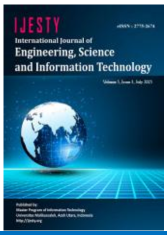
International Journal of Engineering, Science and Information Technology
Volume 5, No. 3 (2025) pp. 515-522
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.1296
Research Paper, Short Communication, Review, Technical Paper



A Novel Hybrid Method for DAP: Differential Evolution with Variable Neighborhood Search

Mamta Thakur[1*], Talluri Sushma[2], Nagaraju Vellanki[3], R. M. Mastan Shareef[4], Peruri Venkata Anusha[5], B Swarna[6], Geno Peter[7]

[1]Department of Mathematics, Chaitanya Bharathi Institute of Technology, Hyderabad, India
[2]Department of CSE-AIML, Aditya University, Surampalem, India.
[3]Department of Mathematics, B V Raju Institute of Technology, Narsapur, India.
[4]Department of FME, St.Martin's Engineering College, Secunderabad, India.
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
[6]University Centre for Research & Development, Chandigarh University, Mohali, India.
[7]School of Engineering and Technology, University of Technology Sarawak, Malaysia

**Corresponding author Email: mamtathakur_maths@cbit.ac.in*

The manuscript was received on 10 January 2025, revised on 10 June 2025, and accepted on 10 July 2025, date of publication 14 July 2025

Abstract

This research investigates MOPFSP-SDST, an advanced and highly computational scheduling difficulty in real-world manufacturing systems. It examines how it correlates with multi-objective permutation flow shops. LS-MOVNS stands for "Learning and Swarm-based Multi-objective Variable neighbourhood Search." It is a better metaheuristic method that combines evolutionary swarm search and adaptive local search techniques to address this Problem. The two main improvements have been discussed: a partial neighbourhood assessment framework that reduces the computational expenses by analysing only a particular portion of the neighbourhood, and an adaptable neighbourhood series selection procedure that rapidly chooses the most beneficial neighbourhood order depending on past performance rates. These improvements aim to make searches more effective and productive by finding a better balance between exploration and exploitation. Particularly in medium to large problem sizes, experimental tests in benchmark instances show that LS-MOVNS frequently outperforms current modern algorithms in convergence and diversity. The results verify the long-term reliability, scalability, and practical applicability of LS-MOVNS for resolving challenging multi-objective scheduling issues.

Keywords: *Variable Neighbourhood Search, Multi-Objective Scheduling, Learning and Swarm-Based Multi-Objective, Adaptive Local Search Technique, Partial Neighborhoods Assessment.*

1. Introduction

It is typical for production and manufacturing systems to encounter the well-known NP-hard computational optimization problem known as the permutation flow shop scheduling problem (PFSP) [1]. Traditionally, PFSP schedules various tasks on several machines in the same sequence to maximize performance factors like making pan or total flowtime. On the other hand, realistic production settings include more complexity, like sequence-dependent setup times (SDST), making the issue much more challenging [2]. Furthermore, decision-makers are usually expected to weigh many opposing goals concurrently—most often, lowering both makes pan and overall flowtime—thereby creating the multi-objective PFSP with sequence-dependent setup times (MOPFSP-SDST).

It is computationally impossible to solve MOPFSP-SDST problems of realistic size using traditional accurate algorithms, leading towards the widespread adoption of metaheuristic algorithms as a replacement [3]. The systematic neighborhood change technique of Variable neighborhood Search (VNS) assists in avoiding local optima for local optimization. Standard VNS, on the other hand, doesn't have good direction methods to balance discovery and exploitation, especially when there are several goals [3].

This paper presents LS-MOVNS, an upgraded hybrid technique, to solve these constraints. This approach uses evolving swarm algorithms' global search capacity, VNS's improved local search, and two unique strategies: adaptive neighborhood sequence choice & partial neighborhood assessment [4]. The method uses adaptive selection to dynamically identify optimal neighborhood directions by considering prior searches, while partial evaluation investigates an accurate subset of neighbors to decrease computation [5]. These developments seek to improve algorithm effectiveness, adaptability, and satisfactory solutions in complicated, large-scale MOPFSP-SDST scenarios [6].

2. Methods

Combining Learning & Swarm-based Mult-Objective Variable Neighborhood Search (LS-MOVNS) solves the MOPFSP-SDST Multi-Objective Permutation Flow Shop Scheduling Problem. This task aims to minimize specifications like make span, total flowtime, and total setup time by determining the ideal sequence of jobs across machines, taking into consideration the intricate relationships in work transitions. The suggested Learning and Swarm-based Multi-objective Variable Neighborhood Search (LS-MOVNS) is intended to successfully solve the Permutation Flow shop Scheduling Problem along with Sequence-Dependent Setup Times (MOPFSP-SDST) by balancing exploration (via swarm-based search) and exploitation (via VNS-based local search). This hybrid strategy uses clustering and learning tools to direct local search activities [7] carefully.

