Title	フローショップ環境における生産再スケジューリングのためのハイブリッド手法
Author(s)	PAKKAPORN, SAOPHAN
Citation	
Issue Date	2023-12
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/18817
Rights	
Description	Supervisor: HUYNH, Van Nam, 先端科学技術研究科, 博士

Doctoral Dissertation

A Hybrid Methodology for Production Rescheduling in Flow Shop Environment

Pakkaporn Saophan

Supervisor: Van Nam Huynh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Knowledge Science]

December, 2023

Abstract

The rescheduling process is indispensable in actual production environments to adapt schedules when significant disturbances render existing ones suboptimal. Manufacturers often face the need to rapidly reschedule production tasks. This research presents a methodology for production rescheduling in flow shop environments with machine failure disturbances, named PPGA-ANN. The primary objective of the methodology is to minimize makespan while ensuring sufficient computational time for rescheduling. Prior to production, the proposed methodology includes a stage of training in which the Perturbation Population Genetic Algorithm (PPGA) is employed to address generated scenarios of flow shop production with machine failure problems. To validate the efficacy of PPGA, its performance is compared to that of other research and the genetic algorithm by using the same data set from a widely used scheduling benchmark. In addition, artificial neural networks (ANNs) are used to store the PPGA-acquired rescheduling knowledge. During the stage of implementing, when a machine failure occurs during production, ANNs provide the rescheduling solution if the machine failure situation matches the generated scenarios. Otherwise, the PPGA, incorporating the initial solution obtained from the ANNs, offers the rescheduling solution. Experimental results consistently demonstrate that PPGA-ANN outperforms benchmark algorithms in terms of makespans, while also providing expedited solutions compared to the genetic algorithm and PPGA used individually. In conclusion, the proposed PPGA-ANN for flexible manufacturing production rescheduling not only exhibits robust performance in handling machine failures in scheduling problems but also provides faster schedules, addressing the limitations of existing state-of-the-art meta-heuristic algorithms that may have impractical computational times for implementation.

Keywords: production rescheduling, machine failure, flow shop production, genetic algorithm, artificial neural network

Acknowledgement

I would like to extend my heartfelt appreciation to Professor Van-Nam Huynh, my esteemed supervisor. His unwavering guidance and persistent assistance have been instrumental in making this thesis a reality.

I am deeply thankful to Professor Hieu Chi Dam, my second supervisor, for his generous support and invaluable insights.

I am also sincerely indebted to Associate Professor Warut Pannakkong, who not only supervised my master's degree but also served as an external committee member for my defense. His unwavering support, kind assistance, and insightful suggestions have been indispensable in the completion of this thesis.

I extend my gratitude to Professor Tsutomu Fujinami for his guidance and unwavering support throughout my minor research project.

I would like to express my sincere appreciation to Professor Takashi Hashimoto and Associate Professor Eunyoung Kim for their vital roles as committee members during my defense. Their critical feedback has significantly strengthened the quality of my research.

I am grateful to my colleagues and friends in the Huynh-Lab for their valuable assistance and encouragement throughout my research journey.

I extend my thanks to the dedicated staff at JAIST for their exceptional support and service during my time there.

Last but certainly not least, I want to express my deepest gratitude to my family for their unwavering love, understanding, and constant support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the dynamic landscape of manufacturing, where unforeseen disruptions can strike at any moment, the ability to swiftly and efficiently reschedule production systems assumes a pivotal role as a critical decision support function. Its significance cannot be overstated in ensuring the overall success of manufacturing processes, particularly in the contemporary fiercely competitive business environment. The effectiveness of production scheduling and rescheduling takes on added importance in the current competitive landscape [22]. Its objectives extend beyond merely facilitating the adoption of innovative production processes; they encompass the strategic maximization of resource utilization, ensuring the sustainable survival of businesses within an intensely competitive marketplace.

Within the context of today's fast-paced manufacturing environments, the imperative for rapid adaptation to unexpected disruptions looms large. Flow shop production, distinguished by its linear arrangement of machines, presents distinctive challenges when it comes to rescheduling in response to disruptive events like machine failures. While traditional optimization techniques remain valuable, their efficacy is constrained when confronted with the dynamic nature of these disruptions. Concurrently, the emergence of machine learning algorithms, with a spotlight on artificial neural networks (ANNs), has opened promising avenues for addressing real-time rescheduling challenges.

To bridge the divide between traditional optimization methods and the realm of machine learning, our research has embraced hybrid methodologies. These innovative approaches are meticulously crafted to harness the distinct strengths of optimization techniques, such as genetic algorithms, and the formidable capabilities of machine learning. The overarching objective is clear: to realize the swift and effective rescheduling of production in flow shop environments. The seamless fusion of these two paradigms holds the promise of expediting rescheduling processes while minimizing production downtime, thereby augmenting the overall efficiency of manufacturing operations.

1.1 Problem Statement

Traditionally, pre-defined timetables have been the standard practice in production manufacturing for many decades. However, achieving optimum productivity while adhering to the predetermined timetable has become increasingly challenging in the current production environment characterized by high uncertainty and complexity. Various studies have explored dynamic scheduling or rescheduling techniques to minimize the impact of deliberate or accidental disruptions on production operations. Despite this, it is widely known that very few practical applications have been implemented in real-world industries [55].

In dynamic manufacturing, where variability often emerges, production rescheduling is typically required to mitigate the consequences of disrupted events while maintaining optimal performance [58]. Substantial disruptions in production, such as machine failure, urgent orders, or changes in delivery dates, make it impossible to adhere to previously established schedules and, thus, are identified as rescheduling factors [13]. While there are various factors that contribute to uncertainty in manufacturing, the failure of technological machinery is still considered to be the central problem [45]. The common causes of rescheduling in the manufacturing industry is machine failure, which frequently results in substandard equipment performance. Consequently, manufacturers face the significant challenge of rescheduling actions promptly and effectively to maintain high production levels. In contrast to actual production environments, where unreliable machines may delay product launch, many researchers in production scheduling have presumed that machines will be available throughout the planning horizon. [31].

In recent years, the emergence of Industry 4.0 has led to an increased focus on new perspectives and challenges in the research of dynamic production scheduling [22]. In light of this, this research proposes a hybridization methodology for rescheduling flow shop production in the event of machine failure. This methodology combines a metaheuristic algorithm, represented by a proposed genetic algorithm, with a supervised learning technique, represented by an ANN. This methodology's primary objective is to quickly generate a new production schedule while minimizing the makespan.

This research first develop the production with machine failure scenarios. The Perturbation Population Genetic Algorithm (PPGA) is proposed and utilized to solve and determine the optimal solution for each scenario of large-scale flow shop production with machine failure. The performance of the PPGA is evaluated by adopting the benchmark from Taillard [51] and comparing the results with the other research algorithms. Furthermore, ANNs are trained using the information obtained from machine failure scenarios and their respective solutions. The solutions generated by the trained ANNs are expected to serve as the best solution for rescheduling, even if they are suboptimal, and can be used as an initial solution for PPGA. The PPGA-based knowledge generated by the trained network is expected to provide a near-optimal solution in a shorter computing time than PPGA without knowledge.

In conclusion, the proposed rescheduling methodology is evaluated based on optimality and computing time and compared to traditional genetic algorithm and the PPGA without knowledge. The results demonstrate the effectiveness of the suggested hybridization methodology and its ability to minimize the impact of machine failure in flow shop production. The proposed methodology has significant potential for practical implementation in real-world manufacturing environments, and it opens up new avenues for future research in the field of dynamic production scheduling.

1.2 Research Objectives

- **Development of an Innovative Rescheduling Methodology:** The foremost objective of this research is to pioneer the development of a cutting-edge production rescheduling methodology tailored explicitly for flow shop environments plagued by machine failure disturbances. The primary aim is to create a methodology that not only minimizes the makespan, a pivotal metric in production efficiency but also ensures that the computational time required for rescheduling remains within practical bounds. This objective underscores the significance of crafting solutions that are not only theoretically sound but also operationally feasible. By achieving this goal, the research addresses the pressing need for agile and efficient rescheduling strategies in contemporary manufacturing landscapes.

- **Practical Relevance and Efficacy Demonstration:** Another crucial research objective is to demonstrate the practical relevance and efficacy of the proposed methodology in effectively managing and mitigating machine failure issues within the broader context of scheduling challenges in flexible manufacturing environments. This entails conducting extensive numerical experiments and empirical validations that provide concrete evidence of the methodology's real-world applicability. By showcasing its performance in a practical setting, the research contributes to bridging the gap between theoretical advancements and industry requirements. It not only provides a theoretical framework but also serves as a practical guide for manufacturers seeking solutions to address machine failures and optimize their production schedules.

- **Advancement of Meta-Heuristic Algorithms:** An overarching research objective is to contribute significantly to the advancement of current meta-heuristic algorithms for production rescheduling. This objective encompasses the introduction of a more efficient and expedient methodology, which stands as a testament to the relentless pursuit of excellence in optimization techniques. By combining the PPGA with ANNs, this research showcases the potential for synergy between different approaches. The aim is to enhance the exploration and exploitation capabilities of algorithms in solving complex production rescheduling problems. This objective underscores the commitment to pushing the boundaries of meta-heuristic algorithms and unlocking new horizons in the field of production scheduling and rescheduling.

In summary, these research objectives collectively represent a holistic and ambitious endeavor aimed at fostering innovation and achieving excellence within the sphere of production rescheduling. They embody a steadfast commitment not only to confronting present challenges but also to establishing a solid foundation for prospective advancements, thereby ensuring that this research constitutes a durable and invaluable contribution to both the academic and industrial realms.

1.3 Chapter Organization

- **Chapter 1**: This pivotal chapter serves as the gateway to the dissertation's exploration. It expertly introduces the concept of production rescheduling, emphasizing its significance in addressing dynamic manufacturing challenges. It succinctly articulates the problem statement and lays out the well-defined research objectives, providing a clear roadmap for the study's trajectory. Additionally, the chapter offers readers a valuable guide by outlining the dissertation's organization, making navigation through the forthcoming content seamless and comprehensible.

- **Chapter 2**: In this comprehensive chapter, a thorough and erudite examination of the pertinent literature is presented. The extensive review encompasses production scheduling, genetic algorithm applications in the context of production scheduling, production rescheduling, and the realm of Artificial Intelligence (AI)-based methodologies, with a particular emphasis on ANNs. This chapter serves as the foundational knowledge base upon which the research is built, showcasing the existing landscape of relevant scholarship and highlighting gaps and opportunities for innovative contributions.

- **Chapter 3**: An essential component of the dissertation, this chapter delves into the intricacies of the flow shop production problem. Its meticulous and exhaustive exploration encompasses various facets and nuances of this complex challenge. By providing a deep understanding of the problem domain, this chapter equips readers with the necessary background to appreciate the complexities addressed by the proposed methodology.

- **Chapter 4**: At the heart of the dissertation lies this chapter, where the innovative and comprehensive methodology for generating rapid production rescheduling solutions takes center stage. Painstakingly designed to counteract the disruptive impact of machine failures in dynamic manufacturing systems, this methodology represents a significant contribution to the field. It aspires to enhance production process performance and efficiency, even under challenging circumstances, making this chapter a cornerstone of the research.

- **Chapter 5**: This empirical chapter embodies the numerical implementation and experimental rigor of the proposed methodology. It offers tangible evidence to validate the effectiveness of the methodology through exhaustive testing and analysis. By evaluating the methodology's performance and its practical applicability, this chapter ensures that the research findings can be confidently applied in real-world manufacturing scenarios.

- **Chapter 6**: This chapter offers a comprehensive exploration and in-depth discussion of critical aspects related to the research findings and methodology.

- **Chapter 7**: This chapter encapsulates the research's contributions, which span methodological implications, practical applicability, and advancements in the field of Knowledge Science. By articulating the multifaceted ways in which this research extends the boundaries of knowledge, this chapter underscores the significance of the study's outcomes.

- **Chapter 8**: In this concluding chapter, the research journey culminates in the presentation of its key findings and their implications. It also humbly acknowledges the limitations encountered along the way, serving as an honest reflection on the study's scope. Furthermore, this chapter provides valuable signposts for future research directions, igniting the torch for continued exploration and innovation in the field.

Chapter 2

Literature Review

A thorough review of related research serves as a critical foundation for understanding the methodologies and strategies previously deployed in addressing the intricate challenges of production scheduling and rescheduling. It enables us to glean valuable insights into the landscape of prior research endeavors and discern effective strategies for the seamless integration of the proposed production rescheduling system. This comprehensive review of earlier research not only illuminates the methodologies employed but also elucidates the problem domains they aimed to tackle. It provides a roadmap for engaging in a meaningful discourse, positioning critical issues, and delineating the underlying purpose that steers the methodology adopted in this research.

Within the annals of scholarly inquiry, this chapter undertakes the arduous task of conducting an extensive literature review, focusing intently on the multifaceted realm of production scheduling and rescheduling, with a specialized emphasis on the intricate dynamics of flow shop operations. Notably, as of our current understanding, the rescheduling conundrum within the context of flow shop production has not received the level of scholarly attention commensurate with its complexity. Thus, the pioneering development of an integrative rescheduling methodology tailor-made for the unique intricacies of flow shop production stands as a significant and highly valuable contribution forged by this research.

The review encompasses not only an examination of the limitations inherent in previous flow shop scheduling methods but also an in-depth discussion of novel perspectives and preexisting research efforts dedicated to the formidable challenge of production rescheduling. This meticulous survey of the literature forcefully underscores the pressing need for innovative approaches to tackle the complexities of production rescheduling and accentuates the potential advantages that the proposed methodology holds in store. Indeed, the proposed methodology, PPGA-ANN, emerges as a direct response to the limitations identified in prior approaches. By harnessing the combined strengths of meta-heuristic algorithms and machine learning, PPGA-ANN offers a beacon of hope in the face of the formidable challenges posed by production rescheduling, charting a course towards more efficient and effective solutions.

2.1 Production Scheduling

In the realm of academia, production scheduling has captured the interest of scholars from various disciplines for many decades. Graves [21] provides a succinct yet comprehensive definition of production scheduling as the intricate task of determining the most optimal schedule or sequence for a group of activities within a production line. This definition highlights the fundamental challenge that lies at the heart of production scheduling—balancing the multitude of factors to achieve efficiency and effectiveness. Notably, this topic has garnered significant recognition, solidifying its position as a well-established field of study. Moreover, it is widely acknowledged to be an NP-hard problem [5], signifying the formidable computational complexities that underlie this domain.

The approach taken to scheduling is heavily contingent on the specific machine environment in which it operates. These environments encompass a diverse range, including the flow shop, job shop, parallel machine, and single machine settings [41]. Among these, the flow shop operation stands out as a prominent and extensively utilized production configuration. A multitude of industries, ranging from the automotive and automobile manufacturing sectors [17, 62] to the plastics industry [3] and the wood industry [16], have embraced the flow shop operation for its applicability in streamlining production processes and enhancing efficiency. This widespread adoption underscores its pivotal role in modern manufacturing.

The historical trajectory of research in this domain traces back to the 1950s when scholars began delving into the challenge of minimizing makespan in production sequencing problems. Johnson's seminal work in 1954 [23] marks a significant milestone, as it introduced a simple yet effective algorithm tailored for two-machine flow shop scheduling. This pioneering work laid the foundation for the emergence of scheduling as an independent and vibrant field of research, characterized by a diverse array of approaches and methodologies. The ensuing decades witnessed a proliferation of research efforts, each contributing to the ever-evolving landscape of production scheduling.

In the realm of production scheduling, the availability of exact optimization algorithms, as exemplified by studies such as [48], [53], and [20], is indeed noteworthy. However, while exact solvers possess the theoretical capability to provide optimal solutions, their practical effectiveness diminishes considerably as the scale and complexity of production scheduling problems increase. In real-world manufacturing scenarios, the need for optimal solutions for moderate- and large-scale problems is relatively rare [61]. This practical constraint has led researchers and practitioners to explore alternative avenues, particularly heuristic algorithms.

Heuristic algorithms, including well-known approaches like those described in [35], [49], and [66], offer a more pragmatic approach to tackling production scheduling challenges. These methods, while delivering reasonably good solutions, grapple with the substantial computational time required for moderately sized problems. Consequently, a notable shift has transpired in recent years, with researchers increasingly turning to meta-heuristic algorithms as a means of striking a balance between solution quality and computational efficiency.

Drawing from the extensive body of literature, it becomes evident that various meta-heuristic algorithms have found utility in addressing the intricacies of flow shop scheduling. Notable examples encompass tabu search [18], iterated greedy [44], simulated annealing [60], ant colony optimization algorithms [14], genetic algorithms [63], and the arithmetic optimization algorithm [1]. Among these, genetic algorithms have emerged as particularly popular and effective tools, garnering attention for their proficiency in solving operational management problems, including scheduling challenges

[24]. Genetic algorithms, with their adaptability and applicability to large and continuous search spaces [25], have carved out a prominent niche within the landscape of meta-heuristic solutions for production scheduling.

2.1.1 Genetic Algorithm (GA)

The genetic algorithm, a powerful meta-heuristic optimization technique, derives its inspiration from the intricate mechanisms of natural selection and evolution. It operates by emulating genetic processes like crossover, mutation, and selection, meticulously navigating the vast solution space in pursuit of optimal or near-optimal solutions to intricate problems. Through the iterative refinement of a population of potential solutions, the genetic algorithm diligently strives to converge upon solutions that exhibit superior performance.

Its prowess in swiftly generating efficient results within reasonable timeframes has rendered the genetic algorithm an invaluable tool in the realm of scheduling problems [42]. Researchers and practitioners have harnessed its capabilities to craft diverse scheduling strategies tailored to address an array of complex challenges. These endeavors encompass optimizing workflow execution costs in cloud computing environments while adhering to stringent deadlines [32], resolving the intricate routing and scheduling problems inherent in parcel delivery services [40], tackling the demanding hybrid flow shop scheduling scenarios encountered in production scheduling [63], and addressing the multifaceted flexible job shop environments replete with sequence-dependent set-up times and job lag times [59].

However, despite its effectiveness, the genetic algorithm is not without its limitations. One significant challenge lies in its susceptibility to becoming ensnared within local optima or succumbing to premature convergence during its exploration of the solution space [4, 56]. This phenomenon, known as getting trapped in local optima, denotes a scenario where the algorithm becomes entrenched within a suboptimal solution, effectively restricting its search to a confined region of the solution space. In essence, the algorithm struggles to venture beyond the boundaries of the local optimum in pursuit of better or globally optimal solutions. This tendency is often a consequence of the algorithm's reliance on local search procedures, which prioritize incremental improvements to the current solution, inadvertently limiting the exploration of alternative regions within the expansive solution space. The implications of getting trapped in local optima can be particularly pronounced in complex optimization problems like scheduling and rescheduling, potentially yielding suboptimal outcomes.

In response to these challenges, the proposed Perturbation Population Genetic Algorithm (PPGA) emerges as an evolutionary leap beyond the conventional genetic algorithm. PPGA is meticulously designed to mitigate the limitations inherent in its predecessor, offering a more robust and efficient approach to solving complex scheduling and rescheduling problems.

However, while meta-heuristic algorithms, including genetic algorithms, demonstrate remarkable prowess in handling large-scale production scheduling challenges, they may still grapple with computational time constraints when faced with the exigencies of rapid production rescheduling in the wake of unforeseen disruptions. Furthermore, it is worth noting that a substantial portion of scheduling research tends to overlook the real-world execution nuances that may manifest when implementing global manufacturing schedules. These studies often make the assumption that the algorithm will execute the schedule with pinpoint precision. Hence, the core objective of this research is to enhance the genetic algorithm's meta-heuristic framework, transforming it into an adept and well-suited solution for the intricate landscape of rescheduling challenges, thereby bridging the gap between optimization and real-world manufacturing execution.

2.2 Production Rescheduling

Unforeseen disruptions in the realm of production operations can introduce delays, especially in environments characterized by high levels of production activity. It becomes imperative to assess whether a new production schedule is warranted to mitigate the repercussions of these delays and uphold commitments to customers. The concept of "production rescheduling" encompasses the dynamic adjustment of production plans to accommodate the shifts necessitated by flexible working conditions and disruptions encountered on the shop floor. These challenges introduce significant complexities, resulting in a growing array of factors that must be considered during the rescheduling process. The task of updating existing scheduling systems and swiftly reaching a new optimal solution for pending work orders is a time-intensive endeavor [61]. Consequently, rescheduling challenges have garnered sustained attention from various quarters, including academic institutions and corporate enterprises. Several comprehensive review studies, such as those conducted by Vieira, Herrmann, and Lin [58], Ouelhadj and Petrovic [38], Cardin et al. [7], Uhlmann and Frazzon [55], and Larsen and Pranzo [29], have delved into the multifaceted domain of production rescheduling.

In pursuit of solutions to the intricate challenges intertwined with production rescheduling, a multitude of research initiatives have been undertaken. Sabuncuoglu and Goren [43] pioneered the concept of proactive scheduling, an innovative approach that melds decision theory with rescheduling policies to establish robustness and stability metrics. Dong and Jang [11] introduced heuristic algorithms and devised the Wilkerson-Irwin algorithm, grounded in an active schedule generation process, to mitigate rescheduling tardiness stemming from machine breakdowns. Kundakci and Kulak [28] unveiled a hybrid genetic algorithm tailored for dynamic scheduling challenges. This algorithm not only delivers high-quality results expeditiously but also minimizes computational demands, offering a practical solution for real-world rescheduling scenarios. These research endeavors collectively exemplify the ceaseless pursuit of innovative methodologies to navigate the intricate landscape of production rescheduling.

2.2.1 Artificial Neural Network (ANN)

The realm of Artificial Intelligence (AI) has witnessed a notable surge in attention within the context of production scheduling and rescheduling. This heightened interest can be attributed to AI's remarkable capability to furnish real-time solutions, addressing the evolving challenges of dynamic manufacturing environments. Evidently, research featured in distinguished journals has consistently advocated for the strategic integration of AI and machine learning as a means to surmount the inherent limitations in these intricate manufacturing landscapes [6, 30, 39, 57].

However, despite the growing recognition of AI's potential, enlightening insights gleaned from the study conducted by Usuga et al. [57] cast a revealing light on the existing research landscape. It emerges that a significant portion, approximately 75%, of potential research avenues in this domain have scarcely delved into the transformative capabilities of machine learning for production planning and control. This revelation underscores the substantial untapped potential awaiting further exploration within this fertile research territory. The call for deeper investigation echoes throughout academic circles, urging scholars to embark on journeys of discovery aimed at unlocking the latent possibilities that lie at the intersection of machine learning and production management.

The ascendancy of AI-based methodologies has left an indelible mark on diverse research domains, propelling the exploration of a myriad of alternative optimization strategies. Within this transformative landscape, machine learning has ascended to the forefront as a formidable tool for crafting optimization algorithms that harness the power of experiential learning from problem instances. A noteworthy instance of this paradigm shift is the introduction of a fuzzy neural network by Zhang et al. [64]. This innovative approach meticulously tailors a decision model for rescheduling, drawing inspiration from the principles of fuzzy logic and neural networks. By adapting to system states and disturbances, this methodology offers a promising avenue for rescheduling in dynamic environments. However, it's important to note that while it demonstrates potential, this approach has yet to be fully implemented or rigorously tested within the complex terrain of manufacturing systems.

Furthermore, the realm of reinforcement learning has garnered considerable attention in addressing intricate optimization challenges. Nazari et al. [36] harnessed the power of reinforcement learning to tackle the formidable vehicle routing problem (VRP). Their approach leverages the computation of rewards and incorporates an attention mechanism to optimize routing decisions. This innovative fusion of machine learning principles and optimization strategies represents a significant advancement in the domain of logistics and transportation, where efficient routing is paramount. Nonetheless, it's imperative to recognize that the applicability of such reinforcement learning approaches extends beyond VRP and holds substantial promise in addressing various production rescheduling challenges within manufacturing systems.

