Research Article Open Access

Prachi Sohoni*, Abdul Ghaffar, Vasant A. Matsagar, K.N. Jha

Optimization of shoring/reshoring levels in high-rise building construction

DOI 10.2478/otmcj-2018-0009
Received February 06, 2018; accepted July 23, 2018

Abstract: Formwork system is a significant constituent and a basic requirement for high-rise cast-in-place reinforced concrete buildings. Usually, the builders are confronted with the decision to choose the safe, optimum number of levels of shores/reshores for a predetermined safety factor and given grade of concrete, giving due consideration to the cost of formwork system. In this study, MATLAB program is developed to calculate the load distribution between the interconnected slabs and levels of shore/reshore of a slab formwork based on a simplified method. This program is further modified by incorporating genetic algorithm for the optimization of cost of construction for high-rise building. The cost of level of shores and reshores per floor is defined as a function of cycle time which directly reflects the increase in the cost of construction. Various combinations of shore and reshore levels with several grades of concrete for various safety factors are checked to minimize the cost of construction. The optimization equation is solved using genetic algorithm considering appropriate constraints to practically ensure feasible solutions. The case of one level of shores and numerous levels of reshores is better than one level of reshores and numerous levels of shores. The result of certain combination of shore and reshore levels is not the same when the level numbers are reversed. A comparative study is carried out to check the optimum cost for various safety factors. The program is useful for the designers to decide the levels of shores and reshores with minimized cost without compromising the safety of construction.

Keywords: formwork, shore, reshore, genetic algorithm, MATLAB, optimization, cost of construction

1 Introduction

Multistorey building construction is exponentially increasing as the shortage of land becomes a major problem. Hence to cater for the huge demand, the builders want to complete the building at a faster rate with as less investment as possible. In a multistorey building construction, both the time and cost play important roles. While going for fast construction, the cost of construction becomes a key factor. The cost of formwork is significant in developing countries, and it cannot be ignored. The formwork shares 30–60% of the total cost of reinforced concrete (RC) construction (Jha, 2011; Nawy, 2008; Senouci and Al-Ansari, 1996). Hence, it becomes necessary to lower the formwork cost to reduce the overall cost of RC construction.

However, while cutting down the cost, the safety of the structure during construction cannot be ignored. Early removal of the forms reduces the total cost of the temporary structure, but it may affect the quality of the construction. If the slabs are subjected to loads before the minimum strength is gained by the concrete, then the construction becomes weak and it increases the chances of failure. Hence, the safety of construction may become the major issue in such cases. The safety of the structure depends on the load-carrying capacity of the slabs and the load applied to it after formwork removal at the time of RC construction. The removal of the formwork increases the load on the lower slabs and form levels. Hence, care should be taken while removing the forms. The transfer of the construction loads between slabs and shore/reshore is studied based on a simplified method. Grundy and Kabaila (1963) proposed this simplified approach to find the loads on slab and shore/reshore at any time during construction. Since then, many researchers worked on the load transfer between the slab and shore/reshore. Several modified procedures have been designed such as the improved simplified method by Duan and Chen (1995), Fang et al. (2001), and Beeby (2001), structural characteristic parameter approach by Dongping et al. (2009), the new simplified procedure by Calderón et al. (2011), etc. However, Liu et al. (1988) showed that the simplified method given by

***Corresponding author: Prachi Sohoni,**
Indian Institute of Technology Delhi, New Delhi, India,
E-mail: prachi.purohit20@gmail.com
Abdul Ghaffar, Vasant A. Matsagar and K.N. Jha, Indian Institute of Technology Delhi, New Delhi, India

Grundy and Kabaila (1963) is still accurate to an adequate degree. Azkune et al. (2010) studied the shore removal pattern to check shore overloading. They showed that the overloads occurring due to early removal of shores could cause failure of the structure. Hence, it becomes necessary to maintain a certain factor of safety (FoS) while reducing the cost of the formwork.

It is observed that very little literature is available, where the cost of the formwork is considered as a part of the discussion. The researchers considered the total formwork cost as one of the components of equation to minimize the total cost. However, the study related to a number of shore–reshore levels for the optimum cost of construction is not reported in the literature. Hence, this study aimed to find the optimum number of levels of shores and reshores that should give the minimum cost of construction ensuring the safety of construction. The cost of construction is based on the grade of concrete used, the levels of shore/reshore formwork, and the overhead charges as the cost of execution of the construction work. The relationship is developed between these components to find the minimum cost of construction. A genetic algorithm is integrated with MATLAB to find the optimized cost for different combinations of levels of shores/reshores. The program is developed for slab formwork used for a multistorey building construction. The study is carried out for 20-, 40-, and 60-storey building to find out the cost of RC construction considering different constraints.

2 Literature review

Grundy and Kabaila (1963) proposed the simplified method to calculate load transfer among slabs, shores, and reshores. The strength gained by slabs at any particular time of construction was compared to the load applied on slabs to check the safety of construction. Jha (2011) provided an illustration of calculation of loads on slabs and shore/reshore based on the simplified method. Liu et al. (1988) provided a method for load distribution on slabs and shores/reshores during the construction. The actual rigidity of shores/reshores and stiffness of slabs depend on time variation.

Coello et al. (1997) used a genetic algorithm for the design of RC beams. The traditional method uses trial-and-error method for beam design; researchers used the mechanism of natural selection to optimize the design. The cost of the rectangular beam was minimized considering the cost of concrete, steel, and shuttering. Hanna and Senouci (1995) derived the equation for minimum cost of slab forms. The material and labor costs of sheathing, joist, stringer, and wooden shores were considered in the equation for the development of cost optimization equation for slab forms. The results were compared to the traditional design method that calculated the distance between sheathing, joist, stringer, and wood shores using design tables and charts. It showed that the saving in the cost of slab form using the optimization programs was 9.9% compared to the traditional methods of design. Senouci and Al-Ansari (1996) presented an optimization method for concrete slab forms. The cost was minimized while ensuring safe slab form design. The costs for sheathing, joist, stringer, shore, and labor were considered for the optimization equation. The cost was reduced up to 21%. The same equation was used by other researchers (Kaveh and Shakouri Mahmud Abadi, 2010; Al-Tabtabai, 2000) to optimize the cost of a concrete slab formwork. The same equation was used by Kaveh and Behnam (2012) to optimize the cost of the composite floor system. The optimization was carried out using charged system search algorithm.

Al-Tabtabai et al. (1999) designed slab formwork considering the optimization of cost. A genetic algorithm was used as an optimization tool. The optimization function was formed considering the cost of sheathing, joists, stringers, shores, and slab volume. The number and spacing of each formwork component were considered to find the total cost of formwork. Malasri et al. (1994) used a genetic algorithm for the design of RC beams to find the sufficient and best cross-section after several iterations of genetic algorithm process. The costs of labor and material for concrete, reinforcement, and formwork were considered for the optimization problem. These parameters were examined by many researchers with different optimization tools to minimize the total cost of construction (Sahab et al., 2005). Similarly, the costs of concrete, steel, and formwork were minimized to lower the cost of a concrete beam (Rao, 1973).

Literature review shows that the works have been carried out by researchers to understand the load distribution between the slab and shore/reshore. But a specific method is not available to find the optimum number of levels of shores and reshores for a particular number of levels of high-rise building considering the cost aspect. This study aimed to compare the cost of different levels of shores and reshores for a high-rise building using the genetic algorithm. The genetic algorithm is incorporated into an already developed MATLAB program by the authors. Hence, a numerical study is conducted to validate the MATLAB program.

3 Numerical study

A MATLAB program is developed to calculate the distribution of construction loads between slabs, shores, and reshores. This program is based on the construction sequence given by ACI 347.2R-05 (2005) for the calculation of load and strength of slabs at a different time during construction. A numerical study is conducted to check the load on slab and shoring/reshoring for various combinations of levels of shores and reshores. The strength of the RC slab is calculated using the formula given by ACI Committee 209R (1997) as shown in Eq. (1).

$$\left(f_c^{'}\right)t=\frac{t}{a+\beta t}\left(f_c^{'}\right)_{28} \tag{1}$$

where $\left(f_c^{'}\right)t$ is the compressive strength of concrete at time t (in day) after casting, $\left(f_c^{'}\right)_{28}$ is the 28-day characteristic compressive strength of concrete, and a and β are constants depending on the type of cement and curing method used. The safety of construction is checked at each level of construction sequence. The results are compared to manual calculation. It is found that the program results match with the manual calculation. Hence, the program output is considered valid for different levels of shoring and reshoring. The applied load and the strength gained by the slab are plotted with respect to time through MATLAB program. The examples given by Jha (2011) and the examples from ACI 347.2R-05 (2005) are solved using this MATLAB program. The program gives the same output values as given in these two references. From the program, it is observed that when the difference between strength gained by the slab and load applied on the slab is minimized, the cost of construction is also minimized. When the strength is less than the applied load, the slab becomes unsafe. Few modifications can be performed to have safe slabs. For example, the stripping time can be increased; hence, the slab gets more time to gain the strength. The higher grade of concrete can be used for construction. If more number of levels of shores/reshores is used, then the construction becomes safe. All these options though make the construction safe; they increase the cost of construction. The objective of this study was to reduce the cost of construction with respect to a number of levels of shoring/reshoring ensuring the safety of construction. Hence, a fitness function is derived to minimize the cost of construction ensuring safety during construction.

