Iterative Optimization-Based Simulation: A Decision Support Tool for Job Release

Nuno O. Fernandes[1,3](✉), Mohammad Dehghanimohammadabadi[2], and S. Carmo Silva[3]

[1] Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco, Av. do Empresário, 6000-767 Castelo Branco, Portugal
nogf@ipcb.pt

[2] Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA
M.dehghani@northeastern.edu

[3] Department of Production and Systems, ALGORITMI Research Unit, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
scarmo@dps.uminho.pt

Abstract. Job release is an essential scheduling function and a core part of every production planning and control system. Essentially, job release has to do with the timing and the jobs to release on to the shop floor, in such way that, a balanced and restricted workload is achieved. In this paper, an Iterative Optimization-based Simulation (IOS) decision support tool is proposed for job release. This is in line with Industry 4.0 paradigm, allowing the autonomous selection of jobs based on the current shop floor situation. This decision support tool is implemented using SIMIO as a simulation manager, MATLAB as an optimization manager and MySQL as a database manager.

Keywords: Decision support system · Load-based job release
Simulation-optimization

1 Introduction

Workload control (WLC) is an approach to production planning and control, specifically designed to meet the needs of high-variety make-to-order shops (e.g. Stevenson et al. [1]). A key feature of WLC is the use of a pre-shop pool. Jobs are held in this pool and released to the shop floor in a controlled manner to meet certain performance criteria, e.g., stabilized work-in-process, reduced manufacturing throughput times, short and reliable delivery dates to costumers.

While there exists a broad literature on controlled job release, most reported load-based release procedures typically use a sequential approach to select jobs from the pool to release to the shop floor. According to this strategy, once the job's workload fits within pre-established load limits, or norms, for the stations in its manufacturing routing, the release of the job is not re-evaluated. As a result, the workload of some stations may be far below their norm value, while others may have reached its workload limit. To

© Springer International Publishing AG, part of Springer Nature 2018
Á. Rocha et al. (Eds.): WorldCIST'18 2018, AISC 746, pp. 706–713, 2018.
https://doi.org/10.1007/978-3-319-77712-2_66

minimize these imbalances, an alternative approach to job release is to use optimization models to select the set of jobs that achieve the best overall workload balance.

The rise of Industry 4.0 has highlighted the use of optimization tools for process control. The Industry 4.0 paradigm also requires modelling of manufacturing via the virtual factory concept. Thus, in this paper we introduce an iterative optimization-based simulation (IOS) decision support tool for controlled job release, imbedding the SIMIO™ virtual manufacturing model and the MATLAB™ numerical computing environment. The aim of the proposed IOS tool is to autonomously select jobs from a pre-shop pool to release to the shop floor, based on the current shop floor situation, by levelling workload along stations and thus improving the production flow.

The rest of this paper is organized as follows: a literature review on load-based job release and simulation-optimization is provided in Sect. 2; and details of the decision support tool for the controlled job release are provided in Sect. 3. Finally, conclusions, future research work and manufacturing implications are discussed in Sect. 4.

2 Literature Review

This section provides a concise literature review regarding load-based job release and simulation-optimization. For a comprehensive review on the former topic we refer interested readers to [1, 2], and for the latter topic we suggest [3, 4].

2.1 Job Release

Job release is an essential scheduling function and a core part of every production planning and control system. Controlled job release aims at firmly controlling shop throughput times of jobs, towards meeting the promised delivery dates. This means that jobs (or orders) are not immediately released to the shop floor as they arrive to the production system, but wait in a pre-shop pool for available capacity at the shop floor. Jobs are then selected from this pool and released, aiming at meeting specific system performance criteria.

As emphasized by Kingsman [5], job release is only effective if the pool of unreleased orders is controlled. Otherwise, jobs may remain in the pool for too long, missing their promised delivery dates. Nevertheless, holding back jobs in a pre-shop pool tend to absorb fluctuations of the incoming flow of jobs and stabilises the performance of the shop. This approach provides an opportunity for companies to benefit from *lean manufacturing* concepts associated with the *work-in-process* reduction and a more *transparent* shop floor situation.