In the context of the rapidly evolving landscape of Industry 4.0, Li et al. [30] unveiled a groundbreaking integrated approach that seamlessly combines the prowess of machine learning with optimization algorithms. Their innovative methodology aims to discern rescheduling trends and effectively address the intricate flexible job shop scheduling problem. By leveraging the capabilities of machine learning, this approach not only identifies critical rescheduling patterns but also harnesses optimization techniques to craft efficient schedules. This interdisciplinary fusion reflects the increasing relevance of AI and machine learning in modern manufacturing, where adaptability and efficiency are paramount.

In response to the unprecedented challenges posed by the COVID-19 pandemic, Wu et al. [61] pioneered a novel solution rooted in ANNs integrated with reinforcement learning. Their innovative approach focuses on the emergency scheduling of medical mask production, where timely and efficient scheduling can be a matter of life and death. While their algorithm has demonstrated remarkable speed in scheduling, there is a recognition that further refinement, particularly in the realm of reinforcement learning, can elevate its capabilities. This opens up avenues for future research to fortify the foundation of reinforcement learning within the context of manufacturing rescheduling.

The integration of ANNs with genetic algorithms (GAs) has been a subject of interest for Takeda-Berger and Frazzon [52]. Their pioneering work introduces an inventory data-driven predictive-reactive production scheduling model that marries the predictive strength of ANNs with the optimization prowess of GAs. This innovative model not only generates predictive schedules using the ANN-GA approach but also incorporates a simulation-based optimization method to craft reactive schedules. This dual-focused approach, encompassing both predictive and reactive schedules, highlights the adaptability and versatility of ANNs in addressing dynamic manufacturing challenges. It further underscores the potential for hybrid methodologies to offer comprehensive solutions for production rescheduling in complex, real-world scenarios.

The integration of ANNs into the proposed methodology is underpinned by several compelling reasons, each enhancing the methodology's effectiveness and adaptability. First and foremost, ANNs exhibit an impressive capacity for capturing intricate patterns and deciphering complex relationships within large and convoluted datasets. In the context of production rescheduling, ANNs shine in their ability to comprehend and represent the multifaceted interconnections among various production parameters and the myriad scenarios of machine failures. This innate capability empowers ANNs to generate precise and insightful initial solutions, laying a robust foundation for the rescheduling process.

Secondly, the trained ANN bolsters the proposed methodology's computational speed, rendering it exceptionally advantageous in time-critical production environments that demand rapid decision-making. Leveraging the computational prowess of ANNs, the proposed methodology excels at swiftly and efficiently producing rescheduling solutions, ensuring timely adaptations in dynamic manufacturing settings. This accelerated decision-making capability is instrumental in minimizing production disruptions and their associated costs.

Furthermore, ANNs exhibit a capacity for learning and adaptation, enabling them to flexibly respond to new and unforeseen instances that may deviate from the training data. This adaptability makes ANNs well-suited for scenarios in which machine failures manifest in ways not explicitly encountered during the training phase. The inherent flexibility and adaptability of ANNs bolster the robustness and applicability of the proposed methodology in tackling diverse and ever-evolving production rescheduling challenges. It's essential to acknowledge that ANNs, like any tool, have limitations. In cases where new instances significantly depart from the training data, the solutions offered by ANNs may fall short of optimality. Hence, the integration of a meta-heuristic component becomes imperative to ensure the attainment of optimal or near-optimal solutions. This synergy between ANNs and meta-heuristics further fortifies the rescheduling methodology's resilience and computational efficiency.

In recognition of the evolving landscape of production rescheduling, this research underscores the importance of continued investigation in this domain. By emphasizing the need for further research, it contributes to the ongoing scholarly discourse and advocates for intensified efforts aimed at refining and expanding the applications utilized in production rescheduling. This acknowledgment underscores the significance of perpetual improvement and the pursuit of innovative scheduling methodologies, which are essential for staying competitive and adaptable within the dynamic manufacturing sector.

Chapter 3

Problem Description

3.1 Flow Shop Production

In the realm of production and manufacturing processes, the flow shop configuration stands out as a distinctive and well-defined production system [33]. Within this configuration, a predetermined sequence of machines or workstations is meticulously arranged in a specific order [34], as visually depicted in Figure 3.1. Within this linear flow of work, each job or product adheres to an unchanging and predefined sequence of operations. This sequence is rigorously followed as the job progresses through the series of workstations. Each machine is dedicated to executing particular tasks, and once a job reaches the culmination of its processing at one machine, it seamlessly transitions to the subsequent machine in the predefined sequence.

At the core of flow shop scheduling lies a fundamental objective: the determination of the most optimal sequence in which jobs should traverse these machines [16]. This optimization endeavor aims to minimize an array of critical performance metrics. These metrics may encompass the minimization of the makespan, which signifies the total time needed to accomplish all jobs, or the reduction of production time and costs. For the purposes of this research, the central focus within the domain of flow shop scheduling is the minimization of the time required to complete a specific job or manufacture a product. This specific metric is commonly referred to as the "makespan," and it takes center stage as a pivotal criterion in our investigation.

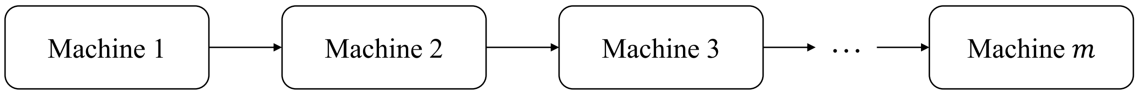


Figure 3.1: Flow shop production

The flow shop production environment, as encountered in practical, real-world scenarios, represents a prevalent and frequently observed manufacturing configuration. It is characterized by a systematic manufacturing methodology in which a series of identical production processes or operations are meticulously executed on a predetermined sequence of machines. This setup, exemplified in the illustrative representation provided in Figure 3.2, features an array of machines, each meticulously engineered and calibrated to perform a specific and well-defined set of tasks or operations. This assembly line-like structure emphasizes the efficiency and orderliness of production.

The flow of work within the flow shop follows a precisely orchestrated sequence. The output generated by one machine seamlessly transitions as the input for the subsequent machine in the predetermined sequence. This intricate choreography of tasks continues unabated until the final product, whether it be an assembled product or a component, is brought to fruition. The hallmark of flow shop production is its strict adherence to this unvarying sequence of tasks, underscoring the systematic and highly organized nature of the manufacturing process.

It's worth emphasizing that tackling the flow shop scheduling problem presents a formidable computational challenge. This complexity becomes particularly pronounced when dealing with three or more machines, denoted as $m \geq 3$. In fact, this problem is officially classified as strongly NP-hard [19], signifying its position among the most computationally demanding problems in the realm of optimization.

In the context of flow shop scheduling, the production process commences with all machines (M_i) and jobs (J_j) simultaneously available and ready to initiate processing without any delays. Here, it's essential to clarify the role of the variables: i corresponds to the indices assigned to machines, ranging from 1 to m, while j represents the indices assigned to jobs, ranging from 1 to n ($i = \{1, 2, ..., m\}$ and $j = \{1, 2, ..., n\}$). This formulation provides a clear and systematic means of representing the various machines and jobs involved in the production process.

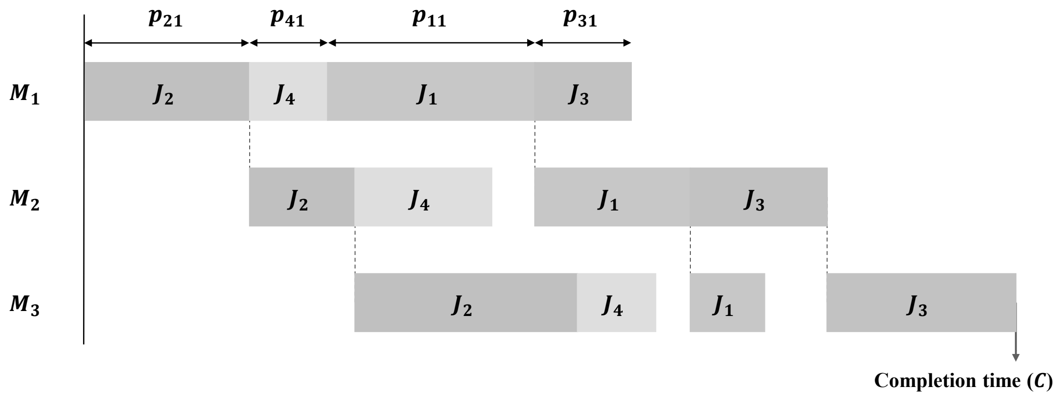


Figure 3.2: Flow shop production with individual processing time

The fundamental goal of the flow shop scheduling problem revolves around the efficient sequencing of n jobs across a sequence of m machines. This sequence entails a predetermined order that all jobs must follow consistently, commencing with M_1, followed by M_2, and so forth until reaching M_m. To effectively manage this process, the concept of processing time plays a pivotal role. Each task executed on a specific machine has a unique processing time, denoted as p_{ij}, where (i, j) symbolizes the execution of job j on machine i. In the computation of processing time, it's crucial to account for the preparation time associated with each job. This preparation time, seamlessly integrated into the overall processing duration, ensures a comprehensive understanding of the time required for each job's completion on a given machine.

Importantly, within this flow shop context, there exist no constraints on the capacity of inter-machine buffers. This design feature permits the free buffering of work between machines without any limits. Nevertheless, it's imperative to emphasize that each machine retains the capacity to handle only a single job at any given moment. This limitation underscores the need for a well-structured scheduling strategy that optimally sequences the jobs to minimize makespan and enhance production efficiency.

The central aim of the flow shop scheduling methodology employed in this research centers around the minimization of the total job completion time, formally represented as C. The overarching goal is succinctly expressed through the mathematical formulation:

$$min \quad C$$

In essence, this optimization objective underscores the pursuit of the most efficient sequence for completing all jobs in the flow shop environment. By achieving this objective, production processes become more streamlined and resource-efficient. Formulating a Mixed Integer Program (MIP) can address the objective of minimizing the makespan in a permutation flow shop with any number of machines [41].

To transform this problem into an MIP, several variables need to be defined. The decision variable x_{jk} takes on the value 1 when job j occupies the k-th position in the sequence; otherwise, it's 0. The auxiliary variable I_{ik} represents the idle time on machine i between processing jobs at positions k and $(k+1)$, while the auxiliary variable W_{ik} represents the waiting time for the job in the k-th position between machines i and $i+1$. Importantly, there's a clear connection between the variables W_{ik} and the variables I_{ik}. It's worth noting that minimizing the makespan is essentially equivalent to minimizing the total idle time on the last machine, which is machine m. Therefore, the MIP formulation is as follows [41].

$$
\begin{aligned}
min \quad & \sum_{i=1}^{m-1}\sum_{j=1}^{n} x_{j1}p_{ij} + \sum_{j=1}^{n-1} I_{mj} & \\
s.t. \quad & \sum_{j=1}^{n} x_{jk} = 1 & k = 1, ..., n \\
& \sum_{k=1}^{n} x_{jk} = 1 & j = 1, ..., n \\
& I_{ik} + \sum_{j=1}^{n} x_{j(k+1)}p_{ij} + W_{i(k+1)} - W_{ik} & \\
& -\sum_{j=1}^{n} x_{jk}p_{(i+1)j} - I_{(i+1)k} = 0 & k = 1, ..., n-1; i = 1, ..., m-1 \\
& W_{i1} = 0 & i = 1, ..., m-1 \\
& I_{1k} = 0 & k = 1, ..., n-1
\end{aligned}
$$

The initial set of constraints dictates that each k-th position must be assigned exactly one job, ensuring that every position is filled. The subsequent set of constraints specifies that each job j must occupy precisely one position. The third set of constraints establish the connections between the decision variables x_{jk} and the physical constraints governing the problem. These physical constraints serve to maintain the necessary relationships between the idle time variables and the waiting time variables. Consequently, the makespan minimization problem in an m-machine permutation flow shop is cast as a MIP. In this MIP formulation, the sole integer variables are the binary decision variables x_{jk}, while the idle time and waiting time variables are continuous and non-negative.

The flow shop configuration and the scheduling challenges it presents are integral facets of production and manufacturing systems. It's crucial to recognize that solving these problems optimally entails substantial computational complexity. Consequently, this research endeavor aspires to make meaningful contributions by devising effective methodologies for addressing flow shop scheduling issues, particularly in scenarios where machine failures introduce additional complexity. Through innovative approaches and the integration of ANNs and PPGA, this research seeks to enhance the efficiency and resilience of flow shop production in the face of disruptions.

3.2 Machine failure

In real manufacturing systems, various unexpected issues can disrupt operations, and one of the most challenging ones is machine failures [45]. When a machine suddenly stops working, it causes problems in the carefully planned production schedule. This disruption can lead to delays, longer production times, and increased costs. Trying to find the best solutions for production in large flow shops is a significant challenge. Rescheduling production operations adds another layer of complexity, especially when dealing with unexpected events like machine failures, as shown in Figure 3.3. These disruptions, primarily caused by machine failures, make the processing times longer and make the whole process even more complicated.

The solution to this intricate challenge lies in the practice of production reschedul-

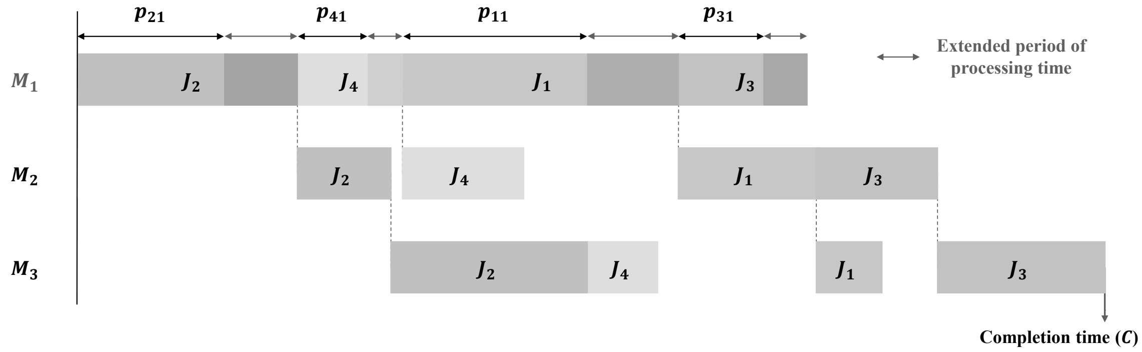


Figure 3.3: Flow shop production with extended period of processing time

ing, a dynamic process adept at accommodating unanticipated modifications or disturbances within the realm of manufacturing. Fundamentally, production rescheduling entails the complex task of strategically reordering job sequences to mitigate the adverse consequences of disruptions [55], with particular emphasis on addressing machine failures. Its primary objective is to ensure the continued achievement of production targets in the face of unforeseen events. Consequently, the imperative for expeditious rescheduling assumes paramount importance in the domain of manufacturing management, functioning as a robust defensive mechanism against the potential issues that these disturbances may engender.

In the special context of flow shops, where machines and jobs are closely connected and work together to achieve production goals, rescheduling when machines might fail is a very tough task. The intricate coordination of jobs and machines, combined with the constant effort to make production better, makes this task extremely hard. This problem is complex and difficult for traditional optimization algorithms to handle. Even methods designed to simplify complex problems encounter challenges when dealing with this issue.

To navigate the labyrinthine intricacies of production rescheduling in flow shops, this research boldly ventures into the realm of hybrid methodologies. This innovative approach artfully amalgamates the prowess of meta-heuristic algorithms, with the genetic algorithm serving as a prime exemplar, and the formidable capabilities of machine learning algorithms, prominently embodied by ANN. The guiding principle underpinning these hybrid methodologies is the harmonious fusion of the strengths intrinsic to each approach. The result is a rescheduling process that emulates the precision of a well-oiled machine, swiftly adapting to disruptions and optimizing production schedules with finesse. Furthermore, our findings underscore the compelling need to cultivate methodologies honed specifically for expediting the rescheduling process within the manufacturing sector, an imperative that resonates resoundingly in the contemporary industrial landscape.

Chapter 4

Methodology

This chapter offers a comprehensive overview of the PPGA-ANN designed for production rescheduling. The methodology is structured into two principal stages, namely training stage and implementing stage, both visually represented in Figure 4.1. Subsequent to this introduction, the intricacies of each stage in the ensuing discourse will be proceeded to explain.

In the stage of training, which precedes the production process, the methodology places significant emphasis on training the ANNs to effectively capture and retain rescheduling knowledge. This is accomplished by leveraging rescheduling solutions generated by PPGA based on various machine failure simulation scenarios. By undergoing this rigorous training, ANNs aim to have the capacity of securely store valuable insights and patterns, thereby enabling the generation of highly effective rescheduling solutions.

Upon the initiation of the production process and the occurrence of a machine failure, the stage of implementing is activated. During this stage, the well-trained ANNs are employed to propose suitable rescheduling solutions depending on the specific scenarios of machine failure. When an actual machine failure situation matches the simulation scenarios utilized during training, *Case 1*, the rescheduled production sequence can be readily acquired from the solutions derived through the PPGA.

However, in situations where the machine failure scenario deviates from the simulation scenarios deployed in the trained ANNs, referred to as *Case 2*, the solution generated by the ANNs assumes the role of the initial solution into PPGA. By in-

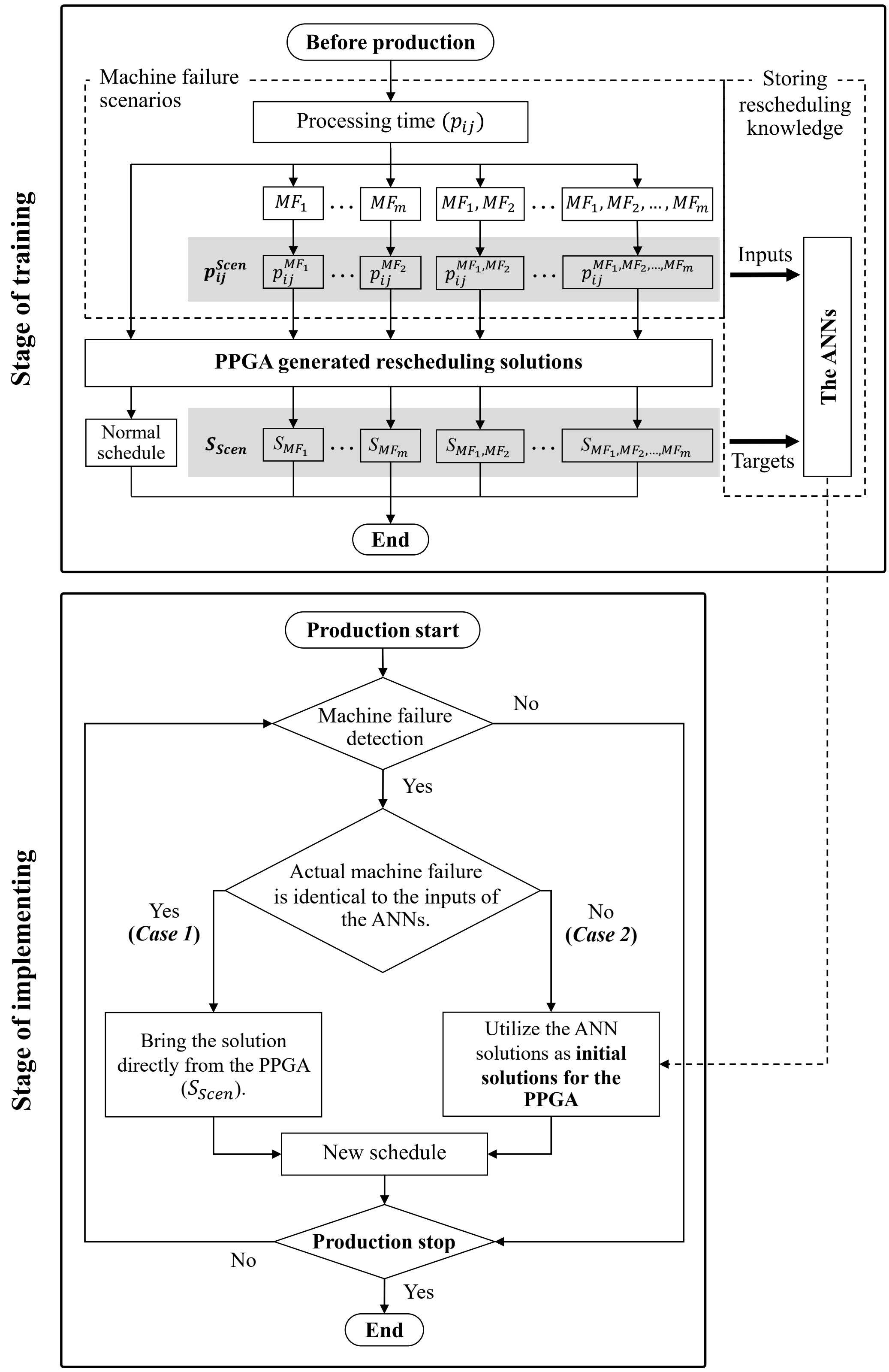


Figure 4.1: The proposed PPGA-ANN methodology

corporating the valuable knowledge acquired from the ANNs, PPGA commences its optimization process from an informed starting point. The seamless incorporation of the PPGA and the ANNs enables a rapid search and optimization procedure, facilitating the identification of rescheduling solutions that approach optimality in a significantly decreased computational time.

The subsequent sections meticulously elucidate the proposed PPGA-ANN. Section 4.1 comprehensively expounds on the stage of training, encompassing the simulation of diverse machine failure scenarios, the utilization of PPGA for generating rescheduling solutions, and the effective storage of rescheduling knowledge by the ANNs. Additionally, Section 4.2 sheds light on the stage of implementing, which entails the proficient utilization of the initial solutions generated by the ANNs, as well as the seamless integration of the rescheduling knowledge obtained from the ANNs into PPGA.

4.1 Stage of Training

The stage of training is executed with the objective of generating valuable knowledge by employing the rescheduling solutions derived from PPGA across a diverse range of simulation scenarios involving machine failure. This knowledge is subsequently preserved within the ANNs through a rigorous training process conducted prior to the actual commencement of production. Section 4.1.1 elaborates on the specific details associated with the simulation scenarios. Moreover, Section 4.1.2 provides an in-depth exposition on the utilization of PPGA for generating the rescheduling solutions. Lastly, Section 4.1.3 offers a comprehensive elucidation of the ANNs employed for effectively storing the acquired knowledge.

4.1.1 Machine failure scenarios

This research aims to investigate the disruptive occurrence resulting from machine failure and its consequential influence on manufacturing processing time. Each machine within the production system is presumed to carry a risk of failure. The processing time required for each job denoted by j when processed on machine denoted by i during regular production is represented by the variable p_{ij}. Here, i ranges from 1 to m, encompassing all machines, while j ranges from 1 to n, encompassing all jobs. The matrix p_{ij} comprises $n \times m$ elements and effectively captures the processing times associated with each job and machine combination.

In the case of machine failure, it is assumed that the processing times will experience an extension. This prolonged processing time is denoted as p_{ij}^{Scen} and assumes that machine overhaul is not required. The set $Scen$ encompasses all potential scenarios of machine failure. Each element within $Scen$ represents a distinct scenario characterized by a set that includes the happening of failed machines, with machine i represented by MF_i. Each scenario can involve the failure of one or multiple machines. The generation of scenario members in $Scen$ follows the power set of i, excluding the empty set, denoted as $\mathcal{P}\{i\} - \emptyset$. As a result, the total number of possible machine failure scenarios amounts to $2^m - 1$, where m signifies the total number of machines.

