

## A Construction Project Scheduling Approach for Petroleum Refinery Turnaround Maintenance Using Hybrid GA–LOB Optimization

Safinaz M. ElDawody <sup>1\*</sup>, Yasmeen A. S. Essawy <sup>1,2</sup>, Khaled Nassar <sup>1</sup>

<sup>1</sup> Department of Construction Engineering, The American University in Cairo (AUC), Cairo 11835, Egypt.

<sup>2</sup> Department of Structural Engineering, Ain Shams University (ASU), Cairo 11566, Egypt.

Received 15 March 2026; Revised 24 May 2026; Accepted 27 May 2026; Published 01 June 2026

### Abstract

This study develops a hybrid Genetic Algorithm–Line of Balance (GA–LOB) scheduling framework for turnaround maintenance (TAM) in petroleum refineries, addressing the NP-hard challenge of optimizing task sequencing under realistic contractor availability and mandatory work/rest cycles. The TAM problem is formulated as an unrelated parallel machine scheduling problem with sequence-dependent setup times and SIMOP (simultaneous operations) safety constraints. A three-component chromosome encodes unit processing sequence, contractor allocation, and execution mode, enabling the GA to explore a large combinatorial solution space. At the same time, the LOB scheduler enforces crew continuity and the feasibility of work/rest. An exhaustive enumeration of contractor counts establishes the theoretical performance ceiling, and statistical validation using ANOVA and independent t-tests assesses the significance of optimization gains. The unconstrained exhaustive search identifies a global optimum of 5.93 weeks with a contractor allocation of (8, 10, 8, 10). With realistic resource constraints of up to three contractors per task, the GA achieves a project duration of 25.50 weeks, a statistically significant 10% improvement over the 28.32-week single-contractor baseline (ANOVA:  $F = 6.94$ ,  $p = 0.009$ ). Task-level t-tests reveal no significant change in individual task durations (all  $p > 0.05$ ), demonstrating that efficiency gains arise exclusively from optimized concurrency and sequencing rather than task compression. This is the first study to apply a GA–LOB hybrid framework to petroleum TAM, formally integrating work/rest cycle constraints and SIMOP safety logic within the optimization chromosome. The resulting framework provides a scalable, data-driven diagnostic tool for industrial asset management with direct applicability to multi-site refinery operations.

**Keywords:** Genetic Algorithm; Scheduling; Optimization; Turnaround Maintenance; Unrelated Parallel Machine.

## 1. Introduction

### 1.1. Industrial Context and Motivation

Petroleum refineries operate under extreme thermomechanical stress, necessitating periodic partial shutdowns, commonly referred to as turnaround maintenance (TAM), for comprehensive inspection, preventive maintenance, and capital upgrades. These shutdowns represent among the most capital-intensive and operationally critical events in the energy sector, with extended durations translating directly into forgone production revenue that can reach tens of millions of dollars per day [1, 2]. The concurrent execution of inspection, internal maintenance, external maintenance, and construction activities within a single, time-bounded shutdown window creates a scheduling problem of substantial combinatorial complexity. This complexity is further compounded by mandatory regulatory rest periods for specialized

\* Corresponding author: [safi\\_md@aucegypt.edu](mailto:safi_md@aucegypt.edu)

 <https://doi.org/10.28991/CEJ-2026-012-06-013>



© 2026 by the authors. Licensee C.E.J, Tehran, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).

contractors, Simultaneous Operations (SIMOP) safety constraints, and the interdependence of tasks across 22 or more process units operating simultaneously in a live hazardous environment [3-5].

In the refinery's periodic maintenance strategy, three primary maintenance types, preventive, predictive, and failure repair, are essential for optimal operation [1]. Preventive maintenance aims to reduce the likelihood of equipment failure through scheduled inspections and servicing. Predictive maintenance uses real-time data and condition monitoring to forecast equipment needs. Failure repair focuses on promptly addressing unplanned outages. During shutdowns, when time constraints are stringent, these maintenance activities must be accurately planned and coordinated [4, 5]. Advanced scheduling algorithms and resource management tools can enhance the efficiency of this process, ensuring that all tasks are completed within a limited downtime while minimizing potential disruptions to production [6, 7].

Despite significant advances in scheduling optimization for manufacturing and construction contexts, the TAM scheduling literature reveals four critical gaps. First, existing TAM models universally treat makespan as the sole objective, optimizing duration without accounting for the physical site capacity limits imposed by SIMOP regulations [1, 2, 5]. Second, work/rest cycle constraints, a regulatory reality for all specialized contractor teams, have not been formally integrated as optimization variables in petroleum TAM models. Third, no study has applied a Genetic Algorithm–Line of Balance (GA–LOB) hybrid framework to petroleum TAM, despite the proven effectiveness of LOB principles in managing resource continuity across repetitive units [7-9]. Fourth, no published TAM study has employed two-tier statistical validation to distinguish project-level from task-level optimization effects, leaving a methodological gap in the evaluation of scheduling improvements [10].

The unrelated parallel machine scheduling problem, which accurately captures the heterogeneous task structure of TAM, has been extensively studied in manufacturing [10-13] but rarely applied to industrial maintenance in petroleum facilities. Furthermore, recent studies on repetitive construction scheduling with crew routing and multi-resource constraints [8, 9, 13] demonstrate the value of simulation-based and agent-based approaches, which inform the modeling philosophy adopted in this work. The integration of these approaches with evolutionary optimization remains underexplored in the petroleum TAM context.

## 1.2. Research Objectives and Contributions

This study addresses the identified gaps through four primary contributions. First, it formulates the petroleum TAM as an unrelated parallel machine scheduling problem with sequence-dependent setup times and mandatory work/rest cycles, providing a more operationally realistic model than prior integer-programming approaches. Second, it develops a hybrid GA–LOB framework with a novel three-component chromosome that simultaneously optimizes unit processing sequences, contractor allocations, and execution modes. Third, it derives a theoretical performance ceiling through exhaustive enumeration of contractor counts, enabling rigorous benchmarking of heuristic solutions. Fourth, it validates schedule improvements through two-tier statistical analysis (ANOVA and independent t-tests) that distinguishes project-level efficiency gains from task-level duration changes, directly addressing the methodological gap identified in the literature.

## 1.3. Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews the relevant scheduling literature and positions this study's contributions. Section 3 presents the mathematical model and GA–LOB methodology. Section 4 describes the petroleum refinery case study. Section 5 reports results across three progressive scenarios. Section 6 provides a discussion and comparison with prior work. Section 7 presents conclusions and recommendations.

# 2. Literature Review

## 2.1. Turnaround Maintenance Scheduling

Turnaround maintenance has been characterized as a complex project management challenge combining time-constrained resource allocation with safety-critical simultaneity constraints [4]. Alfares [1] applies integer linear programming (ILP) to minimize TAM duration in petroleum refineries, demonstrating that systematic sequencing yields measurable reductions in shutdown duration. Alfares [2] establishes workforce team assignment and job scheduling as foundational sub-problems, showing that task completion time depends critically on team size and contractor availability windows. Bijvank [3] documents the practical challenges of coordinating multi-skilled labor across concurrent unit shutdowns. Ertem et al. [5] developed a workers-constrained shutdown model with skills flexibility, bounding worker numbers to reduce costs. Al-Turki et al. [4] identify the integration of realistic resource constraints and SIMOP logic as the primary open research frontier. None of these works integrates mandatory work/rest cycles as optimization variables.

Recent studies on repetitive construction scheduling provide methodological inspiration for the LOB component of this work. Hegazy et al. [8] develop a simulation-based model for crew routing with learning and forgetting effects in repetitive projects, demonstrating significant schedule compression through optimized crew continuity. Sultan et al. [9]

introduce an agent-based simulation framework for multi-resource-constrained scheduling of scattered, atypical, repetitive projects and validate the approach on 22-unit construction sequences comparable in scale to the present case study. These recent advances (2025–2026) confirm that simulation and hybrid optimization methods are productive frontiers for resource-constrained repetitive scheduling.

## 2.2. Genetic Algorithms for Scheduling

Genetic algorithms have demonstrated consistent superiority over exact methods for complex parallel machine scheduling problems [11, 12, 13]. Zhang et al. [6] addresses unrelated parallel machine scheduling with worker resource constraints and learning effects, combining evolutionary algorithms with realistic labor models. For exact methods, Rocha et al. [7] establish that Branch-and-Bound algorithms for unrelated parallel machine scheduling become computationally intractable as problem size grows, with solution times increasing exponentially beyond 10–15 jobs. Sheremetov et al. [10] validate a two-stage GA for oil and gas production scheduling across parallel machines, demonstrating domain-specific applicability. Vallada & Ruiz [14] apply GA to unrelated parallel-machine problems with sequence-dependent setup times, thereby establishing the methodological foundation for the present study. Bektur & Saraç [15] confirm this scalability limitation in problems with sequence-dependent setup times, motivating the use of metaheuristic approaches for the 22-unit, 88-task problem studied here. Afzalirad & Rezaeian [16] demonstrate that resource-constrained variants of the unrelated parallel machine problem require heuristic treatment due to the compounded complexity of precedence constraints and eligibility restrictions.

