Buffer Allocation in Unreliable Production Lines Using Infinitesimal Perturbation Analysis and Genetic Algorithm

Khelil Kassoul, Rakesh D. Raut, Samir Brahim Belhaouari, and Naoufel Cheikhrouhou

1 Introduction

A production line is composed of a sequence of machines, storage buffers, transportation components, and other elements that interact to deliver products [1]. Production lines are stochastic systems of product flow that alternate resources between machines and storage spaces with a finite capacity. Random occurrences like human operation challenges, quality issues, machine failures, and service time variations at various points in the system might temporarily stop machine operations. These sorts of interruptions have an impact on how efficiently production systems operate since even a little change in the design requirements might have a significant negative impact on performance. Optimally designing these lines therefore requires taking into account a large number of variables characterizing these resources (variability of cycle times, maximum storage capacities, repair and failure time distributions, etc.) [2]. The Buffer Allocation Problem (BAP), which concerns the allocation of buffer capacities and their placement inside the line, is the topic of several studies [3]. This article presents the results of the development of a new optimization technique based

K. Kassoul (✉) · N. Cheikhrouhou
Geneva School of Business Administration, University of Applied Sciences Western Switzerland, HES-SO, 1227 Geneva, Switzerland
e-mail: khelil.kassoul@hesge.ch

R. D. Raut
Departmet of Operations and Supply Chain Management, National Institute of Industrial Engineering (NITIE), NITIE, Vihar Lake, Powai, Mumbai, Maharashtra 400087, India

S. B. Belhaouari
Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Ar-Rayyan, Qatar

N. Cheikhrouhou
IFM Business School, 1205 Geneva, Switzerland

A. J. Kulkarni and N. Cheikhrouhou (eds.), *Intelligent Systems for Smart Cities*,
https://doi.org/10.1007/978-981-99-6984-5_28

on the combination of Infinitesimal Perturbation Analysis (PA) and Genetic Algorithm (GA) via simulation for solving the BAP. The paper is structured as follows. Section 2 provides a brief overview of production line optimization approaches for BAP. The mathematical problem formulation and the solution strategy are given in Sect. 3. Section 4 provides the proposed approach, while presenting the associated algorithms. Section 5 presents numerical examples conducted and discussions of the results. Finally, in Sect. 6, conclusions and future research directions are discussed.

2 Literature Review

The BAP is an NP-hard problem in terms of combinatorial complexity [4]. Several researchers addressed the different techniques used to solve this issue [5, 6]. Optimization via simulation is one of the techniques that couples simulation to a solution-finding algorithm. The simulation allows for the evaluation of the performance measure (e.g., the production rate) at each execution, while the optimization algorithm provides directions in the search space for new system design solutions. Authors in [7] conduct symbolic regression using genetic programming. They propose simulation-based meta-models of industrial systems using a sampling strategy modified for genetic programming. Authors in [8] utilize simulation to identify the best buffer allocation that improves the production line's reliability by reducing machine micro-downtime. They take into account the effect of the cost on the buffer allocation in their simulation. Authors in [9] employ a continuous-line approach rather than discrete lines to optimize the PR with respect to buffer capacity in order to identify the optimal buffer allocations. Most of the methods of optimization via simulation require a large number of simulations before finding a good solution to the problem [10] and have, thus, the disadvantage of consuming significant computation time. Therefore, this study focuses to find the best buffer allocation technique while also reducing the amount of time required to attain convergence. This is why we are interested in Perturbation Analysis methods (PA) [11] because they propose a design solution that significantly reduces calculation times by using a unique long simulation of the system. PA can be separated into two main categories: Infinitesimal (or zero order) Perturbation Analysis (IPA) [11] and Finite (or first order) Perturbation Analysis (FPA) [12, 13]. The main difference between IPA and FPA is that a perturbation in the former refers to a small (infinitesimal) delay in a transition time, whereas a perturbation in the latter refers to a jump (finite perturbation) on a sample path from one state to another due to parameter changes. By calculating the function derivatives acquired in a single simulation run, the PA is employed as a method for evaluating the performance of discrete event systems [14–16]. Due to its ease and simplicity of implementation, IPA is often utilized. It has been formerly mostly used with queuing networks, but it is currently principally used with stochastic fluid models [15]. For the purpose of minimizing a cost function, [16] identify, by calculating the gradient of the cost function, the optimal capacities of the finished items and work-in-process buffers. To optimize the parameters (buffer size) of a discrete event system, authors in [17] study the application of several developed IPA estimates based on a stochastic fluid model. Authors in [18] propose an effective method for designing buffer capacities in assembly/disassembly systems.

