



Construction Site Layout Planning Problem Using Metaheuristic Algorithms: A Comparative Study

Ali Kaveh¹ · Yasin Vazirinia¹

Received: 28 December 2017 / Accepted: 2 August 2018 / Published online: 17 August 2018
© Shiraz University 2018

Abstract

Material handling is one of the essential activities in the construction industry. Suitable location of facilities in the construction site can affect the costs and duration of the construction material handling process. The construction site layout planning to supply material and engineering demands within the minimum transportation distance is a quadratic assignment problem. Metaheuristics are widely used to solve construction site layout planning problems. In this article, the performance of four metaheuristic algorithms called charged system search, whale optimization algorithm, vibrating particles system, and enhanced vibrating particles system (EVPS) are compared in terms of their effectiveness in resolving a practical construction site layout problem. Results show that EVPS performs better than the other three methods.

Keywords Construction site layout planning · Quadratic assignment problem · Vibrating particles system · Metaheuristic algorithm · Material handling

1 Introduction

Material handling is one of the essential activities in the construction industry. Appropriate choice of the location of facilities in the construction site affects the costs and duration of the construction material handling process considerably. Based on a study carried out by Tompkins et al. (2010), between 20 and 50% of the total operating costs in manufacturing companies are spent on material handling and a good position for facilities can reduce these costs by at least 10–30% (Tompkins et al. 2010). Another study shows that material handling costs will increase about 36% with an ineffectual layout (Balakrishnan and Cheng 2007). In the recent decades, several research studies have been carried out to provide the best methods to solve the Construction Engineering Optimization Problems (CEOPs). Metaheuristics have been applied extensively to solve construction site layout planning problems (Abdelmegid et al. 2015; Wang et al. 2015; Kaveh et al. 2016;

Kaveh and Rasteghar Moghaddam 2017; Kaveh and Vazirinia 2017, 2018). An appropriate construction site layout can increase the production efficiency. Recently, marketing competition drives many building companies to make good organization and management for their construction sites in order to ensure good productivity and new margins of profitability (Alkriz and Mangin 2005). The construction site layout problems (CSLPs) to supply material and engineering demands within the minimum transportation distance are the interesting CEOPs, because they brought the consideration of layout esthetics and usability qualities into the facility design process. The construction site layout planning (CSLP) is a quadratic assignment problem. Every construction project requires enough spaces for temporary facilities for performing the construction activities in a safe and efficient manner. Construction site-level facilities layout is an important step in site planning. Planning construction site spaces to allow for safe and efficient working status is a complex and multi-disciplinary task as it involves considering a wide range of scenarios. CSLPs are known as combinatorial optimization problems. There are two methods for the solution consisting of metaheuristics for large search sized problems and the exact method with a global search for smaller search sized problems (Kaveh 2017). For example,

✉ Ali Kaveh
alikhavah@iust.ac.ir

¹ Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Narmak, Tehran, Iran

Li and Love (1998) developed a construction site-level facility layout problem for allocating a set of predetermined facilities into a set of predetermined locations, while satisfying the layout constraints and requirements. They applied the genetic algorithm to solve the CSLP by assuming that the predetermined locations are in rectangular shape and are large enough to accommodate the largest facility. Gharai et al. (2006) resolved their model by ant colony optimization. Kaveh et al. (2012) used an improved harmony search algorithm for facility layout optimization. Kaveh et al. (2016) utilized the colliding bodies optimization (CBO) and its enhanced version (ECBO), resolving a practical construction site layout problem.

Solving real-life problems by metaheuristic algorithms has become an interesting topic in recent years. Many metaheuristics with different philosophy and characteristics are developed and applied to a wide range of fields. The objective of these optimization methods is to efficiently explore the search space in order to find global or near-global solutions. Since these algorithms are not problem specific and do not require the derivatives of the objective function, they have received increasing attention from both academia and industry (Kaveh and Ilchi Ghazaan 2017). Metaheuristic methods are global optimization methods that try to reproduce natural phenomena (genetic algorithm (Golberg 1989), particle swarm optimization (Eberhart and Kennedy 1995), whale optimization algorithm (WOA)), humans social behavior (imperialist competitive algorithm (Atashpaz-Gargari and Lucas 2007)), or physical phenomena (ray optimization (RO) (Kaveh and Khayatazad 2013), charged system search (CSS) (Kaveh and Talatahari 2010), colliding bodies optimization (Kaveh and Mahdavi 2014), big bang–big crunch (Erol and Eksin 2006), vibrating particles system (VPS) (Kaveh and Ilchi Ghazaan 2017), thermal exchange optimization (Kaveh and Dadras 2017)). Exploitation and exploration are two important characteristics of metaheuristic optimization methods. Exploitation serves to search around the current best solutions and to select the best possible points, and exploration allows the optimizer to explore the search space more efficiently, often by randomization (Kaveh 2017).

In this paper, charged system search (CSS), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), vibrating particles system (VPS) (Kaveh and Ilchi Ghazaan 2017), and enhanced vibrating particle system (EVPS) (Kaveh et al. 2017) are used to optimize the construction site layout problem. This model is related to the flow of materials among facilities.

