mathematics

MDPI

Article

A Heuristic Guided Genetic Algorithm Applied to Dual Resource Job Shop Scheduling

Ricardo Magalhães *, João M. C. Sousa and Susana M. Vieira

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal; jmsousa@tecnico.ulisboa.pt (J.M.C.S.); susana.vieira@tecnico.ulisboa.pt (S.M.V.)
* Correspondence: ricardo.m.a.magalhaes@tecnico.ulisboa.pt

Abstract

This paper introduces the Knowledge-Guided Genetic Algorithm (KGGA), a hybrid metaheuristic that reimagines crossover as a form of genetic engineering rather than random recombination. By embedding knowledge-guided exploitation principles directly into the crossover operator, KGGA selectively amplifies high-quality genetic material, intensifying the search around promising regions of the solution space. Experimental results on a large scale DRC-FJSSP benchmark show that KGGA outperforms state-of-the-art alternatives—including the Classic Genetic Algorithm (GA), Knowledge-Guided Fruit Fly Optimization Algorithm (KGFOA), and Hybrid Artificial Bee Colony Algorithm (HABCA)—consistently achieving superior solution quality.

Keywords: scheduling; job shop; dual resource; genetic algorithm; genetic engineering; knowledge-guided crossover

MSC: 90B30; 90B90; 90C90

check for updates

Academic Editors: Chia-Ding Hou and Rung-Hung Su

Received: 14 April 2025
Revised: 3 August 2025
Accepted: 17 September 2025
Published: 29 September 2025

Citation: Magalhães, R.; Sousa, J.M.C.; Vieira, S.M. A Heuristic Guided Genetic Algorithm Applied to Dual Resource Job Shop Scheduling. *Mathematics* **2025**, *13*, 3116. https://doi.org/10.3390/math13193116

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Advanced manufacturing refers to the integration of cutting-edge technologies, such as robotics [1], artificial intelligence [2,3], and the Internet of Things (IoT) [4,5], into the manufacturing process. This approach is part of industry 4.0 transformation and not only enhances the efficiency of production but also leads to better quality control [6], increased flexibility, and the ability to produce customized products [7] at a lower cost. The concept of advanced manufacturing is becoming increasingly relevant in today's global market, where companies must remain competitive by staying ahead of the curve in terms of technological innovation.

One of the key areas that advanced manufacturing has transformed is production scheduling [8–10]. With the use of advanced analytics and algorithms, manufacturers can optimize their production schedules to achieve maximum efficiency and reduce downtime [11–13]. This technology enables manufacturers to determine the most efficient production schedule based on factors such as machine capacity, material availability, and customer demand. This, in turn, leads to improved productivity, reduced waste, and cost savings.

To address the challenges of production scheduling in advanced manufacturing, researchers have turned to metaheuristics [14–16], a class of algorithms that are designed to find high-quality solutions to complex optimization problems. Metaheuristics are particularly well-suited to manufacturing environments because they can quickly generate

near-optimal solutions to complex scheduling problems, such as the Job Shop Scheduling Problems (JSSP) [17], without requiring detailed knowledge of the underlying processes [18]. Examples of metaheuristic techniques used in manufacturing include genetic algorithms [19], simulated annealing [20], and ant colony optimization [21]. These algorithms work by generating a set of candidate solutions and iteratively refining them until an optimal solution is found or a given number of iterations is achieved.

The JSSP is a widely recognized production scheduling problem [22]. As proven in [23], it is a NP-hard optimization problem that aims to find the best sequence for processing a set of jobs using a fixed set of machines. Jobs consist of operations that must be processed in a specific order. As stated in [24], the Flexible Job Shop Scheduling Problem (FJSSP) is an extension of the original JSSP where machines are no longer fixed, but rather a set of machines are available to be chosen for each operation, leading to increased flexibility and complexity in scheduling.

As mentioned by [25], a production system can be dual resource constrained (DRC), as when restricted by both worker and machine capacity. This is an effort to more effectively integrate the actual dynamics and constraints of the manufacturing process [26]. In the pharmaceutical industry, for example, tasks are typically carried out by both machines and human analysts.

2. Related Work

This section reviews existing methods that have been developed to solve Dual Resource Constrained Flexible Job Shop Scheduling Problems (DRC-FJSSP), focusing on metaheuristics and knowledge-based strategies. First, classical metaheuristic algorithms, as e.g., Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Tabu Search (TS) are examined, particularly regarding their ability to address dual resource constraints. Then, hybrid methods are discussed, followed by an overview of the only known application of the Knowledge-Guided Fruit Fly Optimization Algorithm (KGFOA) to this domain. The analysis concludes by highlighting the remaining challenges and research gaps that motivate the novel approach proposed in this work.

2.1. Classical Metaheuristics

Metaheuristic algorithms are widely employed for solving DRC-FJSSP due to their ability to handle combinatorial complexity. Among them, PSO has proven effective, especially when adapted for discrete and multi-objective variants of the problem. A study by [27] proposed an enhanced PSO (EPSO) incorporating a particle life cycle and discrete position updates, improving solution diversity and avoiding premature convergence in flexible job-shop scheduling. Similarly, Ref. [28] incorporated workers' boredom as a dynamic human factor in the scheduling process, presenting a two-stage multi-objective PSO that adapts neighborhood size and uses boundary exploration to improve solution diversity and realism. These studies highlight PSO's strength in exploration and its adaptability to problem-specific contexts. However, PSO still suffers from premature convergence and often requires external mechanisms to maintain feasibility.

