Italian Manufacturing Association Conference - XVII AITeM Materials Research Forum LLC
Materials Research Proceedings 57 (2025) 521-529 https://doi.org/10.21741/9781644903735-61

Metaheuristic approaches for solving the scheduling problem in the semiconductor diffusion work area

Roberto Rosario CORSINI[1,a]*, Antonio COSTA[1,a]*, Valeria FICHERA[1,a],
Giuseppe ORITI[1,a] and Daniele PAGANO[2,b]

[1]University of Catania, Italy

[2]STMicroelectronics, Catania, Italy

[a]roberto.corsini@unict.it, [b]antonio.costa@unict.it, [c]valeria.fichera@phd.unict.it,
[d]giuseppe.oriti@phd.unict.it, [e]daniele.pagano@st.com

Keywords: Scheduling, Semiconductor Manufacturing, Smart Manufacturing

Abstract. Motivated by a real case study of a semiconductor company, the paper addresses the scheduling problem in the diffusion area, which is recognized as one of the most critical stages in semiconductor manufacturing. This work area can be modeled as a two-stage hybrid flow shop scheduling problem with re-entrant flows. The first stage is the cleaning process, wherein impurities are removed from the wafer surface. The second one entails several tasks, such as diffusion, deposition and oxidation, to be performed into different furnaces. Based on the product recipe, wafer lots may undergo multiple operations in the second stage. The diffusion area also includes several production constraints, such as p-batching, machine eligibility and Q-time constraint, which increase the complexity of the problem. A novel metaheuristic named the Modified Harmony Search algorithm is proposed to minimize the average flow time of the system.

1. Introduction and background

Semiconductor manufacturing systems are complex, multi-stage production environments where raw silicon wafers are transformed into integrated circuits (ICs) through a large sequence of operations. The production process is divided into front-end and back-end manufacturing. The front-end phase consists of wafer fabrication, which involves hundreds of machines performing multiple operations to produce ICs, while the back-end phase encompasses final operations, such as testing and packaging, before market release. The front-end is particularly challenging due to the technological complexity and the extensive processing requirements involved. As detailed in [1], the front-end semiconductor manufacturing system is organized into different work areas, each dedicated to a specific set of operations, namely diffusion, photolithography, etching, ion implantation, and planarization. Among them, the diffusion area is widely recognized as one of the most critical stages due to long processing times and complexity of production constraints. Given its industrial significance, the scheduling problem in the diffusion work area has recently garnered growing interest from both academic researchers and practitioners (see, e.g., [2-7]).

The diffusion area is typically modeled as a two-stage flexible flow shop with re-entrant flows. Specifically, in the first stage cleaning machines are used to remove material impurities from wafer surfaces, while in the second stage furnaces execute one of three alternative operations, i.e., diffusion, deposition or oxidation. A key complexity factor is the re-entrant nature of the problem, as a wafer lot may require multiple consecutive operations executed by furnaces. Furthermore, the diffusion work area is characterized by several stringent production constraints, including: p-batching with incompatible job operations, machine eligibility, dynamic job arrivals, waiting time limits between two consecutive operations, machine-dependent batch size constraints, operation- and machine-dependent processing times and preloading operations. These scheduling constraints further increase the complexity of the scheduling problem, making it challenging to achieve optimal or near-optimal solutions within a reasonable computational time. To tackle the scheduling problem in the diffusion work area of semiconductor manufacturing systems, metaheuristic algorithms have been widely explored in recent literature. Among these, Simulated Annealing (SA) has been one of the most frequently proposed methods [2,3,5,9], while alternative approaches such as Variable Neighborhood Search (VNS) [4], Harmony Search (HS) [6], and Genetic Algorithms (GA) [7,8] have also been explored. Motivated by the European project named HICONNECTS, this work deals with a real-world scheduling problem in the diffusion work area of a semiconductor manufacturing facility located in the Southern Italy, with the aim of providing an efficient metaheuristic algorithm able to optimize a turnaround performance measure of the system. The main contributions of this work, with respect to the recent literature, are summarized as follows:

