الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة .مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Master's thesis in computer science

Speciality: Réseaux Informatique et Systèmes Répartis (RISR)

Theme

Performance Evaluation of Server Scheduling System

- **Presented by:**

BOUCHIKHI Mustapha

MORADJI FATMA

- **Directed by:**

Dr. MEKOUR Mansour

academic year 2024-2025

Thanks

I would first like to express my deep gratitude to Mr. Mekkour Mansour, my Supervisor, for his availability, his wise advice, his constant support and the quality of his Supervision throughout this work. His scientific rigor and kindness were for I am a great support. I also thank the members of the jury for They do me the honor of accepting To evaluate this thesis, as well as for their constructive and enriching comments. My thanks also go to all the university teachers for the quality of The teaching they gave me throughout my journey, and for their dedication Who greatly contributed to my academic and personal training. Finally, I would like to express my sincere thanks to my family for their support. unconditional, their patience and their encouragement throughout these years of study. Their presence at my side has been a constant source of motivation.

Dedication

To my dear family, the pillar of my life, and to my precious children, infinite sources of inspiration And motivation. To my little family, for your unconditional love and patience during this journey Demanding. To all the people who supported me, with their advice, their help or their presence, I I extend my deepest gratitude to you. This memoir is the fruit of your support and our Shared path.

Bouchikhi Mustapha

إهداء

إلى الذين علّموني أن الإرادة قادرة على كسر القيود،
وأن السعي نحو العلم لا تحدّه الأعمار، ولا تُوقفه الحدود...
إلى والديّ العزيزين،
تاج رأسي ونور دربي،
شكرًا لقلوبكما التي كانت لي وطنًا، ولدعائكما الذي كان سندًا.
أبنائي الأحباء: زواوي، حسّام، وملاهك...
هذه الصفحات ليست مجرّد مذكرة،
بل رسالة مفتوحة تنبض بالأمل، وتهمس لكم:
لا مستحيل مع الإيمان،
ولا شيء يُعيق من يملك عزيمة حقيقية.
مبروك نيلك شهادة الليسانس من جامعة مولاي الطاهر.
كان لي الشرف أن أشاركك هذه الفرحة،
لحظة نادرة نقشتها الذاكرة، وخلّدها الفخر في قلبي.
الذي يشقّ طريقه بثبات نحو التخرج من كلية التكنولوجيا،
أدعو الله أن يفتح لك أبواب النجاح،
ويجعل تعبك بداية لقصة مجد تُروى.
مبروك نجاحك في شهادة التعليم المتوسط .(BEM)
خطوتك الأولى كانت واثقة،
أسأل الله أن تكوني شعلة علم وضياء.
إلى أساتذتي في قسم الإعلام الآلي - جامعة مولاي الطاهر بسعيدة،
لكم كل الامتنان والتقدير،
فكنتم مشاعل علم، وعناوين إخلاص.
إلى زملائي في ثانوية طالبي بوطالب،
كنتم العون والسند،
وشكرًا لكل لحظة دعم، سواء أُظهرت أو خبّأها الصمت.
إلى كل من ترك أثرًا طيبًا في قلبي،
بكلمة، بابتسامة، أو بموقف لا يُنسى...
أهديكم هذا العمل المتواضع،
وفاءً، وامتنانًا، وحبًا لا يُقاس.

Contents

List of Figures

List of Tables

General introduction

In today's rapidly evolving digitalization and growing productivity demands, optimal job scheduling in a production or processing system is becoming a strategic issue. The job shop scheduling problem (JSSP) is one of the most complex combinatorial problems in operations research. It aims to organize the execution of a series of tasks on a set of machines, while respecting strict precedence and capacity constraints. This thesis addresses this issue by implementing a bio-inspired metaheuristic, the Ant Colon Optimization (ACO) algorithm, for solving the JSSP. The main objective is to evaluate the effectiveness of this method via an interactive software platform, through concrete experiments and the analysis of the makespan performance criterion. This work is structured around four progressive chapters, ranging from fundamental concepts related to order management, to the experimental simulation of the results. The aim is to demonstrate the relevance of ACO in a realistic application environment, with an emphasis on convergence, the quality of solutions, and prospects for improvement.

Problems and objectives

The workshop scheduling (Job Shop Scheduling Problem, JSSP) is one of the most complex combinatorial problems in operations research. It consists of determining the optimal sequence of execution of a set of tasks on several machines, while respecting the constraints of precedence and capacity, with the aim of minimizing the makespan (total production time). The JSSP is a problem NP-hard, which makes exact methods impractical for real-size instances. Faced with this complexity, the metaheuristics, and in particular the ant colony algorithm (Ant Colony Optimization, ACO), have proven to be effective approaches to finding quality solutions within reasonable timeframes.

Objectives of the thesis

This research work aims to:

- Study the mathematical modeling of the JSSP and its specific constraints.
- Implement and adapt the ACO algorithm to solve the JSSP.

- Develop an interactive software application providing the user with a graphical interface to:

- the task entry (ID, machine, duration),

- the ACO parameter adjustment (number of ants, number of iterations, α, β, evaporation rate),

- the launch of optimization and the visualization of results.

- Analyze the results obtained, based mainly on the makespan, the convergence curve and the Gantt chart, and compare performance.

Table 1: Summary of objectives

Objective	**Description**
Mathematical modeling	Translate the JSSP constraints into a formal model
Implementation of the ACO	Adapt the algorithm to the specificities of the problem
Software development	Create a graphical interface for data entry, configuration and analysis
Comparative analysis	Evaluate results compared to manual planning

Methodological approach

The methodology followed in this thesis is structured around the following steps:

1. Literature review: analysis of existing work on JSSP, classical approaches and metaheuristics, with a particular emphasis on ACO.

2. Modeling: formalization of the problem and definition of the algorithmic parameters.

3. Application Development: development of software in Python 3.11, integrating a graphical interface PyQt5, allowing the task entry and the ACO settings.

4. Experimentation: simulations on test data sets.

5. Analysis of results: performance assessment through the makespan, the convergence curve and the Gantt chart.

Structure of the thesis

The thesis is structured to guide the reader from theoretical concepts to concrete application:

Table 2: Steps in the methodological approach

Stage	Description
Literature review	Analysis of existing approaches
Modeling	Mathematical definition of the problem
Development	Application and interface programming
Experimentation	Tests on data sets
Analysis of results	Makespan evaluation, convergence, Gantt chart

- Chapter 1: Sample The art of order management for operational excellence.
- Chapter 2: Theoretical framework of the workshop scheduling problem.
- Chapter 3: Evolution of scheduling algorithms.
- Chapter 4: Experimental simulation and analysis of results (makespan, Gantt chart, convergence curve. example Standard instances. Real case study.).
- General conclusion: Summary of contributions, limitations, and research perspectives.

Glossary and Abbreviations

- OCC: Planning customer orders according to priorities and deadlines.
- JSSP: Workshop planning problem with sequencing constraints.
- ACO: Algorithm inspired by the behavior of ants for optimization.
- GA: Genetic algorithm based on natural selection.
- PSO: Particle swarm optimization, inspired by collective behavior.
- ERP: Integrated enterprise resource management system.
- MY: Real-time production monitoring and control system.
- Makespan: Total time to complete all planned operations.
- Metaheuristics: Optimization algorithm inspired by nature.
- Simulation: Numerical modeling of a real system.
- Heuristic: Empirical method to guide the search for solutions.
- NP-hard: Complex problem to be solved exactly in a reasonable time.
- Combined scheduling: Coordination between customer orders and workshop.
- Optimization: Search for the best possible solution.
- Local search: Progressive improvement of an existing solution.
- Dynamic adaptation: Automatic adjustment to changes in real time.

Chapter 1

The art of order management for operational excellence

Introduction

Order management is the foundation of a company's logistics performance. This chapter aims to analyze the different stages of this process, from the handover to The expedition, emphasizing digital tools such as the Management System Orders (OMS). The objective is to highlight the optimization levers Operational which directly influence the quality of service and customer satisfaction.

1.1 Mastering order preparation, a strategic challenge

In the fast-paced world of modern logistics, order fulfillment is much more than just a series of steps. It's the beating heart of the supply chain, the nerve center where the customer promise is delivered. This chapter takes an in-depth look at this essential discipline, seeking to answer a fundamental question: how can order fulfillment processes be mastered to minimize costs and maximize customer satisfaction? We'll delve into the mysteries of picking methods and processes—the techniques that determine the speed and accuracy of fulfillment. We'll then examine the critical importance of order preparation areas and their optimal sizing, the lifeblood of the warehouse. Finally, we'll conclude with an analysis of the key optimization factors that transform a simple logistics operation into a decisive competitive advantage. The goal is clear: to provide you with the tools and knowledge to transform your ordering process into a model of efficiency and profitability.

