Genetic Algorithms for Setup Coordination in Consecutive Stages of a Supply Chain

M. Ciavotta
DIA - Università Roma Tre,
Roma, Italy
ciavotta@dia.uniroma3.it

M. Dotoli, M. P. Fanti, C. Meloni*
DEE - Politecnico di Bari,
Bari, Italy
{dotoli,fanti,meloni}@deemail.poliba.it

S. Hammadi, S. Koubaa
LAGIS - Ecole Centrale de Lille,
Lille, France
{slim.hammadi,sana.koubaa}@ec-lille.fr

Abstract *– This paper considers the use of multi-objective genetic algorithms for solving a typical production chain problem, in which two consecutive production stages have to schedule their internal work while taking into account each other's requirements. We focus on a multi-objective genetic algorithm recently proposed in the related literature, i.e. IGA (Intelligent Genetic Algorithm), comparing the solutions it yields with those obtained by two state-of-the-art genetic optimizers. A set of preliminary computational tests on the mentioned case study using industrial data indicate that IGA is a promising multi objective optimizer for typical supply chain planning and scheduling problems.*

Keywords: Supply Chain, Scheduling, Optimization, Multi-Objective Decision Making, Genetic Algorithms.

*Corresponding author.

1 Introduction

A Supply Chain (SC) may be defined as a network of autonomous or semiautonomous business entities collectively responsible for procurement, manufacturing and distribution activities associated with one or more families of related products [6]. Obviously, the global SC performance depends on the performance of all SC partners. Hence, efficient SC management is increasingly recognized to have a significant impact on accomplishing effective SC operation. The SC management decisions can be broadly classified into three hierarchical levels: strategic (long-term), tactical (medium-term), and operational (short-term and real-time), according to the time horizon of the decisions. Strategic level planning involves SC design, considering time horizons of a few years, and requires approximate and aggregate data. Tactical level planning basically refers to supply planning, which primarily includes the optimization of the flow of goods and services through a given SC network. Finally, operational level planning is short-range planning, which involves production scheduling at all plants on an hour-to-hour basis.

Indeed, the setup or changeover costs minimization is one of the drivers in the modern production management, and in the past decades there has been a significant research effort worldwide associated to it. This research presents three main streams. The first pursues technical improvements of components and equipments of production systems. The second research line is devoted to integrate production planning and operations scheduling. In this context, setup minimization, production scheduling and lot-sizing are considered in the same mathematical models. This approaches may be not always adequate - e.g. for the computational complexity associated to integrated models, and for the structure and the organization of different production systems - to cope with practical management issues. At this aim, different decomposition techinques are considered in the literature. The last research stream focalizes on the problem to minimize the setup costs in the operational level planning assuming a scheduling and a sequencing point of view.

This paper considers a typical problem in SC management at the operational level: the setup coordination of processing operations. More precisely, such a problem arises in the coordination between two consecutive make-to-order SC manufacturing stages (or the departments of a plant), in which parts are processed in given production batches, and each batch is characterized by two distinct attributes. Due to limited inter-stage buffering, the two production stages have to follow the same batch sequence. In the first stage, a setup occurs every time the first attribute of the new batch is different from the previous one. In the downstream stage, there is a setup when the second attribute of the new batch changes. The problem consists in finding a common batch sequence optimizing a set of global (or local) performance indices that assess the setup time cost. Hence, choosing the best sequence is a considerably complex optimization problem that exhibits multiple conflicting objectives.

Recent literature has shown that multi-objective optimization genetic algorithms may be successfully employed for solving supply chain planning and scheduling problems [15, 16]. In this direction, the paper considers the application of multi-objective genetic algorithms to the aforementioned problem of lot sequencing in the production industry. More precisely, we focus on the application of a new genetic algorithmic approach to a multi-criteria version of the problem, recently proposed by Koubaa and Hammadi [13]. This approach, known as Intelligent Genetic Algorithm (IGA), is based on autonomous agent chromosomes. We consider an industrial case study [1, 4] that presents different performance indices based on the number of setups of the involved stages of the SC and conduct a preliminary experimental campaign to test the IGA scheme. Two multi-objective versions of the setup coordination problem are used as test-bed to the novel genetic algorithm and the obtained results are compared to the solutions obtained by two state-of-the-art genetic optimizers [4, 16].

