SIMULATION-BASED ANALYSIS OF A CONCRETE ARMOUR UNITS MANUFACTURING PLANT FOR THE CONSTRUCTION OF A RUBBLE MOUND BREAKWATER

David del Rio Vilas[a], Diego Crespo Pereira[b], Adolfo Lamas Rodriguez[c], Alejandro Garcia del Valle[d]

[a], [b], [c], [d]Grupo Integrado de Ingenieria, University of A Coruna

[a]daviddelrio@udc.es, [b]dcrespo@udc.es, [c]alamas@udc.es, [d]agvalle@udc.es

ABSTRACT

New port facilities at Punta Langosteira, on the northwest coast of Spain, involve the construction of a large rubble mound breakwater that requires thousands of concrete armour units that will have to be supplied as the construction project progresses. For that reason, a provisory plant has been built in situ where the concrete blocks are cast, transported and stacked. Due to the implementation of a DGPS system, the blocks life cycle can be traced and an assessment of strategies for minimizing the number of blocks moves became plausible. In this paper, we describe the operational analysis for the determination of a comprehensive and simple set of rules which may result in a cost-effective operation of the concrete blocks plant whilst meeting the procedural constraints. To do so, a simulation-based approach was adopted both using the commercial tool Delmia QUEST and developing our own ad hoc Java simulator.

Keywords: simulation, optimization, supply chain modelling, manufacturing, stacking yard

1. INTRODUCTION

The project of the New Port Facilities in Punta Langosteira on the northwest coast of Spain is one of the most important building projects under construction in Europe, not only due to its size and complexity but also to its impact on the whole economy of the region. It has an initial budget of more than €429 million partly financed by EU Cohesion funds and by the European Investment Bank. A Joint Venture made up of the Spanish companies Dragados, Sato, Copasa and DRACE was chosen to accomplish this work, which started in April 2005 and is expected to finish by September 2011, although the outer main breakwater will have to be finished by the end of 2009. When finished, the main breakwater will have a total length of 3.4 km and a maximum height of 65m. (Autoridad Portuaria A Coruna 2008).

Prosermar Ingeniería S.L. is the firm that has supplied the DGPS system for tracing the concrete blocks' life cycle and responsible for determining a set of operational policies that may lead to an enhanced plant operation. It is in the framework of this job that the work reported here has been carried out.

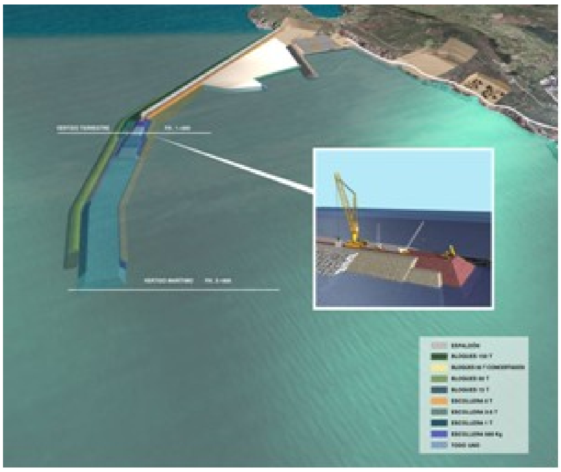

Figure 1: A Schematic General View of the Rubble Mound Breakwater. Layers of Concrete Blocks are shown.

2. THE PLANT

A plant has been built in situ to supply the amount and type of concrete blocks that the construction project requires. Blocks of 15, 50, 70 and 150 ton are manufactured and stacked by Rail Mounted Gantry Cranes in their corresponding yards. As a matter of fact, more than twenty three thousand 150 ton blocks, with a cost nearly €5,000 each will have to be created, transported and stored in this plant.

The plant is divided into two main areas dedicated to the production and stocking processes respectively. There are two stacking yards, one devoted to the 15 ton blocks. This yard is divided into two areas by the casting zone where blocks are cast by means of a continuous pouring system operating around the clock six days a week.

There are six pouring lines in the casting yard on which twenty moulds are moved by a crane following a predefined pouring sequence. Concrete is poured into the shuttering and after six hours it can be removed and moved to the next position. Blocks have to stay ten more hours until they have hardened enough so they can be lifted and transported in pairs by the cranes to their corresponding slots, where they will continue to harden.

Figure 2: A General View of the Plant during its Construction.

Concrete blocks have to meet quality standards related to their curing progression that define their life cycle, as shown in Figure 3. Due to the stochastic nature of this process, they cannot be directly delivered to their final destination in the breakwater and consequently have to be stored. Accordingly, three states are possible for a block, i.e. accepted for final delivery ("Green" Blocks), rejected ("Red" Blocks) and those not yet defined as green or red ("Yellow" Blocks). Besides, the prevailing adverse weather conditions during the winter months make it impossible to carry out maritime operations so that a stock is generated (Stock Phase), whereas during the rest of the year, production and delivery of blocks happen simultaneously (Input-Output Phase).

