UNIVERSITY OF MAKATI

DEVELOPMENT OF OPTICREW: A WORKFORCE MANAGEMENT SYSTEM FOR FIN-NOYS INTEGRATING RULE-BASED WITH GENETIC ALGORITHM OPTIMIZING TASK SCHEDULING

A THESIS SUBMITTED TO THE FACULTY OF THE COLLEGE OF COMPUTING

AND INFORMATION SCIENCES IN CANDIDACY FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

(APPLICATION DEVELOPMENT TRACK)

DEPARTMENT OF COMPUTER SCIENCE

BY

ADAM JAY M. BALONA

EMMAUS L. DIGOL

SUSHMITA T. ESTEBAN

LEIRAMARIE C. SAN BUENAVENTURA

August 2025

CHAPTER II

TOPICAL OUTLINE

1. Workforce Management
    a. Definition
    b. Workforce Planning
    c. Functions
        i. Task Management (Allocation)
        ii. Attendance (Tracking and Monitoring)
        iii. Scheduling
2. Workforce Management System
    a. Definition
        i. Workforce Management
        ii. Workforce Management System
        iii. Functions
            1. Task Management (Allocation)
            2. Attendance (Tracking and Monitoring)
            3. Scheduling
3. Cleaning Service Company
    a. Definition
    b. Characteristics
    c. Labor Laws in Finland

4. Algorithms
    a. Rule-Based Algorithm
        i. Definition
        ii. Types
        iii. Process
        iv. Pseudocode
        v. Mathematical Equation
        vi. Algorithm Performance Metrics
        vii. Applications
    b. Genetic Algorithm
        i. Definition
        ii. Types
        iii. Process
        iv. Pseudocode
        v. Mathematical Equation
        vi. Algorithm Performance Metrics
        vii. Applications
5. Development Tools
    a. Hardware Tools
    b. Software Tools
        i. Mobile application
        ii. Website

6. Testing and Evaluation

    a. System Integration Testing

    b. ISO 25010 Software Evaluation Criteria

    c. User Acceptance Testing

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter presents a comprehensive review of concepts, studies, and frameworks relevant to workforce management and its application in cleaning service companies. It covers key areas such as workforce management processes, workforce management systems, cleaning company operations, algorithms for task allocation and scheduling, development tools, and testing and evaluation standards. The discussion provides the necessary foundation to support the study's objectives and establishes a conceptual framework that connects these themes to the goals of the research.

Workforce Management

Workforce Management can be described as the practice or process of managing employees effectively. According to Porto et al. (2025), workforce management is a hierarchical process consisting of three interconnected decision-making levels across different time horizons: strategic planning, tactical scheduling, and operational assignment. Decisions at higher levels set the boundaries and influence those made at lower levels. The primary goal of workforce management is to enhance employee consistency, efficiency, performance, and productivity in the workplace (Garani et al., 2023). To achieve these goals, workforce management relies on structured processes such as workforce planning, which ensures that the right people with the right skills are available when needed.

Alabi et al. (2024) explain that workforce management includes the process of workforce planning, which focuses on analyzing current workforce demographics, identifying skill gaps, predicting future talent needs, and aligning human resources with organizational objectives. Workforce planning is crucial as organizations must adapt to economic changes, demographic shifts, and the demand for agility in the market. To prepare effectively, organizations need to understand the profile of their workforce, including factors such as age, gender, education, and experience. The study also emphasizes the value of grouping employees based on skills, performance, and career goals to ensure better planning and resource allocation. Additionally, according to Kellermayr-Scheucher et al. (2023), workforce planning includes prediction of manpower demand, future supply of manpower, and discrepancy between supply and demand via workforce scheduling and staffing. These insights form the basis of essential workforce management functions—task allocation, attendance tracking, and scheduling—that ensure employees are effectively assigned, monitored, and organized.

Task management refers to the process of planning, organizing, monitoring, and completing tasks efficiently within a set timeframe. A key aspect of this function is the proper allocation of tasks to employees. Effective workforce planning and allocation are vital in aligning organizational needs with human resources to enhance efficiency, productivity, and long-term success. However, Yanamala (2024) notes that many organizations still depend on static planning, making it challenging to address the growing complexity of workforce dynamics and the changing expectations of employees.

Attendance tracking and monitoring refers to the systematic process of recording employees' work hours, including their arrival, departure, and break times. This function enables organizations to accurately determine employee presence, absences, tardiness, and leave, thereby ensuring effective scheduling and workload management. Saule et al. (2023) emphasize the need for accurate employee tracking, noting that many organizations still rely on manual systems such as logbooks, which are error-prone, time-consuming, and vulnerable to proxy attendance, phishing, and information theft. They encourage companies to adopt more reliable attendance monitoring methods that are permanent, fast, and difficult to forge.

Scheduling focuses on assigning shifts, work hours, or specific time slots for tasks and employees. Kellermayr-Scheucher et al. (2023) noted that in the companies they studied, scheduling is considered the final step of workforce planning, where employees are allocated to particular shifts and workstations. Factors such as overtime, short-term contracts, employee qualifications, and holidays make both weekly scheduling and daily rostering a highly complex process. Effective scheduling ensures that the right number of qualified employees are available at the right time, preventing understaffing or overstaffing. Moreover, well-structured schedules contribute to smoother operations, higher employee satisfaction, and improved organizational productivity.

Workforce Management System

Workforce management is becoming increasingly complex, requiring organizations to meet demanding delivery requirements and adapt quickly to dynamic business environments (Kellermayr-Scheucher et al., 2023). Shabir (2023) highlights that traditional methods of managing employees often fail to address these challenges effectively. Without the use of modern technology and software, organizations face significant disadvantages. Porto et al. (2025) further emphasize that ineffective personnel scheduling leads to inefficient use of human resources, lower service quality, and reduced revenue. In addition, relying on manual processes such as static planning and attendance tracking often results in errors, delays, and security risks, making it even harder for companies to manage their workforce effectively. To address these challenges, many organizations are now turning to workforce management systems, which integrate technology and automation to streamline planning, scheduling, and employee monitoring.

Workforce Management System (WMS) is the tool or software used in order to make workforce management easier, faster, and more accurate. According to Khang (2024), a workforce management system enables organizations to enhance operational efficiency, improve decision-making, and optimize workforce utilization. However, its design and implementation come with challenges, as ethical and legal considerations—such as data privacy, security, and fairness—must be carefully addressed to safeguard employee rights and build trust. The key requirements of an ideal workforce management system include the representation of employee qualification profiles, measurement of individual productivity, forecasting capabilities, simulation and optimization tools to determine the best possible workforce solutions, automated planning, and effective bottleneck reduction and management (Kellermayr-Scheucher et al., 2023). Among the core functions commonly supported by a workforce management system are task management, attendance, and scheduling, which together ensure that organizations can align strategic goals with day-to-day operations.

Task management refers to the comprehensive process of planning, assigning, monitoring, and completing tasks to achieve a specific goal. A common challenge in this process is the planning fallacy—a cognitive bias where people tend to underestimate the time needed to complete tasks, often resulting in delays and added stress. Within task management, task allocation plays a vital role as it ensures tasks are assigned to the most suitable employees. As noted by Derkinderen et al. (2023), achieving efficient task allocation is complex due to factors such as uncertain task durations, operational constraints, and the need to consider employee well-being to maintain satisfaction. For this reason, task allocation is a key function of workforce management systems, as it enables fair and efficient task distribution while aligning organizational demands with employee skills and welfare.

