PROJECT PROPOSAL

STUDENT DETAILS

NAME: MICHAEL OTIENO KASUKU

REGISTRATION NUMBER: COM/B/01-00162/2021

COURSE CODE: BCS 417

COURSE TITLE: COMPUTER SCIENCE PROJECT II

PHONE NUMBER: 0742644460

EMAIL: michaelotienokasuku@gmail.com

SUPERVISOR: MR. APOLLO WALIARO

PROJECT TITLE

Runway Master

PROJECT DESCRIPTION

Simulation and Optimization of Flight Scheduling and Runway Allocation Utilizing Discrete Event Simulation (DES) and Non-dominated Sorting Genetic Algorithm II

ABSTRACT

"Runway Master" presents a comprehensive approach to optimizing flight scheduling and runway allocation at Airport A, addressing complex constraints and objectives through a blend of Discrete Event Simulation (DES) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II).

The problem statement outlines the intricate interplay of airport infrastructure, airline operations, and flight details, emphasizing the need for efficient resource management while considering time constraints and maximizing resource utilization.

1. INTRODUCTION

1.1 Background of the study

The study delves into the intricate domain of airport operations, focusing specifically on the optimization of flight scheduling and runway allocation at Airport A. Amidst the complexities of shared resources and diverse airline operations, the necessity to efficiently manage runways and aprons while adhering to stringent time constraints becomes paramount.

2.PROBLEM DEFINITION

2.1 Current system workflow

The current system workflow of flight scheduling and runway allocation at Airport A entails a series of interconnected processes aimed at managing the complex dynamics of airport operations. Initially, the system gathers information regarding the number of runways, aprons, airlines, and aircraft registered to operate at the airport, establishing the foundational infrastructure parameters.

Subsequently, airlines submit their flight schedules, specifying departure and arrival times for each flight, alongside time requirements for apron and runway usage. The system then undertakes resource allocation, ensuring that only one flight occupies a runway or apron at any given time, while also considering the flight capacity of each aircraft.

Throughout the day, flights are scheduled and managed within the constraints of time and available resources, with the system continuously monitoring and updating the status of runways, aprons, and flights.

2.PROBLEM DEFINITION

2.2 Weaknesses of the current system

- Suboptimal utilization of runways and aprons leading to increased flight delays and prolonged ground times.
- Limited scalability, struggling to efficiently manage an increasing number of flights, airlines, and aircraft.
- Insufficient use of advanced optimization techniques and data analytics for predictive insights and improved decision-making.

2.PROBLEM DEFINITION

2.3 The proposed idea: Runway Master

"Runway Master," is designed to enhance the efficiency of flight scheduling and runway allocation at Airport A through the integration of Discrete Event Simulation (DES) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II).

DES models the operation of the airport as a sequence of time-driven events, allowing precise tracking and allocation of runways and aprons, and ensuring that constraints such as single occupancy are strictly followed. This event-driven approach enhances real-time adaptability and resource utilization.

NSGA-II, on the other hand, is employed to optimize the scheduling process by balancing multiple conflicting objectives, such as minimizing delays and maximizing resource utilization. This algorithm sorts potential solutions into different fronts based on Pareto dominance, maintaining diversity and exploring trade-offs between objectives

2.PROBLEM DEFINITION

2.3.1Why use Discrete Event Simulation(DES)

Event-Driven Model: DES models the operation of a system as a sequence of events in time.

Handling of Resource Allocation: DES can efficiently manage shared resources such as runways and aprons.

Time Management: DES excels at managing systems where time progression is uneven and event-driven, which fits well with your requirement to model flights with specific departure and arrival times.

Scalability: DES is well-suited for complex systems with a large number of interacting components, such as multiple airlines and aircraft in an airport.

Flexibility and Customization: DES models can be easily customized and extended to incorporate additional rules and constraints as needed. For instance, if additional constraints on specific flight schedules or preferential runway assignments are needed, they can be added to the simulation model.

2.PROBLEM DEFINITION

2.3.2 Why use Non-dominated Sorting Genetic Algorithm II(NSGA-II)

Multi-Objective Optimization: NSGA-II is designed to handle multiple conflicting objectives simultaneously, which is essential for this problem where you might need to optimize for minimizing delays, maximizing resource utilization, and balancing the load across runways and aprons.

