

EurOMA

European Operations Management Association

13TH INTERNATIONAL CONFERENCE:

MOVING UP THE VALUE CHAIN

Scheduling mixed-model assembly lines
with genetic algorithms:
the Aprilia case study

De Toni A. F. (University of Udine)

Nonino F. (University of Udine)

18th-21st June 2006
University of Strathclyde
(Glasgow, UK)

SCHEDULING MIXED-MODEL ASSEMBLY LINES WITH GENETIC ALGORITHMS: THE APRILIA CASE STUDY

*Alberto Felice De Toni**, *Fabio Nonino**⁺

Management Engineering Laboratory, University of Udine

via delle scienze, 208, 33100 Udine (UD), Italy

**Email: detoni@uniud.it - Tel. +39 0432 558330*

+Email: fabiononino@uniud.it - Tel. +39 0432 558273

ABSTRACT

The two authors deal with the topic of the final assembly scheduling realized by the use of Genetic Algorithms (GAs). The objective of the research was to study in depth the use of GA for scheduling mixed-model assembly lines and to propose a model able to produce feasible solutions according to the particular requirements of an important Italian motorbike company, the Aprilia group, as well as to capture the results of this change in terms of better operational performances. In the Aprilia case study, the scheduling problem is made more complex by the “chessboard shifting” of work teams. Therefore, a complex model for scheduling mixed-model assembly lines is required. An application of the GAs is proposed in order to test their effectiveness and robustness. The short elaboration time and the robustness of the final assembly plans, obtained during the test-stage, confirm that the choice was right and suggest the use of GAs in other complex manufacturing systems.

Keywords: Genetic Algorithms, Case study, Final Assembly Schedule.

INTRODUCTION

The article examines a case study where a complex model for scheduling mixed-model assembly lines is required. This scheduling problem is often called mixed-model assembly line scheduling (Leu *et al.*, 1996) and can be also referred to the permutation flowshop scheduling problem. In such a production system, the managers want to sequence the different products, thus obtaining a high service level (product mix) without delays in products delivery while respecting the constraints of capacity. The focus of the current work is the application of one of the Genetic Algorithms (GAs) as a technique for the resolution of such complex problems.

The remainder of the paper is organized as follows. In the former part of the article, we describe the principles and the basic characteristics that have made this technique successful and an applicative model is supplied for their use in the best combinatory problems, in particular in the final assembly scheduling phase of mixed-model assembly lines.

The case study of the Aprilia group, an important Italian motorbike company, has been used at the outset of the study. A description of the production system of the plant is presented together with the demand management logic. Afterwards, we propose a scheduling approach consisting of two stages, a macrostep and a microstep. The two steps respectively carry out a “macro-scheduling” and a “micro-scheduling” of the assembly operations; in particular, the Genetic Algorithms are applied in the microstep.

The article focuses on the development of a scheduling prototype and proposes a series of algorithms designed for its implementation. The paper ends with the evaluation of the benefits accessible by the firm by means of the scheduling prototype.

GENETIC ALGORITHMS IN SCHEDULING PROBLEMS

Simulating the natural evolutionary process of human beings results in stochastic optimization techniques, called evolutionary algorithms, which can often outperform conventional optimization methods when applied to complex real-world problems. For instance, the scheduling problems in the manufacturing planning are very complex and hard to be solved through conventional optimization techniques. Therefore, interest in these groundbreaking techniques has grown both among scholars and practitioners. There are three main research streams in the evolutionary algorithms: genetic algorithms (GAs), evolutionary programming (EP) and evolution strategies (ESs) (Ponnambalam *et al.*, 2001b). The GAs approach is different from evolutionary programming and evolutionary strategies. EP and ESs design algorithms to solve specific problems. Instead GAs are customizable because their approach only requires a performance measure, a representation of the problem and the operators who generate the new population. The advantage of this type of algorithms is their wide applicability to different problems. (Levine, 1997).