The structure of $Scen$ can be depicted as $\{\{MF_1\}, \ldots, \{MF_m\}, \{MF_1, MF_2\}, \ldots, \{MF_1, MF_m\}, \ldots, \{MF_1, MF_2, \ldots, MF_m\}\}$. Furthermore, the impact of machine failure is measured through the parameter q_i, which quantifies the tolerance level for increased processing times based on the acceptance criteria of individual manufacturers (Q). The value of Q is a positive real number. The increased processing times are shown in equation (4.1).

$$p_{ij}^{Scen} = q_i \times p_{ij} \quad ; \quad 1 < q_i \leq Q \tag{4.1}$$

Let us consider the machine failure scenarios illustrated in Figure 4.2 for the purpose of illustration an example. In this specific scenario, there exist two machines represented as $i = \{1, 2\}$, and a total of five jobs represented as $j = \{1, 2, ..., 5\}$. Within this context, the set of feasible machine failure scenarios comprises $\{MF_1\}, \{MF_2\}$, and $\{MF_1, MF_2\}$. These scenarios are generated by excluding the empty set from the power set, resulting in a total of three scenarios ($2^2 - 1 = 3$).

The extent of machine failure impact is assessed by the parameter q_i, which, in this particular example, assumes a value of two. This value signifies that in the event of failure, the processing time (p_{ij}^{Scen}) will be twice the duration of the normal processing time (p_{ij}). Consequently, if machine 1 experiences failure (MF_1), the corresponding

Processing time (minutes) under machine failure scenarios: p_{ij}^{Scen}

$p_{ij}^{MF_1}$

$q_1 = 2$	$\boldsymbol{M_1}$	M_2
J_1	**18**	11
J_2	**6**	2
J_3	**12**	7
J_4	**10**	20
J_5	**22**	2

Processing time (minutes): p_{ij}

Base	M_1	M_2
J_1	9	11
J_2	3	2
J_3	6	7
J_4	5	20
J_5	11	2

MF_1

MF_2

MF_1, MF_2

$p_{ij}^{MF_2}$

$q_2 = 2$	M_1	$\boldsymbol{M_2}$
J_1	9	**22**
J_2	3	**4**
J_3	6	**14**
J_4	5	**40**
J_5	11	**4**

$p_{ij}^{MF_1,MF_2}$

$q_1, q_2 = 2$	$\boldsymbol{M_1}$	$\boldsymbol{M_2}$
J_1	**18**	**22**
J_2	**6**	**4**
J_3	**12**	**14**
J_4	**10**	**40**
J_5	**22**	**4**

Figure 4.2: Example of machine failure scenarios

processing time ($p_{1j}^{MF_1}$) will be doubled, while the remaining machines continue to operate at their standard processing times.

4.1.2 PPGA for generating rescheduling solutions

The PPGA serves as an enhanced version of the conventional genetic algorithm. The genetic algorithm, inspired by biological evolution processes, has proven to be extraordinarily effectiveness effective at solving problems characterized by extensive search spaces, including manufacturing scheduling problems. As a widely used meta-heuristic algorithm for solving NP-hard problems, it consistently produces exceptional

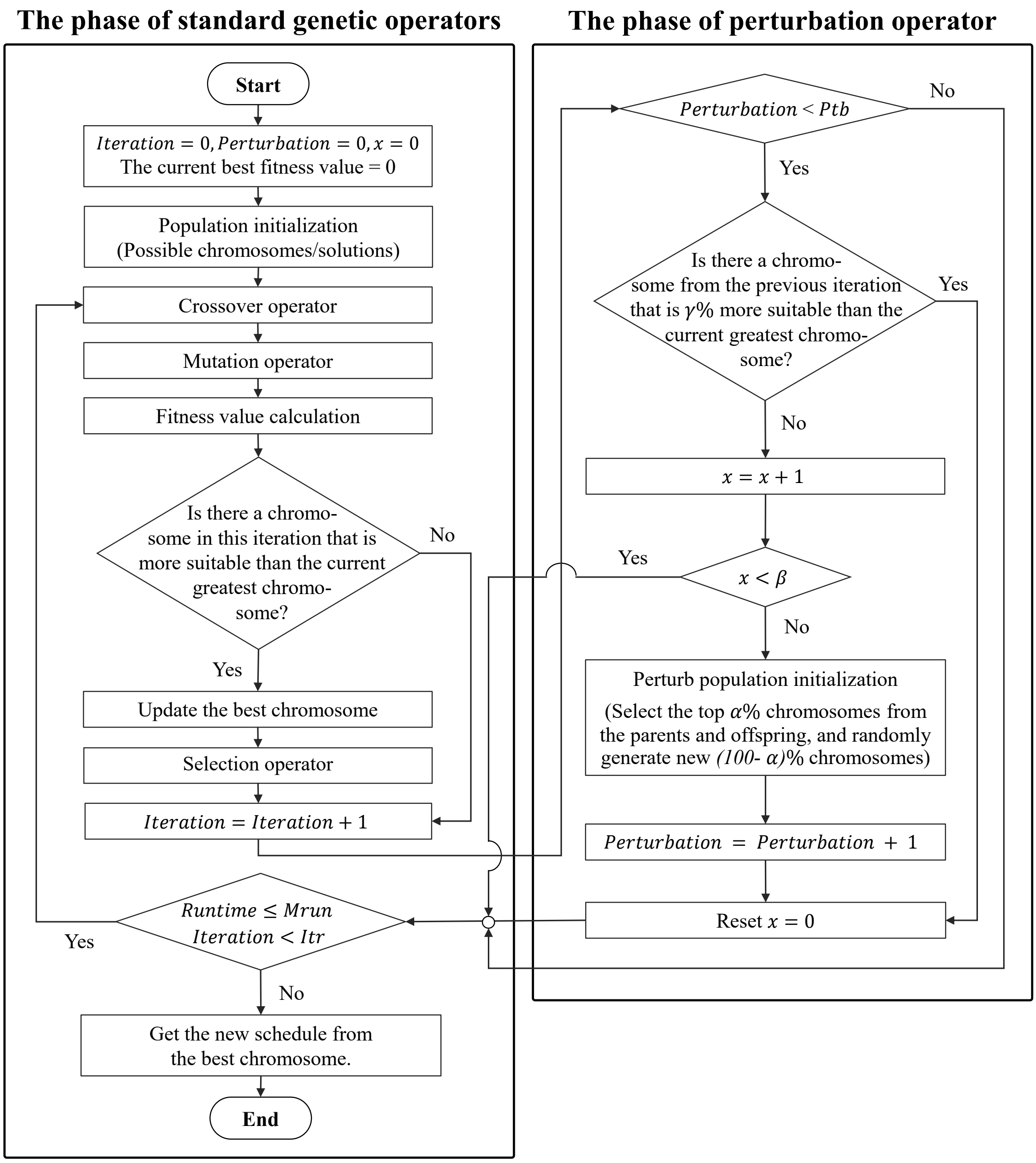


Figure 4.3: Workflow of the proposed PPGA

and efficient results. Furthermore, its computational efficiency makes it well-suited for practical manufacturing decision-making. However, it is important to acknowledge that despite the genetic algorithm's proficiency in investigating solutions in complex spaces via global search, it is prone to being trapped in local optima when conducting local search operations.

This research introduces the PPGA as a solution to the rescheduling problem caused by machine failures in flow shop production. The PPGA is utilized during the stage of training to obtain rescheduling solutions considering various machine failure scenarios. Figure 4.3 illustrates the overview of the PPGA, visually presenting its operation and constituent elements. Moreover, the step-by-step procedure and operations involved in the execution of the PPGA algorithm are outlined in Algorithm 1, which provides the pseudocode for reference. The rescheduling results are initially generated using in a phase of standard genetic operators. However, this research proposes the phase of perturbation operator integrating with the phase of standard genetic operators to avoid the issue of local optimum entrapment during the schedule search process.

PHASE OF STANDARD GENETIC OPERATORS

The first phase of the PPGA is the phase of standard genetic operators. This phase encompasses the utilization of well-established genetic operators, namely population initialization, crossover operator, mutation operator, fitness value calculation, and selection operator, to create the rescheduling solutions. Subsequent paragraphs in this section provide an in-depth elucidation of the specific intricacies and complexities associated with this phase.

Chromosome representation. In a genetic algorithm, a chromosome refers to a potential solution or candidate solution to a problem. Conventionally, the genetic algorithm employs a chromosome representation comprising binary genes, with each gene assuming a value of either 0 or 1. However, such a representation is deemed inadequate for scheduling problems [15]. The chromosome representation utilized within the PPGA is redesigned specifically for flow shop production scheduling. Instead of binary genes, the chromosome now comprises a sequence of jobs. A population consists of a collection of individual chromosomes, representing potential scheduling problem solutions. Each chromosome is characterized by a collection of parameters referred to as genes.

Figure 4.4 visually shows a population containing four chromosomes. Within this

Algorithm 1 Pseudocode of the PPGA

Require: Processing time (p_{ij}), Population size (P), Crossover probability (Cr), Mutation probability (Mu), Number of iterations (Itr), Maximum runtime ($Mrun$), Percentage of improvement (γ), Number of compared iterations (β), Number of perturbations (Ptb)
Ensure: Position of jobs in sequence (S)

```
Iteration = 0
Perturbation = 0
x = 0
while Runtime ≤ Mrun do
  while Iteration < Itr do
    if x < β then
      Population initialization
      Crossover operator
      Mutation operator
      Fitness value calculation
      Selection operator
      Iteration = Iteration + 1
      if fitness value improves ≤ γ% then
        x = x + 1
      else
        x = 0
      end if
    else if Perturbation < Ptb then
      Perturb population initialization
      Perturbation = Perturbation + 1
      x = 0
    else
      x = 0
    end if
  end while
end while
```

context, a specific chromosome fragment, or gene, is represented as "5 3 2 4 1." This sequence signifies that the first position within the gene corresponds to job number "5," followed consecutively by job numbers "3," "2," "4," and "1."

Population initialization. The initial process of the phase of standard genetic operators is characterized by population initialization, where a subset of all possible solutions is generated to form the initial population. This population, consisting of chromosomes, plays a pivotal role in determining the algorithm's effectiveness. Extensive research conducted by Konak et al. [27] underscores the paramount importance

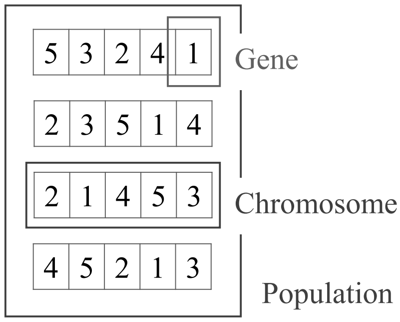


Figure 4.4: Chromosome representation

of maintaining diversity within the population to achieve an approximate global optimum. The presence of diverse individuals is critical to avoid premature convergence, which arises when the algorithm converges prematurely without reaching the optimal solution. To foster diversity and steer the population toward optimality, random initialization is suggested [2]. As a result, the population used in the PPGA is initialized with completely random solutions.

Past research indicates that determining the appropriate population size (P) is best accomplished through empirical experimentation [37]. It is essential to recognize that excessively large population sizes may impede the algorithm's efficiency, while smaller population sizes may fail to provide an adequate mating pool. Consequently, identifying the optimal population size necessitates a systematic trial-and-error process.

Crossover operator. In genetic algorithm, the crossover operator, also referred to as recombination, plays a vital in the algorithm. In order to determine whether the crossover operator should be applied to a given pair of chromosomes, denoted as $pair = \{1, 2, ..., P/2\}$, the algorithm assesses a random probability represented by $ProbC_{pair}$ against a predefined crossover probability, Cr. If $ProbC_{pair}$ is lower than Cr, the crossover operator is executed for that specific pair; otherwise, it is not applied. It is worth emphasizing that the optimal value of Cr depends on the particular problem being addressed and the characteristics of the population. In the process of hyperparameter optimization, Cr is assigned a value ranging from 0.5 to 1.0 [47].

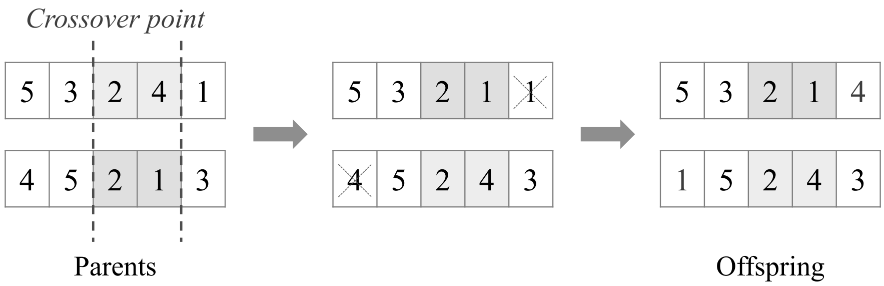


Figure 4.5: Two-point crossover operator

The crossover operator entails combining genes from two parent chromosomes to generate two new chromosomes called offspring. In the PPGA, it utilizes a two-point crossover operator. When chromosomes are encoded with numerical values representing job numbers, exchanging chromosome segments during crossover may introduce duplicate gene fragments in the offspring chromosomes, as illustrated in Figure 4.5 (e.g., job 1 and 4). However, in the context of the flow shop production problem, it is crucial to enforce the constraint that each task position should occur only once, disallowing duplications.

To resolve this concern, this research adopt a strategy where duplicated genes in the offspring chromosomes are randomly replaced with the remaining genes after the selected genes have been swapped. This method is exemplified in the provided example. As a result, the crossover operation is considered complete once the offspring are produced.

Mutation operator. This operator in the genetic algorithm serves the purpose of safeguarding the algorithm against the potential limitation of being trapped in local optima. By introducing genetic diversity between successive generations of chromosomes, the mutation operator fosters exploration of alternative solutions. The specific details of the mutation process within the PPGA are illustrated in Figure 4.6, providing a visual representation of its application.

Within the workflow of the PPGA, the decision regarding the application of the mutation operation to a specific chromosome, denoted as $c = \{1, 2, ..., P\}$, is contingent upon the comparison between a randomly generated probability ($ProbM_c$) and the designated mutation probability (Mu). If $ProbM_c$ is lower than Mu, the

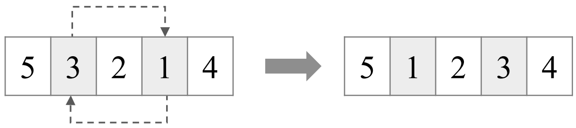


Figure 4.6: Swapping mutation operator

mutation operator is implemented on the corresponding chromosome; otherwise, it is not applied. It is important to acknowledge that while the mutation operator can introduce genetic diversity, it also carries the risk of potentially reducing performance. As a result, the mutation probability is normally assigned a lower value compared to the Cr. The optimal range for Mu is generally between 0.001 and 0.05 [47]. Once a chromosome is chosen for mutation, the swapping mutation technique is utilized, where genes within the chromosome are exchanged by randomly selecting a pair of distinct positions.

Despite the fact that the mutation operator has the potential to induce significant modifications in the structure of a chromosome, there is no assurance that the chromosome's fitness value will necessarily increase as as a result. However, the mutation operator plays a valuable role in the algorithm's overall optimization control [8].

Fitness value calculation. The evaluation of chromosome quality in the PPGA relies on the fitness function. The fitness value directly influences the probability of an individual being selected for genetic operations [12]. A meticulously designed fitness function can expedite convergence and enhance the probability of reaching the optimal solution.

When considering flow shop scheduling, the fitness value (f_c) of chromosome c is computed by evaluating its makespan (C_c). The calculation is carried out using the following equation.

$$f_c = \frac{1}{C_c} \tag{4.2}$$

The primary goal of the PPGA is to identify a chromosome c that maximizes the fitness value f_c, which means minimizing the makespan C_c. This optimization goal corresponds to the minimization of total idle time on the final machine (machine m) in the production line [41]. Using the following equation, the makespan of chromosome c can be determined.

$$C = \sum_{i=1}^{m-1} p_{i1} + \sum_{k=1}^{n-1} I_{mk} + \sum_{j=1}^{n} p_{mj}, \tag{4.3}$$

This equation represents the three components. The first component represents the idle time of the final machine prior to initiating the first job in the given sequence. Here, p_{ik} refers to the processing time on machine i for the k-th position in the sequence of jobs, where $k = \{1, 2, ..., n\}$. The second component represents the idle time of the final machine in the sequence between jobs. I_{ik} represents the idle time on machine i between the execution of the job in the k-th position and the $(k+1)$-th position. The final component represents the total processing time of the final machine, where p_{ij} denotes the processing time of each individual job on machine i.

Nevertheless, because the processing time at the final machine (the final component) remains constant regardless of the job sequence. Therefore, the fitness value calculation can concentrate on optimizing only the first two components in order to minimize the overall value.

Selection operator. The process of selecting chromosomes for recombination is a crucial step in determining the individuals chosen for breeding in the subsequent iteration. In the PPGA, it utilizes the widely recognized and commonly employed roulette wheel selection method from the field of genetic algorithms [46]. This method assigns probabilities to chromosomes based on their fitness values, effectively increasing the likelihood of individuals with higher fitness being selected for recombination. By incorporating roulette wheel selection into the PPGA, it aims to enhance the propagation of promising genetic material and improve the overall quality of the population.

The roulette wheel selection in the PPGA allocates a segment on the wheel to each chromosome based on its relative fitness value (refer to Figure 4.7). This mechanism ensures that chromosomes with higher fitness values have a greater representation on the wheel, thereby increasing their probability of being chosen for reproduction. Consequently, chromosomes with larger fitness values occupy larger segments on the wheel, emphasizing the selection preference towards individuals with superior fitness.

Once the segments are assigned to each chromosome, the roulette wheel is turned, and the chromosome corresponding to the segment where the pointer lands is chosen for reproduction in the subsequent iteration. This selection procedure will proceed until the intended population size P is attained. The probability of choosing a specific chromosome ($Prob\_select_c$) is determined using the following equation:

$$Prob\_select_c = f_c / \sum_{c=1}^{A} f_c, \tag{4.4}$$

The probability ($Prob\_select_c$) of choosing an individual chromosome is determined by dividing its fitness value (f_c) by the sum of fitness values of all chromosomes in the population ($\sum_{c=1}^{A} f_c$), where A represents the total number of parents and offspring. The roulette wheel selection process, depicted in Figure 4.7, employs a population size of four for both parent and offspring populations. In this process, chromosomes with higher fitness values occupy larger sections on the roulette wheel, increasing their likelihood of being chosen for reproduction.

Roulette wheel selection is a straightforward and efficient technique that promotes genetic diversity by giving higher-fitness individuals a greater chance of selection. However, a potential drawback is the possibility of eliminating all offspring generated through the crossover operator. When this occurs, only the parent chromosomes are carried over to the next iteration, which increases the risk of converging to a local minimum. Insufficient diversity in the population presents challenges in identifying globally optimal solutions. To address this, introducing perturbations to the population becomes necessary. These perturbations help maintain genetic diversity, ultimately enhancing the probability of discovering globally optimal solutions.

PHASE OF PERTURBATION OPERATOR

The phase of perturbation operator is significance in the PPGA by aiming to enhance the algorithm's capability to avoid local optimization and exploration of uncharted search spaces. This research introduces the phase of perturbation operator, which is activated when the best chromosome from the previous β iterations shows

	Chromosome	C_c	f_c	$Prob\_select_c$
Parents	5 3 2 4 1	57	1/57 = 0.0175	0.0175/0.1488 = 0.131
	2 3 5 1 4	60	0.0167	0.138
	2 1 4 5 3	52	0.0192	0.120
	4 5 2 1 3	47	0.0213	0.108
Offspring	5 1 2 3 4	59	0.0169	0.136
	3 1 5 2 4	51	0.0196	0.118
	2 4 3 5 1	48	0.0208	0.111
	5 1 2 4 3	60	0.0167	0.138

$$\sum_{c=1}^{8} f_c = 0.1488$$

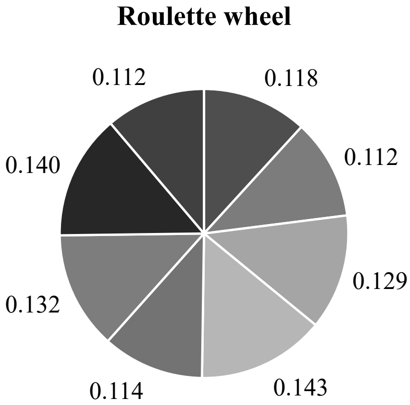


Figure 4.7: Roulette wheel selection operator

no improvement of less than $\gamma\%$. This lack of improvement in the current iteration indicates a lack of substantial progress, possibly implying stagnation within a local optimal area.

Once the phase of perturbation operator is activated, this phase will select the top $\alpha\%$ of the population's chromosomes based on their fitness values in order to preserve the finest chromosomes for the phase of standard genetic operators. $(100 - \alpha)\%$ of the remaining chromosomes are eliminated from the population. Subsequently, new chromosomes are generated at random to replace those that have been eliminated, thereby augmenting population diversity and facilitating exploration of unexplored search spaces. Furthermore, it is possible to set $(100 - \alpha)$ to a value greater than α in order to enhance genetic diversity even further. If the improvement in fitness values over the previous β iterations is still less than $\gamma\%$ compared to the best chromosome obtained from the phase of standard genetic operators, the perturbation operator will persist until the predefined number of perturbations (Ptb) is achieved.

The perturbation operator aims to enhance population diversity, the probability of attaining a near-optimal solution may increase, as it strengthens the capacity to flee local optima. As a result, the standard genetic operators, i.e. the crossover operator, mutation operator, fitness value calculation, and selection operator, are applied to the new population generated by the phase of perturbation operator in order to enhance fitness values.

CRITERIA OF TERMINATION

The criteria of termination hold significance in determining the appropriate cessation point for the search process of genetic algorithm. Various criteria can be employed in a genetic algorithm, with the specific choice depending on the problem at hand and the available computational resources. When choosing the optimal termination criterion, it is essential to establish a balance between solution quality and computation time.

A commonly used termination criterion in genetic algorithms is the maximum number of iterations. This criterion sets a predetermined defined number of iterations for the algorithm to execute before termination. Additionally, the maximum runtime criterion can be utilized to limit the algorithm's execution within a specified time frame. These termination criteria are particularly valuable when time constraints exist or when the algorithm is operating with limited computational resources.

The criteria of termination for the PPGA encompass both the maximum number of iterations (Itr) and the maximum runtime ($Mrun$). The determination of a reasonable maximum number of iterations (Itr) depends on the complexity of the problem and the efficiency of the PPGA, and is established through experimentation. As for the maximum runtime ($Mrun$), it can be defined based on the time constraints imposed by the user.

SOLUTIONS FROM THE PPGA

The PPGA's primary goal is to efficiently determine the optimal solution for a particular rescheduling problem by investigating various search space. In order to accomplish this, the PPGA enhances the phase of perturbation operator, allowing

$$S_{Scen} = \boxed{4 \mid 5 \mid 2 \mid 1 \mid 3}\,, \quad C = 47$$

Figure 4.8: Example solution generated by PPGA

the algorithm to navigate the search space more effectively and overcome the local optima's challenge.

During the stage of training (Section 4.1), the PPGA is used to produce alternate scheduling strategies for a wide range of possible machine failures, then each solution returned by the PPGA reflects a suitable job sequence for that set of specific scenario. As shown in Figure 4.8, the makespan of solution is assessed. These solutions are then preserved for future use by the proposed ANNs, allowing them to leverage the knowledge gained from the PPGA in the subsequent phase of the methodology.

