

Optimization of Tower Crane Location and Material Quantity Between Supply and Demand Points: A Comparative Study

Ali Kaveh^{1*}, Yasin Vazirinia¹

RESEARCH ARTICLE

Received 30 July 2017; Revised 13 February 2017; Accepted 22 February 2018

Abstract

Location optimization of tower crane as an expensive equipment in the construction projects has an important effect on material transportation costs. Due to the construction site conditions, there are several tower crane location optimization models. Appropriate location of tower cranes for material supply and engineering demands is a combinatorial optimization problem within the tower crane layout problem that is difficult to resolve. Meta-heuristics are popular and useful techniques to resolve complex optimization problems. In this paper, the performance of the Particle Swarm Optimization (PSO) and four newly developed meta-heuristic algorithms Colliding Bodies Optimization (CBO), Enhanced Colliding Bodies Optimization (ECBO), Vibrating Particles System (VPS), and Enhanced Vibrating Particles System (EVPS) are compared in terms of their effectiveness in resolving a practical Tower Crane Layout (TCL) problem. Results show that ECBO performs better than other three methods in both cases.

Keywords

Tower Crane Layout Problem, Enhanced Colliding Bodies optimization, Vibrating Particles System, Construction Site Layout Planning, Meta-heuristic Algorithm

1 Introduction

In recent decades, many researchers have tried to provide the best method for solving the Construction Engineering Optimization Problems (CEOPs). Construction site layout problems (CSLPs) are the most interesting CEOPs, because they brought the consideration of layout esthetics and usability qualities into the facility design process [1]. Many building construction projects utilize tower cranes for transporting heavy construction materials. Material transportation is one of the major activities in the building construction industry, lifting and hoisting heavy materials by cranes in construction sites are usual tasks that need precise planning [2]. Every construction project requires enough space for temporary facilities to perform 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 accounting for a wide range of scenarios. CSLPs are known as combinatorial optimization problems. There are two methods to solve large size problems, the meta-heuristics and the exact methods with the global search for smaller search sized problems [3]. For example, Li and Love [4] 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. Gharaie et al. [5] resolved their model by Ant Colony Optimization. Kaveh et al. [6] applied Colliding Bodies Optimization (CBO) and its enhanced version (ECBO).

The tower crane is a major facility in the transportation of materials, especially heavy prefabricated units such as steel beams, ready-mixed concrete, prefabricated elements, and large-panel formworks [7]. TCLP tries to find the best position of tower cranes and supply points in a building construction site for supplying all the requests in a minimum time, has been raised from about twenty years ago. Zhang et al. [8] developed an analytical model by considering the travel time of

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

* Corresponding author, email: alikaveh@iust.ac.ir

tower crane hooks and adopting a Monte Carlo simulation to optimize the tower crane location. However, considered tower crane in their study was a single crane and the effect of location of supply points on lifting requirements and travel time has been neglected. An artificial neural network model by Tam et al. [7] applied for predicting tower crane operations and also they employed a genetic algorithm model to optimize the crane and supply points layout [7, 9]. The case study used by Tam et al. [9] to show the effectiveness of their model was subsequently used in several papers to compare the effectiveness of other optimization methods. For example, mixed integer linear programming (MILP) used by Huang et al. [3] to optimize the crane and supply points location. Result showed that their method reduced the travel time of the hook by 7% compared to the results obtained from the previous genetic algorithms. Kaveh and Vazirinia [10] compared the performance of CBO, ECBO, and VPS in this model and discussed the results. A particle bee algorithm (PBA) with two cases developed by Lien and Cheng [11] to optimize the tower crane layout and material quantity between supply and demand points. Herdani [12] developed an Evolutionary Big-Bang Big-Crunch EBB-BC and applied it for optimizing this model. In this paper, four newly developed meta-heuristic algorithms called Colliding Bodies Optimization (CBO) [13], Enhanced Colliding Bodies Optimization (ECBO) [14], Vibrating Particles System (VPS) [15], and Enhanced Vibrating Particles System (EVPS) [16] are used to optimize the tower crane layout and material quantity between supply and demand points.

