Cuckoo search algorithm for construction site layout planning

Meilinda Fitriani Nur Maghfiroh[1], Anak Agung Ngurah Perwira Redi[2], Janice Ong[3], Muhamad Rausyan Fikri[4,5]

[1]Department of Industrial Engineering, Faculty of Industrial Technology, Universitas Islam Indonesia, Yogyakarta, Indonesia
[2]Department of Industrial Engineering, Faculty of Engineering and Technology, Sampoerna University, Jakarta, Indonesia
[3]Department of Mechanical Engineering, Faculty of Engineering and Technology, Sampoerna University, Jakarta, Indonesia
[4]Department of Information System, Faculty of Engineering and Technology, Sampoerna University, Jakarta, Indonesia
[5]Department of Automation Science and Engineering, Tampere University, Tampere, Finland

Article Info

Article history:

Received Jun 17, 2022
Revised Sep 26, 2022
Accepted Oct 26, 2022

Keywords:

Construction site layout planning
Cuckoo search algorithm
Metaheuristics
Optimization

ABSTRACT

A novel metaheuristic optimization algorithm based on cuckoo search algorithm (CSA) is presented to solve the construction site layout planning problem (CSLP). CSLP is a complex optimization problem with various applications, such as plant layout, construction site layout, and computer chip layout. Many researchers have investigated the CSLP by applying many algorithms in an exact or heuristic approach. Although both methods yield a promising result, technically, nature-inspired algorithms demonstrate high achievement in successful percentage. In the last two decades, researchers have been developing a new nature-inspired algorithm for solving different types of optimization problems. The CSA has gained popularity in resolving large and complex issues with promising results compared with other nature-inspired algorithms. However, for solving CSLP, the algorithm based on CSA is still minor. Thus, this study proposed CSA with additional modification in the algorithm mechanism, where the algorithm shows a promising result and can solve CSLP cases.

Corresponding Author:

Meilinda Fitriani Nur Maghfiroh
Department of Industrial Engineering, Universitas Islam Indonesia
Jalan Kaliurang KM 14.5 Sleman Yogyakarta, Indonesia 55584
Email: meilinda.maghfiroh@uii.ac.id

1. INTRODUCTION

Construction management involves many decision-making stages that need to be tackled, one of them being on-site management. All vital decisions for on-site construction management concern the site layout configuration. Construction site layout planning (CSLP) deals with identifying and determining the function, location, and size; of the temporary facilities; on the construction site [1]. Although CSLP is not directly related to the project's technicality, the poor site layout will impact the overall budget due to an increment in material handling time, more queues, and harmful workers' productivity [2], [3]. The absence of detailed and specific site layout planning might result in problems such as the wrong location of materials stack, inadequate space, and facilities wrongly located concerning their practical use. The above issues will impact the on-site movement of equipment, materials, and workers, with overall project delay. From the managerial point of view, a detailed management site layout planning could improve a project. A thorough and effective site layout planning can lead to better material transfer and waiting times and help create a safer environment. Planning and designing construction site layouts become critical in a project. CSLP can be categorized as a non-polynomial hard (NP-hard) problem. Due to its complex nature, many researchers have solved the problem using two approaches, exact and heuristics.

Several researchers have utilized the exact approach for solving facility layout, including CSLP, branch and bound, simplex method, enumeration algorithm, cutting plane algorithm, and dynamic programming. The branch and bound algorithm (BBA) is employed to solve CSLP by Wong *et al.* [4] and Huang *et al.* [5]. Further, Huang and Wong [6]–[8] continued to employ BBA to solve many types of CSLP. The other type of exact algorithm, which is the enumeration algorithm (EA), is proposed by Park *et al*. [9], and the cutting plane is used by Hammad *et al*. [10]. Liu and Lu [11], continued by El-Rayes and Said [12], used the simplex algorithm to solve the related problem. El-Rayes and Said [12] proposed a dynamic programming model by considering site space reuse, facility relocation, and project duration. Xu and Song [13] utilized dynamic programming to solve the CSLP problem by optimizing the distance between facilities and the layout's cost. Although the exact approach can produce an optimal solution, unfortunately, this approach challenged the increment of computing complexity for large-scale problems. In that regard, a much longer computation time is needed to reach an optimal solution for a large-scale problem, and the issues should be simplified. Another alternative could be by hybridization of exact and heuristic approaches. The hybrid algorithm might lead to an optimal solution but reasonable computing time.

Metaheuristics approaches, on the other hand, comes with numerous advantages, such as having better and reasonable computational time, the ability to facilitate more constraints and objectives [14], [15], and being more flexible as metaheuristics algorithm can be used for many types of problems without having to derive objective functions [16]. The drawbacks, which are related to the optimal solution, are considered tolerable in the trade-off for the computational time, especially for large-scale problems. According to Boussaid *et al*. [17], much optimization-based research agreed upon metaheuristics approaches as efficient. Thus, many researchers are more drawn to metaheuristics approaches than exact approaches. According to Xu *et al*. [18], of all related CSLP articles, genetic algorithm (GA) (29.3%), branch and bound (12%), particle swarm optimization (PSO) (10.7%), and ant colony optimization (6.7%); are employed the most for solving CSLP.

