PERGAMON

Control Engineering Practice 8 (2000) 63–69

www.elsevier.com/locate/conengprac

Assembly line sequencing based on Petri-net T-invariants

Georg Frey*

Institute of Process Automation, University of Kaiserslautern, PO Box 3049, 67663 Kaiserslautern, Germany

Received 5 May 1999; accepted 23 July 1999

Abstract

Mass production of goods with customer specific features is mostly carried out on production lines. Job-dependent processing times in the lines' individual work zones requires line sequencing in order to use the line to capacity. This paper presents a two-step method allowing fast sequencing even for large numbers of jobs and product types. The knowledge about the line characteristics and about the possible jobs is modelled using a Petri-net. The resulting model is independent of the set of jobs to be scheduled. Based on the Petri-net's T-invariants, a valid production sequence is efficiently determined for a particular job set. *© 2000 Published by Elsevier Science Ltd. All rights reserved.*

Keywords: Petri-nets; Discrete-event systems; Flexible manufacturing systems; Invariants; Scheduling algorithms

1. Introduction

Products with customer specific features, e.g. automobiles, are mostly manufactured on production lines as well as in plants combining production lines and team workstations (Buzacott, 1990). In this paper a paced production line with partially overlapping work zones and a fixed launching rate (cycle time) is considered. In accordance with the type of job, it takes the worker in the individual work zone a certain amount of time to perform the necessary tasks. To use the line to capacity, a cycle time, i.e. the time chosen between the launching of two successive jobs is equal to the average processing time. Assuming that the processing times of several succeeding jobs at a particular work zone are always greater than the cycle time, it occurs that a job cannot be processed completely within the zone boundaries. In that case, the job cannot be processed any further on the line, or at least it has to be re-worked later on. Therefore, line sequencing, i.e. determination of a valid job sequence, is necessary. An example of such a situation would be a car assembly line where at most every third car can be equipped with air-conditioning.

The line sequencing problem can be formulated as Mixed Integer Linear Program (MILP). The complexity of this problem grows exponentially with the number of jobs to be scheduled. Therefore, MILP algorithms can only be used for small problems. Hence, algorithms based on heuristics were introduced, see (Bolat, 1994; Bard, Shtub & Joshi, 1994) for an overview and a comparison of different heuristic algorithms. Irrespective of the algorithm used, it is possible to determine critical work zones and to consider only those that are critical within the sequencing algorithm (Rachamadugu & Yano, 1994).

This paper introduces a two-step method for assembly line sequencing. The method avoids complexity, without using heuristics. In contrast to the heuristic method, it guarantees that if a valid sequence exists, it will be found by the algorithm.

In the first step the plant is modelled as a discrete-event system (DES). The line characteristics, i.e. the zone dimensions, as well as knowledge about the possible jobs are incorporated into a Petri-net model. The states of the model are obtained from the system's delay times. Before any job is launched to the line, the system is in its initial state. Every worker begins his task exactly when the first job enters his work zone. In cases where the processing time of the first job in a particular work zone is greater than the cycle time, the worker cannot begin to process the second job when it enters his work zone but has to finish the first job beforehand. This leads to a time delay between the instance when the second job enters the work zone and its processing begins. The vector combining the time delays of all work zones as its components defines the system's state. Consequently, a state

---

* Corresponding author. Tel.: + 49-631-205-3652; fax: + 49-631-205-4462.

E-mail address: frey@technik.uni-kl.de (G. Frey)

PII: S0967-0661(99)00139-2

transition occurs when a new job is launched onto the line. A branch-and-bound algorithm is used to construct a Petri-net model that includes all the states reachable during valid production sequences. Model reduction is carried out by aggregating equivalent states and events; and by grouping the events. The T-invariants of the net are calculated and extended to group invariants (G-invariants) which form the basis of an algebraic system description. All these calculations have to be carried out only once, as long as the line and product characteristics are not altered.

In the second step, a valid sequence for a given job set is determined by solving a system of linear equations and inequalities. The complexity of this system is independent of the number of jobs to be scheduled. So in any case, there is a sequence length, where the new approach is of less complexity than the MILP formulation. The benefit of the new method is the fast computation of valid job sequences for the daily production.