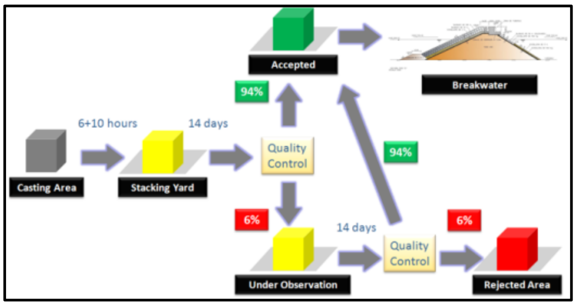

Figure 3: Blocks Life Cycle.

Yard Cranes displacements are monitored and managed by means of a DGPS system so position, identity and state of cranes and blocks can be known. In addition, historical records relative to a previous constructive experience in the nearby port of Ferrol have shown that only 0.36% of the total amount of blocks will not pass the quality tests and consequently will have to be removed. These two key aspects are the foundations on which we have developed our models.

3. THE PROBLEM

Every time a crane has to pick up or drop off a pair of blocks a decision has to be made. In the case of the 15 ton blocks, there are more than 760 candidate slots in the yard where to place them. Besides, they can be stacked on top of each other up to five levels. This case is the most exacting since these blocks present the highest turnover. Consequently we decided to focus our analysis on it. The aim is to minimize the total distance travelled by the cranes while meeting both productive rates and operation constraints.

From a mathematical point of view, this real problem is similar to that of the Stacker Crane Problem (SCP). In the SCP a collection of source-destination pairs (s_i, d_i) is given where for each pair the crane must pick up an object at location s_i and deliver it to location d_i. The goal is to arrange these tasks so as to minimize the time spent by the crane going between tasks, i.e. moving from the destination of one pair to the source of the next one. This can viewed as an Asymmetric Travelling Salesman Problem (ATSP) in which city c_i corresponds to the pair (s_i, d_i) and the distance from c_i to c_j is the metric distance between d_i and s_j. In our specific case, neither sources nor destinations are fixed so the complexity of the problem increases. Additionally, this would be a random dynamic 760-node instance. Even though it were studied as a fixed static case, it would be much larger than most of the instances that still remain unsolved (Gutin 2002). Further research in dynamic routing problems is proposed (Larsen 2000).

Considering this complexity as well as the need of obtaining real solutions, we finally decided to adopt a simulation-based approach by means of which we could evaluate the performance of our proposals and check their operational feasibility. The role of simulation to evaluate alternative management policies is fundamental, especially when the policies are computer generated and the human decision-makers do not have a complete understanding of all their details (Gambardella 2000). Simulation has been employed for supporting decision making processes in manufacturing-oriented supply chain applications (Qiao 2004) as well as in container terminals management. The location problem of containers in port container terminals has been broadly addressed following different approaches using both heuristics and metaheuristics for improving construction methods, but all of them based on specific process peculiarities (Günther 2005).

4. THE SIMULATION

Being aware of the logistic nature of the processes to be modelled, we have used the Delmia QUEST simulator, which allows a good implementation of push-pull policies, queuing logics, and transportation systems while offering an excellent 3D graphical simulation environment on which complex processes can be seen and understood in a very intuitive and practical manner. In fact, one of the most important implications of its utilization has been the possibility for project engineers to visualize the real operation in advance and to anticipate both problems and opportunities. QUEST has been employed to gain more knowledge about the real plant and preliminary information that was used later in the development of our simulator.

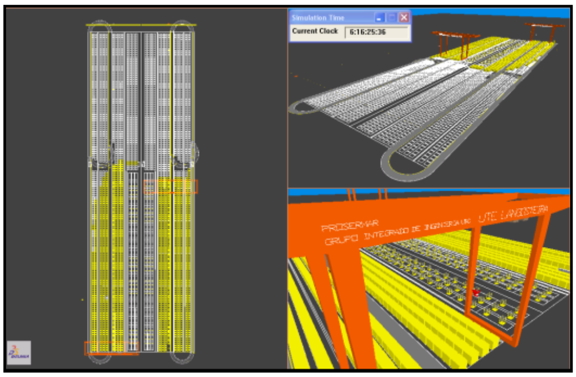

Figure 4: The QUEST Model of the 15ton Yard

We first studied the blocks management in the Stock Phase and then in the Input-Output Phase. The cinematic analogy was achieved by introducing all the geometrical and cinematic data relative to the real elements, i.e. crane dimensions and speeds, blocks dimensions, distance between slots, etc. Models started from CAD information about the real yard. Sources elements represent the cast process by generating three types of parts according to their proportions and following the predetermined sequence. A 760-stack point buffer represents the yard, connected to the sources by means of a Crane AGV System that have been used for modelling the gantry cranes and the logics and points where decisions have to be made. Crane's logics had to be coded in SCL, –a proprietary Simulation Control Language of Delmia– for modelling their kinematics and in order to implement decision rules. Crane movement has been modelled according to a so called "German" operative rather than an "American" one. This means that gantry, cross travel and hoist do not move simultaneously, representing a conservative but safer and more reliable operation from the maintenance point of view.