4.1.3 ANNs for storing rescheduling knowledge

The relevant literature indicates that incorporating deep learning techniques into meta-heuristic algorithms can reduce computational time for rapid rescheduling process in actual industry. This research proposes the integration of the PPGA with ANNs to overcome the limitations of previous publications in the field. The ANNs are utilized as a means to store the knowledge obtained from the solutions generated by the PPGA.

Figure 4.9 depicts the process of storing rescheduling knowledge using the proposed ANNs. The architecture of the ANNs is designed to reflect the relationship between the provided rescheduling problems (inputs) and the PPGA-generated solutions (target outputs). Each ANN is constructed to predict the optimal sequence for a given job in a specific rescheduling problem. The segmenting ANNs into multiple models corresponding to various jobs is a strategy employed to enhance specialization, reduce complexity, and improve the efficiency of neural networks in addressing multifaceted problems. Its effectiveness depends on the specific characteristics of the tasks and the availability of sufficient data for each segment. Consequently, the total number of ANNs is contingent upon the number of jobs. Following steps are involved

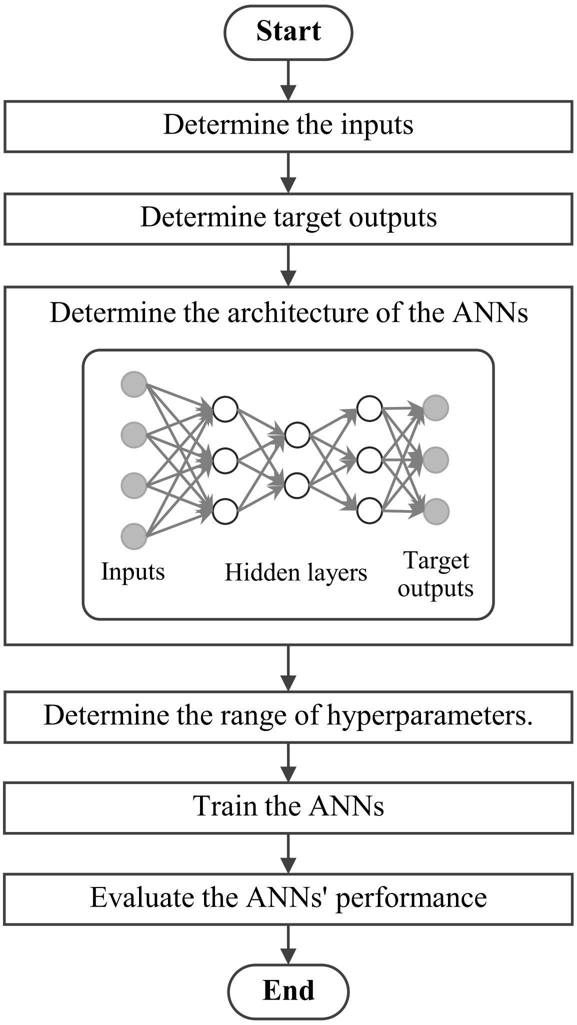


Figure 4.9: Workflow of the ANNs

in preserving rescheduling knowledge in ANNs:

Determine the inputs.

The input layer in ANNs serves as the initial layer responsible for passing the input data to the subsequent layers. Its primary function is to receive and distribute the input data throughout the network for further processing and analysis. Each input layer node represents an input attribute, and the values designated to these nodes correspond to the input attribute values.

The attributes of the input pertain to the processing times of each individual job on each machine. The number of input attributes corresponds to the number of input nodes in the input layer. Specifically, each input layer in this research consists of $n \times m$ nodes, where the number of job represented by n and the number of machine represented by m. Furthermore, the total number of potential machine failure scenarios corresponds to the number of instances in the input data.

To ensure fairness and eliminate bias stemming from varying data scales across the input attributes, all values associated with the input attributes undergo data scaling using a standardization. This data scaling converts each attribute's values into a standard normal distribution. Using the following equation, normalized values for each input attribute can be calculated:

$$y_{ij} = \frac{p_{ij} - \overline{p_{ij}}}{\sigma_{ij}}, \tag{4.5}$$

$$\sigma_{ij} = \sqrt{\frac{\sum(p_{ij} - \overline{p_{ij}})^2}{N}}, \tag{4.6}$$

The equation given by Equation 4.5 demonstrates the process of standardizing the input values in the proposed ANNs. In this equation, y_{ij} denotes the normalized value, while p_{ij} is the primitive input value corresponding to the processing time. The terms $\overline{p_{ij}}$ refer to the mean and σ_{ij} refers to the standard deviation of the processing time. Furthermore, the symbol N refers to the total number of machine failure scenarios. The indices i and j are used to denote the jobs and machines, respectively, where i ranges from 1 to m, and j ranges from 1 to n. To provide further clarity, an example depicting the standardization of input values in the proposed ANNs is presented in Figure 4.10.

Determine the target outputs.

The output layer in ANNs is the ultimate stage responsible for generating output predictions. After processing the input data through the ANNs, the predicted outputs should correspond to the target outputs, which are the PPGA's solutions. As depicted in Figure 4.11(a), these solutions are portrayed as sequences of jobs. However, due to the capability of ANNs to handle only numerical data, it is necessary to transform

$Scen$	p_{ij}									
	J_1M_1	J_1M_2	J_2M_1	J_2M_2	J_3M_1	J_3M_2	J_4M_1	J_4M_2	J_5M_1	J_5M_2
Base	9	11	3	2	6	7	5	20	11	2
MF_1	18	11	6	2	12	7	10	20	22	2
MF_2	9	22	3	4	6	14	5	40	11	4
MF_1, MF_2	18	22	6	4	12	14	10	40	22	4

Mean ($\overline{p_{ij}}$) and standard deviation (σ_{ij}) calculation

$\overline{p_{ij}}$	13.5	16.5	4.5	3	9	10.5	7.5	30	16.5	3
σ_{ij}	5.19615	6.35085	1.73205	1.15470	3.46410	4.04145	2.88675	11.5470	6.35085	1.15470

Normalization of the input value

$Scen$	y_{ij}									
	J_1M_1	J_1M_2	J_2M_1	J_2M_2	J_3M_1	J_3M_2	J_4M_1	J_4M_2	J_5M_1	J_5M_2
Base	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660	-0.8660
MF_1	0.8660	-0.8660	0.8660	-0.8660	0.8660	-0.8660	0.8660	-0.8660	0.8660	-0.8660
MF_2	-0.8660	0.8660	-0.8660	0.8660	-0.8660	0.8660	-0.8660	0.8660	-0.8660	0.8660
MF_1, MF_2	0.8660	0.8660	0.8660	0.8660	0.8660	0.8660	0.8660	0.8660	0.8660	0.8660

Figure 4.10: Data scaling of the inputs in the ANNs

each solution (S_{Scen}) into a binary representation. In this binary format, each binary digit signifies whether a job is assigned to a particular position within the sequence.

Binary matrices of size $n \times n$ are produced to present the target outputs (Figure 4.11(b)), which are formed from the sequences or solutions generated by PPGA (S_{Scen}) in binary format. This is done to highlight the outcomes of the binarization process. In this context, the number of jobs is denoted by n. These matrices include elements with the notation s_{jk}. The value of the variable s_{jk} changes between 0 and 1 depending on whether job j is allocated to position k. For instance, consider a situation with two machines and five jobs ($i = \{1, 2\}$ and $j = \{1, 2, ..., 5\}$, respectively). Therefore, in this instance, there are five positions ($k = \{1, 2, ..., 5\}$). The possible scenarios in this instance include the base case, S_{MF_1}, S_{MF_2}, and S_{MF_1,MF_2}. The position of each job is represented in a 5 × 5 matrix, indicating s_{jk} for five jobs and five positions (i.e., $s_{11}, s_{12}, ..., s_{15}, s_{21}, ..., s_{55}$). As illustrated in Figure 4.11, arrows are used to demonstrate the links between the primitive solutions obtained from the PPGA and the binarized solutions (target outputs) for job 1.

Solution from the PPGA (S_{Scen})

***Scen* = Base**

Sequence	**1**	**2**	**3**	**4**	**5**
Job	4	5	2	1	3

***Scen* = MF_1**

Sequence	**1**	**2**	**3**	**4**	**5**
Job	1	4	3	5	2

***Scen* = MF_2**

Sequence	**1**	**2**	**3**	**4**	**5**
Job	2	4	5	3	1

***Scen* = MF_1, MF_2**

Sequence	**1**	**2**	**3**	**4**	**5**
Job	4	5	2	1	3

(a)

Target outputs

ANN 1 (Job 1)

Scen	***j***	s_{1k}				
		s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
Base	1	0	0	0	**1**	0
MF_1	1	**1**	0	0	0	0
MF_2	1	0	0	0	0	**1**
MF_1, MF_2	1	0	0	0	**1**	0

ANN 2 (Job 2)

Scen	***j***	s_{2k}				
		s_{21}	s_{22}	s_{23}	s_{24}	s_{25}
Base	2	0	0	**1**	0	0
MF_1	2	0	0	0	0	**1**
MF_2	2	**1**	0	0	0	0
MF_1, MF_2	2	0	0	**1**	0	0

ANN 3 (Job 3)

Scen	***j***	s_{3k}				
		s_{31}	s_{32}	s_{33}	s_{34}	s_{35}
Base	3	0	0	0	0	**1**
MF_1	3	0	0	**1**	0	0
MF_2	3	0	0	0	**1**	0
MF_1, MF_2	3	0	0	0	0	**1**

ANN 4 (Job 4)

Scen	***j***	s_{4k}				
		s_{41}	s_{42}	s_{43}	s_{44}	s_{45}
Base	4	**1**	0	0	0	0
MF_1	4	0	**1**	0	0	0
MF_2	4	0	**1**	0	0	0
MF_1, MF_2	4	**1**	0	0	0	0

ANN 5 (Job 5)

Scen	***j***	s_{5k}				
		s_{51}	s_{52}	s_{53}	s_{54}	s_{55}
Base	5	0	**1**	0	0	0
MF_1	5	0	0	0	**1**	0
MF_2	5	0	0	**1**	0	0
MF_1, MF_2	5	0	**1**	0	0	0

(b)

Figure 4.11: Transformation each solution from PPGA (S_{Scen}) into binary representation for target outputs. (a) S_{Scen} and (b) Target outputs

Determine the architecture of the ANNs.

The ANNs' architecture comprises three layers: input, hidden, and output layers. The above paragraphs have already described the input and output layers. The hidden layer, which is determined through fine-tuning experiments, can have a positive integer number of nodes. Figure 4.12 presents an illustration of the ANNs' architecture for a production scenario involving five jobs and two machines.

In the illustrated example, the ANNs' input layer comprises ten nodes, representing the processing times of the five jobs ($j = \{1, 2, ..., 5\}$) and two machines ($i = \{1, 2\}$). To predict the positions of the jobs, five distinct ANNs are constructed. Each network generates output nodes, with a total of five nodes dedicated to each job. These output nodes signify the possible positions that a job can occupy within the sequence. In hidden layers, the precise number of nodes is determined through a series of iterative refinement experiments aimed at optimizing the network's overall performance and accuracy.

Determine hyperparameters' range.

Hyperparameters are used to determine the behavior and performance of a ANN. Unlike the model's trainable parameters, hyperparameters are set prior to the training process and remain fixed throughout. They are determined based on domain expertise, empirical experimentation, or established guidelines. These carefully chosen hyperparameters guide the network's architecture, learning rate, regularization techniques, and other important aspects, influencing its overall effectiveness and generalization capabilities.

- Hidden layers: They act as intermediary layers between the input and output layers, performing figuration on the input data to extract pertinent features before forwarding the results to the subsequent layers.

- Hidden nodes: The number of nodes in each hidden layer and the number of hidden layers are hyperparameters that determine the network's capacity and complexity.

Inputs for all ANNs

Scen	p_{ij}									
	J_1M_1	J_1M_2	J_2M_1	J_2M_2	J_3M_1	J_3M_2	J_4M_1	J_4M_2	J_5M_1	J_5M_2
Base	9	11	3	2	6	7	5	20	11	2

(a)

Target outputs

ANN 1 (Job 1)

Scen	*j*	s_{1k}				
		s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
Base	1	0	0	0	**1**	0

ANN 2 (Job 2)

Scen	*j*	s_{2k}				
		s_{21}	s_{22}	s_{23}	s_{24}	s_{25}
Base	2	0	0	**1**	0	0

ANN 3 (Job 3)

Scen	*j*	s_{3k}				
		s_{31}	s_{32}	s_{33}	s_{34}	s_{35}
Base	3	0	0	0	0	**1**

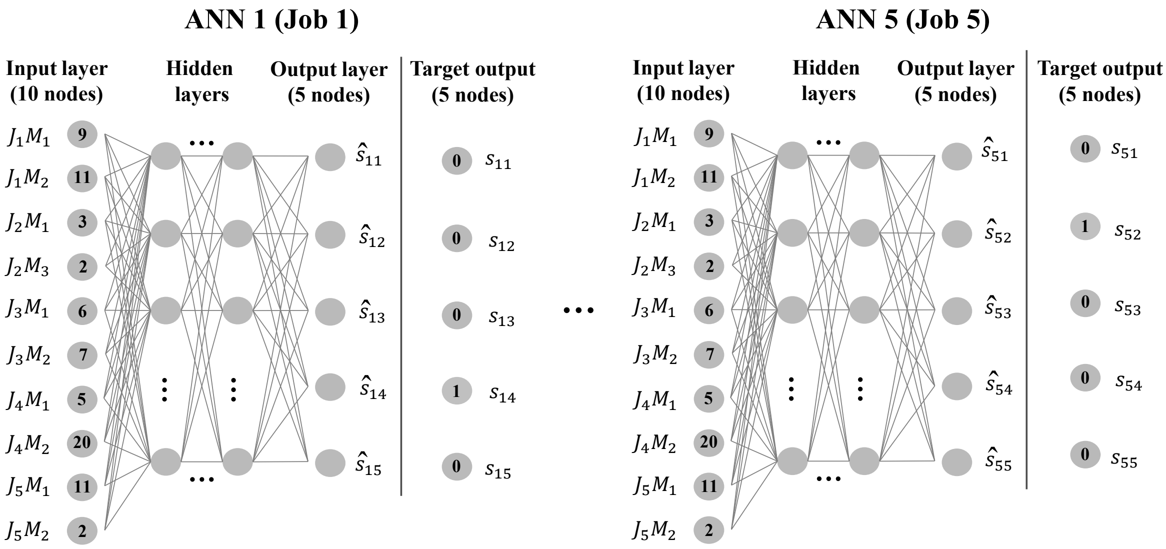
ANN 4 (Job 4)

Scen	*j*	s_{4k}				
		s_{41}	s_{42}	s_{43}	s_{44}	s_{45}
Base	4	**1**	0	0	0	0

ANN 5 (Job 5)

Scen	*j*	s_{5k}				
		s_{51}	s_{52}	s_{53}	s_{54}	s_{55}
Base	5	0	**1**	0	0	0

(b)



(c)

Figure 4.12: Explanation of the ANNs' architecture. (a) Inputs (processing time), (b) Target outputs (the best position in the sequence), and (c) The ANNs' architecture

- Activation function: It is a mathematical operation applied to the aggregated input of nodes within the hidden layers and output layer, producing an output value. Multiple activation functions can be employed in a ANN, enabling the network to capture intricate patterns and relationships within the training data.

- Learning rate: It governs the magnitude of weight and bias adjustments during the training process. A higher learning rate facilitates faster convergence but risks overshooting the optimal weights, while a lower learning rate promotes more precise weights at the expense of longer training time.

- Epochs: The number of epochs specifies how many times the entire training data set is iterated during training. Insufficient epochs may lead to underfitting, where the network fails to grasp underlying patterns, while excessive epochs can result in overfitting, where the network over-adapts to the training data without generalizing well to new instances. Determining the appropriate number of epochs depends on the complexity of the problem and the size of the data set.

- Batch size: Based on the weight and bias updates computed and the average error across each batch within an epoch, the training data is divided into batches. The batch size influences how many samples are utilized during each update phase. The smaller batch size may impede convergence but demands less memory, whereas a larger batch size may expedite convergence but requires more memory resources. The choice of batch size relies on the specific characteristics of the data set and the available computational resources.

- Optimizer: It is an algorithm employed to adjust the network's weights and biases during training with the aim of minimizing the discrepancy or loss between target and predicted outputs. Various optimization algorithms exist, each employing distinct strategies to update the parameters and enhance the network's learning process.

The hyperparameters used in the ANNs are tuned systematically to minimize the loss between the target outputs and the predicted outputs. Through this optimization process, an enhanced model is attained, capable of effectively capturing and retaining the knowledge gained from the PPGA, thereby enabling accurate predictions for novel rescheduling problems.

For ANNs sharing the same architecture across a particular number of jobs and machines, two approaches can be employed regarding the hyperparameter values. Firstly, one may utilize the identical hyperparameter values that have been determined as optimal for a particular ANN (e.g., the ANN for job 1). Alternately, variation can be introduced by deviating from the optimal value parameters of the ANN for job 1. Figure 4.13 elucidates the optimization process of the hyperparameters in the proposed ANNs.

Train the ANNs.

The training of the ANNs involves utilizing machine failure scenarios, which encompass inputs representing the processing times of individual jobs at particular machines, as well as the corresponding optimal production sequences obtained from the PPGA, denoted as target outputs. Notably, all data set derived from the solutions of the PPGA is utilized without partitioning it into separate training, test, and validation sets. This approach is driven by the ANNs' primary goal, which is to capture the intrinsic knowledge embedded within the PPGA-generated data set, rather than making predictions for unfamiliar instances. Moreover, the decision to not separate the data set is justified by the broad distribution of simulation scenarios, encompassing a wide array of potential machine failure scenarios and their corresponding impact on the processing time.

The ANNs employ forward propagation to compute the output of the network for a given input during the training phase. The output of one layer becomes the input of the subsequent layer until the final layer, or output layer, produces the prediction or result of the network. Throughout forward propagation, the input data are multiplied by weights and passed through an activation function at each neuron in the hidden layers. This procedure assists the network in discovering the intricate relationships and patterns within the data.

ANNs then modify the model's weights and biases to minimize a loss function. The loss function measures the difference between the predicted output and the actual target output (ground truth) for a given set of input data, directing the optimization procedure by designating the quantity that the ANNs must minimize to ensure

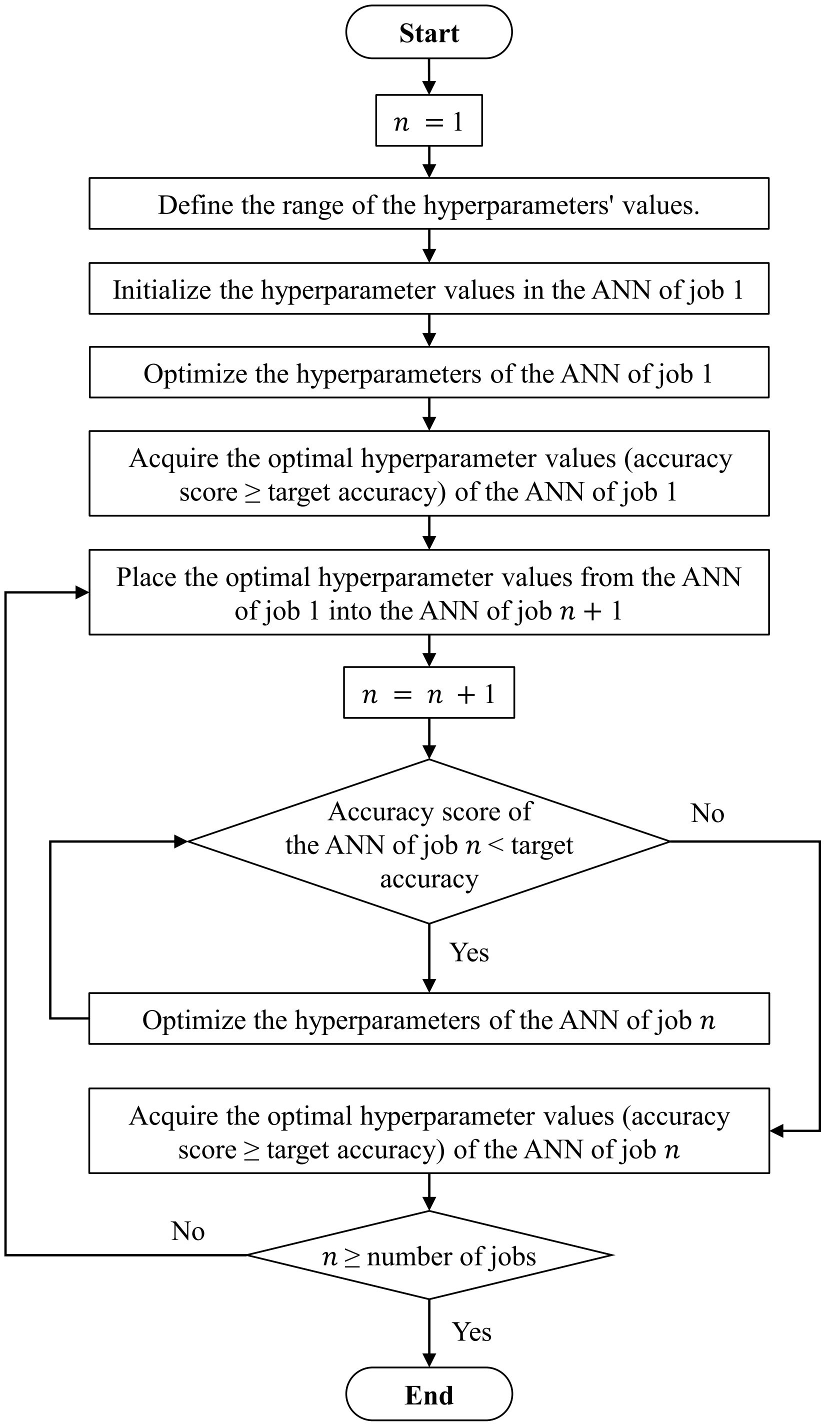


Figure 4.13: The comprehensive hyperparameter optimization process

accurate predictions for the given inputs.

In this research, the *BinaryCrossentropy* loss function is commonly used for optimization during the training procedure in ANN, especially for binary classification tasks. This function quantifies the dissimilarity between the predicted probability distribution and the actual binary labels. It computes the binary cross-entropy loss, which is a measure of how well the model's predictions align with the true binary values. This mechanism effectively steers the ANNs towards making accurate predictions and enhancing the accuracy of binary classification problems, including the specific problem investigated in this research.

Evaluate the ANNs' performance.

The performance evaluation of the trained ANNs relies on the utilization of the *Accuracy* metric. This metric serves as a direct means to assess the prediction capability of the ANN by comparing the target outputs and the predicted outputs of the training data, thereby determining the number of samples that are precisely classified. The performance of the ANNs in relation to the overall classification accuracy can be evaluated by considering the value of this metric.

4.2 Stage of Implementing

The stage of implementing is a significant phase in the proposed PPGA-ANN (Figure 4.1), as it involves implementing the knowledge acquired during the stage of training to make informed decisions regarding rescheduling due to machine failures in the production process. Upon detecting a machine failure during production, which can be classified as either *Case 1* or *Case 2*, this stage is initiated. In *Case 1*, the new schedule can be derived directly from the solutions generated by the PPGA because the machine failure exactly corresponds to the proposed simulation scenarios. In the event that the machine failure situation differs from the inputs used during the training of the ANNs (i.e., a new instance), denoted as *Case 2*, generating a new schedule involves incorporating the solutions obtained from the trained ANNs as initial solutions within the population of the PPGA.

In Section 4.2.1, the procedure of predicting solutions using trained ANNs is explained in detail. Section 4.2.2 elucidates the implementation of the initial solutions derived from the trained ANNs within the PPGA to generate a new schedule.