2 Optimization Algorithms

2.1 Charged System Search

Charged system search is an optimization algorithm proposed by Kaveh and Talatahari (2010) and has its governing rules inspired by electrostatics and Newtonian mechanics. Each candidate solution in this algorithm is assumed as a charged sphere, called charged particle (CP). These charged spheres impose electric forces on each other. The force that each particle exerts on the others is determined according to its electric charge and using Eq. (1):

$$q_i = \frac{\text{fit}(i) - \text{fitworst}}{\text{fitbest} - \text{fitworst}} \quad (1)$$

where $\text{fit}(i)$ is the objective function value of the i th particle; fitbest and fitworst are the best and worst cost function values among all particles, respectively. As well as the electric charges, the force that a particle imports on another one depends on their relative distance, which is defined as a dimensionless quantity by Eq. (2):

$$r_{ij} = \frac{X_i - X_j}{(X_i + X_j)/2 - X_{\text{best}} + \varepsilon} \quad (2)$$

where X_i and X_j are the positions of the i th and j th particles; X_{best} is the position of the best current CP, and ε is a small positive number to avoid division by zero. It covers the probability that the current best record is in the middle point between two charged particles.

All better particles attract worse ones, but only some of the worse particles attract better ones. This rule can be mathematically stated as follows:

$$p_{ij} = \begin{cases} 1 \leftrightarrow \frac{\text{fit}(i) - \text{fitworst}}{\text{fitbest} - \text{fitworst}} > \text{rand, or, } \text{fit}(i) > \text{fit}(j) \\ 1 \leftrightarrow \text{else} \end{cases} \quad (3)$$

where p_{ji} is the probability of the j th particle being attracted by the i th particle. The resultant electric force F_j on the j th particle can be calculated by Eq. (4):

$$f_j = q_j \sum_{i, i \neq j} \left(\frac{q_i}{a^3} \cdot r_{ij} \cdot w_1 + \frac{q_i}{r_{ij}^2} \cdot w_2 \right) \cdot p_{ij} \cdot (X_i + X_j), \begin{cases} w_1 = 1, & w_2 = 2 \leftrightarrow r_{ij} < a \\ w_1 = 2, & w_2 = 1 \leftrightarrow r_{ij} \geq a \\ j = 1, 2, \dots, n \end{cases} \quad (4)$$

where a is the radius of the charged sphere. It helps to discern global and local search phases as discussed by Kaveh and Talatahari (2010). When the final forces are determined, the new position for the j th particle can be calculated by Eq. (5):

Fig. 1 Pseudocode of the charged system search algorithm (Kaveh and Zolghadr 2014)

Pseudo code of Charged System Search (CSS)

```

Initialize algorithm parameters
Initialize a population of random solutions
Evaluate the cost function values and determine the charges of the CPs
While maximum iterations is not fulfilled
  for each particle
    Determine the resultant force acting on the particle
    Determine the velocity vector for the particle
    Determine the new position for the particle
  end for
  Evaluate the cost function values for the particles in new positions
  Record the best particles in Charged Memory
  Update the charges of the particles
end While
end

```

$$X_{j,\text{new}} = \text{rand}_{j1} \cdot k_a \cdot \frac{f_j}{m_j} \cdot \Delta t^2 + \text{rand}_{j2} \cdot k_v \cdot V_{i,\text{old}} \cdot \Delta t + X_{j,\text{old}}, \quad (5)$$

$$V_{j,\text{new}} = \frac{X_{j,\text{new}} - X_{j,\text{old}}}{\Delta t} \quad (6)$$

where rand_{j1} and rand_{j2} are two random numbers uniformly distributed in the range (0, 1); m_j is the mass of the particle that is taken equal to q_j . Δt is the time interval and is taken as unity. k_a and k_v are two coefficients to control the exploration and exploitation tendencies of the algorithm and are called acceleration coefficient and velocity coefficient, respectively. Two coefficients are defined in Eqs. (7) and (8) to maintain more exploration at the early stages of the optimization process and more exploitation at the final stages.

$$K_a = c_1(1 + \text{iter}/\text{iter}_{\text{max}}) \quad (7)$$

$$K_v = c_2(1 - \text{iter}/\text{iter}_{\text{max}}) \quad (8)$$

where iter is the current iteration number and iter_{max} is the maximum number of iterations. In (Kaveh and Talatahari 2010), c_1 and c_2 are both suggested to be taken as 0.5. These parameter values have been determined through trial and error, which can be a time-consuming process. The pseudocode of the CSS algorithm is illustrated in Fig. 1.

