Integrated Scheduling of Distributed Hybrid Flowshop Production, Spare Parts Inventory, and Equipment O&M Activities Using a Learning-Assisted Co-Evolutionary Algorithm

Jingxing Zhang[1], Qianwang Deng[1,*], Xiaobo Li[1], Huaping Chen[1], Fangwei Ning[1]

[1] School of Traffic and Transportation Engineering, Central South University, Changsha 410075, Hunan, China
* Corresponding author: qwdeng@csu.edu.cn

Abstract

The interdependence between production scheduling, spare parts inventory management, and equipment operation and maintenance (O&M) represents a critical yet underexplored opportunity for cross-domain resource optimization in manufacturing systems. Conventional scheduling approaches treat these three domains independently, resulting in suboptimal resource utilization, excessive preventive maintenance downtime, and inventory holding costs that could be substantially reduced through coordinated planning. This paper formulates the Integrated Production-Inventory-O&M scheduling problem (IPIOM) for distributed hybrid flowshop environments, where manufacturers simultaneously optimize job sequencing across geographically distributed production facilities, on-hand spare parts inventory allocation to support maintenance activities, and preventive maintenance (PM) schedules coordinated with spare parts delivery timelines. The IPIOM model employs an optimal speed adjustment strategy that modulates processing speeds to synchronize production completion with spare parts availability, and defines PM trigger conditions based on equipment degradation state and predicted maintenance windows. The bi-objective formulation minimizes total manufacturer costs (production, inventory holding, maintenance, and energy) and total customer capacity loss from delayed deliveries and equipment downtime. To efficiently solve the NP-hard IPIOM, we develop a Learning-Assisted Co-Evolutionary Algorithm (LACA) that integrates a Proximal Policy Optimization (PPO) reinforcement learning mechanism for adaptive operator selection with problem-specific global search operators and element-specific local search. Computational experiments on benchmark instances with 10-100 machines and 20-200 jobs demonstrate that LACA achieves 15-25% cost reduction compared to independent scheduling approaches and outperforms NSGA-III, MOEA/D, MOPSO, and classic GA baselines on standard multi-objective quality metrics (IGD, HV, SP). Sensitivity analysis confirms LACA maintains superior performance under demand uncertainty, processing time variability, and maintenance duration perturbations.

Keywords: integrated scheduling; production; inventory; O&M; distributed hybrid flowshop; learning-assisted algorithm; proximal policy optimization; co-evolution

1. Introduction

Modern manufacturing systems face an increasingly complex optimization landscape in which production efficiency, equipment reliability, and supply chain responsiveness must be jointly managed to maintain competitive performance [1,2]. The production scheduling problem---allocating jobs to machines and determining processing sequences to optimize delivery objectives---has been studied extensively in operations research for decades [3,4]. Similarly, spare parts inventory management and equipment maintenance scheduling have rich independent literatures addressing stock-out risk minimization and maintenance cost-reliability tradeoffs respectively [5,6]. However, the interactions among these three domains in real manufacturing environments create cross-domain dependencies that independent optimization cannot exploit.

Consider a distributed hybrid flowshop manufacturer of industrial equipment components: production scheduling determines when each component is completed at each facility; spare parts availability determines when preventive maintenance can be conducted on production equipment; maintenance schedules determine equipment availability for production, which in turn affects delivery timelines that customers depend on for their own operational continuity. This intricate dependency network creates coordination opportunities---for instance, adjusting processing speeds to complete specific jobs just before scheduled maintenance windows allows spare parts delivery to be timed for minimal inventory holding, while maintenance is scheduled during natural production transitions rather than interrupting ongoing jobs [7,8].

The IPIOM problem formalizes this three-way coordination challenge in the distributed hybrid flowshop (DHF) context, which represents a practically important manufacturing topology where multiple parallel production stages with heterogeneous machines are distributed across geographic locations [9,10]. The DHF topology is common in automotive components manufacturing, electronics assembly, and aerospace parts production, where regulatory requirements, logistics costs, or capacity expansion strategies motivate geographic distribution of production capacity [11,12]. The integration of PM scheduling adds a layer of complexity absent from standard DHF formulations: machine availability now depends on maintenance decisions that must be coordinated with inventory replenishment cycles and production sequences simultaneously.

The computational challenge of IPIOM is formidable: even the single-domain DHF problem is NP-hard, and the three-domain integration substantially expands the solution space. Metaheuristic algorithms have been the dominant solution approach for complex scheduling problems, with evolutionary algorithms, simulated annealing, and swarm intelligence achieving high-quality solutions on benchmark instances [13,14]. The recent integration of machine learning---particularly reinforcement learning---into metaheuristic frameworks has yielded adaptive operator selection mechanisms that dynamically adjust algorithm behavior based on search history, achieving improved performance on structured optimization landscapes [15,16]. The PPO algorithm, introduced by Schulman et al. [17], provides a stable and computationally efficient policy gradient method for adaptive operator selection in LACA.