Solving real-life problems by meta-heuristic algorithms has become an interesting topic in recent years. Many meta-heuristics with different philosophy and characteristics are developed and applied to a wide range of fields. The main 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 derivatives of the objective function, they have received increasing attention from both academia and industry [23]. Meta-heuristic methods are global optimization methods that try to reproduce natural phenomena (Genetic Algorithm [17], Particle Swarm Optimization [18], Water Evaporation Optimization [19]), humans social behavior (Imperialist Competitive Algorithm [20]), or physical phenomena (Charged System Search (CSS) [21], Colliding Bodies Optimization [13], Big Bang-Big Crunch [22], Vibrating Particles System (VPS) [15]). Exploitation and exploration are two important characteristics of meta-heuristic optimization methods. Exploitation serves to search around the current best solutions and select the best possible points, and exploration allows the optimizer to explore the search space more efficiently, often by randomization [23].

After this introduction, tower crane layout with material quantity supply and demand optimization described in section 2, a brief explanation of optimization algorithms presented in

section 3. Numerical examples are studied in Section 4 and the results are discussed in Section 5. Conclusions are derived in Section 6.

2 Problem: tower crane layout with material quantity supply and demand optimization [11]

Many researches have worked on the locating and transporting time of a tower crane. A mathematical model for determining the most suitable tower crane location was developed by Choi and Harris [24]; Zhang et al. [8] developed the Monte Carlo simulation approach to optimize tower crane location; Tam et al. [9] employed an artificial neural network model for predicting tower crane operations and a genetic algorithm model for site facility layout [7, 9]. Huang et al. [3] developed a mixed integer linear programming (MILP) to optimize the crane and supply locations. However, Tam and Huang considered only operation time cost of material operation flow and ignored other important cost factors such as rent, labor and tower crane setup [11]. Lien and Cheng [11] developed a Particle Bee Algorithm (PBA) to solve a TCL model that more practically reflect the actual conditions on a construction site and also considered the rent, labor, and tower crane setup cost. Herdani [12] applied PSO, BB-BC, and EBB-BC meta-heuristic algorithms to solve this PBA model and discussed the results. In this paper, the performance of PSO and four newly developed meta-heuristic algorithms CBO, ECBO, VPS, and EVPS are compared in terms of their effectiveness in resolving a practical TCLP. The meta-heuristics are used to optimize the location of the tower crane. Also, these methods used to optimize the operating distance and frequency between demand and supply points in terms of total operating costs based on the material requirements at demand and supply points.

Travel distance between the supply and demand points can be calculated by the Eqs. (1)–(3) referring to Figs. 1 and 2.

$$\rho(D_j^k) = \sqrt{(D_j^x - Cr_k^x)^2 + (D_j^y - Cr_k^y)^2} \quad (1)$$

$$\rho(S_i^k) = \sqrt{(S_i^x - Cr_k^x)^2 + (S_i^y - Cr_k^y)^2} \quad (2)$$

$$L = \sqrt{(S_i^x - D_j^x)^2 + (S_i^y - D_j^y)^2} \quad (3)$$

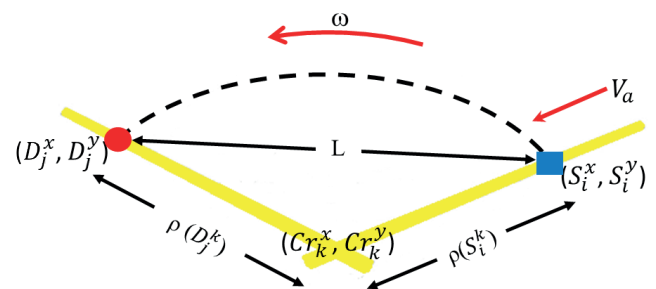


Fig. 1 Radial and tangential movements of the hook

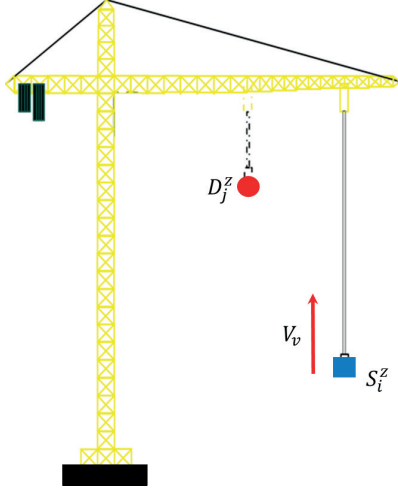


Fig. 2 Vertical movement of the hook

Hook movement time is an important parameter to evaluate the total time of material transportation using a tower crane. The hook movement time has split up into horizontal and vertical paths to reflect the operating costs by giving an appropriate cost-time factor. Corresponding movement paths along different directions can be seen from Figs. 1 and 2.