2.2 Whale Optimization Algorithm (WOA)

The WOA is a novel population-based algorithm which mimics the social behavior of humpback whales. This metaheuristic algorithm has been developed by Mirjalili and Lewis (2016). In this method, the spiral bubble-net feeding maneuver is mathematically modeled to perform optimization. Bubble-net feeding is a unique behavior that can only be observed in humpback whales (Mirjalili and Lewis 2016). In order to update the position of the whales during optimization, two behaviors are identified, the shrinking encircling mechanism and the spiral bubble-net

feeding maneuver. Since the position of the optimum design in the search space is not known, the basic WOA algorithm assumes that the current best candidate solution is the optimum or is close to the optimum and the other search agents will update their positions toward the best search agent (Mirjalili and Lewis 2016). Similar to other multi-agent methods, the WOA starts with a set of random populations. At each iteration, search agents update their positions according to \vec{A} vector's value. Updating mechanism is detailed in following. This process continues until terminating criterion is satisfied. The WOA algorithm has two phases, exploitation and exploration phase. This algorithm transits between exploration and exploitation phase smoothly. The transition is done due to the variation of \vec{A} vector's value. \vec{A} vector's value is decreased during iterations, half of the iterations are assigned to exploration phase when $|\vec{A}| \geq 1$ and the other half is dedicated to exploitation when $|\vec{A}| < 1$. Here, the sign $||$ indicates the absolute value. The vector A is computed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (9)$$

where \vec{a} is linearly decreased from 2 to 0 over the course of iterations and r is a random vector in [0,1].

2.2.1 Exploitation Phase (Bubble-Net Attacking Method)

In order to model the bubble-net behavior of humpback whales mathematically, two approaches are considered: shrinking encircling mechanism and spiral updating position. Since the humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously, WOA assumes that there is a probability (p) of 50% to choose between these two behaviors. Shrinking encircling mechanism is modeled as Eqs. (10), (11), and (12):

$$\vec{C} = 2\vec{r} \quad (10)$$

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{\text{best}} - \vec{X} \right| \quad (11)$$

$$\vec{X}^{\text{new}} = \vec{X}_{\text{best}} - \vec{A} \cdot \vec{D} \quad (12)$$

where \vec{X}^{new} and \vec{X} are new position and previous position of whales, respectively. \vec{X}_{best} is the best position vector of the best solution obtained so far. \vec{A} and \vec{C} are coefficient vectors and $||$ is the absolute value.

Additionally, spiral-shaped movement of whales is simulated by Eqs. (13) and (14):

$$\vec{D}^{\rightarrow} = \left| \vec{X}_{\text{best}} - \vec{X} \right| \quad (13)$$

$$\vec{X}^{\text{new}} = \vec{D}^{\rightarrow} \cdot e^{b \cdot l} \cdot \cos(2\pi \cdot 1) + \vec{X}_{\text{best}} \quad (14)$$

where b is a constant that defines the spiral shape of movement and l is a random number in range $[-1, 1]$.

2.2.2 Exploration Phase (Search for Prey)

If $|A| \geq 1$ exploration phase has happened, WOA updates the position of a whale in the exploration phase according to a randomly chosen whale instead of the best search agent. Thus, new position is computed by Eqs. (15) and (16):

$$\vec{D}^{\rightarrow} = \left| \vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X} \right| \quad (15)$$

$$\vec{X}^{\text{new}} = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D} \quad (16)$$

The pseudocode of the WOA is shown in Fig. 2.

Fig. 2 Pseudocode of the whale optimization algorithm (Mirjalili and Lewis 2016)

Pseudo code of Whale Optimization algorithm (WOA)

```

Initialize the whales population  $X_i$  ( $i = 1, 2, \dots, n$ )
Calculate the fitness of each search agent
 $\vec{X}_{\text{best}}$  = the best search agent
While maximum iterations is not fulfilled
  for each search agent
    Update  $a, A, C, l$ , and  $p$ 
    If 1 ( $p < 0.5$ )
      If 2 ( $|A| < 1$ )
        Update the position of the current search agent by the Eq. (9)
      else if 2 ( $|A| \geq 1$ )
        Select a random search agent ( $\vec{X}_{\text{rand}}$ )
        Update the position of the current search agent by the Eq. (16)
      end if 2
    else if 1 ( $p \geq 0.5$ )
      Update the position of the current search agent by the Eq. (13)
    end if 1
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent
  Update  $X^*$  if there is a better solution
   $t = t + 1$ 
end while
Return  $\vec{X}_{\text{best}}$ 
end

```

2.3 Vibrating Particles System

The VPS is a population-based algorithm which simulates a free vibration of single degree of freedom systems with viscous damping (Kaveh and Ilchi Ghazaan 2017). Similar to other multi-agent methods, VPS has a number of individuals (or particles) consisting of the variables of the problem. In the VPS, each solution candidate is defined as “ X ” and contains a number of variables (i.e., $X_i = \{X_i^j\}$) and is considered as a particle. Particles are damped based on three equilibrium positions with different weights, and during each iteration, the particle position is updated by learning from them: (1) the historically best position of the entire population (HB), (2) a good particle (GP), and (3) a bad particle (BP). The solution candidates gradually approach to their equilibrium positions that are achieved from current population and historically best position in order to have a proper balance between diversification and intensification. Main procedure of this algorithm is defined as:

Step 1 Initialization

Initial locations of particles are created randomly in an n -dimensional search space, by Eq. (9):

$$x_i^j = x_{\min} + \text{rand} \times (x_{\max} - x_{\min}), \quad i = 1, 2, 3, \dots, n; \quad (17)$$

where x_i^j is the j th variable of the particle i . x_{\max} and x_{\min} are, respectively, the minimum and the maximum

allowable values vectors of variables; rand is a random number in the interval [0,1]; and n is the number of particles.

