SCIENTIFIC OASIS

Decision Making: Applications in Management and Engineering

Journal homepage: www.dmame-journal.org
ISSN: 2560-6018, eISSN: 2620-0104

A Multi-Objective Decision-Making Framework for Optimizing Resource Allocation, Project Scheduling, and Financial Performance in Repetitive Construction Projects

Yasser Aboelmagd[1,*] , Remon Eskander[2], Omnia Darwish[3]

1 Vice Dean, College of Engineering, University of Business & Technology (UBT), Jeddah, Saudi Arabia.
2 Mathematics and Physics Engineering Department, Faculty of Engineering, Alexandria University, Egypt.
3 Engineering Programs Director, Field of Engineering, Alamein International University (AIU), New Al-Alamein City, Matrouh - Egypt.
4 Structural Engineering Department, Faculty of Engineering, Alexandria University, Egypt.
5 Construction engineering Department, Higher institute of engineering and technology – ElBheira, Egypt.

ARTICLE INFO

Article history:
Received 11 March 2026
Received in revised form 30 May 2026
Accepted 13 June 2026
Available online 30 June 2026

Keywords:
Mega Construction, Repetitive Projects, Multi-Resources, Multi-Objective Optimization, Optimizing Strategy, Scheduling Software, Tender Data Estimation.

ABSTRACT

Accurate forecasting of critical tender parameters, including anticipated project duration, total gross valuation and cash flow patterns, constitutes a fundamental yet demanding responsibility for contractors operating within the highly competitive construction sector. Given the persistent need to reconcile constrained profit margins with efficient project delivery, this study presents Optimizing Strategy Software (OSS), a Java-based, multi-objective optimisation framework developed specifically for repetitive construction projects characterised by uniform activity durations and single-crew assignments. Rather than relying exclusively on conventional estimation approaches, OSS systematically examines alternative multi-mode resource configurations to identify technically feasible and economically advantageous project execution scenarios. Its principal purpose is to provide practical, scenario-oriented decision support through which project managers can optimise resource allocation while concurrently reducing total project duration, final tender price and peak working capital requirements, alongside maximising net present value (NPV). To ensure compatibility with established computational project management environments, the software incorporates a four-tier architecture consisting of a user interface module, a centralised relational database module, a computational execution module and a dynamic connecting module. The proposed theoretical formulation and computational functionality are subsequently examined through a detailed illustrative case study involving a large-scale repetitive expressway construction project. The application demonstrates the effectiveness of OSS in identifying favourable resource allocation strategies and improving the optimisation of interrelated tender parameters, thereby establishing its practical relevance as an integrated decision-support framework for repetitive construction planning.

1. Introduction

Effective construction project planning requires the coordinated management of financial and

* *Corresponding author.*
E-mail address: yasser@ubt.edu.sa

https://zenodo.org/records/24377851

operational considerations, particularly where competing objectives necessitate careful resource allocation decisions. Planners must simultaneously pursue shorter project durations, lower tender prices and reduced peak working capital requirements while seeking to maximise NPV. Resolving these interconnected objectives involves complex multi-objective optimisation and systematic assessment of trade-offs, typically undertaken during preliminary project planning [1; 2]. Nevertheless, establishing project feasibility requires examining numerous alternative execution scenarios, often beyond the analytical capacity of conventional scheduling approaches. Consequently, advanced computational heuristics, particularly evolutionary techniques such as Genetic Algorithms (GAs) and Particle Swarm Optimisation (PSO), offer effective mechanisms for exploring multidimensional solution spaces [3; 4].

Within this context, the present research develops OSS using a GA-based optimisation framework designed to generate improved Pareto-optimal solutions for competing time, cost and resource objectives compared with existing iterative approaches [5]. The study initially examines the theoretical foundations and applications of contemporary optimisation algorithms in construction management before detailing the development architecture, computational procedures and principal functional components of OSS. Its practical applicability and computational performance are subsequently investigated through an illustrative case study, incorporating a comparative evaluation against conventional manual optimisation procedures. This assessment seeks to establish the potential improvements in computational efficiency and estimation accuracy achievable through the proposed system.

2. Overview

Selecting an appropriate scheduling approach requires consideration of several interdependent project characteristics, including overall scale, technical complexity, expected completion period, workforce availability and stakeholder-specific requirements. Within the conceptual classification presented in Figure 1, construction projects are broadly distinguished according to whether their activity structures are repetitive or non-repetitive. Scheduling non-repetitive construction operations has traditionally relied on two principal methodological approaches. The first comprises bar-chart scheduling, commonly represented through Gantt charts, which became established following their development by Henry Gantt during World War I.

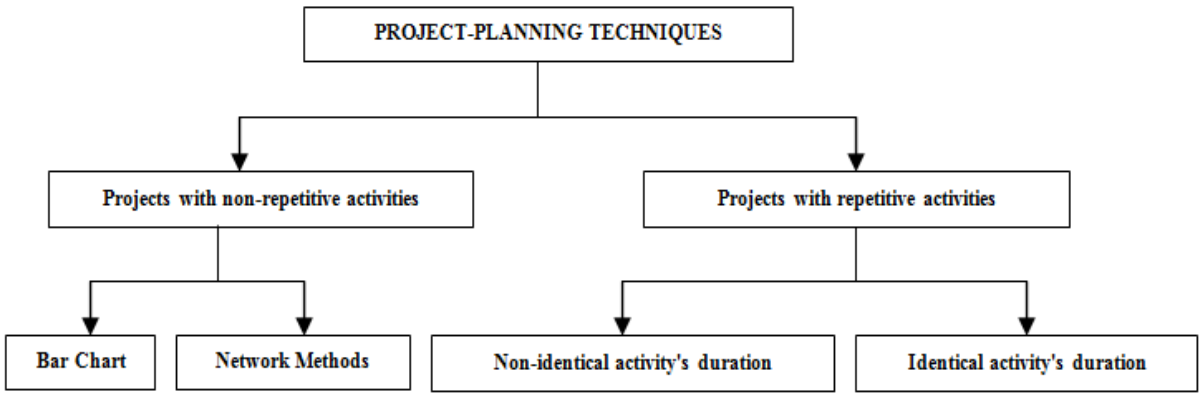


Fig.1: Classification and Analysis of Current Planning Techniques in the Construction Industry

The second encompasses network-based scheduling techniques, notably the Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), independently introduced during the late 1950s to provide mathematically structured representations of activity relationships and

scheduling constraints. Repetitive construction projects, by contrast, may be classified into linear and non-linear activity sequences. Linear repetitive operations involve standardised construction stages in which equivalent activities are performed across successive units with uniform execution durations. Such recurring production cycles, including those associated with multi-storey building floors and residential housing developments, require scheduling methods capable of coordinating resource movement and sustaining uninterrupted workflow. Accordingly, specialised approaches, particularly the Line of Balance (LOB) and Vertical Production Method (VPM), facilitate the planning and control of repetitive operations by maintaining resource continuity, coordinating successive activities and improving productivity throughout project implementation [6; 7].

Optimising construction project planning constitutes a particularly demanding class of combinatorial optimisation problems, attracting sustained scholarly attention since the 1960s. Within computational complexity theory, these problems are generally recognised as Nondeterministic Polynomial-time hard (NP-hard), reflecting the substantial computational effort associated with exploring numerous interacting decision variables, operational constraints and feasible scheduling combinations [8; 9]. Three principal methodological categories have traditionally been employed to address such challenges, as classified in Figure 2. Heuristic methods rely on practical experience, established decision rules and expert judgement to identify workable solutions without necessarily guaranteeing mathematical optimality. Mathematical programming, by comparison, adopts formal analytical formulations in which continuous or integer decision variables are evaluated within specified feasible regions to minimise or maximise defined objective functions.

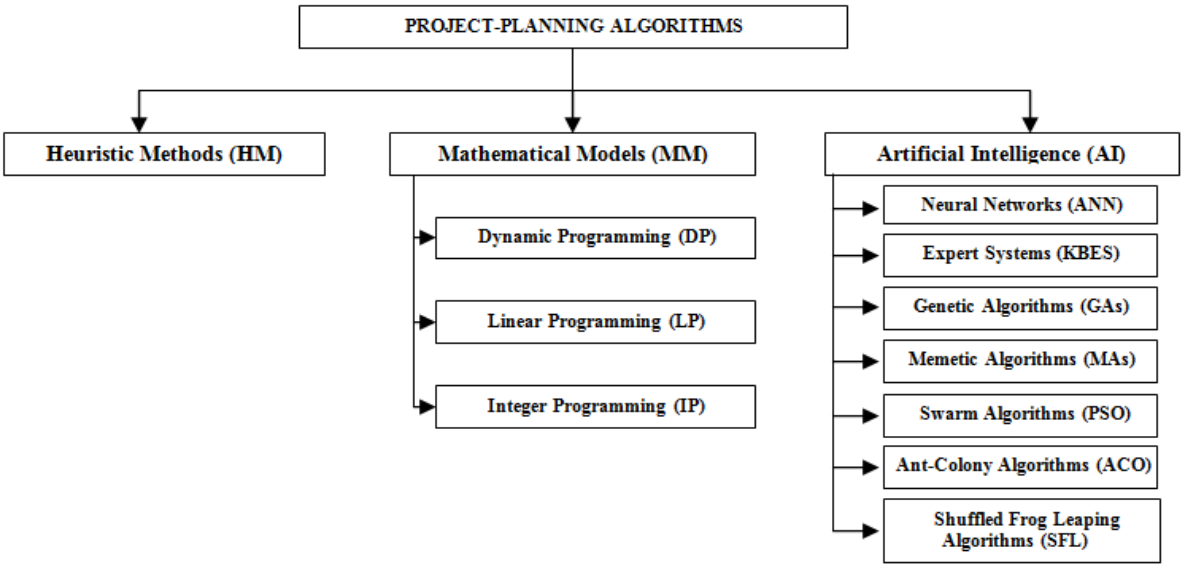


Fig.2: A Comprehensive Classification of Contemporary Optimization Algorithms

Evolutionary algorithms constitute a third methodological category, employing stochastic, population-based search mechanisms inspired by biological evolution and nature-inspired collective behaviour to identify favourable configurations within complex solution spaces [3]. Their applicability to construction planning has been demonstrated repeatedly in recent multi-objective formulations. The non-dominated sorting genetic algorithm II (NSGA-II), for example, was used to minimise duration and cost simultaneously in Indian highway projects, generating Pareto-optimal solutions that surpassed traditional trade-off models [2]. A multi-objective sea horse algorithm was likewise developed to balance time, cost, quality, risk and environmental performance, with a multiple-criteria decision analysis step supporting the final selection [1]. Despite their potential, GAs retain

computational challenges, notably the processing effort required to approach near-optimal solutions in extensive search spaces. A parameter-adaptive improved GA combined with differential evolution, for instance, reduced project duration by up to 22.7% and resource variance by up to 29.9% in a case study [10].

