ECAS: a collaborative multi-agent system for mixed-model assembly lines programming

M. Caridi
Politecnico di Milano, Dipartimento di Economia e Produzione
Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
Tel: +39 02 2399 2787
Fax: +39 02 2399 2700
e-mail: maria.caridi@polimi.it

Abstract

This paper presents a multi-agent model, ECAS, for sequencing a mixed-model assembly line. ECAS implements the concept of deliberative and co-operative agents: the agents contract in order to find a solution which overcomes the objectives of the single agents in order to pursue the global optimum of the system.

ECAS represents the extension of a previous model by the author, AGENT, which implements deliberative but not co-operative agents. One of the aims of this paper is to compare the results of the two models in order to evaluate the value added by co-operation. Moreover, ECAS performance is compared with other traditional approaches to the same problem.

Keywords

Production smoothing, mixed-model assembly lines, multi-agent system, distributed decision-making

1 INTRODUCTION

The topic of this paper is the application of autonomous agents theory to a particular short-term production planning problem: sequencing of mixed-model lines in order to pursue production smoothing. This issue was originally stated by Monden (1983) and has been studied for years (Kubiak, 1993) because it is considered of the keys of success of Just In Time.

K. Mertins et al. (eds.), *Global Production Management*

The model presented in this paper moves away from the traditional approach to this problem: the sequence of products on the assembly line is not determined in a centralised way but through the distributed decision-making process of many intelligent entities.

In particular, on the basis of autonomous agent theory and Multi-Agent Systems (MASs), a MAS architecture, ECAS (*Eterarchical Co-operative Agent System*), has been developed. Each autonomous agent of the architecture is a particular finished product or component or sub-component or the assembly line itself. Agents communicate among themselves through a communication protocol and negotiate in order to determine a sequence of products on the assembly line.

In this architecture, the same approach of EGCM (Miltenburg and Sinnamon, 1989) is recalled. Yet the decision of which product has to be scheduled at each step of the algorithm is not made by comparing the n values of a global objective function but through a negotiation among independent autonomous agents which pursue their own objectives (Wooldridge and Jennings, 1995).

ECAS represents the extension of a previous work by the author, AGENT (Caridi *et al.*, 1998) in that it reproduces the same architecture of deliberative agents but endows them with the power of co-operation (Doran *et al.*, 1997).

In Section 2, the problem of sequencing in mixed-model assembly lines is presented, together with the model usually adopted in literature in order to solve this problem. Section 3 deals with the description of ECAS, while in Section 4 the experimental design is described and ECAS' s performance is compared with that of AGENT and of other outstanding approaches.

2 THE MIXED-MODEL LINES BALANCING PROBLEM

The model applies to a four-level manufacturing system; the levels are numbered as follows (see also Caridi *et al.* (1998)):

- level 1 - products,
- level 2 - sub-assemblies,
- level 3 - components,
- level 4 - raw materials.

Each product (level 1) is made up of a variety of sub-assemblies (level 2), which consist of different components (level 3) which, in turn, are made up of raw materials (level 4).

Given a set of products, various sequences (*schedules*) can be generated. The aim of the model is determining a *balanced schedule*, that is a schedule producing each product concurrently (Miltenburg and Sinnamon, 1989). In this way, small batches of each product alternate on the assembly line, so that the production mix is synchronised with the market demand; as a consequence, inventories are kept small. Such an assembly line is called *mixed-model* line.

The following parameters are introduced:

- n_j : number of outputs at level j; $j = 1, ..., 4$;
- $d_{i,1}$: demand for product i at level 1; $i = 1, ..., n_1$;

- $t_{i,j,l}$: units of output i at level j required to produce one unit of product l; i = 1, ..., n_j; j = 1, ... 4; l = 1, ..., n_1 (for j = 1, $t_{i,1,l}$ = 1 if i = l and 0 otherwise).

