JOBS SEQUENCING IN INDUSTRIAL PLANTS BY MULTI-OBJECTIVE OPTIMIZATION BASED ON A SYSTEM OF AUTONOMOUS GENETIC AGENTS

Roberto Mosca, Filippo Queirolo and Flavio Tonelli
Department of Production Engineering – Savona
University of Genoa
Via Cadorna, I-17100
Savona, Italy
E-mail: filippo.queirolo@dip.unige.it

KEYWORDS

Job sequencing, simulation optimisation, parallel genetic algorithms, evolutionary multi-objective optimisation, autonomous genetic agents

ABSTRACT

Simulation Optimisation is one of the hottest topics in the M&S area. Evolutionary Computation has been shown to have great synergy with simulation both for fitness assessment and constraints description.
In this paper, the Authors discuss a general architecture for job sequencing in a case of multi objective simulation optimisation by agent-directed simulation. Autonomous genetic agents have been used.

INTRODUCTION

This paper addresses a real case involving a manufacturing company producing and commercialising mineral water and soft drinks.
The firm is interested in the enhancement of the:

- weekly production rate, measured in terms of the number of bottles produced over a certain period;
- service level, evaluated by the delay with respect to the due dates.

Notice that the maximisation of the weekly production rate immediately implies to minimise the total set-up time and therefore to determine the optimal production sequence.
The firm plan to achieve these goals by a progressive process of re-organisation, whose first phase has been successfully completed. In this paper the Authors focus on the improvement of the planning, scheduling and sequencing process. This is a real world case of innovation based on information technology and processes re-engineering applied to production management in an industrial plant.
In a previous work [Mosca et al. 2002], the Authors presented the results achieved during the first phase of this re-organisation process; now they discuss the extension of the architecture of the scheduling system and show preliminary results.
In the next section, the Authors briefly present the production process (three production lines alternatively working) and detail the assumptions and the process modelling stages: it is shown that the proposed case belongs to the class of independent job sequencing problems. A short argumentation is then reported in order to formulate it as a Simulation Optimisation (SO) problem over a non-parametric input domain and to shown that Evolutionary Computation (EC) represents the technique most commonly agreed to be suitable for this kind of SO problem. The relevance of the proposed problem has been addressed both in the area of scheduling and sequencing [Pinedo 2002] and SO [Jacobson and Schruben 1989].
The various stages followed during the development of the simulation model are then summarised. This process has been structured according to Williams and Narayanaswamy [Williams and Narayanaswamy 1997], who further refer to [Banks and Gibson 1996], following the key tasks needed for a successful simulation analysis.
The Authors then state that optimisation of the set-up time and earliness-tardiness is required, to be estimated by simulation on the basis of the job sequencing input. This is a Multi Objective Optimisation (MOO) problem.
In order to justify the proposed architecture, with respect to previous literary works, an extreme synthesis of the scientific work developed in the area of MOO by using EC is provided and a special focus (paragraph 5) is dedicated to MOO by Parallel Genetic Algorithms (PGA), in particular to Coarse-Grained Genetic Algorithms (CGGA).
The Authors focus on the algorithm selection process (a sort of CGGA have been adopted), the software architecture (autonomous agents communicating by blackboard protocol) and to the modification of previous PGA paradigms (introduction of an agent for the subpopulations merging and selection of the new individuals). The paper ends with some preliminary results and considerations about the next stages of the projects.

DESCRIPTION OF THE PROBLEM ENTITY

The proposed industrial case refers to a manufacturing process for mineral water and soft drink bottling, with three semi-automated production lines producing twenty-one different product categories. The equipment is connected by rolling tapes and this is a flowshop process. Resources are requested on each line for producing and the number of operators required varies: 8 employees for line 1 and line 2; 10 employees for line 3. The production-aimed resources pool counts 10 persons: simultaneous production by two lines is unfeasible. Equipment allows to produce different

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)

ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

sizes of products (i.e. 500ml, 750ml and 1000ml), but long setup times are required (2-6 hours) for each format change. Production is organised according with a make to stock policy. Nevertheless due dates should be considered, especially in the case of large batches requested by major customers on the basis of an open contract and to the requirement (i.e. product code, date and quantity) of the Planning Department stated in the master production schedule [Aloi et al. 2002]. As a result, from the production manager point of view each batch has a proper product specification, required quantity (i.e. number of pallet) and due date (i.e. day), either derived by direct customer needs or Planning Department requirements.

Producing by line 1 and 2, two persons would remain inactive; thus preforms insuffling or setup activities on another production line can be assigned to them. Notice that performing the set-up of one of the inactive lines allows reducing the total completion time, by partial or full hiding the setup times,.