A continuous type parameter ψ indicates the degree of coordination of the hook movement in radial and tangential directions which depends on the control skills of a tower crane operator, times for horizontal and vertical hook movements can be calculated from Eqs. (4) to (7), respectively.

$$T_a = \frac{|\rho(D_j^k) - \rho(S_i^k)|}{V_a} \quad (4)$$

$$T_\omega = \frac{1}{V_\omega} \arccos \left[\frac{L^2 + \rho(D_j^k)^2 + \rho(S_i^k)^2}{2 * \rho(D_j^k) * \rho(S_i^k)} \right], \quad (5)$$

$$[0 < \arccos(\theta) < \pi]$$

$$T_h = \max\{T_a + T_\omega\} + \Psi * \min\{T_a + T_\omega\} \quad (6)$$

$$T_v = \frac{|D_j^z - S_i^z|}{V_h} \quad (7)$$

The total travel time of tower crane at location k between supply point i and demand point j , $T_{i,j}^k$, can be calculated using Eq. (8) by specifying the continuous type parameter β for the degree of coordination of hook movement in horizontal and vertical planes.

$$T_{i,j}^k = \max\{T_h + T_v\} + \beta * \min\{T_h + T_v\} \quad (8)$$

The objective function of the TCLP was required to satisfy two requirements: (1) The function must be high only for those solutions with a high design preference and (2) The function must be high only for those solutions that satisfy the layout constraints [11]. The objective function of this problem presents as follow:

$$\min TC = \sum_k^K \sum_i^I \sum_j^J T_{i,j}^k * Q_{i,j} * CU_k + R + S + L \quad (9)$$

$$R = \sum_k^K M_k * (\text{int}(DY_k / 30)) \quad (10)$$

$$S = \sum_k^K IS_k + MS_k * MST_k + RSI_k \quad (11)$$

$$L = \sum_k^K LC_k * LA_k * DY_k \quad (12)$$

where TC is total cost; K is the number of potential crane; I is the number of supply points; J is the number of demand points; $Q_{i,j}$ is quantity of material flow from S_i to D_j ; CU_k is cost of material flow from S_i to D_j per unit quantity and unit time by k th crane; R is total rent cost; S is tower crane total setup cost; L is total labor cost; M_k is rent cost per month by k th crane; DY_k is days of renting tower crane/labor work by k th crane; IS_k is tower crane initial setup cost; MS_k is tower crane modified setup cost by k th crane; MST_k is modified setup times by k th crane; RSI_k is disassembly cost; LC_k is labor cost per person by k th crane; LA_k is labor amount by k th crane.

Subject to:

$$\begin{aligned} &\text{If actual supply capacities}(i) > \\ &\text{limit supply capacities}(i) \text{ then } TC = TC + 40,000 \end{aligned} \quad (13)$$

$$\begin{aligned} &\text{If actual supply capacities}(i) < > \\ &\text{limit demand capacities}(i) \text{ then } TC = TC + 40,000 \end{aligned} \quad (14)$$

Notice: the objective function limit of actual supply capacities should be smaller or equal to limit supply capacities. Besides, the objective function limit of actual demand capacities should be equal to limit demand capacities. The subject will give a penalty when objective function breaks the above rules.

3 Optimization algorithms

Particle Swarm Optimization (PSO) is a well-known algorithm and presented in many papers and books, and for brevity it is not repeated in here. However for a through explanation of this method and its enhanced version, the interested reader may refer to Kaveh [23].