4 Objective function

The objective of this study was to minimize the cost of shoring, reshoring, and concrete for the construction of a high-rise building considering safety condition. A MATLAB program is developed to find the load transfer between slabs and shoring/reshoring based on the simplified method. This program is modified to find the economical solution for a multistorey building for a particular number of levels of shores and reshores. The fitness function is defined considering the number of levels of shores, the number of levels of reshores, and grade of concrete as shown in Eq. (2).

$$y=f\left(ns\right)+f\left(nr\right)+f\left(f_{\mathrm{ck}}\right)+OC+PC \tag{2}$$

where y is the fitness function value, ns is a number of levels of shores, nr is a number of levels of reshores, f_{ck} is the characteristic strength of concrete, OC is the overhead charges during construction, and PC is the penalty cost. The life span of a shore is considered as 5 years. The cost of renting the shores is considered to calculate the total shoring cost. The cost of a level of reshores is considered as 20% of the cost of one level of shores. The cost of concrete is taken from Delhi Schedule of Rates (2016) for different grades of concrete. The overhead charges are calculated from the salary of engineer, foreman, and charge hand required to execute the construction of the multistoreyed building. A penalty cost of 2,000 USD is added to check for any infeasible solution. This addition differentiates the infeasible solutions from the final set of solutions. The FoS is taken as the ratio of strength gained by slab to the load on the slab at any instance of time. It should always be more than a specified expected FoS. The FoS values lesser than 1.3 should be treated as unsafe (ANSI10.9, 2003). The expected FoS is considered as 1.5 in the current study. The actual FoS is calculated at each stage of construction. If it is less than this value at any point of time, then the condition is treated as unsafe and a fixed cost is added as a penalty cost to the total cost y. If the FoS is greater than the specified value, then it is a safe condition, and the penalty cost remains zero. The overhead charges for a day are calculated considering the wages to be paid to the workman required to execute the construction work. In this study, it is assumed that two engineers, two foremen, and one charge hand are required for execution and monitoring work. The total wages due to them are 225 USD/day. The slab area is considered as 200 m^2 with a thickness of 0.2 m.

The methodology used for cost optimization is shown in Figure 1. A genetic algorithm is used to optimize the fitness

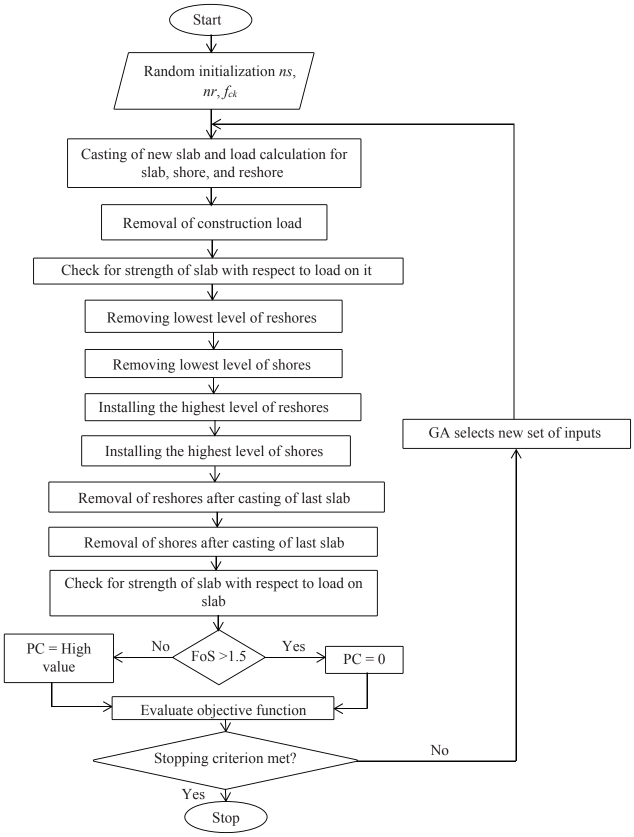

Start
Random initialization ns, nr, f_{ck}
Casting of new slab and load calculation for slab, shore, and reshore
Removal of construction load
Check for strength of slab with respect to load on it
Removing lowest level of reshores
Removing lowest level of shores
Installing the highest level of reshores
Installing the highest level of shores
GA selects new set of inputs
Removal of reshores after casting of last slab
Removal of shores after casting of last slab
Check for strength of slab with respect to load on slab
PC = High value
No
FoS >1.5
Yes
PC = 0
Evaluate objective function
Stopping criterion met?
No
Yes
Stop

Fig. 1: Methodology for cost optimization.

function. The input parameters, i.e., *ns*, *nr*, and f_{ck}, are randomly initialized in the beginning through the genetic algorithm code within the given upper and lower bounds which are integer variables. The ns and nr values are defined based on the number of storeys, *n* value. These sets of values are evaluated in the MATLAB program. The FoS is checked simultaneously for each level of construction, and the PC is decided at the same level. The objective function is evaluated for the set of chosen input values. The individuals of next generation are selected based on the fitness value. These steps are repeated till the stopping criterion is met.

The installation time is taken as 1 day. The cycle time is calculated from installation time and stripping time. The stripping time is the function of the grade of concrete. The stripping time is calculated from Eq. (3).

$$t_s = \frac{aD}{f_{ck} - \beta D} \qquad (3)$$

where t_s is the stripping time, a and β are constants and depend on the type of cement and curing method, f_{ck} is the characteristic strength of concrete, and D is the self-weight of concrete. The maximum value of t_s is obtained with f_{ck} as 20 N/mm^2. The cycle time, T, is the summation of stripping time and installation time. The maximum cycle time is taken as 30 days as given in Eq. (4).

$$T = \left(1 + t_s\right) \qquad (1 \leq T \leq 30) \qquad (4)$$

The program checks all the combination of levels of reshores for a particular number of levels of shores. The cost of each case is calculated considering different constraints.

The constraints for fitness function involve the range of variables. The constraints for the *n* storey building are as follows:

1. The number of levels of shores (*ns*) can range from at least one level of shoring to one minus as many levels as the number of the storey (1≤*ns*≤*n*–1).
2. The number of levels of reshores (*nr*) can range from at least one level of reshoring to one minus as many levels as the number of the storey (1≤*nr*≤*n*–1).
3. The sum of a number of shoring and reshoring levels can attain a maximum equal to the number of storeys. The minimum can be two as at least one level of shoring, and one level of reshoring is allowed (2≤[*ns*+*nr*]≤*n*).
4. The characteristic strength of concrete is taken in the range of 20–50 N/mm^2 with an increase of 5 N/mm^2.

The MATLAB program runs with the algorithm as shown in Figure 1, and the cost output is obtained in the form of excel sheet for the considered *n* storey building giving a cost for all possible combinations of the grade of concrete and the levels of shores and reshores.

5 Results and discussion

Different shore–reshore combinations are checked with various grades of concrete to find the lowest cost for a multistorey building construction. As the cost of shores is more than the cost of reshores, the maximum cost for any *n* storey building is obtained with maximum possible levels of shores and minimum possible levels of reshores. The minimum FoS for the load is considered as 1.3 or 1.4 (Jha, 2011; ANSI10.9, 2003). Hence, an FoS of 1.5 is assumed to ensure the safety during the construction phase. The penalty cost is zero when the specified FoS condition is fulfilled. When the FoS condition is violated, a penalty cost is added to the function, making the total cost a very high value. The cost is considered as an economically undesirable condition when the high penalty cost is added to the total cost for a safety violation.