TS has also been successfully applied to DRC-FJSSP, particularly in multi-agent configurations. Both [29,30] proposed Multi-Start Tabu Search agent-based models (MuSTAM), where multiple Tabu agents explore the solution space in parallel and cooperate to improve search efficiency. These models achieved competitive results in makespan minimization and demonstrated good scalability. Despite its intensification capabilities and strong local search, TS generally requires careful tuning and may struggle with maintaining population diversity, which can lead to stagnation.

Genetic Algorithms (GAs) are among the most frequently used methods for DRC-FJSSP. Recent GA variants have introduced improvements in representation, initialization, and local search integration. For instance, Ref. [31] developed an improved GA with a three-vector chromosome structure and heuristic rules for both resource selection and neighborhood search, achieving significant gains over standard benchmarks. A quantum-inspired GA presented in [32] introduced quantum mutation strategies and adaptive mechanisms to handle differences in workers' operating times and avoid premature convergence. Additional contributions, such as [33], integrated worker proficiency and preparation times, while [34] proposed a dynamic multi-objective GA addressing both completion time and energy consumption. These approaches underscore the flexibility and extensibility of GAs for complex scheduling problems. Nonetheless, they also reveal ongoing limitations, such as parameter sensitivity and the need for knowledge-based guidance to improve the efficacy of crossover and mutation operations.

Overall, while classical metaheuristics offer a powerful foundation, their performance in DRC-FJSSP often depends heavily on customization, parameter tuning, and supplementary repair mechanisms. These weaknesses have motivated the development of hybrid and knowledge-driven strategies.

2.2. Hybrid Methods

Hybrid metaheuristics combine the strengths of different algorithms to improve search efficiency and robustness, especially in problems like DRC-FJSSP where resource constraints and scheduling flexibility interact in complex ways. For example, Ref. [35] introduced a hybrid memetic algorithm for mixed production scheduling, incorporating a four-layer chromosome structure and variable neighborhood search (VNS) to balance global and local exploration. Similarly, Ref. [36] proposed a cooperative evolutionary algorithm integrated with constraint programming (CEAM-CP) to handle sub-problems of resource assignment and sequencing through multi-population co-evolution.

Worker flexibility and learning effects have also been incorporated into recent models. Ref. [37] developed a two-level optimization framework combining rule-based worker allocation with hybrid genetic simulated annealing for scheduling. Ref. [38] proposed a hybrid genetic algorithm with variable neighborhood search to account for learning effects over time. These studies show that hybridization can enhance local exploitation while maintaining global diversity, often outperforming single-method approaches.

Other hybrid strategies include PSO enhanced with simulated annealing [39,40], artificial bee colony algorithms for worker-flexible environments [41], and branch population GAs with elite selection and compressed time windows [42]. Despite their diverse formulations, these methods share a common goal: to overcome the shortcomings of classical metaheuristics by embedding structural problem knowledge, adaptive operators, or cooperative mechanisms.

2.3. KGFOA in DRC-FJSSP

The only known work to apply the Knowledge-Guided Fruit Fly Optimization Algorithm (KGFOA) to DRC-FJSSP is [43]. Unlike traditional Fruit Fly Optimization Algorithm, which relies on random smell-based exploration, KGFOA introduces a knowledge-guided search phase to improve convergence and solution quality.

In their formulation, KGFOA employs a two-layer permutation-based encoding scheme to represent job sequences and dual-resource assignments. The algorithm alternates between smell-based search for exploration and knowledge-guided operators for local refinement. These operators use heuristic rules to adjust operation sequences and resource assignments intelligently, thereby improving feasibility and efficiency. The authors

demonstrated that KGFOA outperforms traditional FOA and other metaheuristics like VNS in terms of makespan and convergence reliability.

Despite its promising results, this study remains an isolated case in the literature. No significant extensions or comparative studies involving KGFOA have been conducted since its introduction, leaving considerable room for further exploration and integration with more advanced evolutionary mechanisms.

2.4. Research Gaps and Opportunities

From the reviewed literature, several patterns emerge. While PSO, TS, and GA variants have achieved strong results in solving DRC-FJSSP, they still face challenges related to scalability, premature convergence, and the need for heavy customization or domain-specific repairs. Hybrid methods have attempted to address these weaknesses, often by embedding heuristic knowledge, co-evolutionary processes, or local search modules. However, many of these methods still rely on traditional crossover and mutation schemes that do not actively integrate domain knowledge into their core operators.

The KGFOA stands out for its direct use of knowledge-guided search, but it lacks structural crossover mechanisms and has not been extended beyond its initial formulation. This creates a valuable opportunity: to merge the population-level diversity of GAs with the domain-awareness and local precision of KGFOA, particularly through knowledge-guided crossover operators that act more like genetic engineering than random recombination.

The proposed approach in this work addresses this gap by restructuring the KGFOA search strategy within a GA framework - applying knowledge-guided principles directly to crossover and exploitation stages, and offering a novel paradigm for solving DRC-FJSSP with improved convergence, adaptability, and solution quality.

The structure of the paper is as follows: Section 3 introduces the numerical model of the DRC-FJSSP, followed by the description of the proposed algorithm in Section 4. Section 5 reports and analyzes the results, while Section 6 summarizes the conclusions and outlines directions for future work.