1. The proposed scheduling model integrates a comprehensive set of production constraints that are specific to the real-world diffusion work area under investigation. While previous studies [2–9] have commonly considered p-batching, machine eligibility, waiting time limits, and dynamic job arrivals, this work introduces additional complexities, i.e., machine-dependent minimum and maximum batch size, operation and machine-dependent processing times and preloading operations. Including these additional constraints enables a more accurate replication of the real operations within the diffusion work area, while simultaneously increasing the complexity of the scheduling problem under investigation.
2. To solve the scheduling problem at hand, a Modified Harmony Search (MHS) algorithm is proposed. The proposed MHS enhances the traditional HS approach by introducing adaptive parameter control and an embedded Local Search (LS) mechanism to improve solution quality and convergence speed. An experimental evaluation based on industrial data is conducted to assess the effectiveness of MHS compared to the standard HS algorithm [6] and the SA algorithm [2]. The flowtime performance measure has been considered as the metric for the experimental comparison

The structure of the paper is as follows. Section 2 describes the problem of the diffusion work area under study. Section 3 presents in detail the proposed MHS metaheuristic algorithm. Section 4 shows the results arising from the experimental comparison between MHS, HS and SA. Finally, Section 5 reports the concluding remarks and directions for future research.

2. Problem description

In the diffusion work area, which is illustrated in Figure 1, N wafer lots, which can be referred to as jobs ($j=1, ..., N$), are processed in a re-entrant two-stage flexible flow shop. In the first stage ($s=1$), cleaning operations are performed by M_1 cleaning machines while in the second stage, diffusion, deposition and oxidation operations are executed by M_2 furnaces. Hereinafter, the two stages are also denoted as cleaning and diffusion stages. The problem is re-entrant since, based on the wafer lot recipe φ, a wafer lot may require two consecutive diffusion operations. Furthermore, some jobs may need only a cleaning operation, while others may require only one or two furnace operations. The recipe φ also specifies the type of operation o_{js} of job j at stage s. The processing time p_{om} depends on the operation o_{js} and the selected machine m_s. The problem is subject to dynamic job arrivals, as job release times r_{js} can be greater than zero at the first stage ($r_{j1} \geq 0$), while at the second stage, they are equal to job completion times of the first stage, i.e., $r_{j2}=C_{j1}$. Each machine operates under p-batching constraint with incompatible job operations. Jobs requiring the same operation o_{js} are collected in batches b_s at each stage and processed in parallel. Batch release time r_b is the maximum job release time in the batch, i.e., $r_b=\max\{r_{js}\}$ where $j \in b_s$. Each machine m_s has a minimum $BMIN_{ms}$ and a maximum $BMAX_{ms}$ batch size constraint. Furthermore, there is the machine eligibility constraint since the o_{js} can be processed by a specific set of machines M_{os}, where $M_{os} \subseteq M_s$. Two different waiting time constraint QW are imposed in the inter-operational buffer between the first and second stage to prevent job deterioration and ensure quality standards. Specifically, if the waiting time wt_j of a job coming from the cleaning stage exceeds QW_{CD}, it must be reworked and returned to the cleaning buffer. Conversely, if the waiting time wt_j of a re-entrant job from the diffusion stage exceeds QW_{DD} (i.e., waiting time constraint between two consecutive diffusion operations), then the job must be scrapped. Additionally, each diffusion furnace can preload a second batch, which is processed immediately after the first batch is completed. The preloading operation allows a batch to be inserted into the machine in advance, thereby protecting it from exposure to environmental conditions that could compromise the quality of the jobs. For this reason, the waiting time constraint for jobs in the preloaded batch is relaxed. In this study, the scheduling problem in the diffusion work area consists of selecting the batches to be loaded at each stage during an 8-hour work shift. The objective is to minimize the average flowtime of the jobs processed within the shift. The average flowtime F is calculated as follows:

$$F = \sum_{j=1}^{N} (C_{j2} - r_{j1}) / N \tag{1}$$

where C_{j2} is the completion time of job j after its final operation in a furnace, r_{j1} is the job release time at cleaning stage and N is the total number of jobs processed during the shift.