The GA is applied for solving site layout problems by Kumar and Cheng [14], Li and Love [19], Hegazy and Elbeltagi [20], Mawdesley and Al-Jibouri [21], Zhou *et al.* [22], Khalafallah and Hyari [23]. Many also utilized PSO to solve site layout problems, including Zhang and Wang [24], Xu and Li [25], Song *et al.* [26], Gharaie *et al.* [27] used ant colony optimization (ACO) and added partial path replacement to avoid infeasible solutions to solve site layout problems. Following that Ning *et al.* [28], Ning and Lam [29], Ning *et al.* [30] also used ACO to solve the site layout problem. Many also attempted to solve variants of CSLP using different metaheuristics algorithms. Some of the other metaheuristics approaches include harmonic search [31], bee algorithm (BA) [32], colliding bodies optimization [33], and symbiotics organism search [3], [34]. A hybrid algorithm using a bacterial and bacterial evolutionary algorithm with several heuristic methods such as clustering algorithm, memetic algorithm, and Hungarian algorithm is done by Kalmár *et al*. [35]. Lam *et al*. [36] proposed a conjoint min and max ant system with a genetic algorithm to solve CSLP, while Kaveh and Moghaddam [37] proposed whale optimization algorithm with colliding bodies optimization (WOA-CBO).

Nature-inspired algorithms demonstrate a high achievement percentage among some successful algorithms to solve CSLP. The trend to use nature-inspired algorithm continues until today as it is proven to be a global optimizer with simple yet powerful tools to reach the nearly optimal solution [38]. As a family of stochastic algorithms, nature-inspired algorithms can solve more applications in real problems extensively [39] and give admirable performance by finding solutions better in a broader range of issues, having a high convergence rate, and producing unbiased exploration and exploitation during the search. In some nature-inspired algorithms, multiple solutions also increase the chance to explore more search space. Yang and Deb [40] introduced a nature-inspired algorithm based on the peculiar behavior of cuckoo birds to exploit a suitable host to raise their offspring, namely the cuckoo search algorithm (CSA). Since being introduced, this algorithm has gained popularity in resolving large and complex problems [41], [42]. The CSA uses a random walk called Levy flight to explore the search space, which is believed to have an advantage as a global searcher. Compared with other nature-inspired algorithms, CSA has fewer parameters to adjust [43].

This study attempted to develop a novel CSA to solve the CSLP problem. To date, we have not found any CSA used for solving CSLP. Nevertheless, CSA has solved many optimization problems, including facility layout. Ouaarab *et al*. [44] proposed a discrete CSA to solve traveling salesman problem (TSP) by adding a mechanism to have a portion *Pc* to improve from the current solution and portion *Pa* to improve from the best solution. The result shows a promising performance by CSA and can outperform the algorithms used for comparison. Ouyang *et al*. [45] also experimented with solving TSP using discrete CSA. Kang *et al*. [46] employed CSA using gamma distribution as a random walk-in levy flight procedure to solve a closed-loop layout problem (CLLP). The algorithm effectively and robustly solved the CLLP for small and large instances up to 30 cells. The CSA was also used by Teymourian *et al*. [47] to solve capacitated vehicle routing problem (CVRP) by improving the discrete CSA by Ouaarab *et al*. [44]. In the case of the hybrid algorithm, CSA is employed as a post-optimization algorithm due to the quality importance of CSA's initial solution. The result shows better performance when the initial explanation of CSA is better. Other applications of CSA include energy efficiency of wireless sensor network optimization [48], manufacturing optimization for milling operation [49], scheduling optimization in flexible manufacturing [50]; knapsack problem [51]; and transportation road network problem [52]. Due to its promising and efficient result, we believe that CSA can also solve the CSLP problem and provide a comparative solution with competitive computation time.

The rest of this paper is structured: section 2 presents the CSLP problem that will be tackled in this study. Section 3 introduces the essential components, the modification, and the procedure of the CSA for solving the CSLP problem. The displays of the comparative result analysis of the proposed algorithm with other similar algorithms are delivered in section 4. Section 5 presents the conclusion and future direction.

2. PROBLEM DESCRIPTION: CONSTRUCTION SITE LAYOUT PLANNING MODEL

The CSLP emphases on the physical prearrangement of temporary facilities at the construction site. The temporary facilities comprise locations, machinery, or divisions such as warehouses, fabrication shops, and maintenance shops, based on the project's size, design, location, and the organization of the construction work. Like most layout planning issues, CSLP contains actions for defining the scope and form of temporary facilities, establishing constraints between temporary facilities, and deciding on temporary facility positions, among others, to optimize each purpose. It also should include safety, site accessibility, material handling, information signs, security, and functional areas such as storage, fabrication shop, staff housing, and office. The goals of an improved and effective operating site layout are to optimize construction operations by lowering project time and expense and boosting workers' productivity.

This study modeled CSLP as a quadratic assignment problem (QAP). The goal is to find the best way to assign *n* facilities to the *n* or *m* (*m*>*n*) available locations. Some assumptions are that each temporary facility may fit into all the pre-set general sites/locations; (ii) each temporary facility will be installed in an identical region; (iii) each assignment is solved separately. The objective is to minimize the distance traversed for activities conducted from all temporary facilities. The overall distance depends on the distance from one temporary facility and the frequency of their journeys. The following equation may represent the optimization model of the research:

$$Min\ TD = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ik} d_{jl} x_{ij} x_{kl} \quad (1)$$

$$Subject\ to\ \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,\dots,n \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\dots,n \quad (3)$$

$$x_{ij} \in \{0,1\}, i = 1,2,\dots,n, j = 1,2,\dots,n \quad (4)$$

The objective, as stated above, is to minimize total travel distance, with n as the number of temporary facilities, and x_{ij} and x_{kl} represent the assignment matrix of the decision variable. x_{ij}=1 if the temporary facility *i* is assigned to location *j*, and $x_{ij} = 0$ if otherwise. Likewise, $x_{kl} = 1$ if temporary facility *k* is assigned to location *l*, and $x_{kl} = 0$ if otherwise. The frequency of trips from one temporary facility *i* to temporary facilities *k* is represented by f_{ik}, and d_{jl} is the distances between temporary facility *j* and *l*.

