Modelling, Simulation and Optimization of a Shipbuilding Workshop

Frédéric Bair, ANAST (University of Liege), Liège/Belgium, F.Bair@ulg.ac.be
Yves Langer, ROGP (University of Liege), Liège/Belgium, Yves.Langer@ulg.ac.be
Jean-David Caprace, ANAST (University of Liege), Liège/Belgium, JD.Caprace@ulg.ac.be
Philippe Rigo, ANAST (University of Liege), Liège/Belgium, Ph.Rigo@ulg.ac.be

Abstract

This study concerns the simulation of a shipbuilding workshop, and its optimization to improve its productivity. Usually, simulation is used to improve efficiency but this optimization is often done manually, particularly for shipbuilding workshop. This is due to the particularity of pieces to be manufactured: almost each of them is different and required thus particular operations. The automation is not so much present as, for example, in automotive industry. In spite of high CPU time for that kind of simulation, we will try to use optimization methods to improve productivity. Algorithm used is a genetic algorithm.

1. Introduction

Nowadays, simulation in shipbuilding becomes more and more important. The use of simulation-based design and virtual reality technologies facilitates higher efficiency in terms of work strategy planning, and offers, as a result, significant productivity gains. Such gains cannot be easily obtained only by using the simulation tools. It is required to link the simulation model with an optimization package.

In the first chapter, we will present briefly the shipbuilding workshop. This research is an academic research funded by a University of Liege research program called “Credit Impulsion”. To obtain relevant results for the shipbuilding industry, we have modelled a real workshop of a shipyard (a workshop of “Chantiers de l’Atlantique”, ALSTOM Marine, Saint-Nazaire, France). They accepted to give us confidential information and all parameters necessary about one of their workshops. The project still remains an academic project because calibration has not been done between the model and the true workshop. For confidentiality reason, the model presented here has been modified (as well as setting figures) and results are thus only indicative values. The workshop studied is a welding workshop: the inputs we have are individual pieces (or small sub-assemblies) and the outputs we have are assemblies. The goal is to realize the maximum number of assemblies in the minimum time.

The second chapter concerns the optimization of the model. Some shipyards (e.g. Flensburger Schiffbau-Gesellschaft) or some research centres use simulation for their production (*Steinhauer, 2003*) but the optimization of the model is generally done “by hand”, without using simulation tools. One of the reasons is the CPU time which is quite important to evaluate production time given by simulation. This is a severe problem because models are always strongly stochastic models and thus we have to execute several simulations to obtain one mean of the global production’s time. Due to the complexity of the system, the best optimization tool to solve the problem seems to be genetic algorithms. To reduce CPU time, our idea is not to make several simulations to have a mean, but to perform only one simulation per individual (see chapter 3 for definition). The main danger of this method is that the algorithm will tend to keep solutions which are maybe not the best, but are good solutions only by luck (remember that the process is strongly stochastic). So the genetic algorithms have to be configured to keep the best solution, but not at any prices because they are maybe not as good as predicted. At the end of the optimization process, we also have to study more precisely best solutions by executing different simulations and observing their variance. This method will probably not give the global optimum, but the goal is to see if the gain obtained by this simplified method is significant. Even if the global optimum could allow a workshop to improve its productivity by 30%, it could be interesting to find a local optimum with a gain of 10%!

2. Description of the model

2.1. Introduction

The main characteristic of this workshop (Fig.1) is its symmetry: left side is identical to the right side. Each side has its own tools and its own arrival of pieces and exit of finished assemblies. However, there is one tool which is shared between the two sides: the automatic welding robot. This is the only link between the two sides.

Of three different production lines, two are identical and the third allows treating bigger pieces. Each line (or "cell") contains two symmetric parts (or "half-cells") which produces their own assemblies. Each half-cell has one area to store containers (area that will be called STK area) and one area to produce the assembly by realising the sequence taking-welding-finishing. Pieces will not arrive in the workshop by the same way: long plates arrive by container in the entry area, long girders come by the side and are stored just before the exit containers, and smallest pieces arrive in small containers put in front of each work area (in the STK area).