Step 2 Evaluation of candidate solutions

The objective function value is calculated for each particle.

Step 3 Updating the particle positions

In order to select the GP and BP for each candidate solution, the current population is sorted according to their objective function values in an increasing order, and then GP and BP are chosen randomly from the first and second half, respectively.

According to the above concepts, the particle's position is updated by Eq. (18):

$$x_i^j = \omega_1 \cdot [D \cdot A \cdot \text{rand1} + \text{HB}^j] + \omega_2 \cdot [D \cdot A \cdot \text{rand2} + \text{GP}^j] + \omega_3 \cdot [D \cdot A \cdot \text{rand3} + \text{BP}^j] \quad (18)$$

where x_i^j is the j th variable of the particle i . $\omega_1, \omega_2, \omega_3$ are three parameters to measure the relative importance of HB, GP, and BP, respectively ($\omega_1 + \omega_2 + \omega_3 = 1$). rand1, rand2, and rand3 are random numbers uniformly distributed in the range of [0, 1], respectively. The parameter A is defined as:

$$A = [\omega_1 \cdot (\text{HB}^j - x_i^j)] + [\omega_1 \cdot (\text{GP}^j - x_i^j)] + [\omega_1 \cdot (\text{BP}^j - x_i^j)]. \quad (19)$$

Parameter D is a descending function based on the number of iterations:

$$D = \left(\frac{\text{iter}}{\text{iter}_{\max}} \right)^{-\alpha}. \quad (20)$$

Fig. 3 Pseudocode of the vibrating particles system algorithm (Kaveh and Ilchi Ghazaan 2017)

Pseudo code of Vibrating Particles System (VPS)

```

Initialize algorithm parameters
Create initial positions randomly by Eq. (17)
Evaluate the values of objective function and store HB
While maximum iterations is not fulfilled
  for each particle
    The GP and BP are chosen
    if P<rand
       $w_3=0$  and  $w_2=1-w_1$ 
    end if
    for each component
      New location is obtained by Eq. (18)
    end for
    Violated components are regenerated by harmony search-based handling approach
  end for
end while
The values of objective function are evaluated and HB is updated
end

```

In order to have a fast convergence in the VPS, the effect of BP is sometimes considered in updating the position formula. Therefore, for each particle, a parameter like p within (0, 1) is defined, and it is compared with rand (a random number uniformly distributed in the range of [0, 1]) and if $p < \text{rand}$, then $\omega = 0$ and $\omega_2 = 1 - \omega_1$.

Three essential concepts consisting of self-adaptation, cooperation, and competition are considered in this algorithm. Particles move toward HB so the self-adaptation is provided. Any particle has the chance to influence on the new position of the other one, so the cooperation between the particles is supplied. Because of the p parameter, the influence of GP (good particle) is more than that of BP (bad particle), and therefore the competition is provided.

Step 4 Handling the side constraints

There is a possibility of boundary violation when a particle moves to its new position. In the proposed algorithm, for handling boundary constraints a harmony search-based approach is used (Kaveh and Talatahari 2010). In this technique, there is a possibility like harmony memory considering rate (HMCR) that specifies whether the violating component must be changed with the corresponding component of the historically best position of a random particle or it should be determined randomly in the search space. Moreover, if the component of a historically best position is selected, there is a possibility like pitch adjusting rate (PAR) that specifies whether this value should be changed with the neighboring value or not.

Step 5 Terminating condition check

Steps 2 through 4 are repeated until a termination criterion is fulfilled. Any terminating condition can be considered, and in this study, the optimization process is terminated after a fixed number of iterations. The pseudocode of the VPS is shown in Fig. 3.

Fig. 4 Pseudocode of the enhanced vibrating particles system algorithm (Kaveh and Vazirinia 2018)

Pseudo code of Enhanced Vibrating Particles System (EVPS)

```

Initialize algorithm parameters
Create initial positions randomly by Eq. (17)
Evaluate the values of objective function and update "Memory"
While maximum iterations is not fulfilled
  for each particle
    Select the GP, BP, and OHB
    if P<rand
       $W_3=0$  and  $w_2=1-w_1$ 
    end if
    for each component
      Update positions by Eq. (21)-(a), (b), or (c) with probability of  $\omega_1$ ,  $\omega_2$  and  $\omega_3$ 
    end for
    Regenerate violated components by harmony search-based handling approach
  end for
  end while
  The values of objective function are evaluated and HB is updated
end

```

2.4 Enhanced Vibrating Particles System

In this method, two new parameters are introduced as "Memory" and "OHB." Memory acts as HB with the difference that it saves NB number of the best historically positions in the entire population, and OHB (one of the best historically positions in entire population) is one row of memory that is selected randomly. HB is replaced with memory in the EVPS algorithm. Another change in the VPS algorithm is that Eqs. (18) and (19) should be replaced with Eqs. (21) and (22). In Eqs. (21) and (22), one of (a), (b), and (c) equations is applied with the probability of ω_1 , ω_2 , and ω_3 , respectively.

