



Automated construction schedule optimisation using genetic algorithm

K. R. Srimathi¹ · A. Padmarekha¹ · K. S. Anandh¹

Received: 11 May 2023 / Accepted: 22 May 2023 / Published online: 7 June 2023
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

Construction project management is a complex process involving numerous resources and activities that present challenges to the scheduling of each activity. Effective scheduling is essential to the success of any construction project, but resource allocation conflicts present a significant challenge to construction managers. In this paper, a genetic algorithm (GA)-based construction schedule optimisation model is proposed that takes into account resource constraints such as construction activity, construction site, labour, machinery, and working hours for labour and machinery. The random schedule, which is created with many constraints and parameters, causes conflicts in its schedule and cannot be used directly. In this work, the genetic algorithm uses selection, mutation, and crossover processes to create a new conflict-free schedule until the desired fitness level is reached or the maximum number of iterations is completed. The proposed model is implemented in Python, and the conflict-free schedule is printed as a result. The novelty of this paper is an attempt to apply the job shop scheduling technique for conflict-free daily schedules in construction using GA. This algorithm can be adapted to optimise the construction schedule for any project, and it can incorporate additional factors that affect the construction schedule. Overall, this paper makes a valuable contribution to construction project management by presenting an effective optimisation model for construction project scheduling.

Keywords Genetic algorithm · Construction schedule optimisation · Resource allocation · Python implementation

Introduction

The construction industry faces numerous challenges, including cost overruns, delays, and low productivity. These challenges can be resolved to some degree using an optimised schedule for construction activity. Scheduling is the development of a plan for the sequence of activities required to complete a construction project. The traditional method of manually scheduling construction projects is not only time consuming, but also prone to errors and inconsistencies (Toklu, 2002). For economical construction, it is necessary to optimise the schedule of construction activities.

Schedule optimisation helps to ensure that the right resources are available at the right time, reducing the risk of delays or downtime due to lack of resources. The complexity of schedule optimisation increases with the size of the project and the constraints placed on resources such as labour and machinery. The execution of work during construction is highly dependent on external factors such as resource availability. For more accurate planning and execution, it is imperative that you consider all these constraints when optimising the schedule. Sometimes, resources such as labour and machinery are used on multiple construction sites. In such cases, constraints arise due to the availability of resources. After understanding the importance of optimisation, you need to use an appropriate algorithm that will give better results considering all the constraints. The various algorithms used for construction schedule optimisation can be divided into mathematical and heuristic optimisation approaches.

✉ K. R. Srimathi
ks5616@srmist.edi.in

✉ A. Padmarekha
padmarea@srmist.edu.in

¹ Department of Civil Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamilnadu, India

Optimization based on a mathematical approach

A mathematical approach to optimising construction schedules involves using practical and experience-based methods to develop a schedule that balances time, cost, and resource constraints. This involves breaking the project down into smaller components, identifying critical activities, and determining the best sequence of those activities to minimise delays and resource conflicts. CPM (critical path method) and PERT (programme evaluation and review technique) are some of the widely used mathematical programming techniques for construction planning. These techniques are limited to deterministic scheduling problems and do not consider the uncertainty and randomness of construction projects.

Traditionally, construction scheduling has relied on the critical path method (CPM) or the programme evaluation and review technique (PERT) to optimise project schedules. However, these techniques require knowledge of the sequence of activities and their interdependencies to produce an optimal schedule. For construction projects with a high degree of uncertainty or a lack of data, it can be difficult to establish the required relationships between activities.

Heuristic approach to optimisation

Heuristic methods have also been proposed for solving construction scheduling problems. Heuristic methods do not guarantee optimal solutions, but they are computationally efficient and can deal with uncertain events (Zhou et al., 2013). Tabu search, simulated annealing, and ant colony optimisation are some of the most popular heuristic methods used in construction planning (Dewantoro & Sihombing, 2019).