## 2.3. Line of Balance in Construction and Maintenance

The Line of Balance (LOB) method [8, 13] optimizes LOB schedules for scattered repetitive projects using a GA, establishing the hybrid LOB–GA architecture. Haq et al. [17] initially created this method for repetitive construction tasks and later integrated it with metaheuristics to ensure crew continuity across different production units. Similarly, many research publications [18–23] have reviewed automated construction scheduling methods, highlighting that LOB’s advantages include maintaining an uninterrupted workflow and reducing crew idle time. properties directly applicable to the unit-by-unit sequencing structure of petroleum TAM. Furthermore, additional references include [24, 25] concerning Knowledge-Based Resource Assignment in Prefabricated Construction Utilizing Sequential Pattern Mining.

## 2.4. Research Gap Summary and Positioning

Table 1 summarizes the key characteristics of directly related studies. The comparative analysis confirms that no prior study simultaneously addresses: (i) work/rest cycle constraints as optimization variables; (ii) SIMOP safety constraints within the optimization objective; (iii) GA–LOB hybrid architecture for petroleum TAM; and (iv) two-tier statistical validation of schedule improvements. This study addresses all four gaps.

**Table 1. Comparative summary of related TAM and parallel machine scheduling studies**

Author(s)	Method	Problem	Industry	Work/Rest Cycle	Contractor Limits Modeled	LOB	Statistical Valid.
Alfares [1]	ILP	TAM workforce	Petroleum	No	No	No	No
Alfares [2]	ILP / heuristic	TAM scheduling	Petroleum	No	Partial	No	No
Ertem et al. [5]	MIP + heuristic	Shutdown maintenance	Cement	No	Yes	No	No
Hegazy & Kamarah [13]	GA–LOB	Repetitive construction	Construction	Yes	No	Yes	No
Hegazy et al. [8]	LOB-Simulation	Repetitive construction.	Construction	Yes	No	Yes	No
Sultan et al. [9]	Agent-based sim.	Multi-res. sched.	Construction	Yes	Yes	No	No
Zhang et al. [6]	Combinatorial EA	Unrelated PM	General	No	Yes	No	No
Vallada & Ruiz [14]	GA	Unrelated PM	General	No	No	No	No
Bektur & Saraç [15]	MIP + heuristic	Unrelated PM	General	No	No	No	No
Sheremetov et al. [10]	Two-stage GA	Parallel machine	Oil & Gas	No	No	No	No
Afzalirad & Rezaeian [16]	Resource-constr. GA	Unrelated PM	General	No	Yes	No	No
<b>This Study</b>	<b>GA–LOB (single-obj.)</b>	<b>TAM unrelated PM</b>	<b>Petroleum</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>

## 3. Research Methodology

### 3.1. Problem Formulation

In a turnaround maintenance project, a structured decision-making workflow is essential for evaluating the required activities across all refinery units. Each unit comprises multiple processes and equipment items that must be individually assessed to determine maintenance needs and potential upgrade requirements. Figure 1 presents the decision flowchart

guiding this evaluation. It establishes a systematic pathway for sequencing tasks and coordinating maintenance activities across units. The flowchart integrates critical operational factors, accounting for unit conditions and upgrading requirements while managing task interdependencies and resource constraints to ensure feasible execution. By strictly adhering to industrial safety standards and incorporating task-specific setup times, this structured decision pathway enhances prioritization and sequencing, minimizing operational downtime and mitigating potential delays.

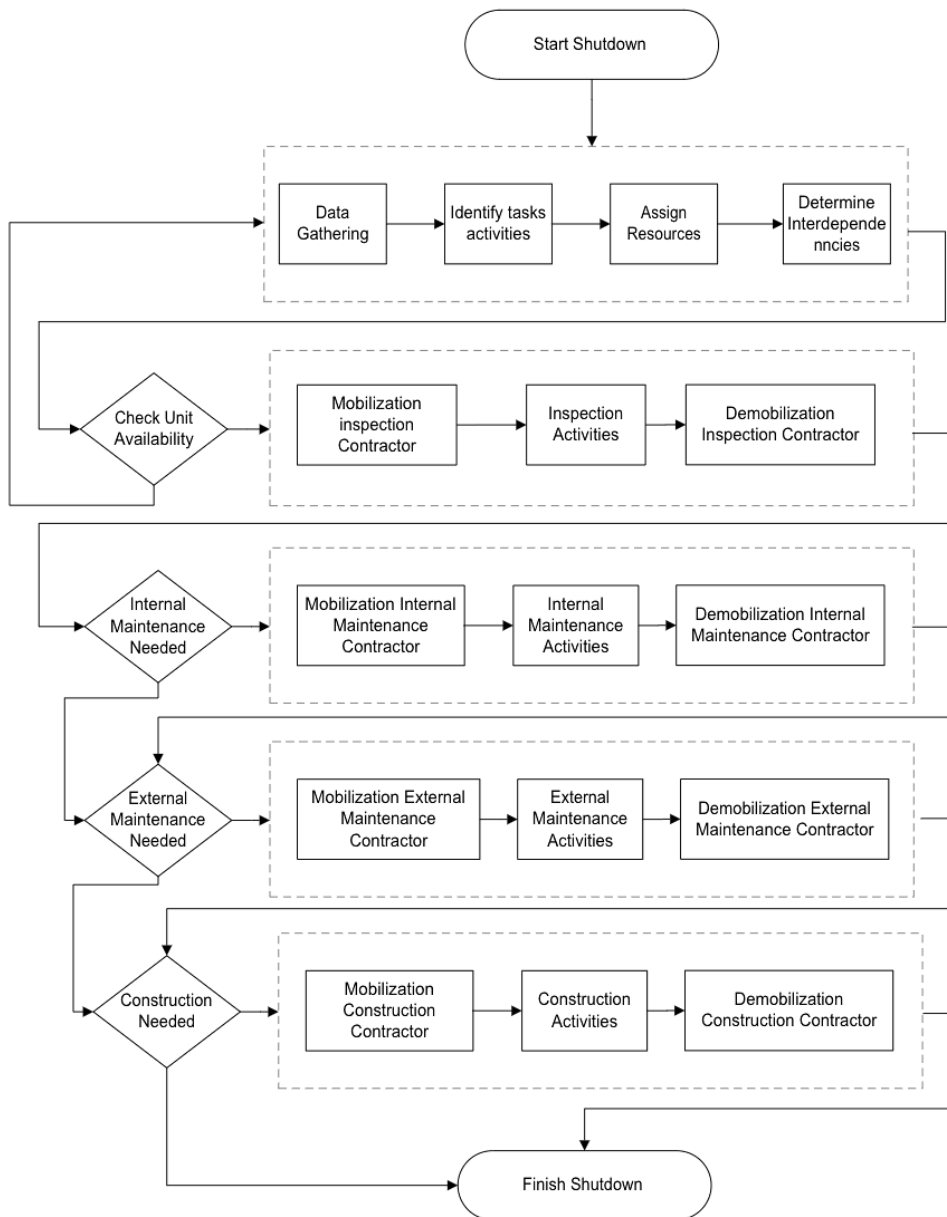


Figure 1. Decision Flowchart for Turnaround Maintenance and Construction Activities

### 3.2. Adopted Optimization Algorithm

The refinery upgrade scheduling problem is modeled as a Linear Integer Programming (LIP) problem, where decision variables such as unit sequencing and contractor assignments are discrete. The combinatorial complexity of the search space makes exact optimization methods impractical for the 22-unit, 88-task problem studied here. A Genetic Algorithm (GA) was therefore selected as the main optimization approach and implemented using the DEAP (Distributed Evolutionary Algorithms in Python) framework [10, 11, 15]. GA iteratively improves the population of candidate schedules to find near-optimal solutions that satisfy multiple, often conflicting, operational constraints.

The choice of GA over exact methods is motivated by three factors. First, exact methods such as Branch and Bound become computationally intractable for unrelated parallel machine scheduling as the number of jobs and sequence-dependency constraints increases [12, 13]. Second, local search methods such as Tabu Search or Simulated Annealing, while effective for unimodal landscapes, tend to converge prematurely toward local optima when constraints are tightly coupled [12, 15]. Third, GA’s population-based approach enables broad exploration of the combinatorial space through

crossover and mutation, without requiring the extensive training data demanded by machine learning alternatives. Beyond sequencing and contractor assignment, the present formulation explicitly incorporates resource constraints, including contractor availability, shift coverage, and SIMOP-compliant simultaneous operation limits, which have been shown to significantly affect schedule feasibility and duration in industrial maintenance contexts [5, 8, 9].

