Computers in Industry 68 (2015) 27–39

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Collaborative routing of products using a self-organizing mechatronic agent framework—A simulation study

Luis Ribeiro [a,*], Andre Rocha [b], Angelo Veiga [b], José Barata [b]

[a] *Linköping University, Department of Management and Engineering, Division of Manufacturing Engineering, SE-58183 Linköping, Sweden*
[b] *Uninova – CTS, Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2825-149 Caparica, Portugal*

ARTICLE INFO

Article history:
Received 11 September 2013
Received in revised form 29 October 2014
Accepted 4 December 2014
Available online 23 January 2015

Keywords:
Multiagent systems
Material handling
Self-organization
Scheduling
Product oriented systems
Mechatronics

ABSTRACT

Scheduling is a fundamental activity in modern shop floors. It is also known to be a highly complex problem which has motivated several sub-formulations and the subsequent models. Traditional approaches, typically enumerative or heuristic, struggle to contain the computation complexity and often present solutions for restricted cases that feature unrealistic assumptions in respect to the system size, flow of products and the system logistics/behaviour. The multiagent-based architecture presented in this paper is aligned with a set of emerging architectures that seek to explore more heterarchical decision and control models to circumvent the limitations of the traditional approaches. The main distinguishing factor of the proposed architecture is that it directly addresses (re)routing/local scheduling of products in plug and produce systems. It does not make any assumptions on the alignment of the orders and, instead, it dynamically handles the potential rescheduling of the orders already on the system based on the available resources, and their state, in a time efficient way. The architecture was tested under a simulation environment, that is geometrically accurate and that supports plug and produce in runtime, to characterize its performance under dynamic conditions.

1. Introduction

The scheduling of products, parts, assemblies or sub-assemblies to different shop floor resources is a well-known and complex problem. It has been typically formulated as an optimization problem. It is also known to be NP hard consisting in the selection of the best possible schedule out of $n!^m$ where n is the number of tasks (or jobs) and m the number of available machines. The complexity of determining effective schedules is subsequently aggravated when [1]: operators and tools are included in the process, optimization occurs both for planning and scheduling, unpredictable conditions impact the system (failures, breakdowns, system changes, production surges).

This has promoted the sub-formulation of the scheduling problem to meet distinct objectives and several performance indicators have been chosen as the optimization target. When the main objective is to improve the system's balance, typical problem formulations include: maximization of line utilization, minimization of number of stations given the cycle time, minimization of the cycle time given the number of stations, a compromise between the number of stations and cycle time [2,3]. When the optimization objective is more focused on performance then makespan [4–13], minimization of tardiness [11,14,15], throughput [16,17], energy efficiency [18], work in progress [12], activity based costing [19], etc., are commonly used to characterize the performance of the scheduling algorithms.

Scheduling has therefore been approached from different perspectives, with different objectives. The conventional approaches are based in enumerative or heuristic algorithms and are normally able to produce near optimal solutions. Known techniques and algorithms include: genetic algorithms [6,14], ant colony optimization [8,10], particle swarm optimization [9,13,15], fuzzy control [12,13] and neural networks [7].

These techniques consider the system and the orders to be scheduled as a whole which is the main strategy for obtaining a near optimal solution. However these strategies are extremely sensible to system perturbations. Any significant system change requires a complete rescheduling. The time efficiency of these algorithms is also directly related with the size of the system. In this context, they are not suitable for highly dynamic systems of a

* Corresponding author. Tel.: +46 0700850630.
E-mail addresses: luis.ribeiro@liu.se (L. Ribeiro), ad.rocha@campus.fct.unl.pt (A. Rocha), a.veiga@campus.fct.unl.pt (A. Veiga), jab@uninova.pt (J. Barata).

http://dx.doi.org/10.1016/j.compind.2014.12.003
0166-3615/

larger size. Computational tractability has been dealt with by introducing constraints in the problem formulation. These limitations normally include: reduced number of stations and jobs, reduced differentiation of stations, inexistence of machine breakdowns or system changes, unlimited buffering capacity, process flows without repeating stations, no accounting for transport costs, etc. The purpose of introducing these constraints is to significantly reduce the search space of the algorithms. However these also limit the application scope in real scenarios and these techniques are normally restricted to small job-shop-based systems.