Connections between elements are required, combining push and pull processes. For example, incoming vehicles for blocks' delivery are modelled by sinks generating pull requests, as well as the buffers for stacking red blocks. Hence, the operational analogy is obtained, and as the simulation can be visualized, the model verification is easier and faster.

An especial effort was made in automating all the processes related to data input and output as well as model geometric definition. This flexibility was obtained by programming BCL scripts –Batch Control Language- with geometric information and characteristic operational parameters rather than using the QUEST GUI. As will be described later, the same intention guided the design of our simulator.

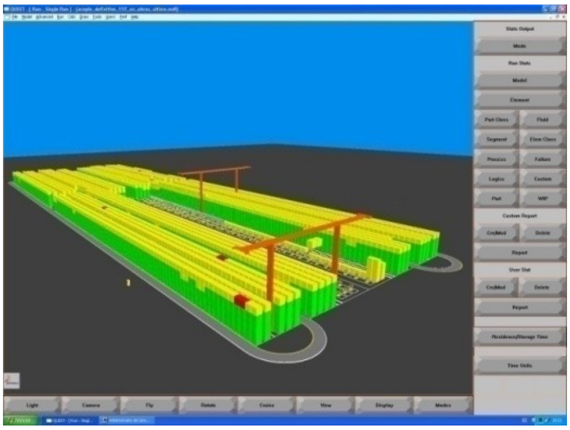

Figure 5: The Yard at the end of the Stocking Phase

Simulation of preliminary models of the Stock Phase was focused on testing a FIFO priority versus a Nearest Neighbour (NN) one when serving sources. Both of them are classical approaches in facing this kind of dynamic problems (Larsen 2000). A basic covering sequence based on maintaining parallelism between sources and stack points visited by the crane was initially proposed. While piling on the region of the yard parallel to the cast area results in an optimum crane performance, results become far from good whenever this parallelism is lost. One solution is to predetermine the maximum distance the crane may travel according to yard's geometry by dedicating farther slots to farther sources and so on. This policy, although simple, has proved to be as good as the NN one for the Stock Phase operation. Even more, we have determined the optimum value for the Stock Ratio in the Stacking Phase –assuming a FIFO pattern when serving sources- by means of the implementation in our simulator of a set of optimal rules. These are basically focused on avoiding those slots that have a lower y-coordinate than the source under consideration whenever a Source-Slot movement has to be done. A value of 86.80 m/block was obtained, which involves only an improvement of 0.3% respect the result derived from the NN procedure.

A Nearest Neighbour rule based on a Euclidean metric was finally adopted when deciding which stack to go to. In Table 1 results relative to different metrics are shown. The yard crane should select the available block corresponding to the pouring order, i.e. should act according to a FIFO policy. The reason is that it only represents less than 2% decrease in performance when compared to the NN operation while it enhances Quality Control and Traceability since it follows the actual cast process. Besides, it has a simpler as well as more robust implementation, in case of a software blackout for instance. But above all, it is less sensitive to the possibility of collapse in the cast area, that is to say, the possibility of overlapping two consecutive pouring processes in the same point. Due to the yard geometry, the NN operation tends to "abandon" the blocks that have been cast in the beginning of the lines, forcing to slow production rate.

Table 1: Stock Ratio (m/block) and Decision Rule in Sources for the NN Policy for Slot Decision in the Stocking Phase.

	FIFO	NN
Euclidean	87.06	85.54
Gantry Distance	87.30	85.93
Manhattan	87.30	85.92
Tchebychev	87.91	87.75

The Rejection Area location for red blocks was also determined. Traditionally, red blocks are piled on an extreme area of the yard so that they do not interfere with other operations. Since the generation of red blocks follows a random pattern, the simulations in QUEST of the stocking phase were first executed to find out whether a centroidal location for red block stacks would be more suitable than the traditional policy. Preliminary results seemed to indicate that both ways are equivalent in terms of distance. Further experimentation in our simulator confirmed this thesis, as will be explained later