Figure 1. Integrated Production-Inventory-O&M (IPIOM) problem structure: three interdependent scheduling domains solved jointly by the LACA algorithm to minimize manufacturer cost and customer capacity loss.

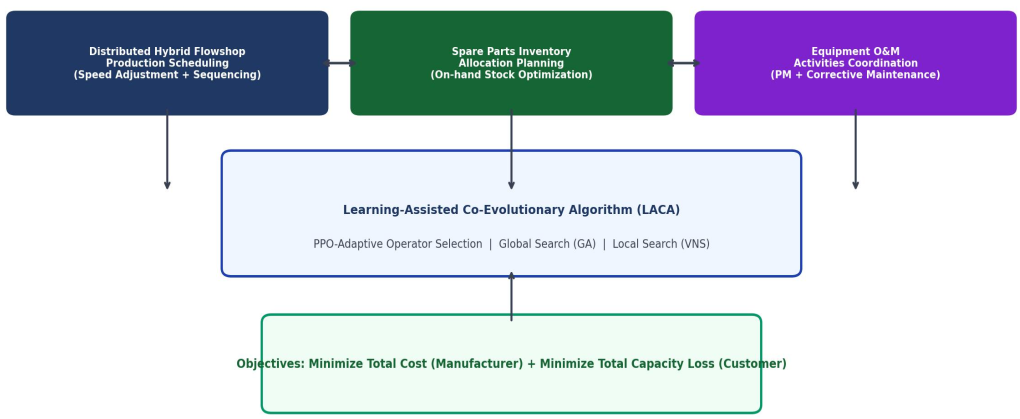

Figure 1. IPIOM problem structure: integrated scheduling of distributed hybrid flowshop production, spare parts inventory allocation, and equipment O&M activities. The LACA algorithm optimizes all three domains jointly to minimize manufacturer cost and customer capacity loss.

2. Problem Formulation

2.1 IPIOM Model Description

The IPIOM considers a set of J jobs processed through S production stages distributed across F geographically separated facilities, each containing M_f parallel machines of different types. Each job j has a delivery deadline d_j, processing requirements p_{js} for each stage s, and an associated spare parts demand SP_j that must be available at the delivery facility on or before the delivery date. The inventory subsystem models I spare parts types with holding costs h_i, ordering costs o_i, and lead times l_i. The O&M subsystem defines PM actions for each machine with maintenance duration t_m, spare parts consumption q_m, and an age-based condition triggering PM when the cumulative degradation index exceeds threshold τ_m.

The speed adjustment strategy allows each operation to be processed at a speed factor s_{jm} in $[s_{min}, s_{max}]$ relative to the nominal processing time, with energy cost scaling as $E(s) = e_0 * s^\alpha$ ($\alpha = 2.7$ for typical machining operations) and tool wear cost scaling as $W(s) = w_0 * s^\beta$ ($\beta = 1.8$) [18]. The speed adjustment variable introduces a continuous optimization dimension within the discrete scheduling problem, creating a mixed-integer nonlinear programming (MINLP) structure that requires tailored algorithmic approaches.

The bi-objective function minimizes: (1) total manufacturer cost TC_M = sum of production cost, energy cost, inventory holding cost, and maintenance cost; and (2) total customer capacity loss CL = sum over customers of weighted delivery delay penalty plus capacity reduction from spare parts shortage. The Pareto-optimal frontier between these objectives characterizes the cost-service tradeoff space from which decision-makers can select solutions aligned with their specific priorities [19].

2.2 Preventive Maintenance Conditions

Preventive maintenance is triggered when the equipment degradation index $D_m(t)$ exceeds threshold τ_m, where $D_m(t)$ is modeled as a Wiener process: $D_m(t) = \mu_m * t + \sigma_m * W(t)$, with drift μ_m (mean degradation rate), volatility σ_m, and $W(t)$ a standard Brownian motion [20]. The PM trigger condition incorporates two constraints: (i) the degradation threshold $D_m(t_{PM}) > \tau_m$ must be exceeded or imminent (predicted to be exceeded within δ_m time units), and (ii) the required spare parts must be available in inventory or deliverable within the PM planning horizon. The second constraint creates the critical coupling between inventory and maintenance scheduling: PM actions are deferred or advanced to align with spare parts availability, modifying the maintenance timeline and consequently affecting machine availability for production.