3. Mathematical Formulation

This paper follows the mathematical model proposed in [44]. The DRC-FJSSP problem has the total of a operations to be executed. These are grouped into jobs J_i in the range $J = \{J_1, \ldots, J_n\}$, which must be processed on a set of machines M_k in the set $M = \{M_1, \ldots, M_m\}$. The machines are operated by a workforce of workers w_l belonging to the set $W = \{w_1, \ldots, w_w\}$. Each job J_i consists of a defined sequence of o_i operations, $J_i = \{O_{i1}, \ldots, O_{ij}, \ldots, O_{io_i}\}$. The processing time p_{ij} is the one of the operation O_{ij}. Every operation c is restricted to a subset of eligible machine–worker pairs. The index c is computed as follows:

$$c = j + \sum_{d=1}^{i-1} o_d \tag{1}$$

A worker–machine pair is considered eligible if the machine can perform the operation and the worker is capable of operating that machine. We define a matrix E_{bc}, where the element (b, c) equals 1 if pair b can process operation c, and 0 otherwise, and $c \in \{1, \ldots, a\}$. The term b is computed as follows:

$$b = (k-1)w + l \tag{2}$$

Further, each machine handles one operation at a time; once started, an operation must finish without interruption (non-preemptive scheduling). It is assumed that the workers, jobs and machines are all available at time $t = 0$. The goal is to minimize the makespan C,

which depends on assigning feasible worker–machine pairs to operations and scheduling their sequence appropriately.

Let s_{ij} denote the time when the operation O_{ij} starts, and r_{kl} the time when the machine M_k operated by worker w_l is ready. The DRC-FJSSP can thus be formulated as follows (where N is sufficiently large number):

$$\min \ C \tag{3}$$

Subject to:

$$C = \max_{i,j}(s_{ij} + p_{ij}),\ J_i \in J,\ j = 1, 2, \ldots, o_i - 1 \tag{4}$$

$$s_{i(j+1)} \geq s_{ij} + \sum_{k}\sum_{l} p_{ij}\eta_{ijkl},\ J_i \in J, j = 1, 2, \ldots, o_i - 1,\ \sum_{c=1}^{a} E_{bc} > 0 \tag{5}$$

$$s_{i'j'} + (1 - \zeta_{ijm-i'j'm})N \geq s_{ij} + \sum_{l} p_{ij}\eta_{ijkl},$$

$$J_i \in J,\ j = 1, 2, \ldots, o_i,\ \sum_{c=1}^{a} E_{bc} > 0 \tag{6}$$

$$r_{k'l} + (1 - \xi_{kl-k'l})N \geq r_{kl},\ w_l \in W,\ \sum_{c=1}^{a} E_{bc} > 0,\ \sum_{c=1}^{a} E_{b'c} > 0 \tag{7}$$

$$r_{kl} + (1 - \eta_{ijkl})N \leq s_{ij},\ J_i \in J, j = 1, 2, \ldots, o_i,\ \sum_{c=1}^{a} E_{bc} > 0 \tag{8}$$

$$\sum_{k}\sum_{l} \eta_{ijkl} = 1,\ J_i \in J,\ j = 1, 2, \ldots, o_i,\ \sum_{c=1}^{a} E_{bc} > 0 \tag{9}$$

where

$$\eta_{ijkl} = \begin{cases} 1, & \text{when } O_{i,j} \text{ is processed on } M_k \text{ operated by } w_l \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$\zeta_{ijk-i'j'k} = \begin{cases} 1, & \text{when } O_{i,j} \text{ is processed before } O_{i'j'} \text{ on } M_k \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$\xi_{kl-k'l} = \begin{cases} 1, & \text{if } M_k \text{ is operated before } M_{k'} \text{ by } w_l \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Equation (4) defines the makespan. Precedence relations are ensured by Equation (5). Machine capacity in (6) and worker capacity in (7) guarantee that no resource is overloaded. Equation (8) enforces that resources are available before an operation starts, and (9) ensures each operation is assigned to exactly one available worker–machine pair.

4. Proposed Knowledge-Guided Genetic Algorithm (KGGA)

The KGGA combines knowledge-guided crossover operations to enhance exploitation with genetic mutation mechanisms that preserve diversity and support exploration. In the crossover phase, each elite chromosome generates S novel child chromosomes, forming an offspring subset. Subsequently, each child chromosome is subjected to one mutation. The encoding vectors for schedule solutions proposed in [43] are also adopted in the KGGA.

The crossover operation combines elite chromosome sequences with experimental chromosome sequences drawn from a knowledge base, which encapsulates knowledge acquired from elite chromosomes on resource allocation and operation sequencing.

The experimental probability $P_{(O_{i,j},r)}(g)$ denotes the chance of assigning the r-th resource combination to operation $O_{i,j}$ during generation g. Experimental operation se-

quences are extracted from the NF top-performing chromosomes of the current population, expressed as $\{\pi_1, \pi_2, \ldots, \pi_{NF}\}$. For a scenario with m machines and k workers, resource pairs are indexed such that the combination of machine M_a operated by worker w_b corresponds to resource index $(a-1)w+b$.

Two distinct procedures are employed for generating the offspring: the Resource Assignment Search (RAS) and the Operation Sequence Search (OSS). The RAS involves the knowledge-guided reallocation of resources for each operation, with the probabilities being adjusted based on the sampling of the experimental possibility of resources. The experimental possibility is initialized using the following approach:

$$P_{O_{i,j}r}(0) = \begin{cases} \dfrac{1}{FR}, & \text{if the r-th resource pair} \\ & \text{is eligible for } O_{i,j} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where FR denotes the number of feasible resource combinations that can be assigned to operation $O_{i,j}$.