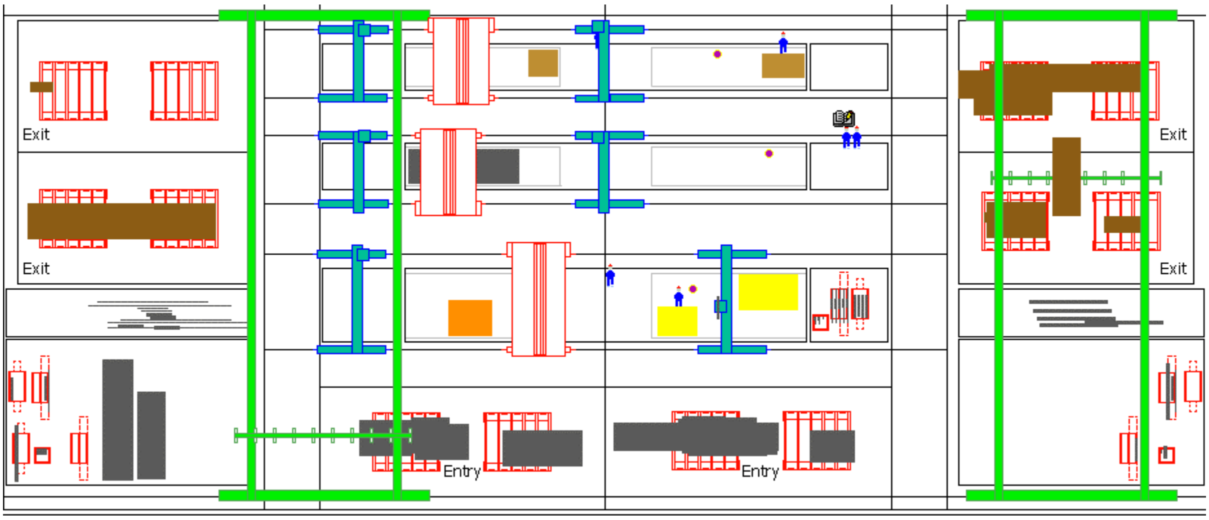

Fig.1: Model of the welding workshop

We will now describe tools used in the workshop and their modelling before to explain more precisely its functioning.

2.2. Tools used and their modelling

Two crane bridges, one on each side, bring input pieces from the entry zone located in the bottom of the workshop, Fig.1, to the work area of the corresponding cell. Sometimes, the pieces we need are not on the top of the entry container, so we have to put upper piece in another area (in the bottom of the workshop, to the extreme right or left depending on the side we are) to create a new pile of pieces.

The crane bridge is also used to evacuate finished assemblies and to evacuate them on one container of the exit (two containers – called PM – are used for the evacuation for each side). Assemblies can be classified in different groups depending on the larger assembly to which they belong (called "panel"). So at the exit, we will pile only same assemblies on the exit containers. We have two different exit containers, so we can have two different groups of assemblies at the same time in the workshop. Unfortunately, it will not be always the case so when we have an assembly to evacuate and if there is no free container, we store it in a temporary area (just before exit containers).

For each half side, we have one mechanized gripper. It is used to bring pieces from the containers of the STK area to the work area and to tack them. For very small pieces, the mechanized gripper is not used but manufacturing and tacking is done manually by workers.

The welding robot is the only tool which is shared between the two sides of the workshop. It is a completely automatic tool, which is used to weld pieces that have been tacked before either manually or by the mechanized gripper. Practically, total time for the welding is calculated before by a special software which knows constraints and limitations of the welding robot, and geometric characteristics of pieces to weld. Right now these data are not yet available so we use a simplified method to evaluate the total welding time. Thus, by assembly we estimate a welding time and this time is fixed (not linked to any randomness).

2.3. General process of the workshop

For the input, we have a list of assemblies which have to be realized. In fact, we have two different lists, one for each side. These assemblies are grouped and constitute what we call a kit. A kit can contain only one assembly, but may also be four or five small assemblies. All assemblies of one kit will be done on the same half-cell at the same time. Our goal is to find the best sequence of all these kits to minimize the total time of production. These kits have to be defined before by the user, and cannot be changed for the moment, but it could be interesting in a further research to try to obtain directly these kits by an optimization tool.

For the simulation the sequences of these kits are inputs but using optimization it is possible to find the best sequences which give – in mean – the shortest production time. There are two different ways to use the simulation and to define the sequence: the sequence could be either a "half-workshop sequence", or a "half-cell sequence".

<u>In the half-workshop sequence</u>, we fix at the start two sequences of kits (one for each side) which is the order of production of the kits. As soon as a half-cell is free, we allocate the next kit in the list to this free half-cell. Thus we don't know at the start of the simulation where kits will be done. Plates are stack at the entry on containers (PM) according to the orders of this sequence. The advantage of this type of sequence is thus to avoid using unnecessarily the crane bridge to stack up plates from PM to a temporary area because plates we need will always be on the top of the stack. Unfortunately the major problem is to equilibrate each side of a cell in order to saturate the welding robot. Indeed this operation is the longest so it is really important to avoid down time of this tool. With this method it is very difficult to satisfy that balance.

<u>In the half-cell sequence</u>, each kit has received its destination (half-cell) at the start of the simulation. So we can try to put kit on the left and on the right of a cell which are very complementary to saturate the welding robot. Problem of this method is to manage breakdown. If there is a problem or a delay in a cell, it could decrease strongly the productivity if we want to keep absolutely the sequence predicted.

Obviously each type of sequence has its own advantages and disadvantages, and simulation can predict which one is the best in different situations.