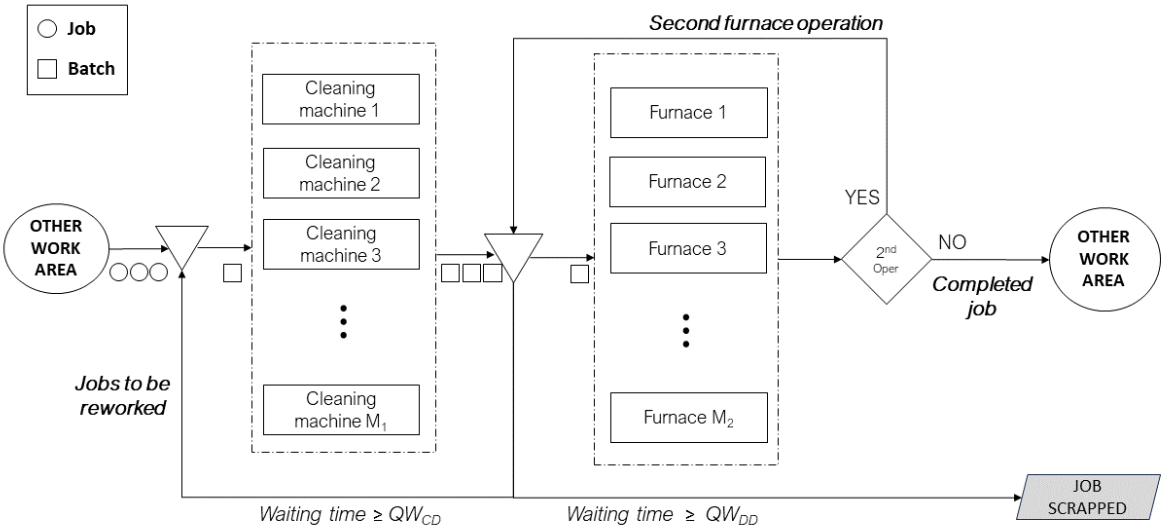


Figure 1. Schematic representation of the diffusion work area.

3. Modified Harmony Search algorithm

The HS algorithm is a well-known metaheuristic inspired by the process of musical improvisation. It is a population-based algorithm, where the population is called Harmony Memory (HM) and consists of a set of candidate solutions, referred to as harmonies:

$$HM = \{X_1, X_2, \dots, X_{HMS}\} \tag{2}$$

where X_k represents a candidate solution (harmony), and *HMS* (Harmony Memory Size) is the total number of harmonies stored in the HM. Each harmony X_k is a N-dimensional real-valued vector:

$$X_k = \{x_k(1), x_k(2), \dots, x_k(N)\} \tag{3}$$

where $x_k(j)$ is a real value within the range $[x_{min}, x_{max}]$. Each harmony is then transformed into a sequence of wafer lots using the Smallest Position Value (SPV) rule. In this study, a MHS algorithm with two key enhancements over the traditional HS is proposed. The former consists of the integration of a LS strategy. To avoid stagnation in local optima, a probabilistic LS mechanism is applied to improve the worst solutions. Precisely, the LS process performs random swap operations on the worst solution of the HMS to explore candidate solutions. The latter is the adaptive control of HS and LS parameters. Instead of using fixed values, the algorithm's parameters are dynamically adjusted during the search process based on the number of iteration ($iter$):

$$\rho(iter) = \rho^U + \frac{(iter - iter_{max})\cdot(\rho^U - \rho^L)}{iter_{max}} \tag{4}$$

where $\rho(iter)$ varies within the range [ρ^L, ρ^U] and can be applied to ρ_{cr}, ρ_{ar} and the LS parameters. The pseudo code of MHS and LS is reported in Algorithm 1 and 2, respectively.

