Journal of Kerbala University, Vol. 23, Issue 1, March , 2026

Available at https://www.iasj.net/iasj
Iraqi Academic Scientific Journals
Journal homepage:
https://journals.uokerbala.edu.iq/index.php/UOKJ

**Research Article**

# A Multi-Objective Programming Approach Using Artificial Intelligence to Enhance Urban Development Strategies in Iraqi Cities

**Meeras Salman Al-Shemarry**
**https://orcid.org/0000-0003-2859-9441**
**Department of Information Technology, College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq**

**Article Info**
Article history:
Received 19 -11-2025
Received in revised form 15-12-2025
Accepted 29-1-2026
Available online 31 -3 -2026
**Keywords: -** Urban Development, Financial Stability, Optimization Algorithms, Artificial Intelligence Decision-Making Models, Iraqi Cities.

## Abstract:

In this study, we suggest an enhanced and systematic model for achieving sustainability in the redevelopment projects of Iraqi cities. Through complex decision-making tools such as Dijkstra’s algorithm, A* search, and linear integer programming models. The research project aims to perfect the most effective ways to invest in urban infrastructure. The results suggest the approach to combine cost functions and evaluation heuristics is viable, generating solutions that are capable of serving dynamic urban requirements while respecting time and finance limitations. Nowadays, the best financial distribution is considered a problem in urban development projects, particularly those for post-conflict Iraqi cities, and answers are not always sufficient due to budget or time constraints. This paper provides a results-oriented, cost-effective decision-making method by integrating traditional optimisation and AI-driven search algorithms. This includes that of Dijkstra's algorithm, A* search, and hill-climbing heuristics involving linear integer programming for providing quality-wise candidate project prioritization and scheduling suggestions. The models developed are verified with otherwise entirely known, researcher-generated data derived from artificial scenarios that model realistic city planning scenarios (although these costs, implementation times , and delay penalties are up to the discretion of the researchers designing each synthetic scenario). The data is for proof-of-concept and decision support, not empirical generalisations at scale. The results indicate that the use of math-based cost models with heuristic assessment schedules can effectively reduce total project costs in order to minimise delay penalty and increase screening efficiency as compared with traditional scheduling methods. The lab presented is a tangible proof-of-concept to enhance the resilience and transparency of urban development processes in environments where resources are scarce.

**Corresponding Author E-mail:** Email: meeras.s@uokerbala.edu.iq

## 1. Introduction

Several recent works have already presented optimization-based algorithms for the solution of some problems arising from urban infrastructure. Wang et al. [1] considered multi-objective path planning in logistics, and they combined A* and genetic algorithms; Li et al. [2] developed a tool to solve the parking assignment problem based on NSGA-II. Hussein et al. [3] put forward the model of urban facility location with rough intervals; Pavon et al. [4] presented a hybrid approach that combines Minimum Spanning Tree (MST) with Mixed-Integer Linear Programming (MILP) for network design. Collectively, these works have stressed the importance of flexible, low-cost, and computationally realistic urban planning frameworks in post-conflict areas such as Iraq. Current conflicts and high population increase, a dilapidated economy, and a detached administration pose major challenges to urban development in Iraq. These factors have had tremendous consequences on the urban infrastructure in most of the country, including housing construction and development, transportation networks, and public facilities. City planning is among the most crucial things Iraq direly needs while it tries to rebuild its urban landscapes following years of war and turmoil. It is the objective of this paper to propose a cost-effective decision-making model for the use of objective programming and artificial intelligence with the aim of producing efficient solutions for urban planning and project prioritisation. Despite the literature reviewing the potential of optimisation and AI techniques for urban planning, a number of shortcomings can be observed. Several related works focus on specific problem components (such as spatial optimisation or routing) and omit the project execution order and the cost penalties for delay. In other research, simple static planning assumptions are made, or the performance of the algorithm is improved without a system that makes cost, time, and schedule integrated decision support available. Therefore, there is a demand for grounded, cost-aware models that can promote a transparent and implementable project prioritisation in capital-deprived post-conflict cities. This gap motivated the subsequent research, which presents an integrated multi-objective framework that includes mathematical cost modelling, heuristic search methods, and linear integer programming to support feasible development planning in urban areas. The primary contributions of this work are:

1.An inexpensive multiobjective planning scheme for urban expansion.
2. Classical and heuristic algorithms ( Dijkstra, Linear Programming, hill climbing, and A* ).
3.Use of the model in Iraqi cities after the war.
4.Demonstrate the potential by a practical case study.
5.Comparison of the proposed approach with other approaches (in terms of performance and generalisation).

## 2. Research Problem and Objective

This paper discusses the illogical and ineffective financing of urban interests in Iraq. It is not uncommon that the government and local authorities experience the challenge in deciding on (the right) project(s) to apply with their funding, time, and resources... once time pressures come into play! This paper aims to propose a model using objective programming, namely linear integer programming and pathfinding algorithms, to enhance the total performance of resource distribution. The principal objectives include:

1. Creation of a protocol to automate the selection of urban development projects for people, considering both their cost and time.

2. A novel hybrid search scheme by combining two major path-selection schemes, namely Dijkstra's algorithm and A* searching , further enhanced by Hill-Climbing for local improvement of time scheduling, is proposed.
3.Advantages of objective optimization for the urban planning problem.

That part is not far off from the goal of this research, since it inscribed readily observable pitfalls in current urban project planning, which our imagined epoch aimed to heal.