To determine the optimal size for each buffer storage, authors in [19] propose a GA integrated to line-search approach. To identify the buffer sizes needed in open serial production lines to optimize the system's average production rate, a hybrid technique-based simulation optimization is presented by authors in [20]. To find potential buffer sizes, the authors develop a hybrid strategy combining a genetic algorithm and simulated annealing, and to obtain the average production rate of the line, they use discrete event simulation model as an evaluative tool. A multi-objective GA was recently developed by authors in [21] to allocate the service times and buffers and provide a range of solutions for several objective functions. Only 4 out of 95 research published after 1998, according to [6], use a hybrid strategy to improve the buffer sizes. There are two contributions in this paper. First, the development of a general optimization technique (IPA coupled with GA) for designing buffer capacities for stochastic production lines. Second, diverse production system sizes may be considered by the methodology.

3 Problem Formulation

Consider the production line shown in Fig. 1, where parts are processed on the sequence of the unreliable machines $M_1, .., M_n$. Consider the vector of decision variables $s = (s_1, s_2, ..., s_{n-1})$ of dimension n-1, where the set $\{s_1, s_2, ..., s_{n-1}\}$ are integers and denotes the storage capacities of the available physical locations and n is the number of machines. B_s is the total storage capacity available for allocation on all the locations of the production line and PR is the Production Rate. The design problem addressed consists of allocating B_s over $(n-1)$ stocks to maximize the production rate PR:

$$\text{Fin s} = (s_1, s_2, .., s_{n-1}) \text{ so as to maximize PR} \quad (1)$$

$$\text{Subjec to}: \sum_{i=1}^{n-1} s_i = B_s; \quad s_i \text{ are positive integers (for each i)} \quad (2)$$

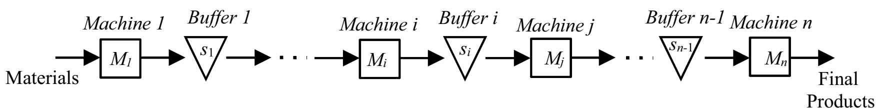

Fig. 1 Open flow production line

4 Proposed GA-IPA

This study proposes a hybrid simulation–optimization method that combines GA with IPA to solve the BAP in unreliable production lines. GA is used to identify the input solutions to the IPA, which in turn is used to estimate the gradients of the PR with respect to buffer capacities. The gradients are subsequently incorporated into a stochastic algorithm to determine the optimal buffer capacities. GA is chosen to cover a wide search area, thanks to its diversification and exploration capabilities, while IPA is employed to investigate each area more extensively, by taking advantage of its intensification and exploitation capabilities. One of the advantages of this method is the use of a single simulation only, which is achieved through the use of IPA to evaluate gradients of the PR at regular intervals during the same simulation run.

4.1 Genetic Algorithm (GA)

One of the often-used evolutionary algorithms is GA [22]. GA requires a population of individuals (or solutions; usually generated randomly). GA then repeats a number of steps (mutation, crossover and selection) for a predetermined generations' number. We use in this study: (1) Tournament selection [23], which is effective and robust selection mechanism usually used by GA, (2) Arithmetic Crossover [24], where the selected parents (actual population) are linearly combined to generate new offspring (new population), and (3) Mutation: to satisfy the constraint (2), an adjustment procedure is used to guarantee that the individual (i.e., buffer size) is feasible (i.e., an integer).