$$x_i^j = \begin{cases} [D \cdot A \cdot \text{rand1} + \text{OHB}^j] & (a) \\ [D \cdot A \cdot \text{rand2} + \text{GP}^j] & (b) \\ [D \cdot A \cdot \text{rand3} + \text{BP}^j] & (c) \end{cases} \quad (21)$$

$$A = \begin{cases} (\pm 1)(\text{OHB}^j - x_i^j) & (a) \\ (\pm 1)(\text{GP}^j - x_i^j) & (b) \\ (\pm 1)(\text{BP}^j - x_i^j) & (c) \end{cases} \quad (22)$$

where (± 1) is applied randomly. It should be noted that OHB, GP, and BP are determined for every particle independently. Other sections of the EVPS are defined exactly the same as in the VPS algorithm.

The pseudocode of the EVPS algorithm is illustrated in Fig. 4.

3 Problem: Site Precast Yard Layout Planning (cheung et al. 2002)

Construction site layout problem was formulated as a QAP. Originally, this formulation presented by Koopmans and Beckmann (1957) associates n facilities to n mutually

individual locations. This layout problem assumes that each facility consumes the same area, and so any facility can be allocated to any site (Koopmans and Beckmann 1957). A CSLP problem called site precast yard layout planning optimization model was presented by Cheung et al. (2002) that was solved by several models such as (Liang and Chao 2008; Wong et al. 2010; Kaveh et al. 2012, 2016; Kaveh and Rasteghar Moghaddam 2017). The performance of a site precast yard in the optimization of the construction site precast yard layout is very much affected by location of the various facilities (Wong et al. 2010).

Here, we are going to explain the site precast yard layout planning optimization problem.

In the model by Cheung et al. (2002), the following assumptions are considered:

- The geometric layout of available locations is fixed and predetermined.
- Just one facility can be located at every location.
- Number of facilities and locations are equal (if, the number of locations is greater than the number of facilities and dummy facilities can be added for computation purpose.).

3.1 Material Flow (Liang and Chao 2008)

In the model by Cheung et al. (2002), n facilities are located to n locations. These facilities are placed in appropriate locations based on their traveling frequencies and distance description. Several types of resources are considered in order to calculate the transportation costs between facilities in the model. The aim of layout planning is to achieve the lowest costs for the transportation of resources to facilities through appropriate site arrangement.

Each replacement by a facility from another facility will increase or decrease the total cost by evaluating the objective function. The total cost is defined as follows:

$$\text{Minimize Total Cost} = \sum_k^n \sum_i^n \sum_j^n \text{TCL}_{Mki,j} \quad (23)$$

where $\text{TCL}_{Mki,j}$ calculates by Eq. (24):

$$\text{TCL}_{Mki,j} = M_{LMi,j} * C_{Mk} \quad (24)$$

where C_{Mk} is the cost per unit distance for resources Mk flow, and $M_{LMi,j}$ is the distance traveled by resource Mk flow per unit time between locations i and location j and is calculated by Eq. (25):

$$M_{LMi,j} = FL_{Mki,j} * D_{ij} \quad (25)$$

where the $FL_{Mki,j}$ is the frequency of resource Mk flow between location i and j per unit time and is calculated using Eq. (24), D_{ij} is the distance between location i and j and is calculated by Eq. (25):

$$FL_{Mki,j} = \begin{bmatrix} FL_{Mk1,1} & FL_{Mk1,2} & \dots & FL_{Mk1,q} \\ FL_{Mk2,1} & FL_{Mk2,2} & \dots & FL_{Mk2,q} \\ \vdots & \vdots & \ddots & \vdots \\ FL_{Mkq,1} & FL_{Mkq,2} & \dots & FL_{Mkq,q} \end{bmatrix} \quad (26)$$

$$D_{i,j} = |XL_j - XL_i| + |YL_j - YL_i| \quad (27)$$

Table 1 Facilities and their corresponding index numbers

Index number	Facilities
1	Main gate
2	Side gate
3	Batching plant
4	Steel storage yard
5	Formwork storage yard
6	Bending yard
7	Cement and sand and aggregate storage yard
8	Curing yard
9	Refuse dumping area
10	Casting yard
11	Lifting yard

Table 2 Four types of materials and transport costs per unit distance

MK	Material	Cost per unit
1	Aggregate, sand and cement/concrete	4
2	Reinforcement bars	5
3	Formwork	8
4	Completed precast units	8.5

Table 3 Coordinates of the available locations

Location number	1	2	3	4	5	6	7	8	9	10	11
X	15	13	22	25	20	12	40	48	48	5	32
Y	40	30	30	20	10	10	10	20	35	20	42