1.2 Order management: When the customer promise takes shape

Order management is the holistic process that begins the moment a customer confirms a purchase and ends when the product or service is delivered. It involves the meticulous orchestration of every step, from order receipt to final delivery, including inventory visibility and shipment coordination. While workflows can vary depending on the specifics of each business, a typical order management process generally revolves around three interdependent phases:

1.3 Ordering: The starting point of the customer journey

This first step is the entry point for the entire supply chain. The customer initiates their order, often through an automated interface such as an e-commerce website, a mobile app, or an electronic ordering system. Once the order is submitted, it is typically reviewed by a sales team member or an automated system that validates the details (availability, price, customer information). Concrete example: Imagine a customer purchasing a new smartphone from an online retailer. Once payment is validated, the order is transmitted to the company's order management system. At this point, the system checks the availability of the model, the chosen color, and the delivery address. If everything is in order, the order is confirmed to the customer via email, marking the beginning of the fulfillment phase.

1.4 Order execution: The operational heart

This is where the action takes place, primarily within the warehouse. A worker or automated system confirms the shipping details, generates the invoice, and initiates the preparation process. This phase includes crucial operations: Collection (picking): Locate and retrieve items from inventory. The packaging: Protect items and prepare them for shipping. The expedition: Hand the package over to the carrier. Concrete example: For our smartphone, the warehouse receives the order. An employee (or a robot in an automated warehouse) goes to the smartphone's precise location, picks it up, and then brings it to the packing station where it is carefully protected in a box along with the accessories. A shipping label is printed, and the package is then handed over to a carrier like Chronopost or DHL.

1.5 Inventory Management: The Pulse of Availability

Inventory management is a continuous and essential step that ensures real-time visibility of inventory. Stock levels are constantly updated as goods are received and orders are shipped. Accurate inventory management is essential to avoid stockouts, optimize replenishment, and ensure reliable delivery promises to customers. Concrete example: As soon as the smartphone is removed from stock, the inventory management system updates the available level. If the company only has five copies of that model, after removal, the stock will display "4." This helps prevent selling a product that is no longer in stock and triggers a restocking alert if the level drops below a critical threshold.

Figure 1.1: The global order management cycle in a warehouse

1.6 The Order Management System (OMS): The digital control tower

An Order Management System (OMS) is the digital backbone that orchestrates the entire order lifecycle. More than just a tracking tool, the OMS centralizes and manages all information and processes related to an order, from its initial entry to its after-sales service. It provides real-time visibility for both the company and the customer, transforming the purchasing experience. Concrete example: For our customer who ordered the smartphone, the OMS allows them to track their package step by step: "Order confirmed," "In preparation," "Shipped," "In delivery," "Delivered." At the same time, the company uses the OMS to have an overview of available inventory, carrier performance, and each customer's order history, thus facilitating returns or exchanges if necessary. It is this transparency and traceability that strengthens customer confidence.

1.7 The critical importance of customer order management

Order management is no longer confined to corporate premises. It is intrinsically linked to virtually every system and process in the supply chain, often involving a multitude of external partners: component suppliers, assembly services, packaging providers, and distribution centers. This increased complexity poses a major risk of loss of visibility and control, leading to costly and error-prone manual processes. An effective OMS then becomes a strategic lever. By automating repetitive tasks and reducing human error, it helps control costs and generate significant revenue.

1.7.1 The direct impact on customer experience

Externally, the quality of order management directly shapes a customer's perception of a company or brand. In an omnichannel environment where customers expect a seamless and consistent experience, impeccable order management is a key differentiator. Concrete example: A customer places their smartphone order online, but may have a question about delivery and contact customer service by phone. OMS ensures that the call center agent has access to all order information in real time, allowing them to respond accurately. During fulfillment, the customer receives updates via SMS or email. If, unfortunately, they need to return the smartphone, they expect to be able to do so through a physical channel, such as a store, without friction. Every interaction, every touchpoint, is an opportunity to deliver an excellent customer experience, strengthening loyalty and generating recurring revenue. Furthermore, a well-managed customer journey opens up opportunities for additional sales (upselling) and cross-selling (complementary products), thus increasing the average order value and revenue.

1.8 The essential features of effective order management

For order management to be truly effective, it must integrate key features that ensure fluidity, control and performance. Full visibility: The ability to visualize the entire supply chain, identify critical events and anticipate problems. Example: A WHO dashboard that shows in real time the location of each order, stock levels in different warehouses, and shipping statuses. Specific information: The ability to refine order management processes based on business rules and company performance objectives. Example: Configure the OMS to prioritize "express" orders or to automatically direct orders to the warehouse closest to the customer. Operational flexibility: The ability to break down orders into single work items, which can be routed to the appropriate systems or resources. Example: A complex order with multiple items can be divided into subtasks assigned to different picking areas or different pickers. Real-time stocks: A unified and accurate view of inventory, including items on hand, in transit, and current demand levels. This reduces the need for costly expedited shipments and excessive safety stock. Example: A customer sees ”in stock” on the website because the OMS is directly connected to the physical inventory and updates it instantly. Optimized planning of deliveries and services: The ability to match delivery commitments with inventory, available resources and required skills, thus ensuring more efficient management of service requests. Example: WHO may estimate delivery date based on carrier availability and customer delivery slots. Customer interaction technologies: Provide customer-facing staff with a complete view of the customer, inventory, and back-end resources, enabling them to execute transactions more efficiently. Example: A customer advisor can, through OMS, see a customer's purchase history, current order status, and even available return options. Execution optimization: Analyze data and recommend the best options for shipping orders, taking into account customer preferences, delivery time and cost. Example: The system can suggest the shortest delivery route or the most economical mode of transport depending on the destination and the urgency of the order.

1.9 Order Processing: From Click to Receipt

Order fulfillment is the entire process that begins when a customer makes an online purchase and ends when they open their package. It's a sequence of key logistics operations whose smooth operation is directly linked to customer satisfaction. To assess your efficiency, ask yourself this simple question: ”What, in my business, constitutes an efficiently processed order?” The most common answer? A fulfilled delivery promise and a loyal customer.

1.10 Key steps in order processing

Order processing is broken down into a series of distinct tasks, from online purchase to final product delivery. Here are the main ones: Verification of payment and delivery address: A basic but crucial step. This involves ensuring that the customer is solvent and that the delivery address provided contains no errors. An error here can lead to costly delays and customer dissatisfaction. Example: An e-commerce platform integrates secure payment gateways and address verification tools to minimize these risks. Stock availability: A sore point. The lack of a powerful IT tool for inventory automation can waste valuable time and hamper efficiency. It's vital to confirm that the items ordered are present and in sufficient quantity. Example: Without an OMS connected to inventory in real time, a business could accept an order for an out-of-stock item, creating major frustration for the customer. Carry out the picking (or sampling): This step involves locating and collecting the ordered item(s) from the warehouse shelves. This is one of the most costly and time-consuming operations in order preparation. Example: An order picker moves around the warehouse, equipped with a scanner and a picking list, to collect all the items in a customer's order. Packaging: Picked items must be carefully packaged to ensure safe transport and protect the product. This is the quintessence of ”order preparation.” Example: Use appropriate cushioning materials, correctly sized cartons, and packaging techniques that protect the product from impact and weather. The expedition: The package is handed over to the selected carrier and begins its journey to its final destination. Example: A partner carrier's truck (La Poste, UPS, etc.) arrives at the warehouse and loads the packages prepared for delivery. Editing the invoice: The invoice corresponding to the purchase is generated and given to the customer, either in paper form in the package or electronically.

1.11 Order preparation processes: The art of picking, sorting and packing

Whatever method is chosen, order preparation is based around three fundamental operations, which follow one another logically: Picking: The collection of items from their storage location. This is the stage where the product moves from stock to the preparation circuit. Sorting: Identifying and grouping items that belong to the same order line, the same complete order, or the same customer. This step is crucial for consolidating complex orders. Packaging: The packaging of prepared items, in order to facilitate handling and transport operations, while protecting the contents.

1.11.1 Sampling methods: Adapting to flows

The different picking methods are not mutually exclusive; it is common to combine them within the same warehouse to accommodate the diversity of orders and items. The ultimate goal is to satisfy requests as quickly as possible. Depending on numerous criteria specific to each manager, you can opt for: Batch Picking: Pick lists are processed as they arrive. Items are picked in the order in which they are received by the picker. This method is simple for small volumes or single orders. Example: An e-commerce merchant receives 10 orders for T-shirts. The order picker takes the first order, picks up the T-shirt, then the second, and so on. Group picking (Wave Picking or Order Picking): The software performs a grouping operation based on numerous customer orders and sorts the order items by reference on the picking lists. The lists are edited periodically. The picking is then followed by physical sorting operations by order and packaging. This method optimizes travel. Example: For the 10 T-shirt orders, the system groups all requests for ”blue T-shirt size M.” The picker retrieves 5 ”blue T-shirts size M” at once, then sorts them by order at a consolidation station. Zone Picking: Similar to group picking, this method has the particularity of sorting order items by storage location in the picking lists. This saves considerable time, as items are picked from the same storage area, minimizing travel. Example: A large warehouse is divided into zones (textiles, electronics, fresh produce). A picker only handles items in their zone. An order containing a T-shirt and headphones will be picked twice by two different pickers and then consolidated.