In Step 1, HS and LS parameters are initialized. The HM is created by generating *HMS* harmonies, where each harmony represents a candidate scheduling solution. Each harmony is constructed as a real-valued vector using the following formula:

$$X = x_{min} + (x_{max} - x_{min}) \cdot rand(N) \tag{5}$$

where rand(N) is a N-dimensional vector of random values in [0,1]. To enhance the initial solution, one randomly selected harmony is replaced by the First-In-First-Out (FIFO) scheduling rule. The objective function $F(X)$ of each harmony is evaluated using a simulation model coded in Python®, which emulates the diffusion work area of the semiconductor manufacturing system. The worst and best harmonies, X_{worst} and X_{best}, are identified and recorded. Step 2 consists of the Harmony Improvement (HI) phase, where new harmonies X_{new} are iteratively generated and evaluated. The number of iterations without any improvement, denoted as γ, is tracked to determine when the LS must be applied. At each iteration, the algorithm dynamically adjusts the parameters ρ_{cr} and ρ_{ar} and α as in Equation 4, so as to ensure a gradual evolution of the algorithm from the exploration to the exploitation phase. α is an adaptive parameter that defines the maximum number of iterations allowed without improvement in the algorithm. In Step 3, a new harmony X_{new} is then generated based on three possible mechanisms. First, with probability ρ_{cr}, the value of $x_{new}(j)$ is taken from an existing harmony in HM. In this case, there is an additional probability ρ_{ar} that $x_{new}(j)$ is adjusted with the real-value in position j, $x_{best}(j)$, of the best solution X_{best}. Otherwise, $x_{new}(j)$ is randomly generated in [x_{min}, x_{max}]. In Step 4, X_{new} is evaluated and if $F(X_{new})$ is lower than $F(X_{worst})$, it replaces X_{worst} in the HM and the new worst harmony is recorded. Additionally, if $F(X_{new})$ is lower than $F(X_{best})$, it becomes the new best solution and γ is set to zero. In Step 5, the LS is applied if γ is greater than α, where α is an adaptive parameter that assumes values within the range [α^L, α^U], as in Equation 4. Further LS parameters are β and $eval_{LS}$, which controls the probability of performing a swap operation and defines the number of solutions evaluated by LS, respectively. The LS procedure is performed on X_{LS}, initially set as X_{worst}, by iteratively applying swap operations with probability β, which is also dynamically updated as in Equation 4. A total of $eval_{LS}$ number of new solutions are evaluated, and the best one, denoted as $X_{LS,best}$, is selected. If $X_{LS,best}$ improves X_{worst}, then $X_{LS,best}$ is inserted into the HM and the new X_{worst} is recorded. Furthermore, if $F(X_{LS,best})$ is lower than $F(X_{best})$, then $X_{LS,best}$ becomes the new best solution. MHS is stopped in Step 6 of Algorithm 1 when the maximum number of iterations, $iter_{max}$, is reached, returning best harmony X_{best}.

Algorithm 1. MHS

```
Step 1:  Initialization
         Initialize HS and LS parameters. Generate the initial HM. Evaluate HM and store $X_{worst}$ and $X_{best}$.
Step 2:  Harmony Improvement
         for iter in $iter_{max}$:
             $\gamma = \gamma + 1$; Calculate adaptive values for $\alpha$, $\rho_{cr}$, $\rho_{ar}$ (Equation 4)
Step 3:  Generation of a new harmony
             $X_{new} = []$
             for j in N:
                 if $rand < \rho_{cr}$ then $x_{new}(j) = x_a(j)$ where $X_a$ is randomly selected from HM
                     if $rand < \rho_{ar}$ then $x_{new}(j) = x_{best}(j)$
                 else $x_{new}(j) = x_{min} + [x_{max} - x_{min}] \cdot rand$
Step 4:  HM update
             if $F(X_{new}) < F(X_{worst})$ then replace $X_{worst}$ with $X_{new}$ in HM and update $X_{worst}$
                 if $F(X_{new}) < F(X_{best})$ then $X_{best} = X_{new}$ and $\gamma = 0$
Step 5:  LS application
             if $\gamma > \alpha$ then $\gamma = 0$; apply LS (see Algorithm 2)
         Stopping the algorithm
Step 6:  Return $X_{best}$ as the optimal solution
```