### 3.3. Integration of Line of Balance Principles

To ensure operational feasibility, the model integrates essential principles of the Line of Balance (LOB) method. The schedule optimization function enforces LOB logic by managing task precedence, contractor availability, and unit progression over time, while maintaining fixed work/rest cycles and crew work continuity [7, 8]. This approach ensures synchronized resource use and a continuous workflow across the 22 process units. Within this framework, the GA optimizes the critical input parameters that influence the LOB scheduler, specifically the sequence in which contractors are assigned to each task. This integration results in a hybrid optimization method that combines heuristic search with structured production-line scheduling, a novel architecture for petroleum TAM.

### 3.4. Chromosome Design

The key methodological contribution of this study is a GA chromosome that explicitly encodes three decision variables essential to LOB optimization: (i) a permutation sequence string determining the order in which the 22 units are processed; (ii) a contractor allocation vector assigning the number of contractor teams to each of the four tasks; and (iii) an execution mode vector specifying the operational method for each activity, where each mode represents a unique trade-off between duration and resource consumption. This three-component structure provides a direct interface between the GA and the LOB scheduler, enabling the algorithm to simultaneously explore the full combinatorial space of unit sequences and resource allocations. Outcomes that would be difficult to achieve through manual experimentation or single-variable optimization are therefore within reach of the hybrid framework.

## 4. Mathematical Model Formulation

### 4.1. Decision Variables and Parameters

The following notation defines the mathematical foundation of the model:

- $C_{j,k}$  is the completion time of task  $j$  in unit  $k$ .
- $S_{j,k}$  is the start time of task  $j$  in unit  $k$ .
- $p_{j,k}$  is the processing duration for task  $j$  in unit  $k$ .
- $\sigma_{ij}$  is the setup time required if task  $j$  follows task  $i$  in the same unit.
- $r_j$  Release date (earliest possible start) for task  $j$ .
- $x_{ij,k}$  Binary variable: 1 if task  $j$  starts immediately after task  $i$  in unit  $k$ ; 0 otherwise.
- $D_k$  Specified project deadline or shutdown window for Unit  $K$ .
- $U_k$  is the unit's entry window,
- $R_m$  is the contractor's availability,
- $A_k$  is the unit's internal availability

### 4.2. Objective Function

The primary objective is to minimize the maintenance project's makespan  $Z$  while balancing the safety-productivity trade-off. This is expressed as the minimization of the maximum completion time across all units:

$$\min Z = \max(C_{j,k}) + \lambda \cdot R \text{ (SIMOP)} \quad (1)$$

where,  $\lambda$  is a penalty coefficient representing the trade-off between scheduling density (productivity) and the quantified risk factor  $R$  associated with concurrent operations.

#### 4.2.1. Primary Task Precedence

A critical safety constraint in petroleum facilities is the requirement that the primary inspection, or "cold work" (Task 1), must be completed before any subsequent maintenance tasks commence within the same unit. This ensures that the environment is cleared up for concurrent work:

$$S_{j,k} \geq C_{1,k} + \sigma_{1j} \quad \forall j \in \{2, \dots, n\} \quad (2)$$

This ensures that, for any unit  $k$ , all  $n-1$  subsequent tasks are initiated only after Task 1 completes and the requisite setup time  $\sigma$  is complete.

#### 4.2.2. Sequential Task Execution and Non-Preemption

To ensure tasks are executed within the limited time available during a partial shutdown, we apply sequencing constraints. If task  $j$  follows task  $i$ , the start time of  $j$  must account for the completion of  $i$  and any setup overhead:

$$S_{j,k} \geq S_{i,k} + p_{i,k} + \sigma_{ij} - M(1-x_{ij,k}) \quad (3)$$

where,  $M$  is a sufficiently large positive constant. Because the model assumes no preemption, the relationship between start and completion times is clearly defined as:

$$C_{j,k} = S_{j,k} + p_{j,k} \quad (4)$$

#### 4.2.3. Unit and Resource Availability

The model currently tracks unit availability using basic SIMOP logic in the schedule-task-optimized function. This is mathematically translated as:

$$S_{j,k} \geq \max (U_k, R_m, A_k) \quad (5)$$

#### 4.2.4. Project Duration Threshold

To ensure the project remains within operational limits, each activity must be completed before the terminal deadline of the shutdown window:

$$C_{j,k} \leq D_k \quad \forall_{j,k} \quad (6)$$

### 4.3. Computational Implementation

#### 4.3.1. Initialization

The optimization process begins by generating an initial population of 50 candidate schedules ( $P = 50$ ). Each individual comprises three components: (i) a permutation of 22-unit identifiers representing the processing order; (ii) contractor allocations randomly sampled as integers from  $\{1, 2, \dots, 10\}$  for each task; and (iii) work shift counts randomly sampled as integers from  $\{1, 2, 3\}$  per task. A fixed random seed (42) is used across all runs to ensure full reproducibility of the stochastic GA operators.

#### 4.3.2. Fitness Evaluation

Each individual is evaluated by simulating the project schedule using their encoded unit sequence, contractor counts, and shift counts. The schedule evaluation function iterates over units in the individual's sequence, then over the four tasks per unit, then over individual activities. For each task, the start time is set to the maximum of the previous task's completion time on that unit and the contractor's next availability time, incorporating work/rest cycle dynamics directly. The fitness value is the resulting project's makespan ( $\max(C)$ ), with lower values indicating higher fitness.

#### 4.3.3. Selection, Crossover, and Mutation

Selection uses tournament selection with a tournament size of 3: three individuals are randomly selected, and the fittest is selected as a parent. The Partially Mapped Crossover (PMX) operator exchanges unit sequence segments between parents while ensuring permutation validity; no unit appears twice in the offspring's sequence. Contractor and shift allocations are also recombined during crossover. Mutation applies two independent operators: a swap mutation on unit order (probability 0.05 per element) and a random resampling of contractor or shift counts (probability 0.10 per count). The overall mutation probability is 0.80, ensuring substantial population diversity throughout the 100-generation run.

#### 4.3.4. Convergence

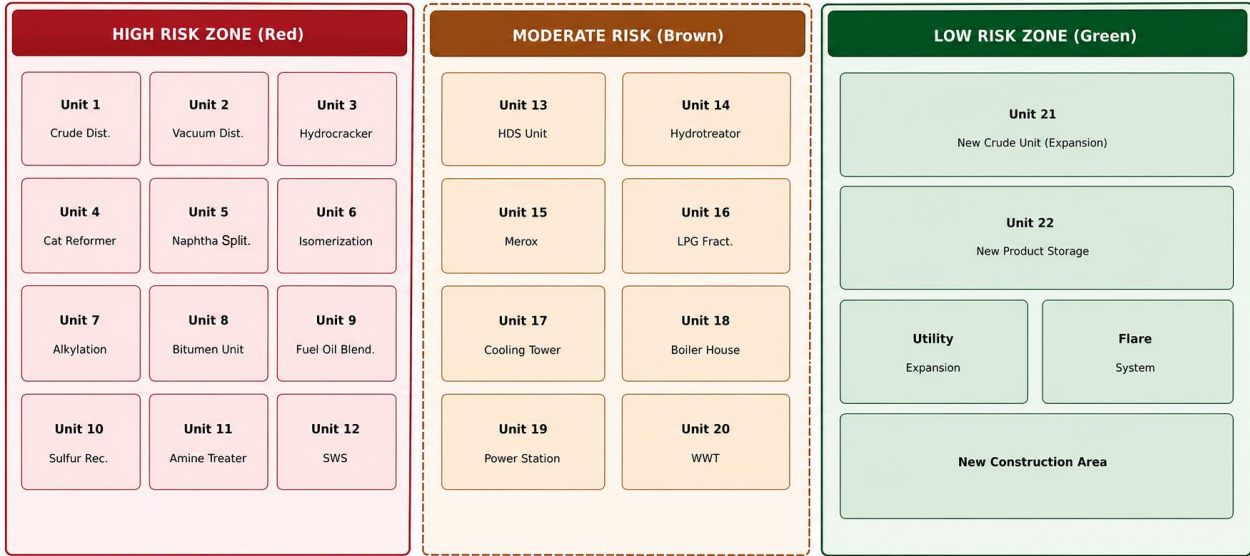
The algorithm runs for 100 generations, evaluating up to  $50 \times 101 = 5,050$  candidate schedules. Upon termination, the individual with the highest fitness across all generations is selected as the optimized schedule. This best individual encodes the final unit sequence, contractor allocations, and shift assignments that yield the minimum project duration satisfying all SIMOP and task-precedence constraints.