Table 4 Distance between locations

Distance	Location	1	2	3	4	5	6	7	8	9	10	11
Location	1	0	12	17	30	35	33	55	53	38	30	19
	2	12	0	9	22	27	21	47	45	40	18	31
	3	17	9	0	13	22	30	38	36	31	27	22
	4	30	22	13	0	15	23	25	23	38	20	29
	5	35	27	22	15	0	8	20	38	53	25	44
	6	33	21	30	23	8	0	28	46	61	17	52
	7	55	47	38	25	20	28	0	18	33	45	40
	8	53	45	36	23	38	46	18	0	15	43	38
	9	38	40	31	38	53	61	33	15	0	58	23
	10	30	18	27	20	25	17	45	43	58	0	49
	11	19	31	22	29	44	52	40	38	23	49	0

Table 5 Flow frequency of the four types of materials between the facilities

	1	2	3	4	5	6	7	8	9	10	11
Facility											
1. Aggregate, sand and cement											
1							20				
2							15				
3							35			35	
4											
5											
6											
7	20	15	35								
8											
9											
10				35							
11											
2. Reinforcement											
1				30							
2				20							
3											
4	30	20				50					
5											
6				50						50	
7											
8											
9											
10						50					
11											
3. Formwork											
1											
2											
3											
4											
5											48
6											
7											
8											
9											
10						48					
11											
4. Complete precast units											
1											28
2											20
3											
4											
5											
6											
7											
8										48	48
9											
10									48		
11	28	20									48

Table 6 Parameter settings of the algorithms

CSS	Pop. size	a	c		
	50	1	eps*		
WOA	Pop. size				
	50				
VPS	Pop. size	α	w_1	w_2	p
	50	0.07	0.3	0.3	0.2
EVPS	Pop. size	α	w_1	w_2	p
	50	0.07	0.3	0.3	0.2

eps: a small positive number to avoid division by zero

where L_i and L_j are the coordinates of the locations within the site area.

4 Numerical Examples

A case study is presented by Cheung et al. (2002) that was solved by several methods such as (Liang and Chao 2008; Wong et al. 2010; Kaveh et al. 2012, 2016; Kaveh and Rasteghar Moghaddam 2017). There are 11 facilities that should be assigned to 11 predetermined locations in the yard. The facilities and their corresponding index numbers are listed in Table 1. Four types of resources and transport costs per unit distance are also presented in Table 2. Coordinates of the available locations are listed in Table 3. With these coordinates, the rectangular distance matrix D_{ij} for the locations was then calculated and presented as Table 4. Flow frequency of the four types of resources between the facilities is listed in Table 5.

5 Results and Discussion

Due to the central limit theorem, if the sample size becomes larger the distribution of the sample mean converges to the normal distribution; the sample size must be equal or more than 30. Therefore, 30 independent experimental runs through 1000 iterations are performed. Employing four optimization methods, the problem is solved by MATLAB R2014a. Since the performance of the CSS, VPS, and EVPS are dependent on the control parameters, several tests have been conducted to select the appropriate parameters for finite-time performance of these algorithms. The parameter settings of algorithms used in this paper are listed in Table 6.

The comparison results of algorithms for CSLP are listed in Table 7. The mean convergence curves of algorithms are shown in Fig. 5. Also, the comparison of best

Table 7 Comparison of algorithms for the CSLP

	Best cost	Mean cost	Std. Dev.	Worst cost
GA (Cheung et al. 2002)	99,788	N/A	N/A	N/A
MIP (Wong et al. 2010)	59,828.7	N/A	N/A	N/A
TS (Liang and Chao 2008)	94,858	N/A	N/A	N/A
HMCSS + LS (Kaveh et al. 2012)	92,758	N/A	N/A	N/A
PSO (Kaveh et al. 2016)	92,758	97,667	3363.1	106,630
CBO (Kaveh et al. 2016)	92,758	97,504	3149	103,038
ECBO (Kaveh et al. 2016)	92,758	96,670	2733.5	102,920
CSS*	92,758	98,074.5	3055	105,046
WOA*	92,758	104,189	4677.2	111,816
VPS*	92,758	97,301.9	2498.2	102,308
EVPS*	92,758	97,178.8	2736.4	103,502