## 3. Literature Review

Many works have studied computational modeling of urban planning and infrastructure optimization. This attention to institutional form in both traditional and modernizing social arrangements, and the position of women within them, was developed by Turner [5], exploring patterns of settlement and housing preferences. Waddell et al. G [6] recommend using UrbanSim, which is a microsimulation platform designed to simulate land use development and location choices in response to changes in policies. Wilson et al. [7] designed a geospatial model for analyzing and mapping the urbanization process based on remote sensing with an emphasis on the spatial pattern of growth. Calabrò, D’Acierno, and Montella [8] developed a model to optimally design feeder bus lines within an urban area thanks to the integration of simulation and optimization approaches. The method was developed to be demand modelled and optimized, making the design routing more efficient than via traditional single objective methodologies. The purpose of this was as much to improve service and reduce costs. Ahmadi et al. [9] in the form of a bi-objective A* that simultaneously computes two objectives by querying paths from both source and target. The new method was much faster than those that had been applied previously, and it assisted in finding improved Pareto paths. Newer techniques also employ multi-objective optimisation. Wang et al. [1] provided a combined A* and Genetic Algorithm-based approach for urban path planning that is able to adapt dynamically to traffic and logistical requirements, but is complex and requires large amounts of input data. Li et al. [2] used NSGA-II in optimising parking locations across distance, cost, and user coverage. Still, it lacks dynamism to model, and the authors cannot find an optimised solution at runtime. Hussein et al. [3] utilised multi-objective location planning with rough interval methods and entropy-weighted criteria for uncertain cases, which, however, causes scalability problems. Finally, Pavon et al. [4] also adopted MST and MILP methodology for the design of urban facility networks, taking into consideration short distance as well as smooth connectivity; however, it considers simple descriptions regarding socio-economic diversity as well as qualitative issues. Combined, these works highlight the range of computational methods and are paving the way for our proposed constraint-aware optimisation framework tailored to Iraqi urban development. In order to give a wider picture of our proposed approach, we compare the method with a variety of six of the most recent urban studies focusing on optimisation of planning issues using various types of computational/nature-inspired and AI methods.

While prior work has given insights into the application of optimization and AI methods to urban planning, a number of mutual limitations can be found. Many of these focus on specific goals (e.g., minimizing the material use, time used for drilling or disruption to a community), not looking at the overall costs related to both project sequencing and delay-based financial penalties. Furthermore, some of the studies are based on static planning assumptions, which makes them less applicable to dynamic and restricted urban environments. Implementation issues, such as scalability, transparency of decision factors and their relative weight, or the logistical feasibility in post-conflict situations or resource-limited settings, are generally weakly considered. Such frequent drawbacks highlight the necessity of integrative trade-off-based frameworks that can take into account costs and balance several planning goals collectively under a hierarchical decision support concept. The main strengths and limitations of those works are listed in Table 1.

The literature survey and also the empirical evidence show some progress, but also the limitations of urban optimization approaches so far, thus motivating an overall general and flexible model like the one presented in this paper. Studies from the literature review reveal progress and stagnation in relation to optimization methods for urban areas, as well as why a comprehensive and flexible model is required (such as that proposed here).

**Table 1:** The comparison results among recent studies that focus on urban optimization methods.

| **Study** | **Methodology** | **Strengths** | **Weaknesses** | **Accuracy/Performance** |
|---|---|---|---|---|
| Wang et al. (2025) [1] | Hybrid A* with Genetic Algorithm for multi-objective path planning | Dynamic weighting, robust under varied scenarios | Computationally intensive, requires detailed traffic data | Pareto convergence >20% better than baseline methods |
| Li et al. (2025) [2] | NSGA-II for smart parking site selection (distance, cost, coverage) | Effective Pareto filtering, marginal utility modeling | Limited to static traffic assumptions | Improved 3D Pareto coverage and cost efficiency |
| Hussein et al. (2024) [3] | MOLP + Rough Interval + WSM and Entropy-based evaluation | Handles uncertainty, rich multi-factor evaluation | Complex for large-scale implementation | The Sen method showed high reliability and better ranking accuracy |
| Pavon et al. (2024) [4] | Greedy + MST + MILP for urban network design | Simple and modular, it balances connectivity | Ignores qualitative factors like social diversity | Reduced total network cost by ~15% |
| Calabrò et al. (2023) [8] | Multi-objective simulation + NSGA + CMA-ES for MRT | High Pareto precision, dynamic flow adaptation | High computational requirement | Improved service level by 25% |
| Ahmadi et al. (2021) [9] | BOBA*: Bidirectional A* for dual objectives | 5× faster, ideal for test networks | Not directly validated on real urban data | Solved 1,000 test cases with high optimality |

## 4. Methodology

Heuristic compute - multi-obj this paper. From the multi-objective urban planning optimisation perspective, it is an involved setting problem for urban planning [10–12]. The two methods also make their contribution at different layers of the proposed formation. Fig. 1 shows the structure of the developed framework as a set of procedures for processing input project data, and through mathematical models, optimization algorithms, and assessment members, gives an achievable solution output. The research methodology of Smart Mechanism for urban projects consists of two: mathematical modelling and methods of integration.