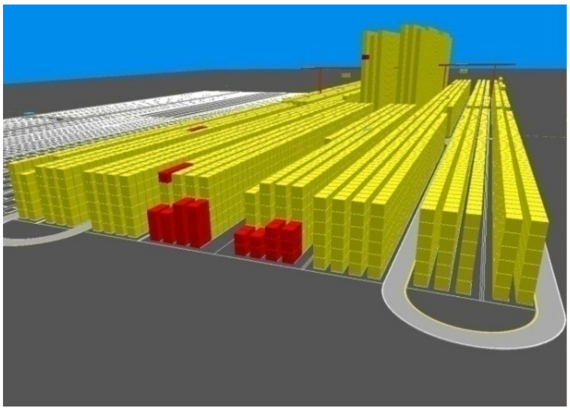

Figure 6: Head Location for Red Blocks. Collapse has happened in Sources

In order to acquire more knowledge and control of the simulation process we developed a Java application, by means of which once we first validated the results obtained in QUEST we could manage simulations in a much faster and detailed way. We could then run long term simulations of 144 working days of the Input-Output model in just a few seconds. Despite the fact of QUEST was extensively used for analyzing the Stock Phase, we could not run the Input-Output QUEST models for more than 20 days of simulation time.

4.1. SIMPA

Our simulator SIMPA - in Spanish, standing for *Simulador de Patios de Apilamiento* - makes use of a combined event-activity oriented simulation approach. The system time evolution consists of an iterative process based on the increment of a time counter by steps corresponding to crane moves. In turn, elements – blocks and sources - are checked for state changes on each step relative to their curing process and the next crane move is decided depending on the rules under consideration. Thus, simulation is event oriented in the sense that crane activities are modelled by starting-ending events. However, blocks casting and curing processes simulation is activity oriented since starting-ending conditions are checked at moments given by crane moves, not by their own activity events.

The underlying software architecture consists of two main modules corresponding to two java packages. One is the system's model module, which contains classes that represent its elements (crane, sources, stacking points and others). The other one is the simulation core module, which contains the main simulation loop and functionality related to the time management of the system and monitoring.

The elements that constitute the model are the following:

1. Crane. The crane is modelled by its position in Cartesian's coordinates and its cinematic properties - gantry, cross travel and hoist speeds - as well as its height.
2. Source. Sources are defined by their position in the ground (given x-y coordinates), casting state (empty, casting or waiting for the block to be moved to the yard), counter of time elapsed since the last casting initiated and properties common to every source (static declared properties) height of the blocks and cycle operational time.
3. Stacking Point. Stacking points are defined by their x-y coordinates, the number of blocks stacked, the list of time counters of the blocks (time passed since the casting of the block), the list of block's states (green, yellow or red) and the general properties to all of them, i.e. time of first and second quality tests and probability of passing them. The stacking point class also defines methods for adding and removing blocks.
4. Delivery Point. It represents a point to deliver a block that has finished its cycle in the yard and is sent to the breakwater (green) or removed (red). Hence, it is used to model the vehicles and the rejection area.
5. Pull Delivery Point. This class is an extension of Delivery Point that models a delivery point which requests a block to be served at constant time intervals. Specifically, it is used to model the vehicles incoming process along the Input-Output phase.
6. Crane Movements. The different types of movements the crane is able to complete are inherited from a super class called *Movement,* making it easier to manage the different actions that can be executed by the crane. The particular types of movements inherited are:
   (a) Source to Stack Point movement. This class is defined for transporting blocks

    that have already been cast and need to be stacked in the yard.
  - (b) Stack Point to Delivery Point movement. This class is defined for the transportation of green or red blocks that have to be delivered.
  - (c) Stack Point to Stack Point movement. This class is defined for reallocation movements.
  - (d) Empty movement. This class is defined to relocate the crane after transporting a block.

The simulation core module contains only one class, called SimCore. Its properties are the time counter of the simulation run, the ending time of simulation and the names of files to monitor the state of the system during simulation. There is a main method that runs the simulation which executes the simulation loop as follows:

1. Initialize time counter to zero.
2. Determine next crane action according to policies.
3. Obtain list of necessary moves to make selected action {mi}.
4. For each move:
   - (a) Calculate move duration.
   - (b) Update time counter.
   - (c) If it is a transportation block move, then:
     - (i) Remove the block from its departing position.
     - (ii) Update transported block time counters.
   - (d) For each source:
     - (i) Update time counters as given by move duration.
     - (ii) If state should change, then update.
   - (e) For each block on each stacking point:
     - (i) Update time counters as given by move duration.
     - (ii) If state should change, then update.
   - (f) If it is a transportation block move, then deliver the block.
5. Save system state into a file.
6. If time counter > time limit, then finish the simulation. Else return to step 2.

Other methods perform certain parts of that loop. The state of every block, the number of green, yellow and red blocks and the distances travelled by the crane are saved into several files at each step.