4.2 Infinitesimal Perturbation Analysis (IPA)

4.2.1 Choice of IPA

This section is devoted to determine an estimate of $\frac{\partial f}{\partial s_i}$ for $i = 1, \ldots, n-1$, and where f denotes the PR of the production line i.e., we search for a value of $\frac{\partial E[f]}{\partial s_i}$ ($E[f]$ is the average PR determined by observing all sample trajectories). These estimators are unbiased if the following condition is fulfilled:

$$\frac{\partial E[f]}{\partial s_i} = E\left[\frac{\partial f}{\partial s_i}\right] \tag{3}$$

Note that with two alternative values of the buffer size, a brute force simulation of the system may often approximate the left-hand side of (3) by $\frac{\sum_{k=1}^{n}\left[f(s_i+\delta,\omega_k)-(f(s_i,\omega_k)\right]}{\partial s_i}$ (where ω_k is the random characteristic of the realization of an event). A single Monte Carlo simulation employing IPA is used to get the right-hand side. According to the IPA theory, sample trajectories do not significantly affect the average value of the process if, for a small discrete perturbation in the size of s_i, both the probability to encounter a discontinuity in f from one path to another and the discontinuity's value, if there is one, are reasonably small [25]. Then, the constraint (3) is met if a long sample trajectory is chosen and:

$$\frac{\partial E\left[f\right]}{\partial s_i} = \frac{f^{\delta}}{\partial s_i} \tag{4}$$

where f^{δ} is the total gain, due to the introduction of an infinitesimal perturbation δ, determined by using generation and propagation rules on a production line.

4.2.2 Rules of Generation and Propagation of Perturbation in Production Lines

The evolution of the production line in a perturbed environment is predicted using different propositions and rules. The prediction is constructed from a nominal sample trajectory using information on each server status at each event [2, 12]. Since the machines are unreliable, they may execute operations (up-state) or are down because of internal failures (down-state). In the up-state, a machine can be Full Output (FO) or blocked (resp. Null Input (NI) or starved) if its downstream (resp. upstream) buffer is full (resp. empty). We assume that the output machine (resp. input machine) cannot be blocked (resp. starved).

IPA algorithm pseudo-code

```
Beginning
Step 1. Initialization: Sum_j = 0     (for i= 1 to n-1)
Step 2. If  M_i    is FO        then (for the first time)
                       Sum_i = t_i · δ

Step 3. If  M_j    is NI        then
          Sum_j = {Sum_i + [NI], Sum_j} − {0, [NI]}

Step 4. If  M_i is FO           then
          Sum_i = {Sum_j + [FO], Sum_i} − {0, [FO]}

Step 5. If  i=n                 then  (Last machine)
          ∂f/∂b_i = −(f/T) · Sum_n    ;   f* = P/(T−Sum_n)    ;
        Stop  f* represents the estimated PR in the
              perturbed path and T is the duration of
              the simulation)
        Else      Go to Step 1
End
```

4.3 Stochastic Algorithm

A stochastic algorithm is used to find the optimal buffer sizes using the gradient estimations provided by IPA. This algorithm is based on the multi-dimensional Robbins and Monro's procedure to optimize stochastic systems [26] which considers gradients from the projection of the PR on the hyperplane of the constraint: $\sum_{i=1}^{n-1} s_i =$ constant. Then, the algorithm updates intermediate buffer sizes by updating in the gradient's direction at each iteration. We develop a single run optimization technique i.e., at each simulation of P units (P < L), where L is the total number of parts to produce during the simulation.

