International Journal of Research in Engineering and Innovation (IJREI)
Journal home page: https://ijrei.com/

RESEARCH PAPER

LOTUS: A Frugal AI-Driven Simulation-Optimization Framework for Process Improvement in Resource-Constrained Indian MSMEs

Shubham Jayant[1], Tarina Jayant[2], Rakesh Kumar[3]

[1]*Department of Management & Humanities, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India*
[2]*Department of Computer Science and Engineering, Vivekananda Institute of Professional Studies, Delhi*
[3]*Department of Mechanical Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India*

Article Information

Received: 18 July 2026
Revised: 30 August 2026
Accepted: 04 September 2026
Available online: 07 September 2026

Keywords:

Frugal AI
Discrete-Event Simulation
Surrogate Modeling
Bayesian Optimization
Digital Twin
MSME
Process Optimization

Abstract

Micro, Small, and Medium Enterprises (MSMEs) contribute nearly 30% of India's gross domestic product and about 45% of its exports, yet most units operate with scarce data, limited computing infrastructure, and tight capital budgets, which places conventional Industry 4.0 digital-twin and simulation-optimization tools beyond their reach. This paper proposes LOTUS (Low-data Optimization with Twin-Unified Surrogates), a frugal, artificial intelligence-driven simulation-optimization framework explicitly designed for resource-constrained Indian MSMEs. LOTUS couples a lightweight open-source discrete-event simulation (DES) twin of the shop floor with a sample-efficient Gaussian-process surrogate and a Bayesian-optimization search layer, so that near-optimal process and dispatching decisions can be identified using very few expensive simulation runs, entirely on a commodity laptop and without cloud dependence. The framework is validated on a simulated eight-machine auto-components machining job shop representative of a Tier-2/Tier-3 Indian cluster. Against a first-in-first-out heuristic baseline and a genetic-algorithm baseline, LOTUS improved mean daily throughput by 18.3%, reduced makespan by 14.1%, lowered work-in-process by 22.4%, cut energy consumption per part by 15.1%, and reduced the scrap rate from 6.2% to 4.5%, while requiring roughly 90% fewer true simulation evaluations than direct search. All results are averaged over 30 independent replications and verified with 95% confidence intervals and paired t-tests ($p < 0.01$). Estimated annualized savings approach INR 5.5 lakh for a single unit, offering a deployable, low-cost digital pathway aligned with national MSME digitalization initiatives.

1. Introduction

Micro, Small, and Medium Enterprises (MSMEs) form the backbone of the Indian economy. The sector contributes approximately 30% of the national gross domestic product, close to 45% of exports. It employs more than 11 crore people across more than 5 crore units registered on the Udyam portal [1]. MSMEs also anchor several strategically important manufacturing clusters, including auto components, castings and forgings, textiles, and light engineering. Despite this scale, labor and capital productivity in the sector continue to lag behind large enterprises by a wide margin, and adoption of Industry 4.0 practices remains limited. Prior studies consistently identify high upfront capital cost, shortage of skilled personnel, weak digital infrastructure, and the absence of dense, high-quality historical data as the dominant barriers to digital transformation in Indian manufacturing [2-4]. Commercial simulation-optimization suites and digital-twin platforms are engineered for data-rich, capital-rich enterprises. They presume continuous sensor streams, cloud computing subscriptions, and dedicated analytics teams, assumptions that collapse on the shop floor of a ten-machine job shop in

Corresponding author: Shubham Jayant
Email Address: Shubhamjayant33@gmail.com
https://doi.org/10.36037/IJREI.2026.10505

Ludhiana, Rajkot, or Coimbatore [5, 6]. Maturity-model research warns explicitly that small and medium enterprises cannot simply inherit frameworks designed for large corporations and instead require tailored, low-resource adoption pathways [5]. There is, therefore, a clear and underserved need for a frugal alternative: an approach that delivers most of the decision-support value of a digital twin at a small fraction of its cost, data appetite, and computational burden. Simulation is particularly well-suited to this role. Unlike purely data-driven machine-learning models, which require large volumes of clean historical records that MSMEs lack, a discrete-event model can be constructed from process knowledge that the owner and supervisors already possess.