Results have been measured by defining performance ratios for every process under consideration. Hence, the distance travelled by the gantry per block transported has been the main relative value on which the comparison of policy goodness relied on. This value has been calculated both for the Source-Slot moves –Stock Ratio– and for the Slot-Delivery ones –Delivery Ratio–. Other important values obtained have been the total distance travelled by each crane, the number of reshuffles (movements required when trying to pick up a green block which is under a yellow one), and the maximum set up times for the loading/unloading operations without production collapse, among others.

4.2. The Input-Output Phase Simulation

During the Input-Output Phase, yards A and B alternate in serving vehicles meaning that whenever a yard is receiving blocks from sources the other one is issuing blocks to the breakwater and vice versa. This sequential operation is a simple but effective and inexpensive way of traffic control that field engineers are used to employ. In our work, this procedure was adopted following the real process guidelines. In addition, we proposed that vehicles should always go to the location where the crane has finished its last move. Taking into account the important difference in speed between the crane -2.4 km/hour- and the vehicles -30 km/hour– we think that this is a reasonable proposal.

Three different approaches have been analyzed for this phase. A so called *Spreading Nearest Neighbour* policy (SNN) was first proposed as it implements a conservative criterion based on balancing the yard's occupancy level whilst avoiding reshuffling. The SNN is a modified NN in the sense that it follows a NN pattern but prioritizing the occupancy of empty slots. The closest free slot is chosen as a first option. If this is not possible then the closest pile with a yellow block on top is chosen. Otherwise the closest pile with a green block on top would be selected.

When delivering a pair of blocks to vehicles the yard crane selects the closest pile with a green block on top in that moment. In case the whole top level is yellow, the closest pile with highest green block is chosen. As reallocations become necessary and the spreading idea is kept, yellow blocks are placed on free slots if possible. Then again, blocks are left on top of the closest yellow block whenever the first option is not possible. Red blocks are picked up and delivered to their area at the head of the yard only when the crane is idle during the stocking sequence as they naturally come out from piles.

A *Greedy* algorithm was also studied. In the Greedy algorithm blocks coming from the cast area are placed on top of the closest possible pile regardless future consequences. However, decisions regarding the Slot-Delivery moves and the Red Blocks moves maintain the SNN approach. We also implemented a modified Greedy algorithm named *Greedy No Green* (GNG). The only difference lies on avoiding stacking on top of green blocks when a slot has to be chosen in a source-slot movement.

First, simulations of the Input-Output Phase were executed to confirm the head location for the rejected blocks' area. This was an extra validating group of simulations for the SIMPA that we could compare with QUEST results so that an SNN policy was adopted.

As shown in Table 2, although the central option seems to lead to a better performance, there is not a significant difference between these two options, so it is not interesting a change in the traditional management of red blocks.

Table 2: Stock Ratio Results relative to Head and Central Location for Red Blocks under an SNN Policy (5 vehicles, set up = 0.5 min, 100 simulations per case)

Location	Mean(m/block)
Head	82.46
Center	82.56

Less intensive experiments with the other policies under consideration lead to the same conclusion.

This previous validating scenario led to two series of simulation experiments. The first has been designed for determining the cranes' performance with regard to the policy under consideration, the number of vehicles and the set up time. Results of the stock ratio in meters per block are shown in Figures 7 and 8. Simulations consisted of a series of 100 experiments for every combination of algorithm, number of vehicles and set up time (Table 3). We call set up time a delay that we have introduced before every single loading and unloading move aimed at determining the influence of the crane's acceleration and set up times over global yard performance. According to Project Engineers, it is quite unlikely for a crane to take thirty seconds in fixing/unfixing its hoist on a block in normal operation. This is the reason why this value of time was taken into account in guiding the simulation process.

Table 3: Average Stock Ratio and Success Rate (100 simulations per case).

Policy	Vehicles	Set up	Success	Stock Ratio
GNG	3	0	100%	56.19
	3	0.5	0%	0.00
	4	0	100%	44.28
	4	0.5	100%	34.27
	5	0	100%	21.20
	5	0.5	100%	13.21
	6	0	100%	10.86
	6	0.5	100%	14.22

Policy	Vehicles	Set up	Success	Stock Ratio
Greedy	3	0	100%	55.43
	3	0.5	0%	0.00
	4	0	100%	46.91
	4	0.5	96%	35.39
	5	0	100%	20.74
	5	0.5	100%	12.27
	6	0	100%	10.30
	6	0.5	100%	11.04

Policy	Vehicles	Set up	Success	Stock Ratio
SNN	3	0	100%	93.85
	3	0.5	0%	0.00
	4	0	100%	86.84
	4	0.5	100%	82.18
	5	0	100%	80.77
	5	0.5	100%	81.41
	6	0	100%	83.32
	6	0.5	100%	82.88

One important result is that three vehicles are not enough to cope with such a production rate when set up times are nearly half a minute regardless the policy imposed. Moreover, the larger the number of vehicles to be dispatched by the crane, the better the performance of the Greedy and the GNG rules. An even more interesting result is that there is a certain improvement tendency when the set up time increases.

