Maximizing the Throughput of Manufacturing Processes Modelled by Event Graphs

Angela Di Febbraro and Riccardo Minciardi

Department of Communications, Computer and Systems Science
The University of Genova
Via Opera Pia, 11 A, 16145 Genova,
ITALY

Abstract: Discrete Event Dynamic Systems can be useful to model and analyse the performances of manufacturing processes. The manufacturing system considered in this paper consists in the repetitive production of different classes of products by means of a set of machines. Such a system is modelled via Timed Event Graphs, a special class of Petri Nets, with the aim at exploiting the relevant results to analyse and optimize its performances. In particular, the optimization problem of maximizing the throughput of a system with constrained production mix while minimizing the work-in-progress is stated and discussed.

Keywords: Discrete event systems; manufacturing processes; event graphs; system analysis; optimization.

Angela di Febbraro was born in Genova, Italy, in 1963. She received the "laurea" degree in Electronic Engineering in 1987, and the Ph.D in computer science and electronic engineering in 1992 from the University of Genova, Italy.
Since 1987, she has done research at the Department of Communications, Computer and Systems Sciences of the University of Genova, where she is now a research assistant. Her research interests include modelling, optimization and control of discrete event systems and queueing networks with particular applications to manufacturing systems and communications networks.

Riccardo Minciardi was born in Catania, Italy, in 1951. He received the "laurea" degree in Electronic Engineering in 1975 from the University of Genova, Italy.
Since 1985, he has been an Associate Professor at the Department of Communications, Computer and Systems Sciences of the University of Genova. His research interests include optimization and control of discrete event systems, with particular applications to manufacturing systems and transportation systems.

1. Introduction

Petri Nets are usually regarded as a suitable, effective tool to design and analyse automated manufacturing systems. Major results achieved are related to structural properties of Petri Nets representing these systems, such as boundedness and liveness, which can be easily interpreted from a practical point of view (see for instance [1], [2] for examples of works related with this research stream). On the other hand, Petri Nets have seldom been used for performance analysis as such, apart from Generalized Stochastic Petri Nets (GSPN) [3], [4], [5], although their use presents some difficulties. In this framework, it is remarkable the case of a class of timed Petri Nets, named Timed Event Graphs (TEG's), for which a series of results have been obtained [6], giving the possibility of determining analytically the time-series of the output transitions firings, given the time-series of the input transitions firings.

Modelling manufacturing systems by TEG's may follow two different approaches. In the first one, operating rules and policies are embedded in the Petri net representation of the manufacturing system. In the second one, only the manufacturing process is explicitly modelled, and the control policies are implemented through a set of controlling places [7].

In this paper, the first of the above approaches is followed and TEG's are used to model a cyclic multiproduction system, in which a certain number of different classes of products are manufactured by means of a certain set of machines, and each product follows a specified routing through the machines.

The analytical tools related with TEG's allow to write and optimize some performance indices with respect to some decision variables. The decisional aspects to be optimized are the lot-sizes of the products and the initial state of the system. The considered performance index is the system throughput under a specified product mix. It will be shown that the optimization objective can be achieved by solving in sequence two mathematical programming problems, namely a linear fractional programming problem and an integer programming one.

2 Event Graph Model of the Manufacturing System

Consider the following model of a manufacturing system in which p different classes of items, $P_1,..,P_p$, are produced by a given set of m machines, $M_1,..,M_m$. Starting from basic components, each product of class P_i, $i=1,..,p$, is obtained through a given set of operations o_{i1}, ... o_{ini}, structured in an oriented precedence-relation graph. No restriction is made upon the structure of such graphs, apart from the absence of cycles. Operations can be simple, i.e., requiring one part in input and giving one part as an output, or assembly ones. As soon as a product of class P_i is completed, the components necessary for manufacturing another product of class P_i enter the system. It makes the functioning of the manufacturing process cyclic. For the sake of simplicity, it is supposed that there are no intersections among the sets of the basic components of the various classes of products. In any case, this could be achieved by considering those basic components related to more than one product to be formally different ones.