Algorithm 2. LS

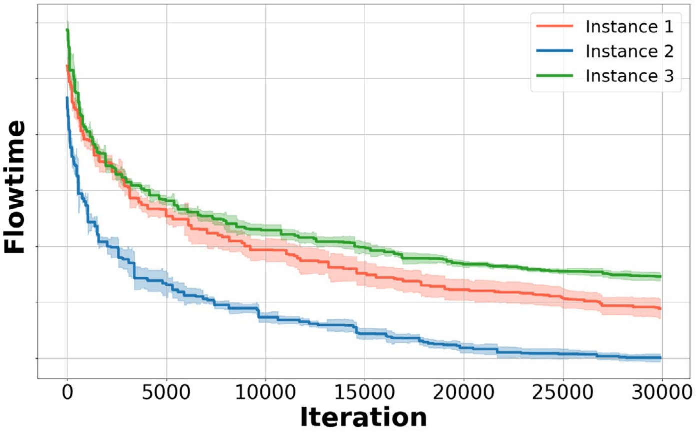
```
Step 5.1:  LS initialization
           Compute adaptive value for $\beta$ (Equation 4)
           for i in $eval_{LS}$:
               $X_{LS} = X_{worst}$
Step 5.2:  Performing swap operations
               for j in N:
                   if $rand < \beta$ then $x_{LS}(j), x_{LS}(k) = x_{LS}(k), x_{LS}(j)$ where k is a random position in [1,N]
               Record $X_{LS,best}$
Step 5.3:  Evaluation of the new solution
           if $F(X_{LS,best}) < F(X_{worst})$ then replace $X_{worst}$ with $X_{LS,best}$ in HM and update $X_{worst}$
               if $F(X_{LS,best}) < F(X_{best})$ then $X_{best} = X_{LS,best}$
```

4. Experimental campaign and results

A comprehensive set of experimental analyses has been conducted to evaluate the effectiveness of the proposed MHS algorithm in addressing the scheduling problem within the diffusion work area of the semiconductor manufacturing system under investigation. The experimental scenarios have been generated using a real-world data set provided by the semiconductor company. The input parameters for the simulation model have been statistically extracted from one month of production data to ensure a realistic representation of the manufacturing environment. Due to non-disclosure agreement restrictions, input parameters (e.g., number of machines) cannot be revealed and the following information can be provided in generic terms. At each work shift, a number between 300 and 600 jobs are available for being scheduled. The number of cleaning and furnaces operations are in the range [60-90] and [150-250], respectively, while the processing times are in the range [60-120] and [240-1560] min. Batch size is in the range [1-5] jobs at the cleaning stage and [1-8] jobs at the diffusion stage. Two different versions of MHS have been tested, i.e., MHS_{AD}, which incorporates only the adaptive parameter mechanism (see Equation 4), and $MHS_{AD,LS}$, which additionally includes the LS component. The experimental evaluation is structured as follows: in section 4.1, a convergence analysis is performed to determine the computational time required for $MHS_{AD,LS}$ to reach a near-optimal solution. Then, in section 4.2, a comparative analysis is conducted, where MHS_{AD} and $MHS_{AD,LS}$ are compared with HS, SA in terms of mean flowtime. The algorithm parameters have been fine-tuned through a trial-and-error procedure, resulting in the following selected values: HMS=100; ρ_{cr}^{L}=0.9; ρ_{cr}^{U}=0.99; ρ_{ar}^{L}=0.1; ρ_{cr}^{U}=0.3; α^{L}=500; α^{U}=900; β^{L}=0.9; β^{U}=0.1; $evals_{LS}$=1500. ρ_{cr} and ρ_{ar} of HS are set equal to the mean value of [ρ_{cr}^{L}, ρ_{cr}^{U}] and [ρ_{ar}^{L}, ρ_{cr}^{U}], respectively. All algorithms have been implemented in Python®v.3.12.2 on a workstation equipped with an INTEL i9-9900 3.6 GHz 10 core CPU, 32Gb DDR4 2666 MHz RAM and Win 10 PRO OS.