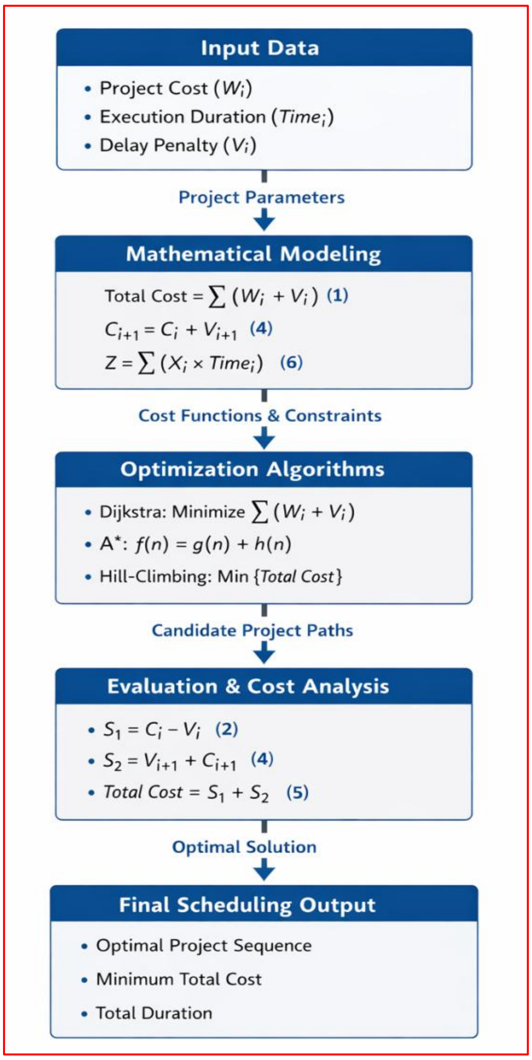


Figure 1. Workflow of the urban project scheduling framework discussed above, with its sequence of data inputs and outputs that begins with data preparation, leading to mathematical model creation and optimisation algorithm implementation, ending in the decision-making of schedules.

The urban project scheduling model is based on a rational algorithmic strategy, including classical and heuristic optimisation methods. The method first builds a planning graph over the project data in terms of cost, time, and delay penalty. Lastly, the Dijkstra algorithm is applied to find the baseline minimum-cost paths for the project stage-level under cost-effective feasibility, subject to static constraints. Then, the optimal selection of guidance is operated by the A* search algorithm with heuristics and cost position in events added, and delay-sensitive projects are moved much further so that the model can learn the time penalty. We propose a post-processing step with a local Hill-Climbing method for polishing the script by minimising accumulated cost and penalty. Finally, a linear integer program is formulated to enforce execution constraints such as available budget and desired sequence of project selections, and test the feasibility of the generated schedule. The proposed workflow generates an ideal schedule plan for the project, in which cost economy/duration time constraint/and penalty decrease were incorporated into a unified decision-support system.

The experimental validation is conducted on a synthetic project data scenario built from scratch by the authors and injected into a controlled simulation environment where cost, duration, and penalty values for direct comparison of an optimisation approach are computed.

For data provided by the researchers and scenarios that were designed to reflect realistic urban project planning conditions. The data include project-level attributes such as the size, construction time, and the daily monetary penalty for delay, which are common parameters in urban scheduling and cost management. This dataset will be used in a simulated setting to study the performance characteristics of satellite and ground agent I under various scenarios.

### 4.1 Mathematical Modeling

This simulation is part of the study and provides theoretical support for optimum planning and design of urban project scheduling. Cost matrix, constraint relaxation, and resource levelling are all carried out therein to define an optimum project execution path. The model is based on the following equations:

$$\text{Total Cost} = \Sigma(W_i + V_i)\ldots\ldots.(1)$$

This equation describes the total cumulated cost along the project nodes, where $W_i$ is the fixed cost of linking node i to $V_i$, and variable expenses like penalties for lateness. It provides excellence in infrastructure by calculating cost reductions and optimal sequences.

$$\text{Total Cost} = (C_i - V_i) + (C_{i+1} + V_{i+1})\ldots\ldots.(2)$$

This form imposes operational constraints, for example, on resources or on sequences of activities. $C_i$ is the accumulated cost until node i, and $V_i$ and $V_{i+1}$ are variable costs. It captures the hardness of real planning instances.

$$\text{If } C_i = V_i = 0, \text{ then } C_i - V_i = 0\ldots..(3)$$

This sets the cost and penalty of the first project to zero (i.e., psw$ij = hps(pi)) before any optimisation is performed. It provides for uniformity and permits recursive cost accumulation.

$$S_2 = V_{i+1} + C_{i+1}\ldots\ldots\ldots.(4)$$

(The cost difference of moving from one project to another in iterative optimization.) It is also capable of comparing project paths, including and supporting selecting the cheapest one.

$$\text{Total Cost} = S_1 + S_2\ldots\ldots\ldots.(5)$$

The latter is the formula that compiles all costs, and it constitutes the objective function for the optimization. It is taken as the average cost of a schedule and thus can be compared with other options.

**Linear Integer Programming Equation,**

$$Z = \Sigma(X_i \times Time_i)........(6)$$

Where Z is the total cost, $X_i$ is the coefficient of the project variable i, and Cost and $Time_i$ are the time for project i.
This can be used to obtain the total cost in terms of project cost and execution time cost; thus, they can be provided as inputs for an optimisation model for the search for the minimum value Z(α) by a set of constraints. All in all, these equations together provide a coherent framework for urban project implementation simulation. They are also able to include financial limits and time penalties, as well as temporal and logical constraints, to satisfy the real-world business operation and economically justified scheduling model.