Hybridisation has therefore become a prominent route to improving convergence and solution quality. A memetic algorithm that integrates a GA, simulated annealing and a resource-levelling heuristic was reported to outperform state-of-the-art levelling heuristics and metaheuristics for the resource levelling problem [8]. A multi-objective resource-constrained model for large repetitive projects used GAs to represent the actual loss incurred when resource utilisation priorities are not met, including the cost of exceeding daily resource availability, moving resources on and off site, and delaying completion [9]. Their case study of 16 repetitive projects comprising 160 activities showed that the need for formal optimisation grows as project size and complexity increase.

The expansion of evolutionary optimisation beyond conventional time–cost relationships has introduced additional dimensions of construction performance. A duration–cost–sustainability model for prefabricated building projects, in which carbon emissions were embedded in the cost objective, was solved with NSGA-II and its Pareto solutions were ranked using an efficacy-coefficient method [11]. In the same prefabrication context, resource-constrained scheduling models based on a GA and an improved elitist NSGA-II were designed to balance time, cost and low-carbon transportation [12]. These studies indicate that environmental performance is increasingly treated as an integral optimisation criterion rather than a supplementary consideration introduced after scheduling decisions have been made.

Resource allocation research has also extended to the portfolio level. Multi-mode resource-constrained multi-project scheduling integrated with supplier selection was addressed using a multi-objective whale optimisation algorithm benchmarked against NSGA-II, with reported financial gains of up to 6% and time savings of up to 16% in Iranian construction projects [4]. Such formulations confirm that selecting an appropriate solution from a Pareto front requires additional decision criteria that reflect the priorities of the decision-maker.

Research on repetitive construction scheduling has similarly progressed towards integrated optimisation formulations capable of representing the operational characteristics of recurring activities. A multi-objective model that selects the number of concurrent crews and their work sequence to minimise project duration and cost achieved reductions of 8% and 0.78%, respectively, relative to earlier models in a benchmark application [6]. Where productivity is uncertain, a Monte Carlo simulation integrated with a multi-objective GA was used to identify crew deployment plans that minimise the expected project duration and cost [13]. A metaheuristic multi-objective model for repetitive projects further incorporated learning effects, soft logic and work-interruption allowances to improve the accuracy of duration, cost and interruption estimates [14]. At the resource level, an adaptive line-of-balance model that allows the number of crews to change at selected units, and that accommodates non-serial repetitive projects, outperformed existing line-of-balance models in minimising resource consumption and fluctuations in a pipeline project [7].

Financial liquidity represents another important dimension of construction optimisation, particularly where project execution depends on maintaining adequate working capital. A GA-based finance-based scheduling model for multiple projects shifted activity start dates within available float to maximise profit while minimising total duration and keeping activity financing within available funds [15]. Extending these considerations to repetitive construction tendering, a multi-objective optimisation methodology implemented through OSS was proposed to determine tender data by simultaneously considering project duration, tender price and working capital requirements while

seeking to maximise NPV [5]. By integrating scheduling decisions with financial evaluation, OSS enables alternative tender scenarios to be assessed against multiple interdependent objectives, providing a more comprehensive basis for construction planning than approaches focused exclusively on duration or direct expenditure.

Subsequent research has increasingly emphasised the integration of evolutionary optimisation with digital construction management technologies. A BIM-based framework that extracts information from the BIM model, applies an ontology constraint rule and then employs a GA was proposed to generate construction schedules automatically [16]. A structured review of BIM-based multi-objective optimisation in construction scheduling similarly highlighted the potential of combining information systems with optimisation tools while identifying gaps for future research [17]. More recently, a BIM-semantic model combining multi-objective particle swarm optimisation with a Deep Q-Network was developed for prefabricated building projects operating under limited resources [18].

Learning-based approaches have emerged as a complementary paradigm. A deep reinforcement learning (DRL) model with valid action sampling and a graph convolutional network was proposed for the automated scheduling and rescheduling of large construction projects, outperforming traditional methods in both project duration and runtime [19]. At the organisational level, DRL agents based on Double Deep Q-Networks and Internet of Things (IoT) data performed autonomous resource allocation across multiple projects without requiring retraining when conditions changed [20].

Large language models (LLMs) and blockchain technologies have also begun to be explored. ChatGPT generated a coherent construction schedule for a simple project that participants evaluated positively, although the technology required further development before wide industrial adoption [21]. A blockchain-based electronic project-recording framework, supported by optimisation models for scheduling and sequencing multiple projects under probabilistic arrival and stochastic durations, provided effective planning and monitoring at minimal idle and overtime costs [22]. Nevertheless, the practical effectiveness of such technologies remains dependent on data reliability, computational interoperability and the capacity of optimisation models to accommodate changing constraints without compromising scheduling feasibility.

Collectively, the reviewed literature demonstrates a progressive transition from conventional combinatorial optimisation towards integrated, multi-objective and data-driven construction planning frameworks. Initial investigations established the applicability of GAs and NSGA-II to resource allocation, schedule generation and time–cost optimization [2; 9], while subsequent developments incorporated environmental performance, resource continuity and financial liquidity into increasingly comprehensive formulations [11; 15]. The integration of BIM and hybrid metaheuristic techniques further improved the representation of project information and strengthened computational search capabilities [8; 17]. More recent developments involving DRL, LLMs and blockchain-based project records indicate growing interest in adaptive optimisation under uncertain operational conditions [19; 21; 22]. Despite these advances, most recent models optimise time–cost pairs or resource profiles, and a fundamental challenge remains in coordinating feasible construction alternatives across competing scheduling, financial, resource and environmental objectives. Addressing this challenge requires integrated decision-support frameworks capable of translating complex computational outcomes into practical construction planning and tendering decisions, which is the gap addressed by OSS.

3. Genetic Algorithms, Gas

Grounded in Darwinian principles of natural selection and biological evolution, GAs are population-based stochastic search techniques designed to identify optimal or near-optimal

solutions within complex, multidimensional decision spaces [3]. Their computational procedure begins with the random generation of an initial population comprising multiple candidate solutions, each represented by a chromosome encoding the decision variables relevant to the optimisation problem. The suitability of each chromosome is subsequently assessed through a predefined fitness function that quantifies its performance against the specified optimisation objectives. Based on these evaluations, a selection mechanism identifies comparatively favourable individuals for reproduction, thereby increasing the likelihood that advantageous solution characteristics will be retained in subsequent generations.

Genetic operators, principally crossover and mutation, are then applied to the selected chromosomes to generate new candidate solutions while maintaining diversity within the search population. Through successive cycles of evaluation, selection and genetic modification, the algorithm explores alternative regions of the solution space and progressively seeks configurations exhibiting improved objective performance. This iterative process reflects the evolutionary principle of survival of the fittest, although improvements between successive generations are not necessarily guaranteed. The search continues until a predefined stopping criterion is satisfied, such as achieving an acceptable fitness level, reaching a maximum number of generations or observing convergence in solution quality.

Figure 3 illustrates the fundamental computational sequence underlying the conventional GA procedure. The process commences with the stochastic establishment of an initial population, followed by the evaluation of individual chromosomes using the designated fitness function. The resulting fitness values guide the selection of parent solutions, which subsequently undergo crossover and mutation to produce offspring representing alternative configurations of the decision variables. These newly generated chromosomes are evaluated and incorporated into the evolutionary population according to the adopted replacement strategy. Repeated execution of these operations enables the algorithm to balance the exploration of previously unexamined solutions with the exploitation of promising regions identified during earlier iterations. The evolutionary cycle terminates when the specified convergence threshold or other predefined stopping condition is reached, yielding the best available solution or a set of near-optimal alternatives. As depicted in Figure 3, this structured sequence of population generation, fitness assessment, parent selection, genetic recombination and iterative refinement provides the computational foundation for applying GAs to complex construction scheduling and resource optimisation problems.

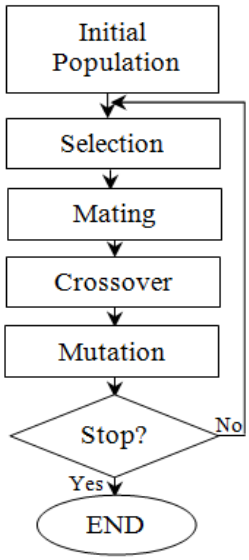


Fig.3: A Canonical Framework for Genetic Algorithms

The fundamental computational structure of GAs follows an iterative evolutionary framework in which successive populations of candidate solutions are progressively refined through biologically inspired search operations, as reviewed by Alhijawi and Awajan [3]. Initially, a population containing n randomly generated chromosomes is established, with each chromosome encoding a feasible candidate solution whose relative suitability is assessed using a predefined fitness function, f(x). Population evolution subsequently proceeds through a selection procedure that favours comparatively high-performing individuals for reproduction. Common selection strategies include roulette-wheel, rank-based, tournament and elitist selection [3]. The selected parent chromosomes then undergo crossover, a principal genetic recombination operation regulated by a user-specified Crossover Probability (CP), typically ranging from 0.1 to 1.0.

Depending on the chromosome representation and problem-specific constraints, crossover may involve exchanging genetic segments at one or two randomly selected positions, probabilistically inheriting individual genes from either parent through uniform crossover, commonly with a selection probability of 0.5, or generating offspring through weighted combinations of parental values using arithmetic crossover for real-valued representations. The three principal crossover paradigms are illustrated in Table 1. To preserve population diversity and reduce the likelihood of premature convergence, mutation introduces stochastic modifications at selected chromosome positions with a relatively low probability. The resulting offspring are incorporated into the succeeding population through an appropriate replacement mechanism, allowing evolutionary refinement to continue across successive generations. This iterative sequence terminates once a predetermined stopping condition is satisfied, after which the chromosome exhibiting the highest fitness is identified as the best solution obtained through the search process, although global optimality cannot necessarily be guaranteed.