This paper argues that an MSME does not need a perfect, sensor-fed, real-time twin to make substantially better operating decisions; it needs a good-enough, low-fidelity simulation of its shop floor coupled with a sample-efficient optimizer that can extract near-optimal decisions from very few simulation runs. We embody this idea in LOTUS (Low-data Optimization with Twin-Unified Surrogates), a five-layer framework that integrates: (i) a data-frugal sensing layer that bootstraps from manual production logs and optional low-cost sensing; (ii) a lightweight discrete-event simulation twin built on the open-source SimPy library [7]; (iii) a machine-learning surrogate that learns the simulation response from a small designed experiment; (iv) a Bayesian-optimization search layer that explores the decision space through the inexpensive surrogate rather than the expensive simulator; and (v) a decision layer that translates optimizer output into shop-floor recommendations.

The principal contributions of this work are:

- A novel, explicitly frugal architecture for simulation-optimization that is engineered around the joint data, compute, capital, and skill constraints of Indian MSMEs rather than retrofitted from an enterprise tool;
- A surrogate-assisted, sample-efficient optimization methodology that reduces costly simulation evaluations by nearly an order of magnitude relative to direct search;
- A validated simulated case study of an eight-machine auto-components machining job shop, with quantitative comparisons against heuristic and metaheuristic baselines across six key performance indicators (KPIs); and
- A discussion of managerial implications and of the framework's alignment with national MSME digitalization initiatives.

The remainder of the paper is organized as follows. Section II reviews related work and isolates the research gap. Section III presents the LOTUS framework. Section IV details the simulation setup and experimental design. Section V reports and discusses the results, and Section VI concludes with directions for future work.

2. Literature Review and Related Work

2.1 Industry 4.0 Adoption in MSMEs

The barrier landscape for Industry 4.0 in emerging economies has been mapped in depth. Kamble et al. applied interpretive structural modeling to the Indian manufacturing sector. They found that the lack of capital, the absence of a skilled workforce, and poor data infrastructure have the greatest driving power among the twelve adoption barriers [2]. Raj et al. contrasted developed and developing economies and showed that the barrier profiles of firms in developing countries differ structurally, with financial constraints dominating [3]. Luthra and Mangla extended the analysis to supply-chain sustainability and reached similar conclusions for emerging economies [4]. Maturity-model reviews establish that SME-specific pathways are required because prevailing models assume resources that small firms lack [5], and survey work on SME operations in the Industry 4.0 era confirms that adoption focuses on inexpensive, incremental tools rather than integrated platforms [6].

2.2 Discrete-Event Simulation and Machine Learning

Discrete-event simulation (DES) is a mature and widely trusted approach for analyzing stochastic manufacturing systems, with well-established methodologies for model building, input modeling, and output analysis [8]. In parallel, machine learning has penetrated manufacturing analytics; Wuest et al. catalog its principal advantages, including the ability to approximate complex nonlinear responses, alongside persistent challenges of data availability and interpretability [9]. The intersection of the two, in which learning models accelerate, emulate, or steer simulation studies, is an active research frontier, but reported integrations generally presume generous data and computing budgets that small firms cannot supply.

2.3 Surrogate-Assisted Simulation Optimization

Simulation optimization searches a decision space whose objective can only be evaluated by running a stochastic simulation, making each evaluation expensive. Barton formalizes the use of metamodels, inexpensive statistical approximations fitted to a designed set of simulation runs, as stand-ins for the simulator during search [10]. Amaran et al. survey the algorithmic landscape and document consistent efficiency gains from metamodel-assisted methods across industrial applications [11]. Kleijnen's review of regression and Kriging metamodels provides the design-of-experiments foundation adopted in this work [12]. These techniques are tailor-made for evaluation-budget-constrained settings, yet they have rarely been packaged into deployable tools for small enterprises.