1.11.2 Sorting modes: Organize the flow of articles

Two main sorting methods are used to optimize order preparation: Computer sorting of items on the collection lists: Its main objective is to facilitate picking and save time. Sorting lists reduces the number of trips to the same location or storage area, thus optimizing picker routes. Example: The picking list is edited so that the picker follows a logical path through the warehouse, going from aisle 1 to aisle 10 without going back. Physical sorting of items after collection: Carried out in dedicated sorting centers, it aims to reconstitute customer orders and optimize delivery transportation. The items collected are grouped by order, by customer, and then by geographic area, thus facilitating shipping and reducing transportation costs. Example: After all the loose items have been collected, they are taken to a sorting area where operators or sorting machines distribute them into bins or boxes dedicated to each customer order.

1.11.3 Order preparation methods: Diversity of approaches

The organization of order preparation varies considerably depending on volumes, the nature of the products and technological investments. Here are the most common methods:

The “Pick then Pack” retail preparation method: Consolidation before shipping

This is a two-step picking method. First, items are picked from their storage location based on picking lists (often group picking or zone picking). They are then moved to a separate order picking area where they will be sorted, grouped by order, and packed. Example: A large distribution center uses pickers who pick entire pallets of various products for multiple orders. These pallets are then transported to a consolidation area where specialized operators sort the items for each individual order and pack them. This method is ideal for optimizing picker movements across large areas.

The “Pick and Pack” retail preparation method: Live packaging

Thanks to a suitable and well-informed computer system, a box with a shipping label is directly prepared and can contain the total volume of items in a single order. Accompanied by the list of order items, this box is sent to the various picking points to be supplied. This method requires high investments in technology and dynamic conveyor systems. Example: In a highly automated warehouse, an empty carton labeled for a customer’s order moves along a conveyor. It stops in front of each picking area where an item from the order is added. Once all the items are loaded, the carton is directly sealed and ready for shipment, minimizing intermediate steps.

The “Pick to Light” assisted preparation mode: The light guides the preparer

Particularly suited to small component storage areas or high-turnover areas, this picking method uses lighting devices to guide the picker. Each storage location (drawer, bin, cell) is equipped with a digital display and an indicator light. The indicator light indicates the picking address, while the digital display indicates the quantities to be picked. A push button allows the picker to signal to the system that the pick has been completed. Example: In an electronics warehouse, an order for 5 resistors and 2 capacitors arrives. The ”Pick to Light” system turns on a green light on the resistor shelf and displays ”5” on the screen. The picker takes 5 resistors and presses the button. Then, a light comes on on the capacitor shelf, displaying ”2,” and so on. This method is extremely fast and significantly reduces picking errors.

Figure 1.2: The “Pick to Light” assisted preparation mode

1.12 The “Pick to belt” preparation mode: Mass picking towards the conveyor

This picking method is typical of storage areas equipped with a drainage conveyor, designed for mass picking. Boxes or bins containing large quantities of items are placed on the conveyor, which transports them directly to a centralized sorting center. Example: In a warehouse handling high volumes of similar products (e.g., books, clothing), pickers pick stacks of these items and place them directly onto a conveyor that winds through the storage aisles. This conveyor then transports the items to a sorting area where they are grouped by order.

1.13 Order preparation areas: Structuring warehouse efficiency

The layout of order preparation areas is crucial to optimize flows and productivity.

1.13.1 The Consolidation Zone: The Gathering Point

This area is dedicated to grouping all the preparations for a single order, especially those that come from multiple pickings (for example, if an order contains items from different areas of the warehouse). However, some warehouses choose to do without it for various reasons: If the majority of orders have only one line item, consolidation is less necessary. If the company decides that multi-line orders can be shipped in several separate packages. If the carriers are responsible for this grouping task. Finally, in the case of automatic transfer of preparations to downstream areas, the handling system can integrate a sorter which will carry out, at least partially, these groupings (for example, an automated sorter with diverging switches).

1.13.2 Packaging areas: The final touch before shipping

When packaging is required, it can be integrated directly into the consolidation area. Packaging can be entirely manual, assisted by semi-automatic equipment, or fully automated. The areas required will depend directly on the technological solutions being considered. It is essential to provide a nearby storage space for packaging materials (cardboard boxes, adhesive tape, cushioning, etc.), as well as for the computer peripherals needed to print shipping labels and slips. This is also the area where pallet wrapping or strapping devices will be installed if necessary, once all inspection operations have been carried out.

1.13.3 Departure control zones: Quality above all

The control of shipments aims to verify: The references of the items in the package. That their number corresponds to the quantities of the customer order. As part of a total quality approach and if an order is processed by a single preparer, this final check is sometimes omitted to save time. If this is not the case, arrangements can be made to facilitate it: Weight control: Comparison of the actual weight of the package with its theoretical weight (calculated from the unit weight of each item). Automatic identification of articles: If each item has an identification tag (barcode, RFID), the controller can scan this information using a reader connected to the system. The calculator then compares the list of identified items to the list of expected items, thus detecting any errors. To size this area, it is necessary to provide a buffer zone to desynchronize the arrival of items and inspection. Work tables, scales, and computer peripherals must be included in the planning.

1.14 Departure waiting areas: Anticipating hazards

The need for relatively large storage areas for orders ready for shipment is due to several reasons: Preparation hazards: Allow for a safety margin for unforeseen events. Carrier risks: Carrier delays require space to temporarily store packages. Smoothing of the activity of the preparers: Allow for longer preparation time slots that are offset from carrier collection times, thereby optimizing resource use. A specific waiting area may be dedicated to ”deadline” orders or export orders, which may remain for several days or even weeks, while customs documents are prepared, customer solvency is checked or special transport is organised. The surface area of these waiting areas will ideally correspond to the surface area of the vehicle platforms. They must be clearly marked (for example, by paint on the floor) to facilitate the work of forklift operators and to ensure compliance with the organization. The traffic aisles must be wide enough (approximately three meters) to allow easy maneuvering of pallet trucks when loading vehicles, as a reduction in width would have a direct impact on loading times. If the size of the docks and agreements with carriers allow it, trailers can be docked during preparation periods, allowing immediate loading and thus freeing up interior space while avoiding reloading.

1.15 Factors for optimizing order preparation: Towards logistics excellence

To achieve optimal effectiveness and efficiency in the order preparation process—whether in terms of costs, lead times, or quality—several areas for improvement can be implemented in a coordinated manner. This list, while not exhaustive, covers the most significant levers.

1.16 Weight control: A double-sided verification

The principle of weight control is appealing due to its apparent simplicity: it involves comparing the theoretical weight of a package (calculated from the unit weights of the items and packaging) with the actual measured weight. If a significant difference is detected, this indicates a preparation error. Often, weighing is done ”on the fly” on a conveyor at the end of the process. However, the reality is more complex, and this system has notable weaknesses, divided into two categories: ”false errors” generated by the system and ”real errors” that go undetected. Some of the most common ”false mistakes” include: Incorrect entry of logistics data: An error occurred when initially saving an item's weight to the database. Change in manufacturing or packaging: A change in the supplier or packaging of a product without updating the weights in the system. Example: If a shampoo bottle manufacturer changes the material of its bottles (from glass to lighter plastic), but the weight in the system remains that of glass, every shampoo order will be flagged as an ”error.” Multiple sources of supply (dual sourcing): Products or packaging from different suppliers may have slightly different weights. Example: Vials from two different glassmakers may not have the same weight, which will result in an erroneous alert if one of them is used. Variation in ambient humidity: Some materials can absorb moisture and become significantly heavier. Example: A rainy week can increase the weight by 10% on some paper or fabric-based items, triggering errors. Weighing system drift: Insufficient calibration or scale malfunction. In the category of undetected ”real errors”: Tolerance setting too lax: Too wide a tolerance range can mask sampling errors. Confusion between references of the same weight: The picker may have picked an incorrect item that has the same weight as expected. Example: Two computer mouse models from different brands but with almost identical weight. Error cancellation: Two errors, one plus and one minus, can cancel each other out in the total weight. Error on a low weight reference: An error on an item whose weight is less than the tolerance will not be detected. Weighing system drift (again): Continued inaccuracy of the scale. These weaknesses explain the strong stance of many critics of weight control. Studies have shown that the reported error rate is often around 22%, while the actual error rate is ten times lower, highlighting the need for a more nuanced and potentially more sophisticated approach.