Table 1
Primary Crossover Paradigms in Genetic Algorithms

Single point crossover		Two point crossover		Uniform crossover	
Parent A:	11001**010**	Parent A:	11**0010**10	Parent A:	1**10**010**10**
Parent B:	**01110**011	Parent B:	**01**1100**11**	Parent B:	**0**11**100**11
Offspring A:	11001011	Offspring A:	11110010	Offspring A:	11101011
Offspring B:	**01110010**	Offspring B:	**01001011**	Offspring B:	**01010010**

Note: Bold underlined digits denote the gene segments exchanged between Parent A and Parent B to produce Offspring A and Offspring B.

Unlike crossover, which recombines genetic information inherited from parent chromosomes, mutation introduces random modifications into individual genetic representations, thereby reproducing the principle of spontaneous variation observed in biological evolution. This operator modifies selected components of an existing chromosome and plays an essential role in preserving genetic diversity throughout successive generations. By introducing previously unexplored genetic configurations, mutation reduces the likelihood of premature convergence towards inferior solutions and improves the algorithm's capacity to explore regions beyond local optima. In binary-encoded GAs, mutation is implemented by reversing the binary state of selected genes according to a predefined Mutation Probability (MP). The appropriate mutation rate depends on the characteristics, dimensionality and constraints of the optimisation problem. In practice, relatively low mutation probabilities, typically below 0.1, are commonly adopted to balance the exploration of alternative solutions against the exploitation of promising configurations, although the appropriate rate remains problem-dependent [3].

3.1 Employed Techniques of Proposed Model

The proposed analytical framework integrates established construction scheduling procedures with financial evaluation and multi-objective optimisation techniques, following the methodological approach of Aziz [5]. Project activities and their logical dependencies are initially represented using the Precedence Diagram Method (PDM), while LOB is subsequently employed to describe the progression of repetitive activities (k = 1, 2, …., K) throughout the construction schedule. To strengthen the computational representation of activity relationships, conventional LOB scheduling information is transformed into a modified CPM network. Operational continuity and scheduling flexibility are addressed through the incorporation of time buffers (TB_) between successive activities at each construction stage (s = 1, 2, …., S).

Additional stage buffers (SB_), specified by the project planner, accommodate technological dependencies and practical execution constraints between sequential operations. From an economic perspective, the framework adopts discrete time–cost–price relationships for alternative resource execution modes, assuming that the selected mode remains consistent across all stages associated with a particular activity. Financial performance is subsequently evaluated through LOB-based cash flow analysis, enabling the integration of construction scheduling decisions with project-level financial requirements. These computational components are combined within a multi-objective optimisation formulation that simultaneously seeks to minimise overall project duration, tender price and maximum working capital requirements while maximising NPV.

3.2 Theoretical Framework and Core Assumptions

The analytical formulation follows the conceptual framework of Aziz [5] and incorporates a defined set of assumptions intended to facilitate construction scheduling and financial performance assessment. From an operational perspective, uninterrupted crew deployment is maintained throughout the project, eliminating idle periods and preserving uniform production rates between the commencement of the first stage and completion of the final stage. This requirement is supported by assigning identical average execution durations to corresponding activities across successive construction stages. Where scheduling adjustments become necessary to accommodate particular project requirements, the revised durations are applied consistently across all stages of the relevant activity. Each construction technique is assigned a single crew, while productivity improvements associated with learning effects are excluded from the computational formulation. Furthermore, individual activities are executed sequentially, with work progressing through one construction unit at a time. Financial assumptions establish a predetermined payment arrangement under which the employer retains 10% of the value of completed work and releases the remaining 90% through interim payments delayed by one payment interval, typically one week for civil engineering projects.

Conversely, the contractor is assumed to discharge all subcontractor and supplier payments immediately and without deductions. The framework additionally assumes constant material prices, excludes mobilisation advances and provides for the immediate reimbursement of accumulated retention upon project handover. To isolate the relationships among scheduling, resource allocation and financial performance, unforeseen disruptions and external incidents are disregarded, overhead expenditure is considered independent of project duration and progress, and interim payment arrangements remain predetermined. Resource allocation flexibility is nevertheless preserved through the availability of alternative execution modes, with the number and characteristics of feasible options varying according to individual activity requirements.

3.3 Technical Implementation Framework

The computational implementation of the proposed methodology comprises three sequential phases that collectively support the identification and refinement of feasible project execution alternatives. The first phase establishes the initial population of u candidate solutions, each representing a particular configuration of resource deployment across construction activities. During the second phase, these alternatives undergo fitness evaluation using four principal performance indicators: total project duration, overall project cost, maximum working capital requirements and NPV. The third phase applies iterative population improvement procedures to progressively enhance solution quality over successive generations. Through the coordinated execution of these phases, the optimisation framework explores competing project objectives and identifies favourable trade-offs among scheduling efficiency, financial expenditure, liquidity requirements and economic returns. The overall computational sequence is presented in Figure 4.

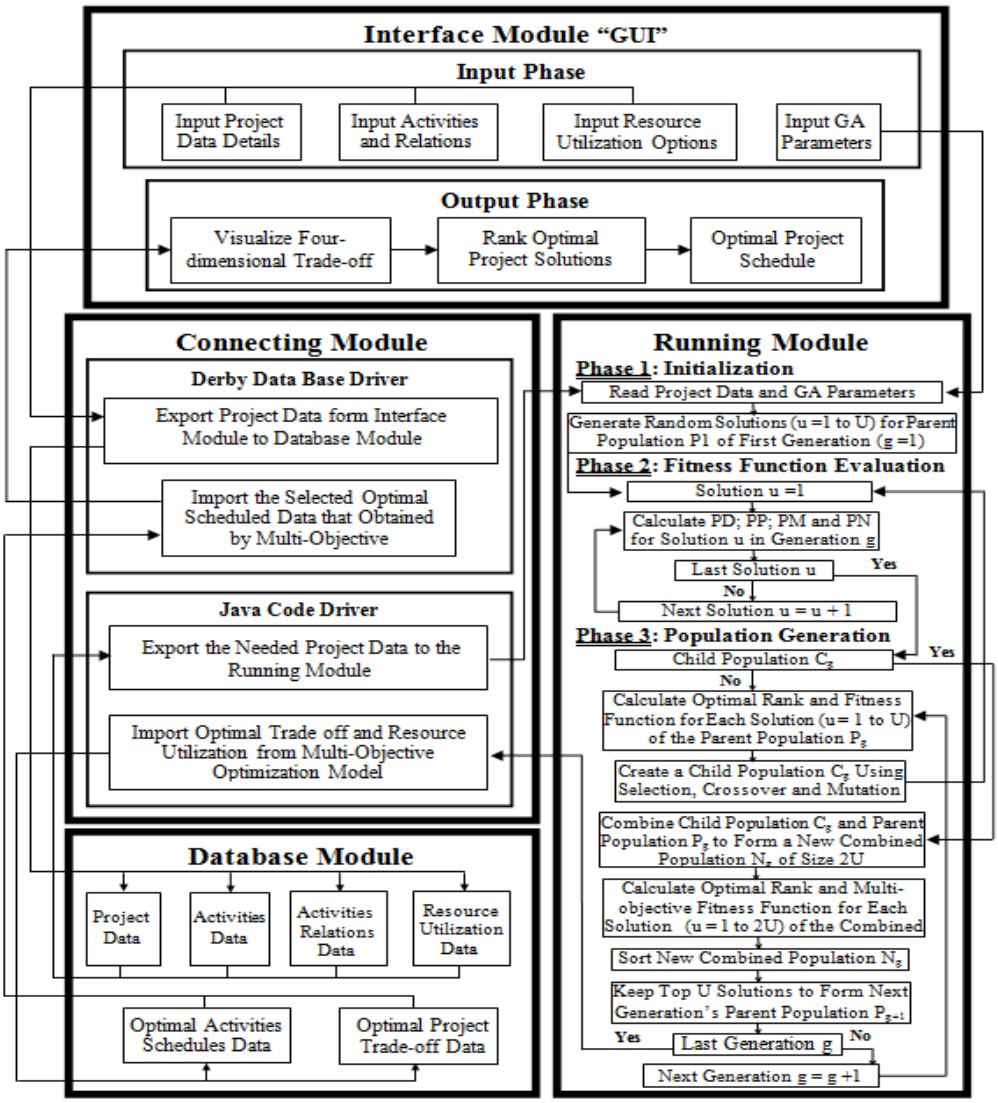


Fig.4: Key Modules of the OSS Framework

3.3.1 Phase 1: Initialization

The initialisation phase establishes the project-specific information and evolutionary parameters required to execute the proposed optimisation procedure. Its implementation comprises two

principal steps. The first involves defining the essential construction project characteristics and configuring the associated GA parameters. Project inputs include the overall scale of construction operations, logical precedence relationships among activities and the available resource execution alternatives for each activity. The evolutionary configuration incorporates chromosome string length, number of generations, population size, MP and CP. Chromosome length is determined according to the total number of project activities (K), providing the basis for specifying the generation count (G) and population size (U) required for the optimisation process. The crossover and mutation parameters are subsequently established with consideration of the selected population size and parent-selection procedure.

The second step generates the initial parent population (P1) for the first generation (g = 1), comprising U randomly constructed candidate solutions indexed from u = 1 to U. Each chromosome represents a distinct combination of resource execution modes assigned to the project's constituent activities. Collectively, these randomly generated configurations establish the initial solution space from which subsequent evolutionary operations identify and progressively refine promising alternatives. The resulting optimisation process seeks resource deployment strategies that reconcile four competing performance objectives: minimising total project duration, tender price and maximum working capital requirements while maximising NPV.

3.3.2 Phase 2: Fitness Functions Evaluation

The second phase evaluates the performance of each candidate solution (u) within the corresponding evolutionary generation (g) through the systematic determination of its fitness. This assessment requires the calculation of four interrelated project indicators: overall construction duration, proposed tender price, maximum working capital demand and anticipated NPV. Each indicator captures a distinct aspect of project performance, enabling alternative resource configurations to be assessed against both operational and financial objectives. The resulting fitness evaluation provides the quantitative basis for comparing candidate solutions and determining their relative suitability for selection, survival and reproduction during subsequent evolutionary iterations. Consistent with the methodological formulation of Aziz [5], the four performance measures are integrated within a multi-objective evaluation procedure that supports the identification of favourable execution alternatives. Through this process, the optimisation model establishes a consistent analytical basis for distinguishing comparatively efficient resource allocation strategies from less desirable configurations.

