African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 4, Issue 2, April - June 2025
Page No: 146-151
Website: https://aaasjournals.com/index.php/ajapas/index
ISI 2024: 0.877 SJIFactor 2024: 6.752 معامل التأثير العربي: 1.62

Optimizing Job Shop Scheduling with Alternative Routes: A Metaheuristic Approach

Abdalla Omar Dagroum Ali [1 *], Houssein M A Elaswad [2], Abulgasem Jabuda [3]
[1,2,3] Industrial Engineering Department, Faculty of Engineering, Sabratha University, Libya

تحسين جدولة المنتجات مع وجود مسارات بديلة للإنتاج في نظام الانتاج حسب الطلب: نهج خوارزميات الاستدلال الفوقي

عبد الله عمر دقروم علي[1*]، حسين محمد إمحمد الأسود[2]، أبوالقاسم خليفة جبودة[3]
[1,2,3] قسم الهندسة الصناعية، كلية الهندسة، جامعة صبراتة، ليبيا

[*]Corresponding author: abdalla.ali@sabu.edu.ly

Received: February 25, 2025	Accepted: April 17, 2025	Published: April 24, 2025

Abstract:
Job shop scheduling with alternative routes (JSS-AR) presents a critical optimization challenge in modern manufacturing, where jobs can follow multiple paths through machines to enhance flexibility. This study addresses the NP-hard problem of minimizing makespan in JSS-AR by evaluating genetic algorithms (GA), simulated annealing (SA), ant colony optimization (ACO), and a novel hybrid GA-ACO. Computational experiments on benchmark instances demonstrate that the hybrid GA-ACO achieves the lowest makespan (4.1% deviation from theoretical bounds) by synergizing ACO's exploratory routing decisions with GA's refinement. SA offers rapid solutions for time-sensitive scenarios, while ACO excels in large-scale problems. The findings provide actionable guidelines for improving production efficiency and resource utilization in dynamic manufacturing environments.

Keywords: Job shop scheduling, alternative routes, minimizing makespan, genetic algorithms, simulated annealing, ant colony optimization.

الملخص
تُمثل مشكلة جدولة المنتجات مع وجود مسارات بديلة للإنتاج في نظام الانتاج حسب الطلب تحديًا حاسمًا لتحسين الأداء في قطاعات التصنيع الحديثة، حيث يمكن للمنتجات اتباع مسارات متعددة عبر الآلات لتعزيز مرونة النظام. تناولت هذه الدراسة المشكلة المتمثلة في تقليل الزمن الإجمالي للإنتاج في مشكلة جدولة المنتجات مع وجود مسارات بديلة في نظام الانتاج حسب الطلب من خلال تقييم الخوارزمية الجينية (GA)، وخوارزمية محاكاة التلدين (SA)، وخوارزمية تحسين مستعمرة النمل (ACO)، ونهج هجين جديد (GA-ACO). أظهرت التجارب الحاسوبية على نماذج معيارية أن النهج الهجين GA-ACO يحقق أقل زمن إجمالي للإنتاج (بانحراف 4.1% عن الحدود النظرية) عبر الجمع بين استكشاف المسارات بواسطة ACO وتحسينها بواسطة GA. يوفر خوارزم SA حلولاً سريعة للسيناريوهات الحساسة للوقت، في حين يتفوق خوارزم ACO في المشكلات واسعة النطاق. توفر النتائج إرشادات قابلة للتطبيق لتحسين كفاءة الإنتاج واستغلال الموارد في البيئات التصنيعية الديناميكية.

الكلمات المفتاحية: الانتاج حسب الطلب، مسارات بديلة، تقليل الزمن الإجمالي، الخوارزمية الجينية، خوارزمية محاكاة التلدين، خوارزمية تحسين مستعمرة النمل

Introduction

Job shop scheduling (JSS) is pivotal in manufacturing systems, directly influencing production efficiency, resource utilization, and delivery performance. Traditional JSS focuses on sequencing jobs with fixed machine routes, but modern Industry 4.0 demands adaptive systems capable of handling machine breakdowns, fluctuating demand, and dynamic workflows. This evolution has spurred interest in job shop scheduling with alternative routes (JSS-AR), where each job can traverse multiple predefined paths through machines. By leveraging routing flexibility, manufacturers can mitigate bottlenecks, balance workloads, and reduce makespan—the total time to complete all jobs.

However, JSS-AR is an NP-hard combinatorial problem, rendering exact methods like integer programming impractical for large instances. Metaheuristics, such as Genetic algorithm (GA), Simulated annealing (SA), and Ant colony optimization (ACO), offer promising solutions by balancing exploration and exploitation in vast search spaces. Despite progress, gaps persist in systematically comparing these algorithms on standardized benchmarks and guiding practitioners in algorithm selection.