The possibility undergoes updates at every generation based on the resource allocation knowledge derived from the NF top-performing chromosomes. The updating process is outlined below:

$$P^{temp}_{O_{i,j}r}(g) = (1-\alpha)P_{O_{i,j}r}(g-1) + \frac{\alpha}{NF}\sum_{s=1}^{NF} I^S_{O_{i,j}r} \tag{14}$$

$$P_{O_{i,j}r}(g) = \frac{P^{temp}_{O_{i,j}r}(g)}{\sum_{r=1}^{R} P^{temp}_{O_{i,j}r}(g)} \tag{15}$$

where $I^S_{O_{i,j}r}$ denotes the expertise regarding resource allocation obtained from the s-th selected chromosome.

$$I^S_{O_{i,j}r} = \begin{cases} 1, & \text{if } O_{i,j} \text{ is allocated to the r-th resource pair} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The OSS is responsible for adjusting the sequence of operations by emulating an example taken from a randomly selected experimental operation sequence within the knowledge base. This emulation process utilizes the block inheritance crossover (BIX) operator, as described in Algorithm 1.

Following this, the child chromosomes are generated and subject to one mutation operation. There is an equal probability of either the Insert (i, j) or Reassign $(O_{i,j})$ permutation-based search operators being applied.

Following this, each child chromosome undergoes a single mutation operation, with equal probability assigned to either the Insert or Reassign permutation-based search operator.

The operator denoted as Insert (i, j) functions by moving operation $O_{p,q}$ from position i into position j within the OSV, while preserving the resource allocation for all operations. In situations where $O_{p,q}$ is positioned ahead of $O_{p,k}$ $(k = 1, \ldots, q-1)$, a repair step is triggered to restore feasibility by adjusting the order of operations within job p. Additionally, the operator Reassign $(O_{i,j})$ is applied to allocate a different admissible worker–machine pair to operation $O_{i,j}$.

After all mutations are applied, the newly generated chromosomes are evaluated. A greedy selection strategy is then used to update the population: each original chromosome is replaced by the best-performing offspring in its subset, but only if the offspring achieves a lower makespan. Figure 1 presents an overview of the KGGA. The following parameter

values were employed: NS = 10, S = 7, α = 0.1, and NF = 5. The stopping criterion involved reaching a maximum number of generations set at g = 1000.

Algorithm 1 Knowledge-Guided Sequence Generation Algorithm

Input: Knowledge base π_{NF}, consisting of the NF top-performing chromosomes. ▷ Input
Output: Child chromosome sequence of operations ▷ Output
Generate a random integer L within the range $[0.5 \cdot T_0, 0.75 \cdot T_0]$, where T_0 is the length of π_{NF}
Randomly extract a subsequence of length L from π_{NF}, label it as the "Block"
Select one experimental operation sequence π_r from the knowledge base at random
Identify the first element of π_r that appears inside the Block, record its position as q_1
for $k = q_1$ to T_0 **do**
    Obtain operation π_k^r from π_r
    if π_k^r is not in the Block **then**
        Position π_k^r either before or after the Block while adhering to precedence constraints with a probability of 0.5
    end if
end for
A subsequence of length $T_0 - q_1 + 1$ is obtained
Place this subsequence after the operations $\pi_1^r - \pi_2^r - \ldots - \pi_{q_1-1}^r$, resulting in a new sequence of operations

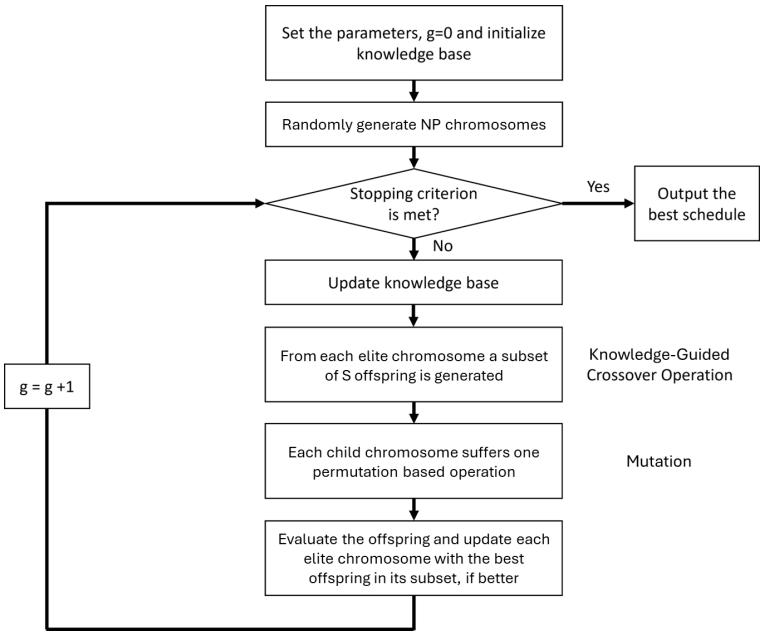


Figure 1. The flowchart of KGGA.

5. Results

The KGGA was tested on a large-scale instance from the widely used MK1-10 benchmark dataset [45]. It was compared with the KGFOA, the HABCA and a classical Genetic Algorithm—each representing a standout approach from the respective subsections of the Related Work review. All algorithms were coded in Python 3.13 and run on a PC with an Intel(R) Core(TM) i7-11800H 2.30 GHz and 32 GB of RAM.

5.1. Hyperparameter Optimization

The comparison algorithms were implemented using the hyperparameters recommended in their original publications, including their stopping criteria. Because execution

times were comparable across most algorithms, this allowed for a fair performance comparison. The exception was HABCA, which exhibited execution times approximately 50 times longer. To account for this, the number of generations in its stopping criterion was reduced. This modification was intended to ensure a more equitable computational effort across all algorithms while preserving fairness in convergence opportunities.