From $d_{i,1}$ and $t_{i,j,l}$ the following quantities can be determined:

- demand for output i at level j; i = 1, ..., n_j; j = 2, ..., 4: $d_{i,j} = \sum_{h=1}^{n_1} t_{i,j,h} \cdot d_{h,1}$;
- total demand for production at level j; j = 1, ..., 4: $DT_j = \sum_{i=1}^{n_j} d_{i,j}$;
- output i at level j for each product assembled; i = 1, ..., n_j; j = 1, ..., 4: $r_{i,j} = \frac{d_{i,j}}{DT_1}$.

DT_1 products must be assembled on the final assembly line during the planning horizon; in other words, there are DT_1 consecutive decisional steps and in each step a product is assigned to the line. The schedule is represented by the variables $x_{i,1,k}$, which are the units of product i assembled during steps 1, ..., k.

The production at the lower levels can be determined as follows:

$$x_{i,j,k} = \sum_{h=1}^{n_1} t_{i,j,h} \cdot x_{h,1,k} ,$$

which expresses the units of output i at level j produced during stages 1, ..., k; i = 1, ..., n_j; j = 2, ..., 4. If production were perfectly levelled, after k stages the total output $x_{i,j,k}$ (actual output) of part i at level j should be $k \cdot r_{i,j}$ (theoretical output); equality is not always possible because $k \cdot r_{i,j}$ is not an integer. The aim of the model is finding a schedule that minimises the difference among actual output and theoretical output of every part at every level. Thus, introducing the *Squared Difference Quantity* (SDQ), which expresses the difference between the actual and the theoretical output at each step k, and the weights w_j, which express the relative importance of deviation at the various levels, the sequencing model is formulated as follows:

$$\min \sum_{k=1}^{DT_1} SDQ_k = \sum_{k=1}^{DT_1} \sum_{j=1}^{4} \sum_{i=1}^{n_j} w_j \left(x_{i,j,k} - k \cdot r_{i,j} \right)^2$$

$$x_{i,j,k} = \sum_{h=1}^{n_1} t_{i,j,h} x_{h,1,k} \qquad \forall i = 1,...,n_1, \forall j = 1,...,4, \forall k = 1,...,DT_1$$

$$x_{i,1,DT_1} = d_{i,1} \qquad \forall i = 1,...,n_1$$

$$\sum_{i=1}^{n_1} x_{i,1,k} = k \qquad \forall k = 1,...,DT_1$$

$$0 \le x_{i,1,k} - x_{i,1,(k-1)} \le 1 \qquad \forall i = 1,...,n_1, \forall k = 1,...,DT_1.$$

3 THE AGENT-BASED MODEL

The context is a mixed-model assembly line where three-levels BOM products are assembled. As a hypothesis each part of the BOM must belong to just one specific level. This model aims at solving the above-stated sequencing problem.

Agents populating the Multi-Agent System are deliberative agents (endowed with mental state and rule bases) and organised into an eterarchical architecture. In particular, the actors of the model are:

- a generator: it creates the agents and initialise them;
- a rooter: it supports the communication among the agents;
- an assembly line agent;
- n_1 product agents;
- n_2 sub-assembly agents;
- n_3 component agents.

Notice that the first and the second agents are not deliberative; they simply support the operative role of the others. Moreover, they are logical entities, whereas the others are physical entities in that they correspond to physical objects of the manufacturing system.

Each deliberative agent is endowed with a local objective that is the minimisation of the difference between its actual and theoretical output, except for the assembly line, whose objective is the minimisation of the expense of "buying" the products.

The output of the model is a sequence of finished products to be assembled on the line. The sequence is built step by step (globally DT_1 steps, as many as the products to be assembled), through a multiple-way one-step negotiation among the line and the products (Lin and Solberg, 1992) which ends when the line chooses which product will be assembled in the next step. In order to make its decision, the line evaluates the price of each product. The price is determined step by step by

each product and takes into consideration the difference between theoretical and actual consumption of the product itself and of all the other parts of the system.