The paper is organized as follows. Section 2 briefly discusses the application of genetic algorithm approaches to lot sequencing in the production industry. Moreover, Section 3 provides a concise description of the IGA scheme. Furthermore, the case study is illustrated in Section 4, while Section 5 reports the computational experiments. Finally, Section 6 draws the conclusions.

2 Using genetic algorithms for setup coordination of operations

The setup coordination of processing operations is a typical problem in serial manufacturing systems or production chains [1]. More precisely, in such environments, often the lack of intermediate buffering between two consecutive stages imposes the coordination of their internal processing sequences. In this case, a considerably complex optimization problem with conflicting objectives arises, since each department aims at minimizing its own total cost (related to the attributes of the processing parts). Clearly, also a global index of performance, e.g. the sum of the costs of each department, must be accounted for in the optimization process. In particular cases, some efficient heuristic algorithms are available in the literature to solve similar sequencing problems considering only a single objective, e.g. the sum of all costs, or the maximum of the costs of each department [1, 2, 7, 14, 17], while the multi-objective case is a complex problem calling for innovative approaches [3, 15, 16]. In particular, as for many manufacturing problems [5], the nature of the problem is particularly suitable for meta-heuristic approaches.

Genetic Algorithms (GAs) are a class of heuristic search techniques inspired by the principle of survival-of-the-fittest in natural evolution and genetics. In single-objective optimization problems, the search is guided by a scalar merit function expressing the fitness of the specific solution with respect to the objectives and the constraints of the problem. Recently, interesting and successful extensions of genetic methods to multi-objective optimization problems have been developed. In these Multi-Objective GAs (MOGAs) the search goals are not expressed or aggregated in a single scalar index of quality, but rather they are considered separately so that each particular solution is associated to a vector of fitness values expressing different objectives. The technical literature describes numerous efficient MOGAs for Pareto Set estimation [4, 8, 9, 10, 11, 12], and some comparative analyses are available [18].

Recent literature has shown that multi-objective optimization genetic algorithms may be successfully employed for solving sequencing problems in production chains [4, 15, 16]. Among these, we consider notable NSGA-II and CLS-EA, both particularly suitable for applications in SC planning and scheduling.

The well-known MOGA proposed by Deb *et al.* [4] called NSGA-II uses the elitism concept and the non-dominated sorting mechanism in the main algorithmic scheme. NSGA-II is designed to store the non-dominated solutions found at any given iteration. Since the number of non-dominated solutions may become excessively large in many optimization problems, it uses a special strategy to limit memory and computational requirements. This strategy is based on the concept of crowding index of a solution belonging to the known Pareto-front providing an higher chances to be stored to solutions belonging to less crowded areas of the front. For brevity, we omit the full description of the NSGA-II, which can be found in [4], while in [16] the algorithm is adapted to solve the class of problems considered in this paper.

In [16], a variant of NSGA-II with combinatorial local search operators has been proposed. The peculiarity of this algorithm, referred as Combinatorial Local Search Evolutonary Algorithm (CLS-EA), is the use of different simple local search functions based on three search rules, which aim to increase the progress rate of the non-dominated solutions during the algorithm's run. Another characteristic of CLS-EA is the use of a selection mechanism contrasting the occurrence of multiple identical solutions in the population, with the intent to increase the overall diversity of the populations. A more detailed description of the structure and the behaviour of this algorithm is provided in [16] and is here omitted for the sake of brevity.

In this context, a novel Intelligent Genetic Algorithm (IGA) [13], proposed by Koubaa and Hammadi, is based on the concept of autonomous agent chromosomes. This feature, together with the IGA characteristic of being a multi-objective evolutionary algorithm, suggest us to employ such a novel approach for setup coordination of operations in SC. The novel genetic algorithm may then be evaluated by comparison with the established algorithms NSGA-II and CLS-EA. The next section briefly describes the main characteristics of IGA.

3 The intelligent genetic algorithm

The Intelligent Genetic Algorithm (IGA) recently proposed by Koubaa and Hammadi [13] is based on the novel concept of autonomous agent chromosomes. The key idea is to implement a realistic behavior of genetic entities and operators. In particular, the proposed multi-objective genetic algorithm tries to emulate the biological evolution concept in which the female cell (F-Cell) “chooses” the ideal candidate among all the male cells (M-Cell). Indeed, recent findings have demonstrated that this mechanism seems to be able to produce superior children according to some specific criterion (fitness).