Left Side		right Side
1-2		
3	≥	1-2
4-5-1-2	≤	3
3	≥	4-5-1-2
4-5-1-2	≤	3
3	≥	4-5
4-5		

Fig.2: Saturation condition of the welding robot

The sequence of the process is: Preparation (1) – Tacking (2) – Welding (3) – Finishing (4) – Evacuation (5). Workers have to alternate work on each side of the cell (tacking in one side followed by finishing on the other one). So to saturate the welding robot, we have to satisfy the following scheme:

The model developed is very detailed as we can see in the list below (non exhaustive) which shows parameters introduced in the simulation:

- supply and evacuation times (different for each container)
- set-up time, speed, acceleration of the crane bridge
- set-up time, speed, acceleration of the mechanized gripper
- tacking time by meter (different if done manually or with the mechanized gripper)
- take down and up time (for crane bridge and robotic gripper)
- maximum capacity of each container
- …

Each time introduced in the model is a mean (with a distribution, variance and bounds associated) and can be changed easily.

Each piece which has to be weld in the workshop has been precisely modelled with all its characteristics (weight, length, wide, …). Attributes of each piece can be easily observed by clicking on the piece. All these characteristics come directly from a database Access which has been modified by many subroutines in order to facilitate data exchange with eM-Plant software.

2.4. Results

Results presented in this section are quite arbitrary because the simulation has not been calibrated yet with the reality, but they are shown to see what kinds of results can be provided. An important point is there are three simulations of all the kits, the first avoids the start with an empty workshop (this is the warm-up period), the second is the simulation which is studied and the third avoids finishing with an empty workshop. This last simulation must not be completed, we stop when all kits of the second simulation are done. Thus all the results shown below concern each time the second simulation. In fact all statistics that we desire can be easily obtain with the simulation. An interesting result is the time of each operation for each kit: this is shown in Fig.3.

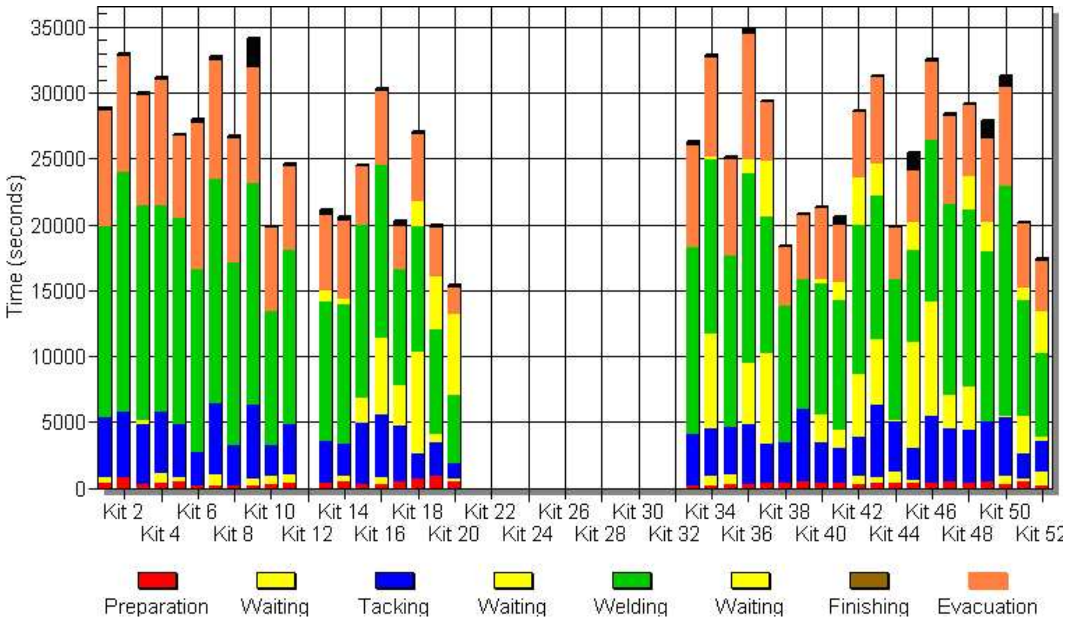

Fig.3: Duration of operations for each kit

Another interesting result is the occupation of crane bridges, and as predicted we can see in the Figure 4 that there are busier in case of a half-cell sequence, but this difference is not really important (in fact, it depends on the sequence chosen, it is not rare to have a difference of 10% or even more):