3.1 Colliding bodies optimization

An efficient algorithm, inspired from the momentum, and energy rules of the physics, named Colliding Bodies Optimization, has been developed by Kaveh and Mahdavi [13]. CBO does not depend on any internal parameter and also it is extremely simple to implement and to use. In this method, one body collides with another body and they move to the lower cost. Each solution candidate "X" at CBO, contains a number of variables (i.e., $X_i = \{X_{i,j}\}$) and is considered as a colliding body (CB). The bodies with masses being assigned are divided to two main equal

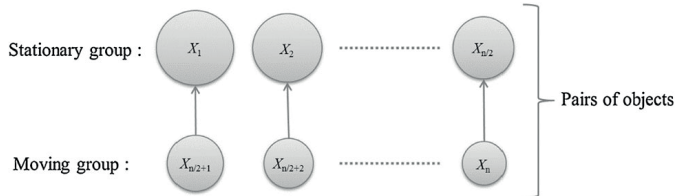


Fig. 3 Pairs of CBs for collision

groups; i.e., stationary and moving bodies (Fig. 3), where the moving bodies move to stationary bodies and a collision occurs between the pairs of bodies. The goal of this process is: (i) to improve the locations of moving bodies and (ii) to push stationary bodies toward the better locations. After the collision, new locations of colliding bodies are updated based on the new velocity by using the collision rules. The main procedure of the CBO is described as:

Step 1: The initial positions of colliding bodies are determined with random initialization of a population of individuals in the search space:

$$x_i^o = x_{\min} + rand \times (x_{\max} - x_{\min}), i = 1, 2, 3, \dots, n \quad (15)$$

where, x_i^o determines the initial value vector of the i th colliding body. x_{\max} and x_{\min} are the minimum and the maximum allowable values vectors of variables, respectively; $rand$ is a random number in the interval $[0, 1]$; and n is the number of colliding bodies.

Step 2: The magnitude of the body mass for each colliding body is defined as:

$$m_k = \left(\frac{1}{fit(i)} \right) / \left(1 / \sum_{i=1}^n \left(\frac{1}{fit(i)} \right) \right) \quad (16)$$

where $fit(i)$ represents the objective function value of the colliding body i ; n is the population size. It seems that a colliding body with good values exerts a larger mass than the bad ones. Also, for maximization, the objective function $fit(i)$ will be replaced by $1/fit(i)$.

Step 3: Then colliding bodies objective function values are arranged in an ascending order. The sorted colliding bodies are divided into two equal groups:

The lower half of the CBs (stationary CBs): These CBs are good agents which are stationary and the velocity of these bodies before the collision is zero. Thus:

$$v_i = 0, i = 1, 2, 3, \dots, \frac{n}{2} \quad (17)$$

The upper half of CBs (moving CBs): These CBs move toward the lower half. Then, according to Fig. 3, the better and worse CBs, i.e. agents with upper fitness value, of each group will collide together. The change of the body position represents the velocity of these bodies before collision as:

$$v_i = x_i - x_{i-\frac{n}{2}}, i = \frac{n}{2} + 1, \dots, n \quad (18)$$

where, v_i and x_i are the velocity and position vector of the i th CB in this group, respectively; $x_{i-\frac{n}{2}}$ is the i th CB pair position of x_i in the previous group.

Step 4: After the collision, the velocities of the CBs in each group are evaluated as:

Stationary CBs:

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} - \varepsilon m_{i+\frac{n}{2}} \right) \times v_{i+\frac{n}{2}}}{\left(m_i + m_{i+\frac{n}{2}} \right)}, i = 1, 2, 3, \dots, \frac{n}{2}; \quad (19)$$

Moving CBs:

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}} \right) \times v_i}{\left(m_i + m_{i-\frac{n}{2}} \right)}, i = \frac{n}{2} + 1, \frac{n}{2} + 2, \frac{n}{2} + 3, \dots, n; \quad (20)$$

where ε is the Coefficient Of Restitution (COR) of the two colliding bodies, defined as:

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \quad (21)$$

with $iter$ and $iter_{\max}$ being the current iteration number and the total number of iteration for optimization process, respectively.

New positions of CBs are updated using the generated velocities after the collision in position of stationary CBs, as follow for each group:

Moving CB:

$$x_i^{new} = x_{i-\frac{n}{2}} + rand^o v_i', i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n; \quad (22)$$

where, x_i^{new} and v_i' are the new position and the velocity after the collision of the i th moving CB, respectively; $x_{i-\frac{n}{2}}$ is the old position of the i th stationary CB pair.

Stationary CB:

$$x_i^{new} = x_i + rand^o v_i', i = 1, 2, 3, \dots, \frac{n}{2}; \quad (23)$$

where, x_i^{new} and v_i' are the new positions, previous positions and the velocity after the collision of the i th CB, respectively. $rand$ is a random vector uniformly distributed in the range of $[-1, 1]$ and the sign "o" denotes an element-by-element multiplication.