Ussher-Eke et al. (2025) emphasize that attendance management is a vital component of human resource (HR) administration, acting as the foundation for workforce monitoring and overall organizational productivity. Traditionally, attendance was tracked using manual methods such as paper logs or punch cards, which were prone to errors, manipulation, and payroll inefficiencies. Implementing a workforce management system with an integrated attendance feature helps prevent issues like “buddy punching” and fraudulent clock-ins, allowing HR managers to maintain accurate and transparent records that support performance reviews and compliance with labor laws. Moreover, the adoption of digital attendance solutions enhances resource planning and operational efficiency. With the use of embedded sensors, mobile GPS tracking, and cloud-enabled dashboards, organizations can now verify employee locations, monitor work hours, and detect irregularities with greater accuracy. This underscores the critical role of attendance tracking and monitoring systems in modern organizations.

Workforce management systems enable organizations to automate their scheduling processes. Alabi et al. (2024) highlight that these tools support dynamic workforce adjustments by automatically modifying schedules based on real-time data. Through this approach, organizations can ensure that the right number of employees with the appropriate skills are available when required. This not only enhances operational efficiency but also boosts employee satisfaction by offering more flexible and responsive scheduling options. In addition, automated scheduling minimizes the risks of understaffing or overstaffing, reducing labor costs while maintaining high service quality. Overall, workforce management systems integrate these functions to create a streamlined, data-driven approach that improves both organizational performance and employee experience.

Cleaning Company

Cleaning service company is a service-oriented business that supplies trained workers and equipment to keep client spaces clean, safe, and well-maintained. Kosta (2024) notes that while cleaning tasks may appear simple, they demand advanced management and continuous quality monitoring to meet rising customer expectations and the fast-changing market environment. These services are commonly delivered in client-occupied spaces such as homes, hotels, cabins, offices, schools, and medical facilities, requiring cleaning methods and products tailored to each location's specific needs. The cleaning service industry is also experiencing rapid growth—for instance, in Poland, the market is expanding by approximately 20% annually, reflecting its dynamic development. Typical services include general cleaning, Airbnb or rental property cleaning, move-in and move-out cleaning, as well as specialized tasks like carpet cleaning, window washing, and deep-cleaning services. To better understand how these companies operate and sustain growth in a competitive market, it is important to examine their key characteristics, which define their structure, workforce practices, and service delivery.

Cleaning service companies are service-oriented, labor-intensive businesses that provide trained workers, equipment, and standardized procedures to maintain cleanliness and safety, while also ensuring client or customer satisfaction through outsourcing, flexible packages, and compliance with regulations. According to Olayiwola et al. (2024), a study on cleaning companies in Finland revealed that effective employee and communication management significantly improved customer satisfaction. They emphasized the importance of customer feedback mechanisms and regular satisfaction surveys using modern technology tools. The study also highlighted that cleaning companies must adopt a customer-oriented approach, recognizing customer satisfaction as their top priority for sustaining profitability. Lastly, organizational goals should be clearly communicated to employees through regular discussions and meetings to align their efforts with the company's objectives. Since these companies rely heavily on their workforce to deliver quality services, understanding and complying with labor laws in Finland is essential to protect employees' rights and ensure fair working conditions.

In Finland, cleaning service companies are often subject to a collective agreement for the property services sector, which establishes minimum standards for wages, working hours, and other employment conditions. The legal framework for these agreements is provided by the Collective Agreements Act (436/1946). Under the Working Time Act (872/2019), standard working hours are limited to eight hours per day and forty hours per week, with overtime pay premiums required in many cases. The Employment Contracts Act (55/2001) further governs the relationships between employers and employees. While there are concerns in some cases about abuses in contracts for foreign workers, formal legal investigations into these issues tend to be more situational rather than covered by broad statutory changes. These regulations highlight the need for cleaning service companies to maintain fair labor practices and transparent workforce management systems, ensuring both legal compliance and the protection of employee rights.

Algorithms

Since cleaning service companies rely heavily on efficient task allocation, fair scheduling, and workforce optimization, solving these challenges requires more than intuition or heuristics; algorithms provide a more systematic path. As Ali Arya (2023) explains, an algorithm is a sequence of clear and logically connected steps, where each action is precise, practical, and leads smoothly to the next. This structure allows complex problems to be broken down into smaller, more manageable parts, making algorithms essential not only for human problem-solving but also for building computer-based systems. In another source, Subarata Saha (2023) defines an algorithm as a finite sequence of well-defined instructions designed to solve specific problems or perform a computation. Each step is precise and clear, ensuring that the process leads to the correct results when executed. An algorithm's structured nature allows it to break down complex problems into small and manageable steps, making it vital not only for programming but also for problem-solving.

Metaheuristic optimization algorithms are high level search strategies designed to solve complex optimization problems that are computationally not feasible for exact methods. According to (Tomar et al., 2024), Metaheuristic algorithms are optimization techniques that are designed to find an adequate solution for a broad range of optimization problems, they are derivative-free, meaning they do not require the calculation of derivatives as in traditional optimization techniques, which enhances their flexibility and helps avoid being trapped in a limited region of search space. On the other hand, stated by (Brahim Benaissa, et al., 2024) Metaheuristic optimization algorithms (MOAs) can be categorized as single-solution–based methods, including examples like Genetic Algorithms, Ant Colony Optimization, and Particle Swarm Optimization. These algorithms are suitable for solving a wide range of problems—continuous, discrete, combinatorial, and mixed-integer—whether the task involves a single objective or multiple objectives. This versatility makes them applicable across numerous domains and real-world applications.

Genetic Algorithm

Diving into the topic of Genetic algorithms, a proceeding published by Spring Nature Singapore by (Jansen, et al., 2024), according to them a genetic algorithm is an optimization technique that mimics natural selection and genetic processes. It simulates biological evolution by retaining high-quality solutions and creating new ones through mechanisms like crossover and mutation to reach optimal outcomes. Another study by (Taha, Z. Y., et al., 2025) Genetic Algorithm is an optimization technique inspired by Darwin's theory of evolution, where every solution is represented by a chromosome and each parameter as a gene. The algorithm evaluates solutions using a fitness function that selects the best candidates for reproduction and generates new solutions through crossover and mutation.

Genetic Algorithms (GAs) can be classified into different types based on their encoding methods, selection strategies, and evolutionary mechanisms, each offering unique advantages for solving optimization problems. A paper by (Tarique, T. A., et al., 2025) conducts a comparative analysis to evaluate Simple Genetic Algorithm (SGA) and the Microbial Genetic Algorithm (MGA) on real-world classification tasks. SGA operates by evolving a population of candidate solutions using selection, crossover, and mutation, while MGA mimics microbial evolution by using a smaller population, tournament-based recommendation, and targeted mutation to efficiently explore the search space. The study found that MGA consistently outperformed SGA across all test cases, suggesting that MGA’s streamlined approach enhances generalization capabilities making a promising alternative to classification problems.