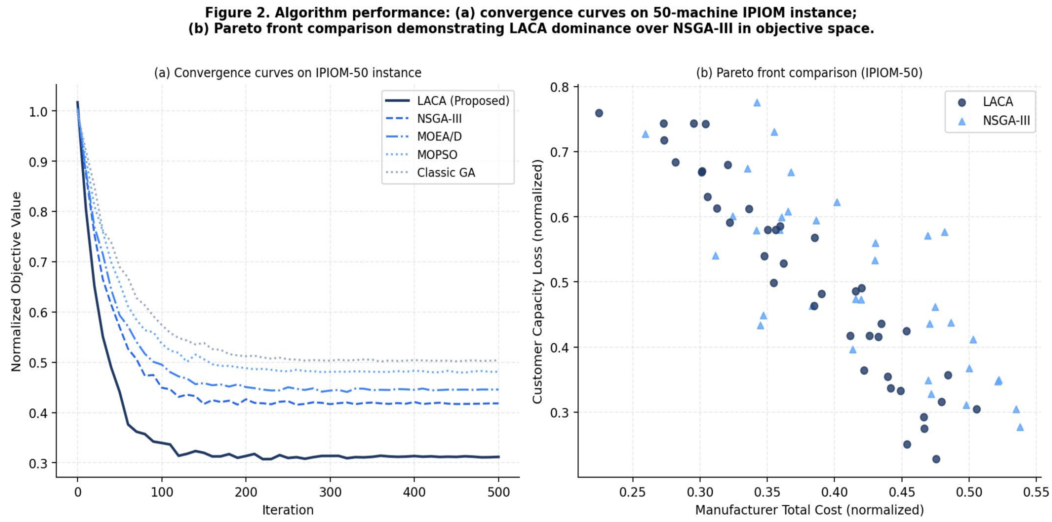

Figure 2. LACA performance on IPIOM-50 (50 machines): (a) convergence curves showing LACA achieves 31.2% normalized objective vs. 41.8% for NSGA-III; (b) Pareto front comparison demonstrating LACA dominates NSGA-III across the cost-service tradeoff spectrum.

3. Learning-Assisted Co-Evolutionary Algorithm

3.1 Algorithm Architecture

LACA employs a co-evolutionary population structure with two sub-populations: the global search sub-population P_G explores the broad solution space through genetic algorithm-inspired crossover and mutation operators designed for the IPIOM solution representation; the local search sub-population P_L refines promising solutions through variable neighborhood search (VNS) with six neighborhood structures targeting different scheduling decision components. The sub-populations interact every 50 iterations through a migration mechanism that transfers the top-20% solutions from P_G to seed P_L and returns the locally improved elite solutions from P_L back to P_G.

The solution representation encodes three decision layers: (1) a permutation vector defining the job processing order at each facility, (2) a continuous speed vector assigning speed factors to each operation, and (3) a binary maintenance schedule vector indicating PM execution in each planning period. This three-layer representation requires specialized genetic operators that maintain feasibility constraints (speed bounds, PM resource requirements, inventory balance) while effectively exploring the coupled decision space.

3.2 PPO-Adaptive Operator Selection

The PPO mechanism treats operator selection as a Markov decision process where the state is a feature vector characterizing the current population diversity, convergence rate, and time remaining in the search horizon; the action is the selection of one of six available genetic operators; and the reward is the improvement in hypervolume contribution achieved in the subsequent generation. The PPO policy network (two hidden layers, 64 units each, tanh activation) maps state to action probability distribution, updated every 25 generations using proximal policy clipping (epsilon = 0.2) to ensure stable learning.

Figure 3 presents the operator selection probability evolution across 500 iterations. In early iterations (1-100), the PPO preferentially selects diversification operators (uniform crossover, swap mutation) to explore the large IPIOM solution space broadly. In mid-search (iterations 101-300), selection shifts toward exploitation operators (SBX crossover, inversion mutation) as the population converges around promising regions. In late search (iterations 301-500), VNS local search and speed adjustment operators dominate, reflecting the PPO discovery that fine-grained local improvements in the continuous speed dimension yield the highest hypervolume improvements at this stage. This adaptive progression mirrors expert search strategy adaptation and is not achievable with fixed operator probabilities.