Non-Dominated Sorting: NSGA-II sorts the population into different fronts based on Pareto dominance. This allows it to maintain a diverse set of solutions and helps in exploring trade-offs between different objectives.

Crowding Distance: The algorithm uses crowding distance to maintain diversity within the population, ensuring a wide exploration of the solution space. This is crucial for avoiding premature convergence and finding a well-distributed set of optimal solutions.

Elitism: NSGA-II incorporates elitism, which ensures that the best solutions found so far are carried over to the next generation. This improves convergence speed and the quality of solutions.

Scalability: NSGA-II is efficient and scalable, making it suitable for large-scale problems with many variables, like the flight scheduling and runway allocation scenario.

3.OBJECTIVES

3.1 Main Objective

To design and develop a sophisticated and efficient system for optimizing flight scheduling and runway allocation at airports.

3.OBJECTIVES

3.2 Specific Objectives

- To conduct a comprehensive analysis of the requirements gathered from stakeholders at the conceptual design phase.
- To develop a detailed system design based on the conceptual design, outlining the architecture, components, and data structures.
- To implement the functionalities and features outlined in the conceptual design.
- To conduct thorough testing at each iteration to ensure that individual components and features meet specified requirements.

4.PROBLEM JUSTIFICATION

4.1 Rationale of automating the current system

- **Increased Efficiency:** Automation can significantly improve the efficiency of resource allocation by leveraging algorithms and simulations that optimize the use of runways and aprons, reducing delays and enhancing overall airport throughput.
- **Scalability:** An automated system can better handle the growing complexity and volume of flights, airlines, and aircraft, scaling seamlessly to accommodate increased demand without a corresponding rise in operational delays or resource contention.
- **Enhanced Decision-Making:** By integrating advanced optimization techniques and data analytics, an automated system can provide predictive insights and more sophisticated decision-making capabilities, leading to improved resource management and operational planning.
- **Consistency and Accuracy:** Automated systems reduce the risk of human error and ensure consistent application of scheduling rules and resource allocation policies, leading to more reliable and predictable airport operations.

4.CONCEPTUAL DESIGN

4.1 Features

- **Discrete Event Simulation (DES) Integration:** Event-driven modeling to simulate airport operations, including flight arrivals, departures, runway occupancy, and apron availability. Real-time resource tracking to ensure accurate and timely allocation of runways and aprons.
- **Non-dominated Sorting Genetic Algorithm II (NSGA-II) Optimization:** Multi-objective optimization to balance conflicting objectives such as minimizing delays and maximizing resource utilization.Pareto dominance sorting to organize solutions into different fronts, maintaining a diverse set of optimal solutions.
- **User Interface:** Intuitive visualization tools to display flight schedules, resource allocation, and operational status.Interactive controls for manual adjustments to schedules and allocations, providing a balance between automation and human oversight.

4.CONCEPTUAL DESIGN

4.2 Technology Infrastructure

- **Programming Languages and Frameworks:**
  Java: Used for the backend logic, simulation modeling, and algorithm implementations due to its versatility and robustness.
  Swing: Employed for the graphical user interface (GUI) development, providing a platform-independent and lightweight framework for creating desktop applications.
- **Data Structures and Algorithms:**
  Priority Queues: Used for event scheduling and management within the discrete event simulation, ensuring efficient handling of events based on their occurrence time.
  Genetic Algorithms: Implemented for population-based optimization in NSGA-II, facilitating the evolution of candidate solutions through crossover, mutation, and selection mechanisms.

4.CONCEPTUAL DESIGN

4.3 Steps to implement Discrete Event Simulation

1. Model Events
Define key events such as flight arrival, flight departure, runway availability, and apron availability.
Each event will include the specific actions taken (e.g., allocate a runway, release an apron).

2. State Variables
Track the state of each runway and apron (occupied or free).
Track the schedule and state of each aircraft (e.g., current flight, next flight, time to next event).

3. Event Scheduling
Implement a scheduling mechanism to queue events based on their occurrence time. Common data structures used are priority queues or event lists.