Genetic algorithms represent a powerful and robust approach to develop heuristics for large-scale combinatorial optimisation problems (Metaxiotis and Psarras, 2004). The fundamental and basic principles of genetic algorithms are described in *Adaptation in Natural and Artificial Systems* by J. Holland (1975); in the book GAs are presented as an abstraction of biological evolution and a theoretical foundation for the concept of adaptation is provided. The distinctiveness of this technique is that it draws inspiration from the natural evolution and it is founded on Darwinian principles of selection and adaptation and, naturally, on the reproduction and genetic mutation mechanisms.

As reported by Metaxiotis and Psarras (2004) GAs have been also used in key business areas, like marketing, banking and finance and forecasting, thus offering real benefits to the business decision support. In a recent review, Chaudry and Luo (2005) highlight that Gas have been applied in the two main areas in the production and operation management, i.e. scheduling problems and facility layout.

As regards scheduling problems, many authors (e.g. Della croce *et al.*, 1995; Ponnambalam *et al.*, 2001c) have proved their applicability and usefulness in the job shop scheduling with a performance criterion which can be the minimization of makespan (the completion time of the last job) or the weighted sum of the multi objectives minimization of makespan, minimization of total idle time of machines and minimization of total tardiness. GAs have been used in the scheduling problem in the two-machine flow shop (Kim and Kim, 2002) with a batch machine followed by a discrete machine in sequence, with the aim of scheduling the jobs to minimize the total completion time. Pongcharoen *et al.* (2002) used GAs for scheduling complex product, thus achieving on time delivery and 63% of cost reduction.

Due to the complexity of scheduling problems, both in practice and theory, heuristics is typically used as a solution methodology and it has been shown that GA tends to perform well in complex situations (Chaudry and Luo, 2005). According to the computational results, genetic algorithm performs better over heuristic algorithms for flow shop problems (Ponnambalam *et al.*, 2001a). Murata *et al.* (1996b) applied the GAs in the flowshop scheduling problems with the aim of minimizing the makespan. Moreover, Murata *et al.* (1996a) proposed the GAs in the flowshop scheduling with the multi-objective to minimize the makespan, the total tardiness and the total flow time (i.e. the sum of completion time over all jobs). Other authors state that the genetic algorithms have proved to be slightly inferior to other search algorithms, just as local search, taboo search and simulated annealing but the hybridizations of genetic algorithms has shown high performance proved by computer simulation and proposed also by Ruiz *et al.* (2006).

THE APRILIA CASE STUDY: COMPLEXITY IN THE MANAGEMENT OF PRODUCTION FLOWS

Aprilia Group is an important Italian brand in the industry of motorbikes and (motor)scooters. The company's factory is located in Scorzè (Venice) and is not concerned with manufacturing but with assembling final products.

Demand management

The market demand for two-wheel motor vehicles is strongly seasonal. Purchases of these vehicles are indeed higher in warmer months; so in the high season, demand is up to six times higher than in the low season. Though Aprilia is a global company, sales are influenced by the seasonality of demand which mainly comes from Europe and the United States. Company's production managers prefer to follow

demand fluctuations rather than keep production volumes constant on an average level throughout the year. Therefore, the calendar year is subdivided in two productive periods: low and high season.

Assembly operations

Vehicles are assembled by different work teams on single- and mixed-model assembly lines. The production plant contains eight assembly lines: four lines are needed to assemble the motorbikes and four lines are used to assemble the scooters. The plant can produce up to 2.500 motorcycles every day, that is, one motorbike every five minutes and one scooter every two minutes.

The assembly lines are subdivided into a determined number of stations and are made of plate conveyers where the semi finished products move forward on a vice. We can talk about *stop & go* lines: the product stops in front of the operator for the assembly time and then moves among the stations without any operation between two positions. The assemblage of the motorcycles develops along the line: it starts upstream by fixing the engine to a vice and then, as this proceeds, all the other components are added till the final product is obtained. Each line is committed to a work team; the number and composition of a team can vary according to seasonality: from March to September (warmer months), the number of operators can be more than double.