Each operation o_{ij}, $i=1, ..., p$, $j=1,.., n_i$, can be executed by any machine $M_k \in \mathcal{M}(i,j)$, where $\mathcal{M}(i,j)$ is the set of machines "compatible" with o_{ij}. Execution times are deterministic and known and no preemption is allowed. Each machine M_h, $h=1,.., m$, is generally made up of several servers, and can require a fixed set-up time between the execution of two subsequent operations. Moreover, for each pair of operations related by a precedence constraint, namely o_{ij} and o_{is}, there is a lower bound on the waiting time between the end of operation o_{ij} and the beginning of operation o_{is}. Finally, for some pairs of operations there can be an incompatibility constraint which prevents the simultaneous execution of the two operations (for instance, due to the necessity of the physical presence of the same basic component or intermediate product).

In this framework, the following decisional problems have to be solved: i) the sequencing of operations linked by incompatibility constraints; ii) the assignment of operations to machines; iii) the sequencing of operations on the single machines.

The presence of set-up times for machine utilization, as well as the fact that the elementary operations needed to manufacture products could be not so relevant to be considered individually, makes it convenient to group the single operations in macro- operations. In the following, it will be assumed that there is a fixed number of pre-selected "production routes" r_i for the realization of products of class P_i. Note that a production route including at least an assembly operation is usually not simply a list of machines, but an in-tree. The size of the lots of product P_i following the s-th route is N_i^s, $s=1,.., r_i$. If t_{ij}^k is the time required to perform the jth elementary operation relevant to the realization of product P_i on machine M_k which has been assigned to, $T_{ij}^{ks} = t_{ij}^k \times N_i^s$ will be the time required to perform the corresponding macro-operation O_{ij}^s on a lot of dimension N_i^s. It is supposed that all the specified routes are actually active, i.e. we impose $N_i^s > 0$, $\forall i, \forall s$. The dimensions of the lots are otherwise free and their determination will be treated in the following. Finally, it is supposed that the local sequencing rule at each machine is a fixed cycle of macro-operations, with the possibility that a single macro-operation is repeated in the same cycle.

The functioning of the manufacturing system complies with a specified production mix. This is guaranteed by the application of sequencing policies to the machines establishing fixed production ratios among the different products, which results in eliminating the necessity for imposing a predefined input sequence on the system. Of course, the system is supposed not to be decomposable into two or more disjoint subsystems.

The assumptions made so far individuate a model that is representable via Timed Event Graphs. From a structural point of view, a TEG is a timed Petri net in which every place has a unique input transition and a unique output transition, which makes it impossible to model any *or* operation among two or more events (transitions). Besides, an important characteristic of TEG's is that the number of tokens keeps constant in every loop during the life of the system.

To clarify the presentation, now an example of a TEG is given, modelling a manufacturing system with two classes of products, P_1 and P_2. There are six machines M_1,.., M_6 involved in this production. Products of class P_1 are obtained through a processing sequence of three operations, where the second one is an assembly operation, whereas products of class P_2 are obtained through a sequence of three simple operations. In this particular case, the production routes for products are simply lists of machines. Namely, it is assumed that the following two routes are active for P_1

$$R_1^1 = (M_1, M_2, M_3)$$

$$R_1^2 = (M_1, M_2, M_6)$$

whereas for P_2 only one route is active:

$$R_2 = (M_2, M_4, M_5)$$

This individuates the macro-operation sequences for the two classes of products, which are, respectively, $(O_{11}^1, O_{12}^1, O_{13}^1)$ and $(O_{11}^2, O_{12}^2, O_{13}^2)$ for P_1, and $(O_{21}^1, O_{22}^1, O_{23}^1)$ for P_2. The sequencings on the machines which have to perform more than a single macro-operation are:

$$S(M_1) = (O_{11}^1, O_{11}^2)$$

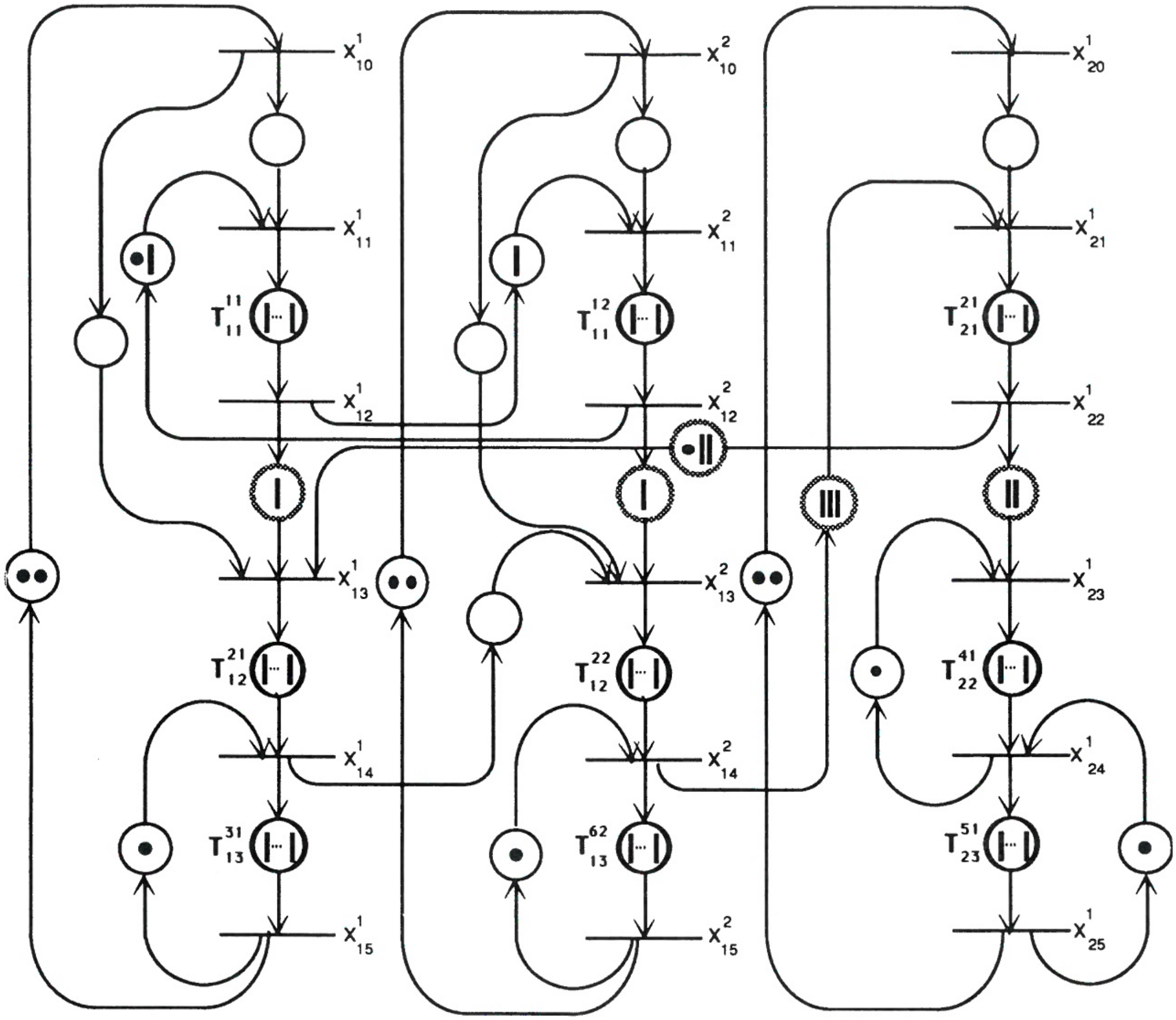
$$S(M_2) = (O_{12}^1, O_{11}^2, O_{21}^1)$$

The structure of the TEG representing this production is shown overleaf.

As a rule, in TEG's transitions are immediate and fire as soon as possible. Token holding times are associated with places, and represented by bars. The notation x_i/x_k is used to denote the place which has x_k as input transition and x_i as output transition. In the TEG above, places represented by thick black circles stand for macro-operation executions. For instance, place x_{12}^1/x_{11}^1 stands for macro-operation O_{11}^1, with duration T_{11}^{11}, which is executed by machine M_1 on lots of product P_1 following route R_1^1. Only the holding times of such places are affected by the choice of the lot-sizes. A second class of places, the grey circles, represents minimum waiting times and set-up times for machines. For instance, place x_{13}^1/x_{12}^1 stands for a waiting time constraint (independent of the lot-sizes), as well as places x_{13}^2/x_{12}^2 and x_{23}^1/x_{22}^1. Place x_{21}^1/x_{14}^2 stands for the set-up time needed by M_2 to switch from macro-operation O_{11}^2 on to macro-operation O_{21}^1, and place x_{13}^1/x_{22}^1 has an analogous meaning. The remaining places model conditions representing the availability of machines and components. For instance, place x_{13}^1/x_{10}^1 represents the availability of at least a lot of components to be assembled by means of macro-operation O_{12}^1, whereas place x_{13}^1/x_{24}^1 represents the availability of machine M_4 to perform a new macro-operation O_{22}^1.