Thus, Jobs' release aims at restricting and balancing workload in the shop floor. A typical approach is to allow the release of a job only if it does not exceed an upper workload limit or workload norm. Load-limiting methods produce an implicit balance of workload among stations. While different load-limiting methods have been presented in the literature, the release procedure executed across these methods is similar and can be formulated as follows:

1. All jobs in the pre-shop pool are sorted according to a pool sequencing rule.
2. The job with the highest priority is considered for release first.
3. If the resulting workload of releasing the job, together with the current workload at stations, fits within the stations' norms, then the job is selected for release. Otherwise, remains in the pool and the next job in the sequence is considered for release.

This sequential approach to select jobs from the pool to release to the shop floor is a greedy algorithm. Most load-based release procedures typically use this approach, e.g., [6–10]. While this is justified by its simplicity, the advent of Industry 4.0 and its advanced scheduling techniques question its adequateness. An alternative approach to job release applies mathematical models to select a set of jobs to be released to the shop floor that optimize a given performance criteria. Authors such as [11–13] apply this approach. Irastorza and Deane [11] propose a mathematical formulation to solve the workload balance problem respecting due dates. Portioli-Staudacher and Tandardini [12] introduce an optimization model that aims at release homogeneous workloads on each station to improve the production flow in a pure flow shop. Recently, Yan *et al.* [13] use an integer linear programming model to manage the trade-off between the timing and load balancing functions of job release.

2.2 Simulation Optimization

Simulation-Optimization (SO) is an alternative to approximate deterministic procedures to deal with highly stochastic, large and complex systems [14]. Mathematical modelling is a preferred approach when the problem size is small and complexity is low [15]; whereas the SO, that is a more favourable approach for problems with large size and higher complexity. By the increase of computational power, SO has seen a remarkable growth [16] and therefore these methods have gained popularity [17].

Iterative Optimization-based Simulation (IOS) is an emerging field of SO [3] which integrates optimization techniques into simulation analysis. Optimization is based on an analytical model and does not need simulation to evaluate system impact. Optimization may be called during simulation execution at periodic time intervals or on an event-driven basis.

This iterative process is applied by [18] in a pipeline management problem. Later, Gupta and Sivakumar [19] presented a Conjunctive Simulated Scheduling framework which combines the concepts of simulation and scheduling of a manufacturing scenario. Here, as soon as a resource is released, a new job must be selected as the next operation on this resource. Also, Sivakumar [20] implemented a discrete event simulation-based "on-line near-real-time" dynamic multi-objective scheduling system. The proposed framework is successfully verified in a semiconductor back-end site. Most recently, Dehghanimohammadabadi et al. [16] designed, developed, and implemented a novel IOS framework by integrating three modules namely, Simulation, Optimization, and Databases, capable of updating the simulation model environment using an optimization solution upon the occurrence of predictable and unpredictable changes.

3 Decision Support Tool for Job Release

This section introduces the decision support tool deployed for load-based job release. Section 3.1 first introduces the developed IOS framework used in this study, and then, the simulation model and the optimization procedure are presented in Sect. 3.2.

3.1 Iterative Optimization-Based Simulation Framework

The proposed decision support tool combines mathematical programming and discrete event simulation using the IOS framework of Dehghanimohammadabadi [21] and Dehghanimohammadabadi and Keyser [22]. In the IOS framework, a MATLAB optimization module is embedded into a simulated model developed in SIMIO. This framework is used in our decision support tool to optimize the job release function. A MySQL database is also used for data interchange between this two software packages using an OBDC interface. Input data to the optimization module and the corresponding output solution data are stored in this database. Input data concerns to the jobs currently waiting release at the pre-shop pool and their respective processing times and the current workloads at workstations, while output data concerns to the jobs that must be released.