1.17 The best solution: Towards integrated and targeted control

Faced with the limitations of global weight control, improvements have been proposed. The idea of placing the control station after the preparation of heavy items and before that of light items has sometimes been put forward. However, this approach does not solve all the problems and may even require a logistical layout that is contrary to the optimization of movements. Although it may seem more expensive in terms of initial investment, the best solution seems to lie in installing a control at the exit of each preparation station. This system would work as follows: Weight recording at entry: The system records the weight of the bin or package as soon as it arrives at the preparation station. Post-collection verification: After the picker adds the items, the system checks that the output weight of the station matches the initial weight plus the weight of the items that are supposed to be picked at that station. Action in case of error: If the weight does not match (within the tolerance), the bin or package is immediately immobilized at the station, triggering an alarm, or is automatically diverted to a dedicated manual control station. This provision is particularly relevant for retail preparation. For the control of standard packages, automatic identification of the reference (by barcode scanning or RFID) remains the most reliable method. In both cases, some sources of error remain, but this localized and integrated approach would achieve a much higher level of accuracy, drastically reducing shipping errors and improving customer satisfaction.

1.18 Reducing journey length: Optimizing the preparer's route

Minimizing the distances traveled by preparers is a major lever for productivity. ABC classification and ordering (Pareto's Law): Position the most frequently ordered items (category A) in the most accessible locations and those closest to the departure stations. Less frequently requested items (categories B and C) can be stored further away. Example: In a clothing store, the best-selling sizes and colors of jeans are within easy reach, while rarer sizes and colors are stored high up or at the back of the warehouse. Optimized preparation circuits: Design picking routes that prevent the picker from having to visit the same location multiple times. The picking order on the list should be logical and sequential. Example: Rather than a random pick list, the system generates a list that allows the picker to walk through an aisle only once, collecting all the necessary items along the way.

1.19 Limiting load breaks: Fluidity of movement

Breaks in load (when the flow of goods is interrupted or transferred from one medium to another) are sources of inefficiency. Rigor in reporting on goods movements: Strict discipline of pickers and forklift drivers is required to update stock locations and merchandise statuses in real time. Using a real-time warehouse management tool (WMS): A good Warehouse Management System allows you to accurately track inventory and movements, avoiding inconsistencies. Efficient replenishment of picking: Keep picking areas stocked so pickers don't waste time waiting or searching for items. Example: Setting up alert thresholds for replenishment, automatically triggering replenishment tasks before picking areas are empty.

1.20 Limitation of damage risks: Preserving the integrity of products

Minimize damage to products during preparation and transportation. Preparation methods respecting the nature of the products: Avoid preparing incompatible items at the same time. Example: Never store or collect toxic or chemical products with food products in the same bin or on the same trolley. Picking order respecting the weight and density of the items: Place heavier, denser items below fragile items to avoid crushing. Example: When the picker collects a batch of items, they start with the cans of food, then the cereal boxes, and finally the bags of chips.

1.21 Optimizing article accessibility: Making the preparer's work easier

Making items easily accessible reduces picking effort and time. Do not place heavy references at height: This prevents injuries to the preparers and makes sampling easier. Do not store items too deep: Avoid shelves that are too deep, which require effort to reach the products. Sufficiently wide preparation aisles: Allow preparers to pass and overtake each other without hindrance, thus reducing waiting times. Limiting the number of simultaneous preparations in the same area: Avoid ”traffic jams” in front of high-turnover reference locations, which can cause delays.

1.22 Using suitable storage furniture: The right tool in the right place

Choosing storage furniture based on product characteristics is essential. Storage furniture based on reference rotation: Use dynamic systems (conveyors, carousels) for fast-moving items to facilitate and speed up picking. Fixed shelving is suitable for slow-moving items. Storage furniture based on the volume and density of items: Adapt the size and robustness of shelves and bins to the dimensions and weight of the products.

1.23 Workstation ergonomics: Well-being at the service of performance

Optimizing the workstations of preparers is a key factor in performance and error reduction. Ergonomic layout of the front sampling: Ensure that the most frequently picked items are at an optimal height and reach for the picker, minimizing unnecessary bending and stretching. Product presentation: Organize items logically and visually clearly in storage locations for easy identification and retrieval. Taking into account the frequency of samples: Inventory management software should allocate locations based on picking frequency, placing high-turnover items in the most accessible and least remote areas.

1.24 Conclusion

Picking retail orders, whether managing thousands or tens of thousands of items, is not a simple execution task; it is a strategic discipline that requires careful study of the layout of picking areas and perfect control of the parameters that govern its supply. It is truly at this stage that a key part of overall performance in terms of productivity and meeting deadlines is played out. The quality of the picking setup, particularly around a mechanized chain, is a determining factor for a successful start and rapid ramp-up of the business. To achieve this, several conditions are necessary: Have or collect reliable and complete data: Without accurate knowledge of inventory, order volumes, product profiles, and movements, any optimization will be based on faulty assumptions. This is the foundation of any informed decision. Adopt a rigorous methodological approach: Implementing new methods or improving existing processes must follow a structured approach, from needs analysis to implementation and performance monitoring. Rely on an available and competent team: The transition to optimized processes, the adoption of new technologies, and daily maintenance require well-trained, committed, and adaptable employees. Their involvement is the key to operational success and continuous improvement. By investing in these pillars, companies can transform their order management into a real competitive advantage, ensuring not only fast and accurate deliveries, but also increased customer satisfaction and a significant reduction in operational costs. Order preparation, far from being a simple operation, is a strategic issue. Optimization of flows, ergonomics, stock management and the choice of methods of Picking not only reduces costs but also improves the experience Customer. Good order management is an essential pillar for any business. Focused on logistical excellence.

Chapter 2

Theoretical framework of the workshop scheduling problem

This chapter introduces the Job Shop Scheduling Problem (JSSP), an optimization problem central to production systems. After defining the key concepts (scheduling, planning, constraints), we mathematically formalize the JSSP and let's examine its characteristics, complexity and industrial applications.

2.1 Definition of the Job Shop Scheduling Problem (JSSP)

The Job Shop Scheduling Problem (JSSP) consists of determining the optimal order of execution of a set of jobs on multiple machines, while respecting precedence and capacity constraints. The main objective is generally to minimize the makespan, i.e., the total production time.
This problem is omnipresent in many industrial and logistics fields, such as manufacturing, project management, IT and even transportation[2].

2.1.1 Overview of the scheduling problem

2.1.2 Difference between planning and scheduling

- Planning: consists of anticipating and organizing all tasks over a time horizon, taking into account the overall objectives and major constraints of the company.

- Scheduling: intervenes after planning to concretely organize the order and timing of task execution, based on available resources and operational constraints. It aims to optimize the use of resources and minimize delays.

2.1.3 Definition of the Job Shop Scheduling Problem (JSSP)

The Job Shop Scheduling Problem (JSSP) involves determining the optimal sequence of operations on a set of machines, each operation having specific technological constraints. This problem is known for its complexity (NP-hard), making exact methods inapplicable on a large scale.

2.1.4 The Job Shop Problem (JSSP)

THE Job Shop Scheduling Problem (JSSP) is one of the most studied scheduling problems in operations research and industrial optimization. It consists of planning a set of jobs (orders or batches), each composed of a sequence of operations to be performed in a specific order on different machines. Each machine can only process one operation at a time, and each operation has a specific processing time.

Main features of JSSP:

- Imposed sequence: each job follows a strict operating range (order of operations).

- Limited resources: each machine can only perform one operation at a time.

- Multiple constraints: resource conflicts, priorities, deadlines, possible setup times.

- Classic goals: minimize total completion time (makespan), reduce delays, maximize resource utilization.

Table 2.1: Characteristics of the JSSP

Element	**Description**
Task (Job)	Ordered sequence of operations to be executed
Operation	Processing on a given machine for a specific duration
Machine	Resource that can process only one operation at a time
Constraints	Precedence, capacity, setup time, priorities, deadlines
Goals	Minimize makespan, delays, maximize resource utilization

- Analysis: JSSP is ubiquitous in manufacturing, logistics, and computing. Its optimal resolution is difficult (NP-hard), which justifies the use of metaheuristics.

The JSSP belongs to the class of NP-hard problems, meaning that finding an optimal solution quickly becomes unattainable as soon as the problem size increases. A very common classification of workshops, from a scheduling point of view, is based on the different machine configurations. The best-known models are those of a single machine, parallel machines, a single-path workshop or a multi-path workshop.