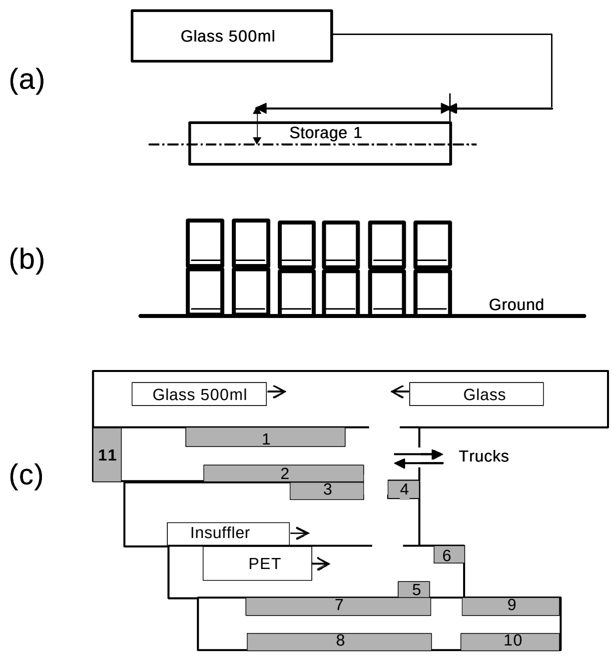

Figure 1: (a) Material Handling from a Production Line to the Pre-assigned Storage Area. (b) Schematic View of a Storing Block. (c) Production Plant.

Lot sizing is performed according to the shift length: a job always starts at the beginning of a shift and ends at the end of (that or a subsequent) shift. In this way, stops due to set-up operations are reduced. Finished goods are then stored in a pre-assigned area before the pick-up for distribution and commercialization. As usual in several SME involved in the large consume market, storage represents a critical issue, especially when trucks arrivals and departures are affected by high variability. The saturation of the storing area should be taken into account during the scheduling process.

ASSUMPTIONS AND MODELING PHASE

Because of the features of the production processes, the scheduling problem crystallises in a job sequencing problem. Indeed, allocating a product on a line means also to define the starting and completion times. It means that the goal is determining the sequence that optimizes the performance indices. One can indeed think of each line as a single block, which transforms raw materials in finished goods by collapsing the whole bottling process in a single black box. In this prospective, the parameters of each line can be estimated on the basis of the production time series of that specific line.

Sequencing is an integer optimization problem on bounded domain (i.e. an integer constrained optimisation problem), that has been shown to be NP-hard [Du and Leung 1990], even in case of total tardiness minimization only.

For these reasons (i.e. complexity and pivotal role in scheduling heuristics) and because the proposed industrial case belong to real world domain, the Authors consider it relevant from a scientific point of view.

IS THIS A SIMULATION OPTIMISATION PROBLEM?

With the great incidence of simulation modelling over a huge number of areas, it has been essential to extend the scope of traditional optimisation to include simulation domain. Simulation Optimisation (SO) is the scientific field that provides a structured approach to determine the optimal values for the input (operating) parameters of a simulation model, according to a performance measure. Simulation models can indeed be used as the objective function and/or constraints functions in optimising stochastic complex systems.

Azadivar [Azadivar 1999] (first formulation pag. 94) and Joshi et al. [Joshi et al. 1996] define SO as the problem of finding an input vector **X** that minimises:

$$\boldsymbol{f}(\boldsymbol{X}) = E[\boldsymbol{r}(\boldsymbol{X})] \qquad (1)$$

subject to the following sets of stochastic and deterministic constraints:

$$\boldsymbol{sc}(\boldsymbol{X}) = E[\boldsymbol{s}(\boldsymbol{X})]$$
$$\boldsymbol{dc}(\boldsymbol{X}) < 0$$

where $\boldsymbol{f}$ and $\boldsymbol{sc}$ respectively are the unknown expected values of the objective function(s) and the set of stochastic constraints evaluated on the basis of the random vectors $\boldsymbol{r}$ and $\boldsymbol{s}$, and $\boldsymbol{dc}$ is a set of deterministic constraints.

Jacobson and Schruben [Jacobson and Schruben 1989] shown that this problem is hard to solve.

Objective function(s) evaluation by simulation leads to enormous advantages, especially if does not exist any analytical expression of the goal function and/or the objectives function(s) and the constraints are stochastic functions of the deterministic decision variables. Moreover simulation allows reaching relevant improvements in the description and characterisation of the problem (e.g. accurate representation of the physical and logical constraints).

On the other side, using simulation as an aid for optimising presents specific challenges, such as those related to the optimisation of complex and highly non-linear functions. Further, the efficiency of the optimisation algorithm is more crucial, since objective function evaluation is performed by simulation run instead of calculation of an analytical expression.