As for the KGGA, the effect four key hyperparameters on its performance was investigated. These hyperparameters are population size (NS), size of offspring subset (S), the updating rate of the knowledge base (α), and the number of elite chromosomes (NF). These were optimized using an adaptation of the Taguchi method of design of experiment (DOE) [46] for a moderate scaled instance of the benchmark dataset. The mutation rate (MR) was fixed at one mutation per offspring, following the effective strategy adopted in [43] to counterbalance the enhanced exploitation effectiveness introduced by the Knowledge-Guided Search mechanism.

We defined four levels for each parameter, as listed in Table 1, and designed the experiments using the orthogonal array $L_{16}(4^4)$ shown in Table 2. Each parameter setting was evaluated by performing 30 independent runs of the KGGA, where every run evolved for 1000 generations.

Table 1. Hyperparameter test values.

	Factor Level			
	1	**2**	**3**	**4**
NS	5	10	15	20
S	3	5	7	9
α	0.05	0.1	0.15	0.2
NF	1	3	5	7

Table 2. Hyperparameter factor test combinations (1–16).

Combinations 1–16				
ID	**NS**	**S**	**α**	**NF**
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	1	4	4	4
5	2	1	2	3
6	2	2	1	4
7	2	3	4	1
8	2	4	3	2
9	3	1	3	4
10	3	2	4	3
11	3	3	1	2
12	3	4	2	1
13	4	1	4	2
14	4	2	3	1
15	4	3	2	4
16	4	4	1	3

Figure 2 illustrates how different parameter values influence the average makespan. Based on the DOE results, the optimal configuration was identified as NS = 10, S = 7, $\alpha = 0.1$, and NF = 5.

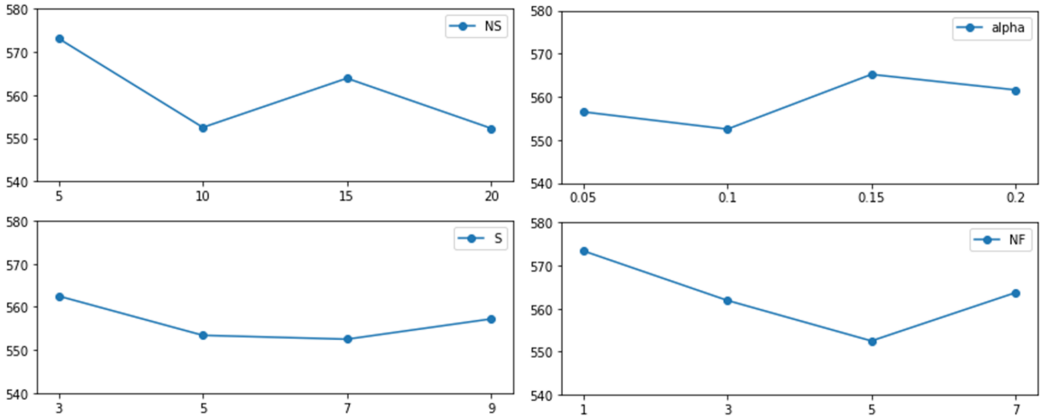


Figure 2. Parameter optimization test results.

5.2. Algorithm Performance Comparison

Table 3 summarizes the characteristics of the benchmark instances, including the numbers of jobs, machines, and workers, as well as the operations per job and their processing times. To account for the stochastic nature of the algorithms, each one was executed 30 independent times. Their performance distributions are illustrated in the box plot in Figure 3. A statistical summary of makespan values for each algorithm is reported in Table 4, and the corresponding *p*-values are presented in Table 5. The makespan distributions for all algorithms passed the Shapiro–Wilk normality test, justifying the use of the *t*-test to assess the statistical significance of performance differences.

Table 3. The MK10 instance configuration—a widely used large-scale benchmark dataset.

	n	m	w	o_i	p_{ij}
MK10	20	15	8	10–15	5–20

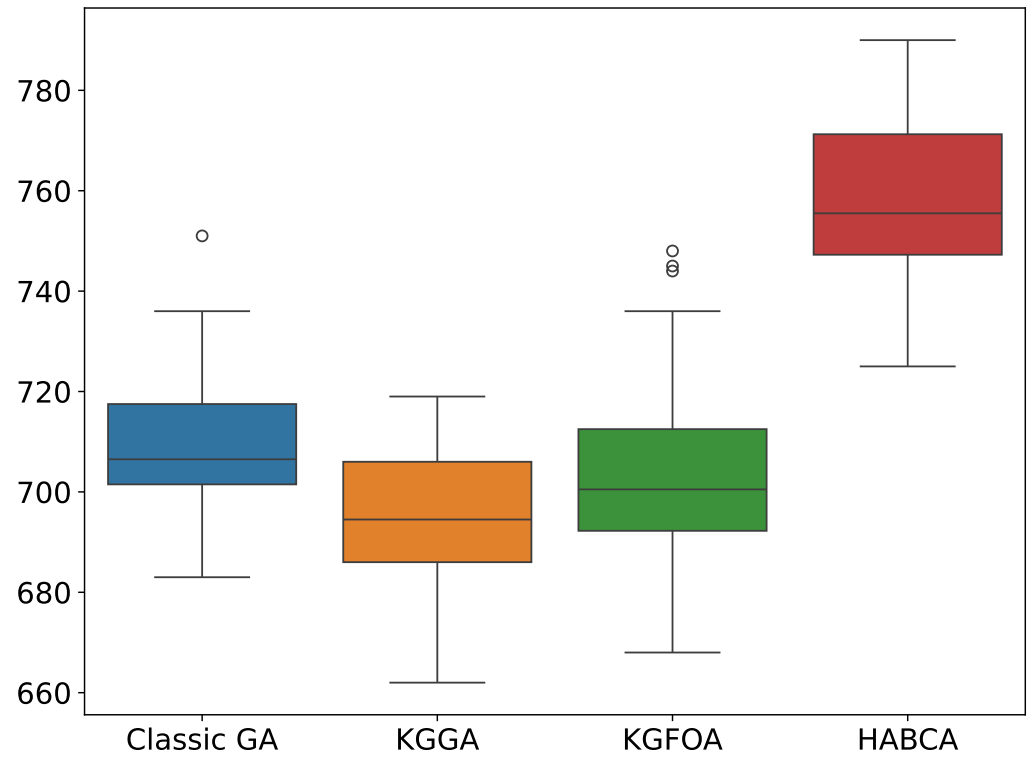


Figure 3. MK10 makespan results distribution.

