Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

Optimizing Multiple-Resource Leveling in Multiple Projects Using Shuffled Frog Leaping Algorithm

A. Hashim[1]*, M. Ehab[1]*, O. Hosny[2], A. Elhakeem[3], Y. A.S. Essawy[4]

[1] Graduate Student, Dept. of Construction Engineering, The American University in Cairo (AUC), Cairo, Egypt
[2] Professor, Dept. of Construction Engineering, The American University in Cairo (AUC), Cairo, Egypt
[3] Professor, Construction and Building Engineering Department, College of Engineering and Technology, Arab Academy for Science Technology & Maritime Transport (AASTMT), Cairo, Egypt
[4] Assistant Professor, Dept. of Structural Engineering, Ain Shams University (ASU), Cairo, Egypt; and Dept. of Construction Engineering, The American University in Cairo (AUC), Cairo, Egypt

ABSTRACT: Efficient management of resources in construction is one of the key goals that project managers thrive to achieve on a regular basis along the lifecycle of a project. On a larger scale, program managers face a more challenging task when trying to allocate different resources of a construction company among the different projects that the company is contracted to, given their variable and urgent requirements to be able to complete the projects on schedule. One of the most significant aspects that contribute to the success of resource management is resource leveling, also commonly referred to as resource smoothing. This paper proposes a user-friendly Excel® model that reduces fluctuation of several resource histograms shared among multiple projects, by optimizing their moments of area in a multi-objective manner. The model variables are the actual starting dates of non-critical activities that are changed within their total floats, while being subject to logical relations and project duration constraints. In search for a near-optimum solution, the model changes its variables following a shuffled frog leaping algorithm that has been programmed using Excel's VBA tool. The model has been applied on a case study, where the resulting resource histograms had a more efficient and less fluctuating profile than the original resource profile. More importantly, it had a more efficient profile than solutions generated by conventional leveling methods.

1. BACKGROUND

The construction industry is facing several challenges that are gradually increasing over the years. The key challenge that every construction manager is encountered with is to achieve the required objective with the least possible use of financial and human resources, and in the least possible duration. Several techniques can be used for addressing such challenges such as optimization and simulation. Optimization models can deliver optimal or near-optimal solutions by investigating variables and their feasible ranges under given their constraints. Different single objective optimization techniques such as linear programming can approach optimal solutions. However, in large problems dealing with several variables, correlated with sophisticated functions, such tools are highly likely not to prove feasible in search for the optimum solution. Evolutionary algorithms (EAs) present promising solutions to these challenges. They are algorithms that have been inspired by observing populations of living creatures and the methods their fittest members survive. Applying such algorithms in construction optimization problems has proven to yield very promising results. In these algorithms, random decision

variables are generated as input for a simulation model, and the output data from the simulation can inform an optimization model. This iterative process refines decision variables based on previous calculations until the best solution is reached within a maximum number of iterations. Genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO) are examples of EAs developed for optimization problems (Fallah-Mehdipour et al. 2012).

There are several typical construction problems which EA models developed through research projects have managed to propose optimal solutions for. For instance, several researches have addressed the site layout problem from different perspectives by proposing ACO models (Ning et al. 2019), (Ning et al. 2018). One of the most common applications for EA models in construction is multi-objective trade off between different parameters such as time, cost, quality or resources. For instance, in a study, GA was utilized to optimize the lately mentioned factors (Elkliny et al. 2023). Another study optimizes both time and cost using social group optimization algorithm (Tran 2020). With respect to resources, several studies have investigated resource-related problems, proposing solutions featuring EA models. For instance, a study has proposed a PSO model that addresses resource allocation problem (Kalpana et al. 2021). Another paper has proposed a GA model to optimize project scheduling that is driven by resources by efficiently allocating and leveling them (Kaiafa and Chassiakos 2015). In another research, an ACO model has been proposed to solve the resource leveling problem (Duraiswamy and Selvam 2022). Accordingly, resource optimization has been extensively featured in the literature showing its significance.

Lack of resources is a typical issue that is encountered by project managers in the construction industry. When planning a project, it is important to consider when resources will be needed. Therefore, resource profiles can be developed through project network methods such as the Critical Path Method (CPM) to visualize the overall resource need along the entire project duration. However, when such resource histograms are left unlevelled, it will have drastic consequences on the project cost. For instance, it is very impractical for contractors to hire and fire the workers required to cope with the changing resource profiles. Thus, during periods of low demand, contractors are bound to bear a portion of idle resources, which lowers project profitability. To avoid unnecessary resource expenditures and adhere to agreed-upon timelines, resources must be managed efficiently (Cheng et al. 2015).