Scheduling is therefore a pressing issue in bigger and more complex systems that have increasing reconfigurability requirements. The key aspects of a reconfigurable system include: module mobility, diagnosability, integrability, convertibility, scalability, "automatibility", modularity and customization [20,21]. All these characteristics promote and support frequent and dynamic system changes that are hardly tackled by the traditional approaches and that address the need for mass customization and sustainability. This problematic has originated, in the past fifteen years, a set of multiagent-based techniques that tend to replace exhaustive or heuristic search with negotiation [1] or some form of collaborative interaction. It is at this point however important to distinguish between:

- multiagent-based approaches that rely in the agent concept as a mere distributed computation construct and seek to explore the availability of distributed computational resources;
- and multiagent-based approaches that encourage an identity relation between purpose-specific agents and their physical counterpart and explore collective self-organizing phenomena.

The first case is closer to the conventional approaches, already described, and shares the limitations considered before.

The article focuses therefore in the latter case which has led to the emergence of several reference architectures. These architectures, collectively, provide the constructs to integrate: planning, scheduling, control, material handling, monitoring and diagnosis.

The multiagent systems community has been particularly active, in the past decade, in the development of agent-base architectures for manufacturing systems. This research has generated a wide range of system specific architectures [22] and set of more generic reference architectures among which one may mention: PROSA [23], HCBA [24], ADACOR [25], COBASA [26], Rockwell Automation Agents [27] and, more recently, ORCA-FMS [4] and the IDEAS Architecture [28]. The architectural design at this level is particularly relevant since it bridges the gap between more abstract concepts and paradigms, such as Holonic Manufacturing Systems [29], Bionic Manufacturing Systems [30], Evolvable Production Systems [31], and system engineering.

The present work positions itself in this context and places particular emphasis in handling manufacturing systems under dynamic conditions. In particular, it addresses systems where:

- stations and transport elements can be plugged and unplugged at any time (dynamic topology);
- and orders are of random nature.

The proposed architecture envisions the system as a directed graph where links and nodes are directly connected to specific agents. These continuously compute transport costs between locations and maintain local routing tables featuring the shortest path, and direction, to reach all relevant locations in the system. The architecture supports the integration of user defined metrics so as to cater for the requirements of a wider range of systems. The architecture was deliberately designed to be independent of other external planning or scheduling tools and targets the orders that are already on the system for which time-effective and runtime decisions need to be considered. In particular, the paper seeks to demonstrate that the architecture is able to improve the overall makespan, in systems supporting plug and produce, while balancing the load on the different components. One fundamental aspect is that all the components are fully decoupled and hence there are no centralized points of failure other than the ones imposed by the mechanical limitations of the system under control. In this context, agents make their own decisions and scheduling is necessarily locally derived and seeks to explore collective phenomena to attain a global consistent state rather than a fully optimized solution.

The subsequent details are organized as follows: Section 2 discusses the related literature in agent based manufacturing scheduling; Section 3 presents the proposed architecture, supporting algorithms and relevant implementation details; Section 4 summarizes the differences between this work and the state of the art; Section 5 presents and assesses the main results; Section 6 features the main conclusion and points future research directions.

2. Related literature

2.1. Brief survey

When dealing with complex and dynamic shop-floors there is the need for frequent rescheduling of the orders already in the system as well as the orders that are about to enter it. Hence, there is an interweaving between material handling and scheduling activities that is often ignored in the traditional models. There have been several contributions towards the definition and tackling of these problems. This section focuses on more recent work where the documented approaches are closely related with the one proposed. A complete survey on other approaches and techniques can be found in [1].

Material handling is a problem that is not exclusive of the manufacturing domain and is a complex problem on its own. In [32] the problematic of developing a generic automated material handling systems (AMHS) is discussed. The authors conclude that most of the AMHSs are sector specific and that the takes on a more generic approach often fail due to the specificities of particular sectors. Also, and not surprisingly, the authors reject centralized approaches due to their rigidity when handling dynamic conditions and voice their concerns regarding pure heterarchical approaches due to the high potential for deviations from optimized solutions. Finally, they propose a hybrid solution in [16] which features a three layer hierarchical system composed of a planner, a resource scheduler and a local traffic control. The local traffic control does specialized optimization in restricted regions of the system using routing rules while the two higher level blocks handle sequencing and coordination issues. The authors also focus their AMHS model on two particular sectors where they identified several structural commonalities. They were therefore also able to verify a high degree of code commonality between the solutions for the different sectors.

A similar and previous solution can be found in [33] where a Holonic-based architecture introduces a global scheduler. The main decision process is mediated by the order holons that are able to negotiate with the material handling holons as well as with the global scheduler to devise the execution plan. The order holons globally announce the transportation task and wait some time for the transportation bids. The local bids from the material handling holons tend to arrive faster and a local schedule is built upon them. The scheduling is later revised if the global scheduler is able to generate a timely solution. A follow up of this proposal is presented in [34] where local schedules are generated at material handling