Step 6: The process is repeated from step 2 until one termination criterion is satisfied. Termination criterion is the pre-defined maximum number of iterations. After getting the near-global optimal solution, it is recorded to generate the output.

The pseudo of CBO is shown in Fig. 4.

3.2 Enhanced colliding bodies optimization

In order to improve CBO for faster and more reliable solutions, Enhanced Colliding Bodies Optimization (ECBO) was developed which uses memory to save a number of historically best CBs and also utilizes a mechanism to escape from local optima [14]. The pseudo of ECBO is shown in Fig. 5 and the steps involved are given as follows:

Step 1: Initialization

Initial positions of all CBs are determined randomly in an m -dimensional search space by Eq. (15).

Step 2: Defining mass

The value of mass for each CB is evaluated according to Eq. (16).

Step 3: Saving

Considering a memory which saves some historically best CB vectors and their related mass and objective function values can improve the algorithm performance without increasing the computational cost [21]. For that purpose, a Colliding Memory (CM) is utilized to save a number of the best-so-far solutions. Therefore in this step, the solution vectors saved in CM are added to the population, and the same numbers of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups

CBs are divided into two equal groups: (i) stationary group and (ii) moving group. The pairs of CBs are defined according to Fig. 3.

Step 5: Criteria before the collision

The velocity of stationary bodies before the collision is zero (Eq. (17)). Moving objects move toward stationary objects and their velocities before collision are calculated by Eq. (18).

Step 6: Criteria after the collision

The velocities of stationary and moving bodies are calculated using Eqs. (19) and (20), respectively.

Step 7: Updating CBs

The new position of each CB is calculated by Eqs. (22) and (23).

Step 8: Escape from local optima

Meta-heuristic algorithms should have the ability to escape from the trap when agents get close to a local optimum. In ECBO, a parameter like *Pro* within (0, 1) is introduced and it is specified whether a component of each CB must be changed or not. For each colliding body, *Pro* is compared with rn_i ($i = 1, 2, \dots, n$) which is a random number uniformly distributed within (0, 1). If $rn_i < Pro$, one dimension of the i th CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + rand \times (x_{j,\max} - x_{j,\min}), i = 1, 2, \dots, n; \quad (24)$$

where x_{ij} is the j th variable of the i th CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the j th variable, respectively. In order to protect the structures of CBs, only one dimension is changed. This mechanism provides opportunities for the CBs to move all over the search space thus providing better diversity.

Step 9: Terminating condition check

The optimization process is terminated after a fixed number of iterations. If this criterion is not satisfied go to Step 2 for a new round of iteration.

3.3 Vibrating particles system

The VPS is a population-based algorithm which simulates a free vibration of single degree of freedom systems with viscous damping [15]. 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

Pseudo Code of the Colliding Bodies Optimization

```
Initial location is created randomly by Eq. (15)
The value of the objective function is evaluated and masses are defined by Eq. (16)
While stop criteria is not attained (like max iteration)
  for each CBs
    Calculating the velocity of Stationary and moving CBs before collision according Eqs. (17) and (18)
    Calculate CBs velocity after collision according by Eqs. (19) and (20)
    Update CBs position according Eqs. (22) and (23)
  End for
End while
End
```

Fig. 4 Pseudo code of the CBO

Pseudo Code of Enhanced Colliding Bodies Optimization

```
Initial location is created randomly by Eq. (15)
The value of objective function is evaluated and masses are defined by Eq. (16)
While stop criteria is not attained (like max iteration)
  for each CBs
    Calculate Stationary and moving CBs velocity before collision according Eqs. (17) and (18)
    Calculate CBs velocity after collision according by Eqs. (19) and (20)
    Update CBs position according Eqs. (22) and (23)
    If rand  $i < Pro$ 
      One dimension of the  $i$ th CB is selected randomly and regenerate by Eq. (24)
    End if
  End for
End while
End
```

Fig. 5 Pseudo code of the ECBO [14]

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: (i) the historically best position of the entire population (HB), (ii) a good particle (GP), and (iii) 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. (25):

$$x_i^j = x_{\min} + rand \times (x_{\max} - x_{\min}), \quad i = 1, 2, 3, \dots, n; \quad (25)$$