Stochastic algorithm pseudo-code

```
Beginning
Step 1. k=1. choose the initial values for all s_i^k (for
        i = 1,…,n − 1)
Step 2. Simulate L parts and calculate gradients ∂f/∂s_i at
        iteration k using IPA.
Step 3. for all s_i do s_i^{k+1} = s_i^k + α/k (∂f/∂s_i − 1/(n−1) Σ_{i=1}^{n−1} ∂f/∂s_i)
Step 4. If s_i^{k+1} ≤ 0 then
            s_p^k = Arg min s_i^k ; s_p^{k+1} = Arg min s_i^{k+1};
            aU (0,1) ; d = |α·s_p^k / (s_p^{k+1} − s_p^k)|
            s_i^{k+1} = s_i^k + d · α/k (∂f/∂s_i − 1/(n−1) Σ_{i=1}^{n−1} ∂f/∂s_i)

Step 5. If |s_p^{k+1} − s_p^k| ≤ ε then s_i= Anint(s_i^{k+1})
            stop
        If P parts are simulated then
            stop
        Else k=k+1, go to step2.
End
```

The slope of function f at the optimum determines the stopping criterion i.e., the algorithm converges, through the optimization process, when a local maximum is found. It follows that the selected criteria $\frac{\alpha}{k} \cdot max_i^n\left(\frac{\partial f}{\partial s_i} - \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{\partial f}{\partial s_i}\right) < \varepsilon$ (which denotes, at a given iteration, the best progression of the buffer capacity on the slope) is a simple acceptable stopping criterion for the previous algorithm at each iteration k (where α is a random constant determined by choosing an initial value to keep, after each iteration, the value of the s_i in the same order). This stopping condition is a guarantee that at least a local optimum will be found.

5 Numerical Experiments

The first set of experiments compare the outcomes of our algorithm with those of state-of-the-art algorithms involving 5, 10 and 20-machines production line. The second set examines the effectiveness of the proposed method on large production lines (i.e., production line with 40 and 50-machines). The simulation models are built using the discrete event simulation software Arena 14.0 [27] and the algorithms are coded in Java. In all experiments, the repair and failure times follow a geometric distribution with probabilities of r_i and f_i, respectively. Determining the appropriate number of replications is crucial as a high number of runs can result in long computational times, whereas a small number of runs may lead to biased solutions. Additionally, the operator parameters of GA play a critical role in the convergence of the algorithm. Well-designed operators can result in faster convergence and in obtaining the best possible solution for a given initial population [28]. To achieve this, we conduct preliminary experiments that show that a number of runs of 30, 20 generations and 30 individuals in each generation is the optimal setting. The algorithm terminates when it reaches a predetermined number generations (i.e., 20 generations) or when there is no improvement in the value of PR.

We compare our method to seven benchmark algorithms: Genetic Algorithm with Finite Perturbation Analysis (GA-FPA) [12], Simulated Annealing with Genetic Algorithm (GA-SA) [20], Tabu Search with Analytical Decomposition Approximation (ADA-TS) [29], Immune Decomposition Algorithm (IDA) [30], Degraded Ceiling method with decomposition approximation technique (DC) [31], Dual Gradient Technique (DGT) [32], and Gradient Technique (GT) [33]. If a method's results are not provided for an experiment, it simply indicates that the results are not published. The best results are bolded in all tables.

5.1 Results on Small/Medium Dimensions

5.1.1 Five-Machine Production Line

Table 1 presents the production line's parameters. The total available buffer space is 31 parts. The simulation model is executed for 50 runs and a total of 100,000 parts.

The average production rates and the corresponding buffer size configuration achieved by the various methods are shown in Table 2. With the proposed GA-IPA, the buffer size configuration {6, 9, 11, 5} yields the best average PR value of 0.4927 which represents an approximately 0.3% lower than the average PR of 0.4943 found by GA-SA [20], ADA-TS [29] and DGT [32]. The results provided in [20, 29, 30, 32] have all the same allocation pattern at the line's extremities as using GA-IPA. Indeed, allocating small capacities at the extremities of the line and large capacities in the middle prevent from any potential line congestion.

Table 1 Parameters of 5-machine line

Machine	1	2	3	4	5
$1/r_i$	11	19	12	7	7
$1/f_i$	20	167	22	22	26

Table 2 Results on 5-machine line

Method	Buffer allocation				Avg. PR
GT	5	11	8	7	0.4914
DGT	7	10	10	4	**0.4943**
ADA-TS	7	10	10	4	**0.4943**
IDA	6	10	11	4	0.4941
GA-SA	7	10	10	4	**0.4943**
GA-IPA	6	9	11	5	0.4927

5.1.2 Ten-Machine Production Line

Table 3 provides the failure and repair rates for a production line of 10 machines. A total buffer capacity of 270 items is available. The simulation model is executed for 30 runs and a total of 10,000 parts.