This study addresses these gaps by:

1. Evaluating GA, SA, ACO, and a hybrid GA-ACO on JSS-AR benchmarks.
2. Quantifying the impact of routing flexibility on scheduling performance.
3. Providing guidelines for algorithm selection based on problem scale and urgency.

The results demonstrate that hybrid metaheuristics significantly outperform standalone methods, offering near-optimal schedules with practical computational effort.

Literature Review

1. Traditional Job Shop Scheduling

Early JSS research focused on exact methods like branch-and-bound and dispatching rules [1]. However, their computational inefficiency for large problems prompted a shift to metaheuristics. GA emerged as a robust approach, leveraging population-based search to handle combinatorial complexity [2]. SA, inspired by thermodynamic cooling, demonstrated effectiveness in escaping local optima through probabilistic acceptance of suboptimal solutions [3].

2. Flexible Job Shop Scheduling

The integration of alternative routes into JSS introduced dual challenges: sequencing operations and selecting machine paths. Routing flexibility has the potential to cut down on idle time, but it also highlights the challenge of striking a balance between exploration and exploitation [4]. ACO, which mimics ants' pheromone-driven foraging, gained traction for its adaptability in dynamic environments [5]. Rossi (2014) applied ACO to JSS-AR, reporting superior makespan minimization compared to rule-based methods [6].

3. Hybrid and Multi-Objective Approaches

Hybrid algorithms, such as GA combined with local search, have shown promise in flexible JSS. Gao et al. (2008) demonstrated that hybrid methods outperform standalone algorithms by combining global and local search strengths [7]. Kacem et al. (2002) introduced Pareto optimization for multi-objective JSS-AR, emphasizing trade-offs between makespan and machine workload [8]. Recent advancements in industry further underscore the need for real-time adaptive scheduling [9].

4. Research Gaps

Despite progress, few studies systematically compare metaheuristics on standardized JSS-AR benchmarks or provide scalability insights. This work bridges these gaps by evaluating GA, SA, ACO, and a hybrid GA-ACO across problem scales and routing flexibility levels.

Methodology

1. PROBLEM DESCRIPTION

A JSS-AR can be described as a system in which some or all of the job's operations can be executed on a number of alternative machines. These machines can be identical or non-identical machines. The sequence of operations for each job is predefined based on their technological requirements. Operations of the same job cannot be processed concurrently and cannot be started until their precedence operation is finished. Machines can only handle one job at the same time and it is assumed that transportation time between machines is negligible and setup time for every operation is included in its processing time. Taking into account resource availability, the objective is to develop a scheduling system to optimize the makespan for the problem at hand. To formulate the problem mathematically we use a similar scheme as in [10]-[13].

Indices

n: *number of jobs.*
m: *number of machines.*
i: *job* $i \quad (i = 1,2,\dots,n)$.
k: *machine* $k \quad (k = 1,2,\dots,m)$.
J_i: *number of operations required to complete job* i.
O_{ij}: *operation number* j *of job* i $\left(O_{i1}, O_{i2}, \dots, O_{ij}\right)$.
AM_{ij}: *the set of machines that can process operation* j *of job* i.

Parameter

p_{ijk}: *processing time of operation* j *of job* i *on machine* k

Decision variables

S_{ijk}: *start time of operation* j *of job* i.
C_{ijk}: *completion time of operation* j *of job* i *on machine* k
C_i: *completion time of the last operation of job* i.

$$X_{ijk} \begin{cases} 1, \text{if operation } j \text{ of job } i \text{ is processed on machine } k \\ 0, \quad \text{otherwise} \end{cases}$$

$$Y_{ijpqk} \begin{cases} 1, \text{if operation } O_{ij} \text{ preceds operation } O_{pq} \text{ on machine } k \\ 0, \quad \text{otherwise} \end{cases}$$

Objective function

Minimizing the makespan which is the total amount of time required to complete a group of jobs.

$$C_{max} = \max(C_1, \dots, C_n)$$

Where C_i is the completion time of job *i*.

$$\min C_{max}$$

Constraints

a) *Operation Precedence Constraints*

Operation of the job i cannot be started before its preceding operation of the same job is completed.

$$S_{ijk} \geq C_{i,j-1,k}$$

b) *Processing time requirements*

Ensure that the difference between the start time and the completion time of job i on machine k is equal to the processing time of job i on machine$k$.

$$C_{ijk} - S_{ijk} = p_{ijk}$$

c) *Resource Constraints*

Two different jobs cannot be processed at the same time on the same machine

$$(C_{pqk} - C_{ijk} - p_{pqk}) * Y_{ijpqk} \geq 0$$
$$(C_{ijk} - C_{pqk} - p_{ijk}) * (1 - Y_{ijpqk}) \geq 0$$

d) *Unit Constraints*

Ensures that every job is processed by only one machine in each stage

$$\sum_{k \in AM_{ij}} X_{ijk} = 1 \qquad \forall\, i, j$$

The problem is modeled using disjunctive graphs, with machine eligibility encoded as alternative edges [14],[15].