4.1 Convergence analysis

A dedicated experimental study has been conducted to assess the convergence behavior of the proposed $\mathrm{MHS}_{\mathrm{AD,LS}}$ algorithm, which will later be used as the stopping criterion for all the algorithms under comparison. Specifically, three instances have been generated for this analysis. Given the stochastic nature of $\mathrm{MHS}_{\mathrm{AD,LS}}$, each instance has been replicated five times with different random seeds. As a result, a total of 3 × 5 = 15 runs has been performed for this convergence analysis. Figure 2 illustrates the convergence behavior in terms of flowtime as a function of the number of iterations. Due to the stochastic approach, each curve in the figure represents the median value across the five replicates for each instance, while the shaded areas indicate the standard deviation. For this analysis, a stopping criterion of 30,000 iterations has been initially considered. However, it can be observed that after 20,000 iterations, the $\mathrm{MHS}_{\mathrm{AD,LS}}$ does not yield any significant improvement in flowtime. Therefore, 20,000 iterations have been selected as the stopping criterion for the comparison among the metaheuristic algorithms. Figure 2 also refers to the time-to-convergence analysis, showing that approximately 13 minutes of computational time are required to reach 20,000 iterations.



Number of iterations	Average Comp. time (s)
5,000	130
10,000	340
15,000	590
20,000	810
25,000	1005
30,000	1175

Figure 2. Convergence analysis of the $\mathrm{MHS}_{\mathrm{AD,LS}}$ algorithm.

4.2 Results from the experimental comparison

To conduct the experimental comparison between $\mathrm{MHS}_{\mathrm{AD,LS}}$, $\mathrm{MHS}_{\mathrm{AD}}$, HS and SA, 10 instances were generated based on the industrial data described in Section 4. Each instance was replicated 10 times using different random seeds to mitigate the influence of stochastic factors on the final results. Consequently, a total of 10 instances × 10 replicates × 4 algorithms = 400 runs were executed. The four metaheuristic algorithms were compared using the Relative Percentage Deviation (*RPD*), computed as follows:

$$RPD_{i,r,a} = \frac{F_{i,r,a} - F_{i,best}}{F_{i,best}} \% \tag{6}$$

where *i, r, a* represent the instance, replicate, and algorithm, respectively. $F_{i,r,a}$ denotes the flowtime obtained by algorithm *a* in instance *i* and replicate *r*, while $F_{i,best}$ corresponds to the lowest flowtime value among those achieved by the four metaheuristic algorithms across all replicates of instance *i*. The results of the experimental comparison are summarized in Table 1, where, for each instance, the median *RPD* (*MRPD*) value across all replicates is reported. The last row presents the average *MRPD* values for each algorithm along with the N_{best} index, which indicates the number of instances in which an algorithm achieved the best solution. The best results are highlighted in bold. The results in Table 1 highlight the effectiveness of the proposed $\mathrm{MHS}_{\mathrm{AD,LS}}$ algorithm for the scheduling problem in the diffusion work area under study. Specifically, $\mathrm{MHS}_{\mathrm{AD,LS}}$ achieves the lowest average *MRPD* value and finds the best solution in 9 out of 10 instances. This suggests that the combination of adaptive parameter tuning and LS mechanisms effectively enhances the search process, reducing the risk of premature convergence and improving solution quality. To further illustrate the comparative performance of the algorithms, Figure 3 presents a boxplot representation of the *RPD* values. The figure clearly demonstrates that the two proposed algorithms, MHS_{AD} and $\text{MHS}_{\text{AD,LS}}$ significantly outperform the benchmark metaheuristic algorithms from the literature (HS and SA). Notably, $\text{MHS}_{\text{AD,LS}}$ exhibits not only a lower *MRPD* but also a reduced variance compared to the other approaches, indicating a more robust and reliable performance across different problem instances. The observed results suggest that introducing adaptivity in HS parameters (MHS_{AD}) already provides a significant improvement over the standard HS, as seen in the reduction of *MRPD* values. The further enhancement achieved by $\text{MHS}_{\text{AD,LS}}$ confirms the crucial role of LS strategies in effectively navigating the solution space. Finally, non-parametric Mann-Whitney tests were conducted to compare $\text{MHS}_{\text{AD,LS}}$ metaheuristic algorithm with the three alternative algorithms. All non-parametric test provided a *p*-value of 0.00 for each comparison, indicating that the differences between $\text{MHS}_{\text{AD,LS}}$ and the other algorithms are statistically significant. The observed reduction in flowtime provided by the proposed $\text{MHS}_{\text{AD,LS}}$ a reduction of the time spent by jobs in the system. The minimization of average flowtime also implies a reduction in the overall system's work in progress (WIP) and has a significant impact on decreasing the cycle time.