Additionally, Hybrid Genetic Algorithm (HGA) is another type of GA that integrates traditional GA with complementary optimization methods to enhance performance. For instance (Abdulghani, 2024) used a hybrid technique, formulated three hybrid algorithms—EDA-GA, HGA-ACO, and IDPSO—and found that they outperformed standalone algorithms, particularly in reducing makespan across experiments with 50, 100, and 150 tasks. These findings highlight the effectiveness of hybrid approaches in addressing the NP-complete and stochastic nature of cloud computing task scheduling, optimizing resource allocation, balancing loads, and improving overall system efficiency. Among these, the IDPSO hybrid approach showed the greatest potential for managing high task volumes.

Expanding the knowledge base of genetic algorithms (Taha, Z. Y., et al., 2025) describe the process of genetic algorithm as an iterative method inspired by natural evolution, particularly suited for feature selection problems. The algorithm operates by applying **selection**, **crossover**, and **mutation** mechanisms to evolve a population of candidate solutions toward optimal outcomes. This approach allows GAs to effectively explore and exploit the search space, enhancing solution quality while preventing convergence to locally best solutions.

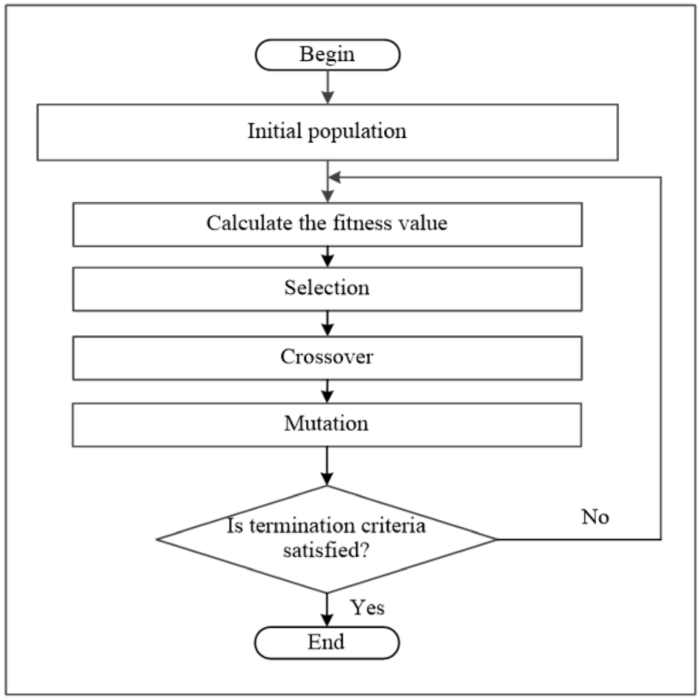

Figure 1. Evolutionary Algorithm Flowchart

With the use of figure 1 to visualize the Genetic algorithm process. The process begins with the **initial population**, which consists of potential solutions generated randomly, with each solution represented as a chromosome. In the **fitness evaluation** step, each individual is assessed using a fitness function that measures how effectively it solves the given problem. During **selection**, individuals are chosen based on their fitness scores to act as parents for the next generation, using methods such as roulette wheel selection, tournament selection, or rank-based selection. In the **crossover** step, selected parents exchange segments of their chromosomes to produce offspring, introducing diversity and allowing the algorithm to explore new regions of the solution space. During **mutation**, random changes are applied to the offspring's chromosomes to maintain genetic diversity and prevent premature convergence. Finally, in the **replacement** stage, the new generation of offspring replaces the previous population, and the cycle repeats from the fitness evaluation step until a termination condition is met.

```
Set parameters;
iterations g = 0;
Initialize population base on chaotic maps (P (g = 0));
Evaluate population (P (g = 0));
while g < Total iterations do
g = g + 1;
Select P(g) from P (g – 1) ;
Reproduce P (g) ;
Mutate P (g) ;
Evaluate P (g) ;
end while
return the best solution
```

Figure 2. Sample of Genetic Algorithm Pseudocode

Figure 2 above shows the pseudocode for the genetic algorithm, adapted from (Wentian Shang, & Jinzhang Jia, 2024). The algorithm begins by setting the parameters and initializing a population using chaotic maps. Each individual in the population is evaluated for fitness. The algorithm then iteratively performs selection, reproduction, and mutation to generate new populations until the maximum number of iterations is reached. Finally, the best solution is returned.

While in the study conducted by (Mohammed et al., 2024) they used a genetic algorithm (GA) to optimize task scheduling in cloud-based AI systems. The research provides a mathematical formulation on how new solutions meaning offspring are generated in a genetic algorithm. The solution of offspring is expressed as:

$$OS = P + Pl_m \times \delta$$

Where *OS* denotes the offspring solution, *P* is the parent solution, *Plm* is the perturbation limit, and δ is a scaling factor. This equation indicates that a new solution is created by taking an existing solution and applying a control adjustment. The scaling factor is then further defined as:

$$\delta = min\left(\frac{(OS - X_t), (X_u - OS)}{X_u - X_l}\right)$$

Where *Xt* is the target solution, *Xu* is the upper bound, and *Xl* is the lower bound of the solution space. This formulation ensures that the offspring remains within feasible limits, preventing solutions from exceeding the problem's upper or lower boundaries. Meaning the equations highlight that genetic algorithms generate new candidate schedules by tweaking existing ones, while maintaining validity within the defined problem space.

Algorithm performance metrics are often evaluated using parameters such as execution time, fitness quality, convergence rate, and scalability. Relating to the comparative review study conducted by (Langazane and Saha, 2024) of modern optimization techniques such as PSO, GA, and AI. The researchers have applied these metrics where they reported that GA generally required longer execution time compared to PSO, but produced higher-quality solution results when parameters such as crossover rate and mutation probability were properly tuned. In their sensitivity analysis they found out that using a 30% crossover rate, 2% mutation rate, and a smaller population size improved the convergence rate and fitness of the genetic algorithm substantially.

Adding to this is a recent study conducted by (Hassan, et al,. 2025), assessed scheduling algorithms in large-scale multiprocessor environments using performance indicators such as Dynamic Load Measure (DLM), Average Response Time (ART), and Average Turnaround Time (ATAT). Their analysis revealed that Earliest Deadline First (EDF) operates with a complexity of O(n log n), while Least Laxity First (LLF) is more computationally demanding at $O(n^2)$. Fuzzy logic–based scheduling (EFSBA) demonstrated a complexity of O(n + m), dependent on the number of inputs and rules. In contrast, the GA-based approach (OPBGA) exhibited a higher complexity of O(G × P × L), where G, P, and L correspond to the number of generations, population size, and chromosome length, respectively. Despite this increased computational burden, the GA method demonstrated superior scalability, performing effectively across scenarios involving 3 to 100 processors and workloads extending to 3000 tasks. These findings suggest that although GA requires greater computational effort, it provides enhanced flexibility and robustness for handling large and complex scheduling problems.

Genetic Algorithms (GAs) are flexible optimization tools that can handle complex and dynamic task allocation problems. (Yao et al.,2024) introduced an improved GA for task reallocation in a human-robot collaborative production environment, where the algorithm adapts based on a dynamic human fatigue model. Using multimodal data to assess the fatigue level of workers, the algorithm reallocates tasks in real time to maintain both efficiency and worker well-being. This approach highlights how GAs can be enhanced with reinforcement learning and real-world considerations, offering insights into their potential for human-centered task assignment systems.