- Single machine

- Parallel machines

- Single-path workshops (Flow Shop)
- Multi-path workshops (Job Shop)

Multi-path workshops (Job Shop)

Workshop scheduling consists of organizing the operation of a workshop over time. Workshop to make the best use of the human resources and materials available in order to produce the desired quantities in the given time.
The following figure shows an example of a multi-path shop with four jobs and six machines

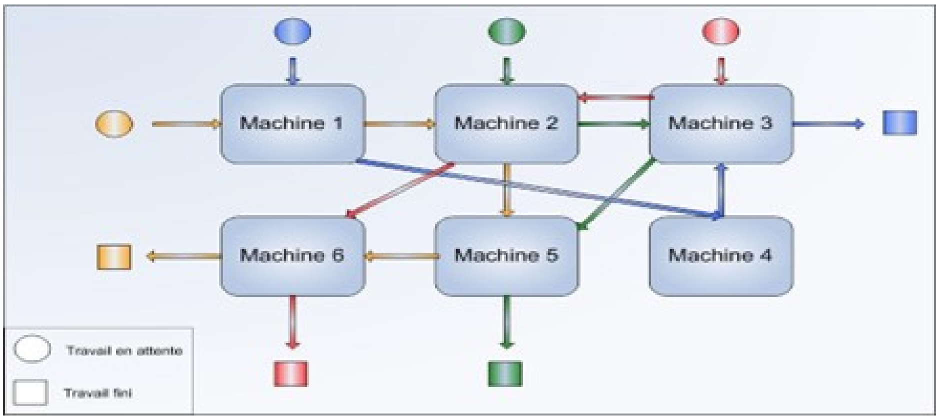

Figure 2.1: Graph illustrating the operations and their sequence in the workshop
Source: https://fr.wikipedia.org/wiki/Ordonnancement$_d$

Among the characteristics of a scheduling problem in a multi-path workshop:
- the number of possible solutions is of the order of $(n!)^m$, where n is the number of tasks to be performed and m the number of machines. Note that a task means the same thing as a job.
- the problem is NP-hard and is considered among the most difficult problems to treat.
- Traditionally, OCC and JSSP issues are addressed separately, which can generate conflicts between business priorities and technical feasibility of production. However, in industrial reality, the dissociation of these two dimensions often leads to delays, additional costs and customer dissatisfaction. The combinatorial complexity of the problem lies in the need to integrate:
- Customer constraints: priorities, delivery deadlines, late penalties,
- Technical constraints: precedents between operations, machine exclusivity, limited capacities

2.2 Mathematical modeling of the JSSP

Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if operation } i \text{ of job } j \text{ is assigned to position } k \\ 0 & \text{otherwise} \end{cases}$$

Objective function (Makespan minimization)

$$\min C_{\text{max}} = \min \left(\max_j C_j \right)$$

where C_j is the end date of the last operation of job j.

Constraints

1. Precedence of operations for each job:

$$S_{j,o+1} \geq S_{j,o} + p_{j,o}$$

   where $S_{j,o}$ is the start date of operation o of job j and $p_{j,o}$ its duration.

2. No overlap on machine:
   For any pair of operations (j, o) and (j', o') assigned to the same machine (m), and $(j, o) \neq (j', o')$:

$$S_{j,o} \geq S_{j',o'} + p_{j',o'} \quad \text{or} \quad S_{j',o'} \geq S_{j,o} + p_{j,o}$$

3. Uniqueness of passage on each machine:
   Each operation must be executed exactly once on the machine assigned to it.

$$\sum_{k=1}^{n} x_{ijk} = 1 \quad \forall i, j$$

2.3 Main formulas of the ACO applied to the JSSP

Probability of an ant selecting an operation

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \text{available}} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

Where:
- τ_{ij}: intensity of the pheromone on the arc (i, j)
- η_{ij}: heuristic information (often $\eta_{ij} = \frac{1}{p_{ij}}$)
- α, β: weighting parameters

Global pheromone update

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}$$

Where:
- ρ: evaporation rate ($0 < \rho < 1$)
- $\Delta\tau_{ij}$: quantity of pheromone deposited, generally:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

with m the number of ants and $\Delta\tau_{ij}^{k}$ the contribution of the k-th ant (often $\Delta\tau_{ij}^{k} = \frac{Q}{L_k}$ if the arc (i, j) belongs to the solution of ant k, 0 otherwise; Q is a constant and L_k the length of the solution).

2.4 Summary of ratings

- C_{max} : makespan (total production time)
  - C_j: end date of job j
  - $S_{j,o}$: start date of operation o of job j
  - $p_{j,o}$: duration of operation o of job j
  - τ_{ij}: intensity of the pheromone on the arc (i, j)
  - η_{ij}: heuristic information for the arc (i, j)

2.5 Problem complexity and computational difficulty

The JSSP is recognized as an NP-hard problem as soon as the number of machines exceeds two. This means that there is no known polynomial-time algorithm to solve it exactly in the general case, making exact methods inapplicable for large instances.

2.6 Examples of applications of scheduling

The JSSP finds applications in:
- Industrial production planning (manufacturing workshops, assembly lines)
- Project management (task scheduling)
- Computer science (process scheduling)
- Logistics (management of resources and delivery times)

2.7 Scheduling constraints in industry

Typical constraints encountered in industrial workshops include:
- Setup times
- The priorities of certain tasks
- Machine maintenance
- Production interruptions and hazards

These constraints must be integrated into the modeling and resolution of the problem, which justifies the use of flexible approaches such as metaheuristics.

2.8 Chapter Summary

This chapter presented the theoretical foundations of JSSP, its mathematical modeling, its complexity and its industrial applications. Understanding these aspects is essential for the development of suitable software tools, as illustrated in the practical part of this thesis, where the user enters tasks, machines and durations in a dedicated interface, thus translating the theoretical model into data usable by optimization algorithms.

2.9 Conclusion

Modeling the JSSP highlights the difficulty of this combinatorial problem. Due to its complexity (NP-hard), exact approaches are quickly limited. It is therefore imperative to turn to heuristic or metaheuristic methods, which are more flexible and adapted to industrial realities, paving the way for the ACO presented in the following chapter.

Chapter 3

Evolution of scheduling algorithms

Introduction

Given the impracticality of exact methods for scheduling problems Complex, metaheuristics have established themselves as powerful alternatives. Chapter traces the evolution of the methods, with emphasis on the colony algorithm Of ants (ACO), its biological foundations, its operating principles and its Optimization mechanisms applied to the JSSP.

Job-shop scheduling (JSSP) is a central problem in operations research and industrial production. Early approaches were based on exact methods such as Branch and Bound or linear programming, but their exponential complexity limits their use to small instances. . With the increase in the size of workshops and the diversity of industrial constraints, interest has turned to heuristic and metaheuristic methods. .

3.1 Limits of classical solutions

Classical methods, although effective for small problems, quickly become unsuitable as soon as the number of tasks or machines increases. Their main drawback lies in the combinatorial explosion of the search space and the inability to easily integrate new industrial constraints[5]. Thus, the use of approximate and adaptive solutions has become essential.

3.2 Metaheuristics and their application to the JSSP

Metaheuristics include tabu search, simulated annealing, genetic algorithms, GRASP, and ant colony optimization (ACO). These approaches allow to efficiently explore large solution spaces and to obtain results close to the optimal in a reasonable time. Their flexibility and ability to adapt to real constraints explain their growing success in the industry.

3.3 The Ant Colony Algorithm (ACO)

Painting3.1: Main parameters of the ACO

Setting	Role
α (alpha)	Influence of pheromone quantity
β (beta)	Influence of heuristic information
Number of ants	Size of the population of simultaneous solutions
Evaporation rate	Controls the gradual disappearance of the pheromone
Iterations	Number of research cycles

3.3.1 Ant System

The fundamental principle of this algorithm is based on a population of artificial ants that cyclically construct solutions to a combinatorial optimization problem.

The algorithm consists of modeling the problem as a graph, on which the ants move from one node to another following the branches (edges), thus forming paths representing potential solutions. Each ant starts from an initial node, and chooses the next node following a State Transition Rule defined as follows:

$$p_{ij}(t) = \frac{\sum_{k \in N_i} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}$$

where: - $\tau_{ij}(t)$ represents the quantity of pheromone present on edge $i \to j$ at time t; - η_{ij} is the heuristic information (often the inverse of the distance); - α and β are parameters which respectively determine the influence of the pheromone and that of the heuristic information.

If a node does not meet the problem constraints, its probability is set to zero. Thus, nodes with a high amount of pheromone and a low heuristic distance have a higher probability of being selected to complete the partial solution.

Once all ants have constructed a complete solution, that is, a valid sequence of nodes representing a solution path, the cycle is completed and a global pheromone update rule is applied:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

where: - ρ is the evaporation rate of the pheromone; - $\Delta\tau_{ij}(t) = Q/L_{best}$ if edge ij belongs to the best solution found so far, otherwise 0. Q is a constant fixing the quantity of pheromone deposited; L_{best} is the value of the evaluation function of the best known solution. This update involves two actions: 1. Evaporating some of the pheromone on all edges of the graph; 2. Adding pheromone only on the edges traveled by the best solution found so far. This allows to strengthen promising paths, while preventing stagnation in local minima, thanks to evaporation. Alternative update strategies: - A biological approach consists of allowing all ants to update the edges they have traveled, proportional to the quality of their solution. - However, simulations show that the elitist strategy, where only the best ant deposits pheromone, gives better results.

The simulator used allows these approaches to be combined and gives more importance to the best ant via an elitist parameter (e). In the rest of this work, the "best-so-far" strategy is the one chosen.

3.3.2 Biological principles

ACO is inspired by the collective behavior of ants, which are able to find the shortest path between their nest and a food source by depositing pheromone traces. This mechanism of self-organization and cooperation has been modeled to solve complex optimization problems.