Recently, evolutionary algorithms have emerged as popular approaches to solve optimisation problems in construction management (Slowik and Kwasnicka, 2020). The genetic algorithm is one such evolutionary algorithm that has been used to optimise construction schedules (Faghihi et al., 2014). Metaheuristic methods commonly used for construction schedule optimisation include democratic particle swarm optimisation, charged system search, magnetic charged system search, force field optimisation, dolphin echolocation optimisation, colliding body optimisation, and radiation optimisation (Kaveh, 2017; Kaveh, 2021). Although each method has its strengths, genetic algorithms (GAs) are often considered the best choice for optimising construction schedules because they balance exploration and exploitation, take a population-based approach that allows simultaneous evaluation of multiple solutions, are flexible in dealing with different problem types and constraints, can be parallelized, and maintain

solution diversity to prevent premature convergence. These characteristics make GAs very suitable for finding optimal or near-optimal schedules in construction projects. GA is based on the principle of natural selection (Albadr et al., 2020) and survival of the fittest and can handle complex, nonlinear, and multi-objective optimisation problems.

GA-based models have been proposed for various planning problems in civil engineering, including resource constrained project scheduling (RCPS), time–cost trade-off, and multi-project scheduling (Xie et al., 2021). A multi-objective genetic algorithm is a powerful optimisation technique that aims to find a set of solutions that optimise multiple competing objectives simultaneously. It uses genetic operators such as selection, crossover, and mutation to iteratively evolve a population of proposed solutions and find a diverse and well-distributed set of Pareto-optimal solutions. Mokhtarimousavi et al., 2015 proposes a three-objective mathematical model to solve the aircraft landing problem using the second version of non-dominated sorting genetic algorithm (NSGA) for two groups of aircraft. Kaveh et al., 2022 introduces a fuzzy multi-mode resource optimisation model with discrete time–cost–resource constraint (F-MRC-DTCRO) that takes into account uncertainties in cost and allows trade-offs in time, cost, and resources whilst satisfying the project constraints. Kaveh et al., 2021 propose a multi-objective optimisation model based on the non-dominated sorting differential evolution algorithm (NSDE-R) to consider multiple objectives such as time, cost, resources, environmental impact, safety risks, and quality during construction. The GA-based models have shown promising results in optimising construction schedules and in some cases have outperformed other optimisation techniques.

The genetic algorithm is an iterative process that starts with an initial population of potential solutions. The population of solutions undergoes an evolution in which the fitter solutions have a higher probability of being selected for reproduction. The evolution process involves selecting the fittest solutions, applying genetic operators such as crossover and mutation, and creating a new population of potential solutions. The genetic algorithm continues this iterative process until a satisfactory solution is found or the maximum number of iterations is reached.

The genetic algorithm has several advantages over traditional optimisation approaches such as linear programming and dynamic programming. The genetic algorithm is a global optimisation approach that searches the entire solution space for the optimal solution, whereas traditional optimisation approaches search a subset of the solution space. The genetic algorithm is also suitable for solving problems that require combinatorial optimisation, such as scheduling, and it does not require the problem to be convex or differentiable. It can also generate feasible schedules in real time without requiring a complete

understanding of the sequence of activities or their interdependencies. In addition, the algorithm can adapt to unexpected changes in resource availability or site conditions.

Table 1 provides a detailed review of the literature in which the genetic algorithm has been used to optimise schedules in construction. Table 1 clearly shows that the genetic algorithm is widely used in automated construction schedule optimisation (ACS). ACS aims to optimise the allocation of resources, the sequencing of tasks, and the determination of project duration to minimise the total cost and duration of the project (Bagshaw, 2021). However, the effectiveness of ACS depends on the quality of input data and parameters used in the optimisation process (Fujisaku et al., 2021). The objective of this paper is to develop and validate a genetic algorithm-based ACS model that considers various constraints and objectives of the construction project to produce a daily construction schedule. The methodology used for the optimisation is explained in the following sections.