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.

$$\begin{aligned} x_i^j = & \omega_1 \cdot [D.A.rand1 + HB^j] \\ & + \omega_2 \cdot [D.A.rand2 + GP^j] \\ & + \omega_3 \cdot [D.A.rand3 + BP^j] \end{aligned} \quad (26)$$

According to the above concepts, the particles position are updated by follow equation:

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 $[0, 1]$. The parameter A is defined as:

$$\begin{aligned} A = & \left[\omega_1 \cdot (HB^j - x_i^j) \right] \\ & + \left[\omega_1 \cdot (GP^j - x_i^j) \right] \\ & + \left[\omega_1 \cdot (BP^j - x_i^j) \right] \end{aligned} \quad (27)$$

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

$$D = \left(\frac{iter}{iter_{\max}} \right)^{-\alpha} \quad (28)$$

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 $p < rand$, then $\omega = 0$ and $\omega_2 = 1 - \omega_3$.

Three essential concepts consisting of self-adaptation, cooperation, and competition are considered in this algorithm. Particles move towards HB so the self-adaptation is provided. Any particle has the chance to have 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 [21]. 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.

Pseudo code of Vibrating Particles System (VPS)

```

Initialize algorithm parameters
Create initial positions randomly by Eq. (25)
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
      W3=0 and w2=1-w1
    end if
    for each component
      New location is obtained by Eq. (26)
    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

```

Fig. 6 Pseudo code of the vibrating particles system algorithm [15]

| Pseudo code of Enhanced Vibrating Particles System (EVPS) | |
|----------------------------------------------------------------------------------------------------------------------|--|
| Initialize algorithm parameters | |
| Create initial positions randomly by Eq. (25) | |
| 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. (29)-(a), (b), or (c) with probability of ω ₁ , ω ₂ and ω 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 | |

Fig. 7 Pseudo code of the enhanced vibrating particles system algorithm [16]

| | | | | | | | | | | | | | |
|----------------|-----------------------------------------------------------------|------------------|-----|--------------------|------------------|-----|------------------|-----|------------------|-----|--------------------|------------------|----------|
| Cell number | 1 | 2 | ... | I-1 | I | ... | 2I | ... | I×(j-1)+i | ... | I×J-1 | I×J | I×J+1 |
| Representation | Q _{1,1} | Q _{2,1} | ... | Q _{I-1,1} | Q _{I,1} | ... | Q _{I,2} | ... | Q _{i,j} | ... | Q _{I,J-1} | Q _{I,J} | ϕ |
| | The quantity of materials between supply and demand pairs (i-j) | | | | | | | | | | | | Location |

ϕ: An integer number in range of [1,K] represents the crane location number

Fig. 8 Definition of the solution candidates for each crane

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 pseudo code of the VPS is shown in Fig. 6.

3.4 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. (26) and (27) should be replaced with Eqs. (29) and (30). In Eqs. (29) and (30), one of (a), (b) and (c) equations are applied with the probability of , ω₁, ω₂ and ω₃, respectively [16].

$$x_i^j = \begin{cases} \left[D.A.rand1 + OHB^j \right] & \text{(a)} \\ \left[D.A.rand2 + GP^j \right] & \text{(b)} \\ \left[D.A.rand3 + BP^j \right] & \text{(c)} \end{cases} \quad (29)$$

$$A = \begin{cases} (\pm 1)(OHB^j - x_i^j) & \text{(a)} \\ (\pm 1)(GP^j - x_i^j) & \text{(b)} \\ (\pm 1)(BP^j - x_i^j) & \text{(c)} \end{cases} \quad (30)$$

where (±1) 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 [16].

Pseudo code of the EVPS algorithm is illustrated in Fig. 7.

3.5 Encoding of solutions

There is a population of individuals in each optimization algorithm. Every individual consists of an encoding of a candidate solution (colliding body in CBO, and vibrating particle in VPS) and a fitness that indicates its quality. Selecting an appropriate encoding method is an important step in applying meta-heuristics to an optimization problem. There are a number of encoding schemes such as binary encoding [25], permutation encoding [26], real value encoding [27], and tree encoding [28] for combinatorial optimization problems. In this paper, a hybrid encoding method which involves two string of number is used (Fig. 8); the first one is a vector of real number strings containing the quantity of material between supply and demand pairs whereas the second is a vector of integer strings contain the code of tower cranes location. The location and material quantity of each crane are modeled as a string like Fig. 8. For instance, solution candidates of the best particle for each numerical example shown in relevant sections.