If the site is a multi-storey building, the layout design must be done on the completed floors of a multi-storey building. In this case, all the vertical movements for transporting construction materials and horizontal movements between temporary facilities are considered. The construction materials are placed in storage rooms, requiring vertical movement from the ground floor if necessary. Some additional assumptions include (a) the building has an *O (o=1, 2…O)* number of floors, where the first p floor can be used as a storage room once finished; (b) there are q types of materials, and each floor might have several storage rooms. Each storage room can only be used for one type of material; (c) each storage room is considered a cell, and there are r cells; (d) the distance traveled includes both horizontal and vertical movement; (e) the physical and demand quantities are fixed and known in advance; (f) there is only one material hoist; (g) there are enough storage rooms to hold all of the needed materials; and (h) loading and unloading are not taken into account in the hoist system and cost calculations.

Annotation:

q = Index of material types.

p = Index of a building's floor used for material storage as supply sources.

r = Index of storage cells on building floors.

o = Index of a building floor requesting the materials.
Q = The total materials.
P = The total number of floors for storage.
R = The total number of cells on the building level.
O = The total number of levels.
S_{oq} = Floor o's demand in a building for material type *m*.
C_q^h= cost of horizontal unit transportation of material type q.
C_{qp}^v= cost of vertical unit transportation of material type *q* to a building's floor *p* from the ground.
C_{po}^v = cost of vertical unit transportation of material type *j* to a building's floor *o* from the ground.
D_{pr} = distance from the material hoist on level *p* to cell *r*.
X_{qpr} = Binary decision variable storing material *q* on level *p* inside cell *r*.
δ_{qpro} = Binary-type variable where one means material *q* is transferred from floor *p* cell *r* to floor *o*, otherwise zero.

The objective function and constraints for multi-storey buildings are revised as:

$$\min\left(\sum_{q=1}^{Q}\sum_{p=1}^{P}\sum_{r=1}^{R} S_{oq}\left(C_{qp}^v + D_{pr}C_q^h\right)X_{qpr} + \sum_{o=1}^{O}\sum_{p=1}^{P} S_{oq}\left(C_{po}^v - C_{qp}^v\right)\delta_{qpro}\right) \quad (5)$$

Subject to

$$\sum_{p=1}^{P}\sum_{r=1}^{R} X_{qpr} = 1 \qquad q = 1,2,3,\dots,Q \quad (6)$$

$$d = \left|p_q - p\right| \quad (7)$$

$$X_{rp} = \sum_{q}^{Q} X_{qpr} \;\in [0,1] \quad (8)$$

$$x_{qpr} \,\epsilon\, \{0,1\}, q = 1,2,3,\dots,Q, p = 1,2,3,\dots,P, r = 1,2,3,\dots,R \quad (9)$$

$$p' \in \{1,2,\dots.,P_T\} \quad (10)$$

In (5), the cost of using a material handler to move materials from the ground floor to the storage floor is calculated. The horizontal cost to transport materials from the elevator to the storage cell is also included. Constraint (7) calculates the vertical movement of the material. Constraint (8) explained that one storage cell could accommodate one material item only. We created a set of binary decision variables representing the materials' storage location in (9). The total cost of material distribution from the ground floor to each floor that requires materials, is the second component. By setting this decision variable, the transportation cost will not be counted if the storage cell is not assigned to store the material. The distribution of materials should generally be from the storage cell to each floor; thus, the quantity becomes S_q/P_T where S_q denoted demand of material item *q* at a defined time and P_T denoted the maximum number of levels in the tower block. The expectation is for requests for material type *q*, materials on every floor, and *o* to be presented for the calculation. The detailed mathematical model can be found Fung *et al*. [53].

3. PROPOSED METHODOLOGY: CUCKOO SEARCH ALGORITHM

As a nature-inspired algorithm, CSA is inspired by the cuckoo behavior of depositing eggs to different nests is believed to minimize the risk of eggs being harmed by other species. The cuckoo lays eggs in the host nest that has just been laid by the host egg, hoping that its egg will hatch faster than the host egg. However, there is also a possibility that the host cuckoo realized the foreign egg. In that case, the host cuckoo will either throw the alien egg or abandon the nest and build a new nest in another location.

In CSA, the eggs in the nest are the number of pool candidate solutions for the problem. If the number of egg deposits on the nest equals one, the nest and the egg are considered the same solution. Iteratively, new solutions or new cuckoo eggs are found by randomly picking a cuckoo and generating a unique solution. Cuckoos with better value will replace some of the so-called "worst" nests. In the simplest form of the algorithm, some modifications of the actual behavior are made by Yang and Deb [40], including (a) each cuckoo laying only one egg at a time and randomly choosing the nest to deposit their egg, (b) only nest with worst solutions is abandoned, and nest with the best solution is kept achieving the best solution in each generation/iteration, and (c) number of nests are assumed to be fixed throughout the whole iteration.