Genetic algorithms (GA) are a type of optimisation algorithm based on the principles of natural selection and genetics. In construction schedule optimisation, GA involves creating a population of potential solutions, evaluating them against a fitness function, and then selecting the best solutions for breeding and generating a new generation of solutions. This process is repeated until a satisfactory solution is found. Hybrid genetic algorithms (HGA) combine the principles of GA with other optimisation techniques to create a more powerful and effective optimisation approach. In the context of construction schedule optimisation, HGA may involve the use of other algorithms, such as simulated annealing or ant colony optimisation, in conjunction with GA to overcome its limitations. A key difference between GA and HGA is that HGA may be better suited for solving more complex optimisation problems that require a more sophisticated approach. For example, when optimising construction schedules, HGA may be better suited for projects that involve multiple constraints and objectives because

Table 1 Review of literature on genetic algorithm

Reference	Factors considered for optimisation	Algorithm	Methodology
Agdas et al. (2018)	Availability of resources, construction sequence, and duration of each activity	Not specified	Genetic algorithm
Alcaraz and Maroto (2001)	Sequence of activities and allocation of resources	Not specified	Genetic algorithm
Zhou et al. (2018)	Stochastic nature of construction activities	Modified particle swarm optimisation algorithm	Genetic algorithm
Bettemir et al. (2015)	Resource-constrained project scheduling problem	Simulated annealing algorithm	Hybrid genetic algorithm
Lotfi and Tavakkoli-Moghaddam (2013)	Precedence relationship	Priority-based encoding scheme	Genetic algorithm
Long et al. (2009)	Critical path method	Not specified	Genetic algorithm
Kim et al. (2008)	Resource-constrained project scheduling problem	Not specified	Genetic algorithm
Wu et al. (2018)	Not specified	Particle swarm optimisation algorithm	Hybrid genetic algorithm
Hamidinia et al. (2012)	Dynamic programming approach	Not specified	Genetic algorithm
Kao and Zahara (2008)	Multi-mode resources	Particle swarm optimisation algorithm	Hybrid genetic algorithm
Song and Li (2022)	Multi-objective optimisation	Differential evolution algorithm	Hybrid genetic algorithm
Madeira et al. (2005)	Multiple constraints	Multi-objective optimisation approach	Genetic algorithm
Dang Quoc and Nguyen Doan (2022)	Resource-constrained project scheduling problem	Particle swarm optimisation algorithm	Hybrid genetic algorithm
Katoch et al. (2021)	Hybrid encoding scheme	Not specified	Genetic algorithm
Wang et al. (2022)	Multiple objectives	Multi-objective optimisation approach	Genetic algorithm
Silva et al. (2022)	Time and safety	Bi-objective optimisation approach	Hybrid genetic algorithm
Salama and Moselhi (2019)	Uncertainty of construction activities	Multi-objective optimisation approach	Genetic algorithm
Afshar et al. (2007)	Trade-off between time, cost, and quality	Multi-objective optimisation approach	Genetic algorithm
Hamdi and Zaied (2019)	Resource allocation	Particle swarm optimisation algorithm	Hybrid genetic algorithm
Goncharov and Leonov (2017)	Resource-constrained project scheduling problem	Dynamic programming approach	Genetic algorithm

it can combine multiple optimisation techniques to find a better solution. Another important difference is that HGA requires more computational resources and expertise to implement because multiple optimisation techniques must be integrated into a single framework.

Methodology

The genetic algorithm works by generating a population of possible schedules, each represented as a chromosome, and then evaluating each chromosome to determine its fitness. In the case of the specific implementation, each chromosome represents a possible schedule of site activities. The fitness function is designed to evaluate how well the schedule satisfies constraints such as the availability of labour and machinery and the duration of each activity. The genetic algorithm then applies genetic operations such as selection, crossover, and mutation to evolve the population of schedules and improve the fitness of the best individuals. Figure 1 illustrates the methodology of the proposed blueprint optimisation model.

Process in GA to optimise the construction schedule

The proposed approach to optimising the construction schedule using a genetic algorithm can be described in the following steps:

Collecting the input data: The input data for construction schedule optimisation should include information such as the name of the activity, the duration, the available resources such as machines and labour, and the resource allocation constraints such as the maximum working time for each

labourer and machine and the availability of a given resource for a given time.