Figure 1 shows the learning and swarm-based multi-objective variable neighborhood optimization. The flowchart demonstrates a hybrid multi-objective optimization approach that employs Swarm Intelligence, Learning, and Variable Neighborhood Search. Before evaluating the solutions using multi-objective fitness criteria, the algorithm generates the initial population of solutions to maintain variety between solutions; a swarm-based search is done first, and afterwards, swarm locations are updated to improve the exploration process. Solutions are grouped; a learning phase refreshes the database of memories with solutions of excellent quality. Using a VNS-based local search, promising candidates have been chosen for further refinement. The findings are applied to modify the Pareto front, which denotes the collection of non-dominated solutions. Subsequently, a predetermined termination condition is satisfied, and this procedure continues until the algorithm produces the last Pareto-optimal solution [8].

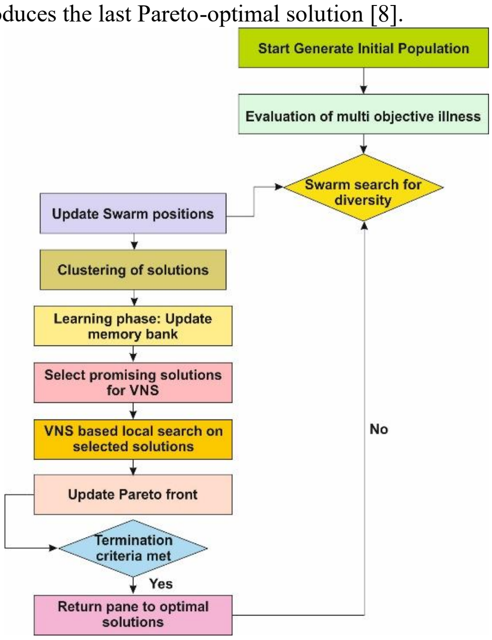


Fig 1. Learning and swarm-based multi-objective variable neighborhood optimization

Mathematical equations can be expressed as:

1. Completion Time Computation

$$C_{i,j} = \begin{cases} C_{i-1,j} + p^{i}_{\pi j} + s^{i}_{\pi j-1,\pi j} \; if \; i > 1 \\ C_{i,j-1} + p^{i}_{\pi j} + s^{i}_{\pi j-1,\pi j} \; if \; j > 1, i = 1 \\ p^{1}_{\pi j} \; if \; j = 1, i = 1 \end{cases}$$

Where, $s^{i}_{\pi j-1,\pi j}$ – Setup time between jobs, $C_{i,j}$- Completion time of job, $p^{i}_{\pi j}$- Processing time of job, π- Job Permutation

2. Total Flowtime

Total Flowtime = $\sum_{j=1}^{n} C_{m,j}$

$C_{m\ i}$ – completion time of job, n – total no. of jobs

3. Total Setup Time

Total setup time = $\sum_{i=1}^{m} \sum_{j=2}^{n} s^{\boldsymbol{i}}_{\boldsymbol{\pi} j-1,\pi j}$

m - Number of machines, n- Number of jobs, $s^{\boldsymbol{i}}_{\boldsymbol{\pi} j-1,\pi j}$- Setup time of machine i

4. Learning-Based Neighborhood Selection

$$P_{select}(\pi_j) = \frac{L(\pi_j)}{\sum_{k=1}^{n} L(\pi_k)}$$

$\sum_{k=1}^{n} L(\pi_k)$ – Normalization factor of utility score, $L(\pi_j)$- learned utility score of job of π_j.

5. Make span

$$Make\ span\ = C_{m,n}$$

C_m n – Completion time of the last task π_n

2.1. Completion Time Computation

The PFSP-SDST problem includes scheduling n tasks $(J_1, J_2, J_3, \dots, J_n)$ on m machines $(M_1, M_2, \dots, M_n)$. Every task $\{J_j\}_{j=1}^{n}$ has to be processed in the same machine sequence, from M_1 to Mm. Job Jj's processing time on machine M_j is represented by pi, j. Apart from processing times, there are additionally sequence-dependent setup times indicating that moving from job (J_k) to job (J_k) on machine (M_i) causes a setup time $S_{i,k,j}$ depending on the order of execution [9].