4.2.1 Initial solutions resulted by the ANNs

In *Case 2*, where the machine failure situations encountered during production deviate from the inputs utilized during the training phase, the ANNs serve the purpose of generating initial solutions for the PPGA. The processing times of the jobs assigned to each machine from new instance are fed into the corresponding ANNs to acquire these initial solutions for a specific machine failure situation, as depicted in Figure 4.14.

For each specific job (j) and position (k) in the sequence, the ANNs generate confidence values, denoted as $\hat{s}_{jk}$, which indicate the suitability of assigning the job to that particular position. Nevertheless, this process may give rise to a situation where multiple jobs exhibit the highest confidence for the same position. Consequently, repetitive positions can emerge in the sequence, thereby violating the fundamental requirement of flow shop production is that each job should be assigned a unique position.

In the provided example, introducing a new instance of machine failure that differs from the training data into the trained ANNs leads to the assignment of multiple jobs (i.e., job 1, job 2, job 3, and job 4) to position 1. Consequently, repetitive positions are observed for multiple jobs in the sequence.

To tackle the repetitive positions, a systematic approach for managing this issue is proposed, as illustrated in Figure 4.15. The process consists of two steps. In Step 1, the jobs' confidences for each position are arranged in a descending order, ensuring a understanding of the job preferences for each position. In Step 2, starting from the first position and moving sequentially towards the last position, the job with the highest confidence is filled to that respective position. However, certain conditions must be satisfied during this assignment process. Firstly, jobs that have already been filled to other positions are excluded from consideration. Additionally, if the difference between the confidences ($\hat{s}_{jk}$) of selected jobs is below a threshold value,

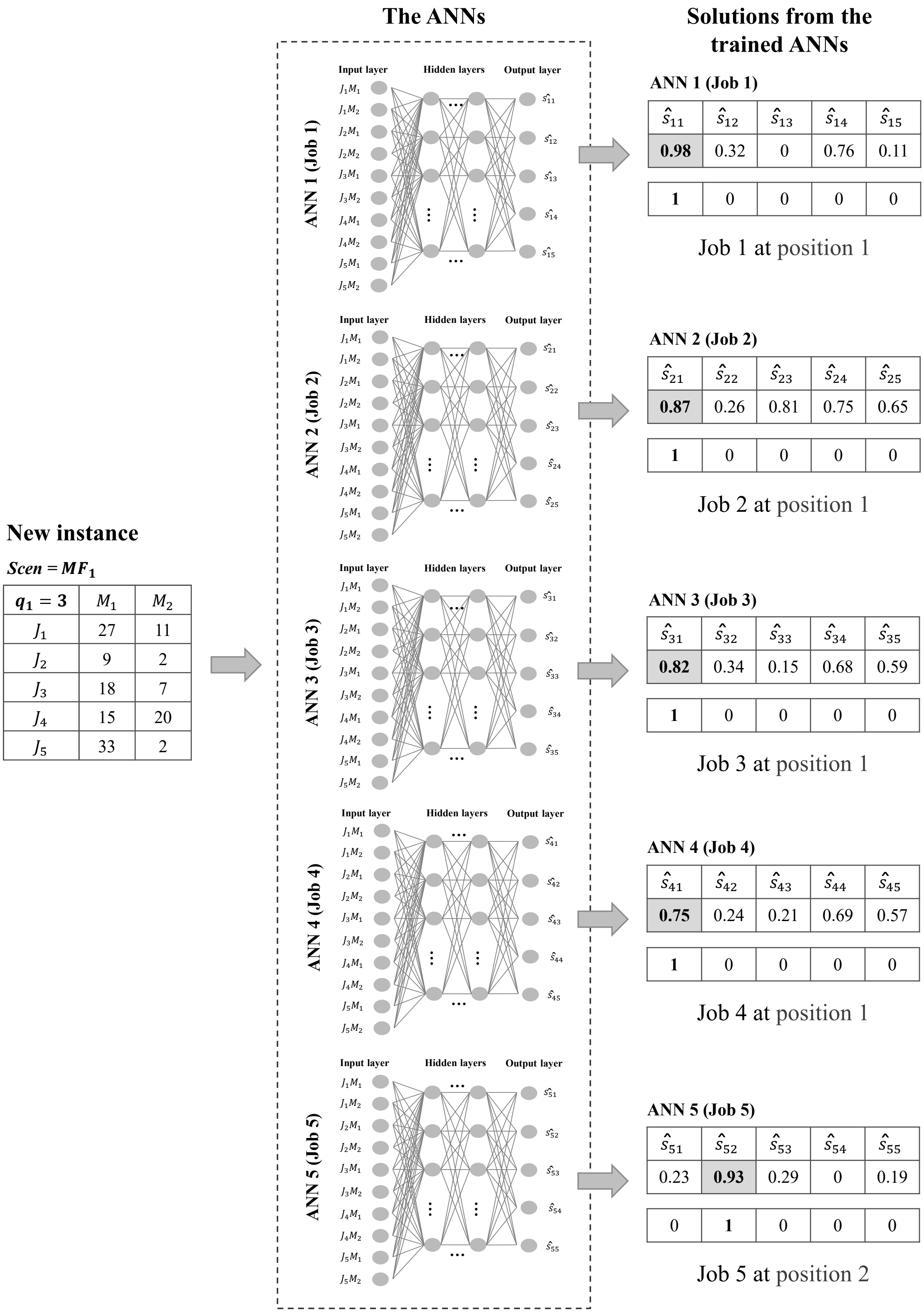


Figure 4.14: Repetitive positions in the sequence

Confidence of Job 1

$\hat{s}_{11}$	$\hat{s}_{12}$	$\hat{s}_{13}$	$\hat{s}_{14}$	$\hat{s}_{15}$
0.98	0.32	0	0.76	0.11

Confidence of Job 2

$\hat{s}_{21}$	$\hat{s}_{22}$	$\hat{s}_{23}$	$\hat{s}_{24}$	$\hat{s}_{25}$
0.87	0.26	0.81	0.75	0.65

Confidence of Job 3

$\hat{s}_{31}$	$\hat{s}_{32}$	$\hat{s}_{33}$	$\hat{s}_{34}$	$\hat{s}_{35}$
0.82	0.34	0.15	0.68	0.59

Confidence of Job 4

$\hat{s}_{41}$	$\hat{s}_{42}$	$\hat{s}_{43}$	$\hat{s}_{44}$	$\hat{s}_{45}$
0.75	0.24	0.21	0.69	0.57

Confidence of Job 5

$\hat{s}_{51}$	$\hat{s}_{52}$	$\hat{s}_{53}$	$\hat{s}_{54}$	$\hat{s}_{55}$
0.23	0.93	0.29	0	0.19

Managing repetitive sequence

STEP 1: Arrange $\hat{s}_{jk}$ for each position by descending order.

STEP 2: Select the job with the largest $\hat{s}_{jk}$ for each position.

Conditions:

1. The jobs that have already been filled cannot be used.
2. If the difference between $\hat{s}_{jk}$ values of the selected jobs is less than δ, then these jobs will be considered to have the same priority for position assignment, which could result in multiple solutions.

Condition 1

Sort	Sequence				
	Position 1	Position 2	Position 3	Position 4	Position 5
Job	**1** (0.98)	**5** (0.93)	**2** (0.81)	~~1~~ (0.76)	~~2~~ (0.65)
	2 (0.87)	3 (0.34)	5 (0.29)	~~2~~ (0.75)	**3** (0.59)
	3 (0.82)	1 (0.32)	4 (0.21)	**4** (0.69)	**4** (0.57)
	4 (0.75)	2 (0.26)	3 (0.15)	**3** (0.68)	5 (0.19)
	5 (0.23)	4 (0.24)	1 (0.00)	5 (0.00)	1 (0.11)

Condition 2

The initial solutions are
2 solutions:

Sequence	1	2	3	4	5
Job	**1**	**5**	**2**	**4**	**3**

and

Sequence	1	2	3	4	5
Job	**1**	**5**	**2**	**3**	**4**

Figure 4.15: Managing repetitive positions in the sequence

denoted as δ that is empirically set as 0.03 in this particular research, these jobs are judged to have equal priority for position assignment. As a result, multiple potential solutions may emerge due to this consideration of equal priority among jobs.

The integration of solutions generated by the ANNs into the initial population of the PPGA is a pivotal step in enhancing the effectiveness of the rescheduling process. This integration serves as a bridge between the machine learning-driven insights of ANNs and the optimization capabilities of PPGA, creating a cohesive and powerful approach to tackling machine failure scenarios in production scheduling.

To maintain diversity and prevent premature convergence within the PPGA's search space, a critical constraint is imposed on the number of initial solutions that can be derived from the ANNs. Specifically, this constraint is set to a maximum value equal to half of the population size ($P/2$). By limiting the number of ANNs-derived solutions in this manner, the methodology ensures that the genetic algorithm retains a degree of randomness and exploration capability during the optimization process. This diversity is crucial for avoiding the entrapment in local optima and promoting the exploration of various solution trajectories.

In cases where the number of solutions obtained from the ANNs exceeds this predefined maximum limit, which can occur when there are repetitive positions in the solutions, an alternative selection criterion based on their makespan (C) is applied. This criterion serves as a pragmatic approach to filter and prioritize the most promising solutions among the surplus provided by the ANNs. By considering the makespan, the methodology focuses on solutions that exhibit the potential for improved production efficiency, aligning with the overarching objective of minimizing makespan during the rescheduling process.

In essence, the constraint and selection criteria implemented in the integration of ANNs into PPGA's initial population strike a delicate balance between preserving diversity and prioritizing solutions with the potential for optimization. This thoughtful approach ensures that the methodology combines the strengths of both machine learning and optimization techniques to navigate the complex landscape of production rescheduling, ultimately leading to the attainment of high-quality solutions in a computationally effective manner.

4.2.2 Utilizing rescheduling knowledge from the ANNs into the PPGA

The incorporation of the PPGA into the implementing stage represents a strategic response to the challenge of handling unforeseen instances in production rescheduling. As production environments are inherently dynamic and prone to various disruptions, it is essential to equip the methodology with adaptability and problem-solving capabilities beyond its initial training data.

In this phase, PPGA takes on a dual role: it leverages the foundational knowledge extracted from ANNs while maintaining its inherent optimization capabilities. This synergistic approach ensures that the methodology is not confined to predefined scenarios but can dynamically respond to real-world machine failure situations that may deviate from the norm.

By commencing the optimization process with initial solutions provided by ANNs, the proposed methodology not only benefits from the ANN's insight but also jump-starts the search for optimal or near-optimal rescheduling solutions. This approach significantly narrows down the solution space and expedites the convergence process, particularly when dealing with unexpected machine failures.

Moreover, the combination of ANNs and PPGA fosters adaptability, a crucial trait in addressing the ever-evolving challenges of manufacturing. As new machine failure scenarios emerge, the methodology can swiftly adapt and evolve, continually improving its performance and effectiveness. This adaptability ensures that the methodology remains a robust and reliable tool for production rescheduling, even in the face of changing operational conditions and unforeseen disruptions.

In summary, the integration of PPGA into the implementing stage, guided by the knowledge acquired from ANNs, epitomizes the methodology's ability to combine machine learning capabilities with optimization prowess. This harmonious fusion not only expedites the rescheduling process but also positions the methodology as a dynamic and adaptable solution for the intricate challenges of modern manufacturing environments.

Chapter 5

Experimentation and Results

The culmination of this research effort is the development of the PPGA-ANN methodology, meticulously designed to address the formidable challenge of rapid rescheduling within the complex landscape of large-scale flow shop production, particularly in the presence of machine failures. As illustrated in Figure 4.1, PPGA-ANN represents a synthesis of machine learning and optimization, harnessing the strengths of both paradigms to expedite rescheduling while striving for optimality. This innovative methodology promises to be a valuable asset for manufacturers operating in dynamic and demanding production environments.

To facilitate the rigorous evaluation of the proposed PPGA-ANN, all experimental procedures have been meticulously orchestrated using the Anaconda platform and implemented through the Python programming language. The choice of this computational environment ensures robustness in the experimentation process, critical considerations when assessing the performance of novel methodologies. Notably, the experiments have been executed on a computer equipped with a 11th Gen Intel(R) Core(TM) i5 CPU running at 2.4 GHz and boasting a substantial 16 GB of RAM, ensuring that computational resources are not a limiting factor in the evaluation of PPGA-ANN's capabilities.

The subsequent sections of this dissertation delve into a comprehensive exploration of the proposed PPGA-ANN, offering a detailed account of the experimental demonstrations and evaluations conducted to validate its effectiveness. Section 5.1 provides a meticulous overview of the process involved in generating machine failure scenarios within the context of flow shop production. This section illuminates the intricacies of simulating machine failures, a crucial aspect of evaluating PPGA-ANN's resilience in real-world manufacturing settings.

Section 5.2 delves into the nuanced domain of hyperparameter tuning. It meticulously outlines the adjustments made to the hyperparameters governing the operation of the genetic algorithm (GA), the PPGA, and the ANNs. This meticulous hyperparameter tuning process aims to fine-tune the behavior of each component, optimizing their interactions within the PPGA-ANN framework.

Finally, in Section 5.3, the performance of the proposed PPGA-ANN undergoes rigorous scrutiny through a comparative analysis with alternative approaches. This evaluation is accompanied by a presentation of the results gleaned from extensive experimentation and a meticulous analysis thereof. The ensuing insights shed light on the efficacy of PPGA-ANN in addressing production rescheduling challenges, offering valuable contributions to the realm of manufacturing optimization and scheduling methodologies.

5.1 Machine Failure Scenarios Generation

The scenarios of the machine failure in this research are evaluated using the benchmark developed by Taillard [51], which has been extensively utilized in various studies, particularly in the context of scheduling problems across various scales. This research focuses on the first data set of the benchmark, which involves a large-scale flow shop production comprising 20 jobs ($n = 20$) and 10 machines ($m = 10$). The processing time (p_{ij}) follows a uniform distribution in the range of $U[1, 99]$.

To simulate machine failures during production, the scenarios are generated according to (4.1). Initially, it begins with a instance from the benchmark data set, where the processing time of all jobs follows a uniform distribution between 1 and 99 minutes. This experiment sets q_i to four different values: 1.5, 2, 2.5, and 3. These values are chosen based on the assumption that the acceptable limit (Q) for changing the processing time, as determined by individual manufacturers, is less than or equal to 3.

With the parameter configuration described above, the total number of scenarios amounts to $4 \times (2^{10} - 1)$ plus one additional case representing the ordinary situation without any failure of machine. Consequently, there are a total of 4,093 scenarios. Each scenario is solved using the proposed PPGA. Subsequently, the ANNs are trained by the scenarios with disrupted processing times as inputs and the corresponding results obtained from PPGA as target outputs. The performance of the ANNs is evaluated based on their ability to generate high-quality solutions when presented with training instances that accurately simulate the perturbed production conditions.

5.2 Modification of Hyperparameters

This research determines the hyperparameters required to resolve the rescheduling challenges in the 20-job, 10-machine flow shop production problem. Extensive experimental research is conducted to determine the optimal values for these hyperparameters. In Table 5.1, the specific hyperparameters utilized by the GA, the proposed PPGA, and the ANNs are enumerated in detail.

The experimental findings demonstrate that, for the proposed PPGA, a population size (P) of 300 chromosomes proves to be suitable for the given scheduling problem. Figure 5.1 illustrates the outcomes obtained in the base case scenario, wherein the PPGA achieves a relatively shorter runtime when employing a population size below 300 chromosomes, albeit resulting in a longer makespan (C). Conversely, for population sizes exceeding 300 chromosomes, both runtime and makespan (C) tend to increase.

Moreover, based on review of the relevant literature reveals, the high crossover probability (Cr) expedites convergence towards a solution, but at the expense of population diversity [47]. Conversely, a high mutation probability (Mu) facilitates exploration of the search space but increases the likelihood of convergence to suboptimal solutions [47]. Therefore, the proposed PPGA uses a crossover probability of 0.80 and a mutation probability of 0.001 based on a trial-and-error method.

Furthermore, termination criteria for the PPGA include the maximum number of iterations (Itr) and the maximum runtime ($Mrun$). Through experimentation, it

Table 5.1: The hyperparameters employed by the GA, the PPGA, and the ANNs

Algorithms	Hyperparameters	Values
GA	Population size (P)	300 chromosomes
	Crossover probability (Cr)	0.8
	Mutation probability (Mu)	0.01
	Number of iterations (Itr)	500 iterations
	Maximum runtime ($Mrun$)	10 minutes
PPGA	Population size (P)	300 chromosomes
	Crossover probability (Cr)	0.8
	Mutation probability (Mu)	0.01
	Number of iterations (Itr)	500 iterations
	Maximum runtime ($Mrun$)	10 minutes
	Percentage of improvement (γ)	0.1
	Number of compared iterations (β)	50 iterations
	Percentage of selection the best chromosomes (α) and randomly creating new chromosomes ($100 - \alpha$)	25 and 75
	Number of perturbations (Ptb)	5 perturbations
ANNs	Hidden layers	5 layers
	Hidden nodes	1^{st} layer: 70 nodes 2^{nd} layer: 60 nodes 3^{rd} layer: 50 nodes 4^{th} layer: 40 nodes 5^{th} layer: 30 nodes
	Activation function of hidden layers	ReLU activation
	Activation function of output layer	Sigmoid activation
	Learning rate	0.001
	Epochs	2000, 3000, 4000, 5000
	Batch size	32
	Optimizer	Adam

has been observed that a value of 500 iterations is appropriate, as the PPGA tends to converge prematurely with fewer iterations and fails to reach the optimal solution. On the other hand, exceeding 500 iterations leads to a longer computational time, even when the optimal solution has already been identified. As for the maximum runtime ($Mrun$), a duration exceeding 10 minutes is considered incompatible with the requirement for rapid rescheduling.

The phase of perturbation operator of the PPGA is activated when the fitness value of the best chromosome in the current iteration shows no improvement exceeding 0.1% (γ) compared to the fitness value of the best chromosome observed in the

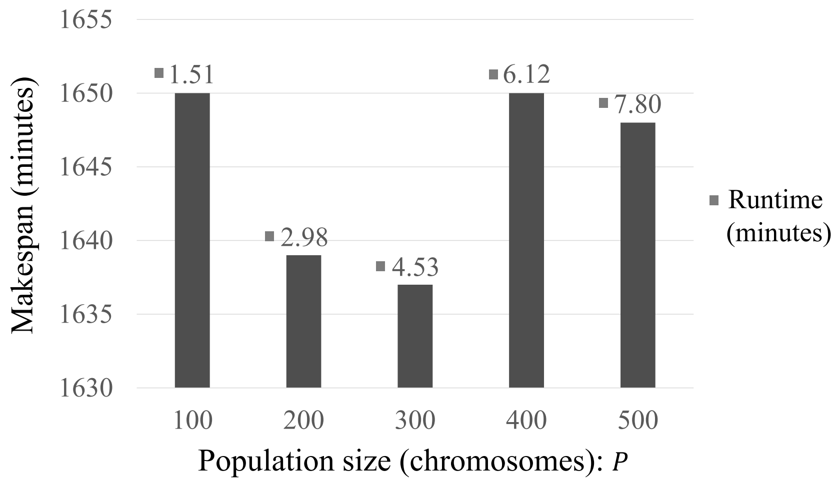


Figure 5.1: Experiment on population size (P)

previous 50 iterations ($\beta = 50$). The specific values for the improvement threshold (γ) and the number of previous iterations considered (β) should be determined according to the user's or manufacturer's preferences. It is worth noting that the improvement threshold (γ) can be expressed in units other than percentages, such as minutes of the makespan or any other unit that individual manufacturers deem appropriate.

In order to preserve the best-performing chromosomes from the phase of standard genetic operators, the top 25 percent of the population size in the present iteration (α) are selected based on their fitness values. The remaining 75 percent of the population ($100 - \alpha$) is generated at random to increase population diversity and investigate unexplored search spaces.

The PPGA applies the perturbation operator up to five times, as determined by the defined maximum number of perturbations (Ptb). However, in certain situations, the maximum number of iterations (Itr) may be achieved before the maximum number of perturbations (Ptb) is achieved. The specific value for the maximum number of perturbations is determined experimentally, taking into consideration factors such as the complexity of the problem and the available computational resources.

The hyperparameters' range used in the ANNs are carefully chosen from a range of values to identify the combination of optimal hyperparameters that can reach high performance, enhance the accuracy, and robustness of the networks. The hyperparameters used in the ANNs remain consistent across all ANNs. Table 5.2 outlines the

Table 5.2: The hyperparameters' range used in the ANNs

Hyperparameters	**Min**	**Max**	**Step**
Hidden layers	1	7	6
Hidden nodes	20	200	18
Learning rate	0.001	0.1	2
Epochs	1000	5000	4
Batch size	32	64	1
Optimizer	Adam, SGD		

range of values considered for each hyperparameter.

The number of hidden layers is tested within the range of 1 to 7 layers, and the number of nodes in each hidden layer is specified between n and $n \times m$, where n represents the number of output nodes (corresponding to the number of jobs), m represents the number of machines, and $n \times m$ represents the number of input nodes. Various learning rates are examined, including 0.001, 0.01, and 0.1, to identify the most effective value. Additionally, the number of epochs is varied between 1000 and 5000 to assess its impact on the ANN's performance. Two options for batch size are considered: 32 and 64. Furthermore, the efficacy of two optimization algorithms, Adam and stochastic gradient descent (SGD), in enhancing the performance of ANNs, is evaluated.

This research extensively investigates the components of ANNs through a systematic exploration of various hyperparameters, such as the learning rate, batch size, and optimizer, employing a trial and error approach. Furthermore, a comprehensive ablation study is conducted to assess the impact of various configurations of hidden layers. The initial experimentation focuses on determining the optimal number of hidden layers, starting with a baseline of two layers for job 1 and training it for a total of one thousand epochs. In subsequent analyses, the influence of the number of nodes in each hidden layer is investigated. Notably, once the accuracy of job 1's ANN exceeds 99%, the configuration of the number of hidden layers and nodes in each hidden layer is defined. Variations in the number of epochs are used to optimize the remaining jobs. This meticulous ablation study provides invaluable insights into the optimal hyperparameter settings and configuration of ANNs, facilitating the generation of precise and efficient rescheduling solutions within the production environment.

However, it is essential to recognize that the applicability of this particular ablation study to other problems or data sets may be limited, as the efficacy of hyperparameter configurations depends heavily on the unique characteristics and complexities of specific problem domains and data sets.

The ANNs involve the construction of 20 separate networks, each corresponding to the processing time data for 20 jobs ($n = 20$) and 10 machines ($m = 10$). These networks consist of 200 input nodes and 20 output nodes. Through experimentation, it has been determined that achieving favorable results involves employing 5 hidden layers with varying numbers of nodes: 70, 60, 50, 40, and 30. The Rectified Linear Unit (ReLU) activation function is utilized in the hidden layers, while the output layer adopts a sigmoid function to map outputs between 0 and 1, facilitating the interpretation of binary values as probabilities.

To achieve high accuracy, the learning rate is set to 0.001, and the number of epochs ranges from 2000 to 5000, depending on the specific ANN for each job, in order to attain high accuracy. Notably, optimal epoch values have been determined, with the ANNs for jobs 1, 2, 4, 5, and 16 achieving high accuracy within 2000 epochs, while the ANNs for jobs 3, 6, 7, 9, 10, 12, 13, 17, and 18 necessitate 3000 epochs. Additionally, the ANNs for jobs 8, 11, 14, and 19 require 4000 epochs, while the ANNs for jobs 15 and 20 employ 5000 epochs. The batch size utilized by the ANNs is 32.