Initialisation: In this step, an initial population of schedules is created. Each schedule consists of a sequence of classes, where each class represents a particular activity to be scheduled along with the resources required for the activity. The size of the population is a parameter that can be set depending on the size of the problem.

Fitness evaluation: In this step, the fitness of each plan is evaluated against the objective function. The objective function aims to fit the activities into the allowable duration of the project whilst meeting the resource allocation constraints.

Selection: Selection selects the fittest schedules from the population to create a new population for the next generation. In this paper, tournament selection is used, where a random subset of schedules is selected and the fittest amongst them is chosen.

Crossover: In this step, the selected schedules are combined to create a new population. To do this, two schedules are taken, and a new schedule is created by combining parts of both schedules. In this work, a one-point crossover is used, where a random point in the schedule is selected and the schedules are split at that point. The first part of one schedule is combined with the second part of the other schedule to create a new schedule.

Mutation: In this step, the new population is randomly mutated to introduce diversity into the population. In this work, the mutation rate is a parameter that can be set according to the problem. In mutation, two classes are swapped in the schedule.

Elitism: In this step, the best schedules of the current population are directly transferred to the next generation without any changes. This ensures that the best schedules are preserved.

Termination: The algorithm terminates when a stopping criterion is met. This can be a maximum number of iterations or a minimum fitness value. The algorithm runs for a fixed number of iterations, and the best schedule is returned as the optimal schedule.

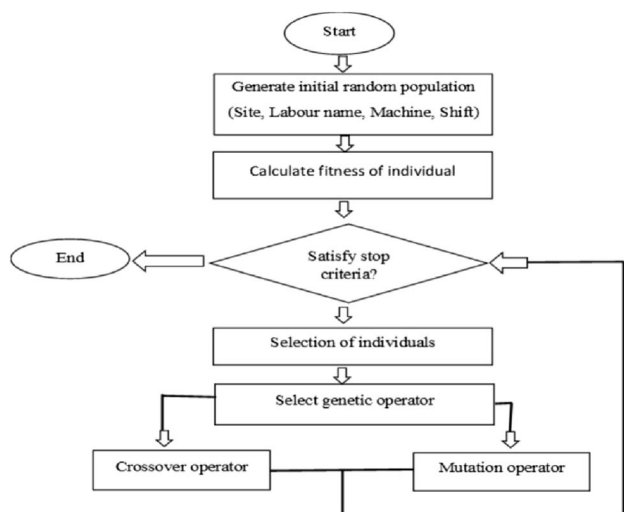


Fig. 1 Methodology flow chart for construction schedule optimisation model

Implementation of the genetic algorithm

The Python for optimising the construction schedule, the classes for representing the entities involved in the construction project can be found in Appendix A. These classes include data, schedule, class, machine, labour name, labour working hour, activity, and construction site. The detailed function of each class is explained below.

The Python code begins by importing the prettyTable(a module in python) and random modules, which are used to display the results and generate random numbers,

respectively. The code then defines the global variables for the population size, number of elite schedules, tournament selection size, and mutation rate. These variables are used by the genetic algorithm to create the initial population, select the best schedules, and create a new schedule that has no schedule conflict.

The data class is then defined, which initialises the entities involved in the construction project. The class initialises the machines, labour working hours, labour names, and activities, and stores them in the appropriate lists. The sites are also initialised in this class, which represent the different construction sites in the project.

The schedule class is then defined, which represents a particular schedule for the project. The class initialises the classes, number of conflicts, fitness, and class number. The class also defines the methods for getting the classes, the number of conflicts, and fitness. The initialise method is used to create a new schedule, which assigns the resources to the activities randomly. The fitness of a schedule which represents the number of conflicts in the schedule is calculated. The fitness is calculated by checking if the machines and labour are available during the scheduled time of the activity.

The class is then defined, which represents a class of activities. The class initialises the class number, construction site, activity, machine, labour name, and labour working hour. The class also defines the methods for getting and setting the construction site, activity, machine, labour name, and labour working hour.