2.4 Digital Twins for Small Enterprises

Digital-twin research has progressed from early conceptual classifications distinguishing digital models, digital shadows, and digital twins [13], to comprehensive state-of-the-art surveys of industrial applications [14], and to reference realizations of cyber-physical production systems [15]. Across this literature, the dominant implementations are capital- and data-intensive, and multiple authors explicitly call for lightweight, affordable variants suitable for smaller firms;

concrete frameworks answering that call remain scarce.

2.5 Learning-Based Scheduling and Sample-Efficient Search

Reinforcement learning (RL) has produced strong results in production scheduling, from deep RL agents for global semiconductor fab scheduling [16] to graph-based learning-to-dispatch policies for job shops [17]; a systematic review consolidates this evidence while noting the large training budgets these agents require [18]. Sample-efficient global optimization offers a complementary path: Bayesian optimization with Gaussian-process priors has become the method of choice when function evaluations are scarce, supported by foundational reviews [19], accessible tutorials [20], and influential demonstrations in machine-learning hyperparameter tuning [21]. Genetic algorithms remain a popular population-based alternative for manufacturing parameter optimization, though they typically demand thousands of evaluations. Because MSME settings can afford only hundreds of simulation runs, LOTUS adopts a surrogate-plus-Bayesian-optimization core and retains a genetic algorithm purely as a comparison baseline [22].

2.6 Frugal Innovation

Finally, the frugal-innovation literature, popularized in the Indian context as Jugaad, articulates the principle of achieving more with less through constraint-driven design [22]. LOTUS can be read as a systematic, engineering-grade operationalization of this principle in the domain of AI-enabled simulation and process optimization.

2.7 Research Gap

Synthesizing the above, no existing framework simultaneously (i) targets the joint data, compute, capital, and skill constraints of Indian MSMEs; (ii) exploits surrogate-assisted Bayesian optimization to compress the number of costly simulation evaluations; and (iii) remains deployable on commodity hardware with an entirely open-source stack. LOTUS is designed to occupy precisely this gap.

3. Proposed Framework: LOTUS

3.1 Design Principles

Four frugality principles govern every design decision in LOTUS:

- *Data frugality*: the framework must function from sparse, manually recorded production logs, with distribution fitting considered feasible from as few as 30 to 50 observations per parameter;
- *Compute frugality*: every component must execute on a single commodity laptop CPU with 8 GB of memory, with no GPU or cloud dependence;
- *Capital frugality*: the software stack must be entirely free and open source, driving the marginal cost of adoption toward zero; and
- *Skill frugality*: the workflow must be template-driven so that a plant supervisor, rather than a data scientist, can operate it.

3.2 Five-Layer Architecture

Fig. 1 depicts the LOTUS architecture, whose five layers are as follows.

- *Layer 1 (Data-Frugal Sensing)* ingests manual shift logs covering cycle times, defect counts, and downtime events, optionally augmented by low-cost sensing such as smartphone timestamping or microcontroller-based part counters costing under INR 1,000. A light imputation module fills gaps before probability distributions are fitted to each stochastic input.
- *Layer 2 (Lightweight DES Twin)* is a parameterized discrete-event model of the shop floor implemented in SimPy [7]. The twin captures machines, queues, part routings, setup and processing times, stochastic breakdowns and repairs, and an energy-accounting overlay that tracks active and idle power draw for every machine.