The input parameters are randomly initialized as shown in Figure 1 in the genetic algorithm. These inputs are given to the MATLAB program developed based on the construction sequence. The cycle time, *T*, is calculated using installation time and stripping time. The FoS is checked for all completed slabs. If the safety condition is violated even for a single slab, then the penalty cost is added to the total cost. The cycle time is increased by 1 day, and the safety of construction is checked for all slabs. This procedure is repeated until the maximum cycle time is reached. After this, the genetic algorithm selects a new set of inputs from the given range, and the entire procedure is repeated. The program is used to find the optimum cost for 20-, 40-, and 60-storey building.

5.1 Application of model on a 20-storey building

Different levels of shores and reshores are considered for cost calculation for a 20-storey building keeping required FoS as 1.5. Figure 2 shows the plot of cost for all the different combinations of levels of shores and reshores for an M20 grade of concrete. The cost for different combinations of shore and reshore levels is calculated using Eq. (2). Consider M20 grade of concrete with *n*=20.

The cost of 1S–16R (one level of shore and 16 levels of reshore) gives the optimum cost as 73332.3 USD with a cycle time of 11 days as shown in Figure 2. This cost is calculated as explained further. The cost of a shore level is a function of *T*, so the cost increases with the increase in *T*. The slab area is considered as 200 m^2 as explained earlier. Therefore, it is estimated that each shore serves the area of 2 m^2. Hence, a total of 100 shores are necessary for the considered slab area. The total cost of one level of shores comes out to be 16.5 USD in this case. The cost of a level of reshores is expressed as 20% of the cost of one level of shores.

Hence, the cost of one reshore level comes out to be 3.3 USD. Therefore, in this study, the cost of 16 levels of reshores is 52.8 USD. The area of the slab is 200 m^2, and its thickness is considered as 200 mm. Therefore, the total volume for slab concreting is 40 m^3. The cost of concrete is calculated for 40 m^3. The cost of M20 concrete is 88.485 USD/m^3 (DSR, 2016) of concrete excluding the labor charges. The total cost for M20 concrete is 70,788 USD for a 20-storey building. It is assumed that two engineers, two foremen, and one charge hand are necessary for the execution and monitoring work. The salary for an engineer is considered as 1,500 USD/month, for a foreman as 1,050 USD/month, and for a charge hand as 750 USD/month. The total wages for two engineers, two foremen, and one charge hand are 6,600 USD/month. The total wages due to them are 225 USD/day. The overhead charges for 11 days are 2,475 USD. The FoS condition is fulfilled in this case, and hence the *PC* value remains zero. A moderate cost of 73,335.6 USD is obtained for the 1S–17R combination for the same cycle time. All other shore/reshore combinations violate the specified safety condition for an M20 grade of concrete with a cycle time of 11 days. The MATLAB program calculates the cost for each combination of the shore–reshore level with different cycle times and checks the specified FoS throughout the construction sequence.

Figure 3 shows different combinations of shore–reshore levels for a 20-storey building with different grades of concrete from M25 to M50. All the reshore combinations for 1S–13S give the economically undesirable cost for M25 concrete. The case of 14S–2R gives the optimum cost with a cycle time of 4 days. It is observed that all other combinations of

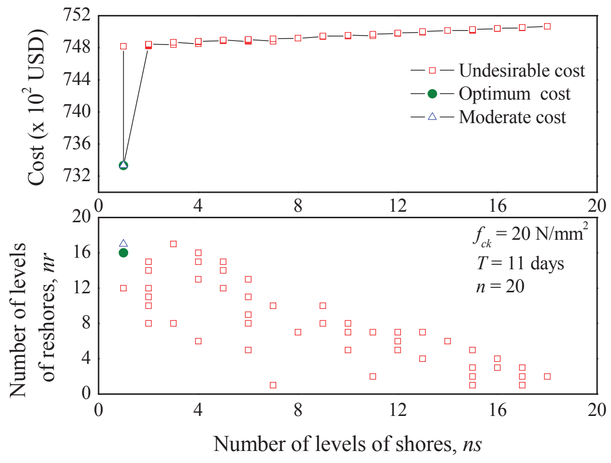


Fig. 2: Cost for different combinations of shore/reshore levels for M20 grade of concrete.

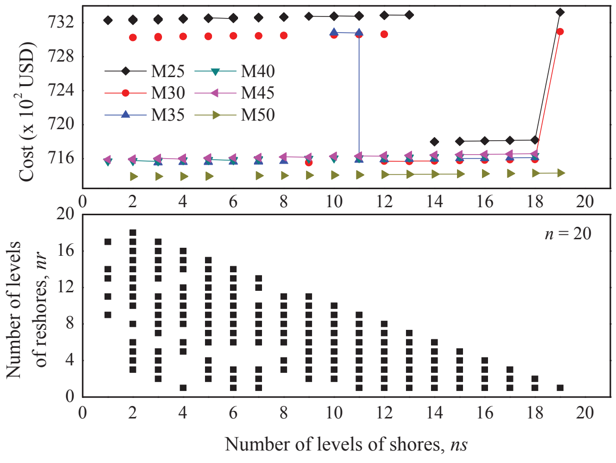


Fig. 3: Cost for different combinations of shore/reshore levels for a 20-storey building.

reshore levels for 14S–18S give moderate cost which is more than the optimum cost, but it does not violate safety requirement. But, the case of 19S–1R again violates the safety condition making the total cost very high. The case of 10S–10R gives economically undesirable case, as the penalty cost gets added to the total cost making it too high. In the case of M30 concrete, all the reshore combinations for the shore level up to 8 show economically undesirable condition. The case of 9S–5R gives the optimum cost of 71,553 USD with a cycle time of 3 days. Other reshore combinations for 9S give the moderate cost. All the reshore combinations for 10 and 11 levels of shore are economically undesirable as it violates the specified FoS. The reshore combinations for 13–18 shore levels are having moderate cost, but the highest level of shore again shows economically undesirable case. For M35 concrete, the case of 3S–8R gives the optimum cost of 71,551.2 USD with a cycle time of 3 days. All the combinations of reshores for 10 levels of shore are undesirable with respect to safety. Similarly, 11S–1R also violates safety condition making the total cost very high. All other combinations of shore and reshore levels are having moderate cost with the desired safety factor.

The case of 1S–11R gives the optimum cost of 71,567.4 USD with a cycle time of 3 days for M40 concrete. All the combinations of shore and reshore levels maintain the FoS of 1.5 with the cycle time of 3 days. Hence, the economically undesirable case is not observed in the case of M40 concrete. The case of 2S–3R gives the optimum cost of 71,587.2 USD with a cycle time of 3 days for M45 concrete. Similar to the case of M40, all the combinations of shore and reshore levels maintain the FoS of 1.5 with the cycle time of 3 days. Hence, the economically undesirable case is not observed. The case of 4S–5R gives the optimum cost of 71,388 USD with a cycle time of 2 days for M50 concrete. All the combinations of shore and reshore levels maintain the FoS of 1.5.

The optimum cost for an M50 grade of concrete is the lowest among all the grades of concrete considered for a 20-storey building. The main reason here is that the cycle time is the lowest for M50 grade of concrete. Therefore, cycle time plays an important role in the total cost of RC construction. But, at the same time, it cannot be lowered randomly as the safety of the structure during construction is also important. Hence, this computer program is useful to achieve the optimum cost of RC construction while maintaining the safety of the structure.

5.2 Application of model on a 40-storey building

Tab. 1: Optimum cost for a 40-storey building for different concrete grades.

Grade of concrete	Optimum cost (USD)	Number of levels of shore	Number of levels of reshore	Cycle time (day)
M20	142,947	31	9	5
M25	142,691.4	28	2	4
M30	142,408.5	10	25	3
M35	142,417.5	2	25	3
M40	142,454.4	1	21	3
M45	142,501.2	3	13	3
M50	142,313.4	1	24	2

Table 1 summarizes the combination of shore and reshore levels giving the optimum cost for different grades of concrete for a 40-storey building. It takes 5 days to reach the optimum value for M20 concrete. The case of 31S–9R gives the optimum cost as 142,947 USD for M20. The cycle time is reduced to 4 days for M25 concrete to get the optimum cost. The combination of 28S–2R gives the optimum cost of 142,691.4 USD. The optimum cost for M25 is lesser than the cost of reinforced cement concrete (RCC) construction for M20 concrete, because the total number of formwork levels is less in the case of M25 concrete and the cycle time is also reduced by 1 day. As the cycle time decreases, the cost of construction decreases, because the cost of formwork and overhead charges is the function of cycle time.