Organization of work teams: the chessboard shifting

The management of workforce changes according to production volumes by increasing the number of work teams with seasonal workers; as a result, the work teams' efficiency is extremely variable. With high production volumes (high season) the number of teams is the same of the number of lines, so that each team works on a single line. On the contrary, with low production volumes (low season), the number of teams is less than that of lines, and each team works on two or three lines.

As a consequence, some assembly lines stay idle for more or less short periods. As a result, the work teams are managed as follows:

- High season: the team works on a single assembly line;
- Low season: the team shifts among the assembly lines.

A criticality arisen from this model of work management lies in the evaluation of the efficiency curve of the work teams. Indeed, in the low season the work teams are made up solely of expert workers while in the high season there are also untrained seasonal workers. During seasonal changes, the expert personnel must support the inexpert one and must frequently stop the assembly line in order to help the new workers to learn their job. As a result, the team efficiency starts from low values, around 40%, and then increases with the rise of assembled vehicles following the *learning by doing* principle.

This choice of workforce management is particularly critical also for another reason: in order to operate with less work teams than assembly lines, a shifting logic called "chessboard shifting" is used. The chessboard shifting consists of moving a work team from one line to another, as soon as the team has finished a production lot and starts to assemble a different model, linked to a different line. These displacements can be even more frequent, according to the resettlement of the final assembly schedule (FAS) and lead to a big loss of time.

Each line is linked to a sole work team, so that this would be able to specialize in relevant models, while each work team can be linked to one or more lines. This method of workforce management is much complex as each line could be in one of the following situations:

- Full line: line with all the vices occupied by semi-assembled motorbikes;
- Empty line.

Work team shifting among lines can be managed in four different situations originated by the state of the left line (empty or full line) by the state of the line found (empty or full line), as shown in Figure 1. The four possible shifting situations of work teams have been analyzed, in order to point out the inefficiencies linked to each case. Two reasonable simplifications have been considered:

- production lines without buffer;
- slight shifting time necessary to work team to move from one line to another one.

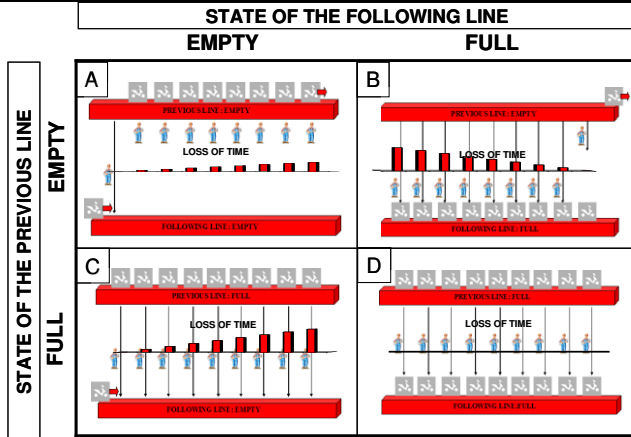


Figure 1 – Loss of time in the four type of chessboard shifting

(A) Empty-empty shifting (Figure 1A). In this situation the workers, as they move on, can start to assembly the new model, but the fastest line has to adapt to the slowest one, therefore the loss of time depends on the speed of both left and found lines. For example, we can consider that the left line is the fastest one, so the time necessary to perform the assembly operation in the following line is higher than the time necessary for the assembly operation on the former line. The first worker moves to the new line and starts to work; afterwards, the second worker has to wait for a period equal to the difference between the lines assembly time before receiving the semi-assembled product from previous position in the new line. Once his task on the left line has been finished, the third worker has to wait for the two former workers to finish their tasks on the new line. This means that waiting time accumulates and moves downstream and turns out to be the highest for the last operation of the line. On the contrary, if the following line is the fastest, the analysis is symmetrical to the previous one. In short, the worker has to wait for a certain time, once he has finished his operations, if the passage empty-empty takes place towards a faster line; on the contrary, he has to wait for a certain time before beginning his operations if the passage is towards a slower line. The total loss of time in chessboard shifting ($\Delta\text{time}_{\text{loss}}$) can be calculated with the following formula:

$$\Delta\text{time}_{\text{loss}} = \frac{\Delta\text{TC}}{2} \cdot (N - 1) \quad (1)$$

where:

- N: number of workers
- ΔTC : difference between cycle time of previous assembly line and of following assembly line

(B) Empty-full shifting (Figure 1B). Once the last operation on the line to leave is finished, the first operator moves to new line. As it is a full line, he can perform the operations committed on the semi-assembled product in his own position, but he can get the line moved forward till the last worker has shifted on the new line. The first operator has the greater and not the lower loss of time. Therefore, in empty-full shifting loss of time decreases from upstream. The total loss of time can be calculated with the following formula:

$$\Delta\text{time}_{\text{loss}} = \frac{N_p - 1}{2} \cdot \text{TC}_p \quad (2)$$

where:

- N_p : number of workers in the previous assembly line
- TC_p : cycle time of previous assembly line

(C) Full-empty shifting (Figure 11C). Once the lot on the line to leave is finished, the work team moves together to a new empty line. For the first operator there is no loss of time, while for the other workers loss is equal to the time necessary to perform forthcoming operations. Therefore, in full-empty shifting loss of time accumulates downstream. Loss of time is perfectly symmetrical compared to the former one. The loss of time can be calculated with the following formula:

$$\Delta\text{time}_{\text{loss}} = \frac{N_f - 1}{2} \cdot \text{TC}_f \quad (3)$$

where:

- N_f : number of workers in the following assembly line
- TC_f : cycle time of following assembly line

(D) Full-full shifting (Figure 1D). In the full-full shifting losses of efficiency are basically slight, as the work team moves together from one line to the following one. In this manner, production stops and restarts after the necessary time for the actual shifting of workers. Full-full passage can be performed if a work team which is leaving a line knows that, when it comes back, it has to assemble a different model from the present one.

Table 1 – Cycle times and number of workers

	TC [min]	Number of workers
Previous line	80	10
Following line	100	10

With reference to two lines with 10 work stations and cycle times of 80 and 100 minutes (see data on Table 1), the loss of time in the four possible chessboards has been calculated by means of the above-presented formulas. Shiftings between empty lines report slight losses of time (90 minutes) depending on the differences between the cycle times of the assembly lines; on the contrary, shiftings between full lines imply no losses of time. The passages between lines with different states, i.e. full-empty and empty-full, are the most unfavourable ones, as they imply big losses of time, respectively 450 and 390 minutes.

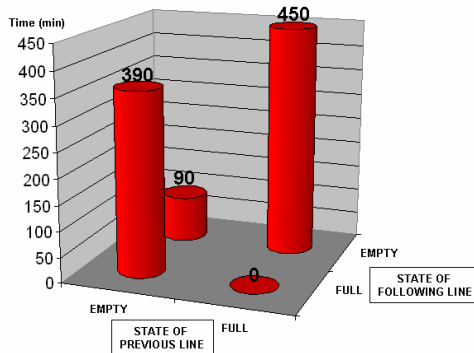


Figure 2 – Loss of time in the four chessboard shiftings

The past experience led Aprilia to work in this way:

- if shiftings are carried out by work teams that operate on two lines, the firm assigns it to two lines whose models have a cycle time similar enough to perform only empty-empty shiftings, in order to reduce both work in process (WIP) (typical of full-full, empty-full, and full-empty) and losses of efficiency (see formula 3).
- if shiftings are carried out on work teams that operate on three lines, full-full shiftings are preferable in order to achieve the maximum efficiency (no loss of time) despite the reduction of WIP.

THE SCHEDULING APPROACH

A software prototype has been designed and developed, in order to manage the complexity in scheduling the assembly plans while considering the constraints of a production system containing mixed-model assembly lines. Although the prototype design started from the Aprilia case study, the scheduling approach was to be a general purpose and customizable to several manufacturing companies. The mixed-model scheduling of assembly lines refers to the flow shop scheduling problem where a number n of jobs (lots for the assembly lines) has to be processed in m machines in the same order.