3. Performance Analysis Tools

The decisional aspects involved in the definition of a system representable by means of the model described in the previous section can be classified into two classes. The first one includes the decisions regarding the number and types of the system machines, the production processes, the definition of the routes for all the classes of products, and of the sequencing rules on the machines. Such decisions affect the structure of the event graph modelling the system. Here, the focus will not be on such decisional aspects, and we will take into account the other class of decisions, which does not result in "structural" changes in the event graph. In particular, we consider the decisions regarding the dimensions of the production lots, and the initial conditions of the



production system. Such decisions are reflected by possible changes in the holding times in the places representing operation executions (but not in those representing set-up times of machines or minimum waiting times), and by the initial distribution of tokens in the event graph. The rest of the paper will focus on such a dependence of some performance indices of interest on the lot-sizes of the products, and on the initial marking in the graph.

The performance analysis methods that can be applied when dealing with event graphs are essentially of two types. The first one is based on an algebraic approach developed to analytically determine the event sequences in the network, given some event sequences as an input. Cohen *et al.* [6] have shown that, using a peculiar algebraic structure, a TEG can be described by state equations in (max, +) algebra (see also [8]) of the type

$$\underline{x} = A \otimes \underline{x} \oplus B \otimes \underline{u} \qquad (1)$$

$$\underline{y} = C \otimes \underline{x} \qquad (2)$$

where $\underline{u}$, $\underline{y}$, $\underline{x}$ are vectors referring to transitions and A, B, C are matrices derived from the TEG structure. The fundamental operations in the algebraic structure mentioned are minimum, maximum, and sum. Two shift operators γ and δ, acting respectively in event domain and in time domain are used. Equations (1)$\div$(2) are written in the 2D time-event domain. As in standard system theory, it is possible [6] to determine the transfer matrix of system (1)$\div$(2), which describes the input/output relationship, and is given by

$$H = CA^*B \qquad (3)$$

where $A^* = \oplus_{i=0}^{+\infty} A^i$, with $A^0 = E$, identity matrix in the considered algebra. Using the algebraic approach mentioned, it is possible to derive the firing sequence of any transition in the graph, given its initial state.

The second method is based on a simple result reported in [9]. Let the cycle time of a circuit γ in the graph

$$C(\gamma) = \mu(\gamma)/M(\gamma) \quad (4)$$

where $\mu(\gamma)$ is the sum of the holding times in the places in γ, and $M(\gamma)$ is the number of tokens circulating in γ. Of course, one must only consider elementary circuits, i.e. those cycles in which no node is repeated. Provided that the holding times are commensurable (and this is supposed to be the case), and that the net N is fully connected, it has been shown [9] that the system settles in a periodic regime after a finite amount of time, and then all the transitions fire with rate λ, being

$$\lambda = \frac{1}{\max_{\gamma \in N} C(\gamma)} \quad (5)$$

As regards the use of the above methods in the analysis of the performance measures of interest, it is apparent that the latter approach allows to evaluate functionals that can be computed only on the basis of the average behaviour of the system (e.g. the throughputs relevant to the different products). On the other hand, functionals whose evaluation requires the detailed determination of the periodic transition firing sequences (e.g. those indices related to product completion times) require the application of the first approach. In the next section, the analysis and the optimization of performance indices of the first class will be under study.