The optimum cost is observed with 3-day cycle time for the M30–M45 grade of concrete as summarized in Table 1. The cost increases from M30 to M45 as the higher concrete grade is used. The total number of formwork level decreases with the same cycle time. For M40 concrete, the required total formwork levels are 22, whereas for M45 concrete, the formwork levels are 16 only. Hence, the total required formwork levels reduce with the increase in the grade of concrete. For M45 concrete, the case of 1S gives safer construction similar to M40 concrete, but that is not the optimum cost. The cost of 3S gives the lowest cost of construction for M45 concrete. Hence, the combination of 3S–13R is considered. The increase in the cost is due to higher concrete grade. The optimum cost for M50 concrete is observed as 142,313.4 USD with 1S–24R having a cycle time of 2 days. This is the lowest cost for the 40-storey RC construction as the lowest cycle time of 2 days is required to maintain the given FoS compared to other grades of concrete.

The figures showing the results of shore/reshore level combinations and the respective cost value are drawn for all the concrete grades. The table for all concrete grades with optimum cost values only is included due to space limitation.

5.3 Application of model on a 60-storey building with fixed reshore level

Now, the comparison of cost is made for *n* number of levels of shores keeping reshore levels fixed as 1 as shown in Figure 4. All the grades of concrete are considered for comparison. These values are calculated with 5-day cycle time.

The cost is economically undesirable for M20 up to 46 levels of shores, and then it becomes safe up to 57 levels leading toward undesirable condition for the remaining two levels, whereas with M25, the initial undesirable value changes at 39 levels of shores to a safe value. It continues to be safe up to 59 levels. For M30 concrete, the initial undesirable condition becomes safe at 10 levels of shores and continues to be safe up to 29 levels. It again becomes economically undesirable for 30–34 levels of shores and becomes safe thereafter.

For M35 concrete, the initial safe condition becomes economically undesirable for 30 and 31 levels of shores and becomes safe thereafter. For M40, M45, and M50 concrete, all the levels of shores follow safety condition and give the total cost with zero penalty cost. The lower concrete grades violate the safety condition when the shore levels are 30, i.e., (*n*/2).

5.4 Application of model on a 60-storey building with fixed shore level

The cost comparison is made for *n* number of reshores for a 60-storey building as shown in Figure 5, keeping shore levels fixed as one level only. These values are calculated with 5-day cycle time.

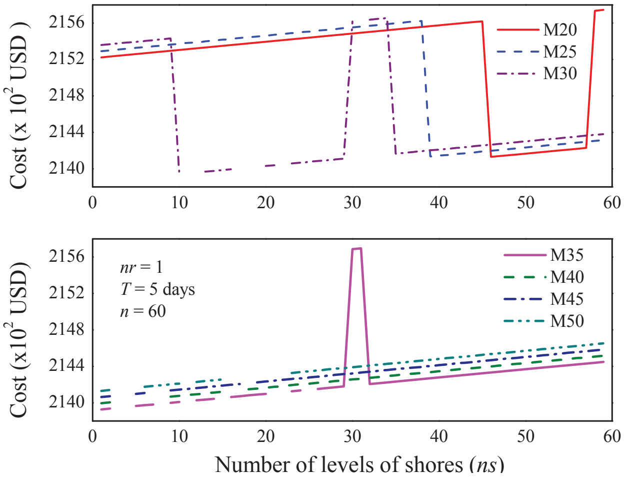


Fig. 4: Comparison of cost for different number of levels of shores for a 60-storey building.

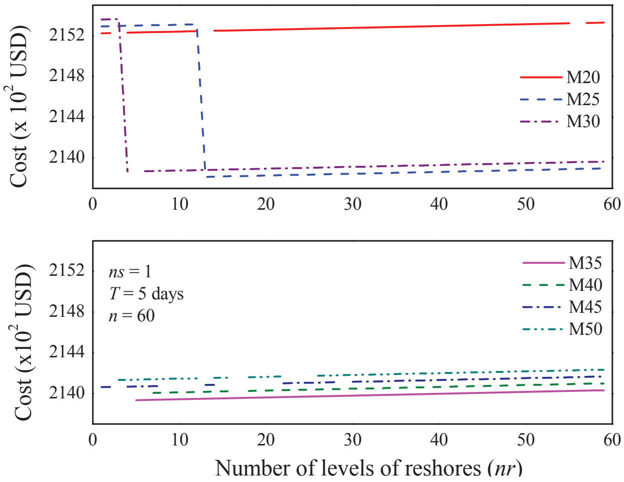


Fig. 5: Comparison of cost for different number of levels of reshores for a 60-storey building.

The cost is calculated for M20–M50 concrete. For M20 concrete, all the combinations of reshores are economically undesirable violating the FoS of 1.5. In the case of M25 concrete, 1–12 levels do not follow the safety condition resulting in undesirable cost. All the further combinations up to 59 levels give cost with an acceptable safety factor. It is an undesirable case in the beginning for M30 concrete till three levels of reshores only. It becomes safe from fourth level and remains safe till the 59th level of reshores. For M35, M40, M45, and M50 concrete, it is observed that all the combinations of reshore levels match with the required safety factor of 1.5 and give an acceptable cost for each combination.

5.5 Variation of FoS to find optimum cost

The FoS used for all the previous calculations is 1.5. Few researchers have recommended the minimum FoS as 1.3 or 1.4 (Jha, 2011; ANSI10.9, 2003). The IS 456:2000 (2000) gives an FoS of 2.0 for working conditions. Hence, a study is carried out to see the effect of variation of required FoS with respect to shore/reshore levels for different grades of concrete.

Table 2 summarizes the results of a 10-storey building for different safety factors. It is observed that the number of levels of shores and reshores giving the optimum cost is more along with the greater cycle time for FoS of 2.0 compared to other FoS values. The optimum cost condition with 5-day (lowest) cycle time is not met in the case of M20 concrete for FoS of 1.4, 1.5, and 2.0. It gives economically undesirable cases for all the combinations of shore and reshore levels with 5-day cycle time for the higher safety factor.

For M20 concrete, the lowest cost combination is given for 1.3 FoS only with 5-day cycle time. With FoS of 2.0, M25 concrete gives the optimum cost with 10-day cycle time. Similarly, the optimum cost for different grades of concrete is tabulated by varying the FoS. The combination of shore/reshore levels is also given for each concrete grade.

Table 3 summarizes the results of a 20-storey building for different safety factors. It is observed that the number of levels of shores and reshores giving the optimum cost is having maximum cycle time for FoS of 2.0 compared to other FoS values. It takes 30- and 23-day cycle time for M20 and M25 concrete, respectively, to maintain the FoS condition. The cycle time of 9 and 11 days is required to obtain the optimum cost with FoS of 1.4 and 1.5 for M20 concrete. It is observed that the total number of levels of formwork required increases with the increase in the FoS when the cycle time is same.

The results of a 40-storey building giving the combination of shore–reshore levels for the optimum cost with different FoS values are summarized in Table 4. It shows that a higher number of levels of formwork are required for M20 concrete for three FoS values with the same cycle time of 5 days. The FoS of 2.0 needs 30-day cycle time for M20 concrete. The lesser cycle time results in safety violation in this case. All the other concrete grades give the combination for the optimum cost with 4 or lesser number of days as cycle time for M30 and above grades of concrete.

Table 5 gives the results of a 60-storey building for different values of FoS for all the considered grades of concrete. The FoS of 2.0 shows the optimum cost at a cycle time of 25 days for M20 concrete. All the other safety factor values take less than 6 days to maintain the safety of the construction while showing the optimum cost for the grades of concrete. Higher grades of concrete up to M45 take 3-day cycle time. For M50 grade of concrete, the lowest cycle time of 2 days is required to have optimum cost with different combinations of shore and reshore levels. The required levels of formwork increase with an increase in the FoS value. More levels of shores are required by the slab to attain the desired strength with higher safety requirement.

Tab. 2: Combination of levels of shores and reshores for a 10-storey building.