In large construction companies, managing multiple projects is the common practice. Accordingly, multiple resource leveling in multiple projects (MRLML) is a significant objective that can lead to reducing gigantic costs. Several studies were found in the literature investigating the resource leveling (RL) problems from different approaches. For instance, mathematical techniques were tried, but in large complex project networks, the combinatorial nature of resource levelling was found to be impractical due to the growing number of decision variables (Savin et al. 1996). Other studies have proposed evolutionary algorithms while solving RL problems. For instance, a study has approached the problem employing minimum moment algorithm, optimizing M_x and M_y. However, the scope of the study was limited to a singular project by implementing GA (Hegazy 1999). Other studies have addressed the MRLML problem. A study has proposed a PSO model to reduce fluctuation and its output was found to be more efficient than the GA algorithm result (Guo et al. 2009). Another study, addressing the same problem, has proposed another algorithm called Discrete Symbiotic Organisms Search (DSOS). It was applied on the same case study mentioned in the latest cited research, and its output was proven statistically to outperform the results obtained from the GA, PSO, and differential evolution algorithms (Cheng et al. 2015).

Results of the literature review have highlighted that different algorithms yield different results. Accordingly, trying other algorithms such as the shuffled frog leaping (SFL) algorithm on the MRLML problem, which has not been covered in the literature, may lead to results that are comparable to outputs of other algorithms in that matter. Therefore, the objective of this research project is to produce a user-friendly model that solves the MRLML using the SFL algorithm, while respecting the project deadlines, and where the model is entirely programmed and operated on Microsoft Excel, one of the easiest tools that construction managers resort to for simple and fast calculations, which would facilitate the process of managing multiple projects for program directors.

2. SHUFFLED FROG LEAPING ALGORITHM

One of the Evolutionary Algorithms that has been inspired by nature is the SFL algorithm. It mimics the behavior of frogs when looking for food. This algorithm is considered a combination of the PSO and the shuffled complex evolution algorithm in order to incorporate their advantages in a more effective manner (Ashuri, et al. 2015). In analogy to the GA, in this system a valid solution is referred to as a frog instead of a chromosome, and a variable is referred to as a meme rather than a gene. An initial population of frogs is generated, and their fitness (F) is evaluated in accordance with the objective function of the problem. Then, based on the calculated fitness values, the frogs are sorted in a descending order showing the best ones at top. After that, frogs are divided into memeplexes, which are local sub-divisions of the global population. Such a division is done in a manner that groups good frogs and bad frogs into local memeplexes, and avoids placing all good ones into certain memeplexes, and all bad ones into others. This division step is considered the key advancement of this algorithm over the PSO, which achieves a faster rate of convergence towards a more optimal solution (Ashuri, et al. 2015). After they are divided, the worst frogs in each memeplex get their memes values enhanced through the multiplication of the memes difference between the best and the worst in the memeplex by a random probability. Then, the property of the new frog is measured to evaluate its fitness and is compared to the original worst frog inside each memeplex. If such fitness is better, the new frog would replace the worst of the memeplex. If not, the new frog memes are re-generated by multiplying the difference between the worst of the memeplex and the global best by a random probability. If such a step failed to enhance the new frog over the worst of the memeplex, the worst frog is replaced by a randomly generated frog. The process is repeated several times in the memeplexes, and then all frogs are re-gathered in the original global population, re-sorted, re-divided among the memeplexes, and the process is repeated until a stoppage criterion is achieved. Figure 1 below shows the steps of the SFL algorithm.