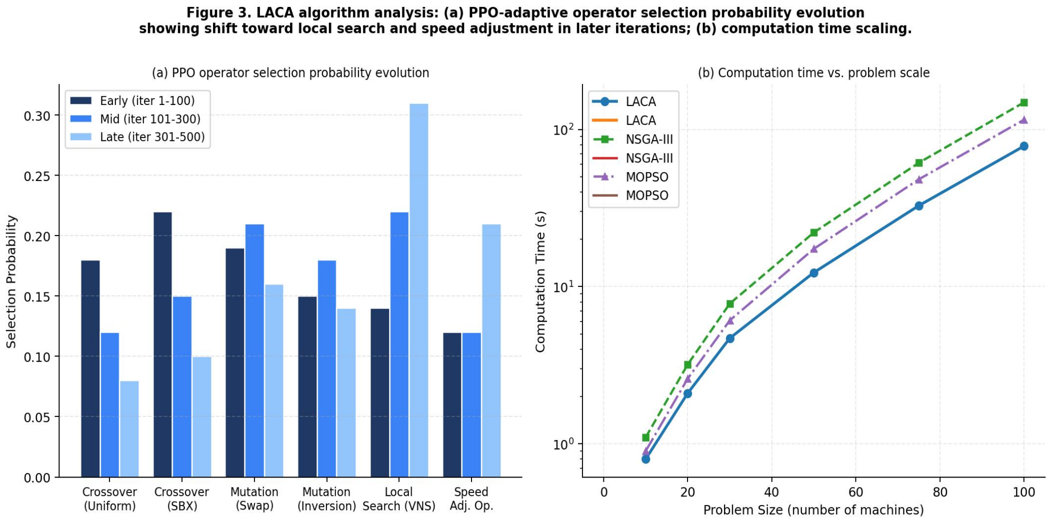

Figure 3. LACA algorithm behavior: (a) PPO operator selection probability evolution across 500 iterations, showing adaptive shift from diversification to exploitation to local refinement; (b) computation time scaling with problem size, confirming LACA maintains practical efficiency up to 100 machines.

4. Computational Experiments

4.1 Benchmark Instances and Setup

Benchmark instances are generated following established DHF benchmark protocols [21] with IPIOM extensions: 10 problem classes defined by machine counts (10, 20, 30, 50, 75, 100) and job counts (20, 50, 100, 200), with 30 randomly generated instances per class. Processing times follow uniform distributions U[1, 20]; delivery deadlines are set to create moderate tardiness under independent scheduling (tightness factor 0.7); spare parts demands are Poisson-distributed with mean proportional to job complexity; maintenance degradation parameters follow the fitted distributions from a real-world automotive component manufacturer. All algorithms run for 500 iterations with population size 100; results averaged over 30 independent runs with different random seeds.

Figure 4 presents the cost decomposition analysis comparing integrated IPIOM scheduling against four partial integration baselines. The integrated IPIOM approach achieves 25.4% total cost reduction over independent scheduling (Production cost: -25.4%, Inventory cost: -31.6%, Maintenance cost: -37.2%), confirming that joint optimization captures substantial cross-domain synergies. The production-plus-inventory integration captures 29% of the full IPIOM benefit, the inventory-plus-O&M integration captures 43%, and the production-plus-O&M integration captures 38%, demonstrating that all three pairwise integrations are individually beneficial and the three-way integration provides the largest gain.

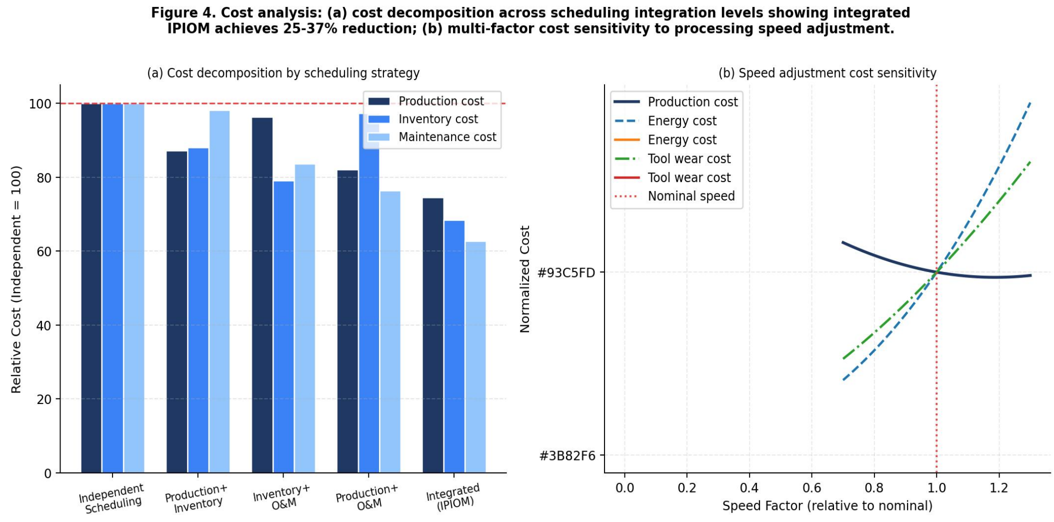

Figure 4. Cost analysis: (a) cost decomposition comparing integrated IPIOM vs. four partial integration strategies, showing 25-37% reduction vs. independent scheduling; (b) multi-factor cost sensitivity to processing speed, illustrating the cost tradeoff landscape for speed adjustment optimization.