SO techniques are generally classified on the basis of the nature (continuous/discrete/non parametric) and structure (quantitative/qualitative) of the input space. Table 1 shows a common classification of SO techniques; Table 2 focuses on the commonly used techniques in each one of the four classes previously highlighted. Notice that SO techniques have been inserted in the most suited class: e.g. Response Surface Methodology or Nelder-Mead Method have been largely used also in discrete domain since 80's [Mosca and Giribone 1985; Mosca et al. 1986a and 1986b].

Table 1**:** Short Classification of SO Techniques

Input Parameter	**Input Structure**	**Reference of Surveys**
Continuous	Quantitative	[SWISHER2000]
Discrete Small	Quantitative	[CARSON1997]
Discrete Large	Quantitative	[CARSON1997]
Non Parametric	Quali/quantitative	[Azadivar1999]

In this mind, when any matter of sequencing and scheduling problems [Pinedo 2002] arise in complex systems and reaching an accurate description of all (relevant) production constraint is required, a SO problem over a non-parametric input domain can be stated. According with Table 1, these problems can be successfully faced by evolutionary computation. With particular attention to the real case described in this paper the problem formulation reported below is appropriated. Notice that it refers to a complex production system and requires that all the relevant production constraints are fully considered.

It is required to determine the weekly schedule (i.e. $\boldsymbol{X}$, the non-parametric input of the simulation) that minimises the expectation of the multi goal function $\boldsymbol{f}$ over a set of replicated simulation runs:

$$\boldsymbol{f}(\boldsymbol{X}) = (E[\boldsymbol{f}_1(\boldsymbol{X})], E[\boldsymbol{f}_2(\boldsymbol{X})]) = \\ = (E[\boldsymbol{r}_1(\boldsymbol{X})], E[\boldsymbol{r}_2(\boldsymbol{X})]) \quad (2)$$

subject to:

- the availability of storing positions, that is definitively due to the stochastic arrivals and departures of the trucks. They are ***sc*** in Equation (1).
- the physical constraints of the production plant and the resource pool. They are ***dc*** in Equation (1).

The goal function **f** can be informally defined as follow:

- f1 is the total set-up time, observed over a week.
- f2 is the service level, evaluated by the total earliness and tardiness penalty (ETP) [Baker1995]:

$$ETP = \sum_{j=1}^{\text{N of Items in Advance}} E(j)\cdot Qt(j)\cdot EW(j) + \\ + \sum_{k=1}^{\text{N of Items in Delay}} T(k)\cdot Qt(k)\cdot TW(k) \quad (3)$$

where E (T) is the earliness (tardiness) of the j-th batch in advance (in delay), measured by time units; Qt is the required quantity of product for a batch; EW (TW) is the penalty due for a unit of product stored in the assigned position in advance (in delay) of a unit of time regarding to its due date;

On the basis of the argumentations and references provided in this section, this problem can formulated as a multi objective SO over a non-parametric domain. The Authors faced it by Evolutionary Multi Objective Optimisation (EMOO).

Table 2**:** SO Techniques

Input	**Mostly Adopted Techniques**
Continuous	Gradient Approaches; Response Surface Methodology; Stochastic Approximation; Nelder-Mead Method; Hooke-Jeeves Method
Discrete Small	Importance Sampling; Ranking and Selection; Multiple Comparison
Discrete Large	Evolutionary Computation; Evolutionary Strategies; Simulated Annealing; Tabu Search;
Non-Param.	Evolutionary Computation

EVOLUTIONARY MULTI OBJECTIVE OPTIMISATION

EC mainly refers to Genetic Algorithms (GAs). In this paper, the Authors avoid any description both of the basic principles of EC and evolutionary mechanisms (i.e. selection, crossover or mating, mutation of the individuals of the population). See [Goldberg 1989] for a general review about GAs and [Mosca et al. 2002] for a detailed explanation of the specific crossover and mutation mechanism adopted for the evolution of both the populations in this application.

Since the proposed case belongs to the class of multi objective SO problems over a non-parametric domain, EMOO is the appropriate paradigm to be adopted as optimisation technique.

MOO requires determining a set of Pareto Optimal solutions instead of a single optimal configuration. "A solution is said Pareto optimal, or non-dominated, if starting from that point in the design space, the value of any of the objective function cannot be improved without deteriorating at least one of the others" [Cardon et al. 1999].

EC seems particularly suitable to solve MOO problems because evolutionary algorithms simultaneously deal with a set of possible solutions, which allows finding an entire set of Pareto optimal individuals in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques [Coello 1999].

Even if no reference has been published, explicitly addressing the problem of Pareto Otimiality in SO, large literature is available in the field of EMOO with respect to this problem. Regardless to the approaches that leads to a degenerate MOO two main concepts can be identified in EMOO, such as aggregating functions [Fonseca and Fleming 1997] or Target Vector Approaches [Coello 1999].

A first research direction aims to determine optimal solutions by directly minimising the vector of objective functions in a Pareto sense (see figure 2). This is the case of MOGA [Fonseca and Fleming 1993], NSGA [Srinivas and Deb 1993], NPGA [Horn and Nafpliotis 1993].