As described above, the final assembly scheduling in the Aprilia group is very complex in the low season because of the difficult work force management and chessboard shiftings. In order to rationalize this criticality, the new scheduling model considers the products as linked to the work teams and not to the assembly lines. The model schedules a feasible final assembly plan managing the work teams and not the assembly lines in order to solve the critical states coming from chessboard shifting and considering the internal constraints of capacity. Indeed, an empirical table (Table 2) has been designed, where the efficiency of work teams is linked to the team composition and the number of assembled motorbikes.

Table 2 – Work teams' efficiency

	Number of vehicles			
	<500	501-2000	2001-6000	>6000
SEASONAL WORKERS				
TEAM COMPOSITION A	0.39	0.61	0.69	0.88
TEAM COMPOSITION B	0.46	0.67	0.80	0.91
TEAM COMPOSITION C	0.47	0.70	0.80	0.91
TEAM COMPOSITION D	0.48	0.73	0.86	0.93
EXPERT WORKERS				
TEAM COMPOSITION E	0.48	0.74	0.86	0.95
TEAM COMPOSITION F	0.49	0.77	0.87	0.95
TEAM COMPOSITION G	0.56	0.79	0.90	0.97
TEAM COMPOSITION H	0.67	0.91	0.96	0.97

We chose to use the genetic algorithms as the core of the scheduling algorithm because, as recent studies highlighted, the GAs outperform the others heuristics and meta-heuristics in large combinatorial complex problems like the permutation flowshop problem (Ruiz and Maroto, 2005). Moreover, as highlighted by Leu *et al.*, (1996) the GAs select a solution based on the evaluation of the fitness function. Any function can be used. This provides great flexibility in capturing particular cost structures in mixed-model sequences and it is also possible to combine several different objectives in a composite evaluation function or to consider several objectives as ranked goals. As a matter of fact, a multi-objective genetic algorithm has been considered.

Macrostep e microstep

The scheduling approach proposed is divided in two stages, a *macro step* and a *micro step* (Figure 3). In the latter one the GAs are applied. The principal input is the Master Production Schedule (MPS), where orders are sequenced by request from the commercial unit which assigns a due date. The other necessary input data for a precise scheduling are: the cycle time of the assembly lines, the link among the products and the assembly lines, the work calendars, the work team composition, the efficiency table, priorities in the orders and the penalty rates for earliness, tardiness, setup and chessboard.

The *macro step* is responsible for:

- Defining the MPS orders subset to be processed with the lower sequence number;
- Assigning the orders and the work teams to the right assembly lines;
- Starting the microstep for building a set of solutions;
- Selecting the solutions to be conserved for the next macrostep and the final state of the assembly lines.

Header Space reserved for Publication

The *micro step* is responsible for building a set of solutions (permutations) including non-preferential orders selected by the macro step. In particular, it uses two methods according to the numbers of MPS orders (N) selected by the macro step:

- If $N < 10$, all the $N!$ permutations are generated, evaluated by the evaluation module and the best one is selected;
- If $N \geq 10$, the genetic algorithms are applied following the scheme described in the next sub section and illustrated in figure 3.