The proposed IGA implementation [13] starts by a randomly generated population of chromosomes, although seeding operations may also be considered. The main functions characterizing IGA are selection, crossover and mutation. The selection function acts as a classification of chromosomes in F-Cells and M-Cells. The classification takes into account the performance or fitness of genetic cells. The cells with better performance indices are considered as F-Cells. The other cells are classified as M-Cells and are grouped in different sets. The crossover function is based on communications between M-Cells and F-Cells. F-Cells collect the requests of crossover from M-Cells and accept only the best of them. The mutation function operates on the children generated in the previous phase. The algorithmic scheme is characterized by a high flexibility, in order to incorporate different genetic operators and fitness functions. In particular, F-Cells and M-Cells may pursue the optimization of different functions and the biological mechanism guides the evolution to solve tradeoffs between objectives. The proposed algorithm can be straightforwardly implemented in a multi-agent development environment and appears to be suitable for applications in multi-criteria and multi-decision makers situations as those arising in supply chain management [6].

4 The problem description

The models and instances considered in this paper arise in the industrial environment described by Agnetis *et al.* [1] and Meloni [14]. In particular, we focus on the coordination issues between two stages of a production chain. Assume that these stages or departments have to produce parts that are grouped in lots that may differ as regards two attributes, called hereinafter *shape* and *color*. In particular, in the first department parts are grouped according to their first characteristic, and in the subsequent department parts are grouped according to their second attribute. In both departments, a changeover occurs when the attribute of a new part changes with respect to the same characteristic of items in the other department. Indeed, if a part exhibits a different shape from the previously produced ones, the cutting machinery must be reconfigured. Similarly, when a new color is used, the painting station must be cleaned in order to remove the residuals of the previous color. In both cases, costs are incurred in terms of time and manpower. Other important issues make the sequencing problem an extremely difficult task, e.g. different costs, precedence constraints, etc. This paper addresses the problem in its basic version, i.e. assuming that all the setup operations in each department and across departments have the same cost, and changes of the processing sequence between the two stages are not allowed. Hence, the sequencing optimization is formulated as a multi-objective problem, in which a tradeoff between different indices of performance has to be solved. Each item to be produced is characterized by its own shape and color. All items having the same shape and color form a single batch. In the first (second) department, a changeover is paid when the new batch has a different shape (color) from the previous one. Otherwise, no changeover is incurred. Since the objective is to minimize the number of changeovers, the actual cardinality of each batch is of no interest. Each given sequence of the batches results in a cost of changeovers to be paid by each department. Thus, referring to the two departments, the objectives that should be simultaneously optimized are the following:

(i) minimization of the total cost of changeovers;
(ii) minimization of the maximum paid cost of changeover;
(iii) minimization of the setups for the cutting department;
(iv) minimization of the setups for the painting department.

While the first objective corresponds to the maximization of overall utility, the second one captures more realistically the need to balance the changeover costs between the two departments. Moreover, the last two objectives represent a frequent tradeoff involving successive working centers.

The problem is formulated in detail as follows. Let $B=\{b_k,\ k=1,\dots,|B|\}$ be the problem input, i.e. a set of batches to be produced. The batches must be processed by two departments of the plant, called D_S (cutting department) and D_C (coloring department), respectively. Each batch is characterized by two attributes, say shape and color. Let $S=\{s_i\ i=1,\dots,|S|\}$ and $C=\{c_j,\ j=1,\dots,|C|\}$ denote the sets of all possible shapes and colors respectively. Therefore, each batch $b_k\in B$ is defined by a pair $b_k=(s_i,c_j)$, with $s_i\in S$ and $c_j\in C$. Moreover, if two batches $b_{k1}=(s_i,c_j)$ and $b_{k2}=(s_h,c_n)$ are in B and b_{k1} is processed immediately after b_{k2}, a changeover is paid in department D_S (D_C) if $s_h\neq s_i$ ($c_n\neq c_j$). Hence, the problem is to sequence the batches in a profitable way from the viewpoint of the number of changeovers. This means that we must find an ordering σ of the elements of B, considering the following situations.

1) If two consecutive batches $b_{k1}=(s_i,c_j)$ and $b_{k2}=(s_h,c_k)$ in σ have no attribute in common, both departments have to pay one changeover when switching from batch b_1 to b_2. We refer to this as a global changeover.

2) On the other hand, if $s_h=s_i$ ($c_n=c_j$), only department D_C (D_S) pays a changeover. This situation is called local changeover.