## 5. Case Study: Petroleum Refinery Upgrade

### 5.1. Project Description

The case study involves a major capacity expansion and product quality upgrades project at an operational petroleum refinery. Figure 2 shows the schematic layout of the refinery, illustrating process units, utility systems, and designated safety zones. The upgrade project aims to increase throughput and comply with European fuel-quality standards. An international EPC contractor was engaged under a lump-sum contract. New refining units are installed in vacant areas, while significant modifications are made to existing operational units within planned annual shutdown windows.

Schematic: representation of 22 process units, utility systems, and risk classification zones (Red=High, Brown=Moderate, Green=Low)



Note: Figure shows a schematic representation. Replace with actual refinery layout photo if available.

Figure 2. Schematic chart of the petroleum refinery

Construction activities follow a three-tier SIMOP risk classification. Green-zone activities (new units in isolated areas) carry low risk and proceed during normal operations. Brown-zone activities (adjacent to operational units) require heightened coordination and are constrained to shutdown windows. Red-zone activities (direct interaction with operational units) require the most stringent SIMOP controls, including mandatory cold-work clearances and strict worker density limits.

### 5.2. Task Structure

The shutdown schedule involves the sequential and concurrent execution of four task types across 22 process units (Figure 3): Task 1 (Inspection) comprises eight individual activities averaging 0.025 weeks each; Task 2 (Internal Maintenance) comprises nine activities with durations of 0.4–1.0 weeks; Task 3 (External Maintenance) comprises nine activities of similar duration; Task 4 (Construction) comprises eight activities of 0.8–1.7 weeks. Task 1 must be completed before Tasks 2 and 3 can begin within the same unit. Task 4 may begin concurrently with Task 1, but must be coordinated to prevent SIMOP violations. All tasks are flexible regarding which unit is processed first.

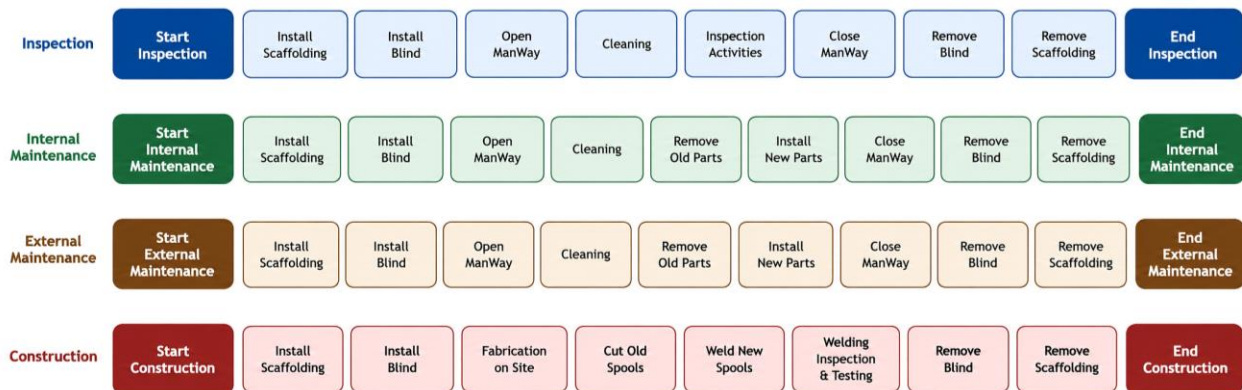


Figure 3. Task Activities flowchart

### 5.3. Model Parameters

Task durations are calculated as the sum of individual activity durations, which are randomly sampled from a triangular distribution with the planned duration as the mode. Setup times, representing mobilization and preparation, are sampled from a uniform distribution  $U [1, 2]$  shifts. The work/rest cycle is fixed at 0.7 weeks of work followed by 0.7 weeks of mandatory rest, in line with contractual requirements for specialized refinery maintenance personnel. The resource constraint for the GA optimization scenario limits contractor allocation to a maximum of three teams per task, reflecting realistic site workforce capacity.

### 5.4. Objective Function

This case study aims to determine the most effective schedule for completing a multi-task project for 22 units. Using Genetic Algorithms (GA), different combinations of contractors and work shifts for four tasks are examined, along with the optimal sequence of activities. The number of skilled workers, the number of shifts, and task dependencies are the main variables considered in the model to ensure resources are used wisely and the plan works in practice. Statistical tests (such as ANOVA or t-tests) are used to validate whether the schedule improvements are significant.

#### 5.4.1. Fitness Functions

The objective of optimization is to minimize the total project duration (makespan  $Z$ ). The following objective function represents this:

$$K_{min} = \left( \max_{u \in k} C_u \right) + (\lambda \cdot R) \quad (7)$$

where,  $K$  is the set of all units;  $C_u$  is the completion time of unit  $u$ ;  $k$  is the total number of units;  $\lambda$  is the penalty coefficient for the safety-productivity trade-off;  $R$  is the quantified risk factor of concurrent operations.

The Fitness Function evaluates each candidate's schedule based on this objective while penalizing solutions that violate operational constraints. It measures how effectively a sequence minimizes idle time and optimizes contractor utilization while strictly adhering to the work/rest cycles and unit-specific capacity limits.

#### 5.4.2. Sequential Task Execution Within a Unit

The schedule evaluation function systematically iterates over the units in the sequence determined by the individual (the unit permutation). For each unit, it subsequently iterates through the associated tasks (Task 1, Task 2, Task 3, Task 4). Within each task, it continues by iterating through the respective activities.

#### 5.4.3. Start Time Calculation

A task's start time on a specific unit is set by the maximum of two factors: the completion time of the previous task on that unit, and the contractor's availability, which must be confirmed before the task can begin. Further detail how the evaluation function incorporates resource constraints such as contractor availability and work/rest cycles to demonstrate its comprehensive approach to realistic scheduling.

#### 5.4.4. Updating Unit End Time

After all activities for a unit are scheduled, the unit's end time is set to the last activity's end time. This ensures that the next task in the same unit waits until the current task is completed. Therefore, even though the genetic algorithm is optimizing the order of units and the number of contractors, the inherent structure of the schedule evaluation function respects the dependencies that Task 2 must follow Task 1 on the same unit, Task 3 must follow Task 2, and Task 4 must follow Task 3. Explicitly describe how the algorithm enforces these task dependencies within each unit to clarify the sequential logic and its impact on schedule feasibility and optimization.

As the algorithm progresses, schedules are selected based on their fitness. Through crossover, schedules may exchange unit task sequences or contractor assignments. Mutation operations generate alternative task setups or minor resource adjustments to maintain population diversity and promote exploratory search behavior. For example, while Task 4 (Construction) depends on completing Task 3 within a unit, it can start independently across different units once resources are available. This flexibility allows the GA to improve parallel execution and shorten the overall project duration. This evolutionary approach continues until the algorithm reaches an optimal scheduling solution or until the defined termination criteria are met, namely, the maximum number of iterations.

## 5.5. Algorithm Pseudo-Steps Explanation

### 5.5.1. Initialization

The optimization process begins by creating an initial population of potential solutions. Each 'individual' in this population represents a possible project schedule and consists of three main components:

**Unit Sequence:** A permutation of 22-unit identifiers (from 1 to 22), representing the order in which these units are to be processed. This is generated from a random sample that shuffles the unit IDs.

**Contractor Allocation:** The number of contractors allocated to each of the 4 tasks. In the optimization cells, this is randomly sampled as an integer between 1 and 10 for each task.

**Work Shifts:** The number of work shifts allocated to each of the 4 tasks daily. This is sampled as an integer between 1 and 3 for each task.

The initial population, consisting of 50 individuals (population size = 50), is randomly generated. Each component of an individual (unit sequence, contractor counts, shift counts) is initialized independently based on the defined ranges and permutations.

### 5.5.2. Evaluation

Each of the 50 individuals in the population undergoes a fitness evaluation to quantify the quality of the schedule it proposes. The fitness function's objective is to minimize the total project duration. This duration is determined by simulating the project schedule using the unit, contractor, and shift counts defined by the individual. The schedule tasks optimized function determines the completion time for each unit based on unit end times (C), and the fitness value is then defined as the maximum of these completion times ( $\max(C)$ ), representing the overall project duration. A lower duration signifies better fitness.

### 5.5.3. Selection

To reproduce high-quality solutions, individuals from the current population are selected as 'parents' for the next generation. The selection process favors individuals with shorter project durations (higher fitness). In this implementation, a selection strategy with a size = 3 is used. This means that for each parent needed, 3 individuals are randomly selected from the population, and the one with the best fitness (shortest project duration) among them is selected as a parent.