Grade of concrete	FoS			
	1.3	1.4	1.5	2.0
M20	8 (level of shore), 2 (level of reshore) (T = 5) (USD 36,582)	1, 9 (T = 7) (USD 36,998.4)	1, 8 (T = 13) (USD 38,369.7)	1, 8 (T = 28) (USD 41,803.2)
M25	1, 9 (T = 4) (USD 36,322.05)	8, 2 (T = 4) (USD 36,355.65)	8, 2 (T = 4) (USD 36,355.65)	1, 9 (T = 10) (USD 37,697.25)
M30	1, 9 (T = 3) (USD 36,104.1)	2, 8 (T = 3) (USD 36,107.7)	7, 3 (T = 3) (USD 36,125.7)	1, 9 (T = 7) (USD 37,020.9)
M35	1, 7 (T = 3) (USD 36,113.55)	1, 8 (T = 3) (USD 36,114.45)	2, 8 (T = 3) (USD 36,118.95)	3, 6 (T = 3) (USD 36,121.65)
M40	1, 6 (T = 3) (USD 36,123.9)	1, 4 (T = 3) (USD 36,122.1)	1, 7 (T = 3) (USD 36,124.8)	2, 5 (T = 3) (USD 36,127.5)
M45	1, 3 (T = 3) (USD 36,132.45)	1, 3 (T = 3) (USD 36,132.45)	1, 3 (T = 3) (USD 36,132.45)	2, 4 (T = 3) (USD 36,137.85)
M50	1, 4 (T = 2) (USD 35,916.9)	1, 7 (T = 2) (USD 35,918.7)	1, 8 (T = 2) (USD 35,919.3)	2, 5 (T = 2) (USD 35,920.5)

FoS, factor of safety.

Tab. 3: Combination of levels of shores and reshores for a 20-storey building.

Grade of concrete	FoS			
	1.3	1.4	1.5	2.0
M20	14 (level of shores), 6 (level of reshore) (T = 5) (USD 72,027)	1, 14 (T = 9) (USD 72,864.3)	1, 16 (T = 11) (USD 73,332.3)	1, 5 (T = 30) (USD 77,628)
M25	3, 17 (T = 4) (USD 71,748.9)	13, 4 (T = 4) (USD 71,793.3)	14, 2 (T = 4) (USD 71,796.9)	1, 5 (T = 23) (USD 76,054.5)
M30	3, 8 (T = 3) (USD 71,528.7)	2, 18 (T = 3) (USD 71,533.2)	9, 5 (T = 3) (USD 71,548.95)	9, 10 (T = 3) (USD 71,557.5)
M35	3, 4 (T = 3) (USD 71,547.6)	2, 6 (T = 3) (USD 71,544.9)	3, 8 (T = 3) (USD 71,551.2)	2, 18 (T = 3) (USD 71,555.7)
M40	3, 2 (T = 3) (USD 71,568.3)	2, 5 (T = 3) (USD 71,566.5)	1, 11 (T = 3) (USD 71,567.4)	2, 9 (T = 3) (USD 71,570.1)
M45	2, 2 (T = 3) (USD 71,586.3)	2, 2 (T = 3) (USD 71,586.3)	2, 3 (T = 3) (USD 71,587.2)	2, 8 (T = 3) (USD 71,591.7)
M50	4, 3 (T = 2) (USD 71,386.8)	4, 4 (T = 2) (USD 71,387.4)	4, 5 (T = 2) (USD 71,388)	4, 7 (T = 2) (USD 71,389.2)

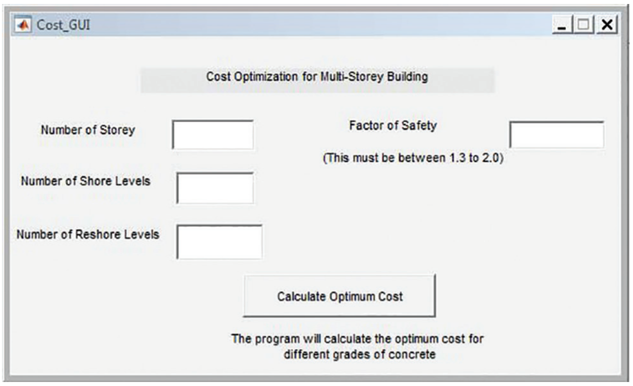
FoS, factor of safety.

5.6 User interface

This MATLAB program is useful for the designers and practitioners to determine the optimum level of shore/reshore with considered level of safety. Figure 6 shows a graphical user interface (GUI) that can be used by the user. The user has to enter the number of storeys, number of shore levels, and number of reshore levels. These values must be positive; otherwise, an error message is popped up. The shore and reshore level values must be at least 1. The FoS value between 1.3 and 2.0 is required. After giving all the inputs, the user has to press "Calculate Optimum Cost" button. The program will calculate the cost of RC construction with different grades of concrete. It gives the cost combinations as an output in an excel file from which one can decide about the suitable combination.

5.7 Discussion

The optimum use of formwork for the multistory building construction is the need of time. The levels of formwork are decided considering the quality, safety, and economic aspect. The researchers have tried to figure out the cost of construction by considering the cost of total formwork as one of the parameters. This study aimed to optimize the cost of construction by optimizing the shore/reshore levels for a slab formwork considering safety and economic aspect. A genetic algorithm is used to find



(This must be between 1.3 to 2.0)

The program will calculate the optimum cost for different grades of concrete

Fig. 6: GUI for the users to find the optimum cost.
GUI, graphical user interface.

Tab. 4: Combination of levels of shores and reshores for a 40-storey building.

Grade of concrete	FoS			
	1.3	1.4	1.5	2.0
M20	28 (level of shores), 7 (level of reshore) (T = 5) (USD 142,921.5)	29, 11 (T = 5) (USD 142,935)	31, 9 (T = 5) (USD 142,947)	1, 10 (T = 30) (USD 148,461)
M25	6, 33 (T = 4) (USD 142,596.6)	8, 21(T = 4) (USD 142,594.2)	28, 2 (T = 4) (USD 142,691.4)	1, 5 (T = 30) (USD 148,461)
M30	2, 18 (T = 3) (USD 142,366.2)	4, 19 (T = 3) (USD 142,376.1)	10, 25 (T = 3) (USD 142,408.5)	18, 18 (T = 3) (USD 142,438.2)
M35	4, 11 (T = 3) (USD 142,413.9)	2, 20 (T = 3) (USD 142,413)	2, 25 (T = 3) (USD 142,417.5)	4, 14 (T = 3) (USD 142,416.6)
M40	1, 12 (T = 3) (USD 142,446.3)	1, 15 (T = 3) (USD 142,449)	1, 21 (T = 3) (USD 142,454.4)	6, 15 (T = 3) (USD 142,471.5)
M45	3, 10 (T = 3) (USD 142,498.5)	3, 10 (T = 3) (USD 142,498.5)	3, 13 (T = 3) (USD 142,501.2)	4, 7 (T = 3) (USD 142,500.3)
M50	5, 9 (T = 2) (USD 142,316.4)	6, 7 (T = 2) (USD 142,318.2)	1, 24 (T = 2) (USD 142,313.4)	4, 12 (T = 2) (USD 142,315.2)

FoS, factor of safety.

Tab. 5: Combination of levels of shores and reshores for a 60-storey building.

Grade of concrete	FoS			
	1.3	1.4	1.5	2.0
M20	44 (level of shores), 1 (level of reshore) (T = 5) (USD 213,820.5)	48, 8 (T = 5) (USD 213,861)	48, 11 (T = 5) (USD 213,865.5)	1, 56 (T = 25) (USD 218,446.5)
M25	4, 54 (T = 4) (USD 213,420.3)	28, 6 (T = 4) (USD 213,506.7)	39, 8 (T = 4) (USD 213,575.1)	52, 2 (T = 4) (USD 213,645.9)
M30	6, 12 (T = 3) (USD 213,211.8)	5, 27 (T = 3) (USD 213,220.8)	9, 44 (T = 3) (USD 213,254.1)	29, 1 (T = 3) (USD 213,305.4)
M35	5, 20 (T = 3) (USD 213,282)	7, 14 (T = 3) (USD 213,285.6)	3, 43 (T = 3) (USD 213,293.7)	7, 34 (T = 3) (USD 213,303.6)
M40	1, 18 (T = 3) (USD 213,329.7)	1, 23 (T = 3) (USD 213,334.2)	1, 30 (T = 3) (USD 213,340.5)	4, 31 (T = 3) (USD 213,354.9)
M45	1, 14 (T = 3) (USD 213,393.6)	1, 20 (T = 3) (USD 213,399)	1, 24 (T = 3) (USD 213,402.6)	2, 20 (T = 3) (USD 213,403.5)
M50	2, 11 (T = 2) (USD 213,213.6)	2, 15 (T = 2) (USD 213,234)	2, 32 (T = 2) (USD 213,244.2)	3, 31 (T = 2) (USD 213,246.6)

FoS, factor of safety.

the optimum cost of construction considering different grades of concrete for a specified FoS. As the cycle time increases, the cost of construction increases. The cost is reduced when the total number of formwork levels is reduced keeping all other parameters constant. A comparative study is carried out to see the effect of constant level of shores for various levels of reshores. It is observed that the case of constant shore level with numerous reshore levels gives better results compared to numerous levels of shores with constant reshore level. The cost of construction increases with the increase in shore levels, whereas the increase in cost is lesser for increased reshore levels. The slabs get more time for strength gain with reshores in place, and the safety of structure during construction can be assured. For the onsite construction, this program can be used to decide formwork levels for slabs to get an idea about the lowest cost of construction. A user interface has been developed which asks user the number of storeys, number of shore/reshore levels, and the expected FoS. By providing these data, the user gets the cost combination for different concrete grades (M20–M50) with different cycle times in the form of an excel file. The program also gives the cycle time that assures the safety during construction. Hence, the formwork can be shifted to the upper levels for slab casting which again saves the cost. By using this program, one can decide about the lowest levels of shore/reshore that gives optimum cost of construction without affecting the safety for a considered grade of concrete.