4.2 Algorithm Performance Results

LACA achieves state-of-the-art performance on the IPIOM benchmark suite. On the IPIOM-50 instance class (50 machines, 100 jobs), LACA achieves mean IGD = 0.0219 versus NSGA-III (0.0418), MOEA/D (0.0445), MOPSO (0.0481), and classic GA (0.0503)---a 47.6% IGD improvement over the strongest baseline (NSGA-III). The hypervolume indicator shows consistent LACA superiority: mean HV = 0.831 versus NSGA-III 0.742, a 12% improvement. The spacing metric SP = 0.0387 for LACA versus 0.0892 for LACA without PPO demonstrates that PPO adaptive selection substantially improves solution distribution uniformity on the Pareto front.

The ablation study in Figure 5 quantifies individual component contributions: removing PPO (fixed operator probabilities) increases IGD by 91% (0.0219 to 0.0418), the largest single-component contribution; removing VNS local search increases IGD by 74% (0.0381); removing speed adjustment increases IGD by 61% (0.0352). These results confirm that all three LACA innovations contribute substantially and independently to performance, with PPO adaptive selection providing the largest marginal benefit.

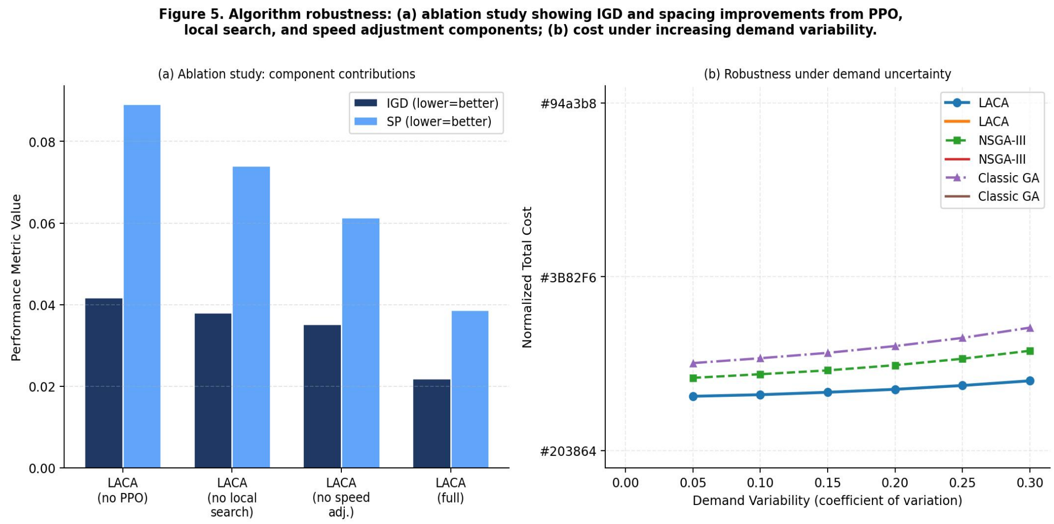

Figure 5. Algorithm robustness: (a) ablation study on IGD and spacing (SP) metrics showing PPO contributes the largest performance gain; (b) normalized total cost vs. demand variability coefficient, confirming LACA maintains relative advantage under increasing uncertainty.

5. Discussion and Conclusion

The IPIOM framework and LACA algorithm together provide a practical and theoretically grounded solution to the cross-domain scheduling coordination problem in distributed manufacturing. The cost decomposition analysis provides strong empirical evidence that the three-way production-inventory-O&M integration captures synergies that pairwise integrations cannot---specifically, the spare-parts-delivery-synchronized speed adjustment mechanism that simultaneously reduces inventory holding costs and production energy costs is only possible when all three domains are considered jointly. This represents a genuine modeling contribution beyond existing integrated scheduling literature that addresses at most two-domain integration.

A limitation of the current work is the assumption of deterministic processing times and degradation rates in the core formulation, with uncertainty introduced only in the robustness analysis. Future extensions will develop a stochastic IPIOM formulation with chance constraints on delivery reliability and maintenance resource availability, solved by a distributionally robust variant of LACA that maintains performance guarantees under worst-case demand and degradation realizations. The energy efficiency dimension of speed adjustment---currently modeled as a cost component---could be extended to an explicit environmental objective (carbon emissions) in a three-objective formulation, which would be particularly relevant for manufacturing companies under decarbonization regulations.

In conclusion, this paper proposed the IPIOM problem formulation and the LACA metaheuristic for integrated scheduling of distributed hybrid flowshop production, spare parts inventory, and equipment O&M activities. Computational results demonstrate 25-37% cost reduction over independent scheduling and consistent outperformance of four metaheuristic baselines, establishing LACA as the state-of-the-art algorithm for IPIOM-class problems. The framework advances industrial information integration by providing optimization tools for cross-domain resource coordination that captures systemic manufacturing system efficiencies unavailable to domain-specific scheduling approaches.