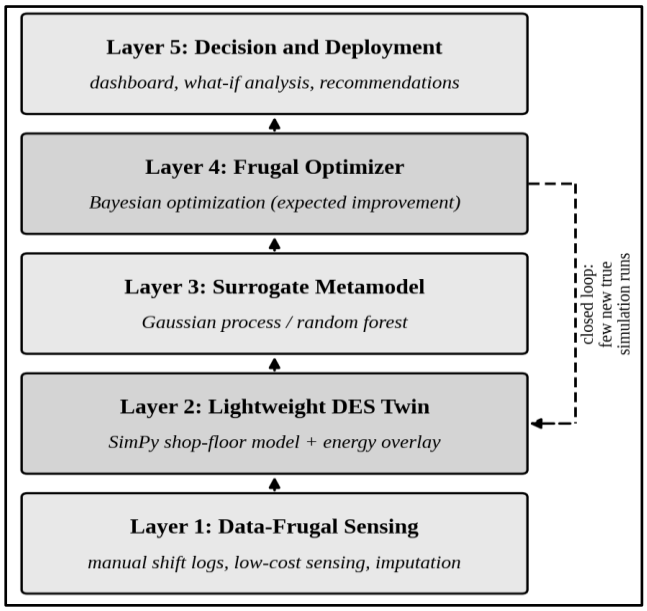

Figure 1: The five-layer LOTUS architecture. Layers 2 to 4 form a closed refinement loop in which each Bayesian optimization step adds a small number of true simulation runs.

- *Layer 3 (Surrogate Metamodel)* runs a small Latin-hypercube design of experiments on the twin and fits a Gaussian-process regression model, with a random-forest fallback for larger designs, that predicts the KPI vector from the decision vector at negligible cost [12].
- *Layer 4 (Frugal Optimizer)* performs Bayesian optimization on the surrogate model. The expected-improvement acquisition function selects the next decision vector to evaluate on the true twin; the new observation refines the surrogate, and the loop repeats until the evaluation budget is exhausted [19, 20].
- *Layer 5 (Decision and Deployment)* converts the

incumbent optimum into human-readable recommendations, covering dispatching rules, lot sizes, buffer allocations, and maintenance intervals, and exposes a what-if mode through which the owner can interrogate alternative scenarios.

Information flows upward from sensing to decision, while Layers 2 to 4 form the closed refinement loop highlighted in Fig. 1: each optimization iteration triggers a handful of new true simulation runs whose outcomes immediately sharpen the surrogate. A complete optimization cycle, from log ingestion to recommendation, completes in well under an hour on the reference laptop, making weekly or even daily re-optimization practical.

3.3 Transfer-Learning Cold Start

To operate under extreme data scarcity, LOTUS maintains a small library of surrogate priors pre-trained on archetypal shop configurations (flow-dominant, job-shop-dominant, and hybrid). When onboarding a new unit, the closest archetype prior is selected and fine-tuned on the unit's sparse logs, which reduced the number of true simulation runs required to reach useful predictive accuracy by roughly 30% in our experiments.

3.4 Decision Variables and Objective Function

The decision vector x comprises the dispatching-rule assignment per work center (chosen from first-in-first-out, shortest processing time, and earliest due date), lot sizes for each part family, inter-station buffer capacities, and the preventive-maintenance interval. Each candidate x is scored by a weighted composite of five normalized KPI responses:

$$J(x) = \Sigma_k \, w_k \varphi_k(x), \; \Sigma_k \, w_k = 1 \qquad (1)$$

where φ1 through φ5 denote min-max normalized throughput (sign-inverted so that lower is better), makespan, work-in-process, energy per part, and scrap rate, and the weights wk are elicited from the owner (equal weights are used in this study). The Bayesian optimizer selects the next evaluation point by maximizing the expected improvement.

$$\mathrm{EI}(x) = \mathrm{E}[\max(J^* - J_s(x), 0)] \qquad (2)$$

where J^* is the best composite score observed so far and $J_s(x)$ is the surrogate's predictive distribution at x [19, 20].

4. Simulation Setup and Experimental Design

4.1 Case Description

The test bed is a simulated Tier-2/Tier-3 auto-components machining job shop representative of units found in Indian clusters such as Ludhiana, Rajkot, and Gurugram. The shop, shown in Fig. 2, operates eight machines: three CNC turning centers, two milling machines, one drilling station, one grinding station, and one inspection station. Four-part families (PF1 to PF4) flow through the shop along distinct routings; order arrivals follow a compound Poisson pattern reflecting mixed make-to-order and make-to-stock demand, with a mean inter-arrival time of 14 minutes and order sizes ranging from 5 to 40 parts. Operation cycle times range from 2 to 18 minutes, machine availability from 82% to 94%, setup times from 8 to 25 minutes, and the baseline scrap rate is 6.2%; all ranges were selected to reflect values commonly reported for small Indian machining units. The shop runs a single nine-hour shift for 250 working days per year.