A schedule is a permutation $\pi = (\pi_1, \pi_2, \pi_3)$ which specifies the task sequence on every computer (i.e., jobs are run in the same order across all machines). The following three goals are concurrently taken into account for optimization:

1. Minimize the make span

$$C_{max} = \max_{j \in \{1,2,\dots,n\}} C_{m,} \pi_j$$

Where, $C_m \pi_j$ – Completion time of task π_j in the last machine M_m

2. Minimize the total flowtime

$$\sum_{j=1}^{n} C_{m,\pi_j}$$

$\sum_{j=1}^{n} C_{m,\pi_j}$- Sum of completion times

3. Total setup time

$$\sum_{i=1}^{m} \sum_{j=2}^{n} s_{\pi j-1,\pi_j}^{i}$$

$s_{\pi j-1,\pi_j}^{i}$ – Setup time in machine M_i

NP-hard, the PFSP-SDST involves complicated decisions among the processing speed and the extra overhead generated by the setup criteria. The multi-objective formulation aims to find a collection of Pareto-optimal plans that effectively balance these incompatible goals [10].

2.2. Initial Population Generation

Step 1: Generate Random or Heuristic-Based Solutions
It is a randomly created permutation or is built using a heuristic such as NEH (Nawaz-Enscore-Ham algorithm) [11].
$\pi^p = Generate\ permutation\ (n), for\ p = 1,2,3 \dots P$
Step 2: Evaluate Each Solution
For each $\pi^p \epsilon P$, mathematical computation is described as
Make span = $C_{m,n}(\pi^p)$

Total flow time = $\sum_{j=1}^{n} C_{m,j}(\pi^p)$

Total setup time = $\sum_{i=1}^{m} \sum_{j=2}^{n} s^i \pi_{j-1,}^{p} \pi_j^p$

Step 3: Store in Initial Archive

P = $(\pi^1, \pi^2, \pi^3, \dots \dots \pi^p)$

After each answer, π^P has been examined, the swarm and learning elements record it for future processing.

2.3. Learning-Based Scoring

Job positions should be given utility ratings L (π_i) according to their past influence on high-quality solutions [12]. Calculate probabilistic selection values P_{select} (π_i) to direct neighbourhood selection. Utility ratings based on prior high-quality solution performance influence neighbourhood search job selection [13].

Step 1: Initialize Utility Scores

$L(\pi_j) = 1$ for all $j = 1,2,3, \dots . n$

Where, $L(\pi_i)$ – Learning score of activity π_i

Step 2: Update Scores Based on Quality

Every solution π^p in the non-dominated archive A is raised based on the input:

$L(\pi_j) \leftarrow L(\pi_j) + \delta_i^P$

$\Delta jP = \delta_i^P$ = favourable position in π^p

Step 3: Normalize Scores to Probabilities

$$P_{select}\left(\pi_j\right) = \frac{L(\pi_j)}{\sum_{k=1}^{n} L(\pi_k)}$$

2.4. Swarm-Based Diversification

Engage swarm intelligence, such as particle swarm optimization ideas, to examine numerous areas of the solution space. Apply velocity-inspired task swaps or mutations to share knowledge among elite solutions [14].

1. Guided Position Exchange (Inspired by Velocity)

$$\pi^{new} = \pi^p \pm \alpha(\pi^{gbest} - \pi^p)$$

$(\pi^{gbest} - \pi^p)$ – Positional differences under the path π^p towards π^{gbest} and $\pm$ – permutation – based combination operator

2. Randomized Blending for Exploration

$$\pi^{new} = Blend\ (\pi^p, \pi^{rand}, \beta)$$

Where, Blend – Randomised operator (e.g., Partial mapping), π^{rand} – Controls the influence of β

2.5. Variable Neighbourhood Search (VNS) Intensification

Variable Neighbourhood Search (VNS) is used in the LS-MOVNS framework as a local intensification tool to improve promising solutions and accelerate integration toward the Pareto-optimal front. This component methodically investigates many neighbourhood structures, including job swaps, insertions, and block reversals, to escape local optima and find better-performing permutations. Using the present neighbourhood structure as a starting point, a shaking phase creates a random neighbour for each solution. Then, a local search technique seeks a non-dominated improvement. Improved neighbours are approved and saved in the external archive; otherwise, the search moves to the next neighbourhood. This methodical investigation helps the algorithm to balance intensity and diversity properly [15]. Moreover, by giving employment positions with earlier high-value priority, the learning-guided process affects neighbourhood selection, directing computational effort toward the most promising areas of the solution space. Everything considered, refining elite solutions and preserving variety along the Pareto front depend strongly on the VNS intensification phase [16].