The machine class is then defined, which represents a machine. The class initialises the machine name and working time. The class also defines the methods for getting and setting the working time.

The labourname class is then defined, which represents a labour name. The class initialises the labour name and defines the method for getting the labour name.

The labourworkinghour class is then defined, which represents a labour working hour. The class initialises the shift and working hours. The class also defines the method for getting the shift and working hours.

The activity class is then defined, which represents an activity. The class initialises the activity number, activity name, labour, and duration.

Once the classes are sorted by fitness, the best schedules are selected as elite schedules. The function breedpopulation uses the elite schedules and a selection process to create a new population of schedules. The new population is created by generating random indexes and selecting the schedules corresponding to those indexes for a tournament selection process. The number of schedules selected is determined by the `tournamentselectionsize` constant, and the best schedule from the selected schedules is added to the new population. The process is repeated until

the new population is the same size as the old population. Once the new population is created, the function `mutatepopulation` is called to mutate the schedules in the new population. The mutation process randomly selects schedules and changes the values of their classes. The number of schedules selected for mutation is determined by the `mutationrate` constant.

The `mutatepopulation` function selects a random schedule from the population and selects a random class from the schedule. The values for the class are then randomly changed, including the machine, labour, and working hours. The `crossoverpopulation` function is called to create a new population by breeding schedules from the current population. The function selects a random schedule from the population and creates a new schedule by selecting classes from the selected schedule and another random schedule. The function then replaces the least fit schedule in the current population with the new schedule. The process is repeated until the desired number of new schedules is created.

Finally, the `printschedule` function is called to print out the best schedule. This function creates a `prettyTable` object and populates it with the data for the schedule. The function prints out the table containing the new schedule with 0 conflicts (elite schedule) to the console.

The detailed conceptual implementation of a genetic algorithm to create a model to optimise the construction schedule is discussed in this section. As an example, the construction of a precast bridge is considered. This genetic algorithm model takes the inputs of activity, duration, construction sites, labour name, labour time (shift-based), construction site, machines, shift times, and machine labour time. The data used for optimisation are listed in Table 2 through Table 6. In Table 2, there are three construction sites, namely construction site 1, construction site 2, and construction site 3. In construction site 1, driving activities, pile caps, and piers are performed. In site 2, the activities are pier and pier cap and in site 3, the activities are pier, pier cap and girder erection. A separate ID was assigned for each activity and the details are shown in Table 3. The duration of each activity and the assigned labour are also listed. Assigning a specific labour for each activity helps in assigning labour with specific skills to perform the activity.

Table 2 Work at different construction site considered for optimisation

Construction site	Activity
Site 1	Piling, pile cap, pier
Site 2	Pier, pier cap
Site 3	Pier cap, girder erection, pier

Table 3 Data on activity, duration and labour details

Id	Activity	Duration	Labour name
A	Piling	8	[L1-Labour 1],[L2-Labour 2]
B	Pile cap	7	['L1','Labour1'],['L2','Labour 2'],['L3','Labour 3']
C	Pier	6	[L1-Labour 1],[L2-Labour 2]
D	Pier cap	7	[L3-Labour 3],[L2-Labour 2]
E	Girder erection	8	[L4-Labour 3]
F	Deck slab	6	[L1-Labour 1],[L3-Labour 3]

Table 4 Machine and machine working time considered for optimisation

Machine	Machine working time
M1	8
M2	7
M3	5

Table 5 Shift wise labour working hour

Id	Labour working hour
SHIFT 1	9:00—12:00
SHIFT 2	12:00—3:00
SHIFT 3	3:00—6:00
SHIFT 4	6:00—9:00

Table 4 lists the machines and the available labour time for the machines. In Table 5, the shifts and shift times are considered. For example, shift 1 is from 9:00 to 12:00.