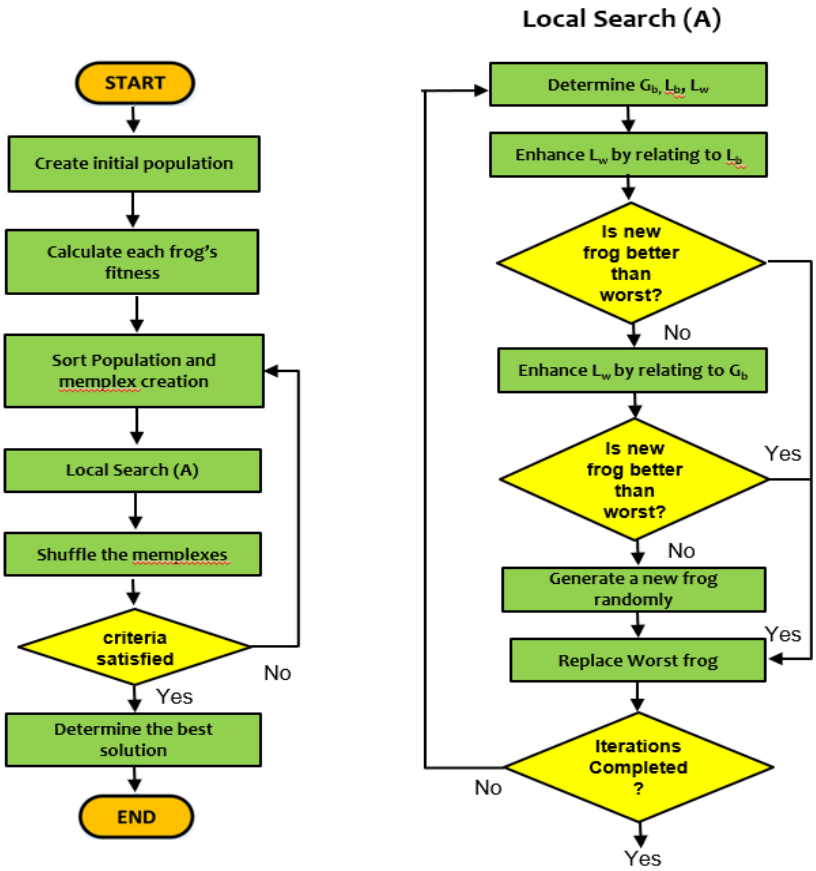

Figure 1: Shuffled Frog Leaping algorithm

3. MODEL DEVELOPMENT AND FEATURES

The developed model is in the form of an Excel file that has six sheets that are linked together through VBA codes. The first sheet features a graphical user interface (GUI) where the user inputs the number of projects, number of resources to be leveled and number of activities in each project. Upon inputting these data, the sheet has a VBA button that the user presses, which would create, in another sheet, a table for each project for the user to input activity IDs, durations, relations and resources. Since it is an experimental not a commercial version, the model has a few architectural boundaries that were set for ease of trial and error process. It allows up to 3 resources to be leveled. The maximum number of predecessors or successors that an activity can have is 5. The maximum project duration is set at 25 time units. Activity interruption is not permitted. The third sheet titled, "CPM" mainly has 2 parts. In the first part, automated CPM calculations are done through built-in functions, identifying critical activities and specifying the projects' durations (PD), activities' early starts (ES), late starts (LS), early finishes (EF), late finishes (LF), total floats (TF) and free floats (FF). Accordingly, the lower and upper limits of the actual start times (AS LL) and (AS UL) for non-critical activities are identified as well. Equations 1 through 9 below summarize the CPM calculations.

[1] $EF_i = ES_i + Duration_i$

[2] $ES_i = \begin{cases} 0, & \text{in case of no predecessors} \\ \text{Maximum of } EF_{\text{preceding activities}}, & \text{in case of having predecessors} \end{cases}$

[3] $PD = \text{Maximum of } EF_{\text{all activities}}$

[4] $LS_i = LF_i - Duration_i$

[5] $LF_i = \begin{cases} PD, & \text{in case of no successors} \\ \text{Minimum of } LS_{\text{succeeding activities}}, & \text{in case of having successors} \end{cases}$

[6] $TF_i = LF_i - EF_i$

[7] $FF_i = \text{Minimum of } ES_{\text{succeeding activities}} - EF_i$

[8] $AS\ LL_i = ES_i$

[9] $AS\ UL_i = AS\ LL_i + FF_i$

This part of the sheet also features automated bar charts, and automated resource histograms for each of the resources, based on their ES. The objective function that the model optimizes is the sum of moments about the x and y axes (M_x) and (M_y) due to the resource demands (D). The model takes into account the user's expertise and preference for importance and scaling of each of the weights of the two parameters (w_x) and (w_y), and accordingly allows the user to input such weights that are multiplied respectively by the calculated M_x and M_y and the output is summed for estimating a fitness value for a variable in the model. In calculating both M_x and M_y, the model also considers possible differences in weights for importance and scaling of each of the resources to be levelled (w_r), and allows the user to input them. For calculation of the resource histograms (RH) on any day, a binary variable (v) having values of either 1, when the activity is performed on the analyzed day, or 0 otherwise, is multiplied by the resource demand on that day, and the resultant are summed for that day. Equations 10 through 13 below demonstrate how each of the above-mentioned parameters are calculated, where equation 13 is the objective function to be minimized by the model, i represents the different activities, j represents the different projects, k represents the time units of the projects, r represents the different resources and t represents the different time points.

[10] $RH_{t,r} = \sum_{i=1}^{I} v * D$

[11] $M_{x\,i,j} = \sum_{k=1}^{K} D^2 * 0.5$

[12] $M_{y\,i,j} = \sum_{k=1}^{K} k * D$

[13] $\text{Min } F = \sum_{j=1}^{J} \sum_{r=1}^{R} w_r * (w_x * M_x + w_y * M_y)$

The second part of the CPM sheet is almost identical to the first part, but with the change of considering the activities actual starts instead of their early starts. Accordingly, this part in linked to the VBA code and to the fourth sheet of the paper titled, “SFL”. In this part, any potential solution to the problem is being evaluated. The SFL sheet proposes solutions to the problem in terms of frogs. Each frog is composed of several memes, each representing the actual start of a non-critical activity. It is very difficult to directly link the proposed AS to the resultant resource histograms M_x and M_y, through mathematical equations in a single function. Therefore, the model takes the proposed AS from the SFL sheet to the second part of the CPM sheet, where bar charts of the projects would be updated, resource histograms would be created, and M_x, M_y and the solution fitness would be calculated. Another important check that the second part of the CPM sheet does is that it assures the feasibility of the proposed solution, by first shifting free floating activities within their FF, and then shifting the preceding noncritical activities within their updated FF depending on the remainder of the consumed TF.

