SIMULATION AND OPTIMIZATION OF A SHIPBUILING WORKSHOP

Frédéric Bair [1], Yves Langer [2], T. Richir[3] and Philippe Rigo [4]

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.

KEY WORDS

modeling, optimization, production, scheduling, simulation.

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. 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 reasons, 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 are assemblies. The goal is to realize the maximum number of assemblies in the minimum time.

---

1 Research Engineer, Dept. of Naval Architecture and Transport System, ANAST, 1 Chemin des Chevreuils, Liège, B-4000, Belgium, Phone +32 (0) 366 96 21, FAX +33 (0) 366 91 33, F.Bair@ulg.ac.be

2 Management Engineer, n-Side, CEI, 6 Chemin du Cyclotron, Louvain-La-Neuve, B-1348, Belgium, Phone +32(0)10390032 Yves.Langer@n-side.be

3 Ph.D. Student, F.R.I.A. Scholarship, ANAST, University of Liege, Bat. B52/3, Chemin des Chevreuils 1, B-4000 Liege, Belgium, Phone +32-4-3664859, Fax +32-4-3669133, thomas.richir@ulg.ac.be

4 Professor, F.N.R.S. Research Director, ANAST, University of Liege, Bat. B52/3, Chemin des Chevreuils 1, B-4000 Liege, Belgium, Phone +32-4-3669366, FAX +32-4-3669133, ph.rigo@ulg.ac.be

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. 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 this solution is 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%!

DESCRIPTION OF THE MODEL

INTRODUCTION

The main characteristic of this workshop (Figure 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.

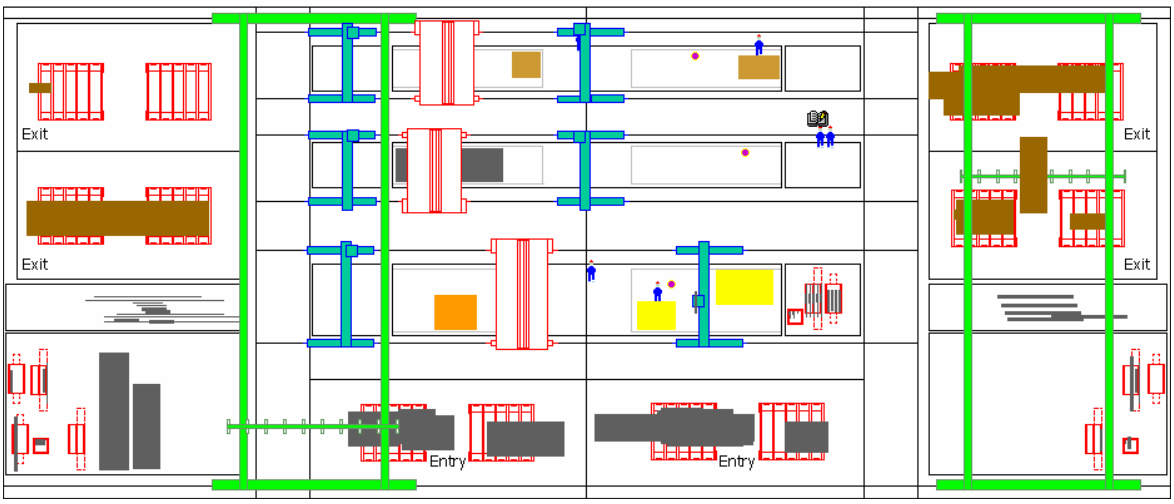

Figure1: Model of the welding workshop

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 realizing 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).

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

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 (see Figure 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 largest 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 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).

GENERAL PROCCESS OF THE WORKSHOP

In inputs we have a list of assemblies that 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 contain 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 by using optimization it is possible to find the best sequences which give – in mean – the shortest production time.

At the start we fix 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 stacked at the entry on containers (PM) according to the orders of this sequence. The advantage of this way of functioning 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.

The model developed is very detailed, parameters introduced in the simulation are (list non exhaustive): supply and evacuation times (different for each container), set-up time, speed, acceleration of the crane bridge and 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 mechanized 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.

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 starting with an empty workshop (this is the warm-up period), the second is the simulation 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.

Simulation can provide us interesting results as occupation of crane bridges, of mechanized grippers or also of workers. It directly tells us which tool in the workshop is a bottleneck. Another interesting result is the time of each operation for each kit (see Figure 2).

A Gantt chart 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.

A lot of others results are available but we cannot show all here. They allow understanding more precisely what exactly happens in the workshop. The impact of breakdowns can also be easily studied once the model has been developed.