4 Numerical examples

A case study presented by Tam et al. that used to evaluate several models such as: [3, 7, 10, 12, 29]. Numerical examples consist of modeling a single tower crane layout and modeling a

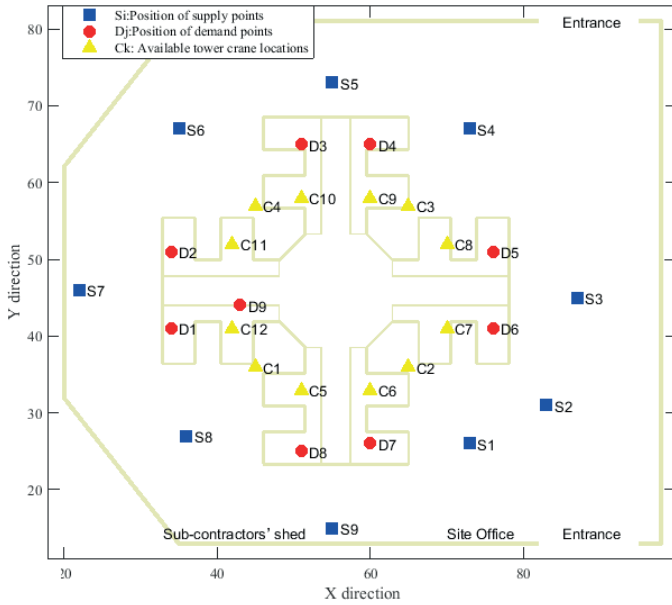


Fig. 9 Plan view of the tower crane layout case [11]

multi-tower crane layout considering 9 available material supply locations and 9 demand locations in a site which also provides 12 potential locations to set up and operate one and two tower cranes. Coordinates of the potential locations are listed in Table 1 and plan view of the TCL case is shown in Fig. 9.

Table 1 Coordinates of the potential locations [11]

| # | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|------------------------|---|----|----|-----|-----|-----|----|----|----|----|----|----|----|
| Demand point j | X | 34 | 34 | 51 | 60 | 76 | 76 | 60 | 51 | 43 | | | |
| | Y | 41 | 51 | 65 | 65 | 51 | 41 | 26 | 25 | 44 | | | |
| | Z | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | | | |
| Supply point i | X | 73 | 83 | 87 | 73 | 55 | 35 | 22 | 36 | 55 | | | |
| | Y | 26 | 31 | 45 | 67 | 73 | 67 | 46 | 27 | 15 | | | |
| | Z | 2 | 2 | 1.5 | 1.5 | 1.5 | 0 | 0 | 1 | 1 | | | |
| Tower crane position k | X | 45 | 65 | 65 | 45 | 51 | 60 | 70 | 70 | 60 | 51 | 42 | 42 |
| | Y | 36 | 36 | 57 | 57 | 33 | 33 | 41 | 52 | 58 | 58 | 52 | 41 |
| | Z | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 |

Table 2 Limit Supply Capacities for single tower crane layout [11]

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--|------|------|------|------|------|------|------|------|------|
| | 1500 | 1000 | 1500 | 1000 | 1500 | 1000 | 1500 | 1000 | 1500 |

Table 3 Limit Demand Capacities for single tower crane layout [11]

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 |

Table 4 Single TCLP Parameter Value [11]

| LA_i | LC_i | RSI_i | MST_i | MS_i | IS_i | DY_i | M_i | V_ω | V_α | β | V_h | α | CU_i |
|--------|--------|---------|---------|--------|--------|--------|-------|------------|------------|---------|-------|----------|--------|
| 5 | 100 | 2000 | 10 | 500 | 5000 | 80 | 1000 | 7.57 | 53.3 | 0.25 | 60 | 1 | 1.92 |

Table 5 Limit Supply Capacities for multi-tower crane layout [11]

| Crane # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Cr # 1 | 750 | 500 | 750 | 500 | 750 | 500 | 750 | 500 | 750 |
| Cr # 2 | 750 | 500 | 750 | 500 | 750 | 500 | 750 | 500 | 750 |

4.1 Example 4.1: Single tower crane layout

In this case study, a single tower crane was used where the results of the actual supply capacities should be smaller or equal to limit supply capacities shown in Table 2. Besides, the result of the actual demand capacities should be equal to the limit demand capacities shown in Table 3. The parameter values of single tower crane layout problem are presented in Table 4.