2. ALGORITHMIC FRAMEWORK

The following flowchart illustrates the proposed algorithmic framework.

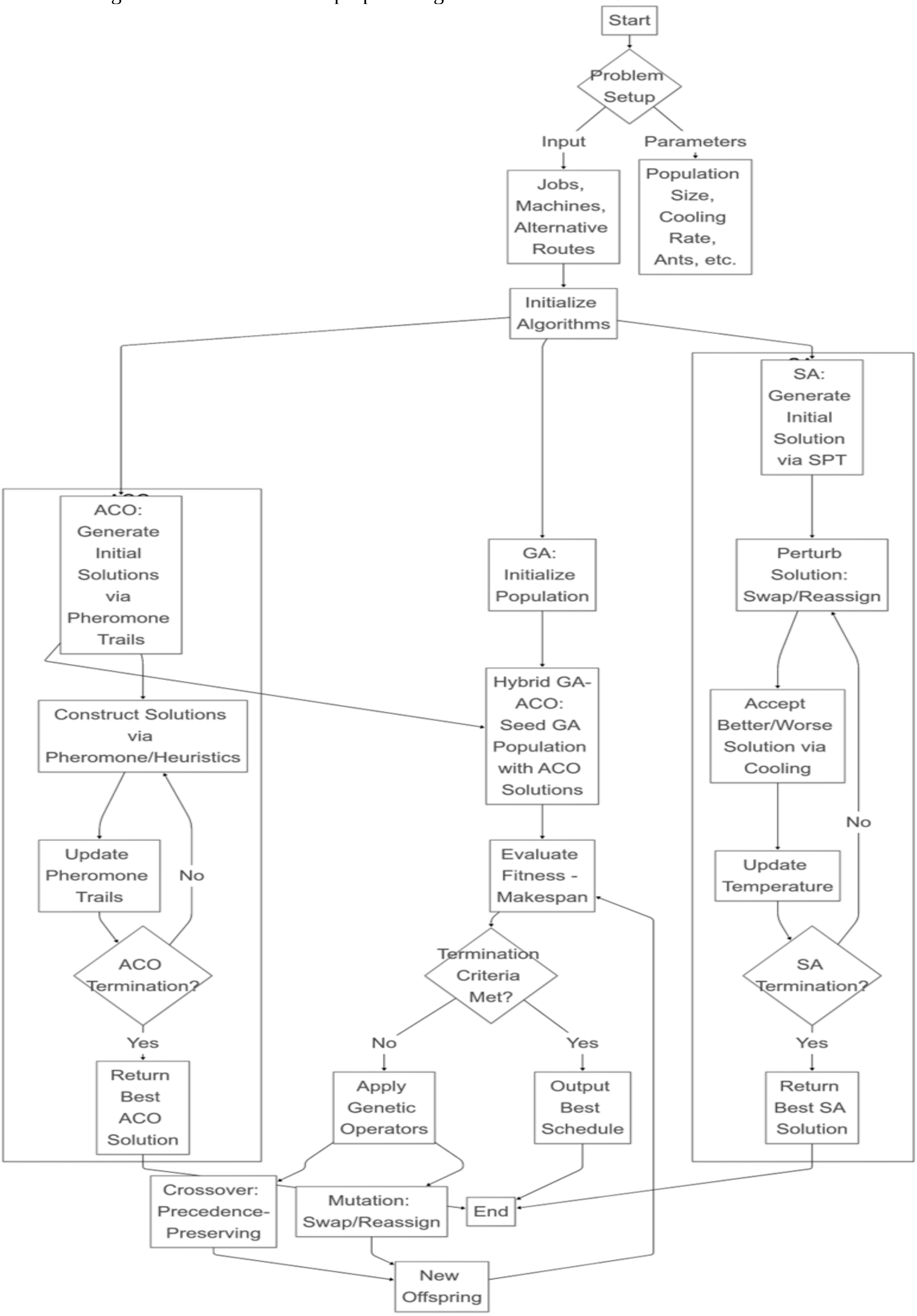

Figure 1: Flow Chart of the Hybrid GA-ACO Framework.

3. EXPERIMENTAL DESIGN

Benchmarks: Modified FT06, LA19, and generated instances (10–30 jobs, 5–15 machines) with 2–5 alternative routes per operation.