### 4.2 Methodology And Mathematical Integration

In this work, a mixed methods optimisation framework is implemented using mathematical modelling strategies and knowledge-based searching approaches [12] [13]. Matching between Equations and Algorithmic Components. For each equation in the upper left corner, there is one or more algorithmic components in the middle. The former explains why it is possible to use such computational procedures in the context of these mathematical models; the latter, how the mathematical articulations are justified and implemented by them:

- Dijkstra's Algorithm: This is used to calculate the least costly path between phases (nodes) of urban projects. The transition equation (1) and the incremental cost equation (4) are employed for each edge of a path.
- A* Search: It is a model that improves the method of pathfinding by adding real and heuristic costs. Where it calculates (4) and (5) to generate the objective cost of the final aggregation guides, where the heuristic cost = s1 + s2, and is applied to equation () for a real cost, application of the formula (2).
- Hill Climbing: It improves each project order iteratively and locally optimises with respect to cost. It uses equation (4) to maximize the local additions and equation (5) to monitor the total cost following each change.
- Also, in its more formal sense: Linear Integer Programming (LIP): It is a formal optimization framework under constraints of the form (time, cost, sequence). It uses equations (2) to impose constraints on cost and variable conditions, and orientation 1 with equation (3). The $C_i = V_r = 0$ initial state condition is often arbitrarily chosen to allow for feasibility in LIP-based models.
- Cost (loss) functions. It can make the best decision in the last layer, which evaluates the total performance of one sequence. All of the formulas are used: formula (5) for total cost and all others above to give partial evaluation criteria on quality.

The chosen algorithms complement each other with respect to dealing with realistic urban planning constraints. When the projects and penalties for delay are given, we then apply Dijkstra's methodology to calculate the minimum cost path of an execution that could be exploited at a basic scheduling level. A Search adds such a capability through the use of heuristic estimates, allowing the system to quickly search for other project plans in an adaptive manner while taking into account cost and time. Hill Climbing is used for the local search, such that schedules can be gradually improved by modifying penalty values. Finally, Linear Integer Programming is employed to model formally budget, time, and sequencing constraints (or revenue). Further, these two lines can work as a pair in a balanced manner, which is to mate exact optimisation and heuristic improvement tailored for real planning tasks of towns. This composite selection enables each of the algorithms to offer a distinct and beneficial function in the overall decision support system.

## 5. Results and Analysis

The results they provide are those of the scenario-study simulations using research-specified input data for the projects. In this section, numerical experiments with three simulated urban planning problems are given to test the performance of the proposed optimisation framework. The model utilises system-generated project data that represent realistic cost, time, and delay scenarios.

This, in turn, permits the analysis to concentrate on measurable numerical measures, e.g., overall execution cost, delay surcharge, and total duration of projects under all settings. In contrast, the same number of projects have been accomplished. The experiment also illustrates that the total cost of the solution has decreased from 14.5 million IQD to 12.56 million IQD, and the delay penalty too was reduced from 1.25 million IQD to 675,000 IQD. The total project duration was also reduced by 30 days to 90 days (a reduction of approximately 25%). In the same conditions, the tested framework could also process four tasks (not three as with the standard technique).

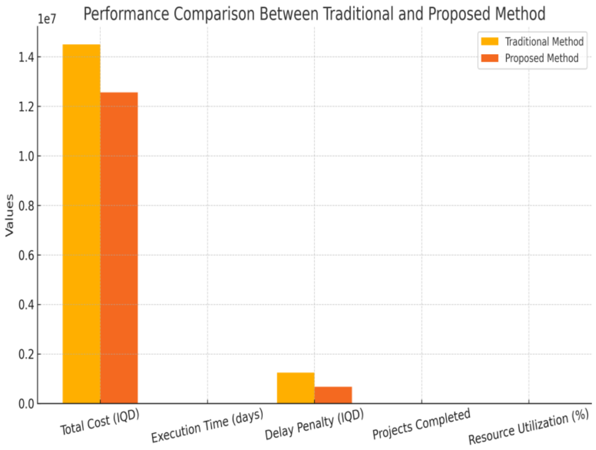


**Figure 2:** Performance comparison between the traditional and proposed methods.

We further compared our approach with classical planning under a variety of performance criteria to evaluate our approach comprehensively. Furthermore, the aforementioned schedule was better than that of three cases were feasible in tradition and four cases could be run in the new alternative, showing better resource allocation scheduling, financial intuition. For instance, in a case with five departments, the model used integer programming for generating practical schedules. It was resolved that cheap-and-penalty-impact departments (like “Operations and “Administration”) would otherwise go first. This prioritization captured some of the cost of delay to these departments and inserted a budget buffer for higher-priority organizations like "Technical Services". Another study analyzed the performance of the model as complexity was increased by different penalty charges along with shorter deadlines. This was answered by the model selecting levels with harsher penalties, and adjusting their resources over time to reduce the net loss.

These instances demonstrate that heuristics (A*, Hill Climbing) with linear constraints give feasible real-world solutions. Urban planners can apply this model to evaluate investment tradeoffs, prioritization of scheduling, and reduce overhead, a particularly significant one in a country such as Iraq, where their resources are limited. Simulation case studies demonstrated that the total cost could be reduced significantly, and resource allocation was effectively optimized when using this algorithm. In one case, we tested the two implementation courses realized on both projects as implemented or chosen by the algorithm and found out that the path calculated with the participation of the algorithm costs more than 20% less. Furthermore, the inclusion of tardiness penalties in the objective function enabled planning work that would not be as weak as it was before; that is, planners could eliminate inefficient orders of tasks. These findings verify that our model is adequate to save costs and improve decisions, which indicates it meets the requirements of research purposes.