Figure 3 shows the expected behaviour of the proposed approach: by selecting the sub-populations recombination interval (i.e. the migration interval in the Coarse-Grained Parallel Genetic Algorithms literature [Cantù-Paz 1998]) it is possible to drive the optimisation process through several parallel directions in order to determine the searched set of Pareto optimal solutions. Notice that GAs are inherently parallel.

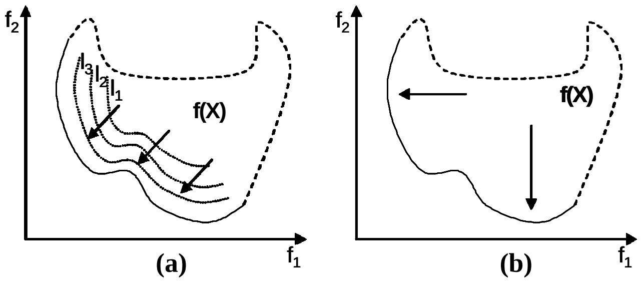

Figure 2 : (a) Pareto Optimality Increases from l_1 to l_3.
(b) Pareto Optimality is Reached by the Minimisation of each Single Objective Function

The second direction, which at the basis of the proposed approach, is largely discussed in [Grefenstette 1984] and in [Shaffer 1985], presenting Shaffer's VEGA. The idea behind VEGA is based on the optimisation of each single objective function by the division of the initial population in two (or more depending on the number of functions) sub-populations. Each sub-populations is generated by selecting individuals according with one of the objective functions; the sub-populations are then shuffled together in order to obtain a population to be mated and mutated in a single GA. See Figure 3.
A main problem of VEGA is speciation that is the excellence of some individuals on specific aspects of performance. This leads to the evolution of "species" within the population due to the mono-directionality of the selection mechanism. A crucial point is that "middling" (i.e. individuals with whole acceptable performance, but not outstanding for any objective functions) can be prevaricated by specialised individuals, avoiding the essential evolution of compromising solutions.

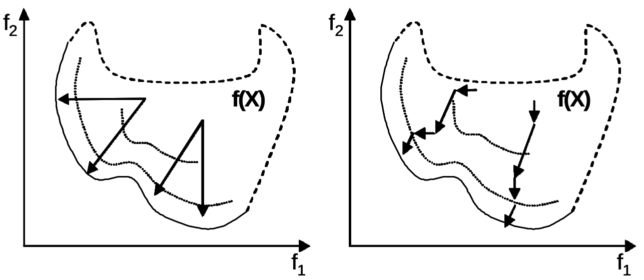

Figure 3 : Expected Behaviour of the Proposed Approach

In this paper VEGA approach to EMOO is extended to the case of multiple parallel genetic algorithms and a specific mechanism for sub-populations recombination has been implemented. Figure 4 shows a schematic representation of the proposed concept. Starting from a random population, two genetic algorithms perform the evolution of the individuals. Within each GA, individuals are selected according with a pre-assigned objective function or with a mixture of some of them. In this way it would be also possible to implement different crossover and mutation mechanisms in different GA, according with the specific issues of the objective function to be optimised by the GA.
Even if this approach could present some advantages, it is still affected by the some behaviour observed in the Schaffer's VEGA. In this mind, a periodic recombination of the sub-populations has been implemented in order to allow the selection of the individuals according to a Pareto ranking criterion.

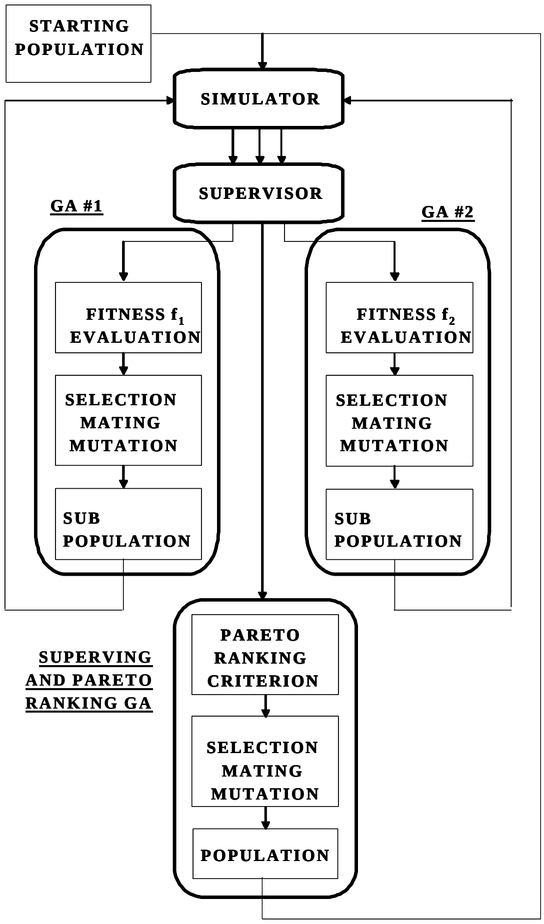

Figure 4 : the Proposed EMOO Architecture

Moreover a really high degree of scalability and flexibility have been reached:

- if *n* objective functions should be optimised, to add a number of GAs is sufficient that enables to reach the desired pool of problem solvers;
- if one of the function to be minimised would requires highly customised mechanism of optimisation, that mechanism should be implemented in the GA, assigned to that function.