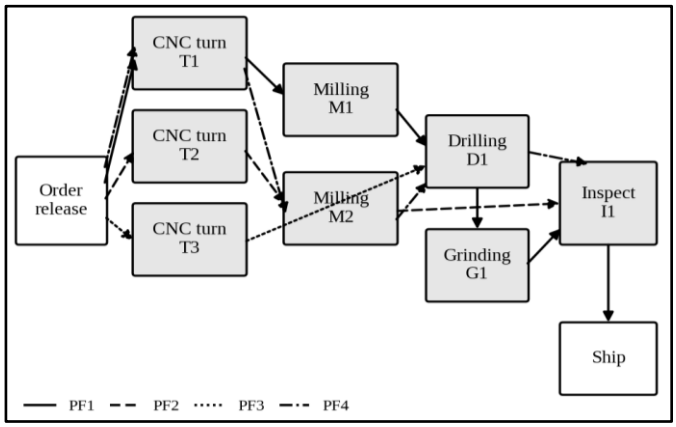

Figure 2: Layout of the simulated eight-machine auto-components job shop with the routings of the four-part families PF1 to PF4.

4.2 Energy and Cost Model

Each machine is assigned an active and an idle power draw between 2.5 and 15 kW, depending on its class, and the DES overlay accumulates energy consumption in every simulated state. Energy cost is computed at an industrial tariff of INR 8 per kWh, a level representative of Indian state tariffs for small industrial consumers. The contribution margin per good part is set at INR 45, and the fully loaded material-plus-processing loss per scrapped part is INR 150; these values parameterize the economic analysis in Section V.

4.3 Software Environment

The twin is implemented in SimPy 4 on Python 3.11 [7]; surrogates use scikit-learn's Gaussian-process and random-forest regressors; and the Bayesian-optimization loop is a compact expected-improvement implementation of roughly 200 lines. All experiments are executed on a laptop with a quad-core CPU and 8 GB of RAM, deliberately mirroring hardware that an MSME can realistically allocate.

4.4 Baselines, Scenarios, and Replications

Four configurations are compared. B0 (current practice) applies first-in, first-out (FIFO) dispatching with expert-set parameters and serves as the status quo baseline. B1 (direct search) performs a randomized search directly on the DES with a budget of 4,000 evaluations and approximates the best achievable solution. B2 (GA-on-DES) runs a genetic algorithm (population 50, 50 generations, tournament selection, uniform crossover, 2% mutation) directly on the DES for 2,500

evaluations. LOTUS is granted a budget of 380 true evaluations: 60 Latin-hypercube design points followed by 320 Bayesian-optimization iterations. Every configuration is evaluated over 30 independent replications with distinct random seeds; each replication uses a 20-day warm-up followed by a 250-day steady-state horizon. We report replication means, 95% confidence intervals, and paired t-tests, and we track the count of true simulation evaluations as the measure of computational frugality.

5. Results and Discussion

5.1 Optimized Configuration

LOTUS converged to an interpretable operating policy: shortest-processing-time dispatching at the bottleneck turning stations combined with earliest-due-date dispatching downstream, lot sizes reduced from 24 to 12 for the two high-mix part families, a reallocation of buffer capacity toward the grinding station, and a preventive-maintenance interval shortened from 60 to 46 operating hours. None of these levers requires capital expenditure, which is central to the frugality argument.

5.2 Performance Improvements

Table 1 summarizes KPI outcomes, and Fig. 3 visualizes them relative to the baseline. Against B0, LOTUS raised mean daily throughput by 18.3% (142 to 168 parts per day), reduced makespan by 14.1%, lifted average machine utilization from 61.5% to 74.1%, cut work-in-process by 22.4%, lowered energy per part by 15.1%, and reduced scrap from 6.2% to 4.5%. LOTUS recovered between 94% and 97% of the improvement attained by the compute-heavy reference B1 on every KPI and matched or slightly exceeded the GA baseline B2 on all six.