Furthermore, the Adam optimizer, a well-known variant of stochastic gradient descent optimization, has been selected as the optimizer for the ANNs. The Adam optimizer uses moving averages of parameters to estimate gradients, promoting stable learning rates and reducing oscillations. It is computationally effective and ideally adapted for solving massive problems [26].

The values presented in Table 5.1 correspond to essential hyperparameters that play a vital role in governing the learning process. Generally, these values are determined through a trial-and-error iterative search process. While the same set of hyperparameter values may be relevant to other problems, it is crucial to periodically explore alternative values to enhance the algorithm's performance.

5.3 Evaluation of the Experiments

This section evaluates the proposed PPGA-ANN, highlighting on the capabilities of its two primary components: the PPGA and the ANNs. In order to evaluate the efficacy of the PPGA, its results are compared to those of existing algorithms. A comparison of scheduling solutions between the PPGA and the genetic algorithm (GA) allows for the unveiling of the benefits of the phase of perturbation operator in mitigating local optima. Furthermore, the efficacy of utilizing the knowledge held in ANNs for rapid rescheduling is evaluated by comparing the convergence of the proposed PPGA-based initial solutions, derived from the trained ANNs, with those of the PPGA and the standard GA.

5.3.1 The performance of PPGA in the stage of training

During the stage of training of the proposed PPGA-ANN, a novel methodology called PPGA is developed to enhance the conventional GA for production scheduling. The proposed PPGA incorporates a phase of perturbation operator with the aim of optimizing production scheduling performance by minimizing the makespan criterion.

This research employs the exhaustive benchmark data set established by Taillard [51], which includes instances of varying dimensions, to evaluate the efficacy of the PPGA in terms of makespan. The PPGA is executed five times, and the best makespan values are recorded for comparison with the GA, discrete water wave optimization algorithm (DWWO) [65], improved iterated greedy algorithm (IIGA) [10], discrete variant of self-organising migrating algorithm (DSOMA) [9], and hybrid genetic algorithm (HGA) [54].

Table 5.3 presents the makespan results obtained from various approaches using the same benchmark dataset. It is important to note that the PPGA, employed to solve problems of varying sizes, does not integrate the maximal runtime criterion ($Mrun$) since the objective is to obtain the optimal solution for comparison with existing methods. Consequently, the PPGA iterates until reaching the predefined number of iterations (Itr). The results indicate that the proposed PPGA consistently outperforms other algorithms, leading to superior makespan values. In certain

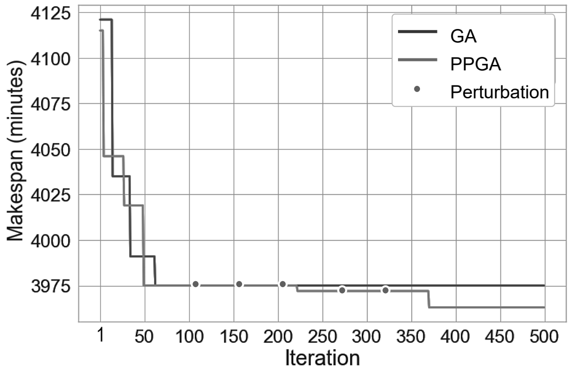
Table 5.3: Makespan comparison of PPGA, GA, DWWO, IIGA, DSOMA, and HGA

Problem	$n \times m$	PPGA	GA	DWWO	IIGA	DSOMA	HGA
Ta011	20x10	**1625**	1655	2044	2011	1698	1955
Ta021	20x20	**2399**	2405	2973	2973	2436	2912
Ta031	50x5	**2740**	2774	3170	3161	3033	3127
Ta041	50x10	**3251**	3256	4274	4274	3638	4251
Ta051	50x20	**4253**	**4253**	6129	6129	4511	6138
Ta061	100x5	**5685**	**5685**	6433	6397	6151	6492
Ta071	100x10	**6113**	6156	8093	8077	7042	8115
Ta081	100x20	**6971**	7046	10727	10700	7854	10745
Ta091	200x10	**11328**	11423	15418	15319	13507	15739
Ta101	200x20	**12489**	12684	19724	19681	15027	20148

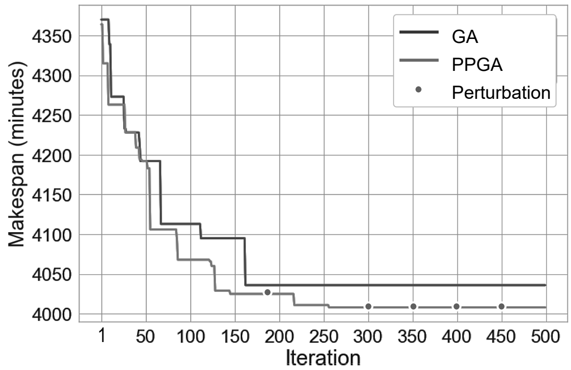
instances, the GA achieves a near-optimal solution comparable to the PPGA, suggesting its proficiency in avoiding local optima. It's noteworthy that the GA used in this research originates from our PPGA, excluding the perturbation operation. Therefore, the performance of both the PPGA and GA relies on the same set of hyperparameters.

In addition, the effectiveness of the phase of perturbation operator in the PPGA is assessed to ascertain its ability to improve population diversity and enhance the likelihood of discovering optimal solutions. This phase ensures the preservation of the best-performing chromosomes from the preceding iteration while introducing randomly generated new chromosomes. The evaluation compares the convergence performance of the PPGA and the GA, both of which utilize the same random seed, for solving flow shop scheduling problems involving machine failure scenarios. The evaluation outcomes are presented in Figure 5.2.

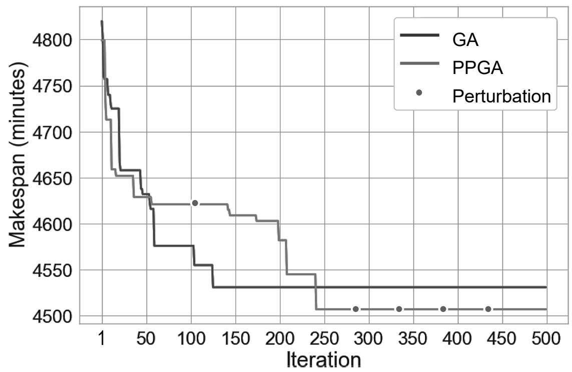
The figure depicted four distinct machine failure scenarios, each representing an extreme case: (a) Machines 1, 5, and 9 ($i = \{1, 5, 9\}$) experience concurrent failures with $q_i = 3$, (b) Machines 2, 4, 6, 8, and 10 ($i = \{2, 4, 6, 8, 10\}$) experience concurrent failures with $q_i = 3$, (c) Machines 3, 4, 5, 6, 7, 8, and 9 ($i = \{3, 4,..., 9\}$) experience concurrent failures with $q_i = 3$, and (d) Machines 1, 2, 3, 4, 5, 6, 7, 8, and 9 ($i = \{1, 2,..., 9\}$) experience concurrent failures with $q_i = 3$. The red points on the graph depict the number of perturbations (Ptb) executed during during the phase of perturbation operator. The outcomes demonstrate that the PPGA, which combines



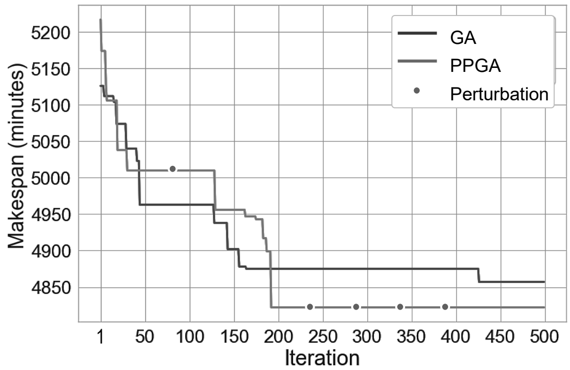
(a) Machine 1, 5, and 9 fail with $q_i = 3$



(b) Machine 2, 4, 6, 8, and 10 fail with $q_i = 3$



(c) Machine 3, 4, 5, 6, 7, 8, and 9 fail with $q_i = 3$



(d) Machine 1, 2, 3, 4, 5, 6, 7, 8, and 9 fail with $q_i = 3$

Figure 5.2: Result comparison between the GA and the PPGA

the phase of perturbation operator with the phase of standard genetic operators, achieves lower makespan (C) values than the GA, indicating superior near-optimal solutions. Notably, the PPGA convergence curves exhibit consistent improvements even in extreme cases, highlighting the efficacy of the phase of perturbation operator in exploring unexplored search spaces.

Furthermore, the comparison between the GA and PPGA in terms of makespan is visually depicted in the box plot (Figure 5.3) using the Ta011 data set from Tailard's benchmark. The box plot provides a concise summary of the distribution of makespan values derived from twenty experimental runs. In the plot, the box represents the interquartile range (IQR), spanning from the lower quartile (Q1) to the upper quartile (Q3). The horizontal line inside the box represents the median value. The whiskers extend from the box to indicate the minimum and maximum values within the range. After a thorough examination of the box plot, it becomes evident that

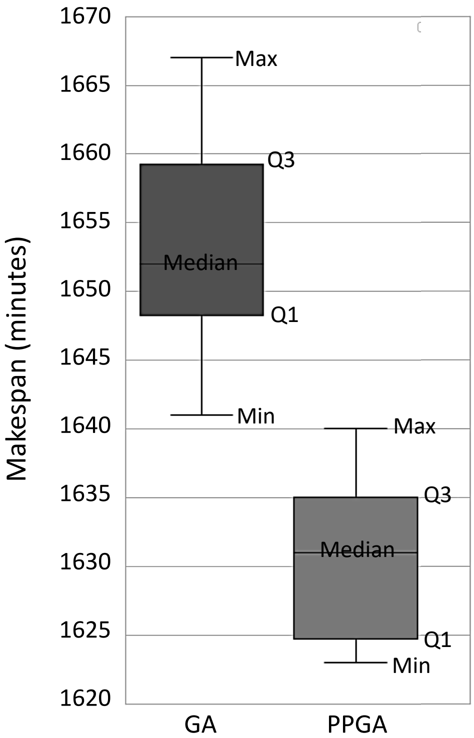


Figure 5.3: The box plot of makespan: a comparison between GA and PPGA

the GA exhibits a median makespan of 1652 minutes, while the PPGA shows a median makespan of 1631 minutes. These median values offer valuable insights into the central tendency of each algorithm's performance. Notably, the PPGA outperforms the GA by demonstrating a lower median makespan, providing compelling evidence of its superior results in terms of makespan. It's worth noting that both the PPGA and GA strive to find the optimal solution, so even a slight change in the median holds significance, as it can influence the lower bound of the near-optimal solution. This discovery underscores the effectiveness of the proposed PPGA in the realm of production scheduling, emphasizing its potential to enhance production efficiency and optimize resource utilization.

5.3.2 Effectiveness of the proposed PPGA-ANN in terms of computational time

The proposed PPGA-ANN introduces the utilization of ANNs during the stage of training to store knowledge obtained from the PPGA. In the subsequent stage of implementing, these ANNs are employed to offer direct solutions or initial solutions for the PPGA, addressing the challenge of machine failures that could lead to suboptimal primary sequence execution.

To assess the performance of the ANNs, loss and accuracy metrics are employed. The numerical results showcase the ANNs' robust memory capabilities and profound comprehension of network relationships, as evidenced by their high accuracy scores and low loss values. The ANNs exhibit an average accuracy score of 99.85% and an average loss of 0.37%, indicating their significant potential in mitigating machine failures and enhancing the PPGA's performance.

The primary objective of the ANNs is to generate initial solutions that expedite the PPGA's search for optimal solutions and enable swift production rescheduling, even when confronted with varying machine failure scenarios not encountered during training.

To evaluate the effectiveness of the initial solutions produced by the ANNs within the PPGA-ANN, new instances are introduced to undergo processing through the ANNs. These instances deviate from the machine failure scenarios encountered during training by introducing a variation in the processing time of the inputs. The extent of this variation is quantified using a percentage change formulation, as depicted by the following equation:

$$Percentage\ change = \frac{|q_{i_{new}} - q_i|}{q_i} \times 100 \tag{5.1}$$

In this equation, q_i signifies the impact of machine failure on the processing time observed in the training scenarios, $q_{i_{new}}$ denotes the new impact of machine failure on the processing time, which has not been employed in the training process. The absolute value operation guarantees that the resulting percentage change value remains positive, regardless of the direction of the change.

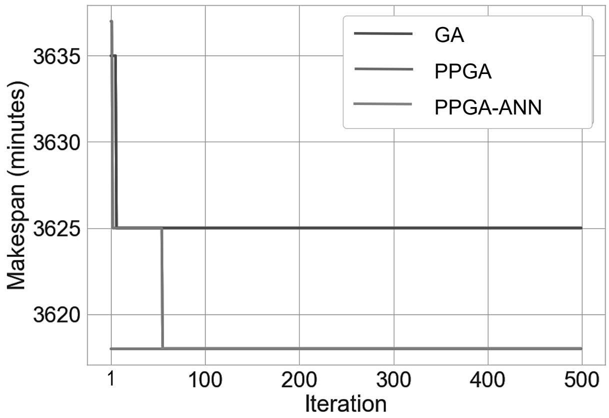
The resulting percentage change value serves as an indicator of the extent to which the processing time has increased or decreased in the new instance compared to the inputs utilized during the training phase. This assessment enables us to measure the effectiveness of the initial solutions and their potential to expedite the rescheduling process in a wider range of scenarios, extending beyond the machine failure scenarios incorporated in the training data.

The evaluation of the PPGA-ANN encompasses the examination of fifteen instances, as illustrated in Figure 5.4. These instances consist of the following categories: (a) three instances that replicate the inputs employed during the training phase (with a percentage change of 0%). Additionally, (b) three instances are generated with a 10% increase in the percentage change of the new instances ($q_{i_{new}} = 3.3$) relative to the inputs ($q_i = 3$). Furthermore, (c) three instances are generated with a 30% increase in the percentage change of the new instances ($q_{i_{new}} = 3.9$) relative to the inputs ($q_i = 3$). Another set of (d) three instances are created with a 50% increase in the percentage change of the new instances ($q_{i_{new}} = 4.5$) relative to the inputs ($q_i = 3$). Lastly, (e) three instances are generated with a 100% increase in the percentage change of the new instances ($q_{i_{new}} = 6$) relative to the inputs ($q_i = 3$). The processing times of the testing instances are normalized using the mean ($\overline{p_{ij}}$) and standard deviation (σ_{ij}) derived from the training set to ensure consistency in the evaluation process.

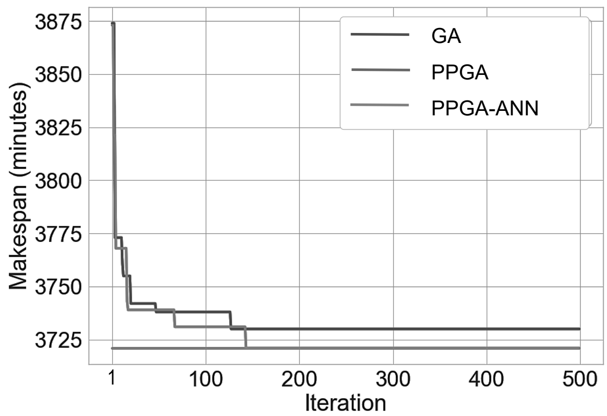
Figure 5.4 illustrates the convergence patterns of the GA, the PPGA, and the PPGA-ANN concerning the percentage change in the new instances. Analyzing these comparisons among instances offers valuable insights into the potential advantages of integrating initial solutions generated by the ANNs in enhancing overall performance.

The results demonstrate that the GA fails to attain optimal solutions for all new instances. Notably, for instances with 0% deviation from the inputs, denoted as (a1), (a2), and (a3), the initial solutions provided by the trained ANNs greatly facilitate the PPGA, resulting in the PPGA-ANN achieving optimal solutions in the first iteration.

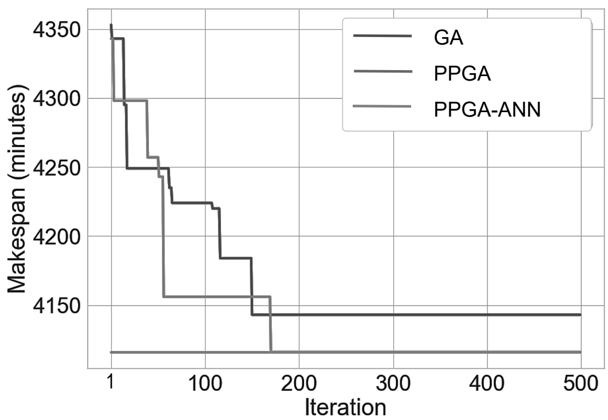
Furthermore, the PPGA-ANN exhibits improved convergence speed compared to the PPGA for instances with 10% and 30% deviation from the inputs. Instances (b1), (b2), and (c1), where finding optimal solutions is relatively straightforward, allow the



(a1) Machine 1 fails with $q_{i_{new}} = 3$



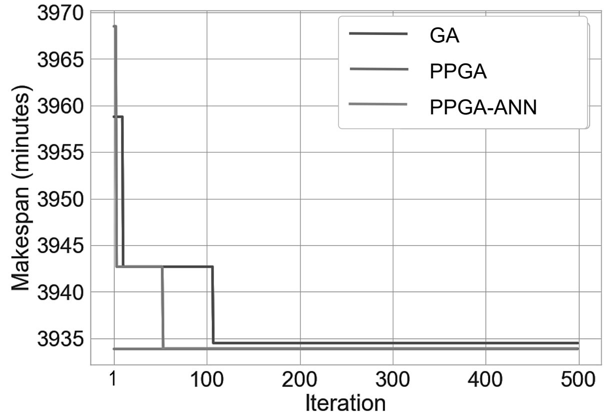
(a2) Machine 2, 6, and 8 fail with $q_{i_{new}} = 3$



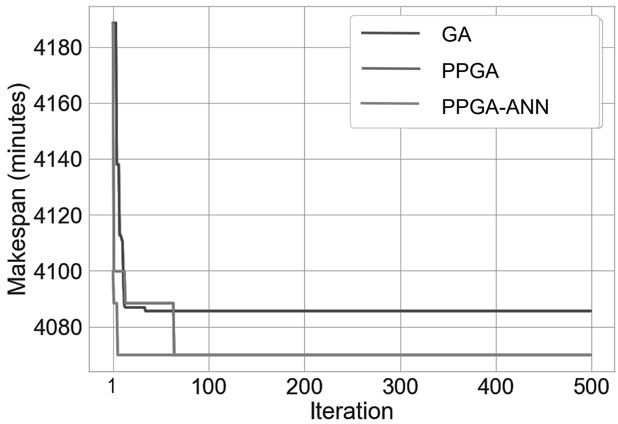
(a3) Machine 1, 3, 5, 7, and 9 fail with $q_{i_{new}} = 3$

(a) The percentage change between the new instances and the inputs is 0%

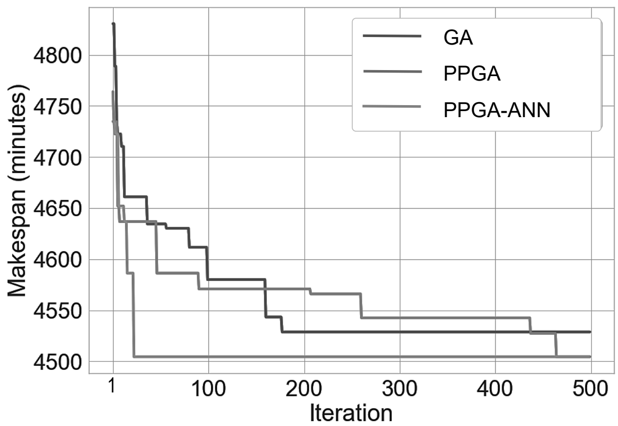
Figure 5.4: Convergence comparison of the GA, PPGA, and PPGA with initial solutions from the trained ANNs (PPGA-ANN)



(b1) Machine 1 fails with $q_{i_{new}}$ = 3.3



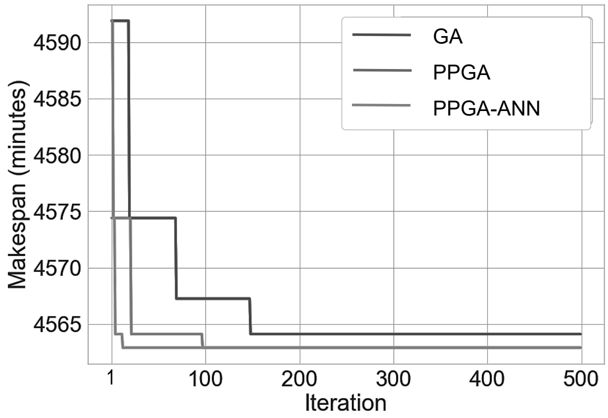
(b2) Machine 2, 6, and 8 fail with $q_{i_{new}}$ = 3.3



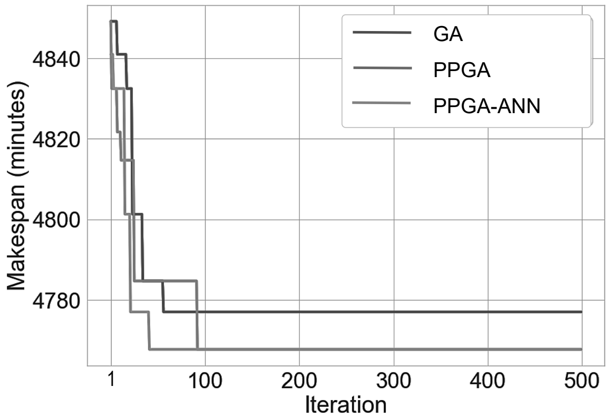
(b3) Machine 1, 3, 5, 7, and 9 fail with $q_{i_{new}}$ = 3.3

(b) The percentage change between the new instances and the inputs is 10%

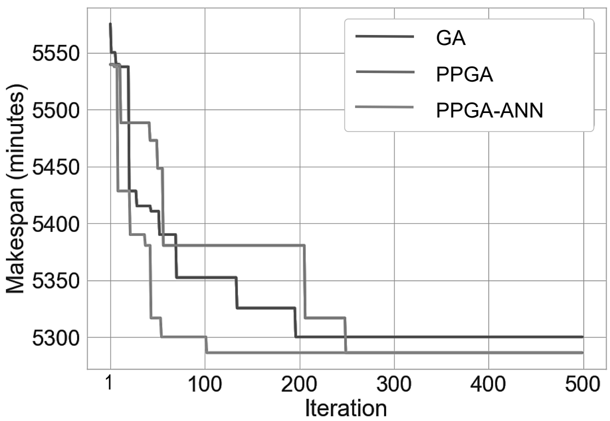
Figure 5.4: Convergence comparison of the GA, PPGA, and PPGA with initial solutions from the trained ANNs (PPGA-ANN) (cont.)