At each step k of the algorithm, $k = 1, \ldots, DT_1$, the following phases take place:

1. *Portfolio determination*: at each step, each product h chooses whether to participate to the product selection. It participates only if almost N_h steps has passed since it has been assembled the last time:

   $$N_h = alfa \frac{DT_1}{d_{h1}};$$

   alfa ($0 < alfa < 1$) is set through a factorial project 2^r (if the experimental space is r-dimensional). This rule prevents from myopic (greedy) decision and is active from the beginning of the algorithm until the last steps. In particular, it is not active starting from the step $DT_1 - \max_h\{N_h\}$. In fact, in the last steps of the algorithm, the mix of products is sensibly different from the beginning (some products has finished the units to be assembled) and the rule looses its meaning of "regulator of mix". We call *active* a product that has decided to submit a bid to the line for its assembly.
2. *Product selection*:
   2.1. *Part-price evaluation*: each part i at level j evaluates its price $C_{i,j,h,k}$ for each active product h:
   $C_{i,j,h,k} = | k\, r_{i,j} - x_{i,j,(k-1)} - t_{i,j,h} |$.
   Each part calculates and sends this value to product h.
   2.2. *Price calculation:* each product h evaluates its global price $P_{h,k}$:

   $$P_{h,k} = \sum_{j=1}^{3} \sum_{i=1}^{n_j} C_{i,j,h,k}$$

   Each active product sends this value to the line.
   2.3. *Product selection*: the line ranks the bids according to the price (the first its the minimum price) and selects the first two products. The final choice between the selected products is made in the following phase (*contracting)*. The contracting phase does not take place (that is the line makes the last choice) only in the following situations:
   - only one product is active,
   - no product is active: in this case, the line forces the assembly of the product which would become active sooner.
3. *Contracting:*
   3.1. *contractors' selection:* the first selected product, say PF_1, determines the *first contractor* (*FC*), that is the part i at level j that maximises the following expression:
   $\max_{i,j} \{C_{i,j,PF1,k} - C_{i,j,PF^*,(k-1)}\}$,
   where PF* is the finished product assembled at k-th step. Let $C_{FC,\ PF1}$ be the part-price of the first contractor if PF_1 is assembled. In the meanwhile, the second selected product determines the *second contractor* (*SC*), that is the part i at level j that maximises the following expression:
   $\max_{i,j} \{C_{i,j,PF2,k} - C_{i,j,PF^*,(k-1)}\}$.

Let $C_{SC,\,PF2}$ be the part-price of the first contractor if PF_2 is assembled.

3.2. *contracting and final assembly choice*: the first contractor determines the *bid*, that is its saving in case PF_2 is assembled in the current step:
$bid = C_{FC,\,PF1} - C_{FC,\,PF2}$.
If *bid* is bigger than zero, *FC* will prefer PF_2 to be assembled. Obviously, this implies a greater expense for the line and, probably, for *SC*. As a consequence, PF_2 will be chosen only if the following relations hold:
$P_{PF2,\,k} - P_{PF1,\,k} < bid$ and $C_{SC,\,PF2} - C_{SC,\,PF1} < bid$.
If the previous relations do not hold, PF_1 is scheduled in the current step.

4 EXPERIMENTAL RESULTS

The agent model described in the previous section has been implemented in *LALO, Langage d'Agents Logiciel Object* (Marchal *et al.*, 1996) that is a programming environment which allows the development of multi-agent systems. The inter-agent communication language is *KQML, Knowledge Query and Manipulation Language* (Finin *et al.*, 1993). The performance of *ECAS* has been tested through a deep experimental design. Proofs have been carried out for a mixed-model line where six different kinds of products ($n_1 = 6$) have to be assembled.