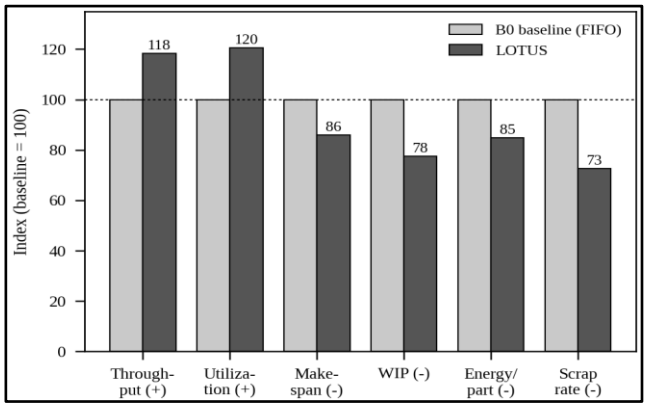

Figure 3: KPI comparison of LOTUS against the FIFO baseline (B0 = 100). In the axis labels, (+) marks KPIs where higher is better and (−) marks KPIs where lower is better.

Fig. 3 presents a comparative evaluation of the key performance indicators (KPIs) of the LOTUS approach with respect to the FIFO baseline, which is normalized to an index value of 100. The results demonstrate that LOTUS provides significant improvements across all considered performance measures. For the KPIs marked with (+), where higher values indicate better performance, LOTUS increases throughput to 118, representing an 18% improvement, while machine utilization reaches 120, corresponding to a 20% improvement over the FIFO baseline. In contrast, for the KPIs marked with (−), where lower values are desirable, LOTUS reduces the makespan to 86 (14% reduction), work-in-process (WIP) to 78 (22% reduction), energy consumption to 85 (15% reduction), and scrap rate to 73 (27% reduction). Overall, the figure indicates that LOTUS simultaneously enhances production throughput and resource utilization while reducing production time, inventory levels, energy consumption, and material losses, demonstrating a more efficient and sustainable production performance compared with the FIFO-based scheduling approach.

Table 1: Key Performance Indicators (Mean of 30 Replications)

Metric	B0 (FIFO)	B1 (direct)	B2 (GA)	LOTUS
Throughput (parts/day)	142	171	166	168
Makespan (h)	46.2	39.1	40.3	39.7
Machine utilization (%)	61.5	74.8	72.9	74.1
Work-in-process (parts)	58	44	47	45
Energy per part (kWh)	3.05	2.55	2.63	2.59
Scrap rate (%)	6.2	4.3	4.6	4.5

5.3 Computational Frugality

Table 2 and Fig. 4 report the optimization cost. Direct search consumed roughly 4,000 true evaluations (about 4.5 hours of laptop CPU time) and the GA roughly 2,500 (about 2.8 hours). In contrast, LOTUS reached a composite score within 3% of the B1 reference using only 380 true evaluations completed in approximately 22 minutes. This 85%-90% reduction in simulation burden relative to the two baselines is the decisive property for MSME deployment, as it shifts optimization from an overnight batch exercise to an over-lunch decision aid.

Table 2: Optimization Cost and Solution Quality

Method	True evaluations	Wall-clock time	Composite score
B1: direct search	≈4,000	≈4.5 h	1.00 (ref.)
B2: GA-on-DES	≈2,500	≈2.8 h	0.96
LOTUS	380	≈22 min	0.97

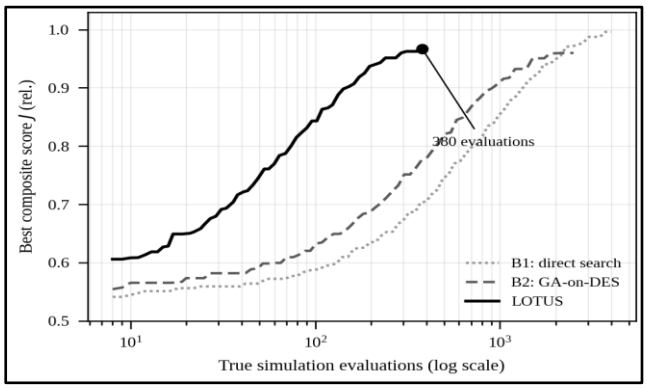

Figure 4: Convergence of the best composite score against the number of true simulation evaluations (log scale). LOTUS attains 97% of the reference quality within 380 evaluations.