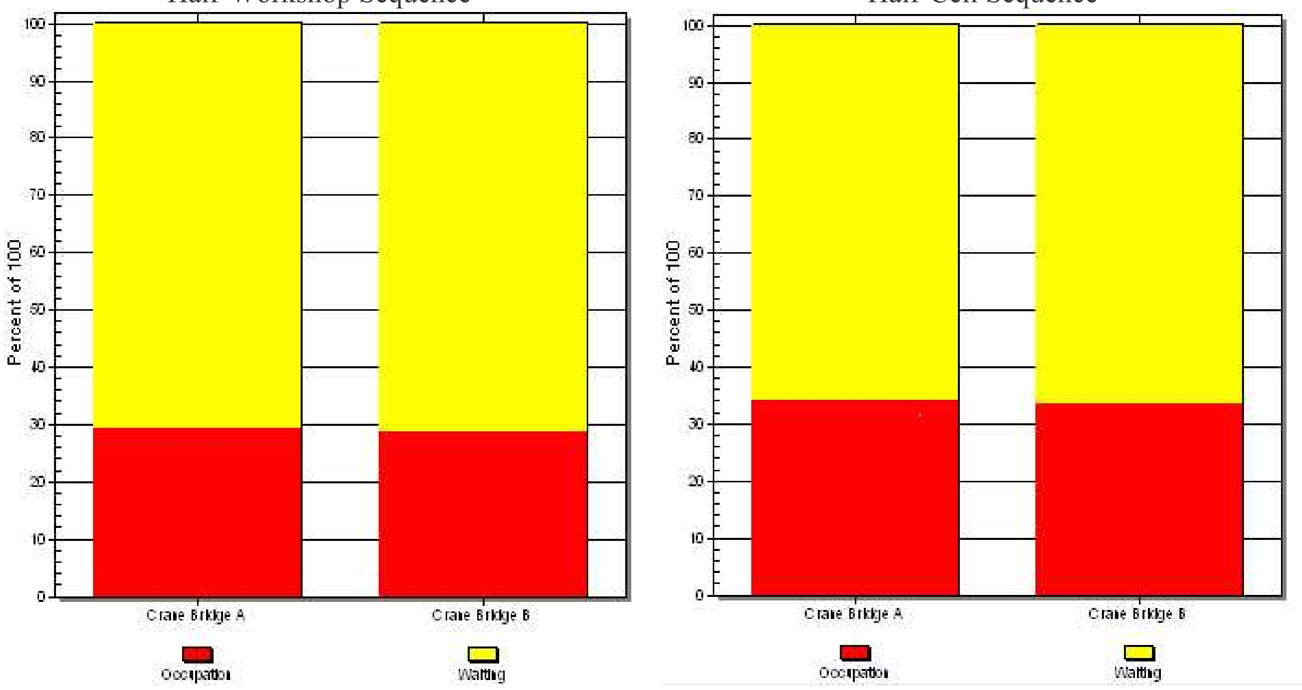

Fig.4: Comparison of crane bridges occupation for different type of sequence

A Gantt chart, Fig.5, can also be obtained. Results are exported in an Excel file which is treated by some Excel macros (developed in VBA) so that they can be shown in a Gantt diagram (we can select one particular half-cell, or see all at the same time to see the balance between the two sides).

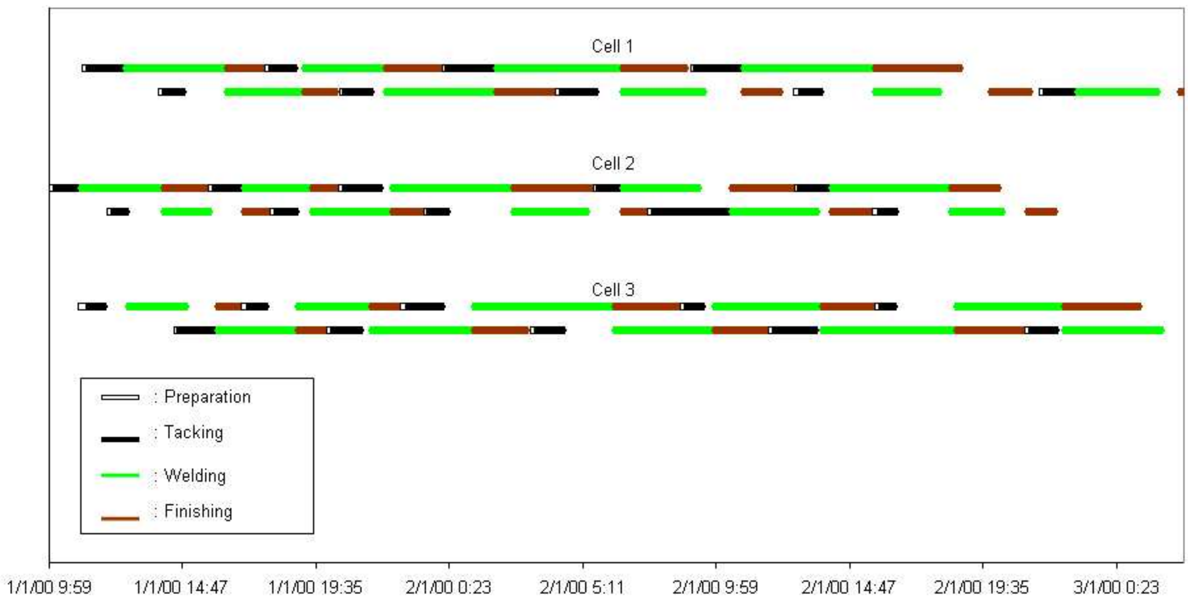

Fig.5: Gantt Chart

A lot of others results are available but we cannot show all here. They allow understanding more precisely what exactly happens in the workshop.