3.1. Solution representation

This study employs a generic decoding method for converting a continuous-based number to an integer-based value. A layout arrangement is generated based on the continuous-based number in each index array, which are ordered ascendingly. The sorting function converts the real-valued CSA solution to an integer value, and the sorted element is returned in ascending order. This layout arrangement depicts the one-to-one relationship between locations and facilities. Each sequence consists of an index and a value associated with the sites and facilities it should be allocated. The example of solutions representation consists of seven locations, and seven facilities that need to be arranged in each area can be seen in Figure 1. A random real value (0~1) is generated for each facility. The real value associated with the facility is then sorted in ascending order, implying the facility's placement in the location.

Location	1	2	3	4	5
Facility	1	2	3	4	5
Real Value associated with facility	0.43	0.28	0.77	0.99	0.18

After sorting

Location	1	2	3	4	5
Facility	5	2	1	3	4
Real Value associated with facility	0.18	0.28	0.43	0.77	0.99

Final solution

Location	1	2	3	4	5
Facility	5	2	1	3	4

Figure 1. Solution representation

3.2. Detailed algorithm

The algorithm generates an initial population of *N* host nests. Each nest will have one egg each, representing solution x_i with *i*=1, 2, …., *N*. In each iteration, a random cuckoo is selected using Levy flight. For each iteration *t*, a cuckoo egg *i*, is chosen randomly, and new solutions x_i^{t+1} are produced. This random search is executed efficiently by using Lévy flights which the steps are defined as the step lengths that follow a particular probability distribution. In Levy flights, the paths of the steps must be isotropic and random. In this case, the general for the Lévy flight is given by:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(s, \lambda) \tag{6}$$

The superscript *t* is used to indicate the current generation, the symbol indicates the entry-wise multiplication, and α>0 indicates the step size. The product $\oplus$ means entry-wise multiplications. This step size determines how far a particle can move by random walk for a fixed number of iterations. The Lévy distribution modulates the transition probability of the Lévy flights in (6). Levy flights essentially provide a random walk while their unexpected steps are drawn from a Levy distribution for a significant step, as (7).

$$Levy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \tag{7}$$

The production of random numbers with Lévy flights is divided into two processes: first, a random direction based on a uniform distribution is chosen; second, a sequence of steps based on the desired Lévy distribution is generated. Figure 2 depicts the suggested CSA's pseudocode. Three principles guide the method used in this study: for each iteration, (i) the best solution is kept for the following generation; (ii) a move along the levy flight provides a new solution, and (iii) a proportion of the worse nests/solutions are excluded and replaced with freshly generated solutions. In this investigation, we assume that a nest may only accommodate one egg; consequently, both may be solutions. The computer chooses a solution randomly from the available solutions and then explores the surrounding surroundings using levy-flight. It takes its position if the new answer has a higher goal value than another randomly chosen option. (iii) the algorithm replaces abandoned nests with best-neighbor nests.

Algorithm 1. Cuckoo search pseudo-code

Input: number of nests *N*, problem size *dim*
Output: the best nest/solution x_i^t
while ($t < MaxGeneration$) or (stop criterion);
    Get a cuckoo (say i) randomly by Levy flights;
    Evaluate its quality/fitness F_i;
    Choose a nest among n (say j) randomly;
    if ($F_i > F_j$),
        Replace j with the new solution;
    End
    Abandon a fraction of worse nests
    [and build new ones at new locations via Levy flights];
    Keep the best solutions (or nests with quality solutions);
    Rank the solutions and find the current best;
end while
Postprocess results and visualization;

Figure 2. CSA pseudocode for CSLP

3.3. Parameter used

In our proposed algorithm, the model parameters to be determined include the number of nests (*n*), number of iterations (*N_IterTotal*), and discovery rate (*dr*). A Taguchi method is used to determine the best combinations of parameters for all case studies. The combination of the parameter values is number of nest (*n*)=15, 25, 50; number of iteration (*N_IterTotal*)=1000, 2000, 3000; and discovery rate (*dr*)=0.1, 0.25, 0.4. One parameter at a time is altered to determine its influence on the objective value. Based on the results of parameters setting, the best solution quality was obtained by setting the parameters number of the nest (*n*)=25; some iteration (*N_IterTotal*)=2000; and discovery rate (*dr*)=0.25.

4. RESULTS AND DISCUSSION

As a stopping criterion, the maximum number of generation G was chosen. The proposed technique was written in MATLAB and ran on an Intel® CoreTM i7 2600 CPU with 3.4 GHz and 4 GB RAM. We employed three different building site layout challenges to test the performance of the proposed CSA method. Case study 1 is taken from Li and Love [19], case study 2 is from Fung *et al*. [53] and case study 3 is derived from Prayogo *et al*. [3]. The first and third case studies are solved according to (1) to minimize the distance traveled for all activities performed in temporary facilities. The second case study is the storage layout problem in multi-stories buildings, with the fitness function as (4). This section discusses the comparative numerical findings of each algorithm.

4.1. Case study 1

The key objective in this case study 1 is to design the temporary construction facility layout to minimize the overall travel distance of employees' sites between various facilities as explained by the objective function (1). The data set includes 11 facilities and 11 sites, and the position of the side gate and the main gate is fixed. Detailed data set for case study 1 can be found in Li and Love [19]. The algorithm is run 20 times to find the best and average objective value. Table 1 shows the comparison results of the proposed algorithm with other algorithm in case 1.