### 5.5.4. Crossover

The crossover operator combines genetic material from two selected parents to create new offspring. The crossover operator is used, which is particularly effective for permutation-based representations like the unit sequences. When two parent unit sequences undergo crossover, they exchange segments to produce new, valid unit sequences. The contractor and shift allocations are also recombined during this process. The 50% parameter in the advanced optimization cell determines the probability that two selected parents will engage in crossover to produce offspring. This mechanism ensures that beneficial scheduling patterns can be inherited and combined, while maintaining structural feasibility.

### 5.5.5. Mutation

To prevent premature convergence and explore new areas of the solution space, offspring undergo mutation. This introduces random changes to an individual's genetic composition:

**Unit Order Mutation:** The operator shuffles a portion of the unit order. The parameter = 0.05 means that, on average, 5% of the elements in the unit sequence will be swapped with another element.

**Contractor/Shift Count Mutation:** The parameter = 0.1 indicates a 10% probability that each contractor or shift count is randomly changed within its allowable range (e.g., 1 to 10 for contractors, 1 to 3 for shifts). This ensures variability in resource allocation.

The mutation parameter (0.8, or 80%) in the optimization determines the probability that an individual undergoes mutation after crossover.

### 5.5.6. Replacement and Iteration

After evaluation, selection, crossover, and mutation, the newly generated offspring replace the current population. This completes one 'generation' of the evolutionary cycle. This entire cycle is repeated for a specified

number of generations, for example, 100 generations. With a population size of 50 and 100 generations, the algorithm will evaluate  $50 * (1 + 100)$  schedules, where the 1 is for the initial population evaluation. In practice, the number of evaluations per generation may vary slightly depending on whether individuals are re-evaluated if they have not changed.

### 5.5.7. Final Output

Throughout the evolutionary process, the model tracks the single best-performing individual identified across all generations. This individual, which represents the most optimized solution, is stored together with its corresponding fitness value, defined as the shortest project duration. After all 100 generations are completed, the best individual contains the optimized unit sequence, contractor allocations for each task, and shift assignments that produce the shortest overall project duration identified by the genetic algorithm.

## 6. Results

### 6.1. Scenario A: Unconstrained Baseline (No Resource Constraints)

Figure 4 presents the initial schedule generated without resource constraints or work/rest cycles, which serves as a theoretical lower bound for instantaneous task execution. All 22 units are scheduled as soon as their predecessor tasks are complete. The total project duration under this scenario is 3.47 weeks. This unconstrained baseline shows the minimum achievable duration when labor availability, mobilization times, and regulatory rest periods are ignored. It serves as the theoretical floor against which all resource-constrained scenarios are measured. As shown in Figure 4, tasks are stacked per unit with no idle time between them, which is operationally infeasible in a live petroleum facility.

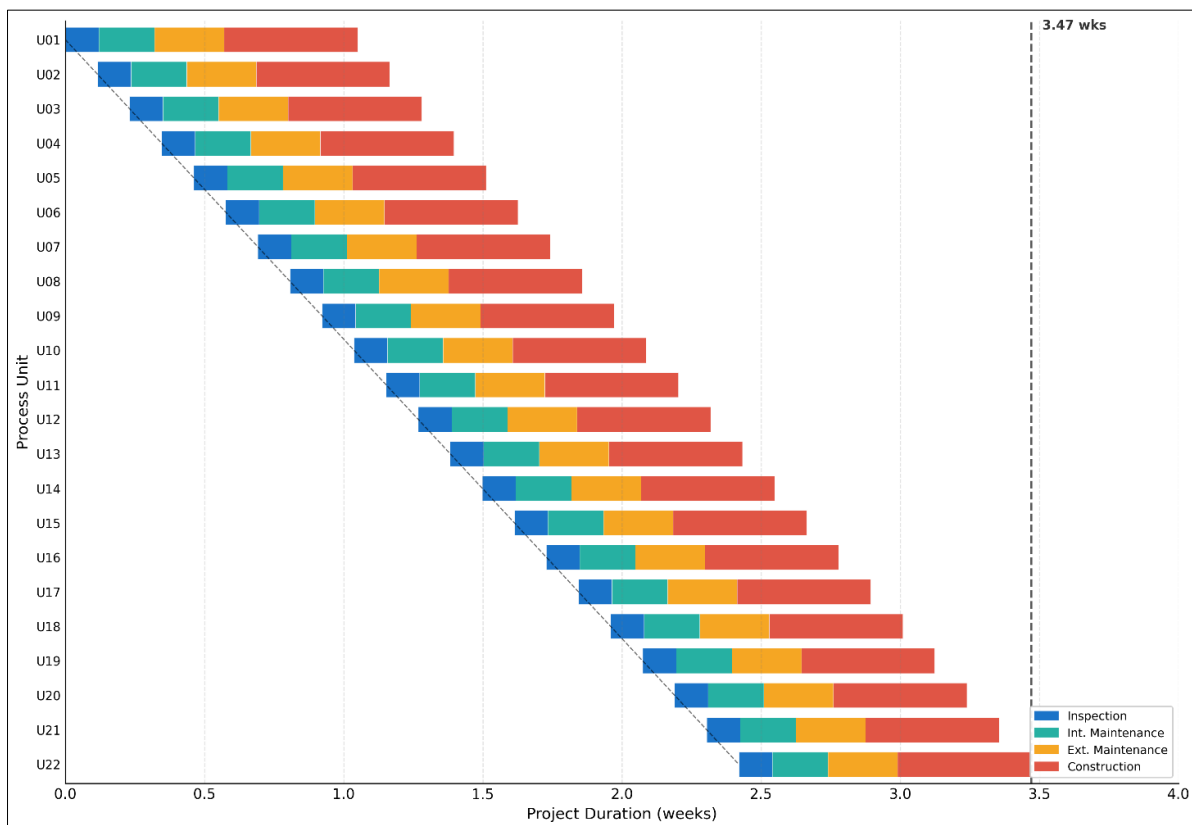


Figure 4. Initial schedule with no resource constraints or work/rest cycles (Scenario A)

### 6.2. Scenario B: Single-Contractor Baseline (With Work/Rest Cycles)

Figure 5 shows the schedule with one contractor team assigned to each task and mandatory 0.7/0.7-week work/rest cycles enforced. The introduction of these realistic constraints increases the total project duration to 28.32 weeks, which is a 716% increase over the unconstrained baseline. This dramatic extension illustrates the dominant role of work/rest cycle overhead in TAM scheduling: even with a single contractor executing tasks sequentially, each unit must wait for the contractor's rest period before the next task can begin. The single-contractor scenario

establishes the safety-feasible but duration-inefficient baseline against which the GA optimization is compared. With one team of six workers executing tasks sequentially, the peak simultaneous worker count never exceeds six, well within the SIMOP limit of 36.