The SFL sheet is the core of the model algorithm. Basically, it is divided into three zones. The first zone has the GUI where the user can initiate the algorithm. Before that, the user is required to input the population size, and number of frogs in each memeplex. Due to the relatively large number of steps that the SFL algorithm has, they were decided to be broken down into six separate codes, where each can be triggered through a separate button, as illustrated in Figure 2 below. Based on the user’s inputs, pressing the first button would generate a random initial population, whose fitness can be evaluated by pressing the second button. After that, frogs are sorted and assigned to memeplexes as per the SFL algorithm by pressing the third button. Pressing the fourth button would result into enhancing the frogs locally in their memeplexes. Local enhancement of frog memes is done as per Equations 14 through 17 shown below, where $\Delta X_{L\,Best}$ and $X_{new,\,L\,Best}$ represent the change in position, and the new position of the frog memes by attributing them to the frog with the current best F in the memeplex. $\Delta X_{G\,Best}$ and $X_{new,\,G\,Best}$ represent the change in position, and the new position of the frog memes by attributing them to the frog with the current best F in the population. $X_{L\,Worst}$ represents the frog with the worst F in the memeplex. Local enhancement of the frogs in the memeplexes is set by the code to take place over 10 iterations. Pressing the fifth button would lead to reassigning the enhanced frogs into other memeplexes, so that good memes can be distributed to other memeplexes as well. The last button titled, “Final Solution” repeats the fourth and fifths steps for the number of iterations specified in the code. The second zone of the SFL sheet is considered the mother table where the initial population is being generated and their fitness values are recorded. The third zone of the SFL sheet is where local memeplexes are created and the enhancement steps take place. In order to be able to trace the convergence rate of the model towards an optimum solution, a fifth sheet has been created, called “Runs”, where results of each iteration are being recorded. The fifth sheet records are summarized in a pivot table and chart available in the sixth sheet, titled “Results”, where only the optimum result of each iteration is plotted.

[14] $\Delta X_{L\,Best} = (X_{L\,Best} - X_{L\,Worst}) * \text{rand}()$

[15] $X_{new,\,L\,Best} = \Delta X_{L\,Best} + X_{L\,Worst}$

[16] $\Delta X_{G\,Best} = (X_{G\,Best} - X_{L\,Worst}) * \text{rand}()$

[17] $X_{new,\,G\,Best} = \Delta X_{G\,Best} + X_{L\,Worst}$

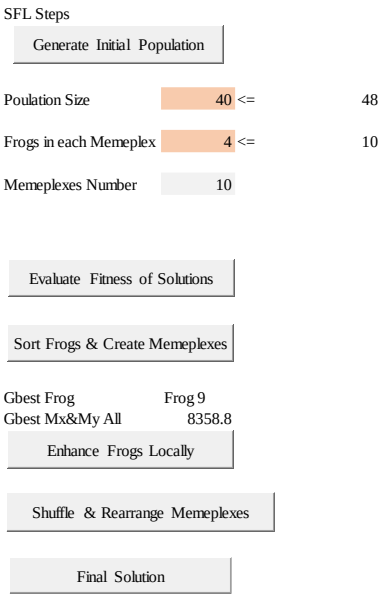

Figure 2: Model SFL algorithm buttons

4. CASE STUDY

One of the most efficient ways for testing a newly developed model is trying it on an example from the literature. The example that has been selected for application was found in two previous researches (Cheng et al. 2015), and (Guo et al. 2009). The problem features two projects with three resources to be leveled, having a total of 12 non-critical activities in the program. Figure 3 below shows a bar chart network representation of the problem highlighting activity durations, relations, FF and resources. Figure 4 shows the initial resource histograms before leveling based on the ES times.