In Table 1, we found that our proposed CSA algorithm can solve the CSLP and achieve an optimal solution. It outperforms the previous studies using GA and tabu search in terms of the best solution. Further, aside from the optimality, the standard deviation is smaller than other algorithms if we compare our CSA algorithm with the CBO, symbiotic organisms search (SOS), and SOS with the local search algorithm. This result shows that the proposed CSA is robust and can effectively solve CSLP. Furthermore, the optimal result is always achieved from 20 runs conducted during our computational experiment.

Table 1. Comparison results between proposed CSA with other algorithms in case 1

Algorithms	Results			Best layout
	Best	Average	Standard deviation	
Genetic algorithm [19]	15,090	N/A	N/A	11 5 8 7 2 9 3 1 6 4 10
Tabu search [54]	12,880	N/A	N/A	8 11 5 7 9 3 6 1 2 4 10
Colliding bodies optimization [33]	12,546	12,558	45.51	9 11 5 6 7 4 3 1 2 8 10
Symbiotic organism search [3]	12,546	12,560.56	42.52	9 11 5 6 7 4 3 1 2 8 10
Symbiotic organism search with local search [3]	12,546	12,553.86	19.37	9 11 5 6 7 4 3 1 2 8 10
Cuckoo search algorithm	12,546	12,546	0	9 11 5 6 7 4 3 1 2 8 10

4.2. Case study 2

Regarding case study 2, we compared it to three other studies that provided varying solutions to the problem. The first study used GA to tackle the problem, the second used mixed integer programming, and the latter two used SOS as the basis for their algorithms. As Table 2, the CSA algorithm and SOS with local search provide the most significant outcomes at the lowest cost compared to other algorithms. Despite finding the optimal solution, our proposed CSA delivers a better deviation than SOS using a local search. Thus, it is evident that the suggested CSA algorithm can tackle the more difficult CSLP example and surpass earlier research. If the number of iterations increases to 3000, regardless of the other parameter values, the method can find an optimal solution with zero variance. In case 2, however, further iterations are required despite the increased processing time, which is acceptable given that the CSLP can be considered a strategic decision.

Table 2. Comparison results between proposed CSA with other algorithms in case 2

Algorithms	Results		
	Best	Average	Standard deviation
Genetic algorithm [53]	4,562,620	N/A	N/A
Mixed integer programming [5]	4,293,020	N/A	N/A
Symbiotic organism search [3]	4,288,196	4,288,196	0.00
Symbiotic organism search with local search [3]	4,287,996	4,288,083.12	61.30
Cuckoo search algorithm	4,287,996	4,288,006.00	28.65

4.3. Case study 3

This case study aims to minimize the overall travel distance of site employees between various facilities, including ten facilities and ten sites. The entrance gate and guard post are permanently installed at locations 4 and 5. A detailed case study can be found by Prayogo *et al*. [55]. As we can see from Table 3, the CSA algorithm gives the minimum cost compared with other approaches in terms of best results. As Table 3, CSA still produces a better deviation than SOS, although both find the best solution. Our proposed algorithm has the best variation compared with SOS with local search. Thus, we can see that the proposed CSA algorithm can be compared and proved to be a robust methodology for solving several cases of CSLP.

Table 3. Comparison results between proposed CSA with other algorithms in case 3

Algorithms	Results		
	Best	Average	Standard deviation
Particle swarm optimization [55]	39,184	39,327.07	303.011
Artificial bee colony [55]	39,184	41,733.77	2,013.849
Symbiotic organism search [55]	39,184	39,243.40	274.206
Symbiotic organism search with local search [3]	39,184	39,184	0.00
Cuckoo search algorithm	39,184	39,184	0.00

4.4. Implication

The computational results imply that the suggested CSA can assist a decision-maker or project manager by selecting the ideal site and facility layout design, which is particularly important for construction projects. In reality, a project manager may need to compare the algorithm's recommended output with the situation on the ground. A well-analyzed and evaluated response provided by the algorithm can increase the quality of the decision. In addition, much attention is necessary based on the findings of case studies 1, 2, and 3, which cover single- and multi-floor facility layout challenges. The proposed CSA adds multiple phases to the local search procedure to avoid becoming stuck in the local ideal. In addition, various real-world scenarios may involve more complex limitations and decision considerations, needing additional modifications to the CSA to handle the issue.

5. CONCLUSION

This study proposed the CSA as a solution to the challenge of construction site layout, which organized many departments according to available space. Despite the situation's seeming simplicity, building site organization is crucial for the smooth flow of materials, workers, and information. This study examined two building site issues, one requiring only horizontal consideration and the other requiring both horizontal and vertical consideration. The suggested approach achieves the optimal solution for three CSLP situations and has a promising performance. Regardless, an extension or hybridization of the CSA method is still required to handle discrete issues, such as construction layout challenges. Future work should include other factors in the model, such as loading and unloading time/cost.

ACKNOWLEDGEMENTS

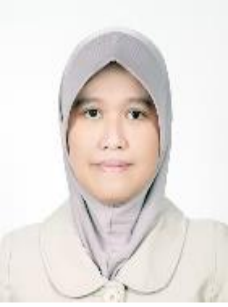
This research is funded by Department of Industrial Engineering Universitas Islam Indonesia and Sampoerna University.