6 Conclusions and future scope

A study is conducted to check the optimum number of levels of shores and reshores that give the optimum cost ensuring the safety of construction. It is observed that economically undesirable condition is obtained for some combinations of shore and reshore levels, especially when the M20–M30 grade of concrete is used. Hence, higher cycle time is required for these concrete grades to maintain the specified FoS. The combinations in which the levels of shores or reshores are equal to the half of the number of stories ($n/2$) are economically undesirable for M20 to the M35 grade of concrete. The safe combination of shore–reshore levels is obtained with a higher cycle time in some cases for M20 concrete. The cost of construction increases with the increase in the cycle time. The case of one level of shores and numerous levels of reshores is better than one level of reshores and numerous levels of shores. As the cost of the level of reshores per floor is lesser than the cost of the level of shores per floor, the overall cost remains less. At the same time, a number of levels of reshore help slabs to gain the adequate strength before it becomes functional and ensures better safety of slabs during construction.

The program considers only slab formwork for the optimum cost with the safety of the construction. This can be further modified by adding the cost of formwork for other structural members to get the total cost of construction for a high-rise building. This program can be added with the cost of other members such as steel where composite sections are used in the construction.

References

ACI 209R-92. (1997). *Predictions of Creep, Shrinkage and Temperature Effects in Concrete Structures*. Committee 209 Report, American Concrete Institute, Farmington Hills, MI, USA, p. 4.

ACI 347.2R-05. (2005). *Guide for Shoring/Reshoring of Concrete Multistory Buildings.* Committee 347 Report, American Concrete Institute*,* Farmington Hills, MI, USA, p. 9*.*

Al-Tabtabai, A. L., Alex, H. A. P., & Jemes, R. (1999). Slab formwork design using genetic algorithm. *Journal of National Research Council Canada, 8*(3), pp. 2407-2418.

Al-Tabtabai, H. M. (2000). Genetic algorithm optimal model for slab formwork cost and design. *Engineering Journal of the University of Qatar, 13*, pp. 177-192.

ANSI10.9. (2003). *Safety Requirements for Concrete and Masonry Work – American National Standard for Construction and Demolition Operations*, p. 11.

Azkune, M., Puente, I., & Santilli, A. (2010). Shore overloads during shoring removal. *Engineering Structures, 32*, pp. 3629-3638.

Beeby, A. W. (2001). The forces in backprops during construction of flat slab structures. *Structures and Buildings, 146*(3), pp. 307-317.

Calderón, P. A., Alvarado, Y. A., & Adam, J. M. (2011). A new simplified procedure to estimate loads on slabs and shoring during the construction of multi-storey buildings. *Engineering Structures, 33*, pp. 1565-1575.

Coello, C. A. C., Christiansen, A. D., & Hernández, F. S. (1997). A simple genetic algorithm for the design of reinforced concrete beams. *Engineering with Computers, 13*, pp. 185-196.

Dongping, F., Haifeng, X., Xiaoming, W., Chuanmin, Z., & Tingsheng, Z. (2009). Load distribution assessment of reinforced concrete buildings during construction with structural characteristic parameter approach. *Tsinghua Science and Technology, 14*(6), pp. 746-755.

DSR 2016. (2016). *Delhi Schedule of Rates*. Central Public Works Department, Government of India, New Delhi, India.

Duan, M. Z., & Chen, W. F. (1995). *Improved Simplified Method for Slab and Shore Load Analysis During Construction*. Purdue University, West Lafayette, IN, USA.

Fang, D. P., Zhu, H. Y., Geng, C. D., & Liu, X. L. (2001). On-site measurements of structural characteristics of reinforced concrete buildings during construction. *ACI Journal, 98*(2), pp. 157-163.

Grundy, P., & Kabaila, A. (1963). Construction loads on slabs with shored formwork in multi-story buildings. *ACI Journal Proceedings, 60*(12), pp. 1729-1738.

Hanna, A. S., & Senouci, A. B. (1995). Design optimization of concrete-slab forms. *Journal of Construction Engineering and Management, 121*(2), pp. 215-221.

IS 456:2000. (2000). *Indian Standard Plain and Reinforced Concrete – Code of Practice*. Bureau of Indian Standards (BIS), New Delhi, India.

Jha, K. N. (2011). *Formwork for Concrete Structures*. Tata McGraw Hill Publisher, Delhi, India.

Kaveh, A., & Behnam, A. F. (2012). Cost optimization of a composite floor system, one-way waffle slab, and concrete slab formwork using a charged system search algorithm. *Scientia Iranica Transactions A: Civil Engineering, 19*(3), pp. 410-416.

Kaveh, A., & Shakouri Mahmud Abadi, A. (2010). Harmony search algorithm for optimum design of slab formwork. *Iranian Journal of Science and Technology Transaction B: Engineering, 34*(4), pp. 335-351.

Liu, X. L., Lee, H. M., & Chen, W. F. (1988). Analysis of construction loads on slabs and shores by personal computer. *American Concrete Institute, 10*, pp. 21-30.

Malasri, S., Halijan, D. A., & Keough, M. L. (1994). Concrete beam design optimization with genetic algorithms. *Proceedings Arkansas Academy of Science, 48*, pp. 111-115.

Nawy, E. G. (2008). *Concrete Construction Engineering Handbook*. CRC Press, Taylor & Francis, FL.

Rao, S. S. (1973). Minimum cost design of concrete beams with a reliability based constraint. *Building Science, 8*, pp. 33-38.

Sahab, M. G., Ashour, A. F., & Toropov, V. V. (2005). Cost optimization of reinforced concrete flat slab buildings. *Engineering Structures, 27*, pp. 313-322.

Senouci, A. B., & Al-Ansari, M. S. (1996). Optimum design of concrete slab forms. *Engineering Journal of the University of Qatar, 9*, pp. 79-93.