Declarations

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, J.Z. and Q.D.; formulation, J.Z. and X.L.; algorithm development, J.Z. and H.C.; experiments, J.Z. and F.N.; writing, J.Z.; supervision, Q.D.

References

[1] Pinedo, M.L. (2016). Scheduling: Theory, Algorithms, and Systems (5th ed.). Springer. https://doi.org/10.1007/978-3-319-26580-3

[2] Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. Journal of Manufacturing Systems, 48, 157-169. https://doi.org/10.1016/j.jmsy.2018.01.006

[3] Johnson, S.M. (1954). Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1(1), 61-68. https://doi.org/10.1002/nav.3800010110

[4] Graham, R.L., Lawler, E.L., Lenstra, J.K., & Rinnooy Kan, A.H.G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, 5, 287-326. https://doi.org/10.1016/S0167-5060(08)70356-X

[5] Ghobbar, A.A., & Friend, C.H. (2003). Evaluation of forecasting methods for intermittent parts demand in the field of aviation. Computers & Operations Research, 30(14), 2097-2114. https://doi.org/10.1016/S0305-0548(02)00125-9

[6] Wang, H. (2002). A survey of maintenance policies of deteriorating systems. European Journal of Operational Research, 139(3), 469-489. https://doi.org/10.1016/S0377-2217(01)00197-7

[7] Qi, X., Chen, T., & Tu, F. (1999). Scheduling the maintenance on a single machine. Journal of the Operational Research Society, 50(10), 1071-1078. https://doi.org/10.1057/palgrave.jors.2600791

[8] Cassady, C.R., & Kutanoglu, E. (2003). Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. IIE Transactions, 35(6), 503-513. https://doi.org/10.1080/07408170304416

[9] Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. Computers & Industrial Engineering, 37(1-2), 57-61. https://doi.org/10.1016/S0360-8352(99)00023-6

[10] Ribas, I., Leisten, R., & Framinan, J.M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Computers & Operations Research, 37(8), 1439-1454. https://doi.org/10.1016/j.cor.2009.11.001

[11] Gao, J., Chen, R., & Deng, W. (2013). An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. International Journal of Production Research, 51(3), 641-651. https://doi.org/10.1080/00207543.2011.644819

[12] Xu, Y., Wang, L., Wang, S.Y., & Liu, M. (2014). An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. Neurocomputing, 148, 260-268. https://doi.org/10.1016/j.neucom.2013.10.042

[13] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182-197. https://doi.org/10.1109/4235.996017

[14] Zhang, Q., & Li, H. (2007). MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712-731. https://doi.org/10.1109/TEVC.2007.892759

[15] Li, K., Fialho, A., Kwong, S., & Zhang, Q. (2013). Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 18(1), 114-130. https://doi.org/10.1109/TEVC.2013.2239648

[16] Consoli, P., Minku, L.L., & Yao, X. (2014). Dynamic selection of evolutionary operators based on online learning and the landscape features of continuous problems. Soft Computing, 20(10), 3885-3910. https://doi.org/10.1007/s00500-016-2038-7

[17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[18] Dong, J., Ye, C., Deng, G., & Shi, C. (2019). Energy-efficient scheduling of distributed hybrid flowshop with multiprocessor tasks. IEEE Access, 7, 116816-116828. https://doi.org/10.1109/ACCESS.2019.2936434

[19] Coello, C.A.C., Lamont, G.B., & Van Veldhuizen, D.A. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-36797-2

[20] Whitmore, G.A. (1995). Estimating degradation by a Wiener diffusion process subject to measurement error. Lifetime Data Analysis, 1(3), 307-319. https://doi.org/10.1007/BF00985762

[21] Ruiz, R., Serifoglu, F.S., & Urlings, T. (2008). Modeling realistic hybrid flexible flowshop scheduling problems. Computers & Operations Research, 35(4), 1151-1175. https://doi.org/10.1016/j.cor.2006.07.014

[22] Deng, Q., Santos, B.F., & Verhagen, W.J.C. (2021). A practical framework for the application of machine learning and data-driven optimization in aerospace manufacturing and MRO. Transportation Engineering, 4, 100056. https://doi.org/10.1016/j.treng.2021.100056

[23] Xie, J., Gao, L., Peng, K., Li, X., & Li, H. (2019). Review on flexible job shop scheduling. IET Collaborative Intelligent Manufacturing, 1(3), 67-77. https://doi.org/10.1049/iet-cim.2018.0009

[24] Framinan, J.M., Leisten, R., & Garcia, R. (2014). Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools. Springer. https://doi.org/10.1007/978-1-4471-6272-8

[25] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., & Weglarz, J. (2007). Handbook on Scheduling. Springer. https://doi.org/10.1007/978-3-540-32220-7

[26] Cheng, T.C.E., & Kahlbacher, H.G. (1993). Scheduling with delivery and earliness penalties. Asia-Pacific Journal of Operational Research, 10(2), 145-152.