3.3.3 ACO principle applied to the JSSP

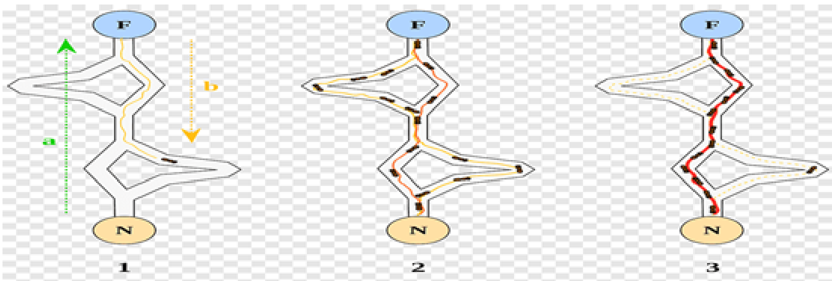

Figure 3.1: Graphical representation of pheromones in the ACO
Source:
https://www.google.com/url?sa=iurl=https%3A%2F%2Fwww.pngegg.com%2Far%2Fpng-naxu, accessed June 7, 2025

The Figure3.1 illustrates the positive feedback mechanism in the ant colony optimization algorithm, where the accumulation and evaporation of pheromones lead to the convergence of ants toward the shortest path between the food source and the nest. Initially, ants explore paths randomly, leaving pheromone trails along the way. Over time, pheromones accumulate on the shorter paths, making them more attractive to other ants, while pheromones evaporate on the longer paths, reducing their attractiveness. This process eventually leads to the convergence of ants toward the shortest path, which represents the optimal solution.

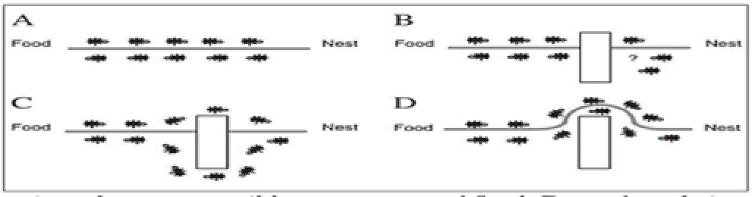

Figure 3.2: Updating pheromone traces
Source: https://ppl-ai-file-upload.s3.amazonaws.com , accessed June 7, 2025

Figure illustrates how ants adjust their pheromone trajectory when faced with an obstacle:

- (HAS) Ants initially follow a path between the nest and the food source.
- (B) An obstacle interrupts this path.
- (C) The ants explore two new alternatives.
- (D) The shortest path becomes dominant due to the reinforcement of pheromones.

In ACO algorithms, this behavior is simulated by artificial agents exploring a graph probabilistically. Each ant constructs a solution (route) recorded in a memory (Tabuk), until reaching a final state.

Two types of information guide decisions:

- I_j (heuristic): fixed, linked to the immediate attractiveness of a movement (e.g. duration, distance).
- τ_{ij} (pheromone): variable, it reflects the accumulated experience, updated at each iteration.

The ACO process consists of three main steps:

1. Construction: each ant develops a solution based on I_j and τ_{ij}.

2. Local improvement (optional): strengthening the solution.

3. Pheromone Update: global evaporation and reinforcement of the best paths.

ACO Procedure: Pseudocode

```
Set parameters, initialize pheromone traces
Scheduled activities:
    Construction of solutions by ants
    Action Server (optional)
    Pheromone Update
```

```
End of scheduled activities
End of procedure
```

The metaheuristic consists of a parameter initialization step and three algorithmic procedures whose activation is regulated by the programmed activities module. These activities are repeated until a stopping condition is reached, such as the maximum number of iterations or a CPU time limit.

The three algorithmic procedures included in the programmed activities are as follows: Ant solution construction is a probabilistic process performed by all ants in the colony, which visit adjacent states of the problem under study. Ants move by applying a stochastic decision policy, using information from pheromone traces as well as heuristic information. This allows them to gradually construct a solution to the problem.

The action server represents centralized actions that modify the algorithm's behavior and cannot be performed individually by ants. The most common action is local optimization or solution improvement, using a local search algorithm. The optimized solutions are then used to adjust the pheromone values to be updated.

Pheromone updating is the process that updates the pheromone trails on each edge τ_{ij}. It is called online or offline a posteriori updating, because it is performed at the end of a path. The amount of pheromone deposited by each ant depends on the total path length (see equation 3). Another approach is to perform an online step-by-step update, i.e., a local or real-time update, performed each time an ant moves from node i to node j. The value of the pheromone trail is then reduced by a constant evaporation rate, which helps avoid premature convergence of the algorithm by gradually eliminating the least traveled paths.

3.3.4 Mathematical modeling

Each virtual "ant" constructs a solution by traversing a graph representing the operations to be scheduled. The probability of choosing an operation depends on the amount of pheromone (collective experience) and heuristic information (for example, the duration or priority of the operation). The alpha (α) and beta (β) parameters control the respective influence of these two factors. The updating of pheromone traces is done globally or locally, allowing the algorithm to avoid premature stagnation.

3.3.5 Search and convergence mechanisms

ACO alternates between exploration (discovery of new solutions) and exploitation (reinforcement of the best solutions found). Pheromone evaporation prevents convergence to local optima and promotes solution diversity. Variants such as Ant

Colony System (ACS), Max-Min Ant System (MMAS) or Elitiste Ant System (EAS) have been proposed to improve search performance and efficiency. .

3.4 Applications of ACO to workshop scheduling

ACO has been successfully applied to various scheduling problems, including JSSP, flow-shop, and project management. Studies show that ACO achieves high-quality solutions even for large instances, thanks to its scalability and consideration of multiple constraints. In practice, integrating ACO into interactive software applications (see Figures 3.1 and 3.2) facilitates data entry, parameter tuning, and results analysis, making the method accessible to industrial engineers and decision-makers.

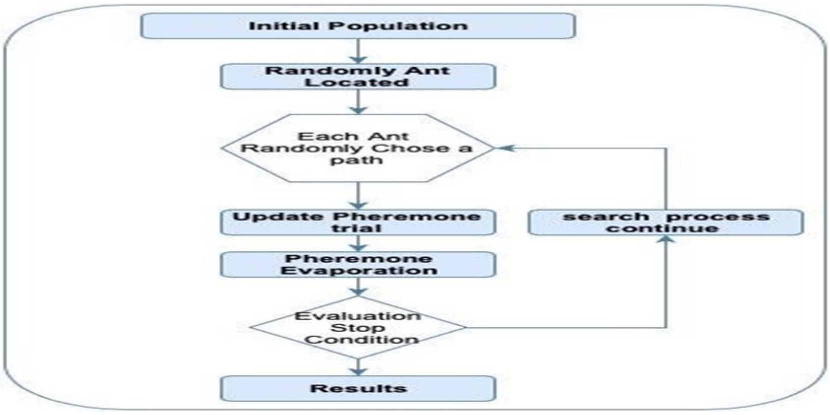

Figure 3.3: The main steps of the ant colony algorithm (ACO)
Source: ResearchGate + Wikipedia, consulted June 8, 2025

Figure diagram3.3 illustrates the main steps of the Ant Colony Algorithm (ACO):

1. Initialization of parameters and pheromones.
2. Construction of solutions by ants according to transition rules.
3. Evaluation of the solutions and update of the best.
4. Global pheromone update (reinforcement and evaporation).
5. Repeat the steps until the stopping criterion is satisfied (number of iterations or convergence).

Algorithm 3.1: Construction of a solution by an ant in the ACO

(Pseudo-code describing the sequential construction of an ordering)

The ACO algorithm for JSSP can be summarized as follows:

1. Initialize the pheromones τ_{ij} to a low and uniform value.
2. Repeat until a stopping criterion is reached:

   - For each ant:
   - Construct a solution (scheduling) by applying the transition rules.
   - Evaluate the quality of the solution (makespan).
   - Update pheromones:
   - Evaporate the pheromones.
   - Place the pheromones on the arcs of the best solutions.

code for global and local pheromone update)

3.5 Graphical representations: disjunctive graph and Gantt chart

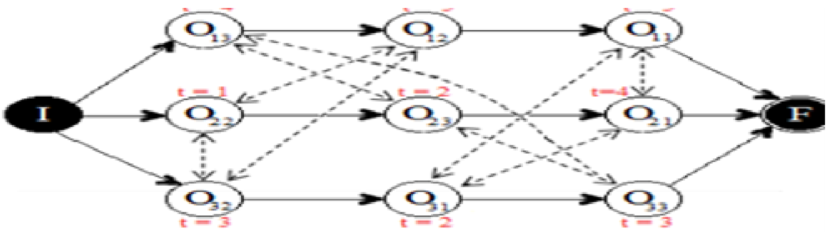

Figure 3.4: Disjunctive graph (Provide a graphical example (3×3 instance))
Roy & Sussmann, 1964

The disjunctive graph (Roy & Sussmann, 1964) illustrates precedents and conflicts on machines. It is written:

$$G = (V, C \cup D)$$

With V: operations, C: oriented arcs (precedences), D: unoriented arcs (conflicts).