Results and discussion

Schedule optimisation is performed to reduce the number of conflict points to zero. Table 6 shows the random schedule created for the first iteration. Since the created schedule (Table 6) is random, there will be problems, practical

difficulties in execution or inconsistencies in the schedule, which are referred to here as conflict. Table 6 and Table 7 contain data such as the fitness and the number of conflicts, followed by the summary of the schedule, which contains the random assignment of the site, the machine and the working hours. For example, in Table 6, machine M3 is assigned to activity B and activity D for the same shift 2, which is not feasible because the machine cannot be used on both sites at the same time. Such conflicts are counted and displayed. To resolve the conflicts described above, a genetic algorithm is implemented, and the schedule is repeatedly created so that there is no conflict in the displayed schedule. This new schedule is conflict free and can be used for execution. It is considered as an elite schedule, as shown in Table 7. In this case, 76 iterations were required to achieve conflict free.

Conclusion

Implementing genetic algorithms provides a flexible and scalable approach to optimising complex scheduling problems in real-world applications, such as construction project management. By leveraging the principles of natural selection and evolution, genetic algorithms can provide efficient and effective solutions to complex scheduling problems, leading to higher productivity and thus lower costs and better project outcomes.

The proposed algorithm uses a set of data classes that represent the construction site, activity, labour and machine. These classes are used to create a schedule, which is then tested for its fitness based on a set of constraints, including resource allocation constraints. The algorithm evolves the population of schedules over time using selection, crossover and mutation operations to produce the best schedule.

After testing the proposed algorithm with a sample dataset, the results show that it is an effective method for optimising construction schedules. The algorithm was able to generate a conflict-free schedule that maximises the use of resources in the given time frame.

Table 6 Generation 0—schedule (with conflicts)

Class	Construction Site	Activity (number, duration)	Machines (working time)	Labour (Id)	Labour working hours
1	Site 1	Piling (A, 8)	M1 (8)	Labour 2 (L2)	9:00–12:00 (SHIFT1)
2	Site 1	Pile cap (B, 7)	M2 (7)	Labour 3 (L4)	12:00–3:00 (SHIFT2)
3	Site 1	Pier (C, 6)	M3 (5)	Labour 1 (L1)	3:00–6:00 (SHIFT3)
4	Site 2	Pier (C, 6)	M1 (8)	Labour 1 (L1)	6:00–9:00 (SHIFT4)
5	Site 2	Pier cap (D, 7)	M2 (7)	Labour 3 (L4)	6:00–9:00 (SHIFT4)
6	Site 3	Pier cap (D, 7)	M3 (5)	Labour 1 (L1)	12:00–3:00 (SHIFT2)
7	Site 3	Girder erection (E, 8)	M3 (5)	Labour 2 (L2)	3:00–6:00 (SHIFT3)
8	Site 3	Pier (C, 6)	M2 (7)	Labour 3 (L4)	9:00–12:00 (SHIFT1)

Table 7 Generation 76—schedule without conflicts

Class	Construction site	Activity (number, duration)	Machines (working time)	Labour (ID)	Labour working hours
1	Site 1	Piling (A, 8)	M1 (8)	Labour 1 (L1)	9:00–12:00 (SHIFT1)
2	Site 1	Pile cap (B, 7)	M1 (8)	Labour 3 (L4)	6:00–9:00 (SHIFT4)
3	Site 1	Pier (C, 6)	M2 (7)	Labour 1 (L1)	3:00–6:00 (SHIFT3)
4	Site 2	Pier (C, 6)	M2 (7)	Labour 1 (L1)	6:00–9:00 (SHIFT4)
5	Site 2	Pier cap (D, 7)	M2 (7)	Labour 2 (L2)	9:00–12:00 (SHIFT1)
6	Site 3	Pier cap (D, 7)	M1 (8)	Labour 3 (L3)	3:00–6:00 (SHIFT3)
7	Site 3	Girder erection (E, 8)	M1 (8)	Labour 3 (L4)	12:00–3:00 (SHIFT2)
8	Site 3	Pier (C, 6)	M2 (7)	Labour 1 (L1)	12:00–3:00 (SHIFT2)

This model can be used in the real world specifically in construction project management where many activities need to be efficiently scheduled within a given time frame and with limited resources such as labour and machinery.

The algorithm is easy to implement and can be extended to solve more complex construction scheduling problems. It is an open-source implementation that can be adapted to any construction project.