As shown in Table 4, identical buffer configuration {14, 19, 30, 54, 45, 27, 23, 24, 34} is obtained in [29] and [31] with an average PR of 0.64135. With nearly the same allocation of buffers of {14, 19, 30, 52, 47, 27, 23, 24, 34}. Authors in [20] obtain a slightly better value of PR of 0.64139. GA-IPA produces the best PR with a modified allocation of buffers of {19, 20, 38, 43, 44, 22, 32, 21, 31}. It should be noted that, in most cases, all approaches provide allocations of buffer capacities with larger storage space assigned in the center of the line. Most likely, this buffer allocation pattern prevents blockages and facilitate the items flows.

Table 3 Parameters of 10-machine line

Machine	1	2	3	4	5	6	7	8	9	10
$1/r_i$	7	7	5	10	9	14	5	8	10	10
$1/f_i$	20	30	22	22	25	40	23	30	45	20

Table 4 Results on 10-machine line

Method	Buffer allocation									Avg. PR
DC	14	19	30	54	45	27	23	24	34	0.64135
ADA-TS	14	19	30	54	45	27	23	24	34	0.64135
IDA	14	19	30	52	47	27	23	24	34	0.64139
GA-SA	7	16	48	61	24	41	20	34	19	0.63016
GA-IPA	19	20	38	43	44	22	32	21	31	**0.64526**

Table 5 Results on 20-machine line

Parameters		Avg. PR		
r_i	f_i	ADA-TS	GA-SA	GA-IPA
0.5	0.5	0.377895	0.382401	**0.419729**

5.1.3 20-Machine Production Line

A total space for 100 items is available. The parameters of the production line are presented in the two first columns in Table 5. Column 3 and 4 provide the results obtained by [19] and [26], respectively. The results of our method are presented in the last column. As shown in Table 5, the developed hybrid algorithm GA-IPA outperforms ADA-TS and GA-SA. Consequently, the developed GA-IPA strategy provides a clear advantage for solving BAP in large-scale production lines.

5.2 Results on Large Dimensions

There have been very few studies devoted to the buffer allocation problem for large production lines (e.g., [20, 34]). Researchers in this field assert that it takes very long computation times to obtain competitive results (typically dozens of hours). In other words, when there are more machines, the calculation takes longer to complete. In this experiment, we consider production lines with 40 and 50 unreliable machines. We compare the GA-IPA algorithm with the GA-FPA proposed in a previous work [12]. The repair r_i and failure f_i rates are geometrically distributed with $r_i = f_i = 0.5$. The experiment is carried out for 5 runs due to the very long computation times needed. The total buffer size is given in the second column of Table 6. Column 3 and 4 provide the comparative results of GA-FPA and GA-IPA (including small and large-scale production lines).

The results of GA-FPA are always better than GA-IPA. This may be explained by the fact that GA-IPA only considers total or zero propagations of perturbation unlike GA-FPA which additionally takes into account partial propagation of perturbations,

Table 6 Results on large production lines

Case	B_{max}	Avg. PR		$\nabla PR(\%)$
		GA-FPA	GA-IPA	
5 machines	31	0.494833	0.492713	0.43
10 machines	270	0.649203	0.645263	0.60
20 machines	100	0.422175	0.419729	0.58
40 machines	200	0.415568	0.415021	0.13
50 machines	245	0.410301	0.410039	0.06

as they refer to creation or deletion of periods of blockage (FO) or starvation (NI). It is noticeable that the differences in results between GA-FPA and GA-IPA are small (see last column of Table 6) and they are very smaller as the production line is larger. This confirms the idea that a long nominal trajectory and a larger system size are more conducive to reproduce the occurrence of all the possible events (even rare events), and to the propagation and accumulation of perturbation at the exit of the system. The results are therefore more reliable as long as the perturbed trajectories are statistically similar to the nominal one.