3.3.3 Phase 3: Population Generation

The third phase develops and manages three interconnected population groups during each evolutionary iteration: the parent population, offspring population and combined population. At generation (g), the existing parent population (Pg) undergoes evolutionary operations consistent with the GA principles [3], producing a corresponding offspring population (Cg). Genetic recombination and stochastic modification of parental chromosomes introduce alternative resource configurations, thereby expanding the exploration of the feasible solution space and maintaining diversity among candidate schedules. The parent and offspring populations are subsequently integrated into a combined population (Ng), allowing previously identified solutions to be evaluated alongside newly generated alternatives. This enlarged population provides the basis for comparative fitness assessment and the preservation of superior candidates. Following evaluation, the most favourable solutions are retained to constitute the parent population of the succeeding generation (P), thereby supporting progressive improvements in the quality of the available execution strategies.

The computational procedure begins by establishing the Pareto-optimal ranking of each candidate solution (u = 1 ……. U) within the parent population. Ranking is undertaken through simultaneous consideration of four optimisation objectives: maximisation of NPV and minimisation of project duration, tender price and maximum working capital requirements. Under the Pareto dominance principle, one solution dominates another when it performs no worse across all objectives and demonstrates a strict improvement in at least one criterion. This classification enables the identification of non-dominated alternatives without requiring a single solution to outperform every competing configuration across all performance measures. Following the dominance assessment, the multi-objective fitness value of each candidate is determined using the formulation presented in Equation 1, which combines normalised fitness components associated with the four optimisation criteria. This integrated evaluation facilitates comparative assessment of alternatives across the Pareto front while supporting the retention of diverse, high-performing solutions and reducing the likelihood of premature convergence.

$$Z^n = W1 * \frac{PD^n - PDmin}{PDmax - PDmin} + W2 * \frac{PP^n - PPmin}{PPmax - PPmin} + W3 * \frac{PM^n - PMmin}{PMmax - PMmin} + W4 * \frac{PNmin - PN^n}{PNmax - PNmin}$$

where Zn represents the multi-objective performance criterion associated with resource execution mode (n), incorporating project duration, tender price, maximum working capital requirements and NPV. The coefficients W1, W2, W3 and W4 denote the relative weights assigned to the four components of the fitness function, subject to the condition that their sum equals one. PDmin and PDmax represent the minimum and maximum project durations, respectively, among all evaluated solutions. Similarly, PPmin and PPmax indicate the lowest and highest project tender prices, while PMmin and PMmax define the minimum and maximum peak working capital requirements across the feasible solution set. Finally, PNmin and PNmax correspond to the minimum and maximum project NPV values obtained from the evaluated alternatives. These mathematical definitions and their associated formulations follow the methodology of Aziz [5].

Create a child population (Cg) using selection, crossover and mutation. The offspring population (Cg) is generated by applying the principal GA operators to the existing parent population, with Pareto-optimal rankings guiding the selection of candidate chromosomes. Individuals exhibiting superior rankings are assigned greater opportunities for reproduction. Selected chromosomes subsequently undergo crossover, during which pairs of parent solutions exchange encoded decision variables at randomly determined crossover positions, producing two offspring with modified genetic configurations. Mutation introduces additional diversity by randomly altering selected variables within individual chromosomes, thereby expanding the search space and reducing the likelihood of premature convergence towards local optima. These procedures follow the evolutionary principles [3]. Once generated, the offspring population is evaluated using the computational procedures described in Phase 2 to determine the corresponding project performance indicators, including cost, duration and quality.

Combine child population (Cg) and parent population (Pg) to form new combined population (Ng) of size 2U. The offspring population (Cg) is merged with the existing parent population (Pg) to establish a combined population (Ng) containing 2U candidate solutions. This integration implements an elitist preservation strategy, allowing high-performing solutions from the preceding generation to remain available for selection alongside newly generated alternatives. Consequently, favourable genetic configurations are protected against inadvertent elimination during successive evolutionary iterations.

Calculate optimal rank and fitness function for each solution (s = 1 to 2U) of the newly created

combined population (Ng). Every candidate solution (s = 1 to 2U) within the combined population (Ng) undergoes renewed Pareto-ranking and fitness evaluation. The computational procedure replicates the operations performed during the first step of this phase, ensuring that parent and offspring solutions are assessed consistently against the established multi-objective performance criteria.

Sort the new combined population (Ng) using "sorting comparison rule". The evaluated candidates are arranged according to the specified sorting comparison rule, which prioritises solutions exhibiting lower fitness values under the adopted minimisation-based formulation. Accordingly, comparatively favourable alternatives are positioned at the beginning of the ranked population and retained for subsequent selection.

Keep top U solutions from the combined population (Ng) to form the parent population (Pg+1) of the next generation. Following the ranking procedure, the highest-performing U solutions are selected from the combined population (Ng) to establish the parent population (Pg+1) for the succeeding generation. This updated population is subsequently returned to the first step of Phase 3, initiating another cycle of offspring generation and evaluation. The coordinated execution of Phases 2 and 3 continues until the predetermined number of generations has been completed.

The optimisation procedure initially determines the relative performance of each resource allocation alternative through the multi-objective fitness measure Zn. Consistent with the formulation of Aziz [5], this measure integrates four competing project objectives: duration, tender price, maximum working capital demand and NPV. Their respective contributions are controlled by the weighting coefficients W1–W4, whose combined value must equal unity. Because these indicators are expressed in different measurement units, normalisation is performed using their corresponding minimum and maximum values, including PDmin and PDmax for project duration. This transformation establishes a comparable numerical basis for evaluating alternative solutions. Following fitness assessment, a new offspring population (Cg) is produced through selection, crossover and mutation. Higher-ranked chromosomes receive preferential consideration during reproduction, while crossover recombines parental genetic information to construct alternative resource configurations. Mutation introduces random modifications that preserve population diversity and reduce susceptibility to local optima, consistent with the evolutionary framework [3].

An elitist population management strategy subsequently combines the existing parent population (Pg) with the generated offspring (Cg), producing an expanded population (Ng) comprising 2U alternatives. Each candidate undergoes renewed fitness evaluation and Pareto-optimal ranking, enabling consistent comparison between previously retained and newly generated solutions. The resulting population is organised through the prescribed sorting comparison rule, which places comparatively favourable fitness values at higher positions in the selection hierarchy. Only the leading U solutions are retained to constitute the parent population (P) for the next generation. Through repeated cycles of fitness evaluation, genetic recombination, mutation and elitist selection, the algorithm progressively refines resource allocation alternatives until the specified generational limit is reached, yielding the best-performing project configurations identified during the evolutionary search.

3.4 Optimization Strategy Software (Oss)

OSS is a Java-based decision-support application developed using the Eclipse Integrated Development Environment (IDE) to facilitate multi-objective optimisation in construction project planning. Its functionality is organised around three principal capabilities intended to strengthen the evaluation and selection of alternative project execution strategies. First, the software enables

planners to rank feasible project configurations through a weighted assessment framework incorporating project duration, tender price, maximum working capital requirements and NPV. User-defined weighting coefficients allow the relative importance of these objectives to be adjusted according to project-specific priorities. Second, an integrated visualisation environment represents the four-dimensional relationships among the performance indicators, supporting the interpretation of competing operational and financial outcomes. Third, OSS provides compatibility with established project management computational procedures, facilitating the application of optimisation results to practical scheduling, resource planning and project control.

3.5 System Architecture and Interface Design

The software architecture, shown in Figure 4, comprises four interconnected functional modules designed to coordinate user interaction, information management, computational processing and system integration. The User Interface (UI) Module provides the principal interaction environment and is organised into input and output components. Through the input component, planners specify essential project information, including activity precedence relationships, available resource alternatives and the computational parameters governing the GA. The output component presents the results generated by the optimisation procedure, enabling users to examine, compare and rank alternative resource deployment strategies and project schedules. Integrated visualisation functions further support the interpretation of multi-objective trade-offs and the identification of configurations consistent with the planner's operational and financial preferences.

3.6 Data Management and Storage

The Database Module provides the centralised information management infrastructure supporting the computational operations of OSS. Implemented through a relational database architecture using the Derby Database Management System, this component facilitates the structured storage, retrieval and exchange of project information. Its data organisation accommodates both initial inputs and processed optimisation outputs, including project calendars, non-working days, activity dependencies, resource availability constraints, optimal execution schedules and four-dimensional trade-off results. Integration with the Java environment enables information to be transferred between the database and computational components while maintaining organised records of project parameters and evaluated alternatives. This persistent storage arrangement supports subsequent retrieval, modification and reassessment of previously established project configurations.

3.7 Computational Logic and Connectivity

The computational infrastructure of OSS is established through the coordinated operation of the Running Module and Connecting Module. The Running Module functions as the principal optimisation engine, executing the Java classes responsible for construction scheduling calculations, financial performance evaluation and multi-objective fitness assessment. Its computational procedures determine project duration, working capital requirements and NPV for alternative resource configurations, thereby providing the quantitative information required for evolutionary optimisation. The Connecting Module facilitates communication among the system components through Java-based interfaces and Derby database drivers, ensuring the controlled transfer of information between the relational database, computational engine and user interaction environment.

This connectivity layer performs two vital functions:

3.8 Driver-Mediated Export/Import

The connecting infrastructure transfers activity information, resource characteristics and other relevant project inputs from the relational database to the computational engine. Following optimisation, the calculated schedules, financial indicators and selected resource configurations are transferred back to the database for organised storage and subsequent retrieval.

3.9 System Synchronization

The Connecting Module maintains consistency between the UI and the underlying relational database by mapping user-supplied project information to the corresponding database tables. It also coordinates the retrieval and presentation of processed results, ensuring that calculated optimisation outcomes are accurately reflected within the software's visualisation and reporting environment developed using Eclipse.

4. Performance Verification of Proposed Model OSS

The effectiveness of the proposed multi-objective optimisation framework is examined through its application to an illustrative construction project, with computational outcomes compared against the manually derived results reported by Aziz [5]. This comparative evaluation provides a basis for assessing the numerical accuracy of OSS and its ability to accommodate interdependent scheduling, resource allocation and financial parameters. The analysis further investigates the practical applicability of the framework in supporting contractors' tendering decisions for repetitive construction projects. By systematically evaluating alternative resource execution modes, the proposed approach seeks to identify economically competitive tender configurations while maintaining operational feasibility. Four competing objectives underpin this assessment: minimising overall project duration, tender price and maximum working capital requirements while maximising NPV.