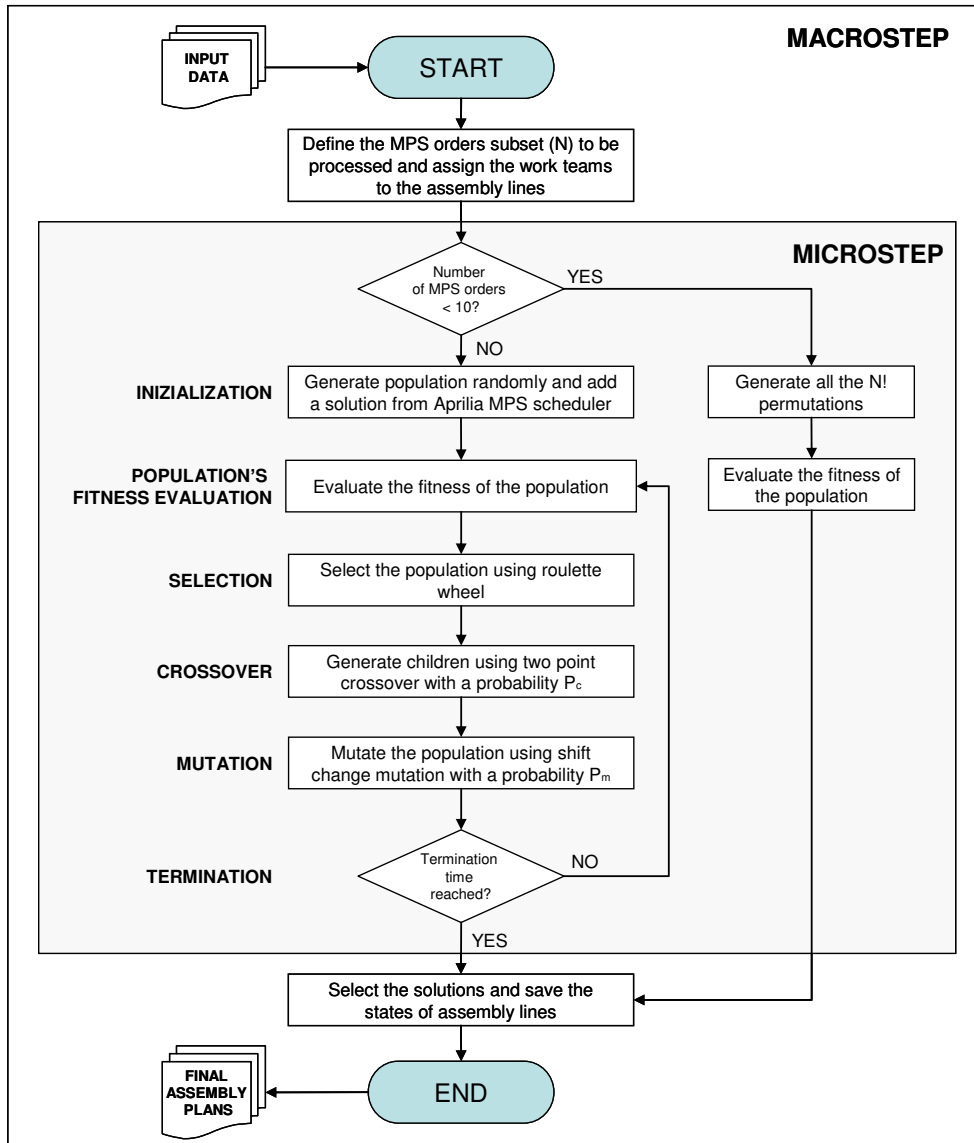


Figure 3 – The scheduling approach

The genetic algorithms

Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and evolution (Goldberg, 1989). The main concepts of genetic algorithms are population, chromosome, gene and fitness. GAs start with a population. A *population* is a set of solutions which could be generated randomly or by another algorithm. The individuals of the population are represented by a *chromosome*, which is a single solution to the problem. A chromosome consists of a finite number of *genes*. Each gene represents an entity: for instance, in the scheduling problem it represents a job. At each generation of the algorithm, the chromosomes with a better *fitness* have higher probabilities to evolve. The evolution occurs with the genetic operators which are crossover and mutation. After numerous generations, the algorithm converges to a chromosome, which is the solution of the problem. In our scheduling algorithm, we use the two-point crossover and the shift change mutation genetic operators, as suggested by Murata *et al.* (1996b); indeed, the authors showed that the two operators are effective for the flowshop problem.

Coding

Genetic algorithms encode the problem into a set of string, each of which is composed of several bits, and then operate on the strings to simulate the process of evolution (Davis, 1991). The solution of an optimization problem is usually encoded as a bit string. In the flowshop problem, which includes a number of jobs, the sequence of job has been encoded as a string like 12345 (the number indicates different lots). More generally, we consider an N-dimensional vector $\{J_1, \dots, J_i, \dots, J_N\}$, where J_i indicates the i-th job. The solution is a permutation of the N given jobs.