4. Optimization Problems

The most sensible optimization objective that can be pursued with respect to the lot-sizes of the different products and the choice of the initial marking is the maximization of the system throughput with a fixed mix, while minimizing the work-in-progress. To state formally such an objective, it is necessary to introduce the concept of cycle time of a machine. The cycle time $C(M_i)$ of machine M_i is defined as the sum of the processing times of the macro-operations in its cycle plus the sum of the relevant set-up times. For instance, in the example shown in section 2 the cycle time of machine M_2 is $C(M_2) = T^{21}_{12} + T^{22}_{12} + T^{21}_{21} + 5$. Note that in general the cycle time $C(M_i)$ can be expressed as an affine function of the vector of the lot-sizes, namely

$$C(M_i) = \underline{h}_i^T \underline{N} + q_i \quad (6)$$

where $\underline{N} = \text{col}\,[N^1_1, N^2_1, \ldots, N^1_2, N^2_2, \ldots]$ is the vector of the lot-sizes, $\underline{h}_i^T$ is a vector of suitable coefficients and q_i is a constant. $N^1_1, N^2_1, \ldots$, are the dimensions of the various production lots relevant to product type P_1. In general, it is not necessarily true that the lot-sizes relevant to the same product type are equal. It is apparent that the mix specification, or more properly, the sequencing rules on the machines performing operations relevant to different product types, impose linear constraints on the components of vector $\underline{N}$. For instance, still referring to the previously detailed example, it must be $(N^1_1 + N^2_1)/N_2 = X_1/X_2$, being X_1/X_2 the mix, i.e. the ratio between the throughput of the two products.

Following the same reasoning line as in [10], once fixed the lot-sizes, the system works at its maximum productivity whenever there is at least one machine working at its cycle time, i.e. acting as a bottleneck machine. That means that there is a machine M_k such that $C(M_k) \geq C(\gamma)$ for any circuit γ in the network. Then, the system can be optimized by choosing the initial marking $\underline{M}_0 = \text{col}[M_0(p_i), \forall \text{ place } p_i \text{ in the graph}]$ in such a way that the work-in-progress is minimized and the bottleneck machine is saturated, i.e. it is always either working or in a set-up time. This could be accomplished the same way as in [10], if two major complications did not arise in the model described in this paper. The first complication results from the constraints imposed on the initial marking of the modelling graph by the presence of assembly operations. For instance, in the example proposed in section 2, it must be $M_0(x^1_{11}/x^1_{10}) = M_0(x^1_{13}/x^1_{10})$ to avoid an initial unbalance of the number of components that should be assembled during operation O^1_{12}. Such an unbalance would remain unchanged during the net evolution, since the system is closed, so it would yield just an inconvenient increase in the work-in-progress. In general, such balance conditions result in a linear vectorial constraint of the type $A\underline{M}_0 = \underline{0}$. The second, more important difference lies in the fact that the lot-sizes of the products are themselves decision variables, so that it is not immediate to determine the bottleneck machine.

Then, the overall optimization procedure may be decomposed into two steps: i) determine the lot-sizes so that the allowable system productivity, i.e. the maximum system throughput allowed, is maximized; ii) having fixed the lot-sizes as above, determine the initial marking so that the allowable system productivity is actually achieved and the work-in-progress in the system (number of workpieces) is minimized.

The maximization of the throughput requires that at least a machine acts as a bottleneck. In such conditions, all the machines have the same cycle time, i.e. that of the bottleneck machine. Let $\overline{C}$ denote such a cycle time, and suppose that no macro-operation appears more than once in the cycle of any machine. Now, the overall throughput relevant to the k-th product class can be computed as

$$X_k = \sum_{j=1}^{r_k} (N_k^j / \mathcal{C}) \tag{7}$$

where N_k^j is the size of the lots of product P_k following the j-th route.

Due to the mix constraints, the maximization of the throughput for a product type automatically yields the maximization of the throughputs of all the other products. Thus, the problem of determining the lot-sizes so as to achieve the maximum throughput can be stated as follows.