(c1) Machine 1 fails with $q_{i_{new}}$ = 3.9



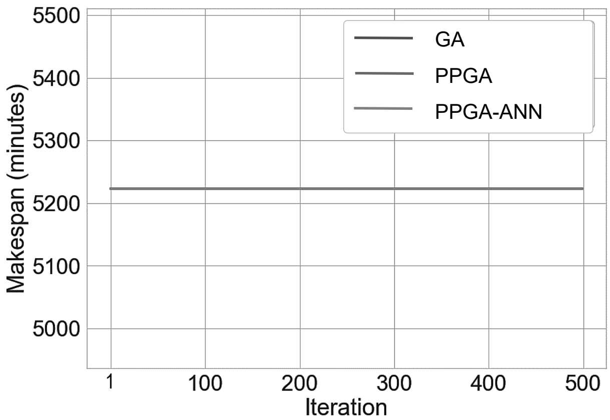
(c2) Machine 2, 6, and 8 fail with $q_{i_{new}}$ = 3.9



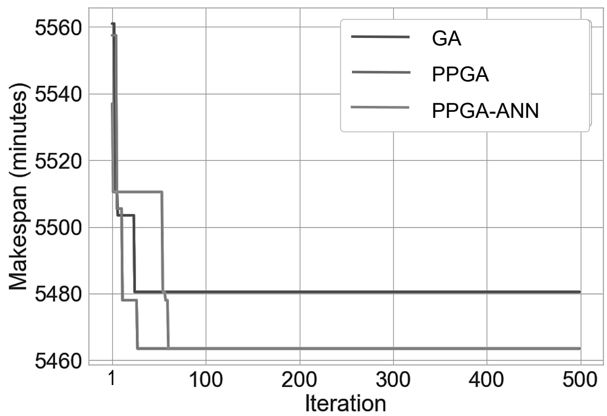
(c3) Machine 1, 3, 5, 7, and 9 fail with $q_{i_{new}}$ = 3.9

(c) The percentage change between the new instances and the inputs is 30%

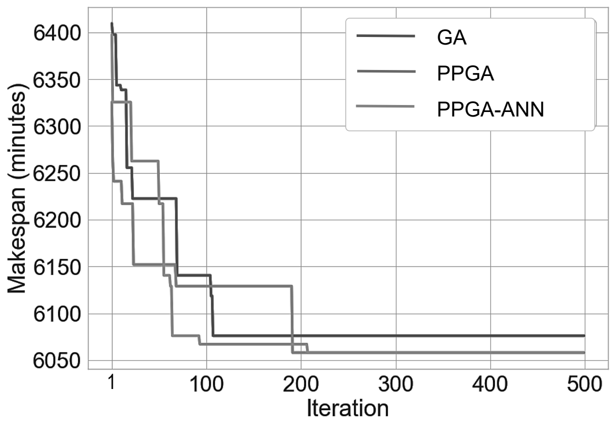
Figure 5.4: Convergence comparison of the GA, PPGA, and PPGA with initial solutions from the trained ANNs (PPGA-ANN) (cont.)



(d1) Machine 1 fails with $q_{i_{new}}$ = 4.5



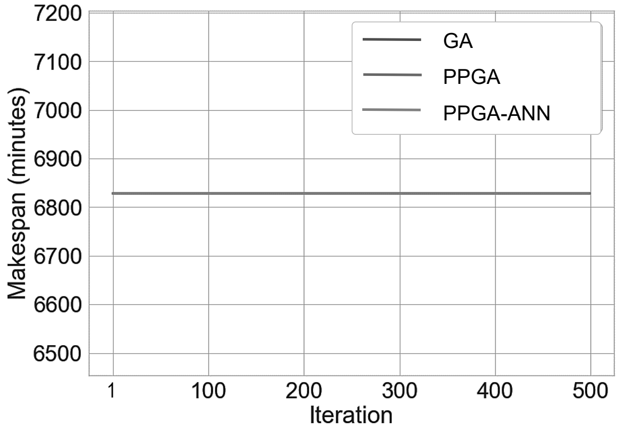
(d2) Machine 2, 6, and 8 fail with $q_{i_{new}}$ = 4.5



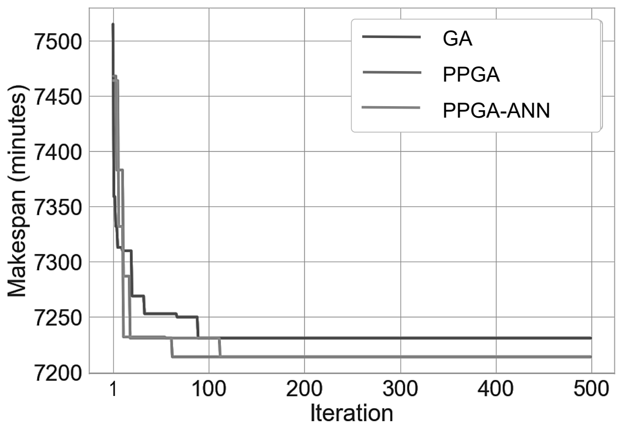
(d3) Machine 1, 3, 5, 7, and 9 fail with $q_{i_{new}}$ = 4.5

(d) The percentage change between the new instances and the inputs is 50%

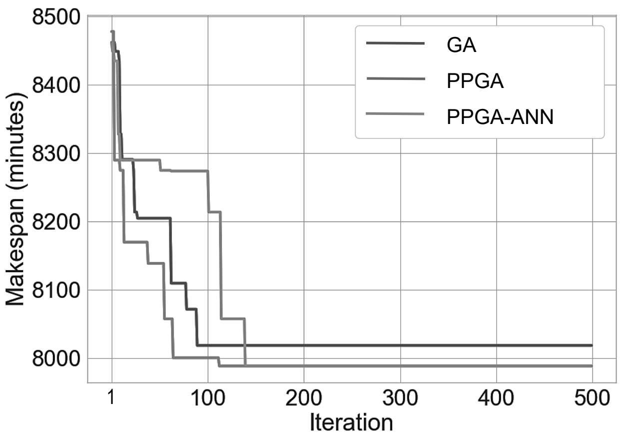
Figure 5.4: Convergence comparison of the GA, PPGA, and PPGA with initial solutions from the trained ANNs (PPGA-ANN) (cont.)



(e1) Machine 1 fails with $q_{i_{new}} = 6$



(e2) Machine 2, 6, and 8 fail with $q_{i_{new}} = 6$



(e3) Machine 1, 3, 5, 7, and 9 fail with $q_{i_{new}} = 6$

(e) The percentage change between the new instances and the inputs is 100%

Figure 5.4: Convergence comparison of the GA, PPGA, and PPGA with initial solutions from the trained ANNs (PPGA-ANN) (cont.)

PPGA-ANN to swiftly reach the optimal solutions with only a few iterations. Additionally, for instances (b3), (c2), and (c3), characterized by a more complex search space for optimal solutions, the PPGA-ANN outperforms the PPGA by achieving the optimal solutions at a faster pace.

However, for instances with a deviation exceeding 50% from the inputs, the initial solutions provided by the trained ANNs prove ineffective. In the case of instances (d1) and (e1), the solutions obtained from the GA, PPGA, and PPGA-ANN are identical since rearranging the job sequence has minimal impact when the new machine failure impact on processing time ($q_{i_{new}}$) is significant and involves only one machine. Instances (d2), (d3), (e2), and (e3) pose challenges in determining the runtime of the PPGA-ANN, even when utilizing initial solutions from the trained ANNs. Nevertheless, the PPGA-ANN can still achieve superior optimal solutions compared to the GA.

The findings suggest that the trained ANNs effectively provide initial solutions for new instances with a deviation of 30% or less. As discussed in Section 4.1.1, this research restricts the acceptable impact on processing time (q_i) within a range that avoids production halt. Consequently, the initial solutions derived from the ANNs can be employed in the PPGA to reschedule previous schedules when the processing time deviation falls within this acceptable range, resulting in faster outcomes compared to the PPGA.

Nevertheless, it is imperative to acknowledge the limitations in generalizing the trained ANNs to entirely novel and unfamiliar instances. The performance of the ANNs may suffer when confronted with instances that exhibit substantial deviations from the training data set. In this research, the initial solutions generated by the ANNs are adopted as the population for the first iteration of PPGA. These initial solutions, depicted by the green line in the first iteration (PPGA-ANN), represent the solutions specifically focused on the makespan aspect of the problem.

In the first iteration of each green line, it is observed that when a new instance exactly aligns with the given inputs (labeled as (a)), the ANNs demonstrate their capability to accurately predict outcomes and yield optimal solutions. However, when the new instance deviates from the given inputs (labeled as (b), (c), (d), and (e)), it is important to acknowledge that the initial solution generated by the ANNs may occasionally fall short of achieving optimality. Nevertheless, even in these cases, the initial solution remains a valuable starting point for the optimization process. In such cases, the complementary of the PPGA becomes evident. The incorporating the PPGA-ANN can effectively leverage the strengths of both algorithms to attain solutions within a reduced computational time.

In practical applications, it is recommended to concurrently execute the PPGA and the PPGA-ANN. This parallel processing strategy is advocated to address the inherent uncertainty associated with the deviation or percentage change in the new impact of machine failure on processing time ($q_{i_{new}}$) in relation to the provided inputs. By running both algorithms simultaneously, it ensures the generation of optimal solutions that outperform the performance of the GA.

The choice between utilizing PPGA or PPGA-ANN to achieve optimal solutions within a reduced computational time relies on the specific instance being considered, taking into account factors such as the complexity of the problem, the unique characteristics of the production environment, and the attributes of the encountered disruption. By executing both algorithms in parallel, manufacturers can leverage the distinct strengths of each approach, enabling them to obtain the most advantageous rescheduling solutions.

Moreover, in order to evaluate the significance of rescheduling, a comparative study was carried out to analyze the makespan (C) of a production process with and without rescheduling using the PPGA-ANN. This investigation encompassed a range of machine failure scenarios, varying from the failure of a single machine to the failure of all ten machines. For each scenario, three sets of failed machines were randomly selected from the entire pool of potential machine failures corresponding to the specific number of failures. The makespans resulting from these three sets of failed machines were subsequently averaged for analysis.

The results presented in Table 5.4 clearly indicate that the introduction of rescheduled production (denoted as Re) leads to significantly shorter average makespans compared to the un-rescheduled production (denoted as Un) in all examined scenarios. Additionally, the runtime for the rescheduling process ranged from 214 to 377

Table 5.4: Comparison of the average makespan (minutes) when machine failure occurs, comparing un-rescheduled (Un) and rescheduled (Re) production.

No. failed machines	$q_i = 1.5$			$q_i = 2$			$q_i = 2.5$			$q_i = 3$		
	Un	Re	Minutes improve	Un	Re	Minutes improve	Un	Re	Minutes improve	Un	Re	Minutes improve
1	1991.67	1910.67	81.00	2483.00	2418.33	64.67	3000.00	2936.00	64.00	3530.00	3462.67	67.33
2	2040.33	1934.33	106.00	2559.33	2457.33	102.00	3086.33	2988.00	98.33	3627.67	3526.00	101.67
3	2172.67	2023.33	149.33	2751.67	2578.33	173.33	3338.00	3128.67	209.33	3934.67	3706.67	228.00
4	2241.67	2133.67	108.00	2877.67	2741.67	136.00	3510.33	3303.00	207.33	4153.33	3910.33	243.00
5	2297.33	2210.33	87.00	2978.67	2867.67	111.00	3660.33	3537.33	123.00	4353.00	4154.33	198.67
6	2329.00	2193.67	135.33	3074.67	2819.00	255.67	3811.33	3473.67	337.67	4557.00	4125.67	431.33
7	2401.33	2298.67	102.67	3173.67	2998.00	175.67	3936.67	3750.67	186.00	4709.00	4452.00	257.00
8	2431.33	2319.67	111.67	3228.00	3066.00	162.00	4014.67	3781.33	233.33	4811.33	4543.00	268.33
9	2453.33	2389.67	63.67	3272.00	3183.67	88.33	4079.67	3962.33	117.33	4898.33	4715.67	182.67
10	2463.00	2463.00	0.00	3282.00	3282.00	0.00	4099.00	4099.00	0.00	4928.00	4928.00	0.00

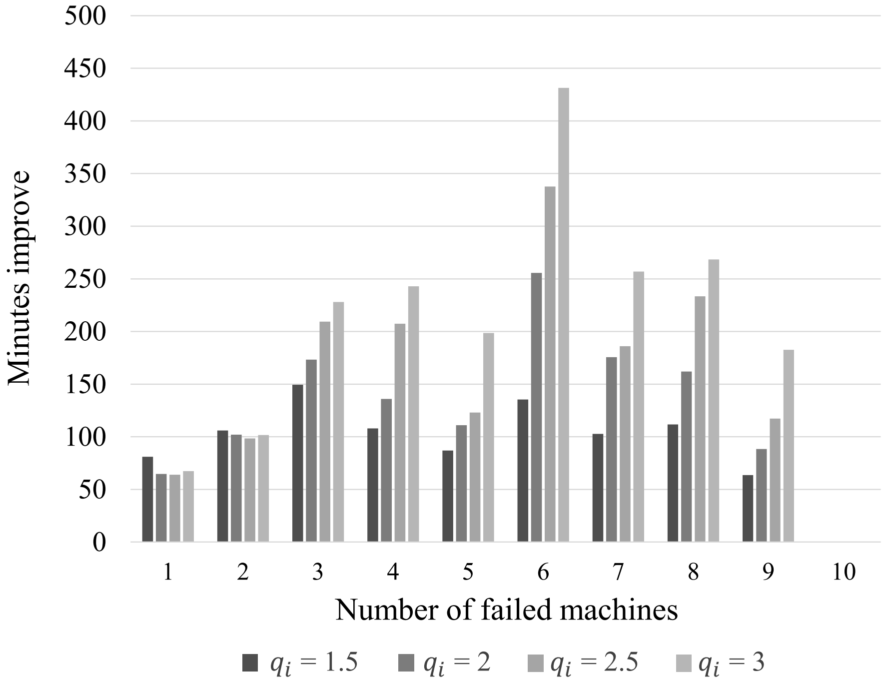


Figure 5.5: The observed in the duration of the average makespan (minutes) when machine failure occurs, comparing un-rescheduled (Un) and rescheduled (Re) production.

seconds, which amounts to less than 6 minutes. These findings underscore the crucial role of rescheduling in enhancing production efficiency by reducing makespans within a limited computational time frame.

The results depicted in Figure 5.5, derived from the data in Table 5.4, further reinforce the effectiveness of rescheduling in improving makespans across the majority of scenarios. In instances where one or two machines experience failure, the enhancement in makespans remains consistent regardless of the impact of processing time (q_i) when a machine fails. However, when three to nine machines encounter failure, there is a notable improvement in makespans, particularly evident in scenarios involving six failed machines and higher values of q_i.

Conversely, when all ten machines fail, the improvement in makespan is observed to be insignificant. This outcome arises due to the uniform distribution of machine failure impact (q_i) among all machines, resulting in an absence of substantial makespan improvement through rescheduling. Although the experiment highlights the criticality of rescheduling when six machines fail, this observation may not necessarily hold true for other data sets.

Chapter 6

Discussion

In this important chapter, a comprehensive analysis is conducted on the results and significant implications arising from this pioneering research. The wider range of applications for the PPGA-ANN methodology is meticulously examined. Additionally, a detailed exploration is undertaken to understand the reasons behind the integration of ANN and PPGA, elucidating their synergistic forces in driving this innovative approach. Simultaneously, the chapter addresses the challenges and limitations encountered during the research journey, highlighting the substantial contributions of this research to the field of study.

6.1 Extending the Scope: Applicability to Structured Problems

A significant insight arising from this research centers on the impressive versatility of the PPGA-ANN methodology. While the primary research focus has been on the complex domain of flow shop production scheduling, it's crucial to recognize that this methodology possesses a natural adaptability that goes beyond its initial purpose. It can easily be reutilized to tackle a wide range of optimization challenges that share a similar structural foundation, especially those that can be formulated as Mixed Integer Programs (MIPs) such as Vehicle Routing Problem (VRP). This inherent flexibility greatly broadens the horizons of the methodology, opening up exciting possibilities for future research efforts. The capacity to repurpose and tailor the approach for structured problems highlights its multifaceted utility, thereby promoting innovation across various domains in the field of operations research and decision-making.

6.2 The Synergy of ANN and PPGA

The use of ANNs, the PPGA, and their combination, PPGA-ANN, in research and problem-solving is driven by their unique strengths and the potential synergies they offer:

6.2.1 Artificial Neural Networks (ANNs):

- Pattern Recognition: ANNs excel in recognizing intricate patterns and relationships within complex datasets. They are particularly valuable in tasks where the underlying patterns are difficult for humans to discern.

- Initial Solutions: ANNs can provide accurate and insightful initial solutions to complex problems. In the context of production scheduling, they can suggest schedules based on historical data and patterns.

6.2.2 Perturbation Population Genetic Algorithm (PPGA):

- The perturbation operator in the PPGA aims to enhance population diversity, the probability of attaining a near-optimal solution may increase, as it strengthens the capacity to flee local optima.

- Meta-Heuristic Optimization: PPGA is a meta-heuristic algorithm, which means it is adept at efficiently exploring solution spaces. It can find near-optimal solutions to complex optimization problems.

- Speed and Adaptability: PPGA can swiftly generate solutions, making it suitable for time-critical manufacturing environments. It adapts well to various problem domains.

6.2.3 PPGA-ANN Combination:

- Complementary Strengths: By integrating ANNs and PPGA, the strengths of both approaches are leveraged. ANNs provide precise initial solutions, while PPGA explores solution spaces efficiently.

- Enhanced Performance: The synergy between ANN's accuracy and PPGA's speed results in a powerful approach for tackling complex production rescheduling challenges.

The use of ANNs, PPGA, and PPGA-ANN is based on their individual capabilities and how they complement each other. ANNs are used for their pattern recognition and initial solution capabilities, PPGA for its optimization and speed, and the combination of both for enhanced performance in production scheduling and rescheduling tasks. This multi-faceted approach is designed to address the complexities and time constraints often found in manufacturing environments.

Central to this research is the harmonious integration of ANNs and the PPGA. This fusion holds profound significance within the domain of production scheduling and rescheduling. ANNs inherently possess the capability to decipher intricate patterns and discern relationships within complex datasets, thus enabling the provision of initial solutions distinguished by their accuracy and insightfulness, particularly within the context of rescheduling. Nevertheless, it is essential to duly acknowledge the limitation of ANNs—their time-intensive training process during the training stage.

The PPGA, a meta-heuristic algorithm, exhibits its complementary strength in this context. It employs meta-heuristic algorithms skillfully to explore solution spaces and produce nearly optimal results within reasonable timeframes. This ability of the PPGA effectively overcomes the time constraint associated with ANNs, providing an efficient way to generate solutions, particularly in time-sensitive manufacturing settings. The coordinated interaction between ANN and PPGA combines the accuracy of machine learning with the flexibility and adaptability of meta-heuristic optimization, resulting in a powerful approach ready to tackle the diverse challenges of production rescheduling.

6.3 Realism of Scenarios

In light of these valuable insights, it is essential to acknowledge the inherent limitations of our meticulously designed scenarios. While these scenarios have been crafted with precision to simulate real-world manufacturing challenges, they inherently fall short in capturing the full spectrum of complexity and unpredictability that characterizes actual production environments. Real-world manufacturing systems operate within dynamic settings replete with variables, uncertainties, and unforeseen events that defy faithful replication in controlled simulations. Thus, there is a resounding call for future research that bridges the gap between simulated environments and the intricate realities of production. The overarching goal is to fortify the applicability and robustness of our methodology.

Furthermore, when considering well-trained ANNs in the context of rescheduling, intriguing insights surface. In the scenario referred to as *Case 2* (see Figure 4.1), where the manufacturing environment operates with firmly established ANN models, the need for the PPGA may potentially decrease. Well-trained ANNs, bolstered by their inherent ability to discern intricate patterns and relationships within complex datasets, might suffice independently for rescheduling tasks. This scenario emphasizes the crucial role of ANNs as a self-sufficient solution, particularly when time constraints allow. It presents a compelling alternative to the combination of ANNs and PPGA. Consequently, the context and urgency of rescheduling decisions emerge as pivotal factors in determining the most suitable methodology, underscoring the dynamic nature of this approach.

In summary, while navigating the intricate landscapes of production scheduling, it is crucial to acknowledge the limitations discovered during this journey with humility. These limitations, rather than hindrances, act as catalysts for innovation and growth. By advocating for bridging the gap between simulated environments and the multifaceted complexities of real-world production, alongside recognizing the potential of well-trained ANNs, an invitation arises for further exploration in the field of production optimization. The mission endures: to reshape the manufacturing landscape, infusing it with efficiency, adaptability, and a profound understanding of the real-world intricacies that characterize modern production environments.

6.4 The Challenge of Finding Global Optima

Throughout this research, a significant challenge emerged in the pursuit of global optima within the extensive domain of large-scale flow shop production problems. The intrinsic computational complexity of these challenges renders the identification of global optima a demanding task within practical time constraints. In recognition of this challenge, the adoption of a structured approach is proposed for forthcoming research endeavors.

The recommended approach involves commencing with small- to moderate-scale problems, where the identification of global optima is more attainable. This initial phase will facilitate the establishment of benchmark solutions for these scenarios, a deeper comprehension of the core issues, and a comprehensive assessment of the performance of the PPGA-ANN methodology. Once a solid foundation and benchmark dataset are established, the research can confidently expand to address larger-scale problems. Essentially, future research will follow a series of progressions, beginning with scenarios where finding global optima is reasonably attainable. This methodical approach will not only aid in benchmarking and validating the methodology but also guarantee its robustness as it expands into the vast domain of large-scale scenarios and understanding the root of the problem.

6.5 Limitations of the Research

While this research represents a significant stride in addressing production rescheduling issues, it's important to acknowledge certain limitations. Chief among these is the time required for training ANNs in the training stage. While ANNs excel in providing accurate solutions, they demand extensive training, especially when dealing with large-scale problems. Therefore, mitigating this limitation necessitates further research and advancements in neural network training techniques, particularly in the context of requiring well-represented training datasets when confronted with new instances.

Furthermore, the careful design of the scenarios is acknowledged. These scenarios were crafted to simulate real-world manufacturing challenges, but it is evident that they may not fully encompass the complexity and unpredictability found in actual production environments. Real-world manufacturing systems contend with dynamic variables, uncertainties, and unforeseen events that are challenging to replicate in controlled scenarios. This recognition emphasizes the necessity for future research dedicated to bridging the gap between simulated environments and the intricate realities of production, thereby enhancing the practicality and reliability of the methodology.

Moreover, to augment the authenticity and statistical significance of this research, the proposal is extended to encompass a broader range of data sources and benchmark datasets in future work. While the primary reliance in this research has been on the Tailard benchmark dataset to validate the methodology, forthcoming investigations should encompass data derived from diverse sources and encompass a variety of benchmark datasets. This approach will yield a more comprehensive evaluation of the methodology's performance across a spectrum of manufacturing scenarios, further substantiating its applicability and reliability. Additionally, the exploration of alternative algorithms to solve the dataset and subsequent comparisons of the methodology's performance with these alternatives can yield valuable insights into its competitiveness and effectiveness.

In conclusion, the research has unveiled the potential of the PPGA-ANN methodology to revolutionize production scheduling and rescheduling. This transformative capacity extends not only to flow shop environments but also to structured optimization problems across various domains. The confluence of ANNs and PPGA adeptly addresses the limitations intrinsic to each approach, rendering it a compelling choice for addressing dynamic manufacturing challenges. Nevertheless, it is duly recognized that ongoing research is imperative to address the limitations identified and to further bridge the gap between simulated scenarios and real-world manufacturing conditions. This journey toward enhanced production scheduling methodologies holds promise in reshaping the landscape of modern manufacturing.

Chapter 7

Research Contributions

The PPGA-ANN methodology introduced in this research offers a multitude of benefits that collectively distinguish it as a significant contribution to the field of manufacturing scheduling. These advantages encompass enhanced optimization capabilities, effective knowledge implementation, and superior overall performance, setting it apart from prior research efforts.

To address the challenge of production rescheduling, it is imperative to apply accurate meta-heuristic algorithms and machine learning techniques within the dynamic context of manufacturing. This research endeavors to provide a substantial contribution by developing an integrated rescheduling methodology tailored explicitly for flow shop production environments. The objective is to bridge the gap between traditional optimization and machine learning methodologies to create a holistic solution.