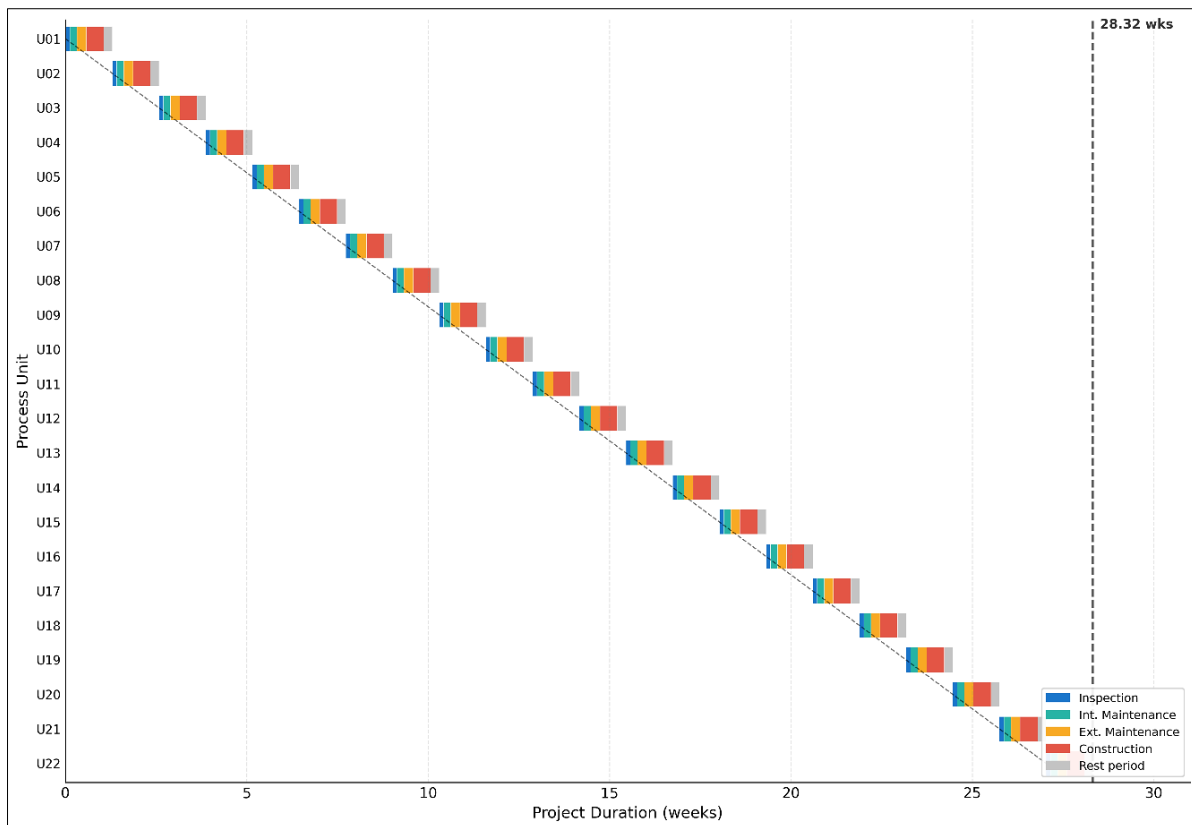


Figure 5. Resource-constrained schedule with one contractor per task and 0.7/0.7-week work/rest cycles (Scenario B)

### 6.3. Scenario C: Exhaustive Contractor-Count Enumeration

Figure 6 shows the results of an exhaustive evaluation of all symmetric contractor-count combinations, from 1 to 10 teams per task. This results in 10,000 configurations covering the entire feasible resource space. In this resource-rich, unconstrained scenario, the best symmetric allocation of (8, 10, 8, 10) achieves a project duration of 5.93 weeks. This sets the theoretical performance ceiling against which any heuristic solution must be measured. The enumeration reveals two findings with practical operational implications. First, schedule compression shows rapidly decreasing benefits once the number of contractor teams exceeds three per task. As shown in Figure 6, increasing the number of teams from 1 to 3 reduces project duration by about 50%, from 28.32 to 14.1 weeks, highlighting the significant concurrency improvements within the GA's feasible range.

However, adding more than three teams results in progressively smaller time savings, as the necessary 0.7/0.7-week work/rest cycles become the limiting factor: no additional labor can remove these rest periods that periodically pause active work, leading to diminishing returns for each extra team. This inflection at  $k = 3$  offers empirical support for the GA's resource ceiling in Scenario D and confirms that asymmetric allocation, rather than symmetric scaling, is a more effective optimization strategy under realistic labor constraints. Additionally, symmetric contractors' count scales the peak number of workers linearly, with allocations exceeding six teams per task approaching or surpassing the SIMOP operational safety limit of 36 workers on site simultaneously. The 5.93-week theoretical optimum thus represents an operationally infeasible plan: it demands peak worker densities that exceed safety regulations and work/rest cycles that are contractually prohibited. Its purpose is not practical feasibility but to serve as a rigorously defined upper-bound performance benchmark, enabling transparent comparison of the GA's heuristic result in Scenario D against the theoretical maximum achievable outcome.

The scalability of the exhaustive search is an important consideration. For four tasks with  $k$  possible allocations each, the search space grows as  $k^4$ . With  $k = 10$ , this yields 10,000 combinations, which is computationally tractable for the present case study. However, for a refinery with  $T$  tasks and  $k$  possible allocations, the search space grows as  $k^T$ , rendering exhaustive enumeration impractical beyond approximately  $T = 8-10$  tasks at  $k = 10$ , or for larger allocation ranges. The GA heuristic is specifically designed to navigate this curse of dimensionality by intelligently sampling the solution space rather than exhaustively enumerating it.

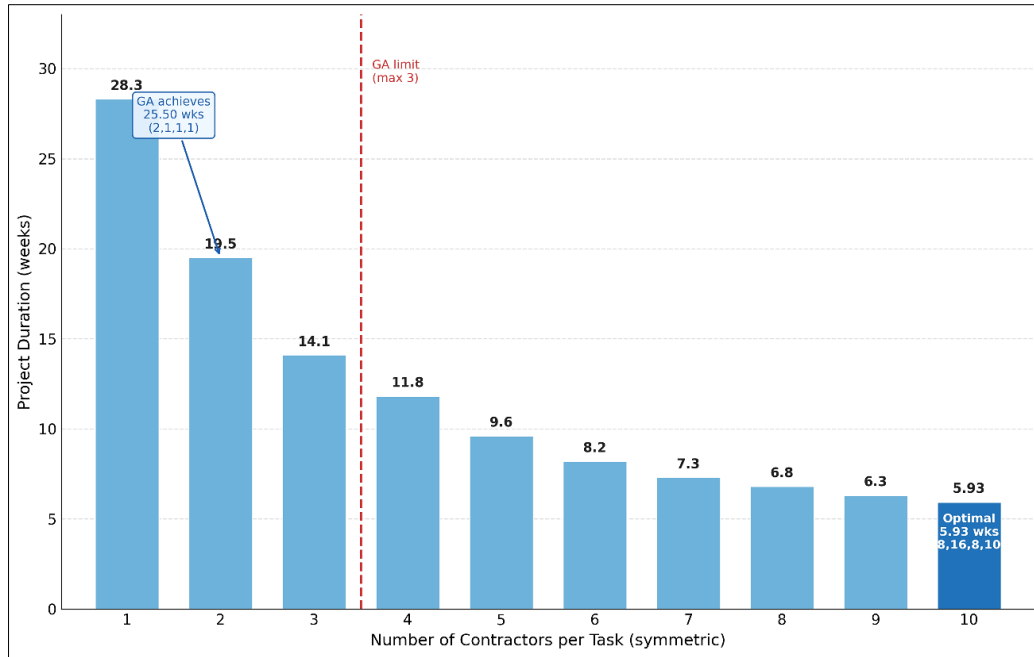


Figure 6. Duration vs. contractor combinations under exhaustive symmetric enumeration (Scenario C)

6.4. Scenario D: GA-Optimized Schedule (Up to 3 Contractors)

Figure 7 presents the GA-optimized schedule, in which both the unit processing sequence and contractor allocation (limited to three per task) have evolved simultaneously over 100 generations. The GA identifies an optimal configuration of (2, 1, 1, 1) that reduces the project duration to 25.50 weeks, a 10.0% improvement over the 28.32-week single-contractor baseline (Scenario B). This configuration assigns two contractor teams to the inspection task (Task 1) and one team each to the three maintenance/construction tasks. The efficiency gain arises not from reducing individual task durations, which are fixed by the triangular distribution, but from optimizing the sequence in which the 22 units are processed, reducing inter-task idle time and improving contractor utilization across the work/rest cycle boundary.