Problem 1. Maximize the throughput of product X_k, for an arbitrary choice of k in {1, .., p}

$$\max_{\underline{N}, \mathcal{C}} X_k = \max_{\underline{N}, \mathcal{C}} \{\sum_{j=1}^{r_k} (N_k^j / \mathcal{C})\} \tag{8}$$

subject to

$$\frac{X_k}{X_s} = \frac{\sum_{u=1}^{r_k} N_k^u}{\sum_{v=1}^{r_s} N_s^v} \qquad s = 1, .., p, \quad s \neq k \tag{9a}$$

$$N_{kmin}^j \leq N_k^j \leq N_{kmax}^j \qquad j=1, .., r_k, \ k=1,.., p \tag{9b}$$

$$\mathcal{C} = \max_{i=1, .., m} C(M_i) = \max_{i=1, .., m} \{\underline{h}_i^T \underline{N} + q_i\} \tag{9c}$$

where

- constraints (9a) guarantee the fulfilment of the production mix;
- N_{kmin}^j and N_{kmax}^j are respectively the a-priori lower and upper bounds of the lot-sizes of product P_k following the j-th route;
- in (9c) the equality constraint can be substituted by the inequalities

  $$\mathcal{C} \geq \underline{h}_i^T \underline{N} + q_i \qquad i = 1, .., m \tag{9d}$$

  since the optimization objective ensures that at least one of the (9d) is fulfilled with an inequality sign. Δ

The optimization problem above is a linear fractional programming one, which can be solved by means of well-known mathematical programming techniques [11].

The second optimization step consists in solving a problem which is fairly similar to that considered in [10], but for the constraints on the initial marking, which are now different for the reasons already explained. Then, the optimization problem relevant to such a step can be stated as follows.

Problem 2. Find, with respect to the initial marking of the np places $p_1,.., p_{np}$ in the network,

$$\min \left\{ \sum_{s=1}^{np} \alpha(s) M_0 (p_s) \right\} \tag{10}$$

subject to

$$M_0(\gamma) \geq M_m(\gamma) \ \forall \text{circuit } \gamma \text{ in the network} \tag{11a}$$

$$A\underline{M}_0 = \underline{0} \tag{11b}$$

$$\mu(\gamma)/\mathcal{C} \leq M_m(\gamma) < \mu(\gamma)/\mathcal{C} + 1 \tag{11c}$$

$$\forall \ \text{circuit } \gamma \text{ in the network}$$

where

$\alpha(s)$ is a weighting coefficient whose value indicates if tokens in place p_s represent basic components or intermediate products;

$\underline{M}_0 = \text{col}[M_0(p_i), \ \forall \ \text{place } p_i, i=1, .., np]$;

constraint (11b) represents the balance conditions imposed by the assembly operations present in the graph;

$M_m(\gamma)$ is the minimum number of tokens such that $\overline{C} \geq C(\gamma)$;

A is a matrix whose elements are 0, 1, or – 1, according to the possible constraints to impose on the initial markings of the places involved in assembly operations;

$\mu(\gamma)$ is the sum of the holding times in the places in γ. Δ

The problem is an integer linear optimization one, which can be solved by means of standard techniques [11]. A major difficulty in doing so is that the statement of such a problem requires the determination of all the elementary circuits in the network. Efficient algorithms can be found in the literature for this purpose [12]. In [13] a heuristic procedure to find a "good" solution to Problem 2 is presented.

A simple numerical example

Consider again the example of the manufacturing process reported in the paper on page 3 and the following. Suppose that the production mix imposes on the throughputs of the two classes of products the constraint $X_1 = 2X_2$. Let the execution times of the elementary operations on the machines which they have been assigned to be the following:

$$t^1_{11}=2\,;\ t^2_{12}=1\,;\ t^3_{13}=2;\ t^6_{13}=3\,;\ t^2_{21}=1;\ t^4_{22}=3;\ t^5_{23}=2$$

and the a-priori lower and upper bounds of the lot-sizes of products be

$$N^1_{1min}=1,\ N^1_{1max}=150\,;\ N^2_{1min}=1,\ N^2_{1max}=200;$$

$$N^1_{2min}=1,\ N^1_{2max}=100$$

In that case, the cycle time of the bottleneck machine results to be

$$\overline{C} = \max_{i=1,..,m} C(M_i) =$$

$$= \max\left\{2N^1_1+2N^2_1+2,\ N^1_1+N^2_1+N^1_2+5,\ 2N^1_2,\ 3N^2_1\right\}$$

Then, to maximize the throughput X_2 (which is equivalent to maximizing X_1) one has to solve the following problem.