The core accomplishment of this research lies in the creation of the PPGA-ANN, a groundbreaking integration of the PPGA with ANNs. This innovative approach empowers rapid production rescheduling in flow shop settings, especially in the presence of machine failures. Notably, this methodology offers several distinctive advantages over existing literature. Firstly, the incorporation of the PPGA addresses the limitations of conventional GAs by effectively mitigating the risk of being trapped in local optima, thereby significantly enhancing scheduling performance. Secondly, the seamless integration of ANNs introduces a new dimension to rescheduling by substantially reducing computational time requirements. These elements synergize to deliver an efficient, effective, and adaptable methodology for production rescheduling.

7.1 Methodological Contributions

1. **Integration of ANNs and PPGA:** This research pioneers a novel methodology by seamlessly integrating ANNs with the PPGA in the domain of production rescheduling. This innovative fusion brings together the capabilities of machine learning and meta-heuristic optimization techniques, marking a significant advancement in solving complex scheduling problems. The resulting synergy not only improves the efficiency of rescheduling processes but also lays the groundwork for future developments at the intersection of these two powerful paradigms.

2. **Comprehensive Understanding of Flow Shop Production with Machine Failure Disturbances:** Through meticulous investigation and empirical analysis, this research contributes to a deeper comprehension of the intricate challenges posed by the flow shop production problem in the presence of machine failure disturbances. By shedding light on the complexities inherent in dynamic manufacturing environments, this work addresses critical knowledge gaps. This newfound understanding serves as a valuable resource for both researchers and practitioners seeking to navigate and optimize production processes in the face of unexpected disruptions.

3. **Advancement in Meta-heuristic Algorithms:** The proposed PPGA represents a notable leap forward in the realm of meta-heuristic algorithms. It stands out as a pioneering hybridization of genetic algorithms with the innovative perturbation process. This fusion leverages the strengths of each constituent element to deliver enhanced exploration and exploitation capabilities, greatly improving the algorithm's ability to efficiently tackle the intricate challenges of production rescheduling. The PPGA thus contributes to the continued evolution of meta-heuristic approaches.

4. **Knowledge Advancement in ANN Applicability:** This research extends the boundaries of knowledge concerning the practical applicability and effectiveness of ANNs in the context of production scheduling and rescheduling. By

capturing complex patterns and relationships within production data, ANNs are shown to be versatile tools for addressing real-world scheduling challenges. This insight into their capabilities contributes to the foundational understanding of employing machine learning techniques to enhance production processes.

5. **Exploration of ANN Limitations and Constraints:** While highlighting the strengths of ANNs, this research also critically examines their limitations and constraints in the rescheduling process. This comprehensive exploration offers valuable insights into the trade-offs and considerations involved in integrating machine learning and optimization approaches. Understanding these limitations is essential for making informed decisions about the applicability of ANNs in specific manufacturing scenarios.

6. **Enhancement of Rescheduling Methodologies:** The research goes beyond theoretical developments and offers practical enhancements to existing methodologies and frameworks for production rescheduling. These enhancements provide fresh insights and perspectives on how to improve the efficiency, effectiveness, and adaptability of rescheduling strategies in real-world manufacturing settings. By addressing the complexities of dynamic production environments, this work contributes to the ongoing evolution of rescheduling practices, ultimately benefiting industries striving for optimal production management.

7.2 Practical Contributions

1. **Development of a Tailored Rapid Rescheduling Methodology:** This research pioneers a rapid production rescheduling methodology meticulously designed to cater specifically to the intricacies of flow shop environments grappling with machine failure disturbances. This tailored approach represents a practical solution that directly addresses the challenges prevalent in dynamic manufacturing systems. By streamlining the rescheduling process, manufacturers gain a powerful tool for swiftly adapting to unforeseen disruptions while ensuring production efficiency remains uncompromised.

2. **Implementation and Empirical Validation:** The proposed methodology isn't confined to theory; it has been implemented and rigorously validated through extensive numerical experiments and empirical testing. These real-world experiments provide tangible evidence of the methodology's effectiveness and efficiency in generating rescheduling solutions. The empirical results not only validate its theoretical foundations but also demonstrate its practical applicability in diverse manufacturing scenarios.

3. **Practical Applicability of ANN-PPGA Integration:** This research goes beyond theoretical proposals and demonstrates the practical applicability of integrating ANNs and the PPGA in the realm of production rescheduling. This integration offers a viable approach that marries the strengths of machine learning and optimization techniques. Manufacturers and decision-makers can now leverage this approach to significantly enhance the efficiency and effectiveness of their rescheduling processes, thereby ensuring operational continuity in the face of disruptions.

4. **Insights and Guidelines for Implementation:** The research provides invaluable insights and practical recommendations for the seamless implementation and deployment of the proposed methodology in real-world manufacturing settings. It offers guidelines for model training, solution generation, and the overall integration of the methodology into existing production systems. This practical guidance empowers practitioners with a roadmap for harnessing the methodology's capabilities to their fullest extent.

5. **Practical Implications and Benefits:** Beyond the theoretical realm, the study identifies and thoroughly discusses the practical implications and benefits of adopting the proposed methodology. These benefits include reduced makespan, a critical metric for production efficiency, and improved scheduling performance, particularly in terms of computational time. Understanding these tangible advantages equips manufacturers with the knowledge needed to make informed decisions about implementing the methodology to enhance their production processes.

6. **Enriching Practical Knowledge:** The research makes a substantial and noteworthy contribution to the body of practical knowledge within the field of production scheduling and rescheduling. It offers valuable insights and recommendations tailored to the needs of practitioners and decision-makers in the manufacturing industry. By bridging the gap between theory and practice, this work empowers industry professionals to navigate the complexities of dynamic manufacturing environments, fostering more agile and efficient production management practices.

7.3 Contribution to Knowledge Science

1. **Introduction of a Novel Hybrid Methodology:** The findings of this research constitute a significant advancement in the field of production rescheduling by introducing a pioneering hybrid methodology. This innovative approach, which integrates the PPGA with ANNs, stands as a testament to the ever-evolving landscape of knowledge science. It not only pushes the boundaries of existing methodologies but also ushers in a new era of problem-solving in production scheduling. This contribution extends the repertoire of tools available to researchers and practitioners in the domain, opening doors to novel strategies and solutions.

2. **Enriching Existing Knowledge:** In addition to introducing a novel methodology, this research enriches the existing knowledge base within the field of production rescheduling. It delves deep into the complexities and nuances of rescheduling challenges, shedding light on previously uncharted territories. By conducting a comprehensive exploration of the synergistic effects of combining meta-heuristic optimization (PPGA) with machine learning (ANNs), this work not only expands the theoretical foundations but also enriches the practical insights available to researchers and industry professionals. It paves the way for a more nuanced and informed approach to addressing production rescheduling challenges.

3. **Promising Avenues for Future Research:** The results of this research not only contribute to the present state of knowledge but also illuminate promising avenues for future research in the field of production rescheduling. The success of the PPGA-ANN methodology in reducing makespan and computational time opens doors to further investigations and refinements. Researchers can explore variations of this hybrid approach, adapt it to different manufacturing scenarios, and fine-tune its parameters for optimal performance. Additionally, the practical applications of this methodology beckon further exploration, potentially leading to its integration into diverse manufacturing systems. In this sense, this research acts as a catalyst for ongoing inquiries and developments, underscoring the dynamic and evolving nature of knowledge science in the realm of production rescheduling.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In the contemporary landscape of highly competitive manufacturing, characterized by a relentless pursuit of agility and responsiveness, the development of efficient production scheduling and rescheduling strategies has assumed a paramount role. The ability to rapidly adapt and make well-informed decisions in the face of unforeseen disruptions has become an imperative for organizations striving to maintain a competitive edge. This research rises to meet this challenge by introducing a novel methodology that seamlessly integrates two powerful paradigms: the PPGA and ANNs.

The PPGA-ANN methodology proposed herein represents a significant advancement in the domain of production rescheduling, particularly within the intricate context of flow shop environments marked by machine failure disturbances. Its distinguishing feature lies in its exceptional proficiency in reducing makespan, a pivotal metric in gauging production efficiency, while concurrently curtailing the computational time required for rescheduling. This noteworthy accomplishment finds its roots in the methodology's two foundational components.

First and foremost, the PPGA assumes a central role in the phase of training. As a meta-heuristic optimization algorithm, it excels in traversing vast solution spaces, diligently seeking near-optimal schedules that hold the potential to minimize production downtime. Through the adept application of the PPGA, this methodology ensures that the resultant schedules are not only operationally efficient but also eminently suited for real-world manufacturing scenarios.

Secondly, the strategic incorporation of ANNs into the methodology introduces an innovative dimension to the rescheduling process. These neural networks serve as invaluable instruments for rapid knowledge implementation. By harnessing ANNs, the methodology taps into the predictive capabilities of machine learning, enabling the swift generation of high-quality initial solutions. These initial solutions function as a robust foundation upon which the PPGA can further refine and optimize schedules, ultimately leading to substantial time savings.

In summation, this research delivers a substantial and well-timed contribution to the realm of rapid production rescheduling. Through the introduction of the PPGA-ANN methodology, manufacturers are presented with a dependable solution that not only elevates production efficiency by curtailing makespan but also achieves this feat within an exceptionally abbreviated computational timeframe. The potential advantages of this methodology reverberate across diverse industries, particularly in scenarios characterized by multiple machine failures. Embracing this innovative approach equips manufacturing processes with the requisite tools to excel in an environment where adaptability and efficiency reign supreme. Looking ahead, the PPGA-ANN methodology holds the promise of revolutionizing manufacturers' responses to unforeseen disruptions, ushering in an era characterized by agile and resilient production management.

8.2 Future Work

To enhance the authenticity and applicability of the research problem while opening doors for future investigations, several promising avenues for extension are worth exploring. Expanding the scope of the problem domain to encompass additional complexities can significantly contribute to a more realistic representation of manufacturing scenarios.

One avenue is to consider augmenting the number of jobs and machines involved in the flow shop production, thus creating a more intricate and demanding scheduling environment. By increasing and decreasing the scale of the problem, researchers can simulate production scenarios that mirror real-world manufacturing facilities more closely, where numerous machines and jobs coexist, generating more complex scheduling challenges. Additionally, the consideration of various processing times, i.e., changing q_i values to non-uniform values, adds an addition of realism to the dataset for training ANNs. This move towards non-uniform values will better represent the variable nature of processing times in actual manufacturing processes, where different jobs may have distinct time requirements for processing on each machine.

Furthermore, broadening the analysis to incorporate diverse production settings, such as job shop production, adds another layer of complexity and can lead to valuable insights into the adaptability of the proposed PPGA-ANN methodology in various manufacturing contexts. Job shop production introduces the concept of routing flexibility, where each job can follow its unique sequence of machines. Exploring how the PPGA-ANN performs in such a flexible environment can provide valuable data on its versatility and effectiveness in handling diverse manufacturing scenarios.

Moreover, to further optimize the computational efficiency of the proposed PPGA-ANN methodology, it may be beneficial to introduce a termination criterion. Implementing a criterion that halts the PPGA when it fails to improve the solution within a predefined number of iterations, for example, after 100 iterations, could prevent unnecessary prolongation of computations. This addition would expedite the rescheduling process without compromising the quality of solutions, particularly when searching for near-optimal schedules in complex scenarios.

Additionally, for real-world manufacturing scenarios, it's often known which machines are more prone to failures or disruptions. Researchers can explore the integration of weighted factors into the methodology, giving higher priority or weight to machines with a history of frequent failures. This adjustment can lead to more tailored and efficient rescheduling solutions that account for the specific vulnerabilities of certain machines.

It is imperative to acknowledge that the performance of the proposed PPGA-ANN methodology hinges on the selection of scenarios used to simulate disrupted situations. To address this inherent limitation and enhance the methodology's applicability to unforeseen disruptions, integrating reinforcement learning techniques, such as Q-learning [50], is recommended. This augmentation would empower the algorithm to acquire and adapt to new knowledge as it encounters unforeseen instances that significantly deviate from the initial scenarios. By doing so, the methodology's adaptability and effectiveness in addressing a wide array of rescheduling scenarios can be significantly bolstered, making it a more robust and versatile tool for real-world manufacturing challenges.

In conclusion, the future of this research promises a more comprehensive and adaptable approach to production rescheduling, addressing real-world complexities and uncertainties with innovative methodologies. By delving into these promising avenues, researchers can contribute to the ongoing evolution of manufacturing optimization and its application in various domains.

Appendix

Table 8.1: The base processing time (p_{ij}) from Taillard's benchmark

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
J_1	74	28	89	60	54	92	9	4	25	15
J_2	21	3	52	88	66	11	8	18	15	84
J_3	58	27	56	26	12	54	88	25	91	8
J_4	4	61	13	58	57	97	72	28	49	30
J_5	21	34	7	76	70	57	27	95	56	95
J_6	28	76	32	98	82	53	22	51	10	79
J_7	58	64	32	29	99	65	50	84	62	9
J_8	83	87	98	47	84	77	2	18	70	91
J_9	31	54	46	79	16	51	49	6	76	76
J_{10}	61	98	60	26	41	36	82	90	99	26
J_{11}	94	76	23	19	23	53	93	69	58	42
J_{12}	44	41	87	48	11	19	96	61	83	66
J_{13}	97	70	7	95	68	54	43	57	84	70
J_{14}	94	43	36	78	58	86	13	5	64	91
J_{15}	66	42	26	77	30	40	60	75	74	67
J_{16}	6	79	85	90	5	56	11	4	14	3
J_{17}	37	88	7	24	5	79	37	38	18	98
J_{18}	22	15	34	10	39	74	91	28	48	4
J_{19}	99	49	36	85	58	24	84	4	96	71
J_{20}	83	72	48	55	31	3	67	80	86	62

Bibliography

[1] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi. The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376:113609, 2021.

[2] C. W. Ahn and R. S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6):566–579, 2002.

[3] A. Allahverdi and H. Aydilek. The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of Intelligent Manufacturing*, 26(2):225–237, 2015.

[4] M. Amirghasemi and R. Zamani. An effective evolutionary hybrid for solving the permutation flowshop scheduling problem. *Evolutionary Computation*, 25(1):87–111, 2017.

[5] J. Berlińska and B. Przybylski. Scheduling for gathering multitype data with local computations. *European Journal of Operational Research*, 294(2):453–459, 2021.

[6] B. Çaliş and S. Bulkan. A research survey: review of ai solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, 26:961–973, 2015.

[7] O. Cardin, D. Trentesaux, A. Thomas, P. Castagna, T. Berger, and H. Bril El-Haouzi. Coupling predictive scheduling and reactive control in manufacturing

hybrid control architectures: state of the art and future challenges. *Journal of Intelligent Manufacturing*, 28(7):1503–1517, 2017.

[8] M. Chen, J. Wen, Y.-J. Song, L.-n. Xing, and Y.-w. Chen. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. *Swarm and Evolutionary Computation*, 65:100912, 2021.

[9] D. Davendra and M. Bialic-Davendra. Scheduling flow shops with blocking using a discrete self-organising migrating algorithm. *International Journal of Production Research*, 51(8):2200–2218, 2013.

[10] J.-Y. Ding, S. Song, J. N. Gupta, R. Zhang, R. Chiong, and C. Wu. An improved iterated greedy algorithm with a tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing*, 30:604–613, 2015.

[11] Y.-H. Dong and J. Jang. Production rescheduling for machine breakdown at a job shop. *International Journal of Production Research*, 50(10):2681–2691, 2012.

[12] Y. Du, T. Wang, B. Xin, L. Wang, Y. Chen, and L. Xing. A data-driven parallel scheduling approach for multiple agile earth observation satellites. *IEEE Transactions on Evolutionary Computation*, 24(4):679–693, 2019.

[13] A. Dutta. Reacting to scheduling exceptions in FMS environments. *IIE Transactions*, 22(4):300–314, 1990.

[14] O. Engin and A. Güçlü. A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Applied Soft Computing*, 72:166–176, 2018.

[15] O. Etiler, B. Toklu, M. Atak, and J. Wilson. A genetic algorithm for flow shop scheduling problems. *Journal of the Operational Research Society*, 55(8):830–835, 2004.

[16] A. M. Fathollahi-Fard, L. Woodward, and O. Akhrif. Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept. *Journal of Industrial Information Integration*, 24:100233, 2021.

[17] P. Fattahi, S. M. H. Hosseini, and F. Jolai. A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 65(5):787–802, 2013.

[18] J. Gao, R. Chen, and W. Deng. An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research*, 51(3):641–651, 2013.

[19] M. R. Garey, D. S. Johnson, and R. Sethi. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2):117–129, 1976.

[20] C. Gicquel, L. Hege, M. Minoux, and W. Van Canneyt. A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. *Computers & Operations Research*, 39(3):629–636, 2012.

[21] S. C. Graves. A review of production scheduling. *Operations Research*, 29(4):646–675, 1981.

[22] Z. Jiang, S. Yuan, J. Ma, and Q. Wang. The evolution of production scheduling from industry 3.0 through industry 4.0. *International Journal of Production Research*, 60(11):3534–3554, 2022.

[23] S. M. Johnson. Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1):61–68, 1954.

[24] S. Katoch, S. S. Chauhan, and V. Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126, 2021.

[25] K. Keskin and O. Engin. A hybrid genetic local and global search algorithm for solving no-wait flow shop problem with bi criteria. *SN Applied Sciences*, 3(6):1–15, 2021.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[28] N. Kundakcı and O. Kulak. Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *Computers & Industrial Engineering*, 96:31–51, 2016.

[29] R. Larsen and M. Pranzo. A framework for dynamic rescheduling problems. *International Journal of Production Research*, 57(1):16–33, 2019.

[30] Y. Li, S. Carabelli, E. Fadda, D. Manerba, R. Tadei, and O. Terzo. Machine learning and optimization for production rescheduling in industry 4.0. *The International Journal of Advanced Manufacturing Technology*, 110(9):2445–2463, 2020.

[31] S. Lu, J. Pei, X. Liu, and P. M. Pardalos. A hybrid DBH-VNS for high-end equipment production scheduling with machine failures and preventive maintenance activities. *Journal of Computational and Applied Mathematics*, 384:113195, 2021.

[32] J. Meena, M. Kumar, and M. Vardhan. Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. *IEEE Access*, 4:5065–5082, 2016.

[33] A. Mishra and D. Shrivastava. A tlbo and a jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs. *Computers & Industrial Engineering*, 124:509–522, 2018.

[34] R. M'Hallah. Minimizing total earliness and tardiness on a permutation flow shop using vns and mip. *Computers & Industrial Engineering*, 75:142–156, 2014.

[35] M. Nawaz, E. E. Enscore Jr, and I. Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95, 1983.

[36] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác. Reinforcement learning for solving the vehicle routing problem. *Advances in Neural Information Processing Systems*, 31, 2018.

[37] M. O. Odetayo. Optimal population size for genetic algorithms: an investigation. In *IEEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pages 2–1. IET, 1993.

[38] D. Ouelhadj and S. Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417–431, 2009.

[39] J. A. Palombarini and E. C. Martínez. Closed-loop rescheduling using deep reinforcement learning. *IFAC-PapersOnLine*, 52(1):231–236, 2019.

[40] K. Peng, J. Du, F. Lu, Q. Sun, Y. Dong, P. Zhou, and M. Hu. A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. *IEEE Access*, 7:49191–49200, 2019.

[41] M. L. Pinedo. *Scheduling*, volume 29. Springer, 2012.

[42] H. Piroozfard, K. Y. Wong, and A. Hassan. A hybrid genetic algorithm with a knowledge-based operator for solving the job shop scheduling problems. *Journal of Optimization*, 2016, 2016.

[43] I. Sabuncuoglu and S. Goren. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing*, 22(2):138–157, 2009.

[44] W. Shao, D. Pi, and Z. Shao. Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms. *Knowledge-Based Systems*, 137:163–181, 2017.

[45] Ł. Sobaszek, A. Gola, and E. Kozłowski. Predictive scheduling with markov chains and arima models. *Applied Sciences*, 10(17):6121, 2020.

[46] M. Squires, X. Tao, S. Elangovan, R. Gururajan, X. Zhou, and U. R. Acharya. A novel genetic algorithm based system for the scheduling of medical treatments. *Expert Systems with Applications*, 195:116464, 2022.

[47] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, 1994.

[48] E. Stafford, F. T. Tseng, and J. N. Gupta. Comparative evaluation of MILP flowshop models. *Journal of the Operational Research Society*, 56(1):88–101, 2005.

[49] S. Suliman. A two-phase heuristic approach to the permutation flow-shop scheduling problem. *International Journal of Production Economics*, 64(1-3):143–152, 2000.

[50] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[51] E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285, 1993.

[52] S. L. Takeda-Berger and E. M. Frazzon. An inventory data-driven model for predictive-reactive production scheduling. *International Journal of Production Research*, pages 1–25, 2023.

[53] F. T. Tseng and E. F. Stafford Jr. New MILP models for the permutation flowshop problem. *Journal of the Operational Research Society*, 59(10):1373–1386, 2008.

[54] L.-Y. Tseng and Y.-T. Lin. A hybrid genetic algorithm for no-wait flowshop scheduling problem. *International Journal of Production Economics*, 128(1):144–152, 2010.

[55] I. R. Uhlmann and E. M. Frazzon. Production rescheduling review: Opportunities for industrial integration and practical applications. *Journal of Manufacturing Systems*, 49:186–193, 2018.

[56] M. S. Umam, M. Mustafid, and S. Suryono. A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *Journal*

of King Saud University-Computer and Information Sciences, 34(9):7459–7467, 2022.

[57] J. P. Usuga Cadavid, S. Lamouri, B. Grabot, R. Pellerin, and A. Fortin. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31:1531–1558, 2020.

[58] G. E. Vieira, J. W. Herrmann, and E. Lin. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1):39–62, 2003.

[59] Y. Wang and Q. Zhu. A hybrid genetic algorithm for flexible job shop scheduling problem with sequence-dependent setup times and job lag times. *IEEE Access*, 9:104864–104873, 2021.

[60] H. Wei, S. Li, H. Jiang, J. Hu, and J. Hu. Hybrid genetic simulated annealing algorithm for improved flow shop scheduling with makespan criterion. *Applied Sciences*, 8(12):2621, 2018.

[61] C.-X. Wu, M.-H. Liao, M. Karatas, S.-Y. Chen, and Y.-J. Zheng. Real-time neural network scheduling of emergency medical mask production during covid-19. *Applied Soft Computing*, 97:106790, 2020.

[62] J. Xu and X. Zhou. A class of multi-objective expected value decision-making model with birandom coefficients and its application to flow shop scheduling problem. *Information Sciences*, 179(17):2997–3017, 2009.

[63] C. Yu, Q. Semeraro, and A. Matta. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research*, 100:211–229, 2018.

[64] J. Zhang, W. Qin, L. Wu, and W. Zhai. Fuzzy neural network-based rescheduling decision mechanism for semiconductor manufacturing. *Computers in Industry*, 65(8):1115–1125, 2014.

[65] F. Zhao, H. Liu, Y. Zhang, W. Ma, and C. Zhang. A discrete water wave optimization algorithm for no-wait flow shop scheduling problem. *Expert Systems with Applications*, 91:347–363, 2018.

[66] Y. Zheng and J. Xue. A problem reduction based approach to discrete optimization algorithm design. *Computing*, 88(1):31–54, 2010.

Publications

International journal

[1] P. Saophan, W. Pannakkong, R. Singhaphandu, and V.-N. Huynh. Rapid production rescheduling for flow shop under machine failure disturbance using hybrid perturbation population genetic algorithm-artificial neural networks (PPGA-ANNs). *IEEE Access*, 11:75794–75817, 2023.

International conference

[1] P. Saophan and W. Pannakkong. A genetic algorithm based artificial neural network for production rescheduling problem. In *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, pages 279–290. Springer, 2022.