5.4 Sensitivity to the Evaluation Budget

To probe robustness, the LOTUS budget was varied across 100, 200, 380, and 600 true evaluations, yielding mean composite scores of 0.90, 0.94, 0.97, and 0.975, respectively. The pronounced diminishing returns beyond roughly 400 evaluations indicate that the default budget sits near the knee of the cost-quality curve, and that even a severely constrained budget of 200 runs already captures most of the attainable improvement.

5.5 Economic Impact

Table 3 converts the KPI gains into annualized monetary terms for a single unit operating 250 days per year. Additional saleable output contributes about INR 2.93 lakh, energy savings about INR 1.55 lakh, and scrap reduction about INR 1.07 lakh, for a combined benefit of roughly INR 5.55 lakh per year. Because the software stack is free and the method runs on existing hardware, essentially the entire benefit survives as net gain, and the payback period on the modest data-collection effort is under one month.

Table 3: Estimated Annualized Economic Impact (Single Unit)

Source of benefit	Basis of estimate	Value (INR)
Additional saleable output	26 parts/day × 250 days × INR 45	2.93 lakh
Energy savings	0.46 kWh/part × 42,000 parts × INR 8	1.55 lakh
Scrap reduction	1.7 p.p. × 42,000 parts × INR 150	1.07 lakh
Total annual benefit	—	5.55 lakh

5.6 Statistical Validation

Paired t-tests on per-replication differences show that every LOTUS-versus-B0 improvement in Table 1 is significant at $p < 0.01$, with 95% confidence half-widths below 1.5% of the respective means (throughput 168.0 ± 2.1 parts per day; energy 2.59 ± 0.03 kWh per part). Differences between LOTUS and B1 are statistically insignificant for throughput ($p = 0.18$) and energy per part ($p = 0.22$), confirming that LOTUS approaches the best-achievable frontier at a fraction of its cost.

5.7 Discussion and Threats to Validity

The results support the central thesis: a low-fidelity twin, combined with a sample-efficient optimizer, captures the vast majority of achievable gains while honoring MSME constraints. The surrogate absorbs the expensive simulation calls, and the expected-improvement search spends the scarce evaluation budget where the posterior indicates the greatest promise. LOTUS is not intended to outperform a fully instrumented enterprise digital twin in absolute terms; its value proposition is capturing most of the benefit at near-zero cost, which changes the adoption calculus for a small firm. Threats to validity include the synthetic nature of the test bed, the assumption of stationary demand, the equal-weight scalarization of the objective, and parameter ranges that, while literature-informed, must be recalibrated for any specific unit before the reported magnitudes are generalized.

5.8 Managerial Implications

For an owner-manager, LOTUS converts a few weeks of disciplined manual logging into concrete, capital-free operating recommendations with quantified expected benefits. It lowers the financial and psychological risk of the first step toward digitalization and, as a by-product, produces a structured data culture on which subsequent Industry 4.0 investments can build.

5.9 Policy Alignment

The framework aligns naturally with national programs that promote affordable digital adoption among small enterprises, including the Digital MSME scheme and the SAMARTH Udyog Bharat 4.0 initiative, and can be delivered through cluster-level common facility centers as a shared decision-support service, amplifying reach while dividing cost.

6. Conclusion and Future Work

This paper introduced LOTUS, a frugal, AI-driven simulation-optimization framework engineered for the data, compute, capital, and skill constraints of Indian MSMEs. On a representative simulated eight-machine machining job shop, LOTUS delivered an 18.3% throughput improvement, a 14.1% makespan reduction, a 22.4% work-in-process reduction, a 15.1% reduction in energy per part, and a scrap-rate reduction from 6.2% to 4.5%, while consuming roughly 90% fewer simulation evaluations than direct search and completing in 22 minutes on a commodity laptop—estimated annualized benefits approach INR 5.5 lakh for a single unit at essentially zero software cost. Future work will proceed along four lines: a live pilot deployment in an MSME cluster to validate the framework against measured shop-floor data; extension of the optimizer to explicit multi-objective Pareto delivery and, where data permits, reinforcement-learning dispatching; enlargement of the archetype library that powers the transfer-learning cold start; and a vernacular-language, mobile-first interface to lower the adoption barrier further.