2.6. Clustering and Archive Management

The LS-MOVNS system has a special strategy for managing clusters and archives to ensure that the new set of solutions is diverse and high-quality [17]. An external archive stores non-dominated solutions across iterations while the search continues. Clustering reduces memory utilization and redundancy in this collection—a distance measure like Euclidean or cosine similarity groups solutions by objective value similarity. Selecting the most elite or representative solution from each cluster helps maintain variety on the Pareto front despite removing duplicates or those with similar characteristics. A well-distributed Pareto front approximation and reduced computing cost are achieved using this method [18]. When a new non-dominated solution is found, it is added to the archive or utilized to take over a weaker, clustered member if it improves spread or performance. This archive management system ensures a restricted but diversified group of high-quality solutions is available during optimization [19].

2.7. Termination

The LS-MOVNS termination criteria balance solution quality and computing economy by ending the search process. The algorithm stops working when a condition is satisfied, which is usually based on a maximum time limit, number of iterations, or convergence threshold. While iterating, a convergence-based termination might keep track of the Pareto front's progress and stop the process when the improvements drop below a certain threshold after a certain number of iterations. The approach incorporates adaptive checks to prevent calculations that are no longer needed if the solution variation and quality are sufficiently high. In the end, decision-makers get a well-approximated Pareto front regarding the MOPFSP-SDST in the form of the final non-dominated archive, which contains various high-quality solutions. This shutdown method keeps LS-MOVNS efficient and resilient in real-world scheduling situations [20].

3. Result and Discussion

Optimizing two competing objectives by make span and total flowtime, the suggested LS-MOVNS (Learning and Swarm-based Multi-objective Variable Neighbourhood Search) method achieves good results in resolving the permutation flow shop scheduling issue with sequence-dependent setup durations. Achieving a solid equilibrium between exploration and exploitation, LS-MOVNS integrates swarm intelligence in global search with a variable neighbourhood search (VNS) supplemented with machine learning-based algorithms for

local search. Using k-means clustering techniques to find practical solutions & a flexible neighbourhood selection system that adapts on the fly depending on prior findings constitute two significant innovations. This study proposes LS-MOVNS, a hybrid metaheuristic that solves the difficult MOPFSP-SDST (Multi-Objective Permutation Flow Shop Scheduling Problem) involving sequence-dependent setup durations. The two objectives are achieving a minimum Make span (Cmax) & total flowtime (TFT).
Table 1 shows the IGD comparison of modern methods. The study highlights that the two cutting-edge methods, MO-MLMA & RIPG, LS-MOVNS, always have the lowest IGD values, suggesting that their Pareto fronts are broader and have better convergence.
The Inverted Generational Distance (IGD) measure for four benchmark cases of the permutation flow shop scheduling issue with sequence-dependent setup durations is compared in the Table by applying three algorithms: RIPG, MO-MLMA, & the suggested LS-MOVNS. LS-MOVNS frequently matches and beats the other techniques in IGD values across all problem sizes. The Inverted Generational Distance (IGD) measure for four benchmark cases of the permutation flow shop scheduling issue with sequence-dependent setup durations is compared in the Table by applying three algorithms: RIPG, MO-MLMA, & the suggested LS-MOVNS. LS-MOVNS frequently matches and beats the other techniques in IGD values across all problem sizes. The three methods achieved an IGD of 0.02 within the small-scale instance (20×5), indicating similar results in integration and diversity. In the medium-sized issue (50×10), LS-MOVNS and MO-MLMA both surpass RIPG, with a decreased IGD of 0.03 compared with RIPG's 0.04. The greater instance (100×10) demonstrates comparable patterns, with LS-MOVNS and MO-MLMA subsequently exceeding RIPG (0.05) by 0.04. Each method accomplishes an identical IGD value of 0.02 for the highest instance (200×20), showing comparable effectiveness at scale. The results emphasize the viability of LS-MOVNS, particularly in large and medium-sized problems, which indicates enhanced convergence and solution variation over the latest techniques.

Table 1. IGD comparison of modern methods

Problem (Jobs x Machines)	RIPG IGD	MO-MLMA IGD	LS-MOVNS IGD (BEST)
20X5	0.02	0.02	0.02
50x10	0.04	0.03	0.03
100x10	0.05	0.04	0.04
200x20	0.02	0.02	0.02

Figure 2 demonstrates the different algorithms with varied problem sizes, which allows LS-MOVNS to efficiently traverse complex and large search spaces through a clearly defined loop that operates on offspring generation, clustering-based selection, adaptive local search, and swarm revision. Based on these findings, LS-MOVNS is a scalable and competitive option for solving multi-objective scheduling issues.