Initialization

Usually, the initial population is chosen completely random. We initialize the population with a random generation; however, we follow Reeve's suggestion (1995) and seed the initial population with a good solution. The solution comes from the macro scheduling and is engendered by a constructive heuristics inserted in an old Aprilia scheduler which is not able to consider the chessboard shifting.

Population's fitness evaluation

The micro step contains a module for evaluating the assembly plans. The evaluation module estimates the quality of the scheduled plans (permutations) generated inside the micro step by using a multi-objective function, which is the sum of weighted quantitative indicators. The objective function assesses the *fitness* of the chromosome. The indicators consider the costs for setup, chessboard, earliness and lateness in assembling a lot. The weights are inserted by the manufacturing planner and represent the real priorities of the firm. The multi-objective function has four objectives: minimize the setup time, minimize the chessboard time minimize the tardiness, minimize the earliness. It is expressed by the following equation:

$$\text{Objective function} = \alpha_i \cdot \sum_{i-1,1} \text{SU} + \beta \cdot \sum_{i-1,1} \text{CB} + \sum_{i-1,1} \gamma_i \text{ET} + \sum_{i-1,1} \delta_i \text{TT} \quad (4)$$

- SU = time lost for the set up the line to start to assemble the lot i [day]
- CB = time lost for the chessboard to start to assemble of lot i [day]
- ET = earliness compared with the due date [day]
- TT = tardiness compared with the due date [day]
- α_i = penalty rate for the setup the line where lot i will be assembled [€/day]
- β_i = penalty rate for the time lost in the chessboard between the lines [€/day]
- γ_i = penalty rate of earliness in assembly of lot i [€/day]
- δ_i = penalty rate of tardiness in assembly lot i [€/day]

Selection

The selection process creates a new population for the next generation, which is selected among the parents and their children. We used a common *roulette wheel* selection (Goldberg 1989). The new population is selected with respect to the probability distribution based upon the fitness values.

Crossover

We use the two-point crossover, in order to generate four children for each selected couple of parents. The crossover operator is applied with a probability P_c ($P_c = \text{crossover probability} = 0.8$). As illustrated in figure 4, two points between two jobs are randomly selected for dividing one parent. For the first two children, the jobs outside the selected two points are kept from one parent to the child, and the other jobs are placed in the order they appeared in the other parent. For the other two children, the job inside are kept and the other jobs are placed in the same manner.

Parent1	1	2	3	4	5	6	7	8
Parent2	4	8	5	6	2	1	7	3
Child 1	1	2	4	5	3	6	7	8
Child 2	4	8	2	5	6	1	7	3
Child 3	8	6	3	4	5	2	1	7
Child 4	1	3	5	6	2	4	7	8

Figure 4 – The Two-Point Crossover

Mutation

The strings generated by the crossover operator are modified with the second genetic operator, the mutation. We applied the shift change mutation with a probability P_m ($P_m = \text{mutation probability} = 0.1$). As illustrated in figure 5, two jobs inside the parent are randomly selected. A job at one position is removed and put at another position and the other jobs shift to the right.

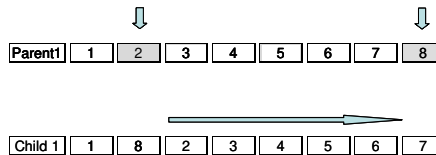


Figure 5 – The Shift Change Mutation

Termination

The process is usually repeated for the fixed number of generations. We designed the scheduling software in order the generations stop after 3 minutes. This computation time is aligned with the requirement of the Aprilia group, where scheduling is carried out daily up to date.

CONCLUSIONS

The software prototype has been implemented and tested on a fourth-monthly master production schedule (MPS). The short elaboration time and the robustness of the final assembly plans, obtained during the test-stage, confirm that the choice of GAs was right. The main objective of the prototype, i.e. a feasible FAS that takes into account the constraints coming from a real manufacturing planning model, has been fully achieved. Furthermore, significant savings in terms of time loss, in particular in the chessboard shiftings, have been attained, thus ensuring a complete adherence to the delivery dates imposed as input data. The other advantages are in terms of flexibility and robustness of the plans.