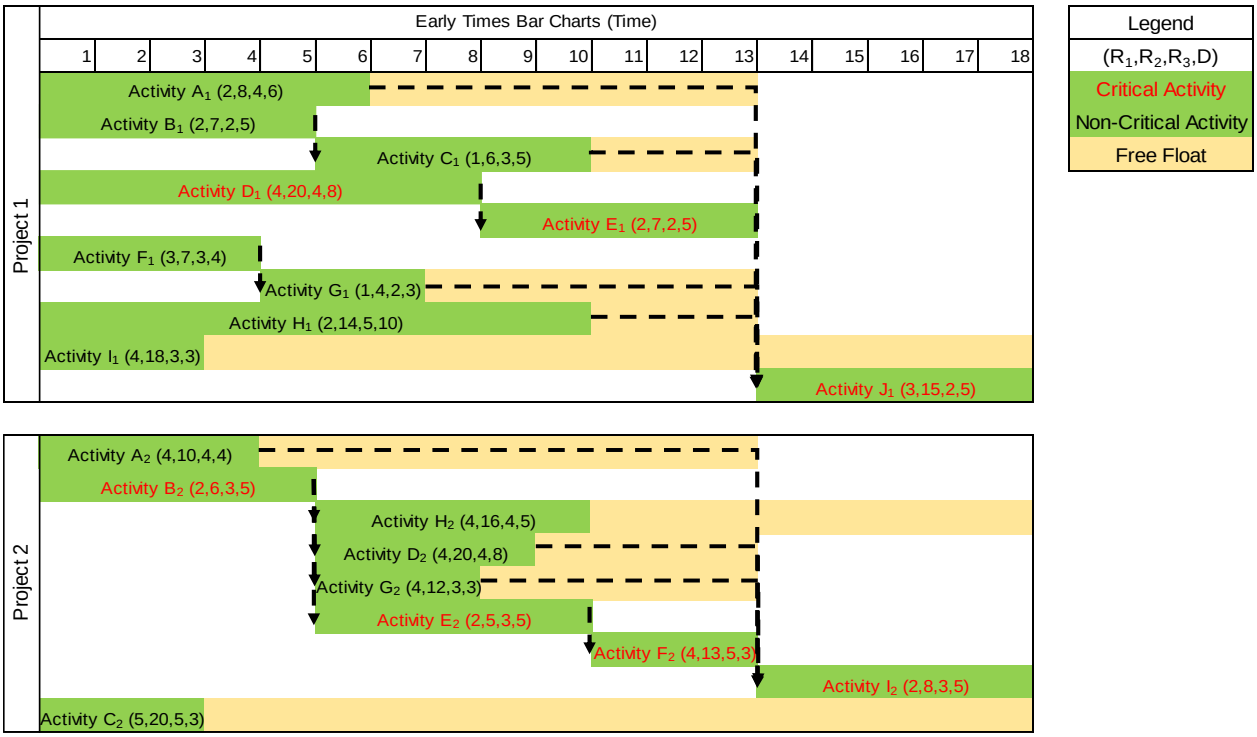

Figure 3: Case study networks

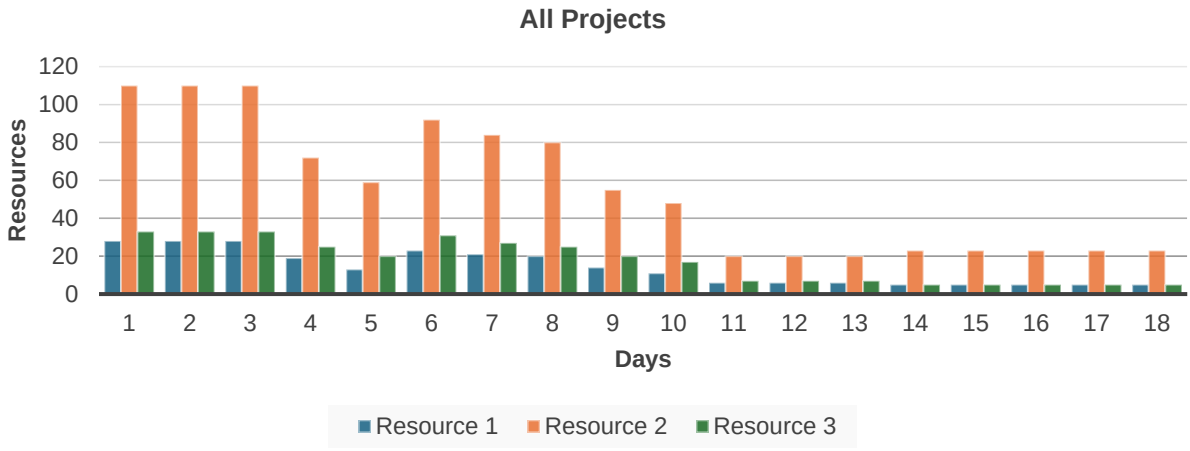

Figure 4: Resource histograms before leveling

5. RESULTS AND DISCUSSION

In order to be able to compare the results of the model with the results obtained from other algorithms presented in previous researches, some factors were set before running the model to have the same values as in the literature. For instance, the total number of iterations in the first research was set at 300 (Guo et al. 2009), and in the second research was set to 200 (Cheng et al. 2015). Accordingly, the model was set to run for 200 iterations. The importance or scaling weights assigned to each of the resources were set to be the same as suggested by the two researches, having values of 0.637, 0.258 and 0.105 for resources 1, 2 and 3 respectively. The fitness function of the two researches minimizes a parameter referred to as the resource intensity (RI) that is calculated as the sum of squares of differences between the resource usage on day and the average resource usage along the project duration. Even though, the fitness function of the model differs from that of the two researches as the model optimizes M_x, and M_y, M_x is the parameter that can be analogous to RI. Therefore, in the model, M_x was assigned a weight of 100%, while M_y was assigned a 0% weight. The model has been run for 50 times in the second research for statistical analysis purposes (Cheng et al. 2015). However, due to time limitations, the model has been run for only 5 times. The size of the population was set in the second research to have 100 members (Cheng et al. 2015). The model was designed having a maximum size of initial population of 48 frogs as indicated in Figure 2. Accordingly, the model was decided to run with a population of 40 frogs.