Table 4. Statistical summary of makespan values for each algorithm.

Algorithm	Mean	Std Dev	Min	Q1	Median	Q3	Max
Classic GA	708	**13.9**	683	697	705	718	751
KGGA	**693**	14.0	**662**	**684**	**694**	**701**	**719**
KGFOA	702	19.1	668	**684**	699	705	748
HABCA	759	16.2	725	747	755	770	790

Table 5. Pairwise algorithm comparisons: p-values from t-test. Significant values ($p < 0.05$) are **bolded**.

	Classic GA	KGGA	KGFOA	HABCA
Classic GA	—			
KGGA	$\mathbf{2 \times 10^{-4}}$	—		
KGFOA	0.3	**0.03**	—	
HABCA	$\mathbf{5 \times 10^{-17}}$	$\mathbf{6 \times 10^{-23}}$	$\mathbf{7 \times 10^{-16}}$	—

The KGGA achieved the best overall performance, surpassing both the Classic GA and the KGFOA, from which it was conceptually derived. This result demonstrates the effectiveness of incorporating Knowledge-Guided Search in enhancing the exploitation capabilities of the crossover mechanism by directing search pressure toward high-quality genetic material.

Moreover, the KGGA demonstrated consistent high performance across runs, with no extreme poor-performing outliers observed. This consistency suggests a well-balanced interplay between exploration and exploitation, as well as a strong ability to escape local optima. Lastly, the HABCA was consistently outperformed by all other algorithms, indicating limited competitiveness within the tested benchmark scenarios.

6. Conclusions

This study introduced a novel hybrid metaheuristic, the KGGA, to address the DRC-FJSSP. By embedding knowledge-guided exploitation principles from the KGFOA directly into the crossover mechanism of a Genetic Algorithm, the KGGA reconceptualizes recombination as a form of genetic engineering—selectively amplifying high-quality genetic material rather than relying on random mixing. This enhanced crossover mechanism intensifies the search in promising regions of the solution space, promoting more efficient and focused convergence.

Experimental results on a large-scale benchmark dataset demonstrated that the KGGA consistently outperformed established baselines, including the Classic GA, KGFOA, and HABCA. The method maintained an effective balance between exploration and exploitation, resulting in consistently superior solution quality. The knowledge-guided crossover significantly improved convergence behavior, while mutation preserved diversity and reduced the risk of premature stagnation.

These findings underscore that Knowledge-Guided Search enhances the precision and efficiency of Genetic Algorithms when addressing complex scheduling tasks. Much like genetic engineering selectively amplifies desirable traits, this approach guides the evolutionary process toward more promising regions of the solution space by leveraging insights from high-quality individuals encountered during the search.

Future work could explore several extensions to further improve and generalize the proposed methodology. One promising direction is applying the KGGA framework to multi-objective variants of the DRC-FJSSP could assess its effectiveness in more complex and realistic manufacturing environments. Another avenue involves hybridizing the KGGA

with local search methods, such as variable neighborhood descent or tabu search, to further refine solutions during later stages of evolution.

Beyond scheduling, the underlying principle of knowledge-guided genetic recombination holds potential for generalization to other combinatorial optimization problems. Finally, a theoretical analysis of the convergence behavior introduced by knowledge-guided crossover could offer valuable insight into the algorithm's long-term dynamics and solution stability. These extensions would not only strengthen the practical impact of the KGGA but also deepen the understanding of how knowledge-guided crossover can be effectively embedded within genetic algorithms.

Author Contributions: Conceptualization, R.M.; methodology, R.M.; software, R.M.; validation, R.M., J.M.C.S. and S.M.V.; formal analysis, R.M.; investigation, R.M.; resources, R.M.; data curation, R.M.; writing—original draft preparation, R.M.; writing—review and editing, R.M.; visualization, R.M.; supervision, J.M.C.S. and S.M.V.; project administration, R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Fundação para a Ciência e Tecnologia (FCT) through the projects: LAETA Base Funding (DOI: 10.54499/UIDB/50022/2020), LAETA Programatic Funding (DOI: 10.54499/UIDP/50022/2020) and the doctoral grant MPP2030-FCT ID 22405888735, under the MIT Portugal Program.