An effective task allocation is crucial for ensuring an efficient resource use, well-organized, less stressed workforce, and improved service quality which leads to better customer satisfaction. (Yan, F. et al., 2024) the study addresses the task allocation problem in multi-UAV systems by applying a Genetic Algorithm to optimize assignments under simultaneous arrival and resource constraints. The GA allowed them to search a large solution space efficiently and generate optimal allocations that improve missions success rates and resource utilization. Their findings reinforce ideas that a well-designed task allocation especially supported by optimization algorithms like GA are essential for performance optimization and effectiveness.

Adding to the topic, a study by (Youssef Msala et al., 2023) applied a genetic algorithm (GA) to solve the travelling salesman problem (TSP), their research focused on distributing a set of tasks with varying difficulty levels among a heterogeneous multi-robot system operating within a defined space. By modeling the task allocation as an open-loop TSP they used Genetec algorithm (GA) to determine the shortest path possible for the robots to execute the assigned task efficiently. This approach not only optimizes the task distribution but also minimizes the travel distance, thereby improving overall efficiency.

Rule-Based Algorithm

To carry on the topic of algorithms, A Scaler's tutorial by (Eshika Shah, 2023) explains that a rule-based system is an AI method that makes decisions by following a set of clearly defined rules. These rules usually follow an "if–then" pattern, meaning that when certain conditions are met, the system knows what action to take to reach a specific outcome. This approach makes the process transparent because you can always see which rule led to which result. Similarly, (Gorchakov et al., 2024) describe a rule-based algorithm as a structure that uses organized sets of rules—expressed as conditions and actions—to systematically analyze parts of a program. Every rule that applies adds to the final outcome, showing how rule-based systems offer a logical and reliable way to solve problems. Together, these explanations show that rule-based algorithms work best when you can clearly define the decision-making steps and put them into a logical order.

In discussing the types of rule-based algorithms, researchers highlight different perspectives on how a rule-based system works. For instance, (Sarker, et al., 2024), in their taxonomy of rule-based AI, they discuss forward chaining as a data-driven reasoning method where the system begins with known facts and applies rules to infer new information until conclusion. This also contrasts with backward chaining, which starts from a goal and works backwards to determine if available information facts supports it. This perspective shows that the algorithms can differ not only in their structural form but also in how they apply rules to reach a decision.

Another type of rule-based algorithm is backward chaining, in a study conducted by (Al-Hakim et al., 2025) developed a rule-based AI system for early pediatric Type 1 diabetes diagnosis by combining backward chaining with certainty factors. The system begins with a diagnostic hypothesis and searches its rule base, structured as *IF <symptom/test result> THEN <diagnosis>*, to verify whether patient data satisfy the conditions. The system model achieved 79.2% accuracy, indicating its potential as a decision-support tool. However, the authors noted limitations such as the small validation sample and reliance on fixed rules, recommending future integration of rule-based reasoning with machine learning to improve adaptability.

A different study by (Turgunbaev, 2025) examined rule-based reasoning as a deterministic approach that ensures consistency and explainability by deriving conclusions from clearly defined conditions. The study outlined the use of forward chaining for diagnostic tasks and backward chaining for hypothesis testing, with applications across healthcare, finance, law, education, and customer support. Despite challenges with uncertainty and scalability, the author recommended hybrid models that integrate rule-based reasoning with machine learning for greater adaptability.

Diving into the process of rule-based algorithms, as reference to the studies above, rule-based algorithms are widely recognized for their systematic application of predefined rules to transform input data into actionable decisions. The process of these algorithms generally involves input acquisition, rule matching, rule selection, execution, and output generation. In the context of task allocation or task management, this process enables the efficient assignment of tasks to resources based on established criteria.

In input acquisition, the system collects relevant information about the task and resources. Task attributes often include priority, deadline, and required skills. While the resource usually contains profiles availability, expertise and workload. During the input acquisition stage it ensures that all the relevant factors are considered before any assignment decisions are made (Li et al., 2025). The next process is the rule matching, this stage involves evaluating each task and resource against the systems rule base. Where the rules are generally structured in the form of "IF <task_condition> AND <resource_condition> THEN <assignment_action>" .that a high-priority task requiring a certain skill it should be assigned to an available employee with matching expertise. The system systematically applies these rules to determine potential matches (Adebowale, 2025). Finally on the output generation stage process, this stage updates task allocation database and notify relevant personnels. This process ensures transparency, traceability and efficiency in task distribution. The type forward chaining is commonly used in such applications where the algorithm usually starts with known facts and iteratively applies rules until all task as assigned (Tatipamula, 2025)

RUMC: A Rule-based Classifier Inspired by Evolutionary Methods

Algorithm 1 RUMC algorithm

Input: Training dataset D_{train}
Output: The set of decision rules R_{set}
procedure RUMC(D_{train})
    Preprocessing
    Create *Initial_Rules*
    Extant_Rules = *Initial_Rules*
    Rule Mutation
    Primary Generalization of *Extant_Rules*
    for $i = 1$ to |*Extant_Rules*| **do**
        for $j = i + 1$ to |*Extant_Rules*| **do**
            Composed_Rule = Rule Composition(R_i, R_{i+1})
            if *Composed_Rule* is better than R_i and R_{i+1} **then**
                Replace both R_i and R_{i+1} with *Composed_Rule*
                Omit all covered rules with *Composed_Rule*
                Update *Extant_Rules*
            end if
        end for
    end for
    Secondary Generalization of *Extant_Rules*
    Sort the *Extant_Rules* Based on Fitness
    R_{set} = *Extant_Rules*
end procedure

Figure 3. Sample of Rule-base Classifier Inspired by Evolutionary Methods

Figure 3 presents the pseudocode for a rule-based classifier inspired by evolutionary methods, as adapted from Melvin Mokhtari (2024). The algorithm begins with preprocessing and the generation of initial rules. These initial rules undergo a mutation process based on evolutionary techniques to promote diversity in the candidate rule set. Next, the algorithm performs primary generalization and evaluates the fitness of each rule through composition and replacement, retaining only the most effective rules. Finally, after a secondary generalization step, the decision rules are sorted by fitness to yield the final set.

A study by (Pagliarin, et al., 2024), emphasizes that rule-based systems remain a cornerstone of interpretable machine learning due to their transparency and logical structure. Unlike opaque black-box models, symbolic classification represents classifiers through mathematical notation allowing rules to be both intelligible and formally defined. Specifically the model can be expressed as:

$$\Gamma = \{\varphi_1 \Rightarrow L_1, \ldots, \varphi_z \Rightarrow L_z\},$$

where Γ denotes the entire set of classification rules, φi represents the logical condition example *(if age > 18 AND income > 30k),* and LiLi the corresponding class. This formulation ensures that rules are mutually exclusive and collectively exhaustive, providing a structured foundation for rule extraction and making decision-making processes transparent and actionable.

Evaluating the performance of algorithms, especially those involving rule-based classification, requires well-defined metrics that balance both predictive accuracy and rule quality. The study of (Mokhtari, M., & Basiri, A. 2024), on rule optimized aggregation classifier(ROPAC), introduces a fitness-base evaluation function to measure the effectiveness of generated rules. Expressed as:

$$Fitness(R_i) = \alpha \times Accuracy(R_i) + \beta \times Coverage(R_i)$$

This formulation considers two important aspects: **accuracy** and **coverage**. Accuracy is computed as the proportion of correctly classified instances over the total number of instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP represents true positives, TN true negatives, FP false positives, and FN false negatives. Accuracy ensures that the rule provides correct predictions, while coverage ensures that the rule applies to a sufficiently large subset of the dataset. The weights α and β allow fine-tuning between these two metrics, balancing precision with general applicability.