The limits of this research lie in the fact that the model and the prototype have been developed and tested on a single case study. Despite this consideration, the model can be easily adapted to several manufacturing companies. In fact we considered a multi-objective genetic algorithm for the scheduling problem with four objectives: minimize the setup time, minimize the chessboard time, minimize the tardiness, and minimize the earliness. The objective function can easily modified by changing the penalty rate of a single objective to zero; for instance the penalty rate of the chessboard time which is typical of the Aprilia group. Thanks to the generalization of the assumption during the design-stage, the software prototype can easily become a general purpose and be customizable to several manufacturing companies. The high applicability of the prototype is widely exploitable, especially where the management of the work teams has a large impact on the productive performances.

REFERENCES

- Chaudry, S.S. and Luo, W. (2005), "Application of genetic algorithms in production and operations management: a review", *International Journal of Production Research*, Vol. 43, No. 19, pp. 4083-4101.
- Davis, L. (Ed.) (1991), *Handbook of Genetic Algorithm*, Van Nostrand Reinhold, New York, NY.
- Della Croce, F., Tadei, R., and Volta, G. (1995), "A genetic algorithm for the job shop problem", *Computers Operations Research*, Vol. 22, No. 1, pp. 15-24.
- Goldberg, D.E., (1989), *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading, MA.
- Hall, R.W. (1991), *Queuing Methods for Services and Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ.
- Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Kim, B. and Kim S. (2002), "Application of genetic algorithms for scheduling batch-discrete production system", *Production Planning and Control*, Vol. 13, No. 2, pp. 155-165.
- Leu, Y., Matheson, L.A. and Rees, L.P. (1996), "Sequencing mixed-model assembly lines with genetic algorithms", *Computers & Industrial Engineering*, Vol. 30, No. 4, pp. 1027-1036.
- Levine, D. (1997), "Genetic Algorithms: a Practitioner's View", *INFORMS Journal of Computing*, Vol. 9, No. 3, pp. 256-259.
- Metaxiotis, K. and Psarras, J., (2004), "The contribution of neural network and genetic algorithms to business decision support: academic myth or practical solution? ", *Management Decision*, Vol. 42, No. 2, pp. 229-242.
- Murata, T., Ishibuchi, H. and Tanaka, H. (1996a), "Multi-objective genetic algorithm and its applications to flowshop scheduling", *Computers & Industrial Engineering*, Vol. 30, No. 4, pp. 957-968.
- Murata, T., Ishibuchi, H. and Tanaka, H. (1996b), "Genetic algorithm for flowshop scheduling problems", *Computers & Industrial Engineering*, Vol. 30, No. 4, pp. 1061-1071.
- Pongcharoen, P., Hicks, C., Braiden, P.M. and Stewardson, D.J. (2002), "Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products", *International Journal of Production Economics*, Vol. 78, pp. 311-322.
- Ponnambalam, S.G., Aravindan, P. and Chandrasekaran S. (2001a), "Constructive and improvement flowshop scheduling heuristics: an extensive evaluation", *Production Planning and Control*, Vol. 12, No. 4, pp. 335-344.
- Ponnambalam, S.G., Aravindan, P. and Sreenivasa, R. (2001b), "Comparative evaluation of genetic algorithms for job-shop scheduling", *Production Planning and Control*, Vol. 12, No. 6, pp. 560-574.
- Ponnambalam, S.G., Ramkumar, V. and Jawahar, N. (2001c), "A multiobjective genetic algorithm for job shop scheduling", *Production Planning and Control*, Vol. 12, No. 8, pp. 764-774.
- Reeves, C.R. (1995), "A genetic algorithm for flowshop sequencing", *Computers Operations Research*, Vol. 22, No. 1, pp. 5-13.
- Ruiz, R. and Maroto, C. (2005), "A comprehensive review and evaluation of permutation flowshop heuristics", *European journal of operational research*, Vol. 165, pp. 479-494.
- Ruiz, R., Maroto, C. and Alcaraz, J. (2006), "Two robust genetic algorithms for the flowshop scheduling problem", *Omega*, Vol. 34, pp. 461-476.