The job-shop scheduling problem (JSSP) is usually represented by a discrete graph indicated ($G = (V, C \cup D)$), where V represents a set of nodes corresponding to processes (function and machine). The nodes exclude the start node (I) and the end node (F). The graph contains two types of connections: C, which includes directed edges connecting processes of the same function in a certain sequence, and D, which consists of undirected edges connecting operations performed on the same device. In addition, the processing time of each operation is indicated at the top of the corresponding node.

Figure 2 illustrates an example graph of a 3x3 JSSP instance. Historically, JSSP has been solved by methods such as Branch and Bound (B&B), Algorithm 3.2: Pheromone Trace Update (Pseudo-

which were introduced in 1960. While these methods are effective, they are limited to cases with up to 220 operations (15x14). Therefore, approximate methods are used to handle larger cases, such as simulated annealing (SA), tabu search (TS), iterative local search (ILS), GRAP, ant colony improvement (ACO), evolutionary algorithms (EA) (e.g., artificial immune system and crop algorithm), among others.

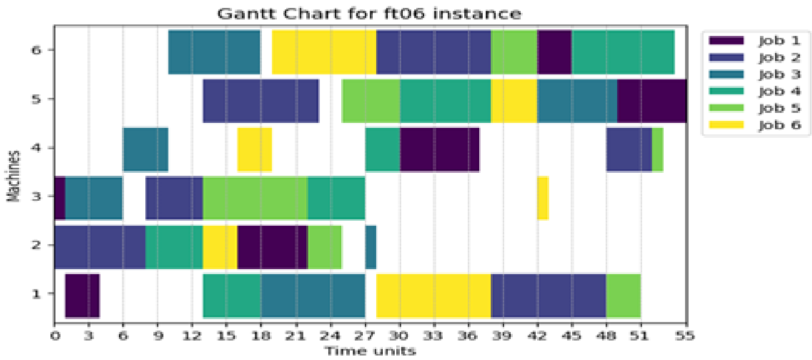

Figure 3.5: Gantt chart

Roy & Sussmann, 1964, accessed June 9, 2025

Figure 3.6: The Makespan Diagram

Roy & Sussmann, 1964, accessed June 10, 2025

Sequence of operations for each task: The diagram shows the order of execution of the operations of each task on the different machines.

Machine Usage: The white spaces between rectangles on the same line indicate the active and inactive hours of each machine.

JSSP Goal: The goal is usually to find a schedule that minimizes the makespan (in this example, 12). This diagram represents a possible solution.

In summary, this diagram visually illustrates how to schedule three jobs, each consisting of multiple operations, on three different machines, in order to execute them all in the shortest possible time (Makespan).

3.6 What does each element of the ant colony represent in the context of the workshop scheduling (JSSP) for the simulation?

ACO Element	Equivalent in the JSSP
Ant	Building a complete schedule (potential solution)
Path	Sequence of operations (possible schedule)
Pheromones	Quality of transition between operations
Node	Individual operation (task on a machine)
Heuristic information	Duration, priority, machine occupancy
Pheromone Update	Strengthening sequences leading to minimal makespan

3.6.1 Conclusion

The ACO represents a promising solution to the JSSP thanks to its intelligent exploration capacity and adaptive behavior. Its iterative and collaborative nature allows it to converge efficiently towards quality solutions, while easily integrating specific constraints. The next step is to experimentally validate its effectiveness in a simulated context.

Chapter 4

Experimental simulation and analysis of results

4.1 Introduction

This chapter presents the practical implementation of the ACO algorithm through an application software developed in Python. After defining the experimental protocol and the cases of Test, we evaluate the performance of the algorithm according to key indicators: makespan, Convergence, robustness and graphical representation.

4.2 Experimental protocol

4.3 Organized the use of the ACO interface for workshop scheduling

The interface shown in the screenshot is a task planning and scheduling application based on the ACO (Ant Colony Optimization) algorithm, with an interface in French. Here is a structured description of the main elements of the task entry interface, as can be deduced from the image:

General operation

- The user begins by selecting a task file using the ”Choose File (.CMY)” button.
- After loading the data, he can start the optimization by clicking on ”Start ACO optimization”.

Table 4.1: Description of the task entry interface

Interface Element	Description
Title bar	Indicates the name of the application: "ACO Scheduling – French Interface".
File selection area	Allows you to choose an input file in .CMY format, probably containing the list of tasks to be scheduled.
Run button	"Start ACO Optimization" button to start the scheduling algorithm.
Gantt chart	Graphical display area (empty here) intended to represent the task execution schedule in the form of a Gantt chart.
Convergence curve (Makespans)	Graphical area (empty here) to visualize the evolution of the solution (makespans) over the iterations of the algorithm.
Validation report	Space reserved for displaying results or validation messages after execution.

- The expected results are then displayed:

- A Gantt chart showing the scheduling of tasks over time.

- A convergence curve illustrating the improvement in makespans (total execution time).

- A validation report summarizing performance or any errors.

Usefulness of the interface

This interface allows the user to:

- Enter and load task data to be scheduled.

- Graphically visualize the solution proposed by the algorithm (Gantt chart).

- Monitor the performance of the algorithm via the convergence curve.

- Obtain feedback on the validity or quality of the proposed scheduling.

This type of interface is typical of scheduling and planning tools, facilitating user interaction and understanding of the results produced by optimization algorithms.

4.4 Presentation of the application

The application offers an intuitive interface (see Figure 3.1) allowing the entry of tasks, the configuration of ACO parameters, the launch of optimization, and the display of numerical and graphical results (Gantt chart, convergence curve).

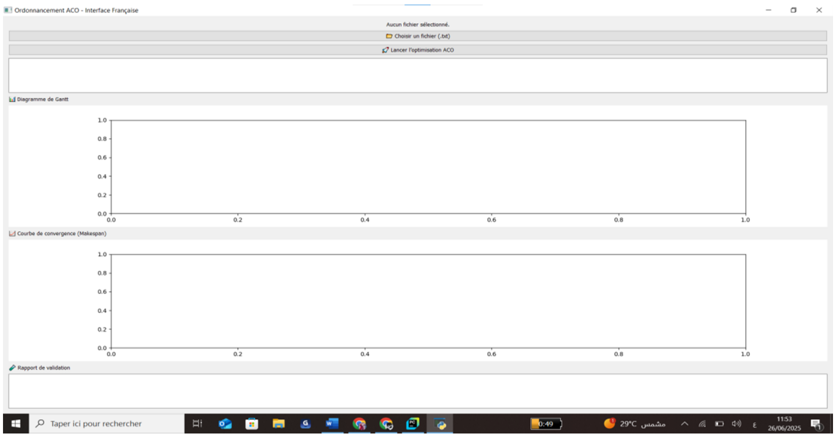

Figure 4.1: Task entry interface in the workshop planning application by ACO

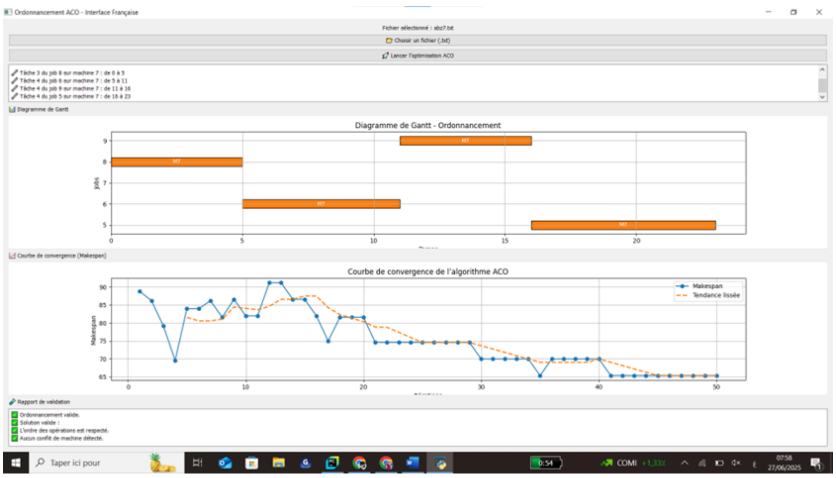

Figure 4.2: Displaying optimization results by ACO instances ft07.txt

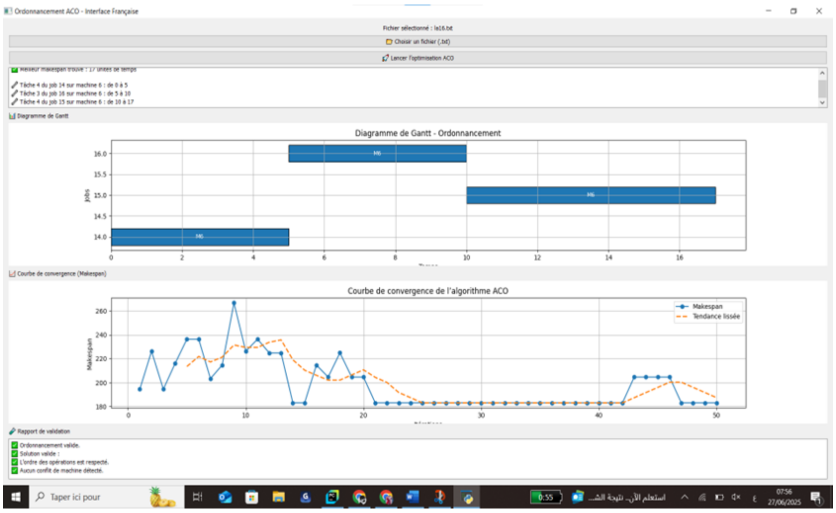

Figure 4.3: Displaying optimization results by ACO instances ft16.txt

4.4.1 Simulation of the ACO algorithm on this case

Step 1: Initialization

- Reading the .txt file containing the data (e.g., ft16.txt)
    - Construction of a directed graph representing the operations
    - Initialization of pheromones and algorithm parameters

Step 2: Ants' Journey (Exploration)

- Each ant starts at the starting node and chooses the next node according to the probability $P = (\text{pheromone}^{\alpha}) \times (1/\text{duration}^{\beta})$

Step 3: Calculating the makespan

- For each path generated by an ant, we calculate the total duration of the scheduling.
    - The Gantt shows a total makespan of 17.