6 Conclusion

In this paper, we combine infinitesimal perturbation analysis with genetic algorithm to design serial production lines with unreliable machines and present an efficient approach GA-IPA for allocating buffer sizes.

Experiments using infinitesimal order analysis are conducted and the results are experimentally compared with first (finite) order analysis (GA-FPA) for some production lines. We notice that the predictions obtained by GA-FPA are closer to those obtained by GA-IPA but are always better than the GA-IPA ones. This may be explained by the use of IPA (infinitesimal variation) with finite discrete parameters (buffer capacity). Therefore, for discrete parameters (e.g., buffer size) or for large parameter changes, the FPA technique offers a good accuracy in the results. Although, to reduce the computational time, the estimators provided by the API technique are retained as the gradients due to the simplicity of calculation and the ease of implementation. Thus, both GA-FPA and GA-IPA algorithms provide good estimates of the performance measure and lead to performant results. However, the choice of the technique to use (FPA or IPA) depends on the system to be studied (i.e., nature of decision variables, performance measure, complexity of the system, etc.) and on the unbiasedness of the gradient estimators.

Exploring the limitations of GA-IPA in terms of solution quality and computation time and assessing whether our proposed method remains effective when dealing with up to 100 machines in a production system is a first research perspective. Furthermore, as this study focuses only on maximizing production rate as a single objective a new direction would be to investigate the extension of our approach to the case when multiple objectives are considered, such as minimizing the work-in-process (WIP) inventory or the lead time. A multi-objective approach could provide more comprehensive insights into the trade-offs between various performance metrics and enable more informed decision-making in the design and planning of AD systems.

References

1. Gershwin, S. (1994). Manufacturing systems analysis. *Prentice Hall, Engle-wood Cliffs, New Jersey, 7632*, 69–93.
2. Cheikhrouhou, N. (2001). *« Apport de l'analyse de perturbation pour le dimensionnement et le pilotage des systèmes de production »*. PhD Thesis, Grenoble INPG.
3. Dolgui, A., Eremeev, A. V., & Sigaev, V. S. (2007). HBBA: Hybrid algorithm for buffer allocation in tandem production lines. *Journal of Intelligent Manufacturing, 18*(3), 411–420.
4. Xi, S., Smith, J. M., Chen, Q., Mao, N., Zhang, H., & Yu, A. (2022). Simultaneous machine selection and buffer allocation in large unbalanced series-parallel production lines. *International Journal of Production Research, 60*(7), 2103–2125.
5. Weiss, S., Schwarz, J. A., & Stolletz, R. (2019). The buffer allocation problem in production lines: Formulations, solution methods, and instances. *IISE Transactions, 51*(5), 456–485.
6. Demir, L., Tunali, S., & Eliiyi, D. T. (2014). The state of the art on buffer allocation problem: A comprehensive survey. *Journal of Intelligent Manufacturing, 25*(3), 371–392.
7. Can, B., & Heavey, C. (2011). Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. *Computers & Industrial Engineering, 61*(3), 447–462.
8. Battini, D., Persona, A., & Regattieri, A. (2009). Buffer size design linked to reliability performance: A simulative study. *Computers & Industrial Engineering, 56*(4), 1633–1641.
9. Gürkan, G. (2000). Simulation optimization of buffer allocations in production lines with unreliable machines. *Annals of Operations Research, 93*(1), 177–216.
10. Shi, L., & Men, S. (2003). Optimal buffer allocation in production lines. *IIE Transactions, 35*(1), 1–10.
11. Ho, Y.-C. (Larry)., & Cao, X.-R. (2012). *Perturbation analysis of discrete event dynamic systems*. Springer Science & Business Media.
12. Kassoul, K., Cheikhrouhou, N., & Zufferey, N. Buffer allocation design for unreliable production lines using genetic algorithm and finite perturbation analysis. *International Journal of Production Research*, *60*(10), 3001–3017. https://doi.org/10.1080/00207543.2021.1909169
13. Kassoul, K., Cheikhrouhou, N., & Zufferey, N. (2023). Simultaneous allocation of buffer capacities and service times in unreliable production lines. *International Journal of Production Research*, 1–21.
14. Ho, Y.-C., Cao, X., & Cassandras, C. (1983). Infinitesimal and finite perturbation analysis for queueing networks. *Automatica, 19*(4), 439–445.
15. Turki, S., & Rezg, N. (2018). Unreliable manufacturing supply chain optimisation based on an infinitesimal perturbation analysis. *International Journal of Systems Science: Operations & Logistics, 5*(1), 25–44.
16. Panayiotou, C., & Cassandras, C. G. (2006). Infinitesimal perturbation analysis and optimization for make-to-stock manufacturing systems based on stochastic fluid models. *Discrete Event Dynamic Systems, 16*(1), 109–142.
17. Markou, M., & Panayiotou, C. G. (2007). Optimization of discrete event system parameters using SFM-based infinitesimal perturbation analysis estimates. In *2007 46th IEEE Conference on Decision and Control* (pp. 5068-5073).
18. Cheikhrouhou, N., & Descotes-Genon, B. (2000). Buffer design in stochastic assembly/disassembly systems. *IFAC Proceedings Volumes, 33*(17), 867–872.
19. Amiri, M., & Mohtashami, A. (2012). Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. *The International Journal of Advanced Manufacturing Technology, 62*(1), 371–383.
20. Kose, S. Y., & Kilincci, O. (2015). Hybrid approach for buffer allocation in open serial production lines. *Computers & Operations Research, 60*, 67–78.
21. Cruz, F. R. B., Duarte, A. R., & Souza, G. L. Multi-objective performance improvements of general finite single-server queueing networks. *Journal of Heuristics*, *24*(5), 757−781. https://doi.org/10.1007/s10732-018-9379-8.