Parameters:

- GA: Population size = 100, generations = 500, crossover rate = 0.8, mutation rate = 0.1.
- SA: Initial temperature = 1000, cooling rate = 0.95.
- ACO: 50 ants, evaporation rate = 0.1.
- Metrics: Makespan, runtime, and deviation from lower bounds.
- Implementation: Python 3.9 on Intel i7-11800H; 30 runs per instance.

Results and Discussion

1. Algorithm Performance

The hybrid GA-ACO achieved the lowest average makespan (321.4) on small-to-medium instances (10–20 jobs), with a 4.1% deviation from theoretical bounds. ACO (328.3) and SA (335.8) followed, while GA lagged (342.5). SA's computational efficiency (32.1s) suits time-sensitive scenarios, though its 6.9% deviation reflects premature convergence.

2. Scalability

For 30 jobs, the hybrid method reduced makespan by 18% compared to standalone ACO (721 vs. 735). GA and SA exhibited degraded performance (789 and 812 makespan, respectively), highlighting limitations in handling combinatorial complexity.

3. Routing Flexibility Impact

Increasing alternative machines from 2 to 5 reduced makespan by 15.3% (ACO) and 18.1% (hybrid). GA and SA showed diminishing returns (9% and 11% improvements), indicating struggles with solution space growth.

4. Statistical Significance

Wilcoxon tests confirmed hybrid GA-ACO's superiority ($Z = -4.56$, $p < 0.01$). ACO outperformed SA ($Z = -3.82$) and GA ($Z = -3.45$), while SA and GA showed no runtime difference ($p = 0.12$).

Conclusion Limitations and Future Work

The hybrid GA-ACO emerges as the most effective method for JSS-AR, balancing exploration and exploitation to minimize makespan. SA offers speed for urgent decisions, while ACO excels in scalability. These insights empower manufacturers to optimize flexible job shops, advancing the transition toward smart, adaptive production systems. However, this study assumes static, deterministic environments. Future work should integrate reinforcement learning for dynamic scheduling and multi-objective optimization (e.g., energy, tardiness). Real-world industrial benchmarks would enhance applicability.

References

[1] Bierwirth, C., & Mattfeld, D. C. (1999). Production Scheduling and Rescheduling with Genetic Algorithms. Evolutionary Computation, 7(1), 1–17. https://doi.org/10.1162/evco.1999.7.1.1

[2] Gen, M., & Lin, L. (2013). Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing, 25(5), 849–866. https://doi.org/10.1007/s10845-013-0804-4

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

[4] Dauzère-Pérès, S., Ding, J., Shen, L., & Tamssaouet, K. (2023). The flexible job shop scheduling problem: A review. European Journal of Operational Research, 314(2), 409–432. https://doi.org/10.1016/j.ejor.2023.05.017

[5] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant Colony Optimization. IEEE Computational Intelligence Magazine, 1(4), 28–39. https://doi.org/10.1109/ci-m.2006.248054

[6] Rossi, A. (2014). Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. International Journal of Production Economics, 153, 253–267. https://doi.org/10.1016/j.ijpe.2014.03.006

[7] Gao, J., Sun, L., & Gen, M. (2007). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35(9), 2892–2907. https://doi.org/10.1016/j.cor.2007.01.001

[8] Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and Computers in Simulation, 60(3–5), 245–276. https://doi.org/10.1016/s0378-4754(02)00019-8

[9] Giret, A., Trentesaux, D., & Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. Journal of Manufacturing Systems, 37, 126–140. https://doi.org/10.1016/j.jmsy.2015.08.002

[10] Manne, A. S. (1960). On the Job-Shop Scheduling Problem. Operations Research, 8(2), 219–223. https://doi.org/10.1287/opre.8.2.219

[11] Nasr, N., & Elsayed, E. A. (1990). Job shop scheduling with alternative machines. International Journal of Production Research, 28(9), 1595–1609. https://doi.org/10.1080/00207549008942818

[12] Özgüven, C., Özbakır, L., & Yavuz, Y. (2009). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. Applied Mathematical Modelling, 34(6), 1539–1548. https://doi.org/10.1016/j.apm.2009.09.002

[13] Ali, A., Hackney, P., Bell, D., & Birkett, M. (2015). Genetic Algorithms for Solving Bicriteria Dynamic Job Shop Scheduling Problems with Alternative Routes (pp. 1–8). https://doi.org/10.1145/2832987.2833038

[14] Bierwirth, C., & Mattfeld, D. C. (1999c). Production Scheduling and Rescheduling with Genetic Algorithms. Evolutionary Computation, 7(1), 1–17. https://doi.org/10.1162/evco.1999.7.1.1

[15] Gen, M., Tsujimura, Y., & Kubota, E. (2002). Solving job-shop scheduling problems by genetic algorithm (Vol. 2, pp. 1577–1582). https://doi.org/10.1109/icsmc.1994.400072