Appendix

Appendix A: Main Function for Optimization: Fitness Function

```
function [y]= funname(x)
      global dat;
      a = 4;
      b = 0.85;
      d = 1;
      fa = 0.1;
      fb = 0.01;
      fc = 0.4;
      cap = 1;
      stiffness = 1;

      n = 20;
      k = zeros(1,n+1);
      coff = 1.5;
      PC = 0;
      C_violated = 0;
      ns = x(1);
      nr = x(2);

      for i = 1:1:n
          k(i) = stiffness;
      end
      k(n+1) = 1/0;

      s = zeros(1,n);

      p = zeros(1,n);

      max = zeros(1,n);
```

```
strength = zeros(1,n);

fos=zeros(1,n);

maxs = 0;

maxslab = 0;

time_slab = 0;

maxp = 0;

maxsho = 0;

timeshore = 0;

fck = [2 2.5 3 3.5 4 4.5 5];

f28 = fck(x(3));

ms = ceil((a*d)/(f28 - b*d));

tins = 1;

T = ms + tins;

k = zeros(1,n+1);
for i = 1:1:n
    k(i) = stiffness;
end

k(n+1) = 1/0;

for T = T:1:30
            fs = 0;
            PC = 0;
            i = 0;
for t = T: T: n*T
            time = t;
            i = i + 1;
            sum = 0;
  for l = 1:1:nr+ns
    if ((i-l)>0)
            sum = sum + k(1,i-l);
    elseif ((i-l) == 0)
            sum = k(1, n+1);
            break;
    end
  end
  for j = (i - nr - ns): 1: (i-1);
    if (j > 0)
            s(1,j) = s(1,j) + (k(1,j)/sum) *(1+fc)*d;
    if s(1,j) > maxs
            maxs = s(1,j);
            maxslab = j;
            time_slab = time;
    end
  end
end
for j = i - ns + 1: 1: i
  if (j > 0)
            slabsum = 0;
  for l = j:1:i;
            slabsum = slabsum + s(1,l);
  end
            p(1,j) = (i-j+1)*d + (i-j+1)*fa*d +fc*d - slabsum;
  if p(1,j) >maxp
            maxp = p(1,j);
            maxsho = j;
            time_shore = time;
  end
  end
end
for j = i - ns - nr + 1: 1: i - ns
  if (j > 0)
            slabsum = 0;
  for l = j:1:i;
            slabsum = slabsum + s(1,l);
  end
          p(j) = (i-j+1)*d + ns*fa*d +(i-ns-j+1)*fb*d + fc*d - slabsum;
  if p(1,j) > maxp
          maxp = p(1,j);
          maxsho = j;
          newtime = time;
  end
end
end
    max = s;
    time1 = num2str(time);

if (fc > 0)
        sum = 0;
for l =1:1:nr+ns
if ((i-l) > 0)
                sum = sum + k(1, i-l);
elseif ((i-l) == 0)
                sum = k(1, n+1);
break;
end
end
for j = (i - nr - ns): 1: (i-1);
if (j > 0)
                s(1,j) = s(1,j) - (k(1,j)/sum)*(fc)*d;
if s(1,j) > maxs
                    maxs = s(1,j);
                    maxslab = j;
                    timely = time;
end
end
end
```

```
for j = i - ns + 1: 1: i
if (j > 0)
                slabsum = 0;
for l = j:1:i;
                    slabsum = slabsum + s(1,l);
end
                p(1,j) = (i-j+1)*d + (i-j+1)*fa*d - slabsum;
if p(1,j) > maxp
                    maxp = p(1,j);
                    maxsho = j;
                    newtime = time;
end
end
end
for j = i - ns - nr + 1: 1: i - ns
if (j > 0)
                slabsum = 0;
for l = j:1:i;
                    slabsum = slabsum + s(1,l);
end
                p(j) = (i-j+1)*d + ns*fa*d + (i-ns-j+1)*fb*d - slabsum;
if p(1,j) >maxp
                    maxp = p(1,j);
                    maxsho = j;
                    newtime = time;
end
end
end
end
for q = 1: 1: i
          strength(q) = ((time - T*q) / (a+(b*(time - T*q))))*f28;
end
    min = 1/0;
    minslab = 0;
if(i > 1)
for q = 1: 1: i - 1
            diff = strength(q) - max(q);
if (diff < min)
                min = diff;
                minslab = q;
end
end
        fs = fs + 1;
        fos(fs) = strength(minslab)/max(minslab);

end
if t == n*T;
break;
end
if (t >= ns*T)
        time = t + ms;
end
if (i >= nr + ns)
if(nr > 0)
if (i > (nr+ns))
```

```
                s(1, i - nr - ns) = s(1, i - nr - ns) - p(i - nr - ns +1);
end
for j = i - ns - nr + 1 :1 :i
if (j > 0)
                    sum = 0;
for l = 0:1:nr + ns - 1
if ((i-l)>0)
                            sum = sum + k(1, i-l);
elseif ((i-l) == 0)
                            sum = k(1, n+1);
break;
end
end
                    s(1,j) = s(1,j) + (k(1,j)/sum)*(p(1, i-ns-nr+1) - fb*d);
if s(1,j) > maxs
                        maxs = s(1,j);
                        maxslab = j;
                        time_slab = time;
end
end
end
for j = i - ns + 1: 1: i
if (j > 0)
                    slabsum = 0;
for l = j:1:i;
                        slabsum = slabsum + s(1,l);
end
                    p(1,j) = (i-j+1)*d + (i-j+1)*fa*d - slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        newtime = time;
end
end
end
for j = i - ns - nr + 2: 1: i - ns
if (j > 0)
                    slabsum = 0;
for l = j:1:i;
                        slabsum = slabsum + s(1,l);
end
                    p(j) = (i-j+1)*d + ns*fa*d + (i-ns-j+1)*fb*d - slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        newtime = time;
end
end
end
            p(i - ns - nr + 1) = 0;
            max = s;
end
end
if (i >= ns)
for j = i - ns + 1 :1 :i
```

```
if (j > 0)
                sum = 0;
for l =0: 1: ns-1
if ((i-l)>0)
                        sum = sum +k(1, i-l);
elseif ((i-l) == 0)
                        sum = k(1, n+1);
break;
end
end
                s(1,j) = s(1,j) + (k(1,j)/sum)*(p(1, i-ns+1) - fa*d);
if s(1,j) > maxs
                    maxs = s(1,j);
                    maxslab = j;
                    timely = time;
end
end
end
for j = i - ns + 2: 1: i
if (j > 0)
                slabsum = 0;
for l = j:1:i;
                    slabsum = slabsum + s(1,l);
end
                p(1,j) = (i-j+1)*(1+fa)*d - slabsum;
if p(1,j) >maxp
                    maxp = p(1,j);
                    maxsho = j;
                    newtime = time;
end
end
end
if (nr ==0)
if( i > ns)
                s(i-ns) = s(i-ns) - p(i-ns+1);
end
end
if(nr > 0)
            sum = 0;
for l = ns : 1 : nr+ns-1
if ((i-l)> 0)
                    sum = sum + k(1, i-l);
elseif ((i-l) == 0)
                    sum = k(1, n+1);
break;
end
end
for j= (i - nr - ns + 1): 1: (i - ns)
if (j > 0)
                    s(1,j) = s(1,j) - (k(1,j)/sum) * p(i-ns+1);
if s(1,j) > maxs
                        maxs = s(1,j);
                        maxslab = j;
                        time_slab = time;
end
```

```
end
end
for j = i - ns - nr + 2: 1: i - ns
if (j > 0)
                    slabsum = 0;
for l = j:1:i-ns;
                        slabsum = slabsum + s(1,l);
end
                    p(j) = (i - ns - j + 1)*(1 + fb)*d - slabsum;
if p(1,j) > maxp
                        maxp = p(1,j);
                        maxsho = j;
                        newtime = time;
end
end
end
end
        p(i - ns + 1) = 0;
end
if (i >= ns)
if(nr == 0)
            max = s;
end
if(nr > 0)
if(i < nr+ns)
                max = s;
else
for q = 1: 1: i
if (max(q) < s(q))
                        max(q) = s(q);
end
end
end
end
        time1 = num2str(time);
end
if (i >= ns)
if(nr > 0)
            sum = 0;
for l = ns:1:nr+ns-1
if ((i-l)>0)
                    sum = sum +k(1, i-l);
elseif ((i-l) == 0)
                    sum = k(1, n+1);
break;
end
end
for j = (i - nr - ns+1): 1: (i-ns);
if (j>0)
                    s(1,j) = s(1,j) - (k(1,j)/sum)*(fb)*d; % + sign is used in
time program instead of -
if s(1,j) > maxs
                        maxs = s(1,j);
                        maxslab = j;
                        time_slab = time;
```

```
end
end
end
for j = i - ns - nr + 2: 1: i-ns
if (j > 0)
                    slabsum = 0;
for l = j:1:i-ns;
                        slabsum = slabsum + s(1,l);
end
                    p(1,j) =fb*d + (i-ns-j+1)*(1+fb)*d  - slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        time_shore = time;
end
end
end
            p(1, i-ns+1) = fb*d;
if p(1,i-ns+1) >maxp
                maxp = p(1,i-ns+1);
                maxsho = i-ns+1;
                time_shore = time;
end
end
for q = 1:1:i
if (max(q)< s(q))
                max(q) = s(q);
end
end
end
    sum = 0;
for l =0: 1: nr+ns-1
if ((i-l)>0)
            sum = sum +k(1, i-l);
elseif ((i-l) == 0)
            sum = k(1, n+1);
break;
end
end
for j = (i - nr - ns+1): 1: i;
if (j>0)
            s(1,j) = s(1,j) - (k(1,j)/sum)*(fa)*d;
if s(1,j) > maxs
                maxs = s(1,j);
                maxslab = j;
                time_slab = time;
end
end
end

for j = i - ns + 2: 1: i-ns
if (j > 0)
            slabsum = 0;
for l = j:1:i;
                slabsum = slabsum + s(1,l);
```

```
end
            p(1,j) =fa*d + (i-j+1)*(1+fa)*d  - slabsum;
if p(1,j) >maxp
                maxp = p(1,j);
                maxsho = j;
                time_shore = time;
end
end
end
for j = i - ns - nr + 2: 1: i-ns
if (j > 0)
            slabsum = 0;
for l = j:1:i;
                slabsum = slabsum + s(1,l);
end
            p(1,j) = fa*d + (ns-1)*(1+fa)*d+(i-ns+2-j)*(1+fb)*d - slabsum;
if p(1,j) > maxp
                maxp = p(1,j);
                maxsho = j;
                time_shore = time;
end
end
end
if (i >= ns)
for q = 1:1:i
if (max(q)< s(q))
                max(q) = s(q);
end
end

for q = 1: 1: i
if time-T*q>=28
            strength(q)=f28;
else
            strength(q) = ((time - T*q) / (a+(b*(time - T*q))))*f28;
end
end
        min1 = 1/0;
        minslab1 = 0;
for q= 1: 1: i
            diff = strength(q) - max(q);
if (diff<min1)
                min1 = diff;
                minslab1 = q;
end
end
        fs=fs+1;
        fos(fs)=strength(minslab1)/max(minslab1);
end
end
time = n*T + ms;
for q = 1: 1: i
        strength(q) = ((time - T*q) / (a+(b*(time - T*q))))*f28;
end
if (i == n)
```

```
    rs= 0;
    j= 0;
for m = i - nr - ns + 1 :1 :i-ns
if(m > 0)
            rs = rs+1;
if (m>1)
                s(m-1) = s(m-1) - p(m);
if s(1, m-1) > maxs
                    maxs = s(1, m-1);
                    maxslab = m-1;
                    time_slab = time;
end
end
            sum = 0;
for l = m: 1: n
                sum = sum +k(l);
end
for j = m: 1: n;
if (j>0)
                    s(j) = s(j) - (k(j)/sum)*(p(m)-fb*d);
if s(1,j) > maxs
                        maxs = s(1,j);
                        maxslab = j;
                        time_slab = time;
end
end
end
for j = m+1: 1: n-ns
if (j>0)
                    slabsum = 0;
for l = j:1:n;
                        slabsum = slabsum + s(l);
end
                    P(j) = (1+fa)*d*ns + (n-ns-j+1)*(1+fb)*d - slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        time_shore = time;
end
end
end
for j= n-ns+1: 1: n
if (j>0)
                    slabsum = 0;
for l = j:1:n;
                        slabsum = slabsum + s(l);
end
                    p(j) = (1+fa)*d*(n-j+1) - slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        time_shore = time;
end
end
end
```

```
            p(m) = 0;
            max = s;
end
end
    shore = 0;
for m = i-ns+1 : 1 : n
if(m > 0)
            shore = shore+1;
if (m > 1)
                s(m-1) = s(m-1) - p(m);
if s(1, m-1) > maxs
                    maxs = s(1, m-1);
                    maxslab = m-1;
                    time_slab = time;
end
end
            sum = 0;
for l = m: 1: n
                sum = sum +k(l);
end
for j = m: 1: n;
if (j>0)
                    s(j) = s(j) - (k(j)/sum)*(p(m)-fa*d);
if s(1,j) > maxs
                        maxs = s(1,j);
                        maxslab = j;
                        time_slab = time;
end
end
end
for j = m+1: 1: n
if (j>0)
                    slabsum = 0;
for l = j:1:n;
                        slabsum = slabsum + s(l);
end
                    P(j) = (1+fa)*d*(n-j+1)- slabsum;
if p(1,j) >maxp
                        maxp = p(1,j);
                        maxsho = j;
                        time_shore = time;
end
end
end
            p(m) = 0;
            max = s;
end
end
end

for jk=1:1:fs
if fos(jk) < coff
            PC = 100000;
else
            PC = 0;
```

```
      end
      end
      if(x(3) == 1)
              cost_conc = 40*6940*0.85*x(3);
      else
              cost_conc = (40*6940*0.85) + (75*(x(3)-1));
      end

      if x(1)+ x(2)> n
              C_violated = 30000000 - (100*T*x(1) + 0.2*100*T*x(2) + cost_conc*n +
      15000*T);
      else
             C_violated = 0;
      end

          y = 100*T*x(1) + 0.2*100*T*x(2) + cost_conc*n + 15000*T + PC + C_violated;

          temp = [x(1) x(2) f28 y PC T C_violated];
          dat = vertcat(dat,temp);
end
end
```