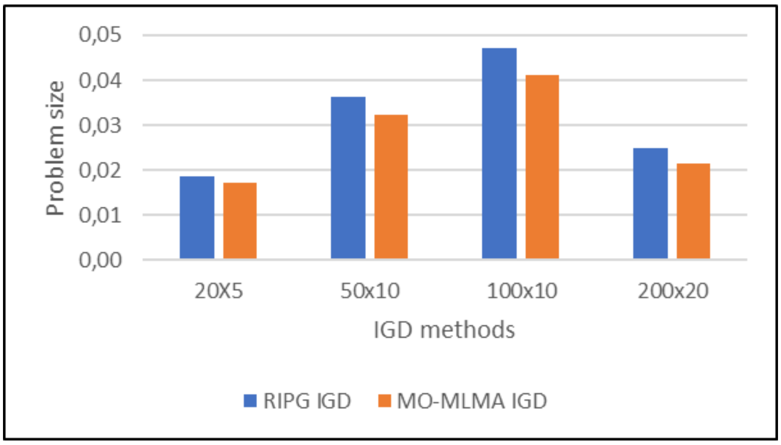


Fig 2. Inverted Generational Distance with four algorithms

Table 2 demonstrates the Hyper Volume comparison. Table 2 compares the Hypervolume (HV) values for different problem sizes using RIPG, MO-MLMA, & the proposed LS-MOVNS method. HV values indicate the convergence and the wide range of the Pareto front. Achieving an optimal HV value of 0.51 in a small-scale business instance (20×10), LS-MOVNS partially exceeds both MO-MLMA and RIPG, suggesting a more robust and well-distributed Pareto front. MO-MLMA, possessing an HV of 0.45, results in the medium-sized issue (50×10), closely beating LS-MOVNS (0.44) and strongly exceeding RIPG (0.41). With a value of 0.42, LS-MOVNS remains competitive in the 100×10 circumstance, just behind MO-MLMA (0.41), whereas both surpass RIPG (0.37), showing greater coverage in objective space. The large-scale instance (200\u00d710) shows the most apparent rise, as LS-MOVNS excels both MO-MLMA (0.63) and RIPG (0.54), reaching the maximum HV of 0.66, proving its excellent scalability and resilience in high-dimensional problem environments. Despite being considered, our investigations encourage the success of LS-MOVNS, specifically in complex or significant instances, to generate high-quality and various Pareto-optimal solutions.

Table 2. Hyper Volume comparison (HV)

Problem (Jobs x Machines)	RIPG IGD	MO-MLMA IGD	LS-MOVNS IGD (BEST)
20X10	0.49	0.50	0.51
50x10	0.41	0.45	0.44
100x10	0.37	0.41	0.42
200x10	0.54	0.63	0.66

Figure 3 depicts the HV comparison of various inverted generational distance techniques. As this bar graph mentions, the X-axis corresponds to problem size, and the Y-axis corresponds to Hypervolume. This data shows that LS-MOVNS IGD (BEST) beats RIPG IGD in problem size and Hypervolume, with better accuracy and performance in multiple Pareto-front optimal solutions.