The illustrative project, presented in Figure 5 and Table 2, comprises four logically interconnected construction activities distributed across ten identical repetitive stages. Multiple resource execution alternatives are available for each activity, allowing the model to examine different combinations of operational capacity and associated expenditure.

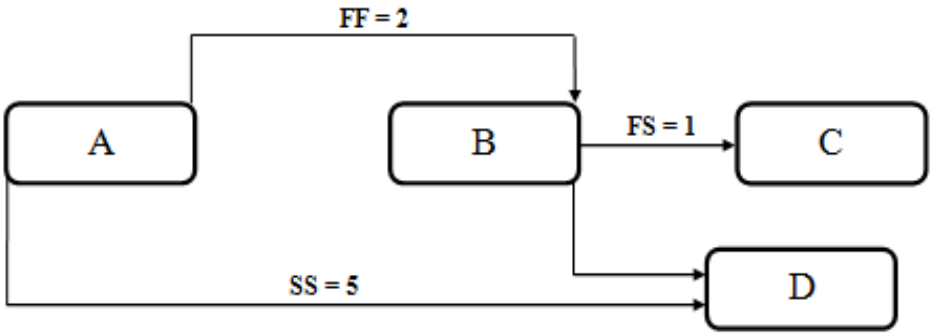


Fig.5: Systematic Case Study Network
Note: Case study network adapted from Aziz [5].

The principal input parameters, summarised in Table 2, include daily production rates, material expenditure, labour and equipment costs expressed in Egyptian pounds (EGP), and lump-sum subcontractor payments. Incorporating these variables enables the framework to represent the interaction between construction productivity, resource expenditure and project cash flow, thereby establishing a comprehensive basis for evaluating alternative scheduling and tendering strategies.

Table 2
Project Activity Data and Resource-Mode Configurations by Lifecycle Stage

Activity Name	Depends On	Relation Type	Lag Value	Stage Buffer	Resource Mode Options	Quantity (units)	Production Rate (units / day)	Material Cost (EGP)	Cost Rate (Labor & Equipment) (EGP / Day)	Subcontractor Lump Sum Cost (EGP)	Total Direct Cost (EGP)	Total Indirect Cost (EGP)	Total Cost (EGP)	Total Price (EGP)
A	--	--	--	--	**1**	1,200	600	4,000	400	350	5,150	150	5,300	6,000
					2		400	3,000	250	300	4,050	150	4,200	4,800
					3		300	2,000	150	250	2,850	350	3,200	3,600
B	**A**	**FF**	**2**	**1**	**1**	600	200	2,400	150	200	3,050	250	3,300	3,900
					2		120	1,900	100	200	2,600	400	3,000	3,500
C	**B**	**FS**	**1**	**0**	**1**	1,500	500	2,200	100		2,500	200	2,700	3,000
D	**A**	**SS**	**5**	**1**	**1**	2,000	500	3,700	350	200	5,300	300	5,600	6,000
	B	**FS**	**0**	**2**	**2**		400	2,800	300	150	4,450	350	4,800	5,000

Note: EGP = Egyptian pounds; FF = finish-to-finish; FS = finish-to-start; SS = start-to-start. Data adapted from Aziz [5].

Table 2 shows that the illustrative project comprises four activities (A–D) repeated over ten identical stages, and that the number of resource modes differs by activity: Activity A offers three modes, Activities B and D two modes each and Activity C a single mode, giving 3 × 2 × 1 × 2 = 12 possible mode combinations per stage. Dividing the stage quantity by the production rate indicates stage durations of 2, 3 and 6 days for Activity A (Modes 1–3), 3 and 5 days for Activity B (Modes 1–2), 3 days for Activity C and 4 and 5 days for Activity D (Modes 1–2). The table also exposes the time–cost trade-off embedded in the case: the fastest mode of Activity A is three times as productive as its slowest mode yet carries a total price of 6,000 EGP compared with 3,600 EGP, a premium of 66.7%. A comparable pattern appears for Activities B and D, whose faster modes are priced 11.4% (3,900 vs. 3,500 EGP) and 20.0% (6,000 vs. 5,000 EGP) higher, respectively. Because the precedence relations combine FF, FS and SS links with lags and stage buffers, the selected mode influences not only the duration of an activity but also the start times of its successors, which explains why the cheapest or fastest mode cannot simply be chosen independently for each activity.