Data Availability Statement: The data utilized in this study consist exclusively of the publicly available classic job shop scheduling benchmark datasets MK1–10 [45]. These data sets are widely used in scheduling research. No new data was created or collected for this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DOE	Design of Experiments
DRC-FJSSP	Dual Resource Constrained Flexible Job Shop Scheduling Problem
GA	Genetic Algorithm
HABCA	Hybrid Artificial Bee Colony Algorithm
KGGA	Knowledge-Guided Genetic Algorithm
KGFOA	Knowledge-Guided Fruit Fly Optimization Algorithm
MR	Mutation Rate
NF	Number of Elite Chromosomes
NS	Population Size
OSS	Operation Sequence Search
RAS	Resource Assignment Search
S	Offspring Subse
α	Updating Rate of the Knowledge Base

References

1. Sherwani, F.; Asad, M.M.; Ibrahim, B. Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). In Proceedings of the 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), Karachi, Pakistan, 26–27 March 2020; pp. 1–5. [CrossRef]
2. Wan, J.; Li, X.; Dai, H.N.; Kusiak, A.; Martínez-García, M.; Li, D. Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges. *Proc. IEEE* **2021**, *109*, 377–398. [CrossRef]
3. Peres, R.S.; Jia, X.; Lee, J.; Sun, K.; Colombo, A.W.; Barata, J. Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. *IEEE Access* **2020**, *8*, 220121–220139. [CrossRef]
4. Tao, F.; Zuo, Y.; Xu, L.D.; Zhang, L. IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1547–1557. [CrossRef]

5. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge Computing in IoT-Based Manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [CrossRef]
6. Ma, Y.; Zhou, H.; He, H.; Jiao, G.; Wei, S. A Digital Twin-Based Approach for Quality Control and Optimization of Complex Product Assembly. In Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, 16–18 October 2019; pp. 762–767. [CrossRef]
7. Sanderson, D.; Antzoulatos, N.; Chaplin, J.C.; Busquets, D.; Pitt, J.; German, C.; Norbury, A.; Kelly, E.; Ratchev, S. Advanced Manufacturing: An Industrial Application for Collective Adaptive Systems. In Proceedings of the 2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, Cambridge, MA, USA, 21–25 September 2015; pp. 61–67. [CrossRef]
8. Dolgui, A.; Ivanov, D.; Sethi, S.P.; Sokolov, B. Scheduling in production, supply chain and Industry 4.0 systems by optimal control: Fundamentals, state-of-the-art and applications. *Int. J. Prod. Res.* **2019**, *57*, 411–432. [CrossRef]
9. Fang, Y.; Peng, C.; Lou, P.; Zhou, Z.; Hu, J.; Yan, J. Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6425–6435. [CrossRef]
10. Rossit, D.A.; Tohmé, F.; Frutos, M. A data-driven scheduling approach to smart manufacturing. *J. Ind. Inf. Integr.* **2019**, *15*, 69–79. [CrossRef]
11. Anap, R.R. Downtime Reduction and Efficiency Improvement by Discrete Event Simulation and Production Planning Optimization for the Semiconductor Industry. Ph.D. Thesis, State University of New York at Binghamton, Binghamton, NY, USA, 2023.
12. Castañé, G.; Dolgui, A.; Kousi, N.; Meyers, B.; Thevenin, S.; Vyhmeister, E.; Östberg, P.O. The ASSISTANT project: AI for high level decisions in manufacturing. *Int. J. Prod. Res.* **2023**, *61*, 2288–2306. [CrossRef]
13. Beldiceanu, N.; Dolgui, A.; Gonnermann, C.; Gonzalez-Castañé, G.; Kousi, N.; Meyers, B.; Prud'homme, J.; Thevenin, S.; Vyhmeister, E.; Östberg, P.O. ASSISTANT: Learning and Robust Decision Support System for Agile Manufacturing Environments. *IFAC-PapersOnLine* **2021**, *54*, 641–646. [CrossRef]
14. Stankovic, A.; Petrović, G.; Cojbasic, Z.; Marković, D. An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem. *Oper. Res. Eng. Sci. Theory Appl.* **2020**, *3*, 13–28. [CrossRef]
15. Zhang, H.; Buchmeister, B.; Li, X.; Ojstersek, R. Advanced Metaheuristic Method for Decision-Making in a Dynamic Job Shop Scheduling Environment. *Mathematics* **2021**, *9*, 909. [CrossRef]
16. Pongchairerks, P. A Two-Level Metaheuristic Algorithm for the Job-Shop Scheduling Problem. *Complexity* **2019**, *2019*, 8683472. [CrossRef]
17. Magalhães, R.; Vieira, S.; Sousa, J. Threat elimination algorithm for Dual Resource Constrained Flexible Job Shop Scheduling Problems. *IFAC-PapersOnLine* **2022**, *55*, 2288–2293. [CrossRef]
18. Xu, J.; Zhang, J. Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; pp. 8633–8638. [CrossRef]
19. Bhatt, N.; Chauhan, N.R. Genetic algorithm applications on Job Shop Scheduling Problem: A review. In Proceedings of the 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), Faridabad, India, 8–10 October 2015; pp. 7–14. [CrossRef]
20. Garza-Santisteban, F.; Sánchez-Pámanes, R.; Puente-Rodríguez, L.A.; Amaya, I.; Ortiz-Bayliss, J.C.; Conant-Pablos, S.; Terashima-Marín, H. A Simulated Annealing Hyper-heuristic for Job Shop Scheduling Problems. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 57–64. [CrossRef]
21. Chaukwale, R.; Kamath, S.S. A modified Ant Colony optimization algorithm with load balancing for job shop scheduling. In Proceedings of the 2013 15th International Conference on Advanced Computing Technologies (ICACT), Rajampet, India, 21–22 September 2013; pp. 1–5. [CrossRef]
22. Pinedo, M.L. *Scheduling: Theory, Algorithms, and Systems*, 5th ed.; Springer International Publishing: Berlin/Heidelberg, Germany, 2018.
23. Garey, M.R.; Johnson, D.S.; Sethi, R. The Complexity of Flowshop and Jobshop Scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
24. Xie, J.; Gao, L.; Peng, K.; Li, X.; Li, H. Review on flexible job shop scheduling. *IET Collab. Intell. Manuf.* **2019**, *1*, 67–77. [CrossRef]
25. Hashemi-Petroodi, S.E.; Dolgui, A.; Kovalev, S.; Kovalyov, M.Y.; Thevenin, S. Workforce reconfiguration strategies in manufacturing systems: A state of the art. *Int. J. Prod. Res.* **2021**, *59*, 6721–6744. [CrossRef]
26. Magalhães, R.; Martins, M.; Vieira, S.; Santos, F.; Sousa, J. Encoder-Decoder Neural Network Architecture for solving Job Shop Scheduling Problems using Reinforcement Learning. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–8. [CrossRef]
27. Teekeng, W.; Thammano, A.; Unkaw, P.; Kiatwuthiamorn, J. A new algorithm for flexible job-shop scheduling problem based on particle swarm optimization. *Artif. Life Robot.* **2016**, *21*, 18–23. [CrossRef]