This approach to performance evaluation is especially important in classification tasks, as high accuracy alone may not ensure real-world usefulness if a rule applies to only a small segment of the dataset. By incorporating both accuracy and coverage, the ROPAC model highlights the need for rules to be both correct and broadly applicable within classification frameworks. Moreover, combining statistical measures like accuracy with rule-based traits such as coverage enables a more comprehensive assessment of an algorithm's performance.

Rule-based algorithms are known for their interpretability and efficiency, especially in time-sensitive applications such as task scheduling. (Barut et al., 2024) proposed a hybrid approach that combines metaheuristic scheduling with machine learning-generated rule sets to enhance performance in dynamic cloud environments. Their method archives previous metaheuristic solutions and extracts interpretable rules through machine learning to quickly identify optimal solutions for similar future problems. This rule-based framework allows for faster execution without compromising solution quality, making it suitable for real-time or resource-constrained environments.

Development Tools

In developing systems, hardware resources play a crucial role in supporting computation, simulation and system development. Researchers commonly rely on the standard computing device such as laptops, desktops, servers, and smartphones. Which provide the processing power and storage needed to execute complex systems. High-performance computing setups are particularly essential for handling large-scale simulations and optimization tasks, as demonstrated in several recent studies. Similarly to a study conducted by (Wang, et at., 2025) experiment on a system equipped with windows 11, 12th gen intel core i7-12700H CPU, and 16 GB of RAM to evaluate the computational efficiency of the Queueing Search Algorithm(QSA). Based on their findings QSA required 1.2 to 6 times more computational time than other algorithms due to its extensive search operation. Although computational demand is higher, the researchers emphasized that in edge computing, the tradeoff is worthwhile since greater processing time can yield better deployment solutions and service quality, underscoring the need for adequate hardware to support advanced heuristic methods.

Another study by (Hussain, et al. 2024) proposed the GA-FiFeS algorithm which combines genetic algorithms with first feasible speed (FiFeS) scheduling for high-performance edge computing systems. Their simulation results indicate ~18.56% energy savings and ~2.78% improved response times compared to FiFeS alone. This study underscores that having the right hardware model like multi-core processors, adjustable speeds, etc… is essential: without hardware which supports multiple speed levels, sufficient cores, and reliable power models, such energy and timing trade-offs can’t be measured or optimized properly.

In addition to this, a study by (Holovati & Mohd Zaki, 2024) developed a smart scheduler that decides which hardware accelerator GPU, TPU, or FPGA should handle a given task in an edge system. By using lightweight machine learning to match tasks with the most suitable processor, their system was able to save up to 35% energy and reduce task delays by 28% compared to traditional scheduling methods. Tested on real devices such as an NVIDIA Jetson GPU, a Google Coral TPU, and an Intel FPGA, the framework showed it can adapt to heavy workloads with very little overhead, making it practical for IoT, embedded AI, and other energy-sensitive applications.

Software tools complement hardware by providing necessary platforms, programming environments, and frameworks that allow algorithms and system designs to be implemented, tested, and optimized. While hardware decides the system's physical capabilities, software makes the resources available and usable. A Recent literature review emphasizes that effective software development practices such as Scrum and DevOps are inseparable from the use of specialized software tools. (Pastrana et al. 2025) highlight that version control systems example like GitHub, GitLab, CI/CD pipelines the likes of Jenkins, Azure DevOps, and automated testing frameworks with example of Selenium, and JUnit are essential in enabling collaboration, traceability, and rapid delivery in agile environments. Their review shows that these tools not only enhance productivity and software quality but also support core Scrum values such as transparency and continuous feedback. However, the study also notes that very small entities (VSEs) often face challenges in fully adopting these tools due to limited resources and infrastructure, underscoring the importance of selecting lightweight yet efficient toolchains that align with organizational needs.

Furthermore another literature review by (Aouni, et al., 2025) concluded that the convergence of Agile, DevOps, and cloud technologies represents a significant shift toward collaborative, scalable, and automated development practices. This integration is essential for accelerating delivery, improving software quality, and maintaining competitiveness in dynamic environments. As organizations increasingly rely on this unified approach to drive innovation and growth, future research should address the challenges of combining these methods and explore their interaction with emerging technologies to optimize efficiency and effectiveness.

Developing and maintaining mobile applications requires not only usability and accessibility considerations but also efficient software delivery practices. In this context, (Ozdenizci Kose, 2024) looked into how DevOps methods are being used in mobile app development. The study pointed out that things like different device types, app store rules, and strong security needs create challenges that aren't as common with web or desktop software. To handle these issues, companies are updating their DevOps approaches by adding better testing for different devices, setting up automated systems to sign and release apps, and making security a constant focus. These insights add to the bigger conversation about mobile app development, showing that making sure apps run smoothly and get updated quickly is just as important as good design and usability. By weaving DevOps into the mobile app world, companies make their apps more reliable, accessible, and useful for everyday life.

A recent comparative study by (Jošt & Taneski, 2025) where the researchers explores the most widely used cross-platform frameworks for mobile applications, focusing on Flutter, React Native, and the newer .NET MAUI, and examines how organizations are adopting them. The findings show that although Flutter and React Native are both highly popular and attract considerable developer interest, Flutter is distinguished by higher long-term developer satisfaction, stronger community activity reflected in GitHub engagement and frequent updates, and increasing visibility in both market demand and search trends. The research emphasizes that selecting a framework involves more than ease of use or the availability of features such as offline support or push notifications. Developers also need to consider performance, available tools, the frequency of updates, and job market relevance.

Although mobile applications provide portability and convenience, websites remain indispensable for ensuring broader accessibility, seamless integration, and platform independence. A study by (Emmanni, 2023) conducted a comprehensive comparative analysis for Angular, React and Veu.js, three of the most popular use of front-end frameworks, within the context of single page application (SPA) development. The study evaluated each framework based on performance benchmarks, developer productivity, ecosystem support and learning curves. The findings aimed to provide guidance for developers and organizations in selecting the framework that best aligns with the specific requirements and constraints of their web development projects. This research highlights the evolving landscape of front-end development and underscores the importance of choosing a framework that balances efficiency, scalability, and ease of use.

Testing and Evaluation

System Integration Testing is the process of verifying how combined modules or subsystems interact, ensuring proper communication, data flow, and functionality cooperation before full system deployment. According to the **ISO/IEC/IEEE 29119** family of software testing standards, the purpose of testing is to establish internationally agreed principles and processes for conducting reliable and repeatable evaluations of software quality. The most recent edition, ISO/IEC/IEEE 29119-1:2022, defines the fundamental concepts and terminology for software testing, while ISO/IEC/IEEE 29119-5:2024 expands the standards to include keyword-driven testing approaches, which are relevant to automation in SIT. (Jayasankar, 2024) study highlights that while unit testing ensures individual component correctness, integration testing is crucial because components-based systems often suffer from dependency issues and evolving interfaces. Furthermore the automation enhances SIT by enabling frequent, repeatable execution of integration tests, however its success depends on alignment with the frameworks and standards like ISO/IEC 29119.