In summary, the genetic algorithm is a promising approach for optimising construction schedules and its implementation in Python provides an efficient and easy-to-use tool for construction managers. This algorithm provides a platform for better project management decisions and can help construction managers save time and reduce costs by utilising resources effectively.

The future of optimising construction schedules with genetic algorithms is promising. Real-time optimisation, collaborative optimisation techniques, and integration with BIM models can improve construction schedules. Genetic algorithms can also optimise schedules under uncertainty. However, the challenges are in problem formulation, exploring a complex search space, developing an accurate fitness function, and accounting for stakeholder uncertainty and preferences. Effective communication and collaboration are essential for successful implementation and widespread application in the construction industry.

Appendix A

Pseudocode for the proposed model

- Define the necessary classes and their attributes for the scheduling problem, including the data, class, activity, construction site, machine, and labour classes.

- Create a schedule class to represent a schedule and initialise its attributes such as classes, number of conflicts, fitness, class number, and is fitness changed.
- Create a data class that initialises the machine, labour working hour, labour name, activity, and sites required for the scheduling problem.
- Define helper functions for the schedule class, such as get classes, get number of conflicts, and get fitness.
- Implement the initialise function of the schedule class, which initialises the classes in the schedule, the corresponding construction site, and the activity.
- Implement the calculate fitness function of the schedule class, which calculates the fitness of the schedule by checking the conflicts between classes.
- Implement the genetic algorithm, which creates an initial population of schedules, and selects the elite schedules, crossover and mutation of the population to generate new schedules.
- Evaluate the fitness of the new schedules and repeat steps 7 and 8 until the termination condition is met.
- Return the best schedule as output.

Author contributions KRS—idea, coding, paper drafting, AP—idea, paper draft preparation, draft paper review, and correction, KSA—draft paper review and suggestions.

Funding The authors declare that no funds, grants, or other support were received for the work presented in this manuscript.

Data availability The data supporting the results of the study are available from the corresponding author upon request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