3. Optimization of the model

3.1. Introduction

As said before, the goal is to find the best two sequences which give the shortest production time. It is crucial to clearly define what will be our objective function (that we will call fitness). To avoid starting with an empty workshop, the first simulation of all the kits is done without recording statistics. This recording is only done for the second simulation (the third will not be totally simulated, its presence is only to avoid also ending with an empty workshop). The problem with that warm-up period is that each half-cell will start to work and to record statistics at a different moment.

At the end of the simulation, some kits of the third simulation are also done – which distorts the true fitness. So we will use the sum of the fabrication time of the second simulation divided by three (because we have three cells). We will thus have one time by side, and the fitness will be the sum of these two times.

If we run different simulations with the same sequence, we obtain a variation of the fitness of about only 1.5% which is very low (with a confidence interval of 90%)!!! This is due to two main reasons:

- The longest operation is the welding and this operation has no variation (time is given and is fixed)
- There is a lot of waiting time between successive operations: these waiting periods thus act like a buffer and small variation of operation times are absorbed in all these waiting times

This characteristic is very important and will be helpful for the optimization.

Because of the complexity of the model and of the kind of optimization (a sequence to optimize), it seems that genetic algorithms are the most appropriate tools to solve this problem. The genetic algorithm used is part of the eM-Plant software. Different types of genetic algorithms exist and they often require to define a lot of parameters (type of mutation, cross-over, …). The next chapter will explain its functioning.

3.2. Description of the genetic algorithm used and its parameters

3.2.1. Introduction

Genetic Algorithms (GAs) are computerized search procedures based on principles of the natural evolution and heredity. There are many reference textbooks and papers about Gas, *Birk et al. (2003)*, *Coley (1999)*, *Davis (1991)*, *Goldberg (1989)*, *Haupt et al. (1998)*, *Man et al. (1999)*, *Michalewicz (1996)*.

The search space of the problem is represented as a collection of “individuals” which are referred as “chromosomes”. A set of individuals in the same time is called a population. Each individual is composed of a set of values, or “genes”, which define the individual’s characteristics. The binary coding (i.e. each gene is a 0 or 1) is usually used for the chromosome. The term “fitness” is used to describe the capability of solving the optimization problem by an individual. An individual’s fitness determines the individual’s likelihood for survival and mating. The fitness is measured by a predefined fitness function. For the calculation of this fitness function the binary chromosome is converted to a real number, and employed in the function equation. The fitness function is useful also for watching how the GAs evolves better chromosomes over time. After initiation, each generation produces new children based on the genetic cross over and mutation operators. It is based on the assumption embedded in the idea of natural selection that, as members of the population mate, they produce offspring (individuals generated by the reproduction process) that have a significant chance of retaining the desirable characteristics of their parents, perhaps even combining the best characteristics of both parents. The population of variants is continually evolving in response to selection pressure described by fitness function. In this manner, the overall fitness of the population can potentially increase from generation to generation. The same manner can also be employed to discover optimal solutions because one can start with a set of solutions which are not necessarily desirable, but the solutions obtained by combining some of the best characteristics of their parental solutions must have a higher fitness. After evolution of generations, the optimal solution can finally be obtained. Different parameters have to be chosen to use efficiently the algorithm and it is important to describe them.

3.2.2. Mutation operator

Two different type of mutation can be used: inversion of a sequence or mutation of two single genes, Fig.6. For an inversion, random inversion range is chosen and then the sequence of genes within this range is inverted. Another mutation operator is to exchange two randomly chosen genes. Fig.7 demonstrates how the operator works.

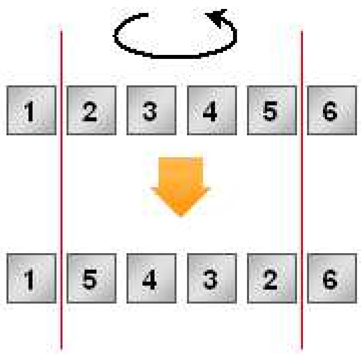

Fig.6: Mutation for an inversion task

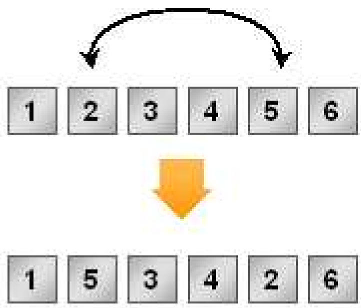

Fig.7: Simple exchange between two random genes

It is also possible to set how the genetic algorithm exchanges the positions by using a probability for exchanging each individual element or by entering a number for the exchanges.

3.2.3. Cross Over

Contrary to the mutation and inversion operators, the crossover operator is applied to two chromosomes. It exchanges elements between these ones. Initially two randomly intersecting points are chosen and then the ranges between these two points are exchanged.