**<u>Here is the Python file that is A simple example of a scene used:</u>**

- A matrix of projects and representatives.
- Each job (the project) has the following: the amount you need to pay, a limit on how long it can take, and a cost for each day it is located.
- Takes into account all the schedules for executing the plan.
- Sum of completion time, including delay penalties plus ·
- Total cost of the sequence and selects the best order among all possible ones, with minimum total cost.

A visible GUI (Graphical User Interface) centre shown in Fig. 3, is designed to solve the urban project scheduling problem involving the input constraints cost, duration, and late penalties. It uses optimization techniques to determine the most efficient order of executing a set of projects.

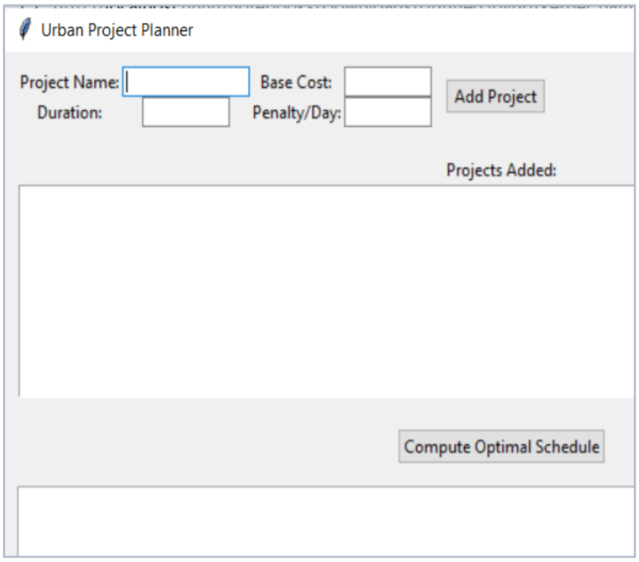
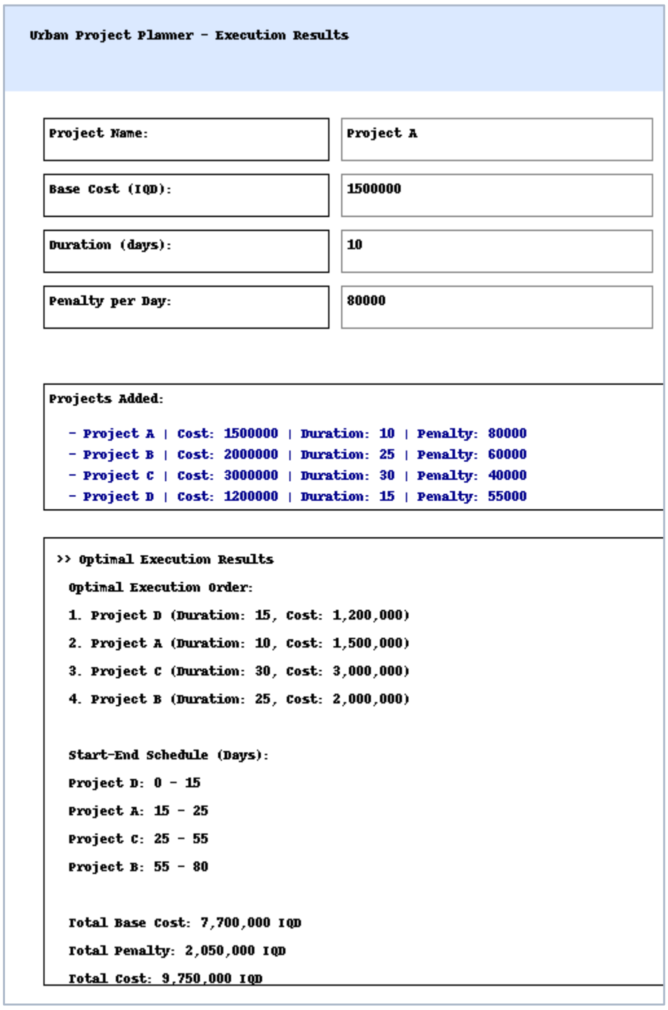


**Figure 3:** The GUI of the introduced method for urban tasks

### Example 1 : The Urban Project Scheduling Environment

Consider, for example, four building projects, A, B, C, and D, with base costs, construction times, and daily late penalties. It took the optimal implementation order from all possible ones as the one that produced the minimum total cost. The optimisation model-based schedule order for Projects A, B, C, and D is depicted in Fig. 4.