4. Simulation Clock
Use a simulation clock to manage the progression of time within the model. The clock advances to the next event time.

5. Resource Allocation Logic
Implement the logic to allocate and release runways and aprons based on event triggers and current system state.

6. Output and Analysis
Collect and analyze data on resource utilization, flight delays, and overall efficiency of the scheduling process.

4.CONCEPTUAL DESIGN

4.4 Steps to implement Non-dominated Sorting Genetic Algorithm II

1. Encoding Solutions
Represent each potential flight schedule and resource allocation as a chromosome. This could be a string of genes where each gene represents a specific flight's departure time, arrival time, runway, and apron allocation.

2. Initialization
Generate an initial population of feasible solutions. Ensure diversity in the initial population to explore a wide solution space from the start.

3. Fitness Function
Define a multi-objective fitness function. Key objectives could include:

Minimizing total delay (sum of individual flight delays).
Maximizing resource utilization (effective use of runways and aprons).
Balancing resource usage (even distribution of flights across runways and aprons).

4. Selection
Use binary tournament selection based on non-dominated sorting and crowding distance to select parent solutions for reproduction.

5. Crossover and Mutation
Apply crossover (e.g., uniform crossover, one-point crossover) to combine parent solutions and produce offspring.
Apply mutation (e.g., random changes to flight times or resource assignments) to introduce variability.

6. Non-Dominated Sorting
Sort the population into different fronts based on Pareto dominance. Solutions in the first front are non-dominated with respect to the entire population.

7. Crowding Distance Calculation
Calculate the crowding distance for each solution to maintain diversity. Solutions with larger crowding distances are preferred during selection.

8. Elitism and Generation Update
Combine the parent and offspring populations and select the best solutions based on non-dominated sorting and crowding distance to form the new generation.

9. Termination
Continue the evolution process for a specified number of generations or until convergence criteria are met (e.g., no significant improvement in the best solutions).

5. METHODOLOGY

5.1 Iterative and Incremental Development Model

1. Requirement Analysis
Gather requirements from stakeholders, including airport authorities, airlines, and air traffic controllers, to understand the specific needs and constraints of the airport environment.
Define objectives, such as minimizing delays, maximizing resource utilization, and balancing runway and apron allocations.

2. Design Phase
Architect the system, dividing it into modular components for simulation, optimization, and user interface.
Design the data structures and algorithms for simulating flight operations, including event-driven modeling of arrivals, departures, and resource allocations.
Define the optimization approach, including the integration of NSGA-II for multi-objective optimization of flight schedules and resource allocations
Design the user interface to provide intuitive controls for interacting with the simulation and viewing results.

3. Development
Implement the simulation engine using Java, incorporating discrete event simulation techniques to model airport operations accurately.
Develop the NSGA-II optimization algorithm, integrating it with the simulation engine to optimize flight schedules and resource allocations.
Implement the user interface using Java Swing, creating components for inputting flight data, configuring simulation parameters, and visualizing simulation results.
Ensure robust error handling and exception management to maintain system stability and reliability.

4. Testing
Conduct unit tests to verify the functionality of individual components, including simulation logic, optimization algorithms, and user interface elements.
Perform integration testing to ensure seamless interaction between simulation, optimization, and user interface modules.
Conduct system testing to validate the system against predefined requirements, including performance testing under varying workload and scenario conditions.
Solicit feedback from stakeholders and end-users to identify any usability issues or functional gaps.

5. Deployment
Package the Runway Master system into executable files or installers for distribution to end-users.
Provide clear installation instructions and documentation to guide users through the setup process.
Ensure compatibility with different operating systems and environments to maximize accessibility and usability.

5. METHODOLOGY

5.2 Functional Requirements

REQ-1: The system shall generate simulated flight schedules based on user-defined parameters, including the number of airlines, aircraft, and flights per aircraft.

REQ-2: The system shall simulate flight arrivals, departures, and resource allocations based on predefined time constraints and resource availability.

REQ-3: The system shall provide real-time updates on the status of flights, runways, and aprons during the simulation.

REQ-4: The system shall implement the NSGA-II algorithm to optimize flight schedules and resource allocations based on multiple objectives, such as minimizing delays and maximizing resource utilization.