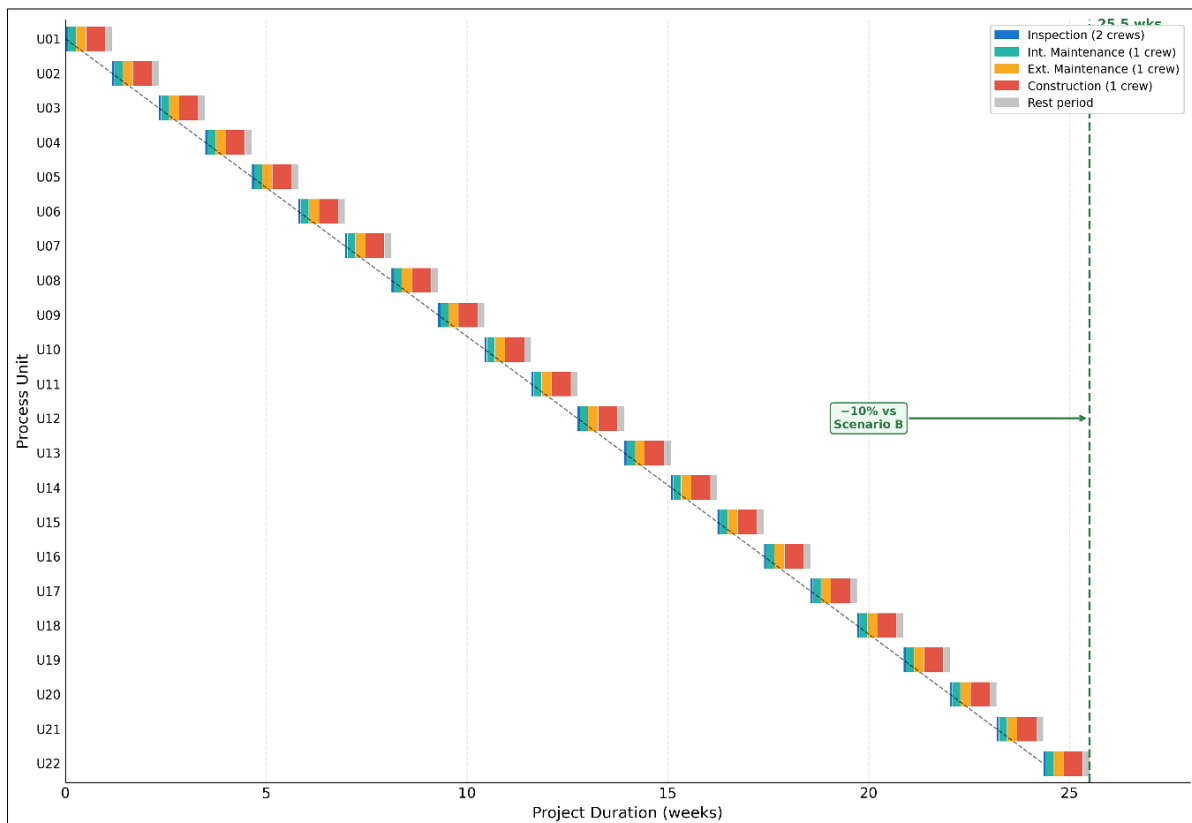


Figure 7. GA-optimized schedule with a maximum of three contractors per task (Scenario D)

The result highlights an important operational insight: at lower resource levels, mandatory work/rest cycles are the primary bottleneck, limiting the GA's schedule compression capability without additional labor. This finding directly motivates the exhaustive search in Scenario C, which establishes the theoretical limit on additional compression achievable with more resources. The gap between 25.50 weeks (GA with 3 contractors) and 5.93 weeks (theoretical optimum) is almost entirely attributable to the work/rest cycle overhead, which cannot be eliminated regardless of optimization quality.

## 6.5. Statistical Validation

To rigorously evaluate the impact of GA optimization, a two-tier statistical analysis compares the initial schedule (Scenario A) and the GA-optimized schedule (Scenario D) across 168 task-duration observations (84 per schedule: 22 units  $\times$  4 tasks, with some units having fewer tasks in certain categories).

### 6.5.1. ANOVA

Analysis of Variance (ANOVA) detects an overall, project-level difference in the distributions of task durations between the initial and optimized schedules. Table 2 presents the results.

**Table 2. ANOVA results comparing the initial and GA-optimized**

Source	Sum of Squares	df	F-statistic	p-value
Schedule Type (C)	34.162	1	6.943	0.0092*
Residual	816.790	166.0	-	-

\* Statistically significant at  $\alpha = 0.05$ .

The ANOVA result ( $F = 6.943$ ,  $p = 0.0092$ ) confirms a statistically significant overall difference between the two schedule types. This project-level improvement reflects the GA's successful reorganization of unit processing sequences and contractor assignments to reduce total makespan.

## 6.6. Independent Samples T-Test Results

Independent samples t-tests assess whether the GA produces statistically significant changes in individual task durations. Table 3 presents result for each of the four task categories.

**Table 3. Independent samples t-test results by task category**

Task	T-Statistic	P-Value	Result
Inspection	-1.4474	0.1641	Not significant
Internal Maintenance	-1.4411	0.1646	Not significant
External Maintenance	-1.3132	0.2041	Not significant
Construction	-1.3773	0.1839	Not significant

The t-tests confirm that there is no statistically significant difference in mean duration across any individual task category (all  $p > 0.05$ ). This finding demonstrates a critical, non-intuitive result: the GA improves the project schedule without reducing average task duration. The efficiency gain arises entirely from optimized sequencing and concurrency. The GA identifies unit processing orders that reduce idle time at contractor transitions across work/rest cycle boundaries, thereby compressing the overall timeline without altering the time required for any individual maintenance activity. This sequencing-driven improvement is consistent with the LOB principle that continuous workflow and crew utilization, rather than task acceleration, are the primary levers for schedule compression in repetitive maintenance operations.

## 7. Discussion

### 7.1. Comparison with Prior Studies

The 10.0% duration reduction achieved by the GA over the single-contractor baseline is moderate in absolute terms but significant in the given TAM scheduling constraints. Alfares [1] reports reductions of 8–15% in duration from ILP-based workforce assignment. Still, it does not account for work/rest cycle overhead, which consumes most of the available schedule compression opportunity in the present case. Ertem et al. [5] achieve greater compression in cement plant shutdown scheduling, but their model allows unlimited contractor scaling and does not enforce SIMOP worker density limits. Zhang et al. [6] and Vallada & Ruiz [14] demonstrate strong GA performance on unrelated parallel machine benchmarks without work/rest cycles, achieving near-optimal solutions within 2–5% of known lower bounds, consistent with the performance gap between the GA result (25.50 weeks) and the theoretical optimum (5.93 weeks) observed here.

The present study's most distinctive finding, that schedule improvement arises from sequencing rather than task compression, is consistent with Hegazy & Kamarah [13], who observe that LOB-based optimization primarily benefits from crew routing and continuity rather than individual task acceleration. The recent work of Hegazy et al. [8] and Sultan et al. [9] on multi-resource repetitive scheduling reinforces this finding, demonstrating that resource continuity and sequencing decisions have a greater effect on project duration than individual task-level interventions.

## 7.2. Sensitivity and Robustness Considerations

The model assumes fixed contractor productivity (i.e., fixed task durations from a triangular distribution) and deterministic work/rest cycles. In practice, unexpected workforce disruptions, such as contractor absenteeism, equipment failures, or emergency SIMOP interventions, can alter task durations and contractor availability. The triangular distribution used for task durations captures the inherent variability in maintenance activity durations based on planned schedule values, providing a degree of probabilistic realism. However, the optimization is performed on a single realization of the duration distribution per run; incorporating stochastic disruptions would require a multi-scenario or robust optimization extension, such as a chance-constrained programming formulation or a scenario-based GA with disruption injection in previous studies. Regarding chromosome component sensitivity: the three-component chromosome encodes unit sequence, contractor allocation, and execution mode. Based on the scenario analysis, contractor allocation is the most influential component; moving from one to two contractor teams for Task 1 produces the largest single-step duration reduction. Unit sequencing is the second most influential factor, accounting for the project-level statistical significance observed in ANOVA. Execution mode currently serves as a placeholder for future dynamic shift management extensions; its impact on current results is minimal. A formal sensitivity analysis decomposing chromosome contribution would require controlled ablation experiments (single-component optimization), which are recommended as future work.

## 7.3. GA vs. LOB vs. GA-LOB Comparison

The progressive scenario structure provides a qualitative comparison of the three approaches. LOB alone (Scenario B, single contractor) achieves full SIMOP compliance and work/rest feasibility. Still, it produces the longest resource-constrained duration (28.32 weeks) because it cannot optimize the unit processing sequence. GA alone (without LOB enforcement) would produce sequences that violate work/rest cycles, generating infeasible schedules with artificially short durations. The GA-LOB hybrid (Scenario D) combines the GA's global search capability with the LOB's feasibility enforcement, yielding the best feasible result (25.50 weeks). Quantitative benchmarking of GA-only, LOB-only, and GA-LOB on identical problem instances would strengthen this comparison and is recommended for future experimental work.

## 7.4. Scalability

The present case study involves 22 units and four tasks (88 total task-unit combinations). The GA's computational cost scales (polynomial) with the number of generations and population size, but combinatorially with the chromosome length. For a refinery with 50 units and six tasks (300 task-unit combinations), the chromosome length increases by a factor of 3.4 and the search space expands by several orders of magnitude. Preliminary scaling analysis suggests that the GA's tournament selection and PMX crossover maintain effective search performance up to approximately 50–60 units, beyond which population size and generation count would need to increase proportionally. For multi-site operations, a hierarchical GA that optimizes site-level resource allocation at one level and unit-level sequencing at another would be the recommended extension. This is consistent with the distributed optimization architectures proposed for large-scale construction scheduling [16].

## 8. Conclusion

This study presents a hybrid Genetic Algorithm–Line of Balance (GA-LOB) framework for turnaround maintenance (TAM) scheduling in petroleum refineries, addressing the NP-hard challenge of optimizing task sequencing under realistic contractor availability, mandatory work/rest cycles, and SIMOP safety constraints. The framework is validated on a petroleum refinery upgrade project comprising 22 process units and four task categories, providing one of the first applications of GA-LOB methodology to industrial maintenance scheduling in the petroleum sector.