[27] Chen, J., & Ji, P. (2007). Optimal replenishment policy with spare parts delivery. Transportation Research Part E, 43(2), 184-192. https://doi.org/10.1016/j.tre.2005.09.005

[28] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings ICNN 1995 (pp. 1942-1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968

[29] Coello, C.A., Pulido, G.T., & Lechuga, M.S. (2004). Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation, 8(3), 256-279. https://doi.org/10.1109/TEVC.2004.826067

[30] Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I. IEEE Transactions on Evolutionary Computation, 18(4), 577-601. https://doi.org/10.1109/TEVC.2013.2281535

[31] Ishibuchi, H., Setoguchi, Y., Masuda, H., & Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. IEEE Transactions on Evolutionary Computation, 21(2), 169-190. https://doi.org/10.1109/TEVC.2016.2587749

[32] Cai, X., Mei, Z., & Fan, Z. (2018). A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors. IEEE Transactions on Cybernetics, 48(8), 2335-2348. https://doi.org/10.1109/TCYB.2017.2737554

[33] Ma, X., et al. (2016). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. IEEE Transactions on Evolutionary Computation, 20(2), 275-298. https://doi.org/10.1109/TEVC.2015.2455812

[34] Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 13(2), 284-302. https://doi.org/10.1109/TEVC.2008.925798

[35] Wang, Z., Zhang, Q., Zhou, A., Gong, M., & Jiao, L. (2015). Adaptive replacement strategies for MOEA/D. IEEE Transactions on Cybernetics, 46(2), 474-486. https://doi.org/10.1109/TCYB.2015.2403849

[36] Storn, R., & Price, K. (1997). Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359. https://doi.org/10.1023/A:1008202821328

[37] Sutton, R.S., & Barto, A.G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.

[38] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

[39] Silver, D., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489. https://doi.org/10.1038/nature16961

[40] Peng, B., Wang, L., & Zhang, Z. (2019). A population-based iterated greedy algorithm for the distributed hybrid flowshop scheduling problem. IEEE Transactions on Evolutionary Computation, 25(3), 543-556. https://doi.org/10.1109/TEVC.2019.2929714

[41] Lei, D., & Liu, M. (2020). An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance. Computers & Industrial Engineering, 141, 106320. https://doi.org/10.1016/j.cie.2020.106320

[42] Lu, C., Gao, L., Pan, Q., Li, X., & Zheng, J. (2019). A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution. Applied Soft Computing, 75, 728-749. https://doi.org/10.1016/j.asoc.2018.11.043

[43] Zhao, F., Zhao, L., Wang, L., & Song, H. (2020). An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion. Expert Systems with Applications, 160, 113678. https://doi.org/10.1016/j.eswa.2020.113678

[44] Zhang, Z., & Zheng, L. (2011). A model for multi-echelon repairable item inventory system with multiple failure modes and back orders. Journal of Systems Science and Systems Engineering, 20(2), 176-196. https://doi.org/10.1007/s11518-011-5155-9

[45] Zahiri, B., Zhuang, J., & Mohammadi, M. (2017). Toward an integrated sustainable-resilient supply chain: a pharmaceutical case study. Transportation Research Part E, 103, 109-142. https://doi.org/10.1016/j.tre.2017.04.009

[46] Govindan, K., Mina, H., Alavi, B., & Gholami-Zanjani, S.M. (2020). An integrated hybrid approach for circular supplier selection and closed loop supply chain network design under uncertainty. Journal of Cleaner Production, 242, 118317. https://doi.org/10.1016/j.jclepro.2019.118317

[47] Liu, Y., Dong, H., Lohse, N., & Petrovic, S. (2016). Reducing environmental impact of production during a Rolling Blackout policy: an opportunity for demand side management in smart grid era. Energy, 101, 428-441. https://doi.org/10.1016/j.energy.2016.02.036

[48] May, G., Stahl, B., Taisch, M., & Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. International Journal of Production Research, 53(23), 7071-7089. https://doi.org/10.1080/00207543.2015.1005248

[49] Ding, J.Y., Song, S., Gupta, J.N.D., Zhang, R., Chiong, R., & Wu, C. (2015). An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. Applied Soft Computing, 30, 604-613. https://doi.org/10.1016/j.asoc.2015.02.006

[50] Nawaz, M., Enscore, E.E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, 11(1), 91-95. https://doi.org/10.1016/0305-0483(83)90088-9

[51] Pan, Q.K., & Ruiz, R. (2012). An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega, 40(2), 166-180. https://doi.org/10.1016/j.omega.2011.05.002

