allows more efficient scheduling for the shipyard. The planner can now test different alternatives and rapidly modify the scheduling to find the best one. But the preparation and verification of data for the simulation is still a major stage to ensure the reliability of the results.

Gains obtained for the shipyard are substantial. The workshop productivity by using the OptiView® tool is increased due to less time needed for the scheduling and better space utilization.

The main advantage of the software is its great flexibility. The software can be easily adapted to different workshops. In shipbuilding, space allocation problems occur in most workshops. We have shown three assembly workshops, but the software could also be used for painting halls, storing areas, etc.

The software is easy to use and intuitive with a lot of useful functionalities: advanced search tool, graphical visualization of results, easy access to each parameter for the user (colour, text, etc.), generation of space allocation plans and scheduling with PostScript, PDF, HPGL and DXF format, etc. Rather than to work in a 2D view, it is also possible to switch into a 3D mode and to navigate easily in this mode.

As a conclusion, a complete and mature software tool has been developed with a large range of application fields in shipbuilding industry, but potentially also in other industries.

**Acknowledgement**

We thank University of Liege, AKER YARDS FRANCE (Saint-Nazaire), for the collaboration within sub-project II.1 of the IP INTERSHIP Project ($6^{th}$ Framework) and the MARSTRUCT - Network of Excellence on Marine Structures.

**References**

FAROE, O.; PISINGER, D.; ZACHARIASEN, M. (2003), *Guided local search for the three-dimensional bin-packing problem*, INFORMS Journal on Computing, Vol.15, No. 3, pp.267-283

GAREY, M.R.; JOHNSON, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York

HUGHES, O.F. et al. (1994), *Applied computer aided design*, Report V.5 of Committee to ISSC, Inst. Marine Dynamics, Newfoundland

LANGER, Y.; BAY, M.; CRAMA, Y.; BAIR, F.; CAPRACE, J.; RIGO, P. (2005), *Optimization of surface utilization using heuristic approaches*, COMPIT'05, Hamburg, pp.419-425

SCHOLL, A.; KLEIN, R.; JURGENS, C. (1997), *BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem*, Computers & Operations Research 24, pp.627-645

VOUDOURIS C.; TSANG E. (1999), *Guided local search and its application to the travelling salesman problem*, European J. Operational Research 113, pp.469-499