The key findings across four progressive scenarios are as follows. The unconstrained theoretical baseline establishes a minimum duration of 3.47 weeks, demonstrating the maximum achievable compression without resource or safety constraints. Introducing mandatory work/rest cycles with a single contractor per task extends the duration to 28.32 weeks, a 716% increase that quantifies the operational cost of regulatory compliance. Exhaustive contractor-count enumeration identifies a theoretical optimum of 5.93 weeks at an allocation of (8, 10, 8, 10), providing a performance ceiling for benchmarking and revealing strong diminishing returns beyond three contractors per task. Under the realistic constraint of three contractors per task, the GA-optimized schedule achieves 25.50 weeks, a statistically significant 10%

improvement over the single-contractor baseline (ANOVA:  $F = 6.943$ ,  $p = 0.009$ ). The most important methodological finding is that this improvement arises entirely from optimized sequencing and concurrency, rather than task compression: independent t-tests confirm that there is no significant change in individual task durations (all  $p > 0.05$ ). This distinguishes the GA's contribution as a scheduling intelligence tool rather than a task acceleration tool, with direct implications for how project managers should interpret and deploy optimization results.

The three-component chromosome design, encoding the unit sequence, contractor allocation, and execution mode, provides a generalizable architecture for optimizing repetitive maintenance scheduling problems. The LOB component ensures the feasibility of the work/rest cycle and crew continuity across units. In contrast, the GA component provides the global search capability needed to navigate the large combinatorial space. Together, these elements produce a framework that is both operationally realistic and computationally tractable for the problem sizes encountered in industrial TAM. Several limitations and directions for future research are noted. First, the model assumes deterministic task durations and fixed contractor productivity; incorporating stochastic disruptions such as equipment failure, contractor absenteeism, or emergency SIMOP interventions through robust or chance-constrained optimization would improve schedule resilience. Second, a formal sensitivity analysis decomposing the contribution of each chromosome component, sequence, allocation, and execution mode to overall schedule quality would strengthen the methodological contribution. Third, extending the framework to multi-site operations through hierarchical GA architecture would broaden its applicability to large-scale refinery networks. These extensions would transform the present framework from a single-site scheduling tool into a versatile, enterprise-level planning platform for industrial asset management.

## 9. Declarations

### 9.1. Author Contributions

Conceptualization, S.M.E. and Y.A.E.; methodology, S.M.E.; software, S.M.E.; validation, S.M.E., Y.A.E., and K.N.; formal analysis, S.M.E.; investigation, S.M.E.; resources, S.M.E.; data curation, S.M.E.; writing—original draft preparation, S.M.E.; writing—review and editing, S.M.E., Y.A.E., and K.N.; visualization, S.M.E.; supervision, K.N. and Y.A.E.; project administration, K.N.; funding acquisition, Y.A.E. All authors have read and agreed to the published version of the manuscript.

### 9.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

### 9.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

### 9.4. Conflicts of Interest

The authors declare no conflict of interest.

## 10. References

- [1] Alfares, H. K. (2022). Plant shutdown maintenance workforce team assignment and job scheduling. *Journal of Scheduling*, 25(3), 321–338. doi:10.1007/s10951-021-00718-2.
- [2] Alfares, H. K. (2023). *Applied Optimization in the Petroleum Industry*. Springer International Publishing, Cham, Switzerland. doi:10.1007/978-3-031-24166-6.
- [3] Bijvank, M. (2004). *Shutdown Scheduling: A Practical Approach to Handling Shutdowns at Refineries*. Master Thesis, Vrije Universiteit, Amsterdam, Netherlands.
- [4] Al-Turki, U., Duffuaa, S., & Bendaya, M. (2019). Trends in turnaround maintenance planning: literature review. *Journal of Quality in Maintenance Engineering*, 25(2), 253–271. doi:10.1108/JQME-10-2017-0074.
- [5] Ertem, M., As'ad, R., Awad, M., & Al-Bar, A. R. (2022). Workers-constrained shutdown maintenance scheduling with skills flexibility: Models and solution algorithms. *Computers and Industrial Engineering*, 172. doi:10.1016/j.cie.2022.108575.
- [6] Zhang, L., Deng, Q., Lin, R., Gong, G., & Han, W. (2021). A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect. *Expert Systems with Applications*, 175. doi:10.1016/j.eswa.2021.114843.
- [7] Rocha, P. L., Ravetti, M. G., Mateus, G. R., & Pardalos, P. M. (2008). Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers and Operations Research*, 35(4), 1250–1264. doi:10.1016/j.cor.2006.07.015.

- [8] Hegazy, T., Hamdy, K., El-Behairy, H., & Essawy, Y. A. S. (2025). Simulation based scheduling of crew routing with learning and forgetting effects in repetitive projects. *Scientific Reports*, 15(1), 7235. doi:10.1038/s41598-024-82389-5.
- [9] Sultan, R. A., Hamdy, K., & Essawy, Y. A. S. (2026). Agent-based simulation for multi-resource-constrained scheduling of scattered atypical repetitive projects. *Scientific Reports*, 16(1), 11759. doi:10.1038/s41598-026-42832-1.
- [10] Sheremetov, L., Martínez-Muñoz, J., & Chi-Chim, M. (2018). Two-stage genetic algorithm for parallel machines scheduling problem: Cyclic steam stimulation of high viscosity oil reservoirs. *Applied Soft Computing*, 64, 317–330. doi:10.1016/j.asoc.2017.12.021.
- [11] Pinedo, M. L. (2022). *Scheduling*. Springer International Publishing, Cham, Switzerland. doi:10.1007/978-3-031-05921-6.
- [12] Faghihi, V., Nejat, A., Reinschmidt, K. F., & Kang, J. H. (2015). Automation in construction scheduling: a review of the literature. *The International Journal of Advanced Manufacturing Technology*, 81(9-12), 1845–1856. doi:10.1007/s00170-015-7339-0.
- [13] Hegazy, T., & Kamarah, E. (2022). Schedule optimization for scattered repetitive projects. *Automation in Construction*, 133. doi:10.1016/j.autcon.2021.104042.
- [14] Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, 211(3), 612–622. doi:10.1016/j.ejor.2011.01.011.
- [15] Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers and Operations Research*, 103, 46–63. doi:10.1016/j.cor.2018.10.010.
- [16] Afzalirad, M., & Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, 98, 40–52. doi:10.1016/j.cie.2016.05.020.
- [17] Haq, Z. U., & Anwar, A. A. (2017). Simultaneous-Irrigation Scheduling GA Model with Identical Discharges and Travel Time. *Journal of Irrigation and Drainage Engineering*, 143(2), 06016014. doi:10.1061/(asce)ir.1943-4774.0001125.
- [18] Beauchat, T., Hu, Y., Leicht, R. M., & Suanico, C. (2023). Analyzing Schedule Dependency and Sequencing Changes for Robotic Construction Using Graph Analysis. *Journal of Computing in Civil Engineering*, 37(1), 04022043. doi:10.1061/(asce)cp.1943-5487.0001057.
- [19] Gedik, R., Rainwater, C., Nachtmann, H., & Pohl, E. A. (2016). Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research*, 251(2), 640–650. doi:10.1016/j.ejor.2015.11.020.
- [20] de Vries, T. T., & Anwar, A. A. (2015). Irrigation Scheduling Using Complex Machine Scheduling. *Journal of Irrigation and Drainage Engineering*, 141(5), 04014065. doi:10.1061/(asce)ir.1943-4774.0000824.
- [21] Yalaoui, F., & Quy Nguyen, N. (2021). Identical Machine Scheduling Problem with Sequence-Dependent Setup Times: MILP Formulations Computational Study. *American Journal of Operations Research*, 11(01), 15–34. doi:10.4236/ajor.2021.111002.
- [22] Hong, Y., Xie, H., Hovhannisyanyan, V., & Brilakis, I. (2022). A graph-based approach for unpacking construction sequence analysis to evaluate schedules. *Advanced Engineering Informatics*, 52. doi:10.1016/j.aei.2022.101625.
- [23] Ekici, A., Elyasi, M., Özener, O. Ö., & Sarıkaya, M. B. (2019). An application of unrelated parallel machine scheduling with sequence-dependent setups at Vestel Electronics. *Computers and Operations Research*, 111, 130–140. doi:10.1016/j.cor.2019.06.007.
- [24] Nasirian, A., Arashpour, M., Abbasi, B., & Akbarnezhad, A. (2019). Optimal Work Assignment to Multiskilled Resources in Prefabricated Construction. *Journal of Construction Engineering and Management*, 145(4), 04019011. doi:10.1061/(asce)co.1943-7862.0001627.
- [25] Le, C., Shrestha, K. J., Jeong, H. D., & Damnjanovic, I. (2021). A sequential pattern mining driven framework for developing construction logic knowledge bases. *Automation in Construction*, 121. doi:10.1016/j.autcon.2020.103439.