REFERENCES

[1] E. Elbeltagi and T. Hegazy, "A hybrid Al-based system for site layout planning in construction," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 79–93, Mar. 2001, doi: 10.1111/0885-9507.00215.
[2] T. W. Liao, P. J. Egbelu, B. R. Sarker, and S. S. Leu, "Metaheuristics for project and construction management – a state-of-the-art review," *Automation in Construction*, vol. 20, no. 5, pp. 491–505, Aug. 2011, doi: 10.1016/j.autcon.2010.12.006.
[3] D. Prayogo *et al.*, "A novel hybrid metaheuristic algorithm for optimization of construction management site layout planning," *Algorithms*, vol. 13, no. 5, p. 117, May 2020, doi: 10.3390/a13050117.
[4] C. K. Wong, I. W. H. Fung, and C. M. Tam, "Comparison of using mixed-integer programming and genetic algorithms for construction site facility layout planning," *Journal of Construction Engineering and Management*, vol. 136, no. 10, pp. 1116–1128, Oct. 2010, doi: 10.1061/(ASCE)CO.1943-7862.0000214.
[5] C. Huang, C. K. Wong, and C. M. Tam, "Optimization of material hoisting operations and storage locations in multi-storey building construction by mixed-integer programming," *Automation in Construction*, vol. 19, no. 5, pp. 656–663, Aug. 2010, doi: 10.1016/j.autcon.2010.02.005.
[6] C. Huang and C. K. Wong, "Optimisation of site layout planning for multiple construction stages with safety considerations and requirements," *Automation in Construction*, vol. 53, pp. 58–68, May 2015, doi: 10.1016/j.autcon.2015.03.005.
[7] C. Huang and C. K. Wong, "Discretized cell modeling for optimal facility layout plans of unequal and irregular facilities," *Journal of Construction Engineering and Management*, vol. 143, no. 1, Jan. 2017, doi: 10.1061/(ASCE)CO.1943-7862.0001206.
[8] C. Huang and C. K. Wong, "Optimization of vertical elevator movements and material storage locations for high-rise building construction with overtime cost effects," *Journal of Computing in Civil Engineering*, vol. 33, no. 1, Jan. 2019, doi: 10.1061/(ASCE)CP.1943-5487.0000794.
[9] M. Park, Y. Yang, H.-S. Lee, S. Han, and S. Ji, "Floor-level construction material layout planning model considering actual travel path," *Journal of Construction Engineering and Management*, vol. 138, no. 7, pp. 905–915, Jul. 2012, doi: 10.1061/(ASCE)CO.1943-7862.0000493.
[10] A. W. A. Hammad, D. Rey, and A. Akbarnezhad, "A cutting plane algorithm for the site layout planning problem with travel barriers," *Computers & Operations Research*, vol. 82, pp. 36–51, Jun. 2017, doi: 10.1016/j.cor.2017.01.005.
[11] C. Liu and M. Lu, "Optimizing earthmoving job planning based on evaluation of temporary haul road networks design for mass earthworks projects," *Journal of Construction Engineering and Management*, vol. 141, no. 3, Mar. 2015, doi: 10.1061/(ASCE)CO.1943-7862.0000940.
[12] K. El-Rayes and H. Said, "Dynamic site layout planning using approximate dynamic programming," *Journal of Computing in Civil Engineering*, vol. 23, no. 2, pp. 119–127, Mar. 2009, doi: 10.1061/(ASCE)0887-3801(2009)23:2(119).
[13] J. Xu and X. Song, "Multi-objective dynamic layout problem for temporary construction facilities with unequal-area departments under fuzzy random environment," *Knowledge-Based Systems*, vol. 81, pp. 30–45, Jun. 2015, doi: 10.1016/j.knosys.2015.02.001.
[14] S. S. Kumar and J. C. P. Cheng, "A BIM-based automated site layout planning framework for congested construction sites," *Automation in Construction*, vol. 59, pp. 24–37, Nov. 2015, doi: 10.1016/j.autcon.2015.07.008.
[15] M. F. N. Maghfiroh, "Solving multi-objective paired single row facility layout problem using hybrid variable neighborhood search," *Jurnal Teknik Industri*, vol. 23, no. 2, pp. 171–182, Dec. 2021, doi: 10.9744/jti.23.2.171-182.
[16] A. Kaveh and Y. Vazirinia, "Construction site layout planning problem using metaheuristic algorithms: a comparative study," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 43, no. 2, pp. 105–115, Jun. 2019, doi: 10.1007/s40996-018-0148-6.
[17] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, Jul. 2013, doi: 10.1016/j.ins.2013.02.041.
[18] M. Xu, Z. Mei, S. Luo, and Y. Tan, "Optimization algorithms for construction site layout planning: a systematic literature review," *Engineering, Construction and Architectural Management*, vol. 27, no. 8, pp. 1913–1938, Jul. 2020, doi: 10.1108/ECAM-08-2019-0457.
[19] H. Li and P. E. D. Love, "Site-level facilities layout using genetic algorithms," *Journal of Computing in Civil Engineering*, vol. 12, no. 4, pp. 227–231, Oct. 1998, doi: 10.1061/(ASCE)0887-3801(1998)12:4(227).
[20] T. Hegazy and E. Elbeltagi, "EvoSite: evolution-based model for site layout planning," *Journal of Computing in Civil Engineering*, vol. 13, no. 3, pp. 198–206, Jul. 1999, doi: 10.1061/(ASCE)0887-3801(1999)13:3(198).
[21] M. J. Mawdesley and S. H. Al-Jibouri, "Proposed genetic algorithms for construction site layout," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 5–6, pp. 501–509, Aug. 2003, doi: 10.1016/j.engappai.2003.09.002.
[22] F. Zhou, S. M. AbouRizk, and H. AL-Battaineh, "Optimisation of construction site layout using a hybrid simulation-based system," *Simulation Modelling Practice and Theory*, vol. 17, no. 2, pp. 348–363, Feb. 2009, doi: 10.1016/j.simpat.2008.09.011.
[23] A. Khalafallah and K. H. Hyari, "Optimization parameter variation: improving biobjective optimization of temporary facility