The parameters of the design are:

- *BOM complexity degree*: the greater the number of parts of each BOM level, the more complex is BOM; then, the complexity degree is expressed by the set of numbers $n_1/n_2/n_3$; in particular during the experimentation, eight different BOM complexity degrees have been considered, as shown in Figure 1.
- *Sharing degree (SD)*: for level 2, it expresses the fraction f_1 of finished products which are parents of each sub-assembly; for level 3, it expresses the fraction f_2 of sub-assemblies which are parents of each component. The combined SDs f_1/f_2 considered during the experiments are shown in Figure 1.
- *Demand distribution*: four types of demand distribution among the finished products have been tested: P1 (homogeneous), P2 (polarised on 50% products), P3 (polarised on 33% products), P2 (polarised on one product). For each demand distribution, six different vectors of demand have been considered.

The experimental space represented in Figure 1 is made up of 192 dots. For each dot, six runs have been carried out, varying the demand values. Globally, 768 runs result.

In order to text the performance of the model, its resulting objective function value has been compared with that obtained by AGENT (Caridi *et al.*, 1998), Bautista *et al.* (1996), Monden (1983) and Inman and Bulfin (1992).

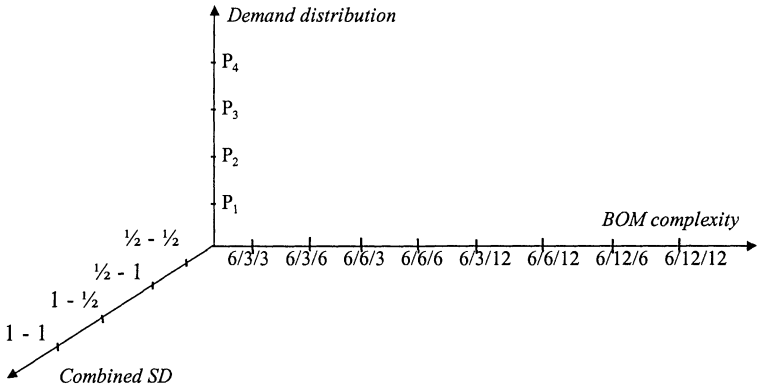

Figure 1 Experimental design space.

In particular, in each experiment the following parameter has been measured for each tested algorithm X, in correspondence of each dot of the experimental space:

$$\Delta SDQ\% = \frac{SDQ_{ECAS} - SDQ_X}{SDQ_X} \cdot 100 .$$

Notice that SDQ is the mean of the six SDQ values resulting from the six repetitions for each dot. Considered the aim of the experiments, weights w_j have been set at 1 in order to filter their influence on the performance of the tested algorithms.

ECAS performance has been compared to that of AGENT (Caridi *et al.*, 1998). Notwithstanding the implementation of co-operation (see *contracting phase*) among the agents, AGENT shows better performance than ECAS when demand is uniform (SDQ% lower than 40%). On the other hand, when demand is polarised, ECAS improves AGENT performance up to 60%, due to the exploitation of co-operation among the agents.

Moreover, the algorithm has been compared to other above-mentioned traditional algorithms. ECAS dominates the cited algorithms, except for Bautista *et al.* (1996), which is up to know the most effective as for the resolution of mixed-model line sequencing problem (SDQ% up to 80% for low complexity BOM).

The numeric results are available in Pompei and Sartore (1999).

5 CONCLUSIONS

In this paper a multi-agent system for mixed-model assembly line programming has been presented. This model is an extension of a previous one, AGENT (Caridi *et al.*, 1998) and implements the co-operation among the agents. Thanks to the co-operation, the performance of the multi-agent system improves in case of polarised demand, reducing the gap between the multi-agent approach and the traditional one, represented by Bautista *et al.* (1996), which is still the most effective. This is not surprising: Bautista *et al.* (1996) applies an algorithm which combines Dynamic Programming and Branch and Bound; thank to this approach the choice taken in each step of the algorithm can take into consideration the impact on the global optimum, preventing from greedy solutions. On the contrary, the multi-agent approach is myopic by definition (each agent performs its function aiming at its local objective).