In a related perspective, (Tramontana, et al., 2024) study examined how software testing education incorporates these standards across universities in four European countries. By using ISO/IEC/IEEE 29119 as a reference framework to their study. Where it revealed that while testing fundamentals and specification-based techniques are widely covered, critical practices such as exploratory and experience-based testing remain underrepresented. This limited coverage highlights the challenge of ensuring that future professionals are adequately prepared to apply standards like ISO 29119 not only in integration testing contexts but also across diverse testing practices. Further supporting the importance of a guide in structuring professional testing practice in evolving software systems.

The ISO/IEC 25010 model outlines nine key qualities that good software should have. However, in this study, only four are considered, as they are measurable within the system being developed. First, **functional suitability** is about how well the software meets what users need and expect it to do. **Compatibility** is all about whether the software can work and share information with other programs or devices. **Security** means protecting data and information from being accessed or changed by the wrong people. **Maintainability** measures how easily the software can be fixed, improved, or updated.

While the ISO 25010 provides a comprehensive framework for the technical qualities of a good software, a recent work by (Prameswari & Setyawam, 2025) where they have combined ISO 25010 to SERVQUAL in order to evaluate websites more fully. The hybrid approach reveals not only how well a system performs technically but also whether it meets users' expectations that are critical for real-world website application success.

In addition to this, the *Proceedings of the Widyatama International Conference on Engineering (2024)* discussed the evaluation of the Ixitask application's quality using the updated ISO/IEC 25010:2023 standard. The study emphasized key dimensions such as functional suitability, reliability, usability, and security in assessing software performance. Its findings highlight that ISO/IEC 25010 provides a structured framework for identifying gaps and ensuring alignment with business objectives. This further reinforces the standard's role as both a benchmark for global quality and a tool for continuous improvement (Widyatama International Conference on Engineering [WICE], 2024).

User Acceptance Test (UAT) is used to ensure the fitness of the system for use, which is conducted by the business users. UAT plays a critical role in validating whether the developed system meets business requirements and provides the expected value to end users before going live. A paper by (Wang, et al., 2024) introduced XUAT-Copilot, a multi-agent collaborative system powered by large language models (LLM). The study addressed the labor-intensive process of manual test script generation by employing three specialized LLM-based agents for action planning, state checking, and parameter selection, along with supporting modules for state sensing and case rewriting. Their experimental results showed that the multi-agent system achieved near-human effectiveness and significantly outperformed single-agent models in terms of accuracy.

More importantly, XUAT-Copilot was successfully deployed in WeChat Pay's formal testing environment, demonstrating its potential to reduce manpower requirements while ensuring reliability in large-scale mobile applications.

In addition, (Isa, et al,. 2023), explored how user acceptance testing (UAT) can help evaluate the Final Year Project Management System (FYPMS) at Universiti Poly-Tech Malaysia. They used a survey with students, lecturers, and coordinators to get feedback on key features like logging in, submitting projects, supervision, and grading. The results showed that users were generally very satisfied, especially with the login and submission features. However, some aspects—like the coordinator interface and the grading system for lecturers—still needed work. This study shows that UAT is essential for finding issues with usability and functionality, making sure the system works well, and guiding future improvements.

Synthesis

The reviewed literature highlights different perspectives on workforce management and task allocation. Studies emphasize that workforce scheduling and allocation are crucial for efficiency, especially in service-based industries like cleaning companies. Research also shows that poor scheduling can lower productivity, reduce service quality, and negatively affect revenue. At the same time, workforce management systems with algorithms such as rule-based and genetic approaches have been shown to automate scheduling, improve fairness, and enhance employee satisfaction. The inclusion of labor laws in Finland provides further context, showing how legal frameworks shape scheduling practices and protect worker rights. Together, these studies reveal that both management strategies and external regulations strongly influence how tasks are allocated and optimized.

Despite the insights provided, several research gaps remain. Many studies focus either on the technical side of algorithms or on organizational management, but few combine both areas to explore how algorithms can practically optimize workforce scheduling. There is also limited research that connects workforce management with real-world industries such as cleaning services under specific labor laws. Additionally, most works lack long-term evaluations of algorithm-based scheduling systems, making it unclear how effective they remain over time. These gaps show the importance of developing an integrated system, like the one proposed in this study, that combines algorithms with practical workforce management needs in a real industry setting.

Conceptual Model of the Study

This section presents the Conceptual Model for OptiCrew, following an Input-Process-Output framework to ultimately deliver the functionalities of the system while integrating the Rule-Based and Genetic Algorithm.