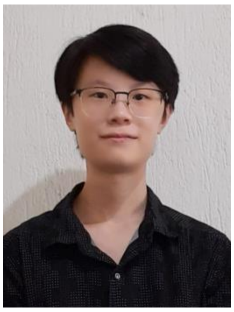
planning," *Journal of Computing in Civil Engineering*, vol. 32, no. 5, Sep. 2018, doi: 10.1061/(ASCE)CP.1943-5487.0000780.
[24] H. Zhang and J. Y. Wang, "Particle swarm optimization for construction site unequal-area layout," *Journal of Construction Engineering and Management*, vol. 134, no. 9, pp. 739–748, Sep. 2008, doi: 10.1061/(ASCE)0733-9364(2008)134:9(739).
[25] J. Xu and Z. Li, "Multi-objective dynamic construction site layout planning in fuzzy random environment," *Automation in Construction*, vol. 27, pp. 155–169, Nov. 2012, doi: 10.1016/j.autcon.2012.05.017.
[26] X. Song, Z. Zhang, J. Xu, Z. Zeng, C. Shen, and F. Peña-Mora, "Bi-stakeholder conflict resolution-based layout of construction temporary facilities in large-scale construction projects," *International Journal of Civil Engineering*, vol. 16, no. 8, pp. 941–964, Aug. 2018, doi: 10.1007/s40999-017-0233-4.
[27] E. Gharaie, A. Afshar, and M. R. Jalali, "Static site layout optimization with ACO algorithm," *WSEAS Transactions on Systems*, vol. 5, no. 4, pp. 685–690, 2006.
[28] X. Ning, K.-C. Lam, and M. C.-K. Lam, "A decision-making system for construction site layout planning," *Automation in Construction*, vol. 20, no. 4, pp. 459–473, Jul. 2011, doi: 10.1016/j.autcon.2010.11.014.
[29] X. Ning and K. C. Lam, "Cost–safety trade-off in unequal-area construction site layout planning," *Automation in Construction*, vol. 32, pp. 96–103, Jul. 2013, doi: 10.1016/j.autcon.2013.01.011.
[30] X. Ning, J. Qi, C. Wu, and W. Wang, "Reducing noise pollution by planning construction site layout via a multi-objective optimization model," *Journal of Cleaner Production*, vol. 222, pp. 218–230, Jun. 2019, doi: 10.1016/j.jclepro.2019.03.018.
[31] R. Gholizadeh, G. G. Amiri, and B. Mohebi, "An alternative approach to a harmony search algorithm for a construction site layout problem," *Canadian Journal of Civil Engineering*, vol. 37, no. 12, pp. 1560–1571, Dec. 2010, doi: 10.1139/L10-084.
[32] L.-C. Lien and M.-Y. Cheng, "Particle bee algorithm for tower crane layout with material quantity supply and demand optimization," *Automation in Construction*, vol. 45, pp. 25–32, Sep. 2014, doi: 10.1016/j.autcon.2014.05.002.
[33] A. Kaveh, M. Khanzadi, M. Alipour, and M. R. Moghaddam, "Construction site layout planning problem using two new meta-heuristic algorithms," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 40, no. 4, pp. 263–275, Dec. 2016, doi: 10.1007/s40996-016-0041-0.
[34] M.-Y. Cheng and N.-W. Chang, "Dynamic construction material layout planning optimization model by integrating 4D BIM," *Engineering with Computers*, vol. 35, no. 2, pp. 703–720, Apr. 2019, doi: 10.1007/s00366-018-0628-0.
[35] B. Kalmár, A. Kalmár, K. Balázs, and L. T. Kóczy, "Construction site layout and building material distribution planning using hybrid algorithms," in *Issues and Challenges of Intelligent Systems and Computational Intelligence*, 2014, pp. 75–88.
[36] K.-C. Lam, X. Ning, and M. C.-K. Lam, "Conjoining MMAS to GA to solve construction site layout planning problem," *Journal of Construction Engineering and Management*, vol. 135, no. 10, pp. 1049–1057, Oct. 2009, doi: 10.1061/(ASCE)0733-9364(2009)135:10(1049).
[37] A. Kaveh and M. Rastegar Moghaddam, "A hybrid WOA-CBO algorithm for construction site layout planning problem," *Scientia Iranica*, p. 0, Aug. 2017, doi: 10.24200/sci.2017.4212.
[38] R. Abu Khurma, I. Aljarah, A. Sharieh, M. Abd Elaziz, R. Damaševičius, and T. Krilavičius, "A review of the modification strategies of the nature inspired algorithms for feature selection problem," *Mathematics*, vol. 10, no. 3, p. 464, Jan. 2022, doi: 10.3390/math10030464.
[39] Y. Zhou, J. Xie, L. Li, and M. Ma, "Cloud model bat algorithm," *The Scientific World Journal*, vol. 2014, pp. 1–11, 2014, doi: 10.1155/2014/237102.
[40] X.-S. Yang and Suash Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.
[41] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," *Applied Soft Computing*, vol. 61, pp. 1041–1059, Dec. 2017, doi: 10.1016/j.asoc.2017.02.034.
[42] A. Iglesias *et al.*, "Cuckoo search algorithm with lévy flights for global-support parametric surface approximation in reverse engineering," *Symmetry*, vol. 10, no. 3, p. 58, Mar. 2018, doi: 10.3390/sym10030058.
[43] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu, "Chaos-enhanced cuckoo search optimization algorithms for global optimization," *Applied Mathematical Modelling*, vol. 40, no. 5–6, pp. 3860–3875, Mar. 2016, doi: 10.1016/j.apm.2015.10.052.
[44] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1659–1669, Jun. 2014, doi: 10.1007/s00521-013-1402-2.
[45] X. Ouyang, Y. Zhou, Q. Luo, and H.-W. Chen, "A novel discrete cuckoo search algorithm for spherical traveling salesman problem," *Applied Mathematics & Information Sciences*, vol. 7, pp. 777–784, 2013.
[46] S. Kang, M. Kim, and J. Chae, "A closed loop based facility layout design using a cuckoo search algorithm," *Expert Systems with Applications*, vol. 93, pp. 322–335, Mar. 2018, doi: 10.1016/j.eswa.2017.10.038.
[47] E. Teymourian, V. Kayvanfar, G. H. M. Komaki, and M. Zandieh, "Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem," *Information Sciences*, vol. 334–335, pp. 354–378, Mar. 2016, doi: 10.1016/j.ins.2015.11.036.
[48] M. Dhivya, M. Sundarambal, and L. N. Anand, "Energy efficient computation of data fusion in wireless sensor networks using cuckoo based particle approach (CBPA)," *International Journal of Communications, Network and System Sciences*, vol. 04, no. 04, pp. 249–255, 2011, doi: 10.4236/ijcns.2011.44030.
[49] A. R. Yildiz, "Cuckoo search algorithm for the selection of optimal machining parameters in milling operations," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 1–4, pp. 55–61, Jan. 2013, doi: 10.1007/s00170-012-4013-7.
[50] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5–8, pp. 951–959, Feb. 2013, doi: 10.1007/s00170-012-4061-z.
[51] A. Gherboudj, A. Layeb, and S. Chikhi, "Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, p. 229, 2012, doi: 10.1504/IJBIC.2012.048063.
[52] O. Baskan, "Determining optimal link capacity expansions in road networks using cuckoo search algorithm with lévy flights," *Journal of Applied Mathematics*, vol. 2013, pp. 1–11, 2013, doi: 10.1155/2013/718015.
[53] W. H. Fung, Ivan, C. K. Wong, C. M. Tam, and T. K. L. Tong, "Optimizing material hoisting operations and storage cells in single multi-storey tower block construction by genetic algorithm," *International Journal of Construction Management*, vol. 8, no. 2, pp. 53–64, Jan. 2008, doi: 10.1080/15623599.2008.10773115.
[54] L. Y. Liang and W. C. Chao, "The strategies of tabu search technique for facility layout optimization," *Automation in Construction*, vol. 17, no. 6, pp. 657–669, Aug. 2008, doi: 10.1016/j.autcon.2008.01.001.
[55] D. Prayogo, J. C. Sutanto, H. E. Suryo, and S. Eric, "A comparative study on bio-inspired algorithms in layout optimization of construction site facilities," *Civil Engineering Dimension*, vol. 20, no. 2, pp. 102–110, Oct. 2018, doi: 10.9744/ced.20.2.102-110.