APPENDIX B: Code for Genetic algorithm Functions

```
clear
clc
global dat
dat = [0 0 0 0 0 0 0 ];

popsize = 13;
dimension = 3;
stringlength = 8;
x_bound = [1,19;1,19;1,7];
pm = 0.8;

pop = encoding(popsize, stringlength, dimension);
pop = decoding(pop, stringlength, dimension, x_bound);

[choice_number,choice_k]= min(pop(:,stringlength*dimension+1));
choice =(pop(choice_k));

for i = 1:1:150
    new_pop = cross_over(pop, popsize, stringlength, dimension);
    pop = mutation(new_pop, stringlength, dimension, pm);
    pop = decoding(pop, stringlength, dimension, x_bound);
    [number,k]= min(pop(:, stringlength*dimension));
if choice_number > number;
        choice_number = number;
        choice_k = k;
        choice =(pop(choice_k,:));
end
    pop=selection(pop, popsize, stringlength, dimension);
    [number,m]= min(pop(:,stringlength*dimension + 1));
    pop(m,:)= choice;

end
```

```
[value,x]= result_guo(pop, stringlength, dimension, x_bound);
xlswrite('result_full.xlsx',dat);

function new_pop = mutation(new_pop,stringlength,dimension,pm)
new_popsize=size(new_pop,1);
for i =1:new_popsize
    if rand<pm
        mpoint=round(rand(1,dimension)*(stringlength-1))+1;
        for j=1:dimension
            new_pop(i,(j-1)*stringlength+mpoint(j))=1-new_pop(i,(j-
1)*stringlength+mpoint(j));
        end
    end
end
end

function selected=selection(pop,popsize,stringlength,dimension)
popsize_new=size(pop,1);
r=rand(1,popsize);
fitness=pop(:,dimension*stringlength+1);
fitness=fitness/sum(fitness);
fitness=cumsum(fitness);
for i=1:popsize
    for j=1:popsize_new
        if r(i)<=fitness(j)
            selected(i,:)=pop(j,:);
            break;
        end
    end
end
end


function [value,x]= result_guo(pop,stringlength,dimension,x_bound)
[value,k]=min(pop(:,stringlength*dimension+1));
temp=2.^(stringlength-1:-1:0)/(2^stringlength-1);
for i=1:dimension
    bound(i)=x_bound(i,2)-x_bound(i,1);
end
for j=1:dimension
    m(:,j)=round(pop(k,stringlength*(j-1)+1:stringlength*j));
end
x=temp*m;
x=round(x.*bound+x_bound(:,1)');
end


function pop=encoding(popsize,stringlength,dimension)
pop=round(rand(popsize,dimension*stringlength+1));
end

function pop=decoding(pop,stringlength,dimension,x_bound)
popsize=size(pop,1);
temp=(2.^(stringlength-1:-1:0)/(2^stringlength-1));
for i =1:dimension
```

```
    bound(i)=x_bound(i,2)-x_bound(i,1);

end
for i=1:popsize
    for j=1:dimension
        m(:,j)=pop(i,stringlength*(j-1)+1:stringlength*j);
    end

    x=temp*m;
    x=round(x.*bound+x_bound(:,1)’);
    pop(i,dimension*stringlength+1)=funname(x);

end


function[child1,child2]=cross_running(parent1,parent2,stringlength,dimension)
cpoint=round((stringlength-1)*rand(1,dimension))+1;
for j =1:dimension
    child1((j-1)*stringlength+1:j*stringlength)=[parent1((j-1)*stringlength+1:(j-
1)*stringlength+cpoint(j)) parent2((j-1)*stringlength+cpoint(j)+1:j*stringlength)];
    child2((j-1)*stringlength+1:j*stringlength)=[parent2((j-1)*stringlength+1:(j-
1)*stringlength+cpoint(j)) parent1((j-1)*stringlength+cpoint(j)+1:j*stringlength)];
end
end

function new_pop = cross_over(pop, popsize, stringlength, dimension)
match = round(rand(1,popsize)*(popsize-1))+1;
for i=1:popsize

[child1,child2]=cross_running(pop(i,:),pop(match(i),:),stringlength,dimension);
    new_pop(2*i-1:2*i,:)=[child1;child2];
end
end
```