In the first three runs, the 40 frogs were divided over 5 memeplexes, each having 8 frogs. In the fourth and fifth runs, the population were divided over 10 memeplexes, each having 4 frogs. Such a change of the memeplex size might have had a positive effect on the results of the model, as the optimum result of the five runs was in achieved in run 4. Table 1 below shows the resultant AS of non-critical activities of the five runs, and the resultant combined fitness values, and fitness for each of the resources. Similarly, table 2 shows the model results in comparison to outputs of other algorithms. The fitness values resulting from AS of non-critical activities obtained by other algorithms have been calculated using the same coefficients, and consistently compared to the output of the SFL algorithm.

Table 1: Model runs results

Run	F	F_1	F_2	F_3	Actual Start											
					A_1	B_1	C_1	F_1	G_1	H_1	I_1	A_2	C_2	D_2	G_2	H_2
1	8,415	1,088	7,040	287	0	1	8	0	8	0	13	6	15	6	10	10
2	8,460	1,094	7,077	290	3	0	5	0	8	3	0	6	13	9	10	13
3	8,370	1,094	6,992	283	3	0	5	0	9	0	15	8	12	8	10	13
4	8,359	1,075	7,002	281	3	0	8	0	9	0	15	5	12	9	9	13
5	8,380	1,077	7,022	282	0	0	8	0	9	2	15	9	12	5	6	13

Table 2: Comparison between different algorithms results

Algorithm	F	F_1	F_2	F_3	Actual Start											
					A_1	B_1	C_1	F_1	G_1	H_1	I_1	A_2	C_2	D_2	G_2	H_2
Initial	11,565	1,507	9,671	388	0	0	5	0	4	0	0	0	0	5	5	5
GA	8,365	1,077	7,007	281	0	3	8	0	9	0	15	8	12	9	6	13
PSO	8,385	1,074	7,031	279	0	0	8	6	10	3	12	0	15	8	5	13
DE	8,379	1,075	7,027	278	0	3	9	0	10	0	12	8	15	9	6	13
DSOS	8,380	1,074	7,025	281	3	0	8	0	8	0	12	8	15	9	5	13
SFL	8,359	1,075	7,002	281	3	0	8	0	9	0	15	5	12	9	9	13

As shown in table 2, results obtained from the SFL algorithm feature a slight improvement in the fitness value over other competing algorithms, indicating the effectiveness of the proposed model in solving the MRLMP problem. More importantly, the output of the proposed SFL model has led to a huge improvement in the resource histogram profile in comparison to its initial state based on the ES of the non-critical activities, managing to reduce the objective function incorporating M_x by at least 28%. Such optimization can be further visualized by investigating the resource histograms resultant from the model for the proposed solution, shown in Figure 5 below.

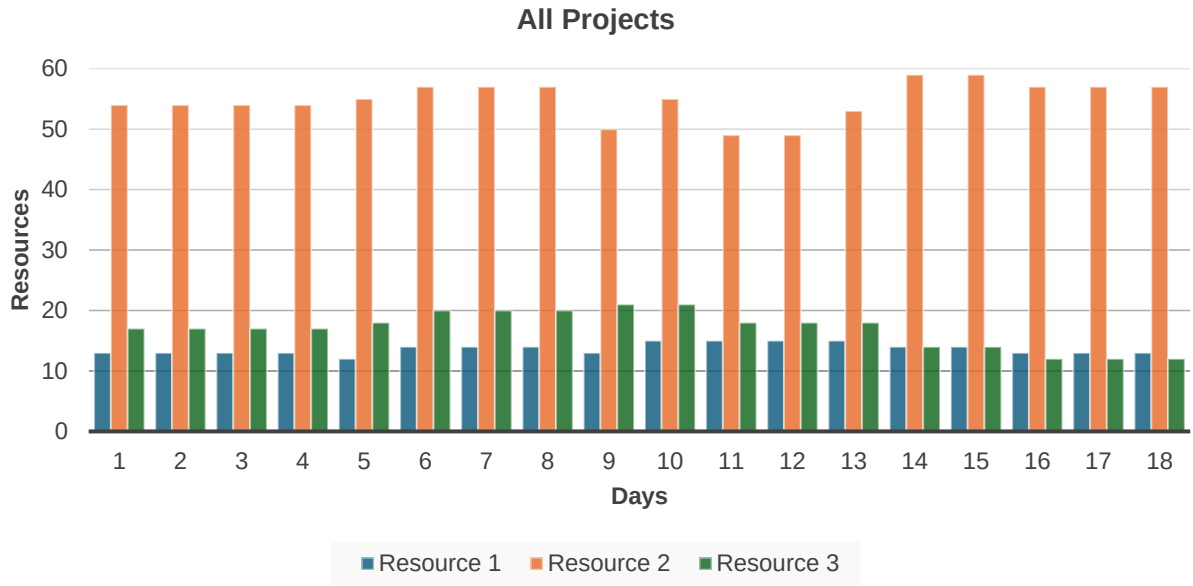

Figure 5: Resource histograms after leveling

Figure 6 shows the bar chart of the proposed updated schedule of the program. Figure 7 shows the convergence trend of the SFL model in approaching the optimum solution in its fourth run. Even though the model has produced promising results, one of its main drawbacks is its relatively long operating time to reach a solution. Due to the large number of steps in the SFL algorithm, on average, the model took between two and three hours to yield a solution for the case study after 200 iterations. Of course, several factors contribute to the time the model takes to solve an MRLMP problem such as the specifications of the used computer, the complexity of the projects and the number of non-critical activities in the program.