Now consider a given sequence σ and let $s(\sigma(q))$ and $c(\sigma(q))$ respectively denote the shape and color of the q-th batch in σ. Moreover, given two consecutive batches $b_{k1}=(s_i,c_j)$ and $b_{k2}=(s_h,c_n)$ in σ, select $\delta_{si,sh}$ ($\delta_{cj,cn}$) equal to 1 if $s_h \neq s_i$ ($c_n \neq c_j$) and 0 otherwise. For the given sequence σ we can therefore easily compute the number of changeovers incurred by each department, called $N_S(\sigma)$ and $N_C(\sigma)$ respectively, as follows:

$$N_S(\sigma) = 1 + \sum_{q=1}^{|B|-1} \delta_{s(\sigma(q)),s(\sigma(q+1))} ; \quad (1)$$

$$N_C(\sigma) = 1 + \sum_{q=1}^{|B|-1} \delta_{c(\sigma(q)),c(\sigma(q+1))} . \quad (2)$$

In our preliminary campaign of experiments, we consider two bi-objective optimization problems for the described case study. In the first problem (referred to as Case 1) the simultaneous objectives are:

(i) minimize $(N_S(\sigma)+N_C(\sigma))$, i.e., minimization of the total cost of changeover;

(ii) minimize $max\{N_S(\sigma),N_C(\sigma)\}$, i.e., minimization of the maximum paid cost of changeover.

In the second problem (Case 2), the objectives are:

(iii) minimize $N_S(\sigma)$, i.e., minimization of the setups for the cutting department;

(iv) minimize $N_C(\sigma)$, i.e., minimization of the setups for the painting department.

5 The computational results

This section describes a preliminary performance comparison between IGA and two well-known multi-objective genetic algorithms, namely NSGA-II and CLS-EA. The comparison of Case 1 and Case 2 is based on the described multi-objective sequencing problem with reference to an industrial case study first introduced in [1] concerning the production process of a furniture manufacturer. The plant produces thirteen basic product models, each to be customized to meet the preferences expressed in the collected orders.

The focus is on the coordination between the cutting and the painting departments. Although thousands of different parts are produced every week, there are only four cutting modes (indicated as A, B, C, D) associated with the shape of the items, three thickness values (in mm), and three different material types (W is natural wood, PW is plywood and MDF indicates a special composite material called medium density fiber). This results in 36 different classes (cutting classes), so that items from the same class can be processed without machine setup in the cutting department. For technical reasons, only 32 of these classes are actually produced, as described in Table I. Similarly, in the painting department items can be classified according to their color and finishing (color classes). There are seven different colors and two types of finishing, yielding 14 classes, reported in Table II.

Table I: Cutting classes.

Cutting class	Shape	Thickness	Material	Cutting class	Shape	Thickness	Material
S1	A	18	W	S17	B	40	MDF
S2	A	18	PW	S18	C	18	W
S3	A	18	MDF	S19	C	18	PW
S4	A	32	W	S20	C	32	W
S5	A	32	PW	S21	C	32	PW
S6	A	32	MDF	S22	C	32	MDF
S7	A	40	W	S23	C	40	W
S8	A	40	PW	S24	C	40	PW
S9	A	40	MDF	S25	D	18	W
S10	B	18	W	S26	D	18	PW
S11	B	18	PW	S27	D	18	MDF
S12	B	18	MDF	S28	D	32	W
S13	B	32	W	S29	D	32	PW
S14	B	32	PW	S30	D	32	MDF
S15	B	40	W	S31	D	40	W
S16	B	40	PW	S32	D	40	PW

Table II: Color classes.

Color class	Color	Finishing	Color class	Color	Finishing
C1	black	natural	C8	black	polished
C2	grey	natural	C9	grey	polished
C3	white	natural	C10	white	polished
C4	yellow	natural	C11	yellow	polished
C5	green	natural	C12	green	polished
C6	red	natural	C13	red	polished
C7	natural wood	natural	C14	natural wood	polished

Table III: Number of batches in the considered instances.

	INST. 1	INST. 2	INST. 3	INST. 4	INST. 5
Batches	251	252	252	252	282

Table IV: Common settings of the considered algorithms.