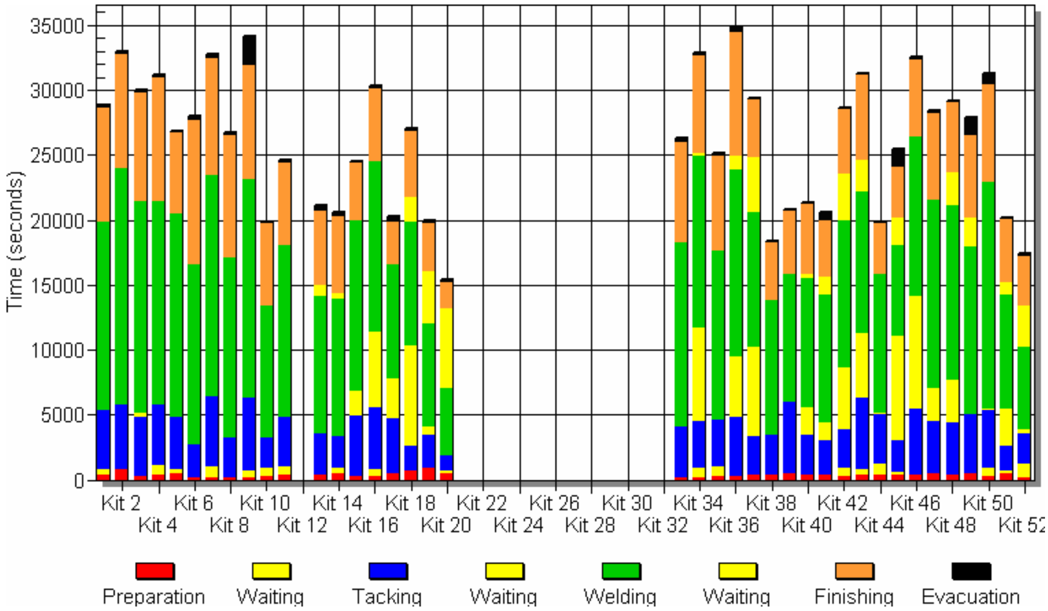

Figure 2: Duration of operations for each kit

Optimization of the model

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).

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 variations 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 defining a lot of parameters (type of mutation, cross-over, etc). The next chapter will explain its functioning.

Reminder of the functioning of genetic algorithm

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 found. Different parameters have to be chosen to use efficiently the algorithm and it is important to describe them.

Mutation operator

Two different type of mutation can be used: inversion of a sequence or mutation of two single genes. For an inversion, random inversion range is chosen and then the sequence of genes within this range is inverted.

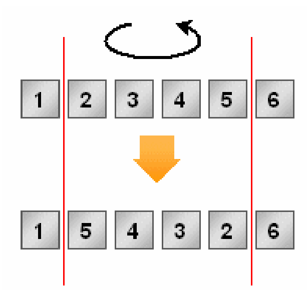

Figure 3: Mutation for an inversion task

Another mutation operator is to exchange two randomly chosen genes. The illustration below demonstrates how the operator works.

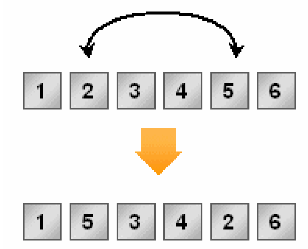

Figure 4: Simple exchange between two random genes

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.

Crossover

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.

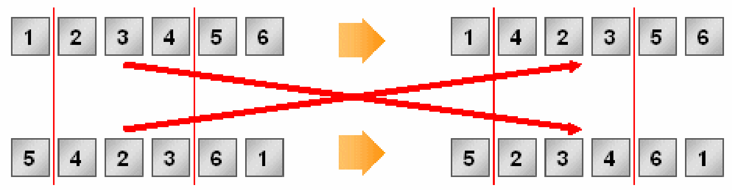

Figure 5: Cross over between two solutions

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.

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.

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)

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.

RESULTS

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 are our first results and some developments must still be done to improve the optimization. The formulation for the genetic algorithm is not very easy. We have 14 chromosomes to make one individuals (detailed explications will not be given here).

Figure 6 below shows the evolution of the fitness (best, mean and worst). The optimization has been done 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).

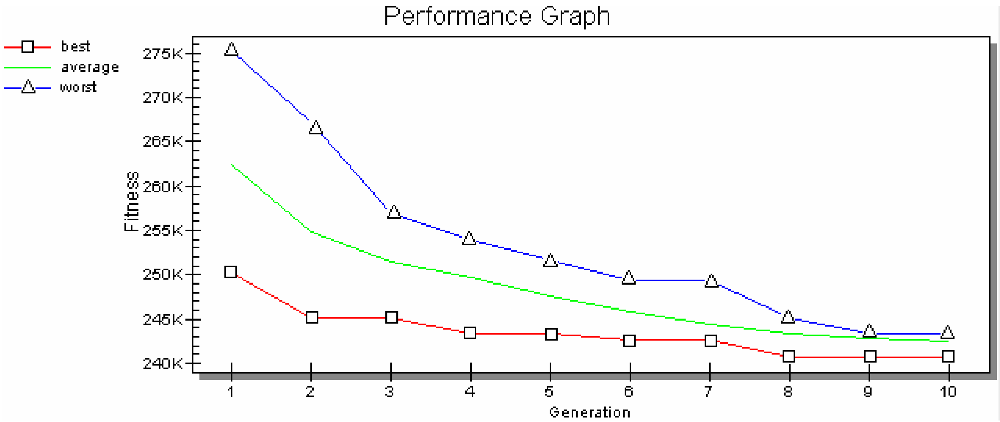

Figure 6: Evolution of the fitness – Half-cell sequence

We have to be careful when we analyze the results: the convergence of the mean and of the worst curve are due to a homogenization of the population. The gain between the best fitness and the mean of the first generation is 8.4% (and 3.6% between the best fitness at the first and the last generation). 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 one optimization. Different optimizations by changing parameters have been done, but we cannot show here all the results.

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 still to be done to improve the process), it is very interesting to choose the results of the optimization 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.

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 show 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

Authors thank University of Liege for funding the research “Credit Impulsion Research Fund”, “Région Wallonne” (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 and Mrs. Céline Boiteux.

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 international EuroConference on computer and IT applications in the marine Industries COMPIT’03, Hamburg.

Tecnomatix*, eM-Plant v.7.0,* User’s Guide.