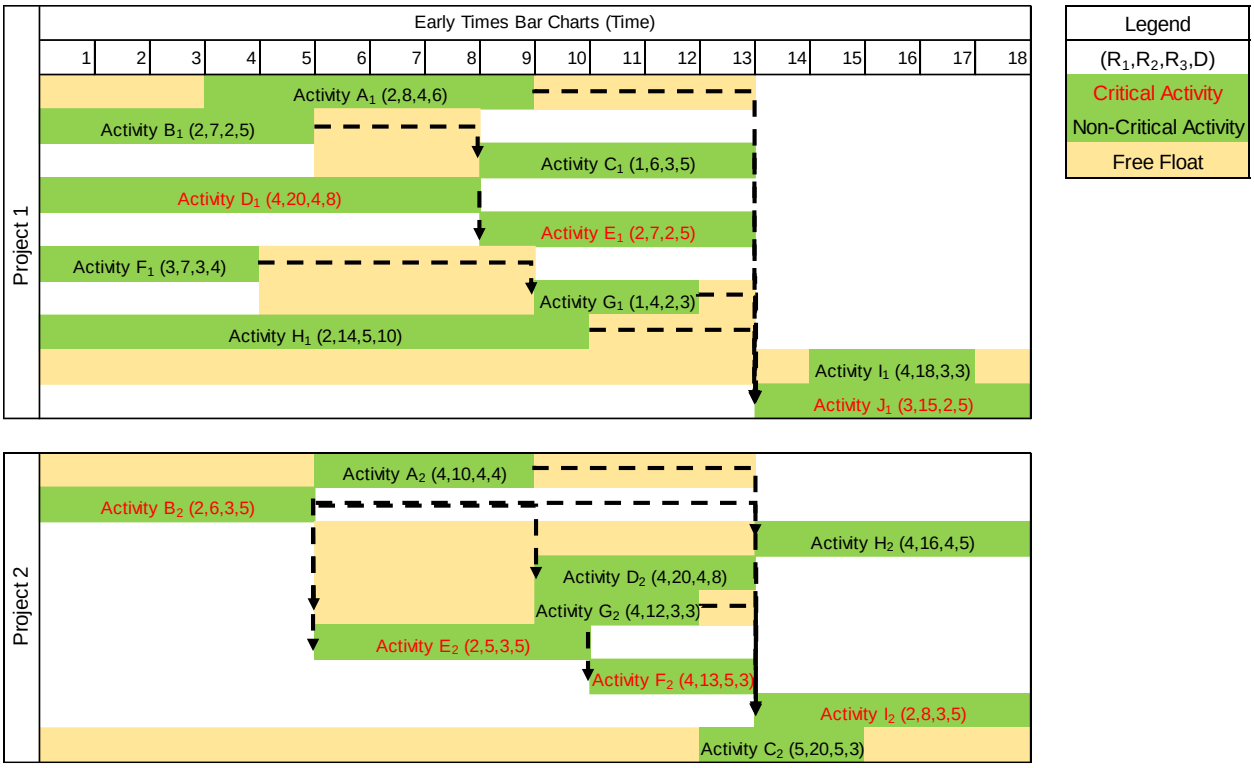

Figure 6: Projects proposed bar charts after leveling

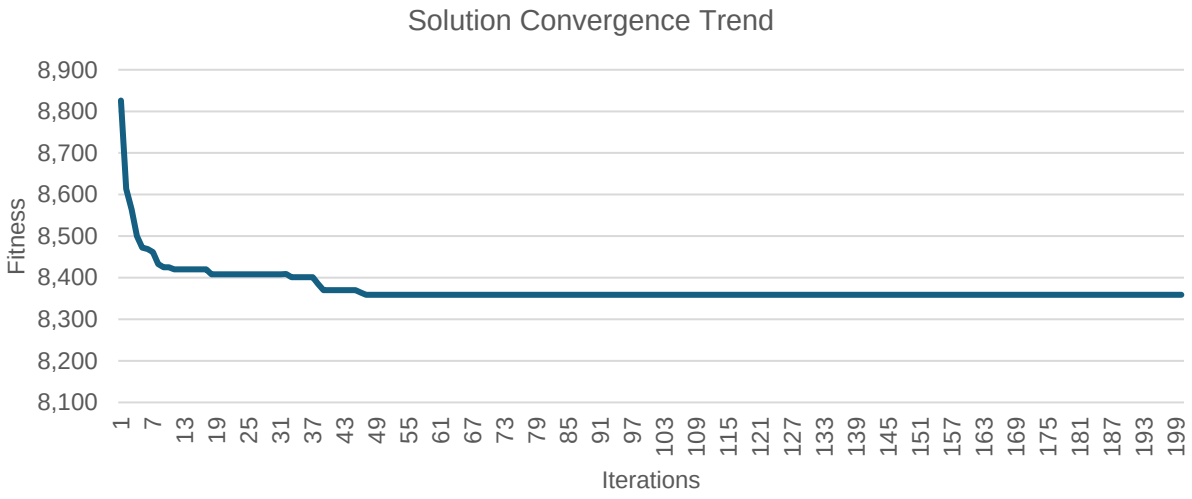

Figure 7: SFL model convergence trend

6. CONCLUSIONS AND RECOMMENDATIONS

This paper proposes a model that implements shuffled frog leaping algorithm (SFL) in solving the multiple resources leveling in multiple projects problem. The proposed model has resulted into optimal scheduling solutions that leads to reducing the fluctuations in multiple resources histograms, by reducing the moments of area of the histograms about its axes. The model has been applied on a case study that was found in previous researches in the literature, and its results were compared to the results of other EAs showing slight advantage of the SFL over other EAs. When compared to the original resource histograms based on

the early start dates, the model was found to have hugely improved the resource histograms, and has reduced M_x by approximately 28%.

The most important feature of the model is that it can be operated entirely on Excel, a software that can be found on almost every computer. Also, it is completely automated and can be applied on any case study within its architectural limitations, without prior adjustment. All that is required from the user is to input the activity durations, relations and resources. The user is also given the advantage of optimizing both M_x and M_y, while being able to assign each of the weights that he/she finds suitable for the program under optimization. Another area where the model is adjustable to the user's preferences is that the user can also input weights representing the importance of each of the resources.

There are several points to consider for future improvement of the model. First, the model should be tested 50 times on the investigated case study, instead of just 5 times, to be able to draw statistical conclusions on the performance of the model, rather than comparing between the optimum outputs of the different EAs. It was only due to time limitation that such a step was not completed. In addition, the model should be tested on several other case studies, where its convergence rates can be investigated to decide statistically whether the population size, the number of memeplexes affect the convergence rate or not. More importantly, the model VBA code can be enhanced to reduce its running time.

REFERENCES