The method is illustrated using an assembly line in the automotive industry.

2. Assembly line as discrete-event system

In this section a formal definition of the problem and a brief overview about the Petri-net theory used in the subsequent text precede the presentation of the discrete-event problem description. An illustrative example closes the section.

2.1. Problem definition

The approach deals with paced assembly lines having partially overlapping work zones and fixed launch rates. Due to the method of transportation through the zones (commonly a steadily moving conveyer), the zone boundaries can be defined in terms of a job's arrival time, relative to its launching time. Integer values for the times are sufficient since these can be chosen as a multiple of a basic time step. See Fig. 1 for an example, it shows the schematic picture of an assembly line with two overlapping work zones.

Definition 1. A paced assembly line consisting of n production zones that can produce different types of products m, is a tuple $L = (c, \mathbf{Z}, \mathbf{P})$ where

(1) c is the cycle time,
(2) $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]^{\mathrm{T}}$ is the matrix of zone boundaries, with $\mathbf{z}_i = [z_{i,l}, z_{i,u}]$ the vectors of the lower and upper boundary of zone i, with $z_{i,l} \leqslant z_{i+1,l}$, and
(3) $\mathbf{P} = [p_{ij}]$ is the $(m \times n)$-matrix of processing times where p_{ij} is the processing time of product type i in production zone j.

The sequencing problem is stated as follows.

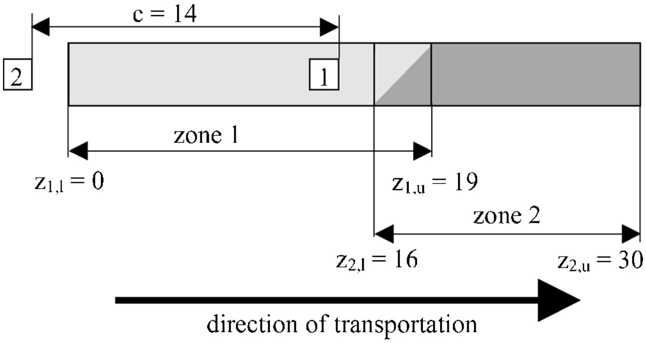

Fig. 1. An example of assembly line.

Definition 2. For a paced assembly line $L = (c, \mathbf{Z}, \mathbf{P})$ and a set of k jobs, described by a job vector $\mathbf{n} = [n_1, \ldots, n_m]$, with n_i as the number of jobs of product type i, find a production sequence $S = \langle s_1, \ldots, s_k \rangle$ so that

(1) all jobs are processed,
(2) the zone boundaries are not overrun,
(3) no job is processed in more than one zone at a time and
(4) the line is in its initial state after all jobs are processed.

The third condition in Definition 2 is often used in practice for safety reasons. Condition 4 guarantees that two separately computed sequences can be concatenated.

2.2. Petri-nets and T-invariants

Here, a very brief overview of Petri-nets is given. For a complete review see for example (David & Alla, 1992; Desroches & Al-Jaar, 1994). To allow a reaction of the model to external events, a special form of synchronised Petri-net is used.

Definition 3. A synchronised Petri-net is a six-tuple $N = (P, T, A, \mathbf{m}_0, E, M_{\mathrm{E}})$, with:

- $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is a set of $|P|$ places;
- $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is a set of $|T|$ transitions;
- $P \cap T = \varnothing$, i.e. the sets P and T are disjoint;
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs;
- the vector $\mathbf{m}_0 : P \to \{0, 1\}$ is the initial marking;
- $E = \{e_1, e_2, \ldots, e_k\}$ is a set of external events and
- M_E: $E^* \varnothing \to T$ is the synchronisation function.

The pre set $\bullet t_k$ of a transition t_k consists of all places $p_i \in P$ with $(p_i, t_k) \in_A$. The post set $t_k \bullet$ of a transition t_k consists of all places $p_i \in P$ with $(t_k, p_i) \in A$. Pre- and post-set of places are defined accordingly. Fig. 2 shows an example of a synchronised PN. It consists of three places (circles) and five transitions (bars). The state of the net is given by the number of tokens marking the places. In the