28. Shi, J.; Chen, M.; Ma, Y.; Qiao, F. A new boredom-aware dual-resource constrained flexible job shop scheduling problem using a two-stage multi-objective particle swarm optimization algorithm. *Inf. Sci.* **2023**, *643*, 119141. [CrossRef]
29. Farjallah, F.; Nouri, H.E.; Driss, O.B. Multi-agent model-based intensification-driven tabu search for solving the dual-resource constrained flexible job shop scheduling. *J. Inf. Telecommun.* **2025**, *9*, 295–316. [CrossRef]
30. Farjallah, F.; Nouri, H.E.; Belkahla Driss, O. Multi-start Tabu Agents-Based Model for the Dual-Resource Constrained Flexible Job Shop Scheduling Problem. In Proceedings of the Computational Collective Intelligence, Leipzig, Germany, 9–11 September 2024; Nguyen, N.T., Manolopoulos, Y., Chbeir, R., Kozierkiewicz, A., Trawiński, B., Eds.; Springer: Cham, Switzerland, 2022; pp. 674–686.
31. Teng, G. An improved genetic algorithm for dual-resource constrained flexible job shop scheduling problem with tool-switching dependent setup time. *Expert Syst. Appl.* **2025**, *281*, 127496. [CrossRef]
32. Zhang, S.; Du, H.; Borucki, S.; Jin, S.; Hou, T.; Li, Z. Dual Resource Constrained Flexible Job Shop Scheduling Based on Improved Quantum Genetic Algorithm. *Machines* **2021**, *9*, 108. [CrossRef]
33. Fan, D.; Wang, C. A Genetic Algorithm for the Dual Resource Constrained Flexible Job Shop Scheduling Problem Considering Preparation Times. In Proceedings of the 2024 12th International Conference on Traffic and Logistic Engineering (ICTLE), Macau, China, 23–25 August 2024; pp. 128–132. [CrossRef]
34. Li, X.; Xing, S. Dynamic scheduling problem of multi-objective dual resource flexible job shop based on improved genetic algorithm. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 28–31 July 2024; pp. 2046–2051. [CrossRef]
35. Lu, X.; Lu, C. Mixed-production flexible assembly job shop scheduling considering parallel assembly sequence variations under dual-resource constraints using multi-objective hybrid memetic algorithm. *Comput. Oper. Res.* **2025**, *176*, 106932. [CrossRef]
36. Li, C.; Meng, L.; Ullah, S.; Duan, P.; Zhang, B.; Song, H. Novel hybrid algorithm of cooperative evolutionary algorithm and constraint programming for dual resource constrained flexible job shop scheduling problems. *Complex Syst. Model. Simul.* **2025**, *5*, 236–251. [CrossRef]
37. Ren, Z.; Qu, H.; Safran, M.; Wan, J. Dual-Resource Optimization Configuration and Collaborative Scheduling for Flexible Job Shop Under Worker Flexibility Constraints. *IEEE Access* **2025**, *13*, 72347–72363. [CrossRef]
38. Wu, R.; Li, Y.; Guo, S.; Xu, W. Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. *Adv. Mech. Eng.* **2018**, *10*, 1687814018804096. [CrossRef]
39. Zhang, J.; Wang, W.; Xu, X. A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility. *J. Intell. Manuf.* **2015**, *28*, 1961–1972. [CrossRef]
40. Zhang, J.; Jie, J.; Wang, W.; Xu, X. A hybrid particle swarm optimisation for multi-objective flexible job-shop scheduling problem with dual-resources constrained. *Int. J. Comput. Sci. Math.* **2017**, *8*, 526–532. [CrossRef]
41. Gong, G.; Chiong, R.; Deng, Q.; Gong, X. A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility. *Int. J. Prod. Res.* **2020**, *58*, 4406–4420. [CrossRef]
42. Li, J.; Huang, Y.; Niu, X. A branch population genetic algorithm for dual-resource constrained job shop scheduling problem. *Comput. Ind. Eng.* **2016**, *102*, 113–131. [CrossRef]
43. Zheng, X.L.; Wang, L. A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. *Int. J. Prod. Res.* **2016**, *54*, 5554–5566. [CrossRef]
44. Magalhães, R.; Santos, F.; Vieira, S.; Sousa, J.M.C. Dual Resource Flexible Job Shop Scheduling Problems: The xBTF Algorithm. In Proceedings of the Information Processing and Management of Uncertainty in Knowledge-Based Systems, Milan, Italy, 11–15 July 2022; Lesot, M.J., Vieira, S., Reformat, M.Z., Carvalho, J.P., Batista, F., Bouchon-Meunier, B., Yager, R.R., Eds.; Springer: Cham, Switzerland, 2024; pp. 115–125.
45. Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. *Ann. Oper. Res.* **1993**, *41*, 157–183. [CrossRef]
46. Montgomery, D.C. *Design and Analysis of Experiments*, 10th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.