Step 4: Update the pheromones

- Overall evaporation of pheromones.
    - Reinforcement on the best paths (those with shorter makespan).

Step 5: Displaying the results

- Gantt chart display (top).
  - Display of the convergence curve (bottom).

Step 6: Validation of the solution

- Checking the order of operations and machine conflicts.
  - Green boxes in the validation report: no errors detected.

4.5 Analysis and Discussion of Results

The case study corresponds to an example of scheduling 4 tasks (Jobs) with scheduling constraints on several machines. The data file used is "ft16.txt", representing 4 jobs with their successive operations.

The generated Gantt chart shows an efficient distribution of operations across machines. Each task is divided into successive operations that follow one another without conflict or overlap, demonstrating compliance with precedence and resource exclusivity constraints.

The resulting makespan is 17 time units. This result is achieved quickly in the iterations, as shown by the convergence curve. This means that the algorithm parameters (alpha, beta, number of ants) were well chosen.

The final validation confirms that all dependencies are met, each operation is scheduled after the previous one, and no machine conflicts are detected.

However, since the makespan remains fixed from a certain point, it is possible that the algorithm has stabilized on a local optimum. Varying the parameters or increasing the exploration could lead to an even better solution.

In summary, the ACO algorithm produced a valid, consistent, and visually well-structured solution, which provides a solid basis for future extensions or comparisons with other approaches such as tabu search or genetic algorithms.

4.6 Conclusion of the case study

In this study, the Ant Colony Optimization (ACO) algorithm was used to solve a Job Shop Scheduling problem with 4 jobs and multiple machines.

The experimental results confirm that ACO can produce efficient solutions to the JSSP in a reasonable time. The developed interface facilitates interaction with the algorithm and provides clear visualizations. However, the quality of the results strongly depends on the initial parameterization, which paves the way for future research on dynamic adaptation and algorithmic hybridization.

4.7 Difficulties encountered

During this work, several difficulties were encountered:

- Setting ACO parameters: Obtaining optimal results strongly depends on the choice of parameters (number of ants, α, β, evaporation rate). Incorrect tuning can lead to slow convergence or suboptimal solutions.

- Managing complexity for large instances: For large problems (e.g., ft40 or complex industrial cases), the calculation time and visualization of schedules become more demanding.

- Integration of specific industrial constraints: Implementing additional constraints (setup time, maintenance, priorities) requires non-trivial algorithmic adaptations.

- User experience: Ensuring an intuitive interface and readable visualizations, even for very dense schedules, required continuous adjustments.

4.8 Recommendations and perspectives

To strengthen and broaden the impact of this work, several areas for improvement are recommended:

1. Automation of parameter adjustment: Develop an auto-tuning module to dynamically adjust ACO parameters based on the size and characteristics of the problem.

2. Parallelization of the algorithm: Implement a multi-threaded version of ACO to speed up solving on very large instances.

3. Integration of new industrial constraints: Take into account setup times, planned maintenance, or job priorities for an even more realistic application.

4. Connection to industrial systems (ERP/MES): Allow automatic import and export of production data for direct use in real environments.

5. User interface improvements: Add advanced visualization features (zoom, filters, interactive editing of schedules) to facilitate analysis and decision-making.

6. Validation on various industrial cases: Test the application on other real workshops to validate its robustness and adaptability to different contexts.

4.9 Conclusion

Experimental results confirm that ACO can produce efficient solutions to the JSSP in a reasonable time. The developed interface facilitates interaction with the algorithm and provides clear visualizations. However, the quality of the results depends heavily on the initial setting, which opens the way for future research on dynamic adaptation and algorithmic hybridization.

General Conclusion

Through this thesis, we were able to confirm the effectiveness of the ACO algorithm in the Solving the Joint Workshop Scheduling Problem (JSSP). The proposed approach, supported Through an intuitive simulation platform, it has been possible to achieve performances Satisfactory in terms of makespan and convergence time. The development of our own application in Python, integrating a PyQt5 interface, has Allowed clear visualization of results (Gantt chart, convergence curve) and flexibility in the algorithm settings. Despite some limitations related to the size From instances and sensitivity to parameters, the ACO has demonstrated real potential in optimization of task scheduling. For the future, several avenues can be explored: hybridization with other Metaheuristics (PSO, GA), dynamic adaptation of parameters or integration into a Real production environment (MES/ERP). These perspectives will open the way to More robust and adaptive solutions, in response to the ever-increasing demands of Industry 4.0.

References

1. Florez, E., Gómez, W., & Bautista, L. (2017). *Job Shop Scheduling Problem*. Universidad Industrial de Santander, Bucaramanga, Colombia.

2. van der Zwaan, S., & Marques, C. (n.d.). Ant Colony Optimization for Job Shop Scheduling. Instituto de Sistemas e Robotica, Instituto Superior Técnico (IST). http://mscmga.ms.ic.ac.uk/jeb/orlib/jobshopinfo.html

3. Bougea, M. E. (2020). *Customer order scheduling, simulation by PSO* [Master's thesis, Mohamed Boudiaf University of M'sila]. Faculty of Mathematics and Computer Science.

4. Pinedo, M. L. (2016). *Scheduling: Theory, Algorithms, and Systems* (5th ed.). Springer.

5. Dorigo, M., & Stitzel, T. (2004). *Ant Colony Optimization*. The MIT Press, Cambridge, Massachusetts, USA.

6. Dorigo, M., Maniiez, V., & Colori, A. (1996). "The Ant System: Optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, and Cybernetics – Part B*.

7. Blum, C. (2005). "Ant colony optimization: Introduction and recent trends." *Physics of Life*.

Python Libraries Used

Table 4.2: Python libraries and their roles

Library	Role
NumPy	Numerical calculations and matrix manipulation
Random	Random generation for operation choices
Matplotlib	Drawing graphs and Gantt charts
PyQt5	Creating the graphical user interface
ReportLab	Generating PDF files
JSON	Reading and interpreting job files
Time	Measurement of execution times
BONE	File management and export paths

ملخص

تتناول هذه المذكرة تقييم أداء نظام جدولة الخوادم.

تم نمذجة المشكلة في بيئة من نوع "ورشة الإنجاز(Job Shop) "، وهي بيئة تتميز بوجود عمليات متسلسلة يجب تنفيذها على آلات محددة.

ولحل هذه المشكلة المعقدة، تم اقتراح منهجية تعتمد على خوارزمية "تحسين مستعمرة النمل.(Ant Colony Optimization - ACO) "

هذه الألية التقديرية، التي يُعترف بكفاءتها في سيناريوهات الجدولة المعقدة، تتيح استكشافاً ذكياً لفضاء الحلول.

كما تم تطوير أداة برمجية تفاعلية باستخدام لغة Python وواجهةPyQt5 ، بهدف محاكاة وتحسين دورة إدارة الطلبات بالكامل.

ومن خلال تحليل النتائج باستخدام مؤشرات الأداء الرئيسية مثل "مدة إنجاز كل المهام" (makespan) ومخططات غانت (Gantt charts)، تبين أن هذه الطريقة تُحدث تحسناً ملحوظاً في كفاءة النظام واستجابته والأداء العام لعملية الجدولة.

Abstract

This master’s thesis focuses on the performance evaluation of server scheduling system.

The problem is modeled within a Job Shop environment, characterized by sequential operations that must be executed on specific machines.

To address this complex problem, an approach based on Ant Colony Optimization (ACO) is proposed.

This metaheuristic, recognized for its effectiveness in complex scheduling scenarios, enables intelligent exploration of the solution space.

An interactive software tool, developed using Python and PyQt5, is designed to simulate and optimize the entire order management cycle.

The results, analyzed through key performance indicators such as makespan and Gantt charts, demonstrate that this method significantly enhances system efficiency, responsiveness, and overall scheduling performance.