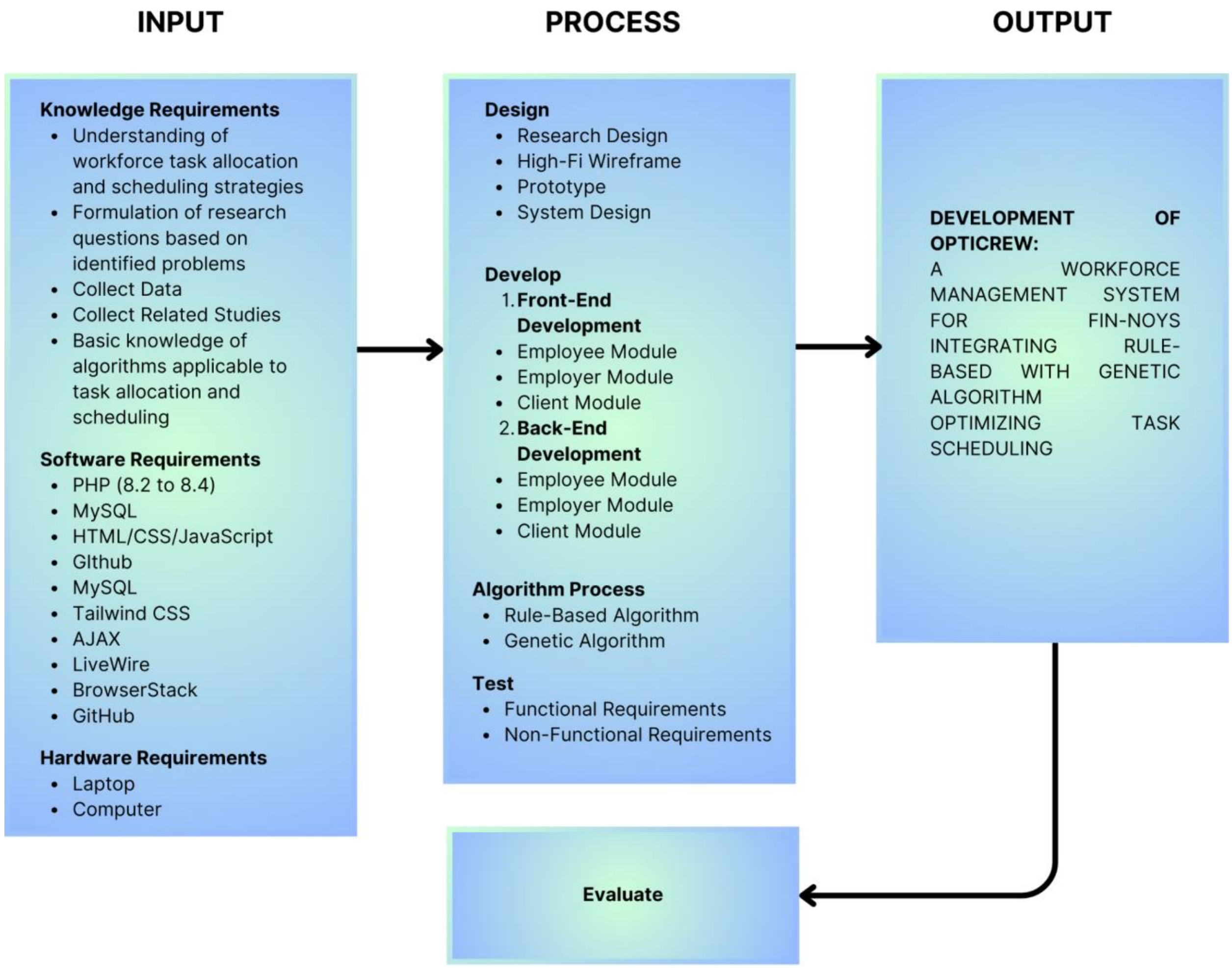

Figure 4. IPO Diagram

The Input-Process-Output (IPO) diagram is a valuable tool for illustrating the flow of information and activities in this study. It consists of three main components:

In this study, inputs consist of knowledge, software, and hardware requirements. The knowledge requirements include an understanding of workforce task allocation and scheduling strategies, as well as algorithms such as the Rule-Based Algorithm and Genetic Algorithm. Software requirements involve programming languages and tools like PHP, MySQL, HTML/CSS/JavaScript, Tailwind CSS, AJAX, and GitHub, while hardware requirements include development computers and client devices. These inputs provide the foundation for building the workforce management system.

The processes in this research involve several development stages. The design stage covers system architecture, database design, and interface planning. Development follows, where modules such as Employee, Employer, and Client Module are implemented. The algorithm process integrates both the Rule-Based Algorithm and Genetic Algorithm for optimized scheduling. Testing is conducted at unit, integration, and system levels to ensure both functionality and usability. These steps collectively contribute to achieving the system's goals.

The outputs of the study consist of both a system and research contributions. The primary output is the OptiCrew Workforce Management System, designed for Fin-Noys and capable of integrating rule-based and genetic algorithms for optimized task scheduling. Beyond the system, the study also produces insights into how hybrid algorithm approaches can improve efficiency in workforce scheduling. The evaluation phase is designed to assess the effectiveness, reliability, and overall performance of the developed system which includes an evaluation of the functionality and performance of features for each module, as well as the algorithm metrics.

References

Porto, A. F., Lusa, A., Herazo, S., & Henao, C. A. (2025). Improving the robustness of retail workforce management with a labor flexibility strategy and consideration of demand uncertainty. *Operations Research Perspectives*, 100345. https://doi.org/10.1016/j.orp.2025.100345

Garani, G., Tolis, D., & Savvas, I. K. (2023). A trajectory data warehouse solution for workforce management decision-making. *Data Science and Management*, *6*(2), 88–97. https://doi.org/10.1016/j.dsm.2023.03.002

Alabi, O. A., Ajayi, F. A., Udeh, C. A., & Efunniyi, C. P. (2024). Predictive Analytics in Human Resources: Enhancing workforce planning and customer experience. *International Journal of Research and Scientific Innovation*, *XI*(IX), 149–158. https://doi.org/10.51244/ijrsi.2024.1109016

Kellermayr-Scheucher, M., Niedermeier, M., & Brandtner, P. (2023). Applications and Perceptions of Workforce Management Systems for Warehouse Operation - Results and Findings from Expert Interviews. *Procedia Computer Science*, *219*, 255–262. https://doi.org/10.1016/j.procs.2023.01.288

Yanamala, K. K. R. (2024). Strategic Implications of AI Integration in Workforce Planning and Talent Forecasting. *Journal of Advanced Computing Systems*, *4*(1), 1–9. https://doi.org/10.69987/jacs.2024.40101

Suale, Y., Kamil, N. M., Musah, M. B., Zakari, M., & Farah, A. M. (2023). The Impact of Biometric Time and Attendance System on Workforce Management Outcomes: The Moderating Role of Managerial Commitment in the Service Sector in Northern Ghana. International Journal of Business and Technology Management, 5(3), 442-455. https://doi.org/10.55057/ijbtm.2023.5.3.37

Shabir, G. (2024). Transforming workforce management: Leveraging advanced analytics, AI, and integrated approaches for performance forecasting.

Derkinderen, V., Bekker, J., & Smet, P. (2023). Optimizing workforce allocation under uncertain activity duration. *Computers & Industrial Engineering*, *179*, 109228.

https://doi.org/10.1016/j.cie.2023.109228

Ussher-Eke, D., Okoh, O. F., & Ijiga, O. M. (2025). The role of biometric and IoT-Based attendance systems in streamlining HR administrative functions, enhancing workforce accountability, and reducing labor inefficiencies. *International Journal for Multidisciplinary Research*, *7*(4). https://doi.org/10.36948/ijfmr.2025.v07i04.52237

Kosta, O. (2024). The SERVQUAL Method as a Tool for Investigating the Quality of Cleaning Services. An International Serial Publication for Theory and Practice of Management Science, 113-130.

Olayiwola, R. K., Tuomi, V., Strid, J., & Nahan-Suomela, R. (2024). Impact of Total quality management on cleaning companies in Finland: A Focus on organisational performance and customer satisfaction. *Cleaner Logistics and Supply Chain*, *10*, 100139. https://doi.org/10.1016/j.clscn.2024.100139

Ali Arya. (n.d.). *Anyone Can Code: Algorithmic Thinking* (First Edition © 2023).

Google Books. (n.d.). https://www.google.com.ph/books/edition/Data_Structures_and_Algorithms_using_Pyt/wt_LEAAAQBAJ?hl=en&gbpv=1&dq=algorithms&pg=PA15&printsec=frontcover

Tomar, V., Bansal, M., & Singh, P. (2024). Metaheuristic algorithms for optimization: A brief review. *Engineering Proceedings*, *59*(1), 238.

Benaissa, B., Kobayashi, M., Al Ali, M., Khatir, T., & Elmeliani, M. E. A. E. (2024). Metaheuristic optimization algorithms: An overview. *HCMCOU Journal of Science–Advances in Computational Structures*, 33-61.

Google Books. (n.d.-b). https://www.google.com.ph/books/edition/Proceedings_of_the_3rd_International_Con/3

RoLEQAAQBAJ?hl=en&gbpv=1&dq=genetic+algorithm+optimization&pg=PA314&printsec=frontcover

Taha, Z. Y., Abdullah, A. A., & Rashid, T. A. (2025). Optimizing feature selection with genetic algorithms: a review of methods and applications. *Knowledge and Information Systems*, 1-40.

Tarique, T. A., Bhadra, D., Ahmed, S. U., Shahjahan, M., & Murase, K. A Comparative Study of Genetic Algorithm and Microbial Genetic Algorithm for Classification problems.

Abdulghani, A. (2024). Hybrid task scheduling algorithm for makespan optimisation in cloud computing: A performance evaluation. *Journal of Artificial Intelligence*, *6*, 241.

Taha, Z. Y., Abdullah, A. A., & Rashid, T. A. (2025). Optimizing feature selection with genetic algorithms: a review of methods and applications. *Knowledge and Information Systems*, 1-40.