MIGRATION RATE AND RECOMBINATION

The problem of periodic recombination of two or more sub-populations is typical of Parallel Genetic Algorithms (PGAs). PGAs have been largely studied since their efficiency in solving combinatorial problems: by dividing the population, the evolution can be speed up, with regard to the communication among the GAs. The determination of the correct migration rate is still an open problem and several recent papers are mainly referred to formally investigate the relation between the selection pressure and the migration rate [Cantù-Paz 1999].

In this work, the Authors implemented a supervising entity (later defined as an agents' supervisor) that is responsible for the migration of the individuals of each sub-population in the large populations: the interval migration rate is defined by a setting parameter that specifies the number of generation between two subsequent migration.

Selection is performed by MOGA's criterion of Pareto optimality: each individual are ranked on the basis of the number of chromosomes by which it is dominated. This choice is justified by the critical comparison provided in [Coello 1999] and by the argumentation of Golberg and Deb [Golderg and Deb 1991] about the rapid converge (even premature) of the algorithm. The Authors consider this property has a favourable one since the proposed framework involves the alternation of one-dimensional and Pareto ranking EMOOs.

DEVELOPMENT OF THE SIMULATION MODEL

The *scope of the simulation model* contains: i) the bottling processes considered as single operations each one performed inside a block; ii) the human resources; iii) the line stops due to machine breakdowns; iv) the material handling and finished goods transportation inside the plant; v) the storing operations and constraints due to the real volume of the storing areas. On the basis of a cost effective evaluation (for similar approaches see [Williams and Narayanaswamy 1997] or [Jayaraman and Agarwal 1996], the Authors didn't model the availability of raw materials, the supplying and distribution chain (i.e. the logistic, transportation and supplying issues), human resources behaviour and they didn't distinguish among the different breakdowns or production interruptions.

This model was derived by the integration of two (already validated) conceptual models, [Mosca et al. 2002] and [Nan et al. 2002]. The Authors coded it by a programming language in order to have a discrete event simulator [Banks et al.1995]. Indeed, since an EMOO approach had been selected, they avoided using a commercial simulation tool, such as Automod and Autosched (optimised by AutoStat) or Micro Saint (optimised by OptQuest), even if they had been already successfully tested in the case of SO since 90's (respectively [Carson 1996] and [Drury and Laughery 1996]). In this way the complete control was reached both on the simulation model and its interfacing with the optimiser; it was crucial for the further development of the scheduling system and to preserve future extension and scalability.

The *choice of a programming language* has been driven by a need for completely manageable software allowing deep integration with the optimisation algorithm (see corresponding paragraph). Considering some specific requirements of the firm (i.e. platform independency, and remote usability and controlling of the software) and because of the suitability for the implementation of agent frameworks and architectures (Bigus framework [Bigus and Bigus 2001], Madkit [Gutknecht and Ferber 2000], Zeus or Jade platforms), the Authors considered Java has the most suitable programming language for the proposed application.

The *objectives* of the proposed simulation model have been preliminary states in a previous paragraph by formulating the optimisation problem: by the simulator the Authors want to assess the performance of a scheduling (i.e. the job sequence provided as input to the simulator), respecting the production constraints. The *output data* measured in order to evaluate the performance of the scheduling provided in input to the simulator are the earliness and tardiness penalties and the total set-up time. Indeed, they are representative of the production rate of the plant (by the total set-up time), the cost effectiveness of the schedule (by time advance with respect to the due dates and total set-up time), and service level (by time delay).

According to [Dileepan 1993], the earliness and tardiness penalties have been calculated by (1):

$$ETP = \sum_{d-C_i>0} b_i(d - C_i) + \sum_{C_i-d>0} b_i(C_i - d) \qquad (2)$$

where the subscript i refers to the i-th job and C_i is the completion time; d is the assigned due date; b_i is the earliness weight; and a_i is the tardiness weight. Notice that generally the earliness and tardiness weights are cost evaluators difficult to be estimated, even with the support of expert. Indeed it is strongly dependent on various voices of the balance sheet and on the organisation of the firm.