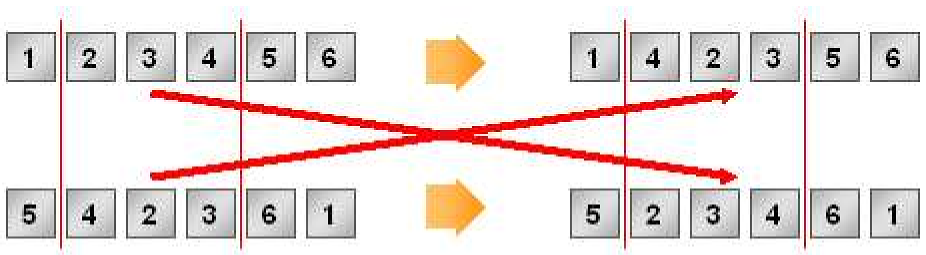

Fig.8: Cross over between two solutions

Using crossover operators will result in better solutions because there is a high probability that short good ranges are preserved and will thus be reproduced in a growing number of solutions. We differentiate two different types of crossover operator. The operators differ in how they stress the relative and absolute position of the individual elements. While Order Cross Over (OX) preserves the relative position (neighbour relation) of the elements of the solution to each other. Partially Matched Cross Over (PMX) stresses the absolute position of the objects. So, when you use OX, small groups of solution objects are kept together while PMX separates the solution objects to preserve the absolute position of individual elements.

3.2.4. Parent selection

We can use two different parent selections: Deterministic or Random. For a deterministic selection, the parents are selected randomly according to their fitness values (roulette wheel selection). For this reason individuals with good fitness values will be used more often as parents for creating the next generation. But individuals with a bad fitness value also have a chance to be used as parents. For a random selection, the fitness values are not used. All individuals have the same likelihood to be used as parents.

3.2.5. Offspring Selection

This parameter allows us to select family member used in the next generation. We have four different possibilities:

- Only use the two child solutions and selects the solution with the better fitness value (1of2)
- Use parent and child solutions and selects the best solution (1of4)
- Use parent and child solutions. It employs a stochastic selection. Selection probabilities are proportional to the quality of the solution (Prob)
- Select the individual taken for the new generation regardless of the fitness value (Random)

3.2.6. Fitness Reference

After a new generation has been generated and evaluated, we need to determine the individuals of the parent generation used for creating the individuals of the next generation. The fitness value of the individual proposed solutions can be evaluated by two different methods.

- Absolute: for this setting the fitness value of the individual solutions is going to be evaluated relative to absolute zero. At the beginning of the optimization runs, it produces good results because of the variance of the individuals in the generation. With this setting the selection pressure for homogenous generations of individuals is extremely low.
- Relative: for this setting the fitness value of the individual solutions is going to be evaluated relative to the worst solution of the generation of individuals. For homogenous generations of individuals an even selection pressure is maintained.

3.3. Results

3.3.1. Introduction

The goal is to find the best kits sequence which minimizes production time. As explained before, there are some constraints on this sequence because assemblies which belong to the same panel must be realized successively (because at the exit of the workshop we book each container for one type of panel and thus we cannot evacuate assemblies if they belong to more than two different panels). Thus some kits will be grouped together and we will do the hypothesis that the fabrication’s order of these groups cannot be changed. In fact it is logical that we cannot modify our sequence as much as we want because the arrival of pieces in the workshop depends also on previous workshops. An order for these groups is chosen and cannot be modified; the optimization consists to find the best sequence of kits in each group. Results showed further that our first results and some developments must still be done to improve the optimization.

3.3.2. Results for a Half-workshop sequence

The formulation in this case is quite easy: we have 6 chromosomes, one for each group, the four first constituent the sequence for one side, the two last for the other side. Remember that in this half-workshop sequence, we do not fix in advance where the kit will be assigned.

Fig.9 below shows the evolution of the fitness (best, mean and worst) and an optimization with the following parameters:

- Size of generation: 20
- Number of generations: 10
- Fitness reference: Absolute
- Parent Selection: Deterministic
- Cloning Best solution: Yes
- Offspring Selection: 1of4
- Cross Over: PMX
- Mutation: Simple exchange between two random genes (probability of 0.1 to occur)

