Computers in Industry 47 (2002) 123–138

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

Integrating Petri Nets and hybrid heuristic search for the scheduling of FMS

A. Reyes[a], H. Yu[b,*], G. Kelleher[c], S. Lloyd[d]

[a]*iSOCO Laboratory, Alcalde Barnils 64/68 A, 08190 Sant Cugat del Valles, Barcelona, Spain*
[b]*School of Engineering and Computer Science, Exeter University, North Park Road, Exeter EX4 4QF, UK*
[c]*CMS, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK*
[d]*School of Engineering, Sussex University, Brighton BN1 9QT, UK*

Received 19 July 1999; accepted 19 May 2001

Abstract

This paper studies modelling and scheduling of Flexible Manufacturing Systems (FMS) using Petri Nets (PNs) and Artificial Intelligence (AI) based on heuristic search methods. A subclass of PNs, *Buffer nets* or *B-nets* is obtained by the systematic synthesis of PN models from FMS formulations. Scheduling is performed as heuristic search in the reachability tree, which is guided by a new heuristic function that exploits PN information. This heuristic is derived from a new concept, the Resource Cost Reachability (RCR) matrix which builds on the properties of *B-nets*. To mitigate the complexity problem, a hybrid search algorithm is proposed. The algorithm combines dispatching rules based on analysis information provided by the PN simulation with a modified *stage-search* algorithm. Experimental results are provided and indicate the effectiveness of the approach and the potential of PN-based heuristic search for FMS scheduling. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Heuristic search; Flexible Manufacturing Systems; Modelling; Scheduling; Petri Nets

1. Introduction

An FMS usually consists of several numerically controlled manufacturing machines and automated material handling systems that transport work-pieces between machines and tool systems. In a facility with routing flexibility, each product can be manufactured via one of several available routes. A high-level control system must decide which resources to assign to which product at what time, to optimise some criteria, (for example, makespan). The planning/scheduling of an FMS is the process of determining the allocation of parts to machines and the sequence of operations so that the constraints of the system are met and performance criteria are optimised.

Unfortunately flexibility in manufacturing systems comes at a price. In the case of FMS the price is one of operation complexity, which means that, without an effective means of scheduling and controlling production FMS, economic returns will be poor. Consequently, the problem has been of considerable interest to both academic and industrial researchers over the last three decades [2,14,18,26].

The recent integration of PN as a representation tool with AI problem solving methods as a reasoning paradigm appears to be a promising answer to the

* Corresponding author.
E-mail address: h.yu@exeter.ac.uk (H. Yu).

0166-3615/02/$ – see front matter
PII: S0166-3615(01)00124-5

need to develop models that factor in the full complexity of the FMS, yet are efficient enough to obtain good solutions.

A Petri Net (PN) [16] is a mathematical formalism and graphical tool that may be used for the modelling, formal analysis and design of discrete event systems. It has been widely used in industrial applications [32]. The performance of production systems, involving simple production, job shops, robotic assembly cells and FMS, has been extensively studied by the PN community [5,19,27,31]. As a graphical tool, a PN works like a flow chart and provides a visualisation of a dynamic system. As a mathematical tool, a PN model can be described by a set of linear algebraic equations, which allows formal checking for properties related to the behaviour of the underlying system.

The combination of PN simulation capabilities and AI-based systematic search within the PN reachability graph is proposed in [7]. Branch and Bound (B&B) search is employed in [1,4,12]. The A^* algorithm has also been applied in [9,12,30]. An A^* algorithm which limits the backtracking capability of the algorithm by introducing irrevocable decisions is implemented in [8,10,28]. Reyes et al. [20], have also studied a hybrid search algorithm based on relaxation of the evaluation scope of an A^*-based algorithm. A hybrid algorithm combining *Best-First* and B&B search methodologies has been considered in [29]. *Beam search* as on-line decision support is implemented in [25].

A typical problem observed is the difficulty in balancing the control of the search effort with the potential of PN to guide a heuristic search process. This has lead to the application of non-admissible and non-well-informed heuristic functions that reduces the applicability of using information from the PN to guide the search process. In this paper, we extend the work begun in [17]. The aim is to reduce the scope of evaluation of the A^* algorithm so that we may apply a more exhaustive local search and to enhance the usefulness of a new admissible heuristic function based on the PN model. The research, thus, aims to provide effective decision modules integrated with knowledge-based architectures that employ a PN as a representation paradigm [15] or hybrid methodologies based in random optimisation such as [21].

2. Scheduling of FMS based on PN structures: representation and reasoning

2.1. Petri Nets

A timed PN [16] can be defined as a bipartite directed graph formed by three types of entities: places, transitions and directed arcs (connecting places to transitions and transitions to places). The concept of time is assigned to transitions in this paper.

A PN can be defined formally as the tuple: $\mathrm{PN} = (P, T, I, O, \tau, M_0)$ where P is the finite set of places; T the finite set of transitions $P \cup T \neq 0$, and $P \cap T = 0$; I: $(P \times T) \rightarrow N^+$ is an input function that defines directed arcs from places to transitions; O: $(P \times T) \rightarrow N^+$ is an output function that defines directed arcs from transitions to places; τ: $T \rightarrow R^+ \cup \{0\}$ the firing time function; M_0: $P \rightarrow N^+ \cup \{0\}$ is the initial marking.

Places are represented as circles, and transitions as bars or boxes. Places and transitions are interconnected by means of directed arcs whose *weight* is given by I and O. A place p is said to be an *input* (*output*) *place* of a transition t if$I(p,t) > 0$ $(O(p,t) > 0)$.

The PN marking expresses the state of the net and is given by a concrete distribution of tokens (solid dots) in the places of the PN. Due to the concept of time a token can have two states: available or unavailable.

Starting from the initial state M_0, the following two rules are used to govern the flow of tokens in the net, which simulates the dynamic behaviour of the modelled system.

1. *Enabling rule*: A transition t is *enabled* if every input place p contains at least as many available tokens as the weight of the directed arc connecting p to t.
2. *Firing rule*: A firing of an *enabled* transition t removes from each input place p as many tokens as the weight of the directed arc connecting p to t. It also deposits in each output place p the number of tokens equal to the weight of the directed arc that connects t with p. The new tokens deposited in p are unavailable and will become available after $\tau(t)$.