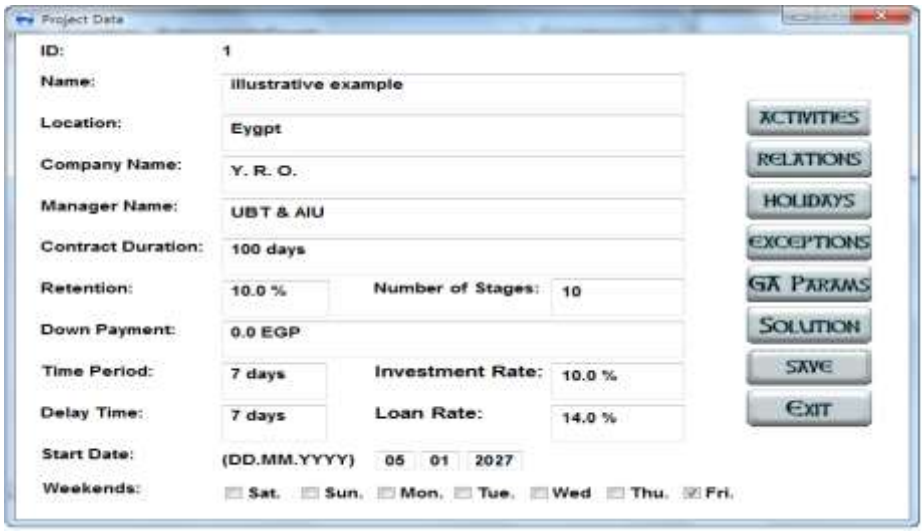


Fig.6: Formal Project Intake Form

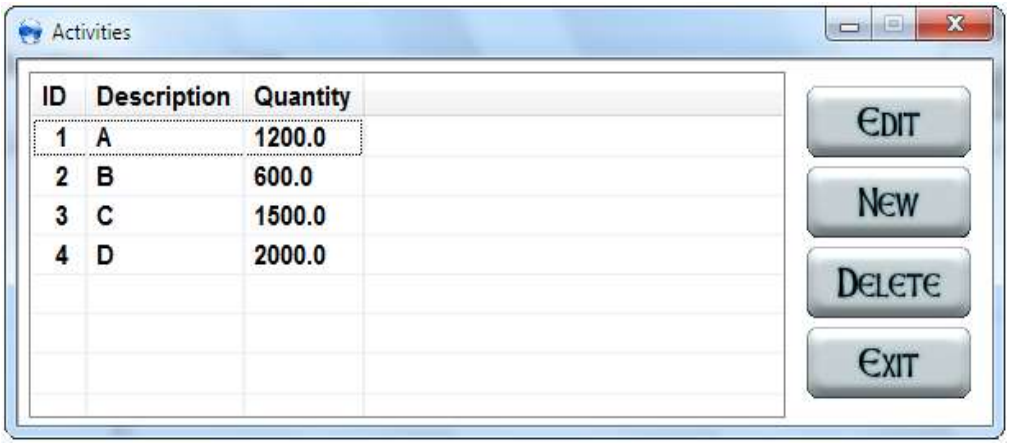


Fig.7: Project Activity Documentation Form

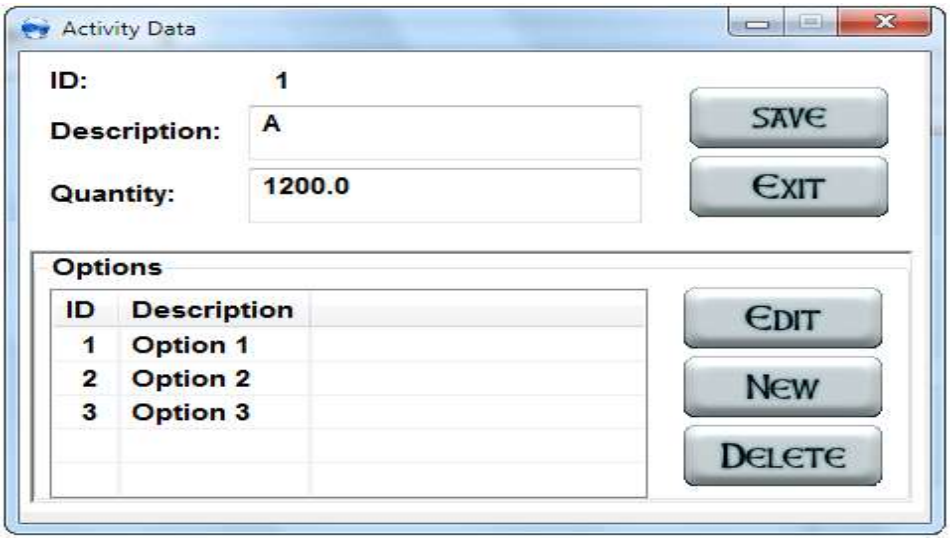


Fig.8: Activity Log & Description Form

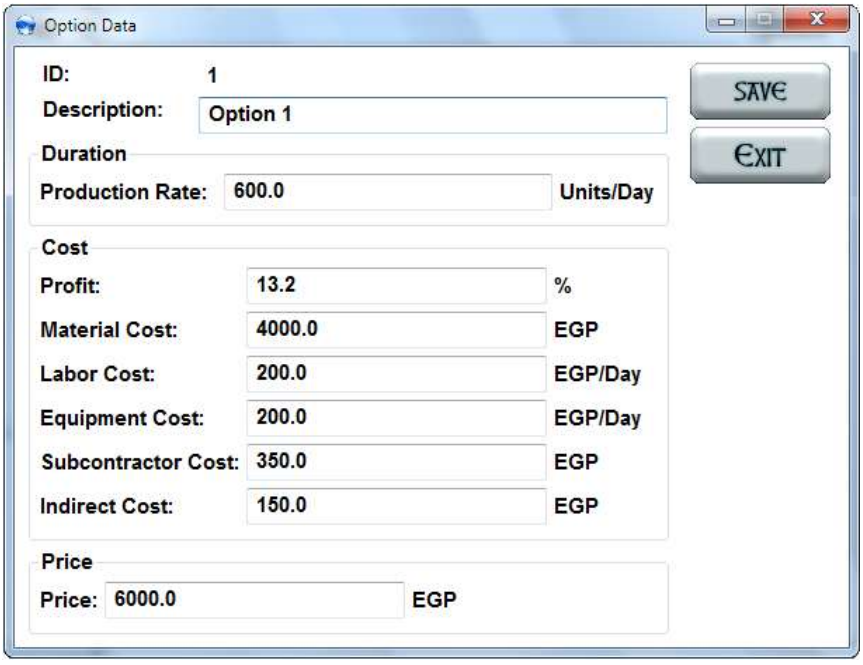


Fig.9: Project Resource Allocation and Utilization Form

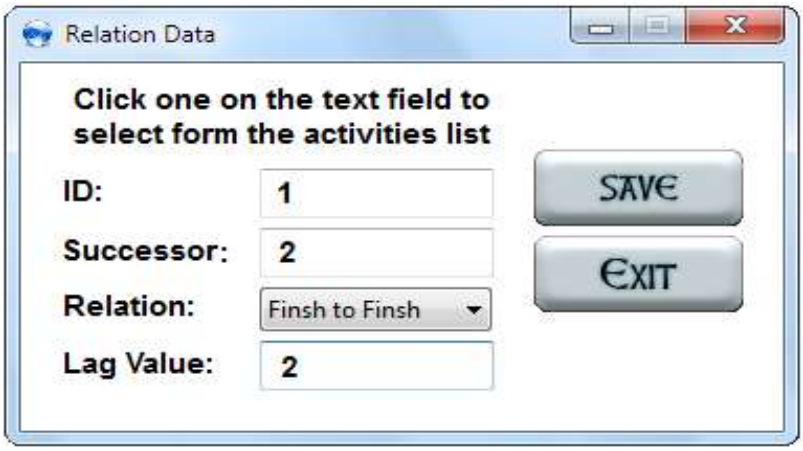


Fig.10: Activity Dependency Details Form

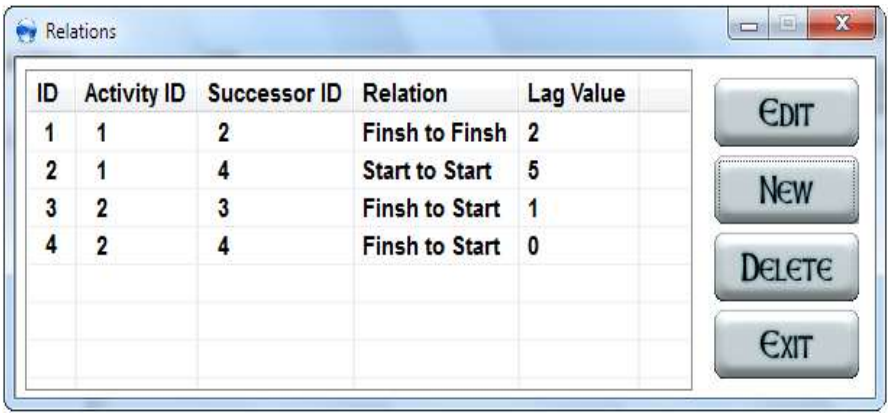


Fig.11: Task Relationship Specification Form



Fig.12: Project Holiday Schedule Form



Fig.13: Project Exceptions Schedule Form

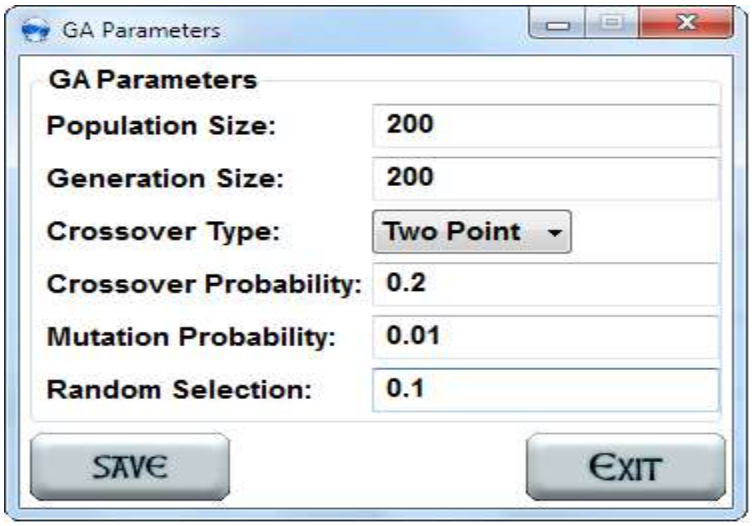


Fig.14: Genetic Algorithm Optimization Settings Form

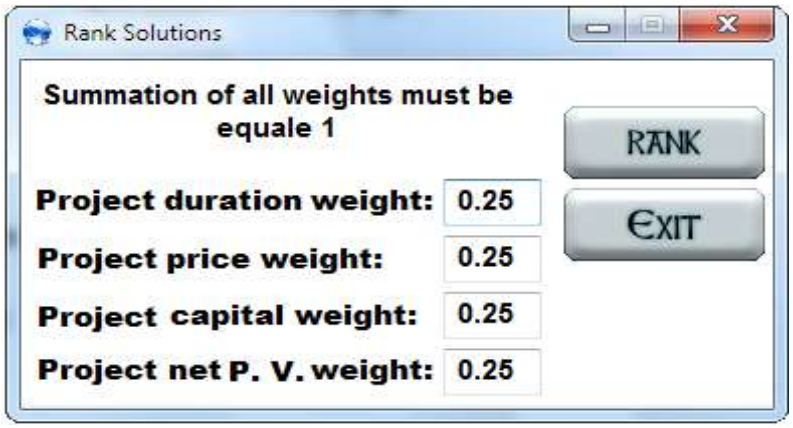


Fig.15: Project Solutions Ranking Form

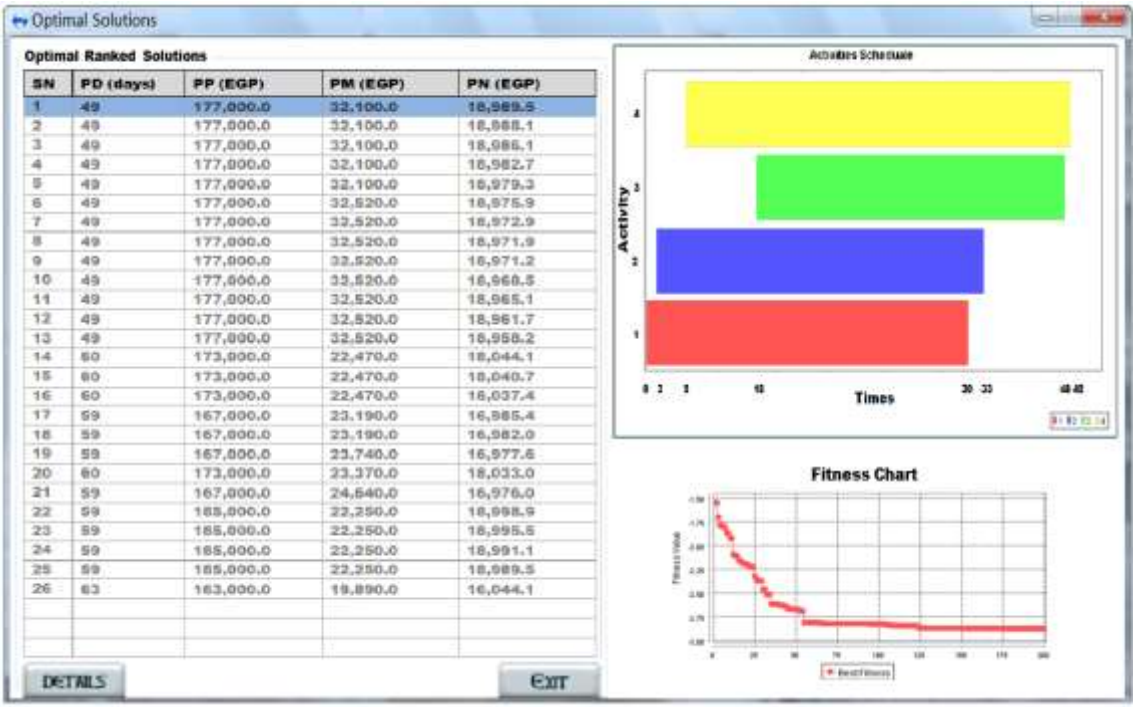


Fig.16: Optimization of Ranked Solutions Utilizing OSS Outputs

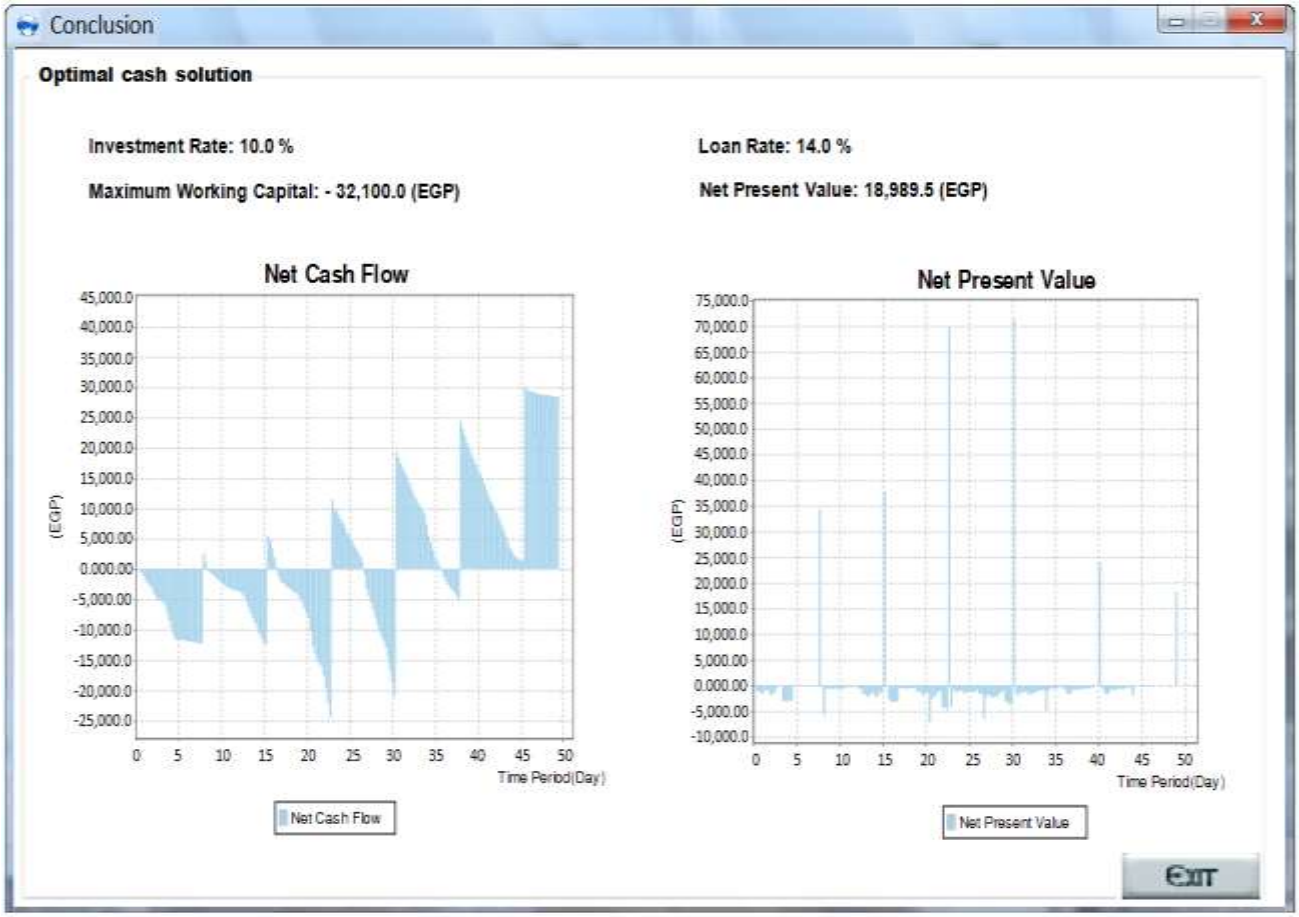


Fig.17: Cumulative Net Cash Flows and NPV Analysis of the Optimal Solution

The OSS input interfaces presented in Figures 6–14 enable users to specify the project characteristics and computational parameters required for optimisation: project identification (Figure 6), activity documentation and description (Figure 7 and Figure 8), resource allocation and utilisation (Figure 9), activity dependencies and task relationships (Figure 10 and Figure 11), calendar data comprising holidays and exceptions (Figure 12 and Figure 13) and the GA settings (Figure 14). As shown in Figure 14, the search was configured with a population of 200 chromosomes evolved over 200 generations, two-point crossover with a crossover probability of 0.2, a mutation probability of 0.01 and a random-selection rate of 0.1; the mutation probability lies within the low range (below 0.1) discussed in Section 3 and the crossover probability within the customary 0.1–1.0 range. The process wizard illustrated in Figure 15 supports the selection of operational scenarios, here an equal weighting of 0.25 for project duration, price, working capital and NPV. Subsequently, the output wizard shown in Figure 16 consolidates feasible execution alternatives generated from different combinations of resource modes across successive project stages, each evaluated according to four principal performance indicators: project duration, associated cost, maximum working capital requirements and NPV. The final output interface, presented in Figure 17, provides graphical representations of the net cash flow and NPV profiles corresponding to the identified optimal solution. The resulting comparative assessment is summarised in Table 3, which ranks the most favourable alternatives identified from 15 distinct evaluated scenarios according to their relative optimisation performance.

Table 3
Primary Resolution for All Defined Scenarios

ID	*Project Duration (Days)*	*Project Price (EGP)*	*Start time of Activity*				*Max. Working Capital*	*Net Present Value Investment=10% Loan =14%*	Remarks
			A	B	C	D			
01	48	189,000.0	0	2	18	8	34,860.0	19,956.3	W1 = 1.0
02	64	151,000.0	0	4	32	14	21,200.0	14,059.6	W2 = 1.0
03	63	163,000.0	0	3	32	13	19,890.0	16,044.1	W3 = 1.0
04	48	189,000.0	0	2	18	8	34,860.0	19,956.3	W4 = 1.0
05	49	177,000.0	0	3	18	9	32,100.0	18,989.5	W1 & W2 = 1/2
06	49	177,000.0	0	3	18	9	32,100.0	18,989.5	W1 & W3 = 1/2
07	48	189,000.0	0	2	18	8	34,860.0	19,956.3	W1 & W4 = 1/2
08	64	151,000.0	0	4	32	14	21,200.0	14,059.6	W2 & W3 = 1/2
09	49	177,000.0	0	3	18	9	32,100.0	18,989.5	W2 & W4 = 1/2
10	59	185,000.0	0	2	29	19	22,250.0	18,998.9	W3 & W4 = 1/2
11	64	151,000.0	0	4	32	14	21,200.0	14,059.6	W1 & W2 & W3 = 1/3
12	49	177,000.0	0	3	18	9	32,100.0	18,989.5	W1 & W2 & W4 = 1/3
13	48	189,000.0	0	2	18	8	34,860.0	19,956.3	W1 & W3 & W4 = 1/3
14	63	163,000.0	0	3	32	13	19,890.0	16,044.1	W2 & W3 & W4 = 1/3
15	49	177,000.0	0	3	18	9	32,100.0	18,989.5	W1 & W2 & W3 & W4 = 1/4

Note: W1 = project duration weight; W2 = tender price weight; W3 = maximum working capital weight; W4 = net present value weight. Start times are in days; price, maximum working capital and net present value are in EGP. Net present value was computed at an investment rate of 10% and a loan rate of 14%.

Interpretation of the Optimisation Results