Both Greedy and GNG policies behave in an almost identical manner confirming that the rule of not stacking on top of green blocks does not provide any significative advantage. On the other hand, and even more stable the SNN policy results in stock ratio values more than four times higher than the Greedy and GNG ones.

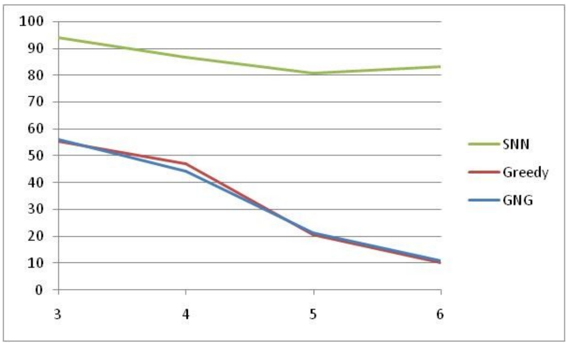

Figure 7: Stock Rate Dependence with Number of Vehicles for each Policy (set up time = 0)

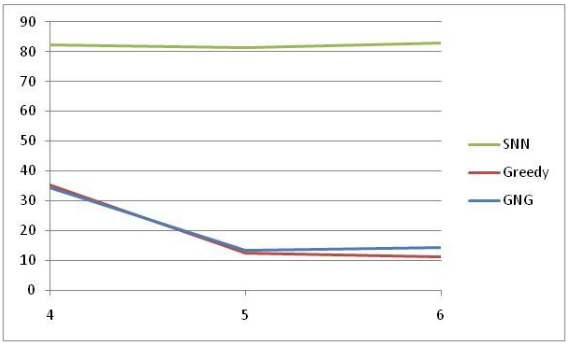

Figure 8: Stock Rate Dependence with Number of Vehicles for each Policy (set up time = 0.5)

Attending to crane performance, a validating operational scenario was proposed (Figure 9) confirming the agreement between production capacity and operational crane capacity. The Greedy policy was simulated for a null set up time and four vehicles model.

Crane capacities correspond to the average results from the capacity test previously described. The crane can cope with the maximum production rate of 114 blocks/day presenting a surplus of 15 blocks/day. This difference justifies the fact that reallocations have to be performed during the stocking sequence. Besides, the crane operates at its maximum capacity during the delivery sequences, reaching its operational ceiling of 153 blocks/day. This is a very important conclusion since more vehicles do not mean a higher degree of progress of the construction project. However, as previously explained, five vehicles would be more desirable than four as the investment in an additional vehicle is lower than the savings in the crane's operational costs.

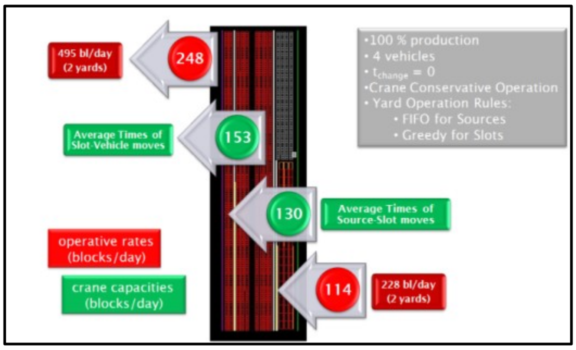

Figure 9: Logistic Input-Output Model: Rates and Capacities.

Results related to this model are presented in the next figures. In Figure 10, the distance travelled by the gantry for a 144 day season is depicted. Simulations showed that a total distance of 700 km is quite likely. The initial slope is higher because the crane has to operate in farther areas of the yard until the steady state is reached.

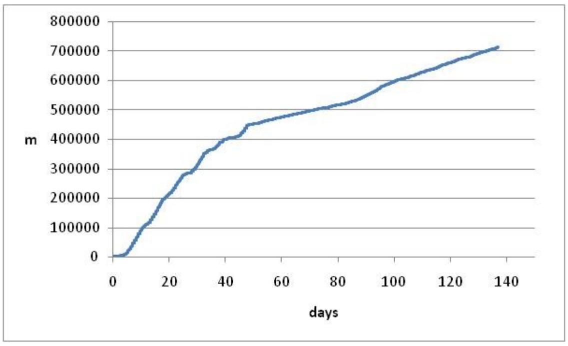

Figure 10: Evolution of Gantry Crane Distance Travelled (6 months, Greedy, 4 vehicles, set up = 0).