BIOGRAPHIES OF AUTHORS



Meilinda Fitriani Nur Maghfiroh graduated from the Tokyo Institute of Technology, Japan, and currently works as an Industrial Engineering Lecturer in Universitas Islam Indonesia. Her research focuses on optimization and metaheuristics and different application, including facility layout, vehicle routing problems, and location problems. She can be contacted at email: meilinda.maghfiroh@uii.ac.id.

Anak Agung Ngurah Perwira Redi is currently a Lecturer in the Department of Industrial Engineering -at Sampoerna University. Formerly a research fellow at Monash University, Australia. He received his Ph.D. in Industrial Management from the National Taiwan University of Science and Technology in 2017. His current research interests focus on developing algorithms for optimization and their application in logistics and SCM problems. He can be contacted at email: wira.redi@gmail.com.



Janice Ong is a freshman at Sampoerna University majoring in Mechanical Engineering. She has high interests in music, technology, and aviation. She can be contacted at email: janice.ong@sampoernauniversity.ac.id.

Muhamad Rausyan Fikri is an Information Systems lecturer at Sampoerna University. He received his Bachelor of Science (B.Sc.) at Universitas Gadjah Mada, Department of Computer Science and Electronics, in 2015 and Master of Engineering (M.Eng.) at Tokyo Institute of Technology, Department of Systems and Control Engineering in 2018. Currently, he is pursuing his Ph.D. degree at Tampere University, Finland. His research interests include bioinspired robots, cooperative control, and IoT. He is currently also the Coordinator of IoT Lab and Cyber-Physical Research group at Sampoerna University. He can be contacted at email: muhamad.fikri@sampoernauniversity.ac.id.