N/A not available, *present work

Fig. 5 Convergence curves obtained for the CSLP problem

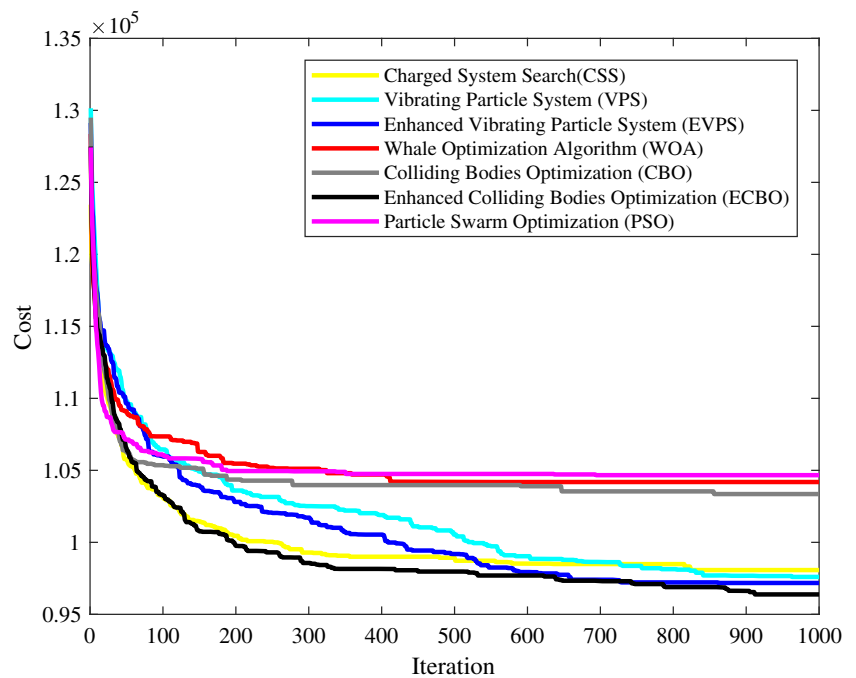
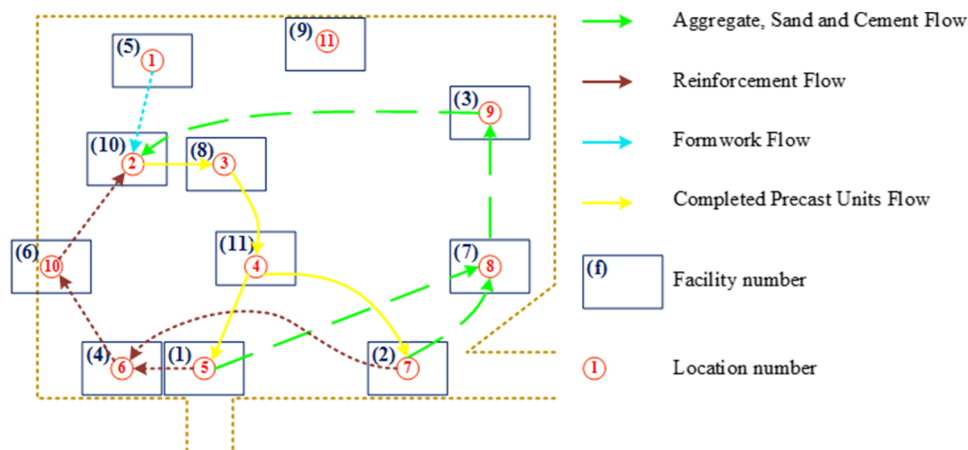


Table 8 Best layouts of this paper and previous researches

Algorithm	Best layout										
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
GA (Cheung et al. 2002)	1	10	9	6	8	5	11	3	7	4	2
MIP (Wong et al. 2010)	1	10	8	6	7	5	9	3	11	4	2
TS (Liang 2008)	5	7	10	8	1	9	6	3	11	2	4
HMCSS + LS (Kaveh et al. 2012)	5	7	9	6	1	10	8	3	11	2	4
PSO (Kaveh et al. 2016)	5	7	9	6	1	10	8	3	11	2	4
CBO (Kaveh et al. 2016)	5	7	9	6	1	10	8	3	11	2	4
ECBO (Kaveh et al. 2016)	5	7	9	6	1	10	8	3	11	2	4
CSS*	5	7	9	6	1	10	8	3	11	2	4
WOA*	5	7	9	6	1	10	8	3	11	2	4
VPS*	5	7	9	6	1	10	8	3	11	2	4
EVPS*	5	7	9	6	1	10	8	3	11	2	4

N/A not available, *present work

Fig. 6 Proposed layout arrangement plan for the site pre-cast yard



results of this paper and previous studies is listed in Table 8.

As shown in Table 7 and Fig. 5, the EVPS algorithm converges faster than other algorithms and has better efficiency in the mean (97,276) and worst (102,794) costs and the standard deviation of the VPS (2498.2) is better than other three algorithms. Totally, ECBO still performs better solutions for this problem.

As shown in Table 8, facilities 1 through 11 are closet to locations 5, 7, 9, 6, 1, 10, 8, 3, 11, 2, and 4, respectively. These results are similar to those obtained by (Kaveh et al. 2012, 2016). The proposed layout arrangement plan and flow diagram for the site pre-cast yard are depicted in Fig. 6.

6 Conclusions

In this paper, four recently developed metaheuristic algorithms consisting of CSS, WOA, VPS, and EVPS are applied for resolving a practical construction site layout planning algorithm. The performances of these metaheuristic algorithms are compared in terms of their effectiveness. Results show that all of these metaheuristics are able to reach the best cost. The VPS, and EVPS algorithms present more reliable solutions than others. Future researches can focus on integrating this model with tower crane layout planning problem for presenting a model that considers three dimensions of material handling in the construction site.