Data collecting has been performed in strict collaboration with practitioners, being known that it is a critical task in simulation [Amico et al. 2000]. Data validation has been performed during all the collecting process and initially unavailable data has been estimated on the basis of the standards (adopted by the production department) and the experience of the production personnel.

No warm-up period is here calculated since this is the case of terminated simulation [Banks et al.1995] over a time [0; T_w], where T_w equals a working week. The Authors searched for a trade off between the confidence interval (estimated according with the MSpe in [Mosca et al. 1982]) and the number of replications to be performed. Since the goal function **f** is a vector of objective functions, the analysis of the number of iterations has been performed by the total weekly production (i.e. number of pallets). Figure 5 shows the amplitude of the interval of confidence versus the number of simulation runs. For each value of simulation runs, 10 replications have been performed, using different random stream. As a results, the average amplitude (solid line) seems to stabilise when 14 simulation runs are performed. Notice that the vertical lines progressively reduce until experiments are performed over 11 simulation runs; it provides an estimation of the standard deviation of the amplitude of confidence interval. According with the accuracy required for the specific testing case, the Authors estimate each value of the goal function by running 6 simulation runs. In this way the experimental error can be reasonably considered lower than 1.1% (basis the weekly production).

Since the Authors previously developed and tested the conceptual models now integrated in the described simulator, the Verification and Validation (V%V) phase especially focused on the testing and of the computerized model by comparing the simulation model with the validated ones. Moreover, a full validation of the system has been performed by the following V&V techniques [Sargent 1999]:

- structured walkthrough of the model logic;
- Operational Graphic;
- Predictive Validation, quantitative statistical comparison of the estimated working rates of the equipment (i.e. the utilization rates) with the corresponding values observed during the simulation;
- Deep analysis of the simulation trace;
- Turing Tests.

Last two stage of the V&V process have been performed in strict collaboration with the production personnel.

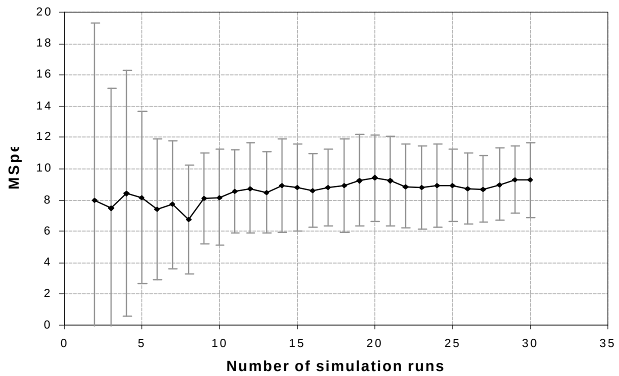

Figure 5: Amplitude of the Interval of Confidence versus the Number of Simulation Runs. For each Number of Simulation Runs, 10 Replications with different random stream have been performed

WHY AN AUTONOMOUS GENETIC AGENTS ARCHITECTURE?

Referring to Prof. Ören's Invited Paper at the St. Petersburg Workshop on Simulation [Ören 2001c], the Authors implemented the proposed concept in an agent software architecture. As an evolution of the application of artificial computation in statistics, software agent is emerging as a key research area. Especially considering simulation, software agents sounds as a promising paradigm for agent-support in simulation, which is design of experiments, simulation-based optimisation, and analysis of simulation results [Ören 2001a and 2001b].

Genetic Agents represents hence an important paradigm since they are suitable for optimising complex system over a non parametric domain (one of the hottest research area [Law and McComas 2000]) and have cognitive abilities such as autonomy, goal processing and input evaluation.

Notice that according the whole argumentation provided in this paper, these exciting features always require coupling each genetic agent with a validated simulation model.

The agent architecture presented in this paper uses a blackboard communication protocol. Coherently with its primary aim [Erman et al. 1980], it is used to share indirectly data by a common knowledge exchange place.

An agents' supervisor monitors the evolution of the genetic agents and decide for the recombination of the sub-population according with the framework previously discussed.

PRELIMINARY RESULTS

The proposed architecture has been preliminary tested on the real problem presented at the beginning of this paper. Some results have been obtained. They showed a certain instability of the average value of the fitness function, probably due to the re-combination of multiple research approaches. Nevertheless the definitive performance results in a significantly improvements of the production schedule with respect to the previous scheduling system: the total set-up time has been reduced by 4% and the earliness/tardiness function records savings higher than 5.8%. The computational effort is significantly higher than in [Mosca et al. 2002]: even if the number of simulation runs to be performed in this case (i.e. 5) is lower than the correspondent value adopted in the previous simulations (i.e. 11 runs), more than twice the time needed for standard GA is now required.