Table 3 summarises the outcome of the 15 weighting scenarios, which correspond to every non-empty combination of the four objective weights (W1 = project duration, W2 = tender price, W3 = maximum working capital and W4 = NPV). Although 15 scenarios were evaluated, they converge on only five distinct solutions: (a) 48 days, 189,000 EGP, 34,860 EGP of working capital and an NPV of 19,956.3 EGP (Scenarios 01, 04, 07 and 13); (b) 64 days, 151,000 EGP, 21,200 EGP and 14,059.6 EGP (Scenarios 02, 08 and 11); (c) 63 days, 163,000 EGP, 19,890 EGP and 16,044.1 EGP (Scenarios 03 and 14); (d) 49 days, 177,000 EGP, 32,100 EGP and 18,989.5 EGP (Scenarios 05, 06, 09, 12 and 15); and (e) 59 days, 185,000 EGP, 22,250 EGP and 18,998.9 EGP (Scenario 10). This convergence indicates that the objectives are not independent. Whenever duration or NPV carries weight, the optimiser favours the fast, high-price configuration, whereas tender price and working capital pull the solution towards slower, cheaper configurations. Because two of the three objectives in Scenario 11 (price and working capital) are minimised by the slow configuration, their combined weight outweighs the duration weight and the solution shifts to 64 days.

The extreme solutions quantify the trade-off. Relative to the fastest solution (a), the cheapest solution (b) lengthens the project by 16 days (33.3%) but reduces the tender price by 38,000 EGP (20.1%), the peak working capital by 13,660 EGP (39.2%) and the NPV by 5,896.7 EGP (29.5%). The minimum-working-capital solution (c) requires 19,890 EGP of peak working capital, 42.9% below solution (a), at the expense of 15 additional days (31.3%) and a 19.6% lower NPV. The compromise solution (d), obtained when all four weights are equal (Scenario 15) and ranked first in Figure 16, is only 1 day (2.1%) longer than the fastest solution while lowering the price by 12,000 EGP (6.3%) and the peak working capital by 2,760 EGP (7.9%), at an NPV reduction of 966.8 EGP (4.8%). Solution (d) therefore retains most of the NPV of the fastest plan while shedding part of its price and financing burden, which explains its selection under balanced preferences. Solution (e) attains the highest NPV among the non-fastest options (18,998.9 EGP) with a peak working capital 30.7% lower than solution (d) (22,250 vs. 32,100 EGP), although it requires 10 additional days and a 4.5% higher price (185,000 vs. 177,000 EGP).

The start-time columns of Table 3 show how these trade-offs are realised. Activity A always starts on day 0, whereas the start of Activity B ranges from day 2 to day 4, that of Activity C from day 18 to day 32 and that of Activity D from day 8 to day 19. Activity C is deferred by up to 14 days (day 32 vs. day 18) in the low-price and low-working-capital solutions, and Activity D by up to 11 days (day 19 vs. day 8) in solution (e). This pattern suggests that the optimiser exploits available float to postpone expenditure and reduce the cumulative outflow incurred before interim payments are received, thereby lowering peak working capital. Such behaviour is consistent with finance-based scheduling, in which shifting activities within their float improves cash-flow performance without altering the precedence logic [15].

Figure 16 shows that OSS ranks 26 candidate solutions under the equal-weight setting of Figure 15. The 13 highest-ranked solutions all complete in 49 days at a price of 177,000 EGP, with peak working capital of 32,100–32,520 EGP and NPVs between 18,958.2 and 18,989.5 EGP. The optimiser therefore retains a dense cluster of near-equivalent alternatives rather than a single point solution, which gives planners flexibility when site conditions favour a different start-time pattern. The lower-ranked solutions (59–63 days) trade longer durations for lower prices (163,000–185,000 EGP) and lower peak working capital (19,890–24,640 EGP), with NPVs between 16,044.1 and 18,998.9 EGP. The fitness chart in Figure 16 shows a rapid decline of the best fitness value during roughly the first 55 generations (from about −1.5 to about −2.7), a much slower improvement until approximately generation 125 and a plateau thereafter, indicating that the 200 generations specified in Figure 14 were sufficient for convergence under the minimisation-based formulation.

Figure 17 characterises the financial profile of the selected 49-day solution at an investment rate of 10% and a loan rate of 14%, with a maximum working capital of 32,100 EGP and an NPV of 18,989.5 EGP. The net cash flow chart alternates between deficit phases and positive pulses that coincide with the delayed interim payments; the deficits reach roughly −12,000 EGP before the first payment and approach −25,000 EGP around day 22, and the profile closes with a sustained inflow of about 29,000 EGP at the end of the project, consistent with the immediate release of the accumulated retention upon handover assumed in Section 3.2. The NPV chart indicates that the discounted value is dominated by a small number of payment events, namely the periodic interim payments and the final retention release, whereas the intervening days contribute small negative values from continuing expenditure.