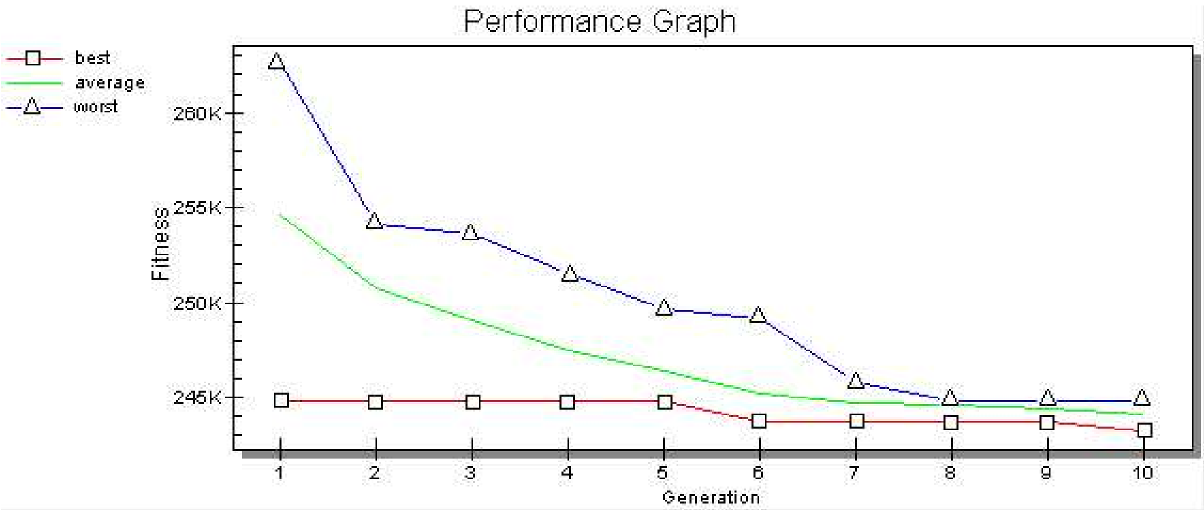

Fig.9: Evolution of the fitness

These results seem good because we have a gain between the mean of the first generation and the best fitness of 4.7%, and the mean and the worst value of the last generation are also much better. Unfortunately we have to be careful because we tend to have homogenous generations of individuals. The mean curve and the worst curve do not improve because of the efficacy of the algorithm, but because the individuals who constitute these curves become less and less different. For example, if we evaluate the fitness of a lot of randomly individuals (without any optimization), we find practically the same best fitness!!! We also have a weak improvement of the best fitness during modifications of the generation. Improvements must thus have to be done, even if this method could give us a good first sequence.

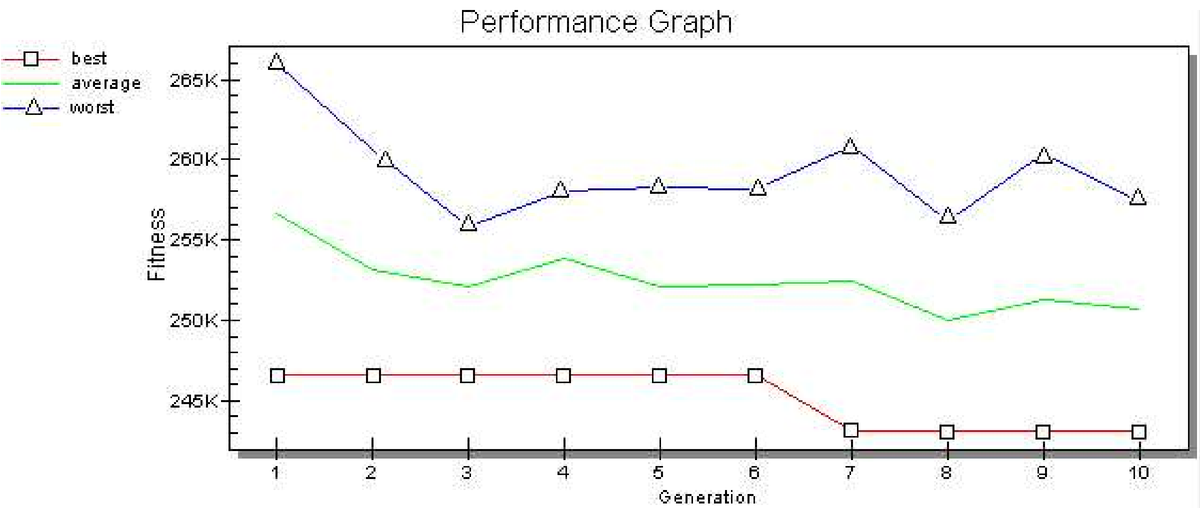

Fig.10: Evolution of the fitness (Offspring Selection: 1of2)

We can see in Fig.10 that if we try to avoid homogenous generations (for example by choosing an offspring selection of 1of 2), the convergence is not so obvious.
But we still have a gain of about 5% in comparison of the mean of the individuals of the first generation. We also observe a slight improvement of the mean, but it is not very marked. Different optimizations by changing parameters have been done, but we can't show here all the results.

3.3.3. Results for a Half-cell sequence

The optimization of this type of sequence seems more logical because this optimization can really try to improve the balance between the two sides. It was also the case for the previous sequence but results were difficult to interpret: the same sequence could lead to completely different results since kits will not necessary be done each time on the same half-cell. Nevertheless the half-cell sequence is more difficult to adapt for the optimization algorithm and we now need 14 chromosomes to make one individuals (detailed explications will not be given here).

With the followings parameters, we obtain the results shown in Fig.11.