REQ-5: The system shall identify Pareto optimal solutions representing trade-offs between conflicting objectives, allowing users to select preferred solutions based on their priorities.

5. METHODOLOGY

5.2 Functional Requirements

REQ-6: The system shall offer intuitive forms for users to input parameters such as flight schedules, runway capacities, and apron availability.

REQ-7: The system shall provide controls for starting, pausing, and resetting the simulation, allowing users to manage the simulation process.

REQ-8: The system shall display visual representations of flight schedules, runway allocations, and resource utilization using charts, graphs, and diagrams.

REQ-9: The system shall allow users to configure simulation parameters, including time constraints, resource capacities, and optimization objectives.

REQ-10: The system shall enable users to interact with simulation elements, such as dragging flights to different runways or adjusting departure times.

5. METHODOLOGY

5.2 Functional Requirements

REQ-11: The system shall calculate and display performance metrics such as total delay, resource utilization, and balance across runways and aprons.

REQ-12: The system shall store and retrieve historical simulation data for analysis and comparison, allowing users to track performance trends over time.

REQ-13: The system shall enable users to export simulation results and reports in various formats, such as CSV or PDF, for further analysis or documentation.

REQ-14: The system shall allow users to create and manage multiple simulation scenarios, enabling testing of different configurations and scenarios.

REQ-15: The system shall support customization of simulation constraints and rules, allowing users to model specific airport environments and operational requirements.

5. METHODOLOGY

5.2 Functional Requirements

REQ-16: The system shall provide options to adjust parameters of the NSGA-II algorithm, such as population size and crossover rate, to fine-tune optimization performance.

REQ-17: The system shall detect and report errors or inconsistencies in input data, simulation parameters, or optimization results, providing feedback to users for correction.

REQ-18: The system shall log simulation events, optimization iterations, and user interactions for auditing, troubleshooting, and performance analysis purposes.

5. METHODOLOGY

5.3 Non Functional Requirements

NFR-1:Performance: The system shall be capable of handling large-scale simulations with thousands of flights and extensive optimization computations, ensuring responsive performance within acceptable time frames.

NFR-2: Scalability: The system architecture shall be designed to scale vertically and horizontally to accommodate increasing computational demands and user loads without significant degradation in performance.

NFR-3: Reliability: The system shall maintain high availability and uptime, minimizing downtime for scheduled maintenance or unexpected failures, with a target uptime percentage of at least 99.9%.

NFR-4: Security: The system shall implement robust security measures, including user authentication, data encryption, and access controls, to protect sensitive information and prevent unauthorized access or data breaches.

NFR-5: Usability: The system interface shall be intuitive, user-friendly, and accessible to users with varying levels of technical expertise, promoting ease of use and reducing the learning curve for new users.

5. METHODOLOGY

5.3 Non Functional Requirements

NFR-6: Accessibility: The system shall comply with accessibility standards (e.g., WCAG) to ensure equal access and usability for users with disabilities, including support for screen readers, keyboard navigation, and alternative input methods.

NFR-7: Compatibility: The system shall be compatible with a wide range of operating systems (e.g., Windows, macOS, Linux), ensuring seamless access and functionality across different platforms.

NFR-8: Interoperability: The system shall support interoperability with external systems and APIs (e.g., airline databases, airport management systems) for seamless data exchange and integration, facilitating collaboration and data sharing.

NFR-9: Maintainability: The system shall be designed with modular, well-documented code and clear architectural patterns to facilitate ease of maintenance, updates, and future enhancements by development teams.

NFR-10: Performance Efficiency: The system shall optimize resource utilization (e.g., CPU, memory, network bandwidth) to maximize computational efficiency and minimize energy consumption, promoting environmental sustainability.

5. METHODOLOGY

5.3 Non Functional Requirements

NFR-11: Data Integrity: The system shall ensure the integrity and consistency of data stored and processed, implementing mechanisms for data validation, error detection, and recovery to prevent data corruption or loss.

NFR-12: Compliance: The system shall comply with relevant regulatory requirements and industry standards (e.g., GDPR, FAA regulations) governing data privacy, security, and aviation operations, ensuring legal and ethical compliance.