22. Chaudhry, S. S., & Luo, W. (2005). Application of genetic algorithms in production and operations management: A review. *International Journal of Production Research, 43*(19), 4083–4101.
23. Lei, D., Zheng, Y., & Guo, X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *International Journal of Production Research*, *55*(11), 3126-3140. https://doi.org/10.1080/00207543.2016.1262082.
24. Duman, S. (2018). A modified moth swarm algorithm based on an arithmetic crossover for constrained optimization and optimal power flow problems. *IEEE Access, 6*, 45394–45416.
25. Cao, X.-R. (1987). First-order perturbation analysis of a simple multi-class finite source queue. *Performance Evaluation, 7*(1), 31–41.
26. Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics, 22*(3), 400–407.
27. Altiok, T., & Melamed, B. (2010). *Simulation modeling and analysis with Arena.* Elsevier.
28. Pavai, G., & Geetha, T. V. (2019). New crossover operators using dominance and co-dominance principles for faster convergence of genetic algorithms. *Soft Computing, 23*, 3661–3686.
29. Demir, L., Tunali, S., & Løkketangen, A. (2011). A tabu search approach for buffer allocation in production lines with unreliable machines. *Engineering Optimization, 43*(2), 213–231.
30. Massim, Y., Yalaoui, F., Amodeo, L., Chatelet, E., & Zeblah, A. (2010). Efficient combined immune-decomposition algorithm for optimal buffer allocation in production lines for throughput and profit maximization. *Computers & Operations Research, 37*(4), 611–620.
31. Nahas, N., Ait-Kadi, D., & Nourelfath, M. (2006). A new approach for buffer allocation in unreliable production lines. *International journal of production economics, 103*(2), 873–881.
32. Gershwin, S. B., & Schor, J. E. (2000). Efficient algorithms for buffer space allocation. *Annals of Operations research, 93*(1), 117–144.
33. Ho, Y.-C., Eyler, M. A., Chien, et T.-T. (1979). A gradient technique for general buffer storage design in a production line. *International journal of production research*, 17(6), 557–580.
34. Spinellis, D., Papadopoulos, C., & Smith, J. M. (2000). Large production line optimization using simulated annealing. *International journal of production research, 38*(3), 509–541.