Table 1. Experimental results in terms of MRPDs.

Instance	$\mathbf{MHS_{AD}}$	$\mathbf{MHS_{AD,LS}}$	**HS**	**SA**
1	6.82%	**4.39%**	10.68%	14.74%
2	2.21%	**0.80%**	6.28%	11.88%
3	4.21%	**1.83%**	8.52%	11.84%
4	2.27%	**1.40%**	5.71%	10.48%
5	3.16%	**1.48%**	6.82%	9.64%
6	3.66%	**1.31%**	8.48%	11.54%
7	**2.51%**	2.56%	7.11%	12.18%
8	2.38%	**2.19%**	6.48%	11.07%
9	2.80%	**1.06%**	6.16%	10.25%
10	5.19%	**2.97%**	8.85%	13.62%
Average (N_{best})	*3.52% (1)*	***2.00% (9)***	*7.51% (0)*	*11.73% (0)*

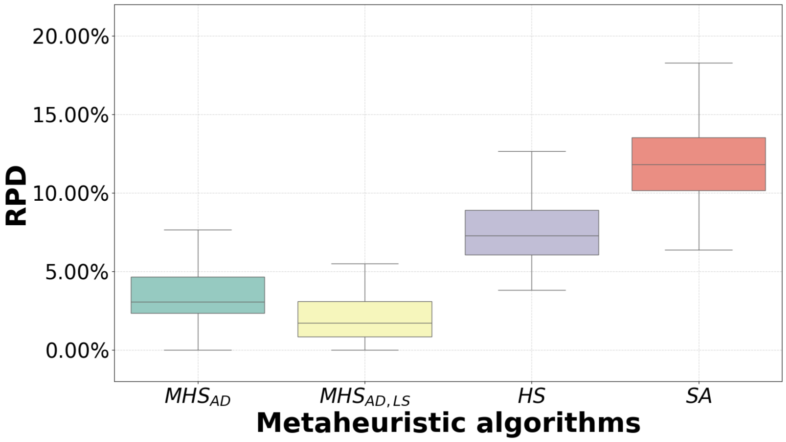


Figure 3. Boxplots to compare metaheuristic algorithms.

5. Conclusion

This paper addressed the scheduling problem in the diffusion work area of a semiconductor manufacturing system, which is modeled as a two-stage flexible flow shop with re-entrant flows and multiple complex production constraints, including machine-dependent batch sizes, operation-and machine-dependent processing times, waiting time constraints, preloading operations and other constraints. To tackle this problem, a MHS metaheuristic algorithm was proposed, incorporating adaptive parameter control and an embedded LS procedure to enhance the effectiveness of the original algorithm for the problem at hand. Two versions of MHS were tested: MHS_{AD}, which includes only the adaptive parameter mechanism, and $MHS_{AD,LS}$, which also integrates the LS procedure. The experimental results demonstrated that the introduction of adaptive parameters improves the exploration and exploitation capabilities of the algorithm, leading to better solutions compared to the standard HS. Moreover, the inclusion of LS further refines solution quality, preventing the algorithm from getting trapped in a local optimum. From an industrial perspective, the proposed MHS may offer a valuable metaheuristic framework for semiconductor manufacturers facing the complex scheduling challenges of the diffusion work area, contributing to a reduction in terms of average wafer flowtime of the system.