The resulting evolution of the Stock Ratio is shown in Figure 11. Then again, the Stock Ratio presents a region of high values as long as the stock generated during the winter months is not delivered. The total number of reallocations and its evolution during a 144 day simulation is presented in Figure 12. Reallocations only appeared after day 80 when the stock is finished. Were the total number of reallocations the criterion on which the selection of policies would rely on, an SNN policy would be definitely chosen since the Greedy one is almost six times higher. The Yard's state evolution can be seen in Figure 13 relative to the number and type of blocks in the yard.

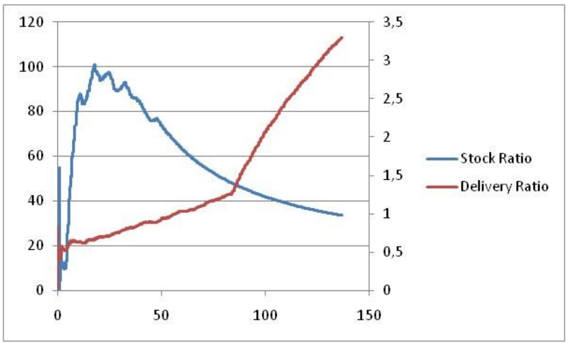

Figure 11: The Stock and Delivery Ratios Evolution (6 months, Greedy, 4 vehicles, set up = 0).

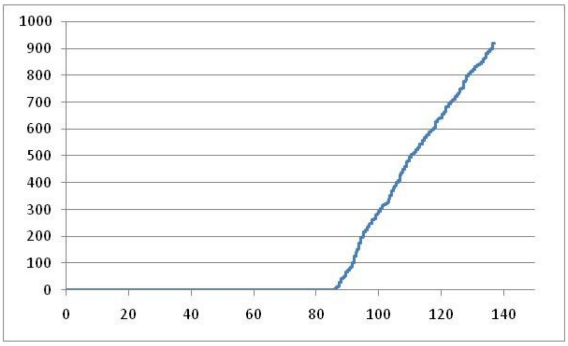

Figure 12: Number of Reallocations (Greedy, 4 vehicles, set up = 0).

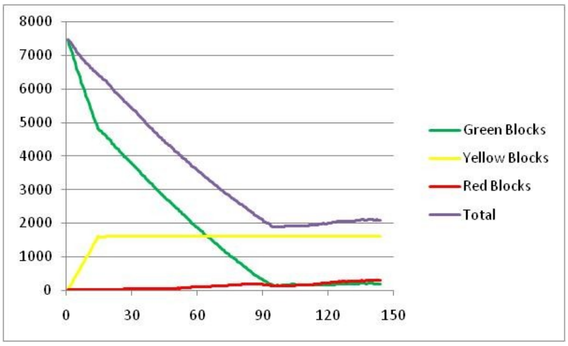

Figure 13: Yard State Evolution for a 6 Month Simulation (Greedy, 4 vehicles, set up = 0).

In order to test the robustness of the Greedy policy a set of 50 simulations was run (Figure 14). The improvement tendency relative to the increment in the set up times is now clearly depicted. For a 6 vehicles operation under a Greedy approach, there is a plateau between 0.4 and 0.9 minutes where the Stock Ratio presents excellent values. These results are not only stable but robust as it is necessary to impose a 1.2 minutes set up time for reducing the success rate from 100% to

90%. Then again, a ratio of 12 m/block is obtained under a Greedy policy operating with 5 vehicles and assuming a set up time of about 36 seconds. An effort in determining real set up times should be done so this value could be optimally adjusted together with appropriate reductions in global crane speed. This would imply an enhanced crane operation both from maintenance and distance minimization approaches. This is a non intuitive or predictable conclusion almost possible to attain by other means but simulation.

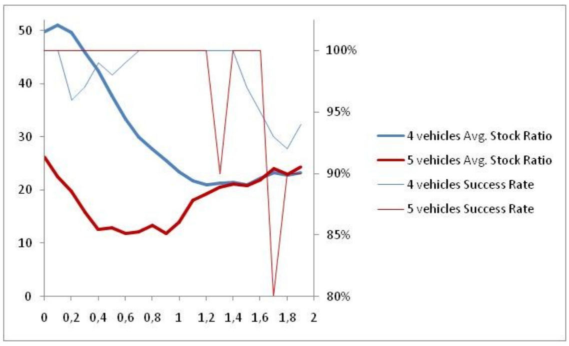

Figure 14: Robustness Test: Stock Ratio and Success Rate for Set Up Times between 0 and 2 minutes (Greedy, 50 simulations per case).

The Greedy proposal has been also analyzed under a very pessimistic value of rejection rate. At present, the actual rejection rate of 6% is being held. However, the typical initial starting up problems in these production environments usually imply a transient state when variables are out of control so it is recommendable to determine whether the actual operation should be maintained or a specific one should be selected and applied during that stage. Moreover, changes during the project related to concrete manufacturing equipment or concrete components –proportions of additives, cement, sand, etc. - may lead to the same situation.