- Agdas, D., Warne, D. J., Osio-Norgaard, J., & Masters, F. J. (2018). Utility of genetic algorithms for solving large-scale construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, 32(1), 04017072.
- Afshar, A., Kaveh, A., & Shoghli, O. R. (2007). Multi-objective optimization of time-cost-quality using multi-colony ant algorithm.
- Albadr, M. A., Tiun, S., Ayob, M., & Al-Dhief, F. (2020). Genetic algorithm based on natural selection theory for optimization problems. *Symmetry*, 12(11), 1758.
- Alcaraz, J., & Maroto, C. (2001). A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research*, 102(1), 83.
- Bagshaw, K. B. (2021). PERT and CPM in project management with practical examples. *American Journal of Operations Research*, 11(4), 215–226.
- Bettemir, Ö. H., & Sonmez, R. (2015). Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling. *Journal of Management in Engineering*, 31(5), 04014082.
- Dang Quoc, H., & Nguyen Doan, C. (2022). An Effective Hybrid Algorithm Based on Particle Swarm Optimization with Migration Method for Solving the Multiskill Resource-Constrained Project Scheduling Problem. *Applied Computational Intelligence and Soft Computing*, 2022.
- Dewantoro, R. W., & Sihombing, P. (2019, September). The combination of ant colony optimization (ACO) and tabu search (TS) algorithm to solve the traveling salesman problem (TSP). In 2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM) (pp. 160–164). IEEE.
- Faghihi, V., Reinschmidt, K. F., & Kang, J. H. (2014). Construction scheduling using genetic algorithm based on building information model. *Expert Systems with Applications*, 41(16), 7565–7578.
- Fujisaku, T., So, F. T. K., Igarashi, R., & Shirakawa, M. (2021). Machine-Learning optimization of multiple measurement parameters nonlinearly affecting the signal quality. *ACS Measurement Science Au*, 1(1), 20–26.
- Goncharov, E. N., & Leonov, V. V. (2017). Genetic algorithm for the resource-constrained project scheduling problem. *Automation and Remote Control*, 78, 1101–1114.
- Hamdi, M., & Zaied, M. (2019). Resource allocation based on hybrid genetic algorithm and particle swarm optimization for D2D multicast communications. *Applied Soft Computing*, 83, 105605.
- Hamidinia, A., Khakabimamaghani, S., Mazdeh, M. M., & Jafari, M. (2012). A genetic algorithm for minimizing total tardiness/earliness of weighted jobs in a batched delivery system. *Computers & Industrial Engineering*, 62(1), 29–38.
- Kao, Y. T., & Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8(2), 849–857.
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80, 8091–8126.
- Kaveh, A. (2017). *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*. Springer.
- Kaveh, A. (2021). *Advances in metaheuristic algorithms for optimal design of structures* (3rd ed.). Springer International Publishing.
- Kaveh A., Rajabi F., & Mirvalad S. 2021, Many-objective optimization for construction project scheduling using non-dominated sorting differential evolution algorithm based on reference points, *Scientia Iranica*, Volume 28, Issue 6, November and December 2021, Pages 3112–3128.
- Kaveh, A., & Rajabi, F. (2022). Fuzzy-multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using ENSCBO". *Periodica Polytechnica Civil Engineering*, 66(1), 50–62.
- Kim, J. L., & Ellis, R. D., Jr. (2008). Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling. *Journal of Construction Engineering and Management*, 134(11), 904–913.
- Long, L. D., & Ohsato, A. (2009). A genetic algorithm-based method for scheduling repetitive construction projects. *Automation in Construction*, 18(4), 499–511.
- Lotfi, M. M., & Tavakkoli-Moghaddam, R. (2013). A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. *Applied Soft Computing*, 13(5), 2711–2726.
- Madeira, J. A., Rodrigues, H., & Pina, H. (2005). Multi-objective optimization of structures topology by genetic algorithms. *Advances in Engineering Software*, 36(1), 21–28.
- Mokhtarimousavi, S., Rahami, H., & Kaveh, A. (2015). Multi-objective mathematical modeling of aircraft landing problem on a runway in static mode, scheduling and sequence determination using NSGA-II. *International Journal of Optimization in Civil Engineering*, 5(1), 21–36.
- Salama, T., & Moselhi, O. (2019). Multi-objective optimization for repetitive scheduling under uncertainty. *Engineering, Construction and Architectural Management*.
- Silva, P. M., Gonçalves, J. N., Martins, T. M., Marques, L. C., Oliveira, M., Reis, M. I., & Fernandes, J. M. (2022). A hybrid bi-objective optimization approach for joint determination of safety stock and safety time buffers in multi-item single-stage industrial supply chains. *Computers & Industrial Engineering*, 168, 108095.
- Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32, 12363–12379.
- Song, E., & Li, H. (2022). A hybrid differential evolution for multi-objective optimisation problems. *Connection Science*, 34(1), 224–253.
- Toklu, Y. C. (2002). Application of genetic algorithms to construction scheduling with or without resource constraints. *Canadian Journal of Civil Engineering*, 29(3), 421–429.
- Wang, M., Li, X., & Chen, L. (2022). An enhance multimodal multi-objective optimization genetic algorithm with special crowding distance for pulmonary hypertension feature selection. *Computers in Biology and Medicine*, 146, 105536.
- Wu, X., Bai, W., Xie, Y., Sun, X., Deng, C., & Cui, H. (2018). A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs. *Applied Soft Computing*, 73, 735–747.
- Xie, F., Li, H., & Xu, Z. (2021). Multi-mode resource-constrained project scheduling with uncertain activity cost. *Expert Systems with Applications*, 168, 114475.
- Zhou, H., Pang, J., Chen, P. K., & Chou, F. D. (2018). A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes. *Computers & Industrial Engineering*, 123, 67–81.
- Zhou, J., Love, P. E., Wang, X., Teo, K. L., & Irani, Z. (2013). A review of methods and algorithms for optimizing construction scheduling. *Journal of the Operational Research Society*, 64(8), 1091–1105.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.