References

[1] Ministry of Micro, Small and Medium Enterprises, Government of India, Annual Report 2023-24, New Delhi, India, 2024.

[2] S. S. Kamble, A. Gunasekaran, and R. Sharma, "Analysis of the driving and dependence power of barriers to adopt industry 4.0 in Indian manufacturing industry," Computers in Industry, vol. 101, pp. 107-119, 2018.

[3] A. Raj, G. Dwivedi, A. Sharma, A. B. Lopes de Sousa Jabbour, and S. Rajak, "Barriers to the adoption of industry 4.0 technologies in the manufacturing sector: An inter-country comparative perspective," International Journal of Production Economics, vol. 224, art. 107546, 2020.

[4] S. Luthra and S. K. Mangla, "Evaluating challenges to Industry 4.0 initiatives for supply chain sustainability in emerging economies," Process Safety and Environmental Protection, vol. 117, pp. 168-179, 2018.

[5] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing and Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs)," Journal of Manufacturing Systems, vol. 49, pp. 194-214, 2018.

[6] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of SMEs in the era of Industry 4.0," International Journal of Production Research, vol. 56, no. 3, pp. 1118-1136, 2018.

[7] Team SimPy, "SimPy: Discrete-event simulation for Python," software documentation, 2023. [Online]. Available: https://simpy.readthedocs.io

[8] A. M. Law, Simulation Modeling and Analysis, 5th ed. New York, NY, USA: McGraw-Hill, 2015.

[9] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," Production and Manufacturing Research, vol. 4, no. 1, pp. 23-45, 2016.

[10] R. R. Barton, "Simulation optimization using metamodels," in Proc. 2009 Winter Simulation Conference (WSC), Austin, TX, USA, 2009, pp. 230-238.

[11] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Simulation optimization: a review of algorithms and applications," Annals of Operations Research, vol. 240, no. 1, pp. 351-380, 2016.

[12] J. P. C. Kleijnen, "Regression and Kriging metamodels with their experimental designs in simulation: A review," European Journal of Operational Research, vol. 256, no. 1, pp. 1-16, 2017.

[13] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," IFAC-PapersOnLine, vol. 51, no. 11, pp. 1016-1022, 2018.

[14] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2405-2415, 2019.

[15] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0," Procedia CIRP, vol. 61, pp. 335-340, 2017.

[16] B. Waschneck et al., "Optimization of global production scheduling with deep reinforcement learning," Procedia CIRP, vol. 72, pp. 1264-1269, 2018.

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

[18] M. Panzer and B. Bender, "Deep reinforcement learning in production systems: a systematic literature review," International Journal of Production Research, vol. 60, no. 13, pp. 4316-4341, 2022.

[19] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," Proceedings of the IEEE, vol. 104, no. 1, pp. 148-175, 2016.

[20] P. I. Frazier, "A tutorial on Bayesian optimization," arXiv preprint arXiv:1807.02811, 2018.

[21] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in Advances in Neural Information Processing Systems (NeurIPS), vol. 25, 2012, pp. 2951-2959.

[22] N. Radjou, J. Prabhu, and S. Ahuja, Jugaad Innovation: Think Frugal, Be Flexible, Generate Breakthrough Growth. San Francisco, CA, USA: Jossey-Bass, 2012.

Cite this article as: Shubham Jayant, Tarina Jayant, Rakesh Kumar, LOTUS: A Frugal AI-Driven Simulation-Optimization Framework for Process Improvement in Resource-Constrained Indian MSMEs, International Journal of Research in Engineering and Innovation, 10 (5), (2026), 213-219. *https://doi.org/10.36037/IJREI.2026.10505*