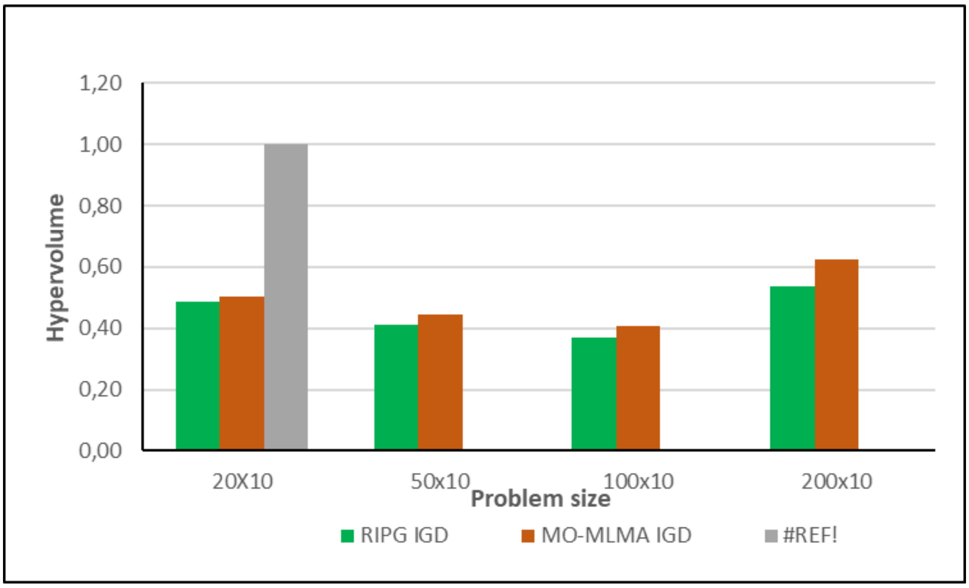


Fig 3. HV comparison of various techniques

The predicted LS-MOVNS strategy and the randomized equivalent are discussed in Table 3. The initial approach utilizes clustering to identify probable solutions, whereas the other approach employs the Inverted Generational Distance (IGD) parameter to evaluate efficiency across three different issue sizes. Every combination achieves an IGD of 0.02 in the small instance (20×10), revealing that learning-based solution preference offers a less significant influence at smaller sizes. However, in the case of the medium-sized sample (50×20), LS-MOVNS showed more successful results than its randomized equivalent, with an IGD of 0.04 as opposed to 0.06. This emphasizes the efficacy of directed solution choice via clustering in improving diversity and convergence. The randomized variation unexpectedly gets a superior IGD of 0.03 compared to 0.06 from LS-MOVNS in the large-scale instance (200×20), making it quite enjoyable. This may imply that, on occasion, randomization may improve performance for huge and complicated problems by adding beneficial variation. The findings showcase that the learning-based method in LS-MOVNS might enhance the quality of solutions in general, though its efficiency may be based on the size and structure of the Problem.

Table 3. IGD Differentiation Between LS-MOVNS And LS-MOVNS Randomised Strategy

Problem size	LS-MOVNS	LS-MOVNS randomised
20X10	0.02	0.02
50x20	0.04	0.06
200x20	0.06	0.03

Figure 4 shows the differentiation of IGD between LS-MOVNS and LS-MOVNS randomised. In this bar chart, the X-axis is LS-MOVNS randomised and the Y-axis is LS-MOVNS. In these two methods, problem sizes ranging from (200x20) in LS-MOVNS and LS-MOVNS randomised (50x20) perform excellent metrics in accuracy,

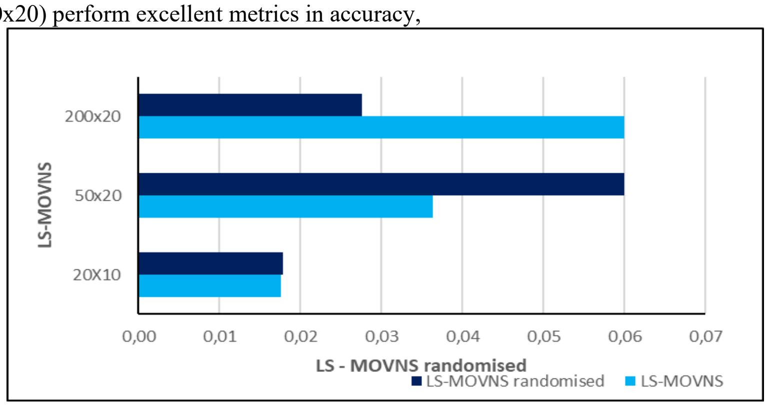


Fig 4. Differentiation of IGD between LS-MOVNS and LS-MOVNS randomised

Table 4 depicts the hypervolume comparison of LS-MOVNS HV and MO-MLMA HV. It contrasts the Hypervolume (HV) measure of the proposed LS-MOVNS method using the reference MO-MLMA system covering 20x10 & 50x20 problems. The HV metric evaluates Pareto front convergence & diversity. In the 20x10 Problem, LS-MOVNS slightly beats MO-MLMA with an HV of 0.42 vs. 0.40, suggesting superior solution spread and closeness towards the ideal front. In the greater 50x20 scenario, LS-MOVNS obtains a significantly larger HV of 0.65 compared to MO-MLMA's 0.26. This striking difference indicates LS-MOVNS' better scalability and durability in maintaining varied, high-quality solutions despite increasing problem complexity. These results highlight LS-MOVNS's ability to offer well-distributed Pareto fronts, even in intricate scheduling environments.

Table 4. Hypervolume comparison of LS-MOVNS HV and MO-MLMA HV

Problem size	LS-MOVNS HV	MO-MLMA HV
20X10	0.42	0.40
50x20	0.65	0.26

Figure 5 shows the Hypervolume of LS-MOVNS and MO-MLMA. The X-axis corresponds to the Hypervolume, and the Y-axis represents the methods of IGD. LS-MOVNS HV performs 0.65 in 50x20 and 0.42 in 20x10 in these two techniques.