Unlike existing simulation-optimization frameworks where optimization is used to find the best scenario of the simulated system, in the IOS approach, the optimization is an inner component of the simulation engine, and can be triggered several times within a single simulation run. In other words, periodically the simulation run is paused, and the optimization is triggered. After optimization, simulation is re-primed with the updated results from the optimization manager. Figure 1 depicts the iteration of a simulation run in which the optimization is triggered twice to optimize a system's setting.

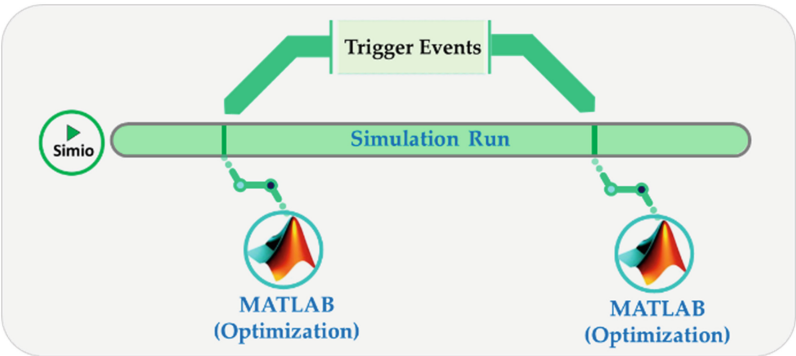

Fig. 1. Iterative Optimization-based Simulation (IOS) approach (Dehghanimohammadabadi [21])

3.2 Simulation Model and Optimization Procedure

As case study, a simulation model of a make-to-order production line was developed using SIMIO software, which is depicted in Fig. 2. The production line contains six

workstations, where each is a single server. The SIMIO virtual manufacturing environment provides the means and the visibility to make decisions at job release level, but also a visual representation of the entire production system, allowing management to see how their products flow in the system.

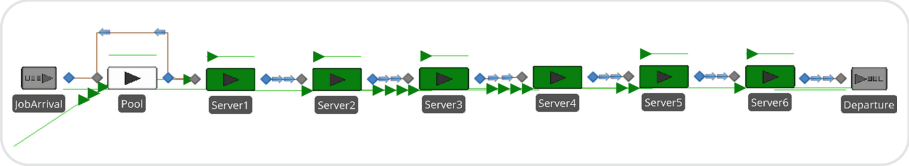

Fig. 2. Simulation model of a production line using SIMIO software.

In this simulation model, arriving jobs flow into a pre-shop pool where they wait release to the production system. Job release is activated periodically, e.g., at the beginning of each working shift. At that moment, simulation data, concerning the jobs currently waiting release at the pre-shop pool and their respective processing times and the current workloads at workstations, is updated in the MySQL database using the Simio "DbWrite" step (as depicted in Fig. 3). Then, in case there are waiting jobs in the pool, the MATLAB optimization module is triggered using the "CallMatlab" step developed by Dehghanimohammadabadi and Keyser [22]). This will select the jobs that must be released to the shop floor by updating the status of waiting jobs in the MySQL database. This information is then *uploaded* into the simulation model using the "DbRead" step, as depicted in Fig. 3. Finally, a "Decide" step is used to route the job to the shop floor if selected for release or to maintain the job in the pool otherwise.