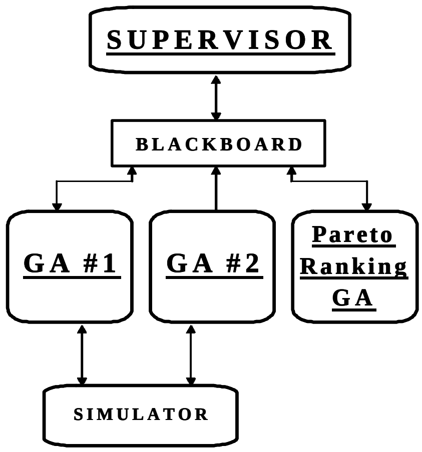

Figure 6: the Proposed Agents Architecture

CONCLUSIONS AND FUTURE WORK

In this paper a general architecture has been introduced starting from a real industrial problem. The proposed logical scheme uses EMOO in a case of SO over a non-parametric domain. Two genetic agents and a supervisor communicate by a blackboard protocol in order to implement collaborative problem solving based on evolutionary algorithm. Simulation has been shown to be crucial in EC both for fitness assessment and constraints description. As a result, agent-directed simulation optimisation has been discussed and architecture proposed. Novelty refers to the extension of Shaffer's VEGA approach, in order to mitigate the "middling" effect, and to the structured formulation of a scheduling problem as a multi objectives SO case.

Future work will be mainly devoted to investigate the relationship among the different parameters and to implement an algorithm for novelty detection in the sub-populations in order to adaptively decide whether combine (i.e. perform migration of) the individuals.

REFERENCES

Aloi U; R. Mosca; F. Queirolo; and F. Tonelli. 2002. “Strategic Planning and Production Control: Deriving an Useful Master Production Schedule from Sales Forecasts”. *Proceedings of Summer Computer Simulation Conference*. (SCSC'02). San Diego (CA)

Amico V.; R. Guha; and A.G. Bruzzone. 2000. "Critical Issues in Simulation", *Proceedings of Summer Computer Simulation Conference*, Vancouver, July

Azadivar F.. 1999. “Simulation optimization methodologies”. *Proceedings of the 1999 Winter Simulation Conference*, 93-100.

Baker K.R.. 1995. “Elements of Sequencing and Scheduling”. Amos Tuck School of Business Administration. Dartmouth College, Hanover, New Hampshire.

Banks J.; J.S. Carson; and B.L. Nelson. 1995. *Discrete Event System Simulation. Second Edition*. Prentice Hall, Upper Saddle River, New Jersey

Banks J. and R.R. Gibson. 1996. “Getting started in simulation modelling. Industrial Engineering Solutions”. 28(11):34-39.

Bigus, J.P. and J. Bigus. 2001. *Constructing Intelligent Agents Using Java* John Wiley & Sons

Cantu-Paz E.. 1998. “A survey of parallel genetic algorithms. Calculateurs Paralleles”, Reseaux et Systems Repartis. Vol. 10, No. 2. pp. 141-171. Paris.

Cantú-Paz E.. 1999. “Migration policies, selection pressure, and parallel evolutionary algorithms”. Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference

Cardon A.; T. Galinho; and J.P. Vacher. 1999. “A Multi-Objective Genetic Algorithm in Job Shop Scheduling Problem to Refine an Agents' Architecture”, In *Proceedings of EUROGEN'99*, Jyväskylä, Finland. University of Jyváskylä.

Carson J.S.. 1996. “AutoStat Output Statistical Analysis for AutoMod Users”, *Proceedings of the 1996 Winter Simulation Conference*, 492-499

Carson Y. and A. Maria. 1997. “Simulation optimization. Methods and applications”. *Proceedings of the 1997 Winter Simulation Conference*, 118-126.

Carlos A.; and C. Coello. 1999. “An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends”, In *Congress on Evolutionary Computation*, pages 3-13, Vol. 1, Washington, D.C.. IEEE Service Center.

Dileepan P.. 1993. “Common due date scheduling problem with separate earliness and tardiness penalties”, European Journal of Operational Research 120, 375-381

Drury, C. and K.R. Laughery. 1996. “Advanced Uses for Micro Saint Simulation Software”. *Proceedings of the 1996 Winter Simulation Conference*, 510-516.

Du J. and J.Y.T. Leung. 1990. “Minimizing total tardiness on one machine is NP-hard”. Mathematics of Operational Research, 15:483--495.

Erman L.D.; F. Hayes-Roth; V.R. Lesser; and R.D. Reddy. 1980. “The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty”. ACM Computing Surveys, 12(2):213-253.

Fonseca C.M. and P.J. Fleming. 1993. “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization”. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416--423, San Mateo, California, Morgan Kauffman Publishers.

Fonseca C. M. and P.J. Fleming. 1997. “Multiobjective Optimization”, In *Handbook of Evolutionary Computation*, pages C4.5:1-C4.5:9. Institute of Physics Publishing and Oxford University Press.

Goldberg D.. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison- Wesley, Reading, MA.