- Size of generation: 20
- Number of generations: 10
- Fitness reference: Absolute
- Parent Selection: Deterministic
- Cloning Best solution: Yes
- Offspring Selection: 1of4
- Cross Over: PMX
- Mutation: Simple exchange between two random genes (probability of 0.1 to occur)

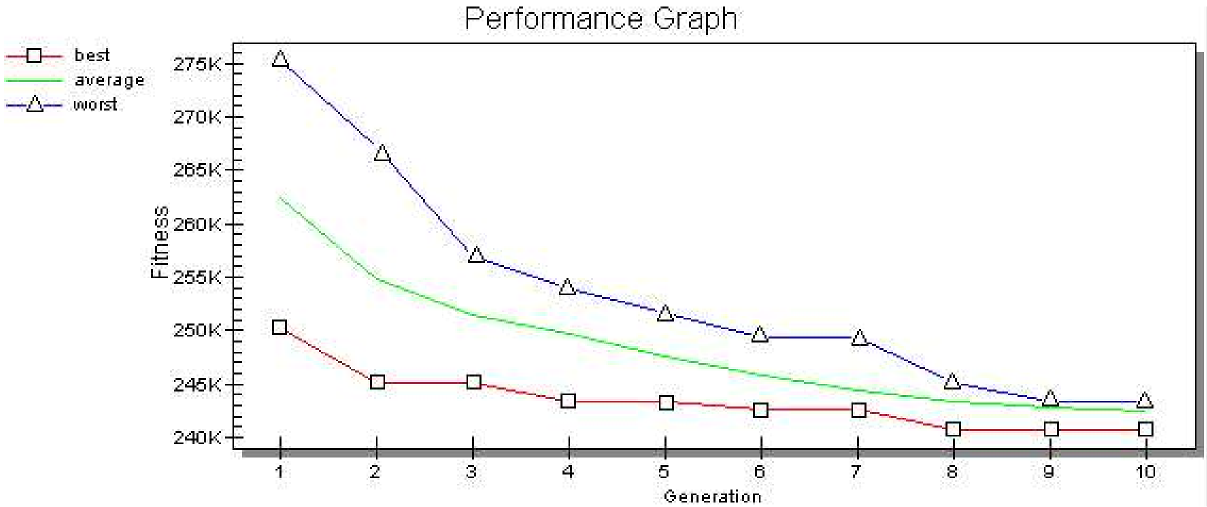

Fig.11: Evolution of the fitness – Half-cell sequence

Same remarks can be done here: the convergence of the mean and of the worst curve are due to a homogenization of the population. At the first generation the mean has a higher fitness than the previous generation. It is conforming to our waiting time because the crane bridges have to do more operations in this type of sequence. The gain between the best fitness and the mean of the first generation is now 8.4% (and 3.6% between the best fitness at the first and the last generation). Then it seems that optimization is better in this case. This is confirmed when we run a great number of different random sequences, without any optimization. The best result obtained is 249K, which is not as good as the result after an optimization.

In conclusion, even if the optimization capability has not yet been fully validated (keep in mind that these results are the first results, different parameters have till to be done to improve the process), it is very interesting to choose the results of the optimisation as a first sequence (gain of more than 8% on a random distribution), which could be modified manually after a more precise analysis of all the results.

4. Conclusions and perspectives

Simulation is obviously a powerful tool and even without optimization can be very useful. The main difficulty is the development of the model because it takes a lot of time, particularly for our model which is a very detailed model. But once it has been developed gains and information about productivity can easily be obtained.

First results obtained shows that coupling simulation with an optimization tool can be done and encourage us to do further investigations. To really assess gain that can be obtained, a calibration of the model and the workshop has to be done. Results will also depend a lot on the variance of the model. The lower it is the better it is for the optimization. If variance is too high, it will be necessary to optimize the mean of a sequence which will increase strongly the CPU time. Nevertheless automatic optimization is finally not so hard to develop compared with time required to build the simulation model and give quickly good results. Optimization and simulation will be more and more coupled in the following years.

After an improvement of this optimization, the next optimization step is to automatically define the kit content. We will start with a list of assemblies and group them to create kits. This optimization is more complex because we raise considerably the solution space, and we have an additional constraint: the use of the ground space in the work area. Each assembly has its own size and thus it is not so easy to determine automatically which assemblies will constitute one kit.

Acknowledgments

The authors thank University of Liege for funding the research "Credit Impulsion Research Fund", "Region Walonne" (RW) and "Fond Social Européen" (FSE). Special thanks to "Chantiers de l'Atlantique", ALSTOM Marine, Saint-Nazaire, France for the comprehensive set of data used to model one of their workshops and more especially to Mr. François Ellie.

References

BIRK, L.; HARRIES, S. (2003), *Optimistic: Optimization in Marine Design*, Mensch & Buch, Berlin

COLEY, D.A. (1999), *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, Singapore

DAVIS, L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York

GOLDBERG, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc

HAUPT, R.L.; HAUPT, S.E. (1998), *Practical Genetic Algorithms*, John Wiley & Sons, Inc., New York

MAN, K.F.; TANG, K.S.; KWONG, S. (1999), *Genetic Algorithms*, Springer-Verlag, London

MICHALEWICZ, Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin Heidelberg

STEINHAUER, D. (2003), *The Virtual Shipyard – Simulation in production and logistics at Flensburger*, 2nd COMPIT'03, Hamburg