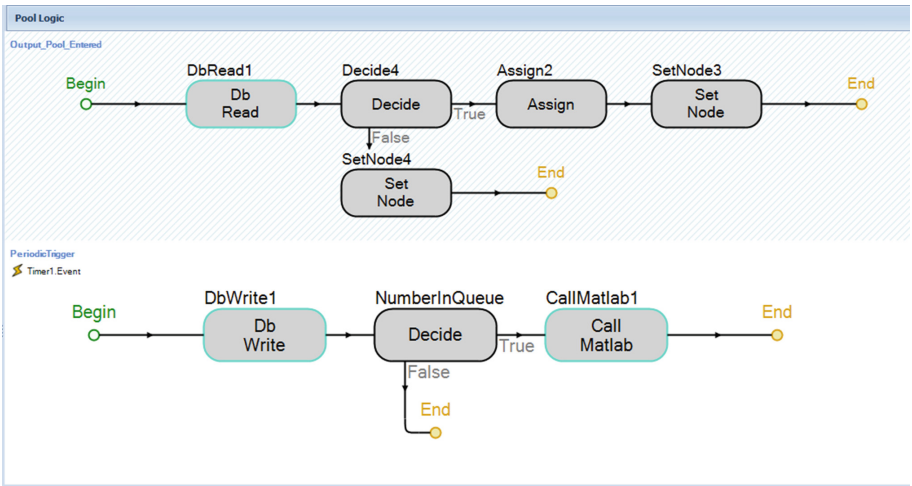

Fig. 3. SIMIO processes to call the MATLAB optimisation module and to write (read) from (to) the MySQL database.

Note that releasing a set of jobs is only feasible if it does not violate the workload norms of the stations in the routing of the job. An integer linear programming model is used, at the MATLAB optimization module, to solve the jobs' selection problem. The model is given by equations below:

$$\text{Max} f = \sum_{s=1}^{S} \frac{L_s}{\lambda_s - W_s} \tag{1}$$

Subject to:

$$L_s \leq \lambda_s - W_s, \forall s = 1 \ldots S \tag{2}$$

where s is the station index, S is the number of stations in the production system, $\lambda_s - W_s$ indicates the load gap at station s, λ_s the load norm at s, W_s the current load at s and L_s the load released to s after the release decision, and is obtained as follows:

$$L_s = \sum_{j=1}^{J} y_{sj} x_j \tag{3}$$

where j is the job index, J is the number of jobs in the pre-shop pool when job release is activated, x_j are the decision variables, taking the value of one or zero to indicate whether job j is selected for release or not, respectively and y_{sj} the load job j will add to station s. The release decision is thus clearly defined by setting the decision variables for the jobs in the pre-shop pool.

The optimization model sets the decision variables so that the objective function is maximized given constraint defined in (2), i.e., the released load approaches the load gap at stations as much as possible. Once the release decision has been taken, the workload of released jobs is credited to the stations' load and jobs are removed from the pre-shop pool list.

4 Conclusions and Managerial Implications

In this paper, an iterative optimization-based simulation (IOS) decision support tool for job release is proposed. The tool works within a framework that imbeds a SIMIO virtual manufacturing model, a MATLAB numerical computing environment, and a MySQL database that interconnects both software's. SIMIO is used to run the virtual model of a manufacturing system, while MATLAB enables simulation to solve the optimization problem using integer linear programming.

The proposed decision support tool could be used to autonomously select jobs from a pre-shop pool to release to the shop floor, based on real-time data from intelligent logistic objects, such as machines, carriers, etc. This implements a step forward in the intelligence of commonly used greedy algorithms to solve the job release function.

Future research work should extend the applicability of the proposed IOS decision support tool to other manufacturing scheduling decisions, such as order entry and priority dispatching at the shop floor level.

Acknowledgments. This work had the financial support of FCT - Fundação para a Ciência e Tecnologia of Portugal under the project PEst2015-2020: UID/CEC/00319/2013.