Goldberg D. E. and K. Deb. 1991. “A comparison of selection schemes used in genetic algorithms”. In *Foundations of Genetic Algorithms*, pages 69--93. Morgan Kaufmann, San Mateo, California.

Grefenstette J.J.. 1984. “GENESIS: A system for using genetic search procedures”. In *Proceedings of the 1984 Conference on Intelligent* Systems and Machines, pages 161-165.

Gutknecht O. and J. Ferber. 2000. “The Madkit agent platform architecture”. In *1st Workshop on Infrastructure for Scalable Multi-Agent Systems*.

Horn J. and N. Nafpliotis. 1993. “Multiobjective Optimization using the Niched Pareto Genetic Algorithm”. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.

Jacobson S.H. and L.W. Schruben. 1989. “Techniques for simulation response optimisation”, *Operations Research Letters*, 8:1-9.

Jayaraman, A. and A. Agarwal. 1996. *Simulating an engine plant.* Manufacturing Engineering 117(5):60-68.

Joshi B.D.; R. Unal; N.H. White; and W.D. Morris. 1996. “A Framework for the Optimization of Discrete-Event Simulation Models”. *Proceedings of the 17th ASEM National Conference*.

Law A.M. and M.G. McComas. 2000. “Simulation-based optimisation”. *Proceedings of the 2000 Winter Simulation Conference*: 46-49

Mosca R. and P. Giribone. 1985. “The Nelder-Mead Method: A Setter Sequential Technique To Get Optimum Region In Discrete Simulation Of Complex Plants”. *Proceeding III IASTED International Conference "Modelling and Simulation"*, pp 441-446, Lugano (CH)

Mosca R; P. Giribone; and P. Guglielmone. 1986. “The steepest ascent method in the optimisation of discrete simulations of complex industrial plants: a new approach”. *International Journal of Modelling and Simulation Acta Press* – Volume 6”, n 3 pp96/101

Mosca R; P. Giribone; and P. Guglielmone. 1986. “The simplex method in the optimisation discrete simulations of complex industrial plants: theory, applications, comparison”. *International Journal of Modelling and Simulation Acta Press* – Volume 6”, n 2 pp58/64

Mosca R.; F. Queirolo; and F. Tonelli. 2002. “Job sequencing problem in semi-automated production process”. *Proceedings of the 14th European Simulation Symposium.* Dresda. Germany.

Nan P.; F. Queirolo; M. Schenone; and F. Tonelli. 2002. “Warehouse layout design: minimizing travel time with a genetic and simulative approach - Methododology and case study”. *Proceedings of the 14th European Simulation Symposium.* Dresda. Germany.

Ören T.I.. 2001. *Advances in Computer and Information Sciences: From Abacus to Holonic Agents*. Special Issue on Artificial Intelligence of Elektrik. 9:1, 63-70.

Ören T.I.. 2001. “Impact of Data on Simulation: From Early Practices to Federated and Agent-Directed Simulations”. In *Proceedings of EUROSIM 2001*, Delft, the Netherlands

Ören T.I.. 2001. “Software Agents for Experimental Design in Advanced Simulation Environments”. In *Proceedings of the 4th St. Petersburg Workshop on Simulation*, pp. 89-95.

Pinedo, M.. 2002. *Scheduling – theory, algorithms, and systems. Second edition.* Prentice Hall, Englewood Cliff, N. J.

Sargent R.G. 1999. “Validation and Verification of Simulation Models”. In *Winter Simulation Conference*, IEEE, Piscataway, NJ, 39-48.

Schaffer J. D.. 1985. “Multiple objective optimization with vector evaluated genetic algorithms. In Genetic Algorithms and their Applications”. In *Proceedings of the First International Conference on Genetic Algorithms*, pages 93--100. Lawrence Erlbaum.

Srinivas N. and K. Deb. 1993. “Multiobjective optimization using nondominated sorting in genetic algorithms”. *Technical report, Department of Mechanical Engineering*, Indian Institute of Technology, Kanput, India.

Swisher J.R.; P.D. Hyden; S.H. Jacobson; and L.W. Schruben. 2000. “A Survey Of Simulation Optimization Techniques And Procedures”. *Proceedings of the 2000 Winter Simulation Conference*.

Williams E.J. and R. Narayanaswamy. 1997. “Application of simulation to scheduling, sequencing, and material handling”. In: *Winter Simulation Conference*. IEEE, Piscataway, NJ, 861-865.

AUTHORS’ BIOGRAPHIES

Roberto Mosca is full professor of "Industrial Plants Management" and "Economy and Business Organization" and he is currently the head of Department of Production Engineering at University of Genoa. He involved in several simulation project involving both academic and industrial partners.

Filippo Queirolo is a member of DIP Research Group at University of Genoa.

Flavio Tonelli is a complex systems management researcher in the Department of Production Engineering at University of Genoa.