Find

$$\max_{\underline{N},\overline{C}} X_2 = \max_{\underline{N},\overline{C}} N^1_2/\overline{C}$$

subject to

$$\frac{X_2}{X_1} = \frac{N^1_2}{N^1_1+N^2_1} = \frac{1}{2}$$

$$1 \leq N^1_1 \leq 150$$

$$1 \leq N^2_1 \leq 200$$

$$1 \leq N^1_2 \leq 100$$

$$\overline{C} \geq 2N^1_1+2N^2_1+2$$

$$\overline{C} \geq N^1_1+N^2_1+N^1_2+5$$

$$\overline{C} \geq 2N^1_2$$

$$\overline{C} \geq 3N^2_1$$

An optimal solution to this problem is $N^1_1 = N^2_1 = N^1_2 = 100$, which yields $\overline{C} = 402$. Then, the maximum throughputs achievable in the system are $X_1 \cong 0.5$ and $X_2 \cong 0.25$.

5. Conclusions

In this paper it has been shown that a class of performance analysis problems related to manufacturing processes can be studied through Timed Event Graphs. Referring to the model of a repetitive production system, an optimization procedure has been formally stated, whose objective was the maximization of the system throughput with a fixed production mix, while minimizing the work-in-progress. It has been shown that the maximization of the maximum allowable throughput of the system with respect to the lot-sizes of the components consists in solving a linear fractional programming problem.

Thus, the proposed approach generalizes that presented in [10], because it allows to consider assembly operations and multiple alternative routes for the single classes of products. It is worth observing that such an approach could also apply, though in an indirect way, to optimizing the system performance with respect to the production routes. In fact, whenever the optimum value of a lot-size for a route equals its lower bound, this suggests to "rebuild" the Petri net of the system eliminating that route, and then find the optimal lot-sizes for the new system.

REFERENCES

1. ZHOU, M.C. and DI CESARE, F., **Parallel and Sequential Mutual Exclusion for Petri Nets Modelling of Manufacturing Systems with Shared Resources**, IEEE TRANS. ROBOT. AUTOMAT. 7, 4, 1991.
2. KOH, I. and DI CESARE, F., **Modular Transformation Methods for Generalized Petri Nets and Their Application to Automated Manufacturing Systems**, IEEE TRANS. SYST. MAN CYBERN. 21, 6, 1991.
3. AL-JAAR, R.Y. and DESROCHERS, A.A., **Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets**, IEEE TRANS. ROBOT. AUTOMAT. 6, 1990.
4. WATSON, J.F. and DESROCHERS, A.A., **Applying Generalized Stochastic Petri Nets to Manufacturing Systems Containing Nonexponential Transitions**, IEEE TRANS. SYST. MAN CYBERN.21, 5, 1991.
5. VISWANADHAM, N. and NARAHARI, Y., **Performance Modelling of Automated Manufacturing Systems**, Prentice-Hall, 1992.
6. COHEN, G., MOLLER, P., QUADRAT, J.P. and VIOT, M., **Algebraic Tools for the Performance Evaluation of Discrete Event Systems**, Proc. IEEE 77, 1, 1989.
7. HOLLOWAY, L.and KROGH, B.H., **Synthesis of Feedback Control Logic for a Class of Controlled Petri Nets**, IEEE TRANS. AUTOM. CONTROL 35,5, 1990.
8. CUNINGHAME-GREEN, R., **Minimax Algebra**, SPRINGER-VERLAG, Lecture Notes in Economics and Mathematical Systems, 1979.
9. CHRETIENNE, P., **Les Reseaux de Petri Temporisés**, Thése d'Etat, Univ. Paris VI, Paris, 1983.
10. HILLION, H.P. and PROTH, J.M., **Performance Evaluation of Job-shop Systems Using Time Event-Graphs**, IEEE TRANS. AUTOM. CONTROL 34, 1989.
11. NEMHAUSER, G.L., RINNOOY KAN, A.H.G. and TODD, M.J., **Optimization**, NORTH HOLLAND, 1989.
12. MARTINEZ, J. and SILVA, M., **A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Net**, SPRINGER-VERLAG, Lecture Notes in Computer Sciences, 52, 1980.
13. BAZARAA, M. and SHETTY, C., **Nonlinear Programming**, JOHN WILEY,1980.