References

1. Stevenson, M., Hendry, L.C., Kingsman, B.G.: A review of production planning and control: the applicability of key concepts to the make-to-order industry. Int. J. Prod. Res. **43**(5), 869–898 (2005)
2. Fredendall, L.D., Divesh, O., Patterson, J.W.: Concerning the theory of workload control. Eur. J. Oper. Res. **201**, 99–111 (2010)
3. Figueira, G., Almada-Lobo, B.: Hybrid simulation optimization methods: a taxonomy and discussion. Simul. Model. Pract. Theory **46**, 118–134 (2014)
4. Tekin, E., Sabuncuoglu, I.: Simulation optimization: a comprehensive review on theory and applications. IIE Trans. **36**(11), 1067–1081 (2004)
5. Kingsman, B.G.: Modelling input–output workload control for dynamic capacity planning in production planning systems. Int. J. Prod. Econ. **68**, 73–93 (2000)
6. Land, M., Gaalman, G.: The performance of workload control concepts in job shops: improving the release method. Int. J. Prod. Econ. **56–57**(20), 347–364 (1998)
7. Land, M.: Parameters and sensitivity in workload control. Int. J. Prod. Econ. **104**(2), 625–638 (2006)
8. Thürer, M., Fernandes, N.O., Stevenson, M., Qu, T.: On the backlog-sequencing decision for extending the applicability of ConWIP to high-variety contexts: an assessment by simulation. Int. J. Prod. Res. **55**(16), 4695–4711 (2016)
9. Fernandes, N.O., Silva, C., Silva, S.C.: Order release in the hybrid MTO-FTO production. Int. J. Prod. Econ. **170**(Part B), 513–520 (2015)
10. Fernandes, N.O., Thurer, M., Silva, C., Carmo-Silva, S.: Improving workload control order release: incorporating a starvation avoidance trigger into continuous release. Int. J. Prod. Econ. **194**, 181–189 (2017)
11. Irastorza, J.C., Deane, R.H.: A loading and balancing methodology for job shop control. AIIE Trans. **6**(4), 302–307 (1974)
12. Portioli-Staudacher, A., Tantardini, M.: A lean-based ORR system for non-repetitive manufacturing. Int. J. Prod. Res. **50**(12), 3257–3273 (2012)
13. Yan, H., Stevenson, M., Hendry, L.C., Land, M.J.: Load-Oriented Order Release (LOOR) revisited: bringing it back to the state of the art. Prod. Plann. Control **27**(13), 1078–1091 (2016)
14. Lin, J.T., Chen, C.M.: Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. Simul. Model. Pract. Theory **51**, 100–114 (2015)
15. Klemmt, A., Horn, S., Weigert, G., Wolter, K.J.: Simulation-based optimization vs. mathematical programming: a hybrid approach for optimizing scheduling problems. Robot. Comput.-Integr. Manuf. **25**(6), 917–925 (2009)
16. Dehghanimohammadabadi, M., Keyser, T.K., Cheraghi, S.H.: A novel Iterative Optimization-based Simulation (IOS) framework: an effective tool to optimize system's performance. Comput. Ind. Eng. **111**, 1–17 (2017)
17. Dehghanimohammadabadi, M., Keyser, T.K.: Tradeoffs between objective measures and execution speed in iterative optimization-based simulation (IOS). In: Proceedings of the Winter Simulation Conference, pp. 2848–2859. IEEE Press (2015)

18. Subramanian, D., Pekny, J.F., Reklaitis, G.V.: A simulation–optimization framework for addressing combinatorial and stochastic aspects of an R&D pipeline management problem. Comput. Chem. Eng. **24**(2), 1005–1011 (2000)
19. Gupta, A.K., Sivakumar, A.I.: Conjunctive simulated scheduling. Int. J. Adv. Manuf. Technol. **26**(11–12), 1409–1413 (2005)
20. Sivakumar, A.I.: Multiobjective dynamic scheduling using discrete event simulation. Int. J. Comput. Integr. Manuf. **14**(2), 154–167 (2001)
21. Dehghanimohammadabadi, M.: Iterative Optimization-based Simulation (IOS) with Predictable and Unpredictable Trigger Events in Simulated Time. Western New England University (2016). http://gradworks.umi.com/10/03/10032181.html
22. Dehghanimohammadabadi, M., Keyser, T.K.: Intelligent simulation: integration of SIMIO and MATLAB to deploy decision support systems to simulation environment. Simul. Model. Pract. Theory **71**, 45–60 (2017)