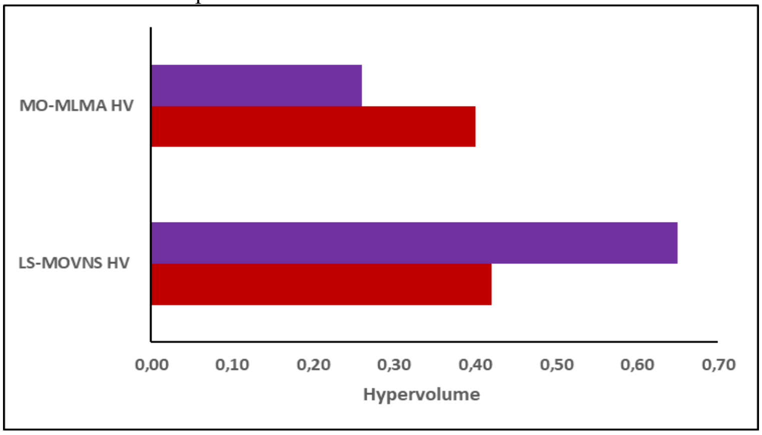


Fig 5. Hypervolume comparison of LS-MOVNS and MO-MLMA

In MOPFSP-SDST, the Variable neighbourhood Search (VNS) concept is a powerful metaheuristic that examines additional neighbourhoods to avoid local optimal settings. The study encompassed flexible neighbourhood series generation & partial neighbourhood evaluation using VNS. The dynamic sequence decision automatically selects the most beneficial neighbourhood order based on prior performance, optimizing search efficiency, especially in higher problem instances, by targeting greater potential search routes. Simply randomly choosing 50% of every neighbourhood, the partial assessment method drops computation stress & accelerates the search without lowering the quality of the solution. Three configurations—LS-MOVNS with both upgrades, LS-MOVNS random select (randomized neighbourhood sequence), & LS-MOVNS full assessed these improvements. LS-MOVNS consistently surpassed the other variations in convergence (lower IGD values) & computational speed. The adaptive technique performed well in medium to large-scale challenges, indicating the necessity of intelligently connecting neighbourhood searches towards balanced exploitation and exploration.

Across three problem instances, Table 5 depicts the corresponding efficiency of all three variations based on the LS-MOVNS algorithm: random neighbourhood series (LS-MOVNS random select), performing neighbourhood search (LS-MOVNS full), and suggested adaptable in partial evaluation (LS-MOVNS). The three different patterns offer similar IGD values for the smaller problem size (50×5); LS-MOVNS full performs somewhat better (IGD = 0.0235) than the others, indicating that complete neighbourhood exploration could be more successful if computation complexity is relatively low. However, the benefits of the adaptive and partial techniques become clearer as the issue size increases. Illustrating its superior convergence, LS-MOVNS across the medium-sized instance (100×10) acquires the lowest IGD of 0.0392, overcoming both the random preference (0.0435) and total search (0.0449) variants. In contrast, LS-MOVNS achieves the highest IGD of 0.019, significantly lower than randomized and complete setups; the large-scale issue (200×20) shows the same pattern. These findings verify that, particularly in bigger and more complicated scheduling situations, the adaptive neighbourhood selection paired with partial assessment improves search efficacy and the effectiveness of solutions.

Table 5. Variable Neighbourhood Search

Problem size	LS-MOVNS random select	LS-MOVNS full	LS-MOVNS
50X5	0.0242	0.0235	0.0239
100x10	0.0435	0.0449	0.0392
200x20	0.0231	0.0247	0.019

4. Conclusion

Finally, the experimental findings show that the proposed LS-MOVNS method, especially in adaptable neighbourhood sequence choice & partial neighbourhood evaluation improvement functions. Inverted Generational Distance (IGD) is more accurate for LS-MOVNS than randomized selection and complete neighbourhood search, especially when issue size rises. In addition to improving convergence, the adaptive technique maintains solution diversity & reduces computation cost. The ability to dynamically direct local searches according to performance history & concentrate computational resources towards probable solution space areas is extremely valuable. Since it balances exploration and exploitation, LS-MOVNS is stable and scalable for complicated multi-objective scheduling problems involving sequence-dependent setup periods.

References

[1] A. A. Almazroi and C. A. U. Hassan, “Nature-inspired solutions for energy sustainability using novel optimization methods,” *PLoS One*, vol. 18, Nov. 2023, doi: 10.1371/journal.pone.0288490.

[2] E. A. Aldhahri, A. A. Almazroi, and N. Ayub, "Nature-inspired approaches for clean energy integration in smart grids," *Alexandria Eng. J.*, vol. 105, pp. 640–654, Oct. 2024, doi: 10.1016/j.aej.2024.08.003.

[3] F. Alsokhiry, P. Siano, A. Annuk, and M. A. Mohamed, “A Novel Time-of-Use Pricing Based Energy Management System for Smart Home Appliances: Cost-Effective Method,” *Sustain.*, vol. 14, Nov. 2022, doi: 10.3390/su142114556.