Ashuri, B. and Tavakolan, M. 2015. Shuffled Frog-Leaping model for solving Time-Cost-Resource optimization problems in construction project planning. *Journal of Computing in Civil Engineering,* 29: 04014026.

Cheng, M.Y., Prayogo, D. and Tran, D.H. 2016. Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search. *Journal of Computing in Civil Engineering,* 30: 04015036.

Duraiswamy, A. and Selvam, G. 2022. An Ant Colony-Based Optimization Model for Resource-Leveling Problem. *Advances in Construction Management,* Springer, Singapore, 191: 333-342.

Elkliny, A.F., Sanad, H.M. and Etman, E.E. 2023. Time-cost-quality tradeoff considering resource-scheduling problems. *Ain Shams Engineering Journal,* 14: 102524.

Fallah-Mehdipour, E., Haddad, O.B., Tabari, M.R. and Mariño, M. 2012. Extraction of decision alternatives in construction management projects: Application and adaptation of NSGA-II and MOPSO. *Expert Systems with Applications*, 39: 2794-2803.

Guo, Y., Li, N. and Ye, T. 2009. Multiple Resources Leveling in Multiple Projects Scheduling Problem Using Particle Swarm Optimization. *2009 Fifth International Conference on Natural Computation,* IEEE, Tianjian, China, 6: 260-264.

Hegazy, T. 1999. OPTIMIZATION OF RESOURCE ALLOCATION AND LEVELING USING GENETIC ALGORITHMS. *Journal of Construction Engineering and Management,* ASCE, 125: 167-175.

Kalpana, N., Gai, H.K., Kumar, A.R. and Sathya, V. 2021. Optimal Resource Allocation Based on Particle Swarm Optimization. *Advances in Communications, Signal Processing, and VLSI,* Springer, Singapore, 722: 199-211.

Kaiafa, S. and Chassiakos, A.P. 2015. A Genetic Algorithm for Optimal Resource-driven Project Scheduling. *Procedia Engineering,* 123: 260-267.

Ning, X., Qi, J., Wu, C. and Wang, W. 2019. Reducing noise pollution by planning construction site layout via a multi-objective optimization model. *Journal of Cleaner Production,* 222: 218-230.

Ning, X., Qi, J., Wu, C. and Wang, W. 2018. A tri-objective ant colony optimization based model for planning safe construction site layout. *Automation in Construction,* 89: 1-12.

Savin, D., Alkass, S. and Fazio, P. 1996. Construction resource leveling using neural networks. *Canadian Journal of Civil Engineering,* 23: 917-925.

Tran, D.H. 2023. Optimizing time–cost in generalized construction projects using multiple-objective social group optimization and multi-criteria decision-making methods. *Engineering, Construction and Architectural Management,* 27: 2287-2313.