5. Implications and Limitations

These findings agree with the wider literature showing that the objectives of repetitive projects conflict and cannot be reconciled by a single solution: multi-objective GA and NSGA-II models likewise yield a spread of Pareto-optimal alternatives for repetitive and highway projects [2; 6], and resource-focused formulations confirm that the need for formal optimisation grows with project size and complexity [9]. The distinctive contribution of OSS is that financial indicators (tender price, working capital and NPV) are optimised alongside duration, whereas many recent models concentrate on time–cost pairs or on resource levelling [7; 8]. In practice, a contractor could adopt solution (b) to maximise price competitiveness when financing is ample, solution (c) when liquidity is scarce, solution (a) when delay penalties or early-completion incentives dominate, or solution (d) as a balanced default.

Several limitations qualify these results. The case study is a single deterministic example comprising four activities and ten stages, with constant material prices, immediate subcontractor payments and no learning effects (Section 3.2); the results therefore illustrate the behaviour of OSS rather than support statistical generalisation. Future work could incorporate stochastic productivity [13], exploit BIM-derived quantities [16; 17], employ learning-based search [19; 20] and use language models to translate planner preferences into objective weights [21].

6. Conclusion

This study develops and implements OSS as an integrated multi-objective optimisation framework for identifying economically advantageous tendering strategies in repetitive construction projects. Through the systematic examination of alternative resource deployment modes across constituent activities, the proposed model seeks to minimise project duration, tender price and maximum working capital requirements while simultaneously maximising NPV. The software architecture comprises four interconnected functional modules. The user interface facilitates project data entry and the visualisation of four-dimensional trade-offs among competing objectives, whereas the relational database supports the systematic storage, retrieval and management of scheduling information, resource utilisation parameters and optimal execution alternatives. The computational execution module performs the iterative calculations and optimisation procedures required to evaluate feasible project configurations, while the Java-based connecting module coordinates information exchange and maintains consistency among the system components. The operational capabilities of OSS address two principal planning requirements: optimising resource allocation and scheduling decisions throughout repetitive construction operations, and determining favourable activity commencement times by exploiting available total float to improve overall schedule efficiency. Together, these functions enable the systematic identification of advantageous execution strategies within complex, multidimensional solution spaces. The proposed mathematical formulations and computational procedures are subsequently examined through an applied case study, providing a practical basis for assessing the framework's effectiveness in supporting competitive tender preparation and informed construction planning decisions.

Quantitatively, the 15 weighting scenarios evaluated in the case study converged on five distinct solutions. The fastest plan (48 days, 189,000 EGP) and the cheapest plan (64 days, 151,000 EGP) differ by 16 days, 20.1% in tender price, 39.2% in peak working capital and 29.5% in NPV, whereas the balanced 49-day solution (177,000 EGP, 32,100 EGP peak working capital, NPV of 18,989.5 EGP) lies only 1 day and 4.8% of NPV away from the fastest plan while reducing price and peak working capital by 6.3% and 7.9%. These results confirm that OSS can expose the trade-offs among duration, price, liquidity and profitability, and can rank alternatives according to user-defined priorities. Future research should address stochastic productivity and payment behaviour [13], integrate BIM-derived data [17] and examine learning-based and language-model-assisted decision support [19; 21].

7. ABBREVIATIONS

For convenience of discussion, the following abbreviations are used without special explanations:

ACO:	**A**nt **C**olony **O**ptimization;
AI:	**A**rtificial **I**ntelligence;
ANN:	**A**rtificial **N**eural **N**etworks;
COPS:	**C**onstruction **O**peration and **P**roject **S**cheduling;
CP:	**C**rossover **P**robability;
CPM:	**C**ritical **P**ath **M**ethod;
DP:	**D**ynamic **P**rogramming;
GAs:	**G**enetic **A**lgorithm**s**;
GWP:	**G**lobal **W**arming **P**otential;
HM:	**H**euristic **M**ethods;
KBES:	**K**nowledge **B**ased **E**xpert **S**ystems;
LOB:	**L**ine **O**f **B**alance;
LP:	**L**inear **P**rogramming;
MAs:	**M**emetic **A**lgorithm**s**;
MM:	**M**athematical **M**ethods;
MP:	**M**utation **P**robability;
OSS:	**O**ptimizing **S**trategy **S**oftware;
PD:	**P**roject **D**uration;
PDM:	**P**recedence **D**iagram **M**ethod;

PERT:	**P**rogram **E**valuation and **R**eview **T**echnique;
PM:	**P**roject **M**aximum working capital;
PN:	**P**roject **N**et present value;
PP:	**P**roject/bid **P**rice;
PSO:	**P**article **S**warm **O**ptimization;
RCMPSP:	**R**esource **C**onstrained **M**ulti-**P**roject **S**cheduling **P**roblem;
SFL:	**S**huffled **F**rog **L**eaping Algorithms.
TCEI:	**T**ime, **C**ost, and **E**nvironment **I**mpact;
VPM:	**V**ertical **P**roduction **M**ethod.

References

[1] Nguyen, D. A., Nguyen, N.-T., Tran, Q., & Tran, D.-H. (2023). Tradeoff different construction project goals in using a novel multi-objective sea horse algorithm. *Alexandria Engineering Journal*, *82*, 55-68. https://doi.org/10.1016/j.aej.2023.09.059

[2] Kumar, K. M., Agrawal, D., Vishwakarma, V. K., & Eirgash, M. A. (2024). Development of time-cost trade-off optimization model for indian highway construction projects using non-dominated sorting genetic algorithm-II methodology. *Asian Journal of Civil Engineering*, *25*(8), 5975-5988. https://doi.org/10.1007/s42107-024-01157-y

[3] Alhijawi, B., & Awajan, A. (2024). Genetic algorithms: Theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, *17*(3), 1245-1256. https://doi.org/10.1007/s12065-023-00822-6

[4] Ghoroqi, M., Ghoddousi, P., Makui, A., Shirzadi Javid, A. A., & Talebi, S. (2024). Integration of resource supply management and scheduling of construction projects using multi-objective whale optimization algorithm and NSGA-II. *Soft Computing*, *28*(11-12), 6983-7001. https://doi.org/10.1007/s00500-023-09467-0

[5] Aziz, R. F. (2013). Optimizing strategy for repetitive construction projects within multi-mode resources. *Alexandria Engineering Journal*, *52*(1), 67-81. https://doi.org/10.1016/j.aej.2012.11.003

[6] Abdelbasset, E., Hassan, A., Amer, N. H., & Khalifa, Y. A. (2023). Optimizing the number of crews working in parallel and their work sequence to minimize project duration and cost for repetitive construction projects. *Innovative Infrastructure Solutions*, *8*(11), 288. https://doi.org/10.1007/s41062-023-01233-3

[7] Altuwaim, A., Alagha, E., Mahmoud, A. B., & Saad, D. A. (2024). Adaptive LOB scheduling for optimizing resource leveling and consumption. *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT*, *30*(3), 220-233. https://doi.org/10.3846/jcem.2024.20952

[8] Iranagh, M., Sonmez, R., Atan, T., Uysal, F., & Bettemir, Ö. H. (2023). A memetic algorithm for the solution of the resource leveling problem. *Buildings*, *13*(11), 2738. https://doi.org/10.3390/buildings13112738

[9] Lazari, V., Chassiakos, A., & Karatzas, S. (2024). Multi-objective resource-constrained scheduling in large and repetitive construction projects. *Algorithms*, *17*(8), 351. https://doi.org/10.3390/a17080351

[10] Jia, T. (2025). Optimizing construction project plan management using parameter-adaptive improved genetic algorithm. *Tehnički vjesnik*, *32*(1), 98-106. https://doi.org/10.17559/TV-20240720001868

[11] Peng, J., Feng, Y., Zhang, Q., & Liu, X. (2023). Multi-objective integrated optimization study of prefabricated building projects introducing sustainable levels. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-29881-6

[12] Yang, C. (2024). Construction scheduling optimization in the prefabricated buildings through

multi-objective GA on the principle of sustainable development. *Archives of Civil Engineering*, *70*(3), 459-472. https://doi.org/10.24425/ace.2024.150995

[13] Hassan, A., El-Rayes, K., & Attalla, M. (2023). Stochastic scheduling optimization of repetitive construction projects to minimize project duration and cost. *International Journal of Construction Management*, *23*(9), 1447-1456. https://doi.org/10.1080/15623599.2021.1975078

[14] Albayrak, G. (2024). Metaheuristic multi-objective optimization approach for repetitive construction project scheduling. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, *11*(24), 438-454. https://doi.org/10.54365/adyumbd.1477358

[15] Hasan, M. F., & Mohammed, S. R. (2022). Time and finance optimization model for multiple construction projects using genetic algorithm. *Journal of the Mechanical Behavior of Materials*, *31*(1), 267-281. https://doi.org/10.1515/jmbm-2022-0032

[16] Wu, Z., & Ma, G. (2023). Automatic generation of BIM-based construction schedule: Combining an ontology constraint rule and a genetic algorithm. *Engineering, Construction and Architectural Management*, *30*(10), 5253-5279. https://doi.org/10.1108/ecam-12-2021-1105

[17] Essam, N., Khodeir, L., & Fathy, F. (2023). Approaches for BIM-based multi-objective optimization in construction scheduling. *Ain Shams Engineering Journal*, *14*(6), 102114. https://doi.org/10.1016/j.asej.2023.102114

[18] Zhang, Y. (2025). Development of an integrated intelligent BIM-based model for multi-objective optimization in engineering assembly processes. *PLOS One*, *20*(11), e0333354. https://doi.org/10.1371/journal.pone.0333354

[19] Yao, Y., Tam, V. W. Y., Wang, J., Le, K. N., & Butera, A. (2024). Automated construction scheduling using deep reinforcement learning with valid action sampling. *Automation in Construction*, *166*, 105622. https://doi.org/10.1016/j.autcon.2024.105622

[20] Soleymani, M., Bonyani, M., & Wang, C. (2024). Simulation of autonomous resource allocation through deep reinforcement learning-based portfolio-project integration. *Automation in Construction*, *162*, 105381. https://doi.org/10.1016/j.autcon.2024.105381

[21] Prieto, S. A., Mengiste, E. T., & García de Soto, B. (2023). Investigating the use of ChatGPT for the scheduling of construction projects. *Buildings*, *13*(4), 857. https://doi.org/10.3390/buildings13040857

[22] Al-Refaie, A., Al-Hawadi, A., Lepkova, N., & Abbasi, G. (2023). Blockchain of optimal multiple construction projects planning under probabilistic arrival and stochastic durations. *Journal of Civil Engineering and Management*, *29*(1), 15-34. https://doi.org/10.3846/jcem.2023.17927