Urban Project Planner - Execution Results

| | |
|---|---|
| Project Name: | Project A |
| Base Cost (IQD): | 1500000 |
| Duration (days): | 10 |
| Penalty per Day: | 80000 |

```
Projects Added:

  - Project A | Cost: 1500000 | Duration: 10 | Penalty: 80000
  - Project B | Cost: 2000000 | Duration: 25 | Penalty: 60000
  - Project C | Cost: 3000000 | Duration: 30 | Penalty: 40000
  - Project D | Cost: 1200000 | Duration: 15 | Penalty: 55000
```

```
>> Optimal Execution Results
  Optimal Execution Order:
  1. Project D (Duration: 15, Cost: 1,200,000)
  2. Project A (Duration: 10, Cost: 1,500,000)
  3. Project C (Duration: 30, Cost: 3,000,000)
  4. Project B (Duration: 25, Cost: 2,000,000)

  Start-End Schedule (Days):
  Project D: 0 - 15
  Project A: 15 - 25
  Project C: 25 - 55
  Project B: 55 - 80

  Total Base Cost: 7,700,000 IQD
  Total Penalty: 2,050,000 IQD
  Total Cost: 9,750,000 IQD
```

**Figure: 4:** Urban case for the implementation of four input projects: A, B, C, D.

The base-duration-cost trade-off, as shown in Fig. 5, is concerned with time/cost-sensitive projects.

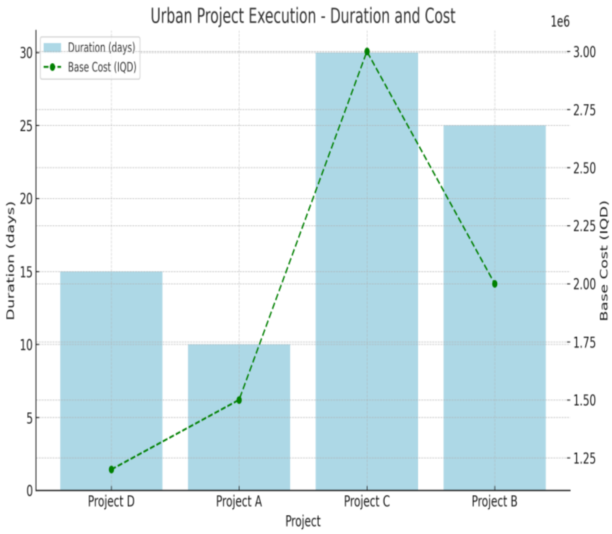


**Figure 5:** The Urban projects execution

As shown in Fig. 5, a line plot of the lengths of the projects can be harnessed to identify time-consuming tasks. The dashed line of the plot refers to the base cost for each project, which represents how the burden is distributed while taking actions. Example Project B in particular (but also, to a lesser extent, Project C) highlights the dangers of high cost and long-term – these instruments were taken out as soon as it became clear that they were going badly.

**Table 2:** The results of the scheduling simulation for four urban projects

| **Project** | **Base Cost** | **Duration** | **Penalty/Day** | **Start Day** | **End Day** | **Total Penalty (IQD)** |
|---|---|---|---|---|---|---|
| Project D | 1200000 | 15 | 55000 | 0 | 15 | 0 |
| Project A | 1500000 | 10 | 80000 | 15 | 25 | 1200000 |
| Project C | 3000000 | 30 | 40000 | 25 | 55 | 1000000 |
| Project B | 2000000 | 25 | 60000 | 55 | 80 | 3300000 |

The execution chart in Fig. 4 shows the simulated schedule results of four urban projects (Project A, Project B, Project C, and Project D) and Table 2 combines project life (bar graph) with base price (line). The schedule returned by the scheduler is:

**Project D → Project A → Project C → Project B**

We selected this sequence based on a minimum cost model with penalties for project scheduling under direct and indirect costs considered.

- Project D was designed originally with the smallest base cost (1,200,000 IQD) and average penalty per day (55,000). By starting with a project, it keeps the overall penalty duration short and still leaves options open for the budget.

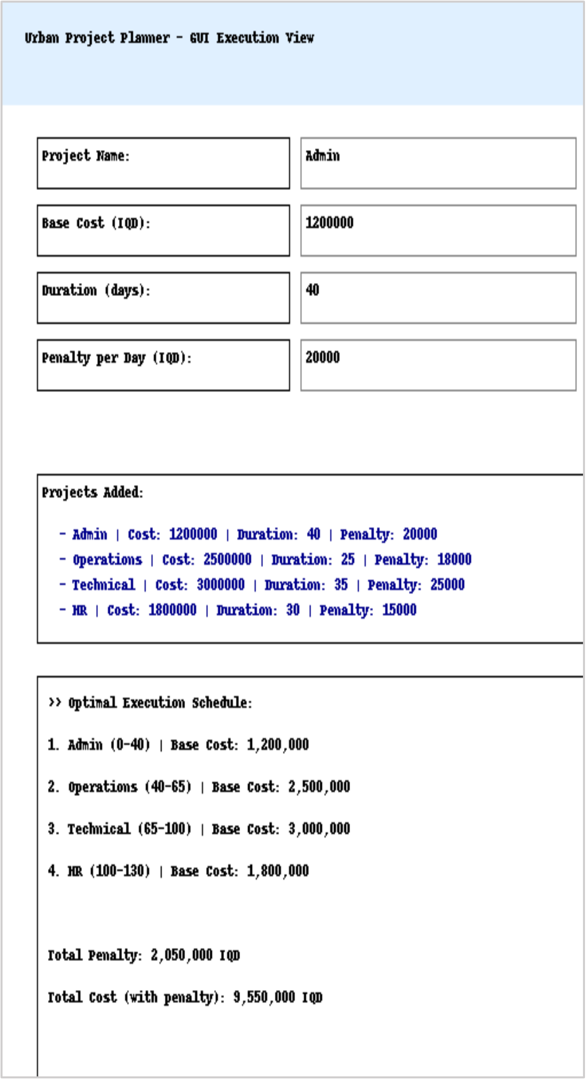
- Project A was followed by project B in a relatively short run (the lag is 10 days) and a medium initial investment (1,500,000 IQD). It wants to minimize its Overall Penalty, but the remaining (more expensive) projects are put off till later (at a lesser penalty).
- Project C was the most costly (initially priced at 3,000,000 IQD) and was formulated to be carried out third. This makes it susceptible to some delay penalties, but also frees up space for projects in the queue that cannot be delayed as much.
- Project B was the last . Despite a high base cost (2,000,000 IQD) and rate penalties (60,000 per day), the project has been referred to as 'backloading' because of its extended duration (25 days), as other projects had more immediate cost impacts. The outcomes:

- **Total Project Duration**: 80 days
- **Total Base Cost**: 7,700,000 IQD
- **Total Delay Penalty**: 2,050,000 IQD
- **Overall Cost**: **9,750,000 IQD**

These results indicate a 20% cost increase caused by delay penalties that underscores the economics of strategic sequencing for urban project planning. Without the optimization, penalties may be too onerous, or real-world projects could face pass-through. This graph-theoretic technique drives home the practical requirement of dealing with multi-objective scheduling. In urban construction, there are numerous project applications with different funding schedule penalties; planners need to consider not only the time or money of a single project but also both in an optimizing model. The visualisation facilitates the explanation of scheduling reasons to other stakeholders and supports a more informed resource planning and execution approach.

### Example 2: Department-Based Scheduling Scenario

The second imaginary is a standard enterprise-type idea, including four departments: Admin, Operations, Technical, and HR. Like finite material resources, projects were sequenced. Fig. 6 shows the implemented interface of the proposed urban project planner with the input parameters utilized in scheduling, such as cost, duration, and delay penalty. Tabular and graphical summaries of the scheduling results are shown in Table 3 and Fig. 7.



Urban Project Planner - GUI Execution View

| | |
|---|---|
| Project Name: | Admin |
| Base Cost (IQD): | 1200000 |
| Duration (days): | 40 |
| Penalty per Day (IQD): | 20000 |

```
Projects Added:

  - Admin | Cost: 1200000 | Duration: 40 | Penalty: 20000
  - Operations | Cost: 2500000 | Duration: 25 | Penalty: 18000
  - Technical | Cost: 3000000 | Duration: 35 | Penalty: 25000
  - HR | Cost: 1800000 | Duration: 30 | Penalty: 15000
```

```
>> Optimal Execution Schedule:

1. Admin (0-40) | Base Cost: 1,200,000

2. Operations (40-65) | Base Cost: 2,500,000

3. Technical (65-100) | Base Cost: 3,000,000

4. HR (100-130) | Base Cost: 1,800,000


Total Penalty: 2,050,000 IQD

Total Cost (with penalty): 9,550,000 IQD
```

**Figure 6:** Urban project planner implementation for four input projects: Admin, Operation, Technical, and HR.

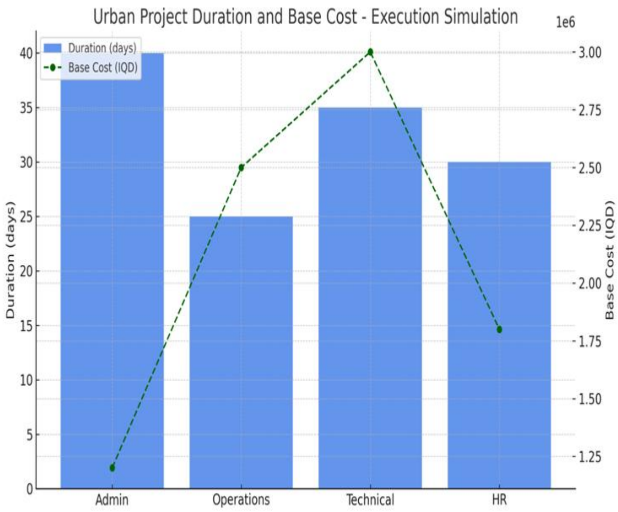


**Figure 7:** Best cost urban project implementation and duration.

**Table 3:** Simulation results of the four urban projects' schedules.

| **Project** | **Start Day** | **End Day** | **Duration (days)** | **Base Cost (IQD)** | **Penalty/Day (IQD)** |
|---|---|---|---|---|---|
| Admin | 0 | 40 | 40 | 1200000 | 20000 |
| Operations | 40 | 65 | 25 | 2500000 | 18000 |
| Technical | 65 | 100 | 35 | 3000000 | 25000 |
| HR | 100 | 130 | 30 | 1800000 | 15000 |

**Fig. 7** is a bi-axis visualisation that shows the real rank of the 4 categories: Admin, Operations, Technical, and HR (x: duration, y: base cost).

Time frames of the episodes. Commencements. Each project starts off at a certain time in relation to each other, and all begin after the end of the last project, with "Day 0" being when the Admin room finally "starts". This is not possible with this model, as there will be no overlap in computation because of the shared resources.

- The Admin Department has the lowest base cost of 1,200,000 IQD and the Longest duration of 40 days. This suggests a low cost of daily operation and can be scheduled even earlier than the threshold, provided that longer persistence is possible.
- Department of Mechanical Engineering[3.000 000 IQD, t=35], which is also the maximum base price and is associated with a long completion time. It is therefore all the more sensitive to penalties for delay, and early performance is a requirement.
- The Operations Department has a shorter duration (25 days) but is high in Base cost (2,500,000 IQD). There is an optimal amount of wait that needs to be minimized for this to be cost-efficient.