[4] F. Koopmans, K. W. Li, R. V. Klaassen, and A. B. Smit, “MS-DAP Platform for Downstream Data Analysis of Label-Free Proteomics Uncovers Optimal Workflows in Benchmark Data Sets and Increased Sensitivity in Analysis of Alzheimer’s Biomarker Data,” *J. Proteome Res.*, vol. 22, pp. 374–386, Feb. 2023, doi: 10.1021/acs.jproteome.2c00513.

[5] X.S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, pp. 78–84, 2010, doi: 10.1504/IJBIC.2010.032124.

[6] B. A. Kumari and K. Vaisakh, “Ensuring expected security cost with flexible resources using modified DE algorithm based dynamic optimal power flow,” *Appl. Soft Comput.*, vol. 124, Jul. 2022, doi: 10.1016/j.asoc.2022.108991.

[7] B. Aruna Kumari and K. Vaisakh, “Integration of solar and flexible resources into expected security cost with dynamic optimal power flow problem using a Novel DE algorithm,” *Renew. Energy Focus*, vol. 42, pp. 48–69, Sep. 2022, doi: 10.1016/j.ref.2022.03.008.

[8] B. Aruna Kumari, K. Vaisakh, and N. C. Sahoo, “A Novel DE algorithm for solving dynamic OPF for expected security cost incorporating solar and flexible resources in smart grids.”

[9] V. Lai, Y. F. Huang, C. H. Koo, A. N. Ahmed, and A. El-Shafie, “A Review of Reservoir Operation Optimisations: from Traditional Models to Metaheuristic Algorithms,” Aug. 2022, *Springer Science and Business Media B.V.* doi: 10.1007/s11831-021-09701-8.

[10] H. Youssef, S. Kamel, M. H. Hassan, E. M. Mohamed, and S. Mouassa, “Cost-efficient smart home energy management with hybrid quadratic interpolation tuna swarm optimization,” *Cluster Comput.*, vol. 28, Oct. 2025, doi: 10.1007/s10586-024-05078-y.

[11] J. L. K. Grace, A. Maneengam, P. K. V. Kumar, and J. Alanya-Beltran, “Design and Implementation of Machine Learning Modelling through Adaptive Hybrid Swarm Optimization Techniques for Machine Management,” *Evergreen*, vol. 10, pp. 1120–1126, Jun. 2023, doi: 10.5109/6793672.

[12] Ravi Yadav, Ashok Kumar Pradhan, Wavelet probability distribution mapping for detection and correction of dynamic data injection attacks in WAMS, International Journal of Electrical Power & EnergySystems,Volume134, 2022, 107447, ISSN01420615, https://doi.org/10.1016/j.ijepes.2021.107447.

[13] I. Dagal *et al.*, “Prioritized Multi-Step Decision-Making Gray Wolf Optimization Algorithm for Engineering Applications,” *Eng. Reports*, vol. 7, May 2025, doi: 10.1002/eng2.70154.

[14] S. Mimi, Y. Ben Maissa, and A. Tamtaoui, “Optimization Approaches for Demand-Side Management in the Smart Grid: A Systematic Mapping Study,” Aug. 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. Doi: 10.3390/smartcities6040077.

[15] W. Alexan, M. Gabr, E. Mamdouh, R. Elias, and A. Aboshousha, “Color Image Cryptosystem Based on Sine Chaotic Map, 4D Chen Hyperchaotic Map of Fractional-Order and Hybrid DNA Coding,” *IEEE Access*, vol. 11, pp. 54928–54956, 2023, doi: 10.1109/ACCESS.2023.3282160.

[16] E. Uwimana and Y. Zhou, “A Novel Two-Stage Hybrid Model Optimization with FS-FCRBM-GWDO for Accurate and Stable STLF,” *Technologies*, vol. 12, Oct. 2024, doi: 10.3390/technologies12100194.

[17] S. Kadam and D. I. Kim, “Knowledge-Aware Semantic Communication System Design and Data Allocation,” *IEEE Trans. Veh. Technol.*, vol. 73, pp. 5755–5769, Apr. 2024, doi: 10.1109/TVT.2023.3333350.

[18] M. A. Ali, F. Naeem, N. Tariq, I. Ahmed, and A. Imran, “Bioactive Nutrient Fortified Fertilizer: A Novel Hybrid Approach for the Enrichment of Wheat Grains With Zinc,” *Front. Plant Sci.*, vol. 12, Dec. 2021, doi: 10.3389/fpls.2021.743378.

[19] L. Zhang and R. Tang, “Dispatch for a Continuous-Time Microgrid Based on a Modified Differential Evolution Algorithm,” *Mathematics*, vol. 11, Jan. 2023, doi: 10.3390/math11020271.

[20] L. Nasser and T. Jamshid, “A Hybrid Method based on SA and VNS Algorithms for Solving DAP in DDS,” https://ibn.idsi.md/vizualizare_articol/138676.