Thus, we decided to test our Greedy algorithm under an increment of 100% in the individual rejection rate value. The proportion of Red Blocks was then fixed to 12% resulting in a total level of 1.44%. This final value involves a 200% increment in the expected number of Red blocks.

Table 4: Average Stock Ratio and Success Rate for 100 simulations per case.

Policy	Vehicles	Set up	Success	Stock Ratio
Greedy	4	0.5	100%	35.12
	5	0.5	100%	12.52

The average Stock Ratio of a series of 100 simulations with a set up time of 0.5 minutes and 4 vehicles was 35.12 m/block. For 5 vehicles, the value was 12.52 m/block. This behaviour is almost coincident to the results obtained for the 0.36% rejection rate. In addition, in both cases the success rate is 100%. Even though the concrete manufacturing process failed, the Greedy policy could cope with such amount of rejected blocks.

Finally, the proposed rules were compared to those previously applied in the construction of a similar breakwater in the Outer Port of Ferrol (traditional operation) and to a random strategy. In Ferrol, it was not possible to follow a strategy based on selecting optimal positions due to the absence of a GPS system. Moreover, at that moment there was not a clear estimation of the blocks' rejection rate, so a conservative stacking operation was adopted. To ensure the traceability of the block's life cycle it was necessary to assign predetermined positions in such a way that piles of blocks were filled sequentially. A pile was not initiated until the previous one was completely filled to prevent red blocks from scattering. This policy is far from optimal. Furthermore, it leads to a poor distance rate because with just a parallel assignment between sources and destinations the rate would be better.

The random policy consists of randomly selecting the next move for the crane at any stage. This way, an efficiency rate measurement of a completely non-controlled system was obtained.

Simulations of the traditional policy were run for a complete Stocking Phase (since we did not have detailed information about the Input-Output phase). Because of the time independent approach proposed by this policy, the purpose of simulation is only to ensure that collapse does not occur. The resulting average distance rate per block was 110.54 m/block. Were this traditional operation assumed, it would imply a 25% increase in the total distance travelled by the crane.

Table 5: Traditional and Random Operation. Average Stock Ratio (100 simulations per case).

Policy	Mean(m/block)
Traditional	110.54
Random	112.80

The random policy was tested for the Stocking Phase resulting very similar to the traditional operation. The obtained average rate was 112.80 m/block. This implies that traditional policy is equivalent to not following any, under a total distance minimization effort.

5. CONCLUSIONS

A simulation-based analysis of the concrete armour units manufacturing plant at Punta Langosteira has been described. Simulation has proved to be especially effective for the analysis of large and complex systems as is this case. It has also allowed an enhanced dynamic project management, highly valued by the engineering team.

We have employed a commercial tool -Delmia QUEST- as well as our own java developed discrete event simulator SIMPA. A group of policies have been simulated under different scenarios leading to a simple set of rules according to its easy implementation and low cost.

A Greedy algorithm is proposed as well as the indication of employing 5 vehicles for the transportation of blocks. Besides, a reduction in the crane operational speed would lead to a more profitable crane's utilization. This Greedy rule has proven to be a robust operation faced with variations in set up times and with a doubled rejection rate.

Simulations indicate that our model may result in a 20% reduction in the total distance travelled by the crane in comparison with traditional operation. Even though this is a very conservative guess since it only takes into account the Stocking Phase, it has an immediate and proportional effect in reducing the plant operational cost.

ACKNOWLEDGMENTS

We wish to express our gratitude to the Xunta de Galicia, which has funded this work through the research project "Investigación e Desenvolvemento dun Sistema de Xestión Integral de Terminais de Contenedores" (PGIDIT05TIC051E).

REFERENCES

Autoridad Portuaria de A Coruña, 2008. Available from: http://www.puertodelatlantico.com [accessed 4 May 2008].

Gambardella, L.M., Rizzoli, A.E., 2000. *The Role of Simulation and Optimization in Intermodal Container Terminals*. Istituto Dalle Molle di Studi sull'Intelligenza Artificiale. Available from: http://www.idsia.ch/luca/abstracts/papers/ess2000.pdf [accessed 4 May 2008]

Günther, H.O., Kim, K.H. (Eds.), 2005. *Container Terminals and Automated Transport Systems*. New York: Springer.

Gutin, G.; Punnen, A.P. (Eds.), 2002. *The Travelling Salesman Problem and its Variations*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Larsen, A., 2000. *The Dynamic Vehicle Routing Problem.* Thesis (PhD). Technical University of Denmark.

Qiao, G., Riddick, F., 2004. Modeling information for manufacturing-oriented supply-chain simulations. *Proceedings of the 36th Conference on Winter Simulation*, pp. 1184-1188. December 5-8, Washington D.C. (USA).