Shang, W., & Jia, J. (2025). Theoretical knowledge enhanced genetic algorithm for mine ventilation system optimization considering main fan adjustment. *Complex & Intelligent Systems*, *11*(1), 16.

Mohammed, A. S., Mallikarjunaradhya, V., Sreeramulu, M. D., Boddapati, N., Jiwani, N., & Natarajan, Y. (2024, September). Optimizing Real-time Task Scheduling in Cloud-based AI Systems using Genetic Algorithms. In *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 7, pp. 1649-1653). IEEE.

Langazane, S. N., & Saha, A. K. (2024). A comparative review of current optimization algorithms for maximizing overcurrent relay selectivity and speed. *IEEE Access*, *12*, 53205-53223.

Hassan, H. E., Ibrahiem, K. H., & Madian, A. H. (2025). Optimizing multiprocessor performance in real-time systems using an innovative genetic algorithm approach. *Scientific Reports*, *15*(1), 3842.

Yan, F., Chu, J., Hu, J., & Zhu, X. (2024). Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm. *Expert Systems with Applications*, *245*, 123023.

Msala, Y., Hamed, O., Talea, M., & Aboulfatah, M. (2023). A new method for improving the fairness of multi-robot task allocation by balancing the distribution of tasks. *Journal of Robotics and Control, 4*(6), 743–753.

Shah, E. (2023, June 8). *Rule-based systems in AI - Scaler topics*. Scaler Topics. https://www.scaler.com/topics/artificial-intelligence-tutorial/rule-based-system-in-ai/

Gorchakov, A. V., Demidova, L. A., & Sovietov, P. N. (2024). A Rule-Based Algorithm and Its Specializations for Measuring the Complexity of Software in Educational Digital Environments. *Computers*, *13*(3), 75.

Sarker, I. H., Janicke, H., Ferrag, M. A., & Abuadbba, A. (2024). Multi-aspect rule-based AI: Methods, taxonomy, challenges and directions towards automation, intelligence and transparent cybersecurity modeling for critical infrastructures. *Internet of Things*, *25*, 101110.

Al-Hakim, R. R., Setiawan, R. A., Hidayat, R., Arkananta, E. R., Samodra, G., Jayusman, H., ... & Famuji, T. S. (2025). Rule-based ai system for early paediatric diabetes diagnosis using backward chaining and certainty factors. In *BIO Web of Conferences* (Vol. 152, p. 01020). EDP Sciences.

Turgunbaev, R. (2025, August 10). *Rule-based reasoning and its role in intelligent decision making*. https://altumnova.com/index.php/tsir/article/view/9

Li, J. (2025, March). Intelligent Task Allocation Model for University Teaching Based on Knowledge Graph and Hierarchical Soft Actor-Critic Algorithm. In *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence* (pp. 280-286).

Adebowale, J. (2025, August). *Intelligent task automation in project management using generative AI and conversational agents*. University of Ilorin. https://www.researchgate.net/publication/394535835_INTELLIGENT_TASK_AUTOMATION_IN_PROJECT_MANAGEMENT_USING_GENERATIVE_AI_AND_CONVERSATIONAL_AGENTS

Tatipamula, S. (2025). The Evolution of AI Workflow Automation: From Rules to Adaptive Intelligence. *IJSAT-International Journal on Science and Technology*, *16*(2).

Mokhtari, M. (2024). RUMC: A Rule-based Classifier Inspired by Evolutionary Methods. *arXiv preprint arXiv:2412.07885*.

Pagliarini, G., Ponsanesi, E., Sciavicco, G., & Stan, I. E. (2024). Towards Modern Rule-Based Learning.

Mokhtari, M., & Basiri, A. (2024). ROPAC: rule optimized aggregation classifier. *Expert Systems with Applications*, *250*, 123897.

Wang, B., Sun, X., & Song, Y. (2025). Edge Server Deployment Strategy Based on Queueing Search Meta-Heuristic Algorithm. *Algorithms*, *18*(4).

Energy efficient Real-Time tasks scheduling on High-Performance Edge-Computing systems using genetic algorithm. (2024). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10499799

Holovati, J. L., & Zaki, F. M. (2025, June 17). *Energy-Aware task scheduling in heterogeneous GPU/TPU–FPGA embedded platforms*. https://www.eccsubmit.com/index.php/congress/article/view/92

Pastrana, M., Ordoñez, H., Cobos-Lozada, C. A., & Muñoz, M. (2025). Best Practices Evidenced for Software Development Based on DevOps and Scrum: A Literature Review. *Applied Sciences*, *15*(10), 5421.

El Aouni, F., Moumane, K., Idri, A., Najib, M., & Jan, S. U. (2025). A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits. *Information and Software Technology*, *177*, 107569.

Kose, B. O. (2024). Mobilizing DevOps: exploration of DevOps adoption in mobile software development. *Kybernetes*. https://doi.org/10.1108/k-04-2024-0989

Jošt, G., & Taneski, V. (2025, April). State-of-the-Art Cross-Platform Mobile Application Development Frameworks: A Comparative Study of Market and Developer Trends. In *Informatics* (Vol. 12, No. 2, p. 45). MDPI.

Emmanni, P. S. (2023). Comparative Analysis of Angular, React, and Vue. js in Single Page Application Development. *International Journal of Science and Research*, *12*(6), 2971-2974.

Jayasankar, S. (2024). *Research on software testing and automation*. University of Illinois Chicago. https://www.cs.uic.edu/~i440/Grad%20Papers/440%20Fall%202024/Papers/Sanjith%20Jayasankar%20-%20Research%20on%20Software%20Testing%20and%20Automation.pdf

Tramontana, P., Marín, B., Paiva, A. C., Mendes, A., Vos, T. E., Amalfitano, D., ... &

Fasolino, A. R. (2024, May). State of the practice in software testing teaching in four European countries. In *2024 IEEE Conference on Software Testing, Verification and Validation (ICST)* (pp. 59-69). IEEE.

Prameswari, A. P. T., & Setyawan, M. Y. H. (2025). Development of a hybrid model for website functional evaluation based on ISO/IEC 25010 and SERVQUAL. *Matrix: Jurnal Manajemen Teknologi dan Informatika*, *15*(1), 32-39.

Proceedings of the Widyatama International Conference on Engineering 2024 (WICOENG 2024). (n.d.). Google Books. https://books.google.com.ph/books?hl=en&lr=&id=EfY6EQAAQBAJ&oi=fnd&pg=PA139&dq=System+Integration+Testing+ISO+25010+&ots=Wj7f1i5pIZ&sig=kld3kzrNG1ZpWgPq-F2SS_DBAkc&redir_esc=y#v=onepage&q&f=false

Wang, Z., Wang, W., Li, Z., Wang, L., Yi, C., Xu, X., ... & Zhou, J. (2024). Xuat-copilot: Multi-agent collaborative system for automated user acceptance testing with large language model. *arXiv preprint arXiv:2401.02705*.

Raznida, I., Shuhadah, O., Madya Dr Adam Shukry, A., & Noraliza, A. (2024). Journal of Advanced Research in Applied Sciences and Engineering Technology-2024: FYPMS User Acceptance Testing: Ensuring User Satisfaction and System Success. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *61*(1), 23-31.