References

Abdelmegid MA, Shawki KM, Abdel-Khalek H (2015) GA optimization model for solving tower crane location problem in construction sites. *Alex Eng J* 54:519–526. <https://doi.org/10.1016/j.aej.2015.05.011>

- Alkriz K, Mangin J-C (2005) A new model for optimizing the location and construction facilities using genetic algorithms. In: *Proceedings of the 21st annual conference. Association of researchers in construction management, ARCOM*, pp 981–991
- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *IEEE congress on evolutionary computation. CEC 2007*, pp 4661–4667. <https://doi.org/10.1109/cec.2007.4425083>
- Balakrishnan J, Cheng CH (2007) Multi-period planning and uncertainty issues in cellular manufacturing: a review and future directions. *Eur J Oper Res* 177:281–309. <https://doi.org/10.1016/j.ejor.2005.08.027>
- Cheung SO, Tong TKL, Tam CM (2002) Site pre-cast yard layout arrangement through genetic algorithms. *Autom Constr* 11:35–46. [https://doi.org/10.1016/S0926-5805\(01\)00044-9](https://doi.org/10.1016/S0926-5805(01)00044-9)
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science. Nagoya, Japan*, pp 39–43. <https://doi.org/10.1109/mhs.1995.494215>
- Erol OK, Eksin I (2006) A new optimization method: big bang-big crunch. *Adv Eng Softw* 37:106–111. <https://doi.org/10.1016/j.advengsoft.2005.04.005>
- Gharaie E, Afshar A, Jalali MR (2006) Site layout optimization with ACO algorithm. In: *Proceedings of the 5th WSEAS international conference on artificial intelligence, knowledge engineering and data bases. World Scientific and Engineering Academy and Society (WSEAS)*, pp 90–94
- Golberg DE (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, Reading
- Kaveh A (2017) *Advances in metaheuristic algorithms for optimal design of structures*, 2nd edn. Springer, Berlin
- Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Adv Eng Softw* 110:69–84. <https://doi.org/10.1016/j.advengsoft.2017.03.014>
- Kaveh A, Ilchi Ghazaan M (2017) A new meta-heuristic algorithm: vibrating particles system. *Sci Iran* 24:1–32. <https://doi.org/10.24200/sci.2017.2417>
- Kaveh A, Khayatad (2013) Ray optimization for size and shape optimization of truss structures. *Comput Struct* 117:82–94. <https://doi.org/10.1016/j.compstruc.2012.12.010>
- Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. *Comput Struct* 139:18–27. <https://doi.org/10.1016/j.compstruc.2014.04.005>

- Kaveh A, Rasteghar Moghaddam M (2017) A hybrid WOA-CBO algorithm for construction site layout planning problem. *Sci Iran* 25(3):1094–1104. <https://doi.org/10.24200/sci.2017.4212>
- Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213:267–289. <https://doi.org/10.1007/s00707-009-0270-4>
- Kaveh A, Vazirinia Y (2017) Tower cranes and supply points locating problem using CBO, ECBO, and VPS. *Int J Optim Civ Eng* 7(3):393–411
- Kaveh A, Vazirinia Y (2018) Optimization of tower crane location and material quantity between supply and demand points: a comparative study. *Period Polytech Civ Eng* 62(3):732–745. <https://doi.org/10.3311/PPci.11816>
- Kaveh A, Zolghadr A (2014) Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints. *Adv Eng Softw* 76:9–30. <https://doi.org/10.1016/j.advengsoft.2014.05.012>
- Kaveh A, Shakouri MAA, Moghaddam SZ (2014) An adapted harmony search based algorithm for facility layout optimization. *Int J Civil Eng* 10(1):37–42
- Kaveh A, Khanzadi M, Alipour M, Rasteghar Moghaddam M (2016) Construction site layout planning problem using two new meta-heuristic algorithms. *Iran J Sci Technol Trans Civ Eng* 40:263–275. <https://doi.org/10.1007/s40996-016-0041-0>
- Kaveh A, Hosseini Vaez SR, Hosseini P (2017) Enhanced vibrating particles system algorithm for damage identification of truss structures. *Sci Iran*. <https://doi.org/10.24200/sci.2017.4265>
- Koopmans TC, Beckman M (1957) Assignment problems and the location of economic activities. *Econometrica* 25:53–76
- Li H, Love PED (1998) Site-level facilities layout using genetic algorithms. *J Comput Civ Eng* 12:227–231. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1998\)12:4\(227\)](https://doi.org/10.1061/(ASCE)0887-3801(1998)12:4(227))
- Liang YL, Chao WC (2008) The strategies of tabu search technique for facility layout optimization. *Autom Constr* 17:657–669. <https://doi.org/10.1016/j.autcon.2008.01.001>
- MATLAB Software (2014) url: <http://www.mathworks.com/>
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Tompkins JA, White JA, Bozer YA, Tanchoco JMA (2010) *Facilities planning*, 4th edn. JWiley, Hoboken
- Wang J, Zhang X, Shou W, Wang X, Xu B, Kim MJ, Wu P (2015) A BIM-based approach for automated tower crane layout planning. *Autom Constr* 59:168–178. <https://doi.org/10.1016/j.autcon.2015.05.006>
- Wong CK, Fung IWH, Tam CM (2010) Comparison of using mixed-integer programming and genetic algorithms for construction site facility layout planning. *J Constr Eng Manag* 136:1116–1128. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000214](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000214)