One limitation of the proposed approach lies on its computational time. In a real-world production environment, it is crucial to make scheduling decisions rapidly at the beginning of each shift. In future works, the algorithm will be further optimized with the aim of being embedded into a digital twin architecture. Another limitation is that the current model does not explicitly account for minimizing the number of reworked or scrapped jobs per shift. Future developments will focus on considering a multi-objective function to minimize the number of jobs that do not respect the waiting time limit. As for future research, if the results from the real-world case study confirm those observed in the experimental analysis, the MHS algorithm can be tested for other work areas, such as photolithography or etching. Furthermore, alternative AI-based scheduling method, such as Deep Reinforcement Learning (DRL), can be developed and compared with MHS for the scheduling problem of the diffusion work area. Finally, the proposed MHS can be compared with new metaheuristic algorithms to maximize the system’s throughput measure.

Acknowledgment

The work presented in this paper has received funding from the Chips Joint Undertaking (JU) under grant agreement No 101097296. The JU receives support from the European Union’s Horizon EU research and innovation programs and Austria, Belgium, Denmark, Finland, France, Germany, Greece, Hungary, Israel, Italy, Netherlands, Romania, Sweden, Switzerland and Turkey.

References

[1] L. Mönch, J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, O. Rose, A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations, J. Sched. 14 (2011) 583-599. https://doi.org/10.1007/s10951-010-0222-9

[2] C. Yugma, S. Dauzère-Pérès, C. Artigues, A. Derreumaux, O. Sibille, A batching and scheduling algorithm for the diffusion area in semiconductor manufacturing, Int. J. Prod. Res. 50(8) (2012) 2118-2132. https://doi.org/10.1080/00207543.2011.575090

[3] S. Knopp, S. Dauzère-Pérès, C. Yugma, A batch-oblivious approach for complex job-shop scheduling problems, Eur. J. Oper. Res 263(1) (2017) 50-61. https://doi.org/10.1016/j.ejor.2017.04.050

[4] Y. Tan, L. Mönch, J. W. Fowler, A hybrid scheduling approach for a two-stage flexible flow shop with batch processing machines, J. Sched. 21 (2018) 209-226. https://doi.org/10.1007/s10951-017-0530-4

[5] K. Tamssaouet, S. Dauzère-Pérès, S. Knopp, A. Bitar, C. Yugma, Multiobjective optimization for complex flexible job-shop scheduling problems, Eur. J. Oper. Res. 296(1) (2022) 87-100. https://doi.org/10.1016/j.ejor.2021.03.069

[6] H. Zhong, M. Liu, L. Bao, A job-priority based soft scheduling approach for uncertain work area scheduling in Semiconductor Manufacturing, Int. J. Prod. Res. 60(16) (2022) 5012-5028. https://doi.org/10.1080/00207543.2021.1948134

[7] K. Wu, E. Huang, M. Wang, M. Zheng, Job scheduling of diffusion furnaces in semiconductor fabrication facilities, Eur. J. Oper. Res. 301(1) (2022) 141-152. https://doi.org/10.1016/j.ejor.2021.09.044

[8] L. Chen, Y. Qiao, N. Wu, S. Zhang, J. Li, B. Liu (2023, August). Scheduling Analysis of Diffusion Area in Semiconductor Manufacturing. In 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE) (pp. 1-6). IEEE. https://doi.org/10.1109/CASE56687.2023.10260420

[9] F. Da Ros, L. Di Gaspero, M. L. Lackner, N. Musliu, F. Winter (2025). Multi-neighborhood simulated annealing for the oven scheduling problem. Comput. Ind. Eng. 106999. https://doi.org/10.1016/j.cor.2025.106999