Population size	100
Encoding scheme	path representation
Stopping criteria	900 s
Fitness functions	$N_S(\sigma)$
	$N_C(\sigma)$
	$N_S(\sigma)+N_C(\sigma)$
	$max\{ N_S(\sigma), N_C(\sigma)\}$

The common settings of the different considered algorithms are summarized in Table IV. The specific parameters for NSGA-II and CLS-EA are set to the suggested values proposed in the respective references, i.e. order-based crossover and two-point crossover, both with a crossover probability p_c=0.8, and insertion mutation, both with a mutation probability p_m=0.3. Moreover, the current version of IGA is equipped with a meta-ordering crossover operator and an inversion mutation operator.

The set of test problems consists of a sample of five (over 32) instances taken from Agnetis *et al.* [1] and contains data describing 5 different weekly production plans requiring different products. For each instance, the number of required production batches is indicated in Table III.

Each of the compared algorithms tries to solve the two cases described in Section 4 of the mentioned batch ordering problem. The three algorithms are tested on runs of 15 minutes length and is launched 3 times on each instance. The evaluation of the algorithms is based on a recently proposed comparative metric for MOGAs. The metric is the *average coverage*, defined in [18] as follows. Given two sets of solutions X', $X'' \subseteq X$, the function C maps the ordered pair (X',X'') in a number in [0,1] as follows:

$$C(X', X'') = \frac{\left|\{ \mathbf{a}'' \in X''; \exists \mathbf{a}' \in X' : \mathbf{a}' \succeq \mathbf{a}'' \}\right|}{\left|\{X''\}\right|}, \qquad (3)$$

where the notation $\mathbf{a} \succeq \mathbf{b}$ indicates that **a** *covers* **b**, i.e.

$$\mathbf{a} \succeq \mathbf{b} \;\Leftrightarrow\; \mathbf{a} \succ \mathbf{b} \text{ or } f(\mathbf{a}) = f(\mathbf{b}), \qquad (4)$$

and $\mathbf{a} \succ \mathbf{b}$ indicates that **a** dominates **b**. In other words, a solution **a** is said to cover a solution **b** if and only if either **a** dominates **b** or **a** and **b** have the same fitness vector. Therefore, the overall coverage obtained with function *C* provides a clear indication of the relative quality of the compared solution sets.

Tables V to VIII report the relative coverage of the three algorithms. More precisely, in each of these tables a pair of algorithms is compared on the basis of function (3) and each row of the tables indicates which algorithm provides set *X'*. The first value in each cell indicates the coverage of the *overall non-dominated solution sets* obtained as the union of the three solutions sets (obtained in the three different runs) for each algorithm. The values in the brackets indicate the cardinality of the overall non-dominated solution set. Tables V and VI clearly indicate that, while IGA and NSGA-II exhibit comparable performances in Case 1. In the same case, neither NSGA-II nor CLS-EA dominate the performance of IGA.

Table V: Relative coverage of algorithms IGA and NSGA-II in Case 1.

	INST. 1	INST. 2	INST. 3	INST. 4	INST. 5
IGA	0 [13]	0 [10]	0 [12]	0 [10]	1 [12]
NSGA II	0 [1]	0 [1]	0 [1]	0 [2]	0 [2]

Table VI: Relative coverage of algorithms IGA and CLS-EA in Case 1.

	INST. 1	INST. 2	INST. 3	INST. 4	INST. 5
IGA	1 [13]	1 [10]	0 [12]	1 [10]	0 [12]
CLS-EA	0 [1]	0 [1]	0 [2]	0 [1]	0 [2]

Table VII: Relative coverage of algorithms IGA and NSGA-II in Case 2.

	INST. 1	INST. 2	INST. 3	INST. 4	INST. 5
IGA	0.127 [18]	0.105 [22]	0.053 [14]	0.113 [14]	0.102 [25]
NSGA II	0.333 [134]	0.682 [133]	0.714 [133]	0.357 [142]	0.560 [157]

Table VIII: Relative coverage of algorithms IGA and CLS-EA in Case 2.

	INST. 1	INST. 2	INST. 3	INST. 4	INST. 5
IGA	0.098 [18]	0.090 [22]	0.016 [14]	0.033 [14]	0.068 [25]
CLS-EA	0.444 [61]	0.636 [78]	0.999 [61]	0.786 [61]	0.720 [73]

However, the IGA algorithm exhibits lower performances in Case 2. In fact, for each test, the current version of the algorithm underperforms both NSGA-II and CLS-EA (see Tables VII and VIII). Note that the latter algorithm is equipped with problem-specific combinatorial local search procedures [16] that contribute to this results.