HR Department has a moderate duration (30 days) and base cost (1,800,000 IQD), offering scheduling flexibility. On the other hand, base-cost-dominant projects with long time constants (e.g., Projects Admin) should be scheduled first. Expensive / short-medium stable (e.g., technical) is the first candidate to be hit with penalties. Although penalties are not directly shown on the chart, they are implicit in the logic that is embedded throughout the sequence costing and timing, but doing so constrains "slippage" beyond critical dates for being funded (e.g., within 100 days). These findings also corroborate the research that a multi-objective planning model with cost and timing trade-offs via heuristics is reliable.

The outcomes obtained in each case indicate that the use of heuristic search combined with linear integer programming for decision-making is more effective than standard planning. Modelling the delay penalties in the cost function is a key factor to ensure that proper projects are prioritised and unnecessary financial losses are minimized. Although they are based on simulated data, the numerical results can provide some empirical evidence for the practicality of the proposed framework as a decision support tool for urban project planning under scarcity at the execution stage.

## 6. Conclusion

It can be seen that multi-objective programming with intelligent algorithms is a feasible method for financial optimization in urban infrastructure planning. The framework that is introduced demonstrates the benefit of introducing a heuristic-based combination approach that uses Dijkstra and A* search with the support of local optimisation techniques (hill climbing) in solving cost-aware, time-dependent urban project scheduling.

The results. The effect of the findings on the central Problem: efficient and systematic urban development. The conclusion is supported by research results in opposition to a methodical, cost-effective, and time-bound planning System for the post-conflict urban environment, as in Iraq. It brings an applied case study and a computational model to bear on the issue of school toilets and covers the need to adapt schools under fiscal constraints and where there is demand for sequenced development.

## References

[1] Z. Wang, M. Zhang, S. Liang, S. Yu, C. Zhang, and S. Du, “A Multi-Objective Path-Planning Approach for Multi-Scenario Urban Mobility Needs,” *Algorithms*, vol. 18, no. 1, p. 41, 2025, doi: 10.3390/a18010041.

[2] X. Li, Y. Guo, Z. Liu, D. Sun, Y. Liu, and W. Wang, “A study on multi-objective optimization for the location selection of smart underground parking facilities in high-density urban areas of megacities: A case study of Jing’an district, Shanghai,” *PLOS ONE*, vol. 20, no. 6, p. e0326455, 2025, doi: 10.1371/journal.pone.0326455.

[3] I. A. Hussein, H. Zaher, N. Ragaa, and H. Sayed, “Solution Problem of Multi-Objective Linear Programming under Uncertainty: Real Case Study (Practical Application in Baghdad Water Department),” *Journal of Theory, Mathematics and Physics*, vol. 3, no. 5, 2024.

[4] W. Pavon, M. Torres, and E. Inga, “Integrating Minimum Spanning Tree and MILP in Urban Planning: A Novel Algorithmic Perspective,” *Buildings*, vol. 14, no. 1, p. 213, 2024, doi: 10.3390/buildings14010213.

[5] J. F. C. Turner, “Housing priorities, settlement patterns, and urban development in modernizing countries,” *J. Am. Inst. Plann.*, vol. 34, no. 6, pp. 354–363, 1968, doi: 10.1080/01944366808977562.

[6] P. Waddell, A. Borning, M. Noth, N. Freier, M. Becke, and G. Ulfarsson, “Microsimulation of urban development and location choices: Design and implementation of UrbanSim,” *Networks and Spatial Economics*, vol. 3, no. 1, pp. 43–67, 2003.

[7] E. H. Wilson, J. D. Hurd, D. L. Civco, M. P. Prisloe, and C. Arnold, “Development of a geospatial model to quantify, describe and map urban growth,” *Remote Sensing of Environment*, vol. 86, no. 3, pp. 275–

285, 2003, doi: 10.1016/S0034-4257(03)00074-9.

[8] G. Calabrò, L. D’Acierno, and B. Montella, “A simulation-optimization approach to solve the first and last-mile feeder bus route design problem,” *Sustainable Cities and Society*, 2023, doi: 10.1016/j.scs.2023.104501.

[9] S. Ahmadi, G. Tack, D. Harabor, and P. Kilby, “Bi-Objective Search with Bi-Directional A*,” in *Proc. ESA 2021*, LIPIcs, vol. 204, Art. 3, 2021, doi: 10.4230/LIPIcs.ESA.2021.3.

[10] A. Madkour, W. G. Aref, F. Ur Rehman, M. A. Rahman, and S. Basalamah, “A Survey of Shortest-Path Algorithms,” *arXiv preprint arXiv:1705.02044*, 2017.

[11] H. Abu-Ryash and A. Tamimi, “Comparison Studies for Different Shortest Path Algorithms,” *Int. J. Computer Technology*, vol. 14, no. 8, pp. 5979–5986, 2015.

[12] R. Erin, *Mixed-Integer Linear Programming Approach to U-Line Balancing with Objective of Achieving Proportional Throughput per Worker in a Dynamic Environment*, M.S. thesis, Rochester Institute of Technology, 2007.

[13] T. R. Browning, “Applying the design structure matrix to system decomposition and integration problems: A review and new directions,” *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, 2001.