Let us give a hint to the potential of the multi-agent approach to this problem. The model of manufacturing system represented in this paper is an ideal environment, where no lack of materials takes place, where the assembly line is always available to assembly. What happens if materials (raw materials, components and subassemblies) are not available for assembly at a generic step k? The schedule previously determined has to change since a sub-set of products can not be assembled until the materials become available (step t). ECAS can simply model this situation excluding from the set of active products the sub-set from step k to step t. Whereas, Bautista *et al.* (1996) will recalculate the sequence from step k to step t and, then, from step $t+1$ to step DT_1. Thus, the overall schedule (from step 1 to step DT_1) will be the sum of three sub-optimal sequences. In this situation, the optimality of Bautista *et al.* (1996)'s approach, based on the application of dynamic programming, fails to effect the result.

In other words, when the ideal hypotheses of the model do not hold, that is in any real manufacturing system, the multi-agent system seems to naturally adapt itself to manage the unforeseeable events of the system, whereas the optimal approach of Bautista *et al.*(1996) probably looses its advantage. Obviously this thesis has to be proved and in fact is currently studied at Politecnico di Milano. The results are forthcoming in a following paper.

6 REFERENCES

Bautista, J., Companys, R., Corominas, A. (1996) Heuristics and Exact Algorithms for Solving the Monden Problem. European Journal of Operational Research, **88**, 101-113.

Caridi, M., Carnicella, S., Colombo, S., Sianesi, A. (1998) A multi-agent application to the mixed-model lines balancing problem. Proceedings of the Seventeenth Workshop of the UK Planning and Scheduling Special Interest Group, Huddersfield, UK, September 9-10, 31-47.

Doran, J.E., Franklin, S., Jennings, N.R., Norman, T.J. (1997) On Cooperation in Multi-Agent Systems. The Knowledge Engineering Review, **12**.

Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McGuire, J., Pelavin, R., Shapiro, S., Beck, C. (1993) Specification of KQML as an Agent-Communication Language. Draft by the DARPA Knowledge Sharing Effort, http://www.cs.umbc.edu/kqml.

Inman, R., Bulfin, R.L. (1992) Quick and Dirty Sequencing for Mixed-Model Multi-Level Just In Time Production System. International Journal of Production Research, **30**, 2011.

Kubiak, W. (1993) Minimizing Variation of Production Rates in Just-In-Time Systems: A Survey. European Journal of Operational Research, 66, 159-271.

Lin, G.Y., Solberg, J.J. (1992) Integrated shop-floor control using autonomous agents. IEE Transactions, **14**, 57-71.

Marchal, H., Gauvin, D., Delle Donne, V. (1997) LALO - C++. http://www.crim.ca/sbc/english/lalo.

Miltenburg, J., Sinnamon, G. (1989) Scheduling mixed-model Multi-Level Just-In-Time Production Systems. Management Science, **27**, 1487-1509.

Monden, Y. (1983) Toyota Production System. Institute of Industrial Engineering Press, Necross, Georgia.

Pompei, C., Sartore, M. (1999) Co-ordination and Co-operation in Multi-Agent Systems: an application to mixed-model lines scheduling. Master Degree Thesis (in Italian), Politecnico di Milano, Italy.

Wooldridge, M., Jennings, N.R. (1995) Intelligent Agents: Theory and Practise. The Knowledge Engineering Review, **10**.

7 BIOGRAPHY

Maria Caridi graduated at Politecnico di Milano, where she works as a researcher in Industrial Production Management. Her main interest is production planning and control: as for the techniques, she has been studying the application of Multi-Agent System theory to the control of manufacturing systems; as for the instruments, she is concerned in how modern APS systems cover manufacturing system requirements and how they can be effectively integrated with ERP systems.