6 Conclusions

The paper considers a well known sequencing problem arising in supply chains. An experimental campaign is conducted to test the behavior in such a field of a new genetic optimizer, called Intelligent Genetic Algorithm (IGA). The considered case study comprises two complex multi-objective problems, defined on a set of real-world test instances, and a metric evaluating the algorithm. The obtained results show that in the first case IGA is able to perform at least at the same level (but in some cases better) than current state-of-the-art algorithms, in terms of quality of non-dominated obtained solutions. Hence, this first campaign of computational experiments indicates that the novel IGA approach is a promising multi-criteria evolutionary solver for typical problems of supply chain management. Future perspectives include the investigation of additional metrics to capture and compare different aspects of the behavior of the considered algorithms with additional tests as well as the application of the IGA scheme to some extension of the problem, e.g. to cope with different structures of setup costs and the cases in which production quantities and capacities have to be considered.

References

[1] Agnetis, A., P. Detti, C. Meloni, D. Pacciarelli. Set-up coordination between two stages of a supply chain, *Annals of Operations Research*, Vol. 107, pp. 15-32, 2001.

[2] Al-Haboubi, M.H., Selim, S.Z., A sequencing problem in the weaving industry, *European Journal of Operational Research*, Vol. 66, pp. 65-71, 1993.

[3] Ciavotta, M., Detti, P., Meloni, C., Pranzo, M., A bi-objective coordination setup problem in a two-stage production system, *Proceedings of ORP-2005 Conference, Valencia*, pp. 335-344, 2005.

[4] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182 –197, 2002.

[5] Dimopoulos, C., Zalzala, A. M. S., Recent developments in evolutionary computation for manufacturing optimisation: problems, solutions, and comparisons, *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 2, pp. 93-113, 2000.

[6] Dotoli, M., Fanti, M.P., Meloni, C., Zhou, M.C., Design and optimization of integrated e-supply chain for agile and environmentally conscious manufacturing, *IEEE Transactions on Systems, Man and Cybernetics*, part A, Vol. 36, No. 1, pp. 62-75, 2006.

[7] Fanti, M.P., Maione, B., Piscitelli, G., Turchiano, B., Heuristic scheduling of jobs on a multi-product batch processing machine, *International Journal of Production Research*, Vol. 34, No. 8, pp.2163-2186, 1996.

[8] Fonseca, C.M., Fleming, P. J., An overview of evolutionary algorithms in multi-objective optimization, *Evolutionary Computation*, Vol. 3, No. 1, pp. 1-16, 1995.

[9] Fonseca, C.M., Fleming, P. J., Multiobjective optimization and multiple constraint handling with evolutionary algorithms - part I: a unified formulation, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 28, No. 1, pp. 26-37, 1998.

[10] Gao, Y., L. Shi, P., Study on multi-objective genetic algorithm, *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Vol. 1, pp. 646 – 650, 2000.

[11] Ishibuchi, H., Murata, T., A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol.28, No. 3, pp. 392 –403, 1998.

[12] Knowles, J., Corne, D., The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization, *IEEE CEC 99, Proceedings of the 1999 Congress on Evolutionary Computation*, Vol.1, pp. 98-105,1999.

[13] Koubaa, S., Hammadi, S., Intelligent genetic algorithm based on autonomous agent chromosomes, *Proceedings of AISTA'04: International Conference on Advances in Intelligent Systems: Theory and Applications*, Luxembourg, 2004.

[14] Meloni, C., An evolutionary algorithm for the sequence coordination in furniture production, *Lecture Notes in Computer Science*, Vol. 2264, pp. 91-106, 2001.

[15] Meloni, C., Naso, D., Turchiano, B., Multi-objective genetic algorithms for a class of sequencing problems in manufacturing environments, *2003 IEEE International Conference on Systems, Man & Cybernetics*, Washington, D.C, October 5-8 2003.

[16] Meloni, C., Naso, D., Turchiano, B., Setup coordination between two stages of a production system: a multi-objective evolutionary approach, to appear on *Annals of Operations Research*, 2006.

[17] Reeves, C.R., A genetic algorithm for flowshop sequencing, *Computers and Operations Research*, Vol. 22, pp. 5-13, 1995.

[18] Zitzler, E., Deb, K., Thiele, L., Comparison of multiobjective evolutionary algorithms: empirical results, *Evolutionary Computation*, Vol. 8, No. 2, pp. 173-195, 2000.