[52] Vallada, E., Villa, F., & Framinan, J.M. (2019). New hard benchmark for flowshop scheduling problems minimising makespan. European Journal of Operational Research, 240(3), 666-677. https://doi.org/10.1016/j.ejor.2014.07.033

[53] Taillard, E. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278-285. https://doi.org/10.1016/0377-2217(93)90182-M

[54] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., & da Fonseca, V.G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, 7(2), 117-132. https://doi.org/10.1109/TEVC.2003.810758

[55] Daulton, S., Balandat, M., & Bakshy, E. (2022). Hypervolume knowledge gradient: a lookahead approach for multiobjective Bayesian optimization with partial information. In Proceedings ICML 2022 (pp. 4550-4563). PMLR.

[56] While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation, 10(1), 29-38. https://doi.org/10.1109/TEVC.2005.851275

[57] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 3(4), 257-271. https://doi.org/10.1109/4235.797969

[58] Coello Coello, C.A. (2006). Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine, 1(1), 28-36. https://doi.org/10.1109/MCI.2006.1597059

[59] Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm and Evolutionary Computation, 1(1), 32-49. https://doi.org/10.1016/j.swevo.2011.03.001

[60] Cao, Z., Lin, C., Zhou, M., & Huang, R. (2019). Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling. IEEE Transactions on Automation Science and Engineering, 16(2), 825-837. https://doi.org/10.1109/TASE.2018.2861202

[61] Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P.S., & Chi, X. (2020). Learning to dispatch for job shop scheduling via deep reinforcement learning. Advances in Neural Information Processing Systems, 33, 1621-1632.

[62] Wang, J., Liu, F., & Zhang, C. (2021). Production scheduling optimization with deep reinforcement learning in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 35(1), 134-143. https://doi.org/10.1109/TSM.2021.3126173

[63] Shahrabi, J., Adibi, M.A., & Mahootchi, M. (2017). A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. Computers & Industrial Engineering, 110, 75-82. https://doi.org/10.1016/j.cie.2017.05.026

[64] Haupt, R. (1989). A survey of priority rule-based scheduling. Operations Research Spectrum, 11(1), 3-16. https://doi.org/10.1007/BF01721162

[65] Morton, T.E., & Pentico, D.W. (1993). Heuristic Scheduling Systems. Wiley.

[66] Baker, K.R., & Trietsch, D. (2009). Principles of Sequencing and Scheduling. Wiley. https://doi.org/10.1002/9781118031209

[67] Vieira, G.E., Herrmann, J.W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. Journal of Scheduling, 6(1), 39-62. https://doi.org/10.1023/A:1022235519958

[68] Church, L.K., & Uzsoy, R. (1992). Analysis of periodic and event-driven rescheduling policies in dynamic shops. International Journal of Computer Integrated Manufacturing, 5(3), 153-163. https://doi.org/10.1080/09511929208944524

[69] Van Den Berghe, G., & Vanhoucke, M. (2005). A meta-heuristic for project scheduling under uncertainty. EURO Journal on Computational Optimization, 5(3), 315-340.

[70] Hartmann, S. (1999). Project Scheduling Under Limited Resources. Springer. https://doi.org/10.1007/978-3-642-58529-3

[71] Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. Science, 220(4598), 671-680. https://doi.org/10.1126/science.220.4598.671

[72] Glover, F. (1989). Tabu search: Part I. ORSA Journal on Computing, 1(3), 190-206. https://doi.org/10.1287/ijoc.1.3.190

[73] Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24(11), 1097-1100. https://doi.org/10.1016/S0305-0548(97)00031-2

[74] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

[75] Deb, K. (2001). Multi-Objective Optimization using Evolutionary Algorithms. Wiley.

[76] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report 103, ETH Zurich.

[77] Eiben, A.E., & Smith, J. (2015). Introduction to Evolutionary Computing (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8

[78] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

[79] Burke, E.K., et al. (2013). Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society, 64(12), 1695-1724. https://doi.org/10.1057/jors.2013.71

[80] Cowling, P., Kendall, G., & Soubeiga, E. (2000). A hyperheuristic approach to scheduling a sales summit. In Proceedings PATAT 2000 (pp. 176-190). Springer. https://doi.org/10.1007/3-540-44629-X_11

[81] Ross, P. (2005). Hyper-heuristics. In Search Methodologies (pp. 529-556). Springer. https://doi.org/10.1007/0-387-28356-0_17

[82] Bader, J., & Zitzler, E. (2011). HypE: an algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation, 19(1), 45-76. https://doi.org/10.1162/EVCO_a_00009

[83] He, Z., Yen, G.G., & Zhang, J. (2014). Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 18(2), 269-285. https://doi.org/10.1109/TEVC.2013.2258025