4.2 Example 4.2: Multi-tower crane layout

In this case study, two tower cranes are utilized where the results of the actual supply capacities should be smaller or equal to the limit supply capacities shown in Table 5. Besides, the result of the actual demand capacities should be equal to the limit demand capacities shown in Table 6. The parameter values of multi-tower crane layout problem are presented in Table 7.

5 Results and discussion

According to the central limit theorem, if the sample size gets 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 are performed for each problem and algorithm through 5000 iterations for single tower crane layout problem and 10000 iterations for

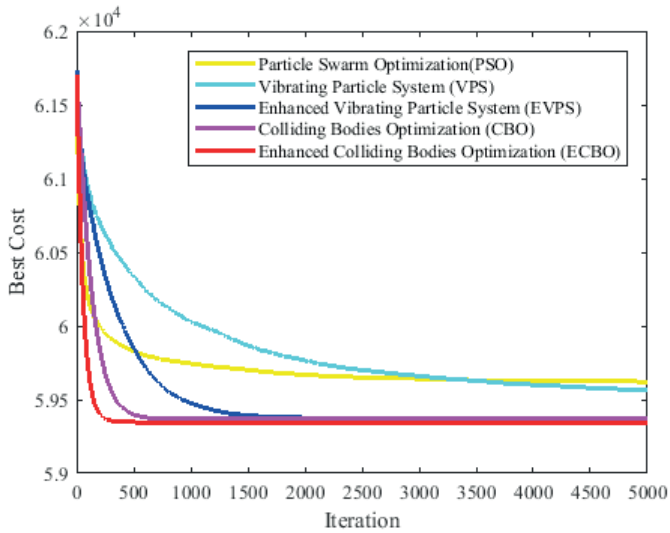


Fig. 10 Average cost convergence curves of single tower crane layout problem

multi-tower crane layout problem. Employing four optimization methods, the problem is solved by MATLAB R2014a [30]. Since the performance of the ECBO, VPS, PSO, 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 for both single tower crane layout and multi-tower crane layout are listed in Table 8.

5.1 Results and discussion for single tower crane layout

As can be seen from Table 9 and Fig. 10, the best mean costs are 59346.6, 59371.5, 59373.8, 59563.6, and 59622 for ECBO, CBO, EVPS, VPS, and PSO, respectively. These values are better than those obtained using either PSO (59828.7), BB-BC (59842.6), or EBB-BC (59842.6). It shows the robustness of CBO, ECBO, and EVPS. Also, the best cost of ECBO (59307.8) is better than other used algorithms. Thus, ECBO obtained a better evolution result than either PSO, BB-BC, EBB-BC, VPS, EVPS, or CBO. Moreover, the performance of both of ECBO, and EVPS are better than their standard versions (CBO, and VPS).

Table 6 Limit Demand Capacities for multi-tower crane layout [11]

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 |

Table 7 Multi-TCLP Parameter Value [11]

| Crane # | LA_i | LC_i | RSI_i | MST_i | MS_i | IS_i | DY_i | M_i | V_ω | V_α | β | V_h | α | CU_i |
|---------|--------|--------|---------|---------|--------|--------|--------|-------|------------|------------|---------|-------|----------|--------|
| Cr # 1 | 5 | 100 | 2000 | 10 | 500 | 5000 | 80 | 1000 | 7.57 | 53.3 | 0.25 | 60 | 1 | 1.92 |
| Cr # 2 | 5 | 100 | 2000 | 10 | 500 | 5000 | 80 | 1000 | 2.8 | 33.1 | 0.25 | 35 | 1 | 1.92 |

Table 8 Parameter settings of the algorithms

| | | | | | |
|------|-----------|----------------|---------|-----|-----------|
| CBO | Pop. size | | | | |
| | 100 | | | | |
| ECBO | Pop. size | CM size | pro | | |
| | 100 | 5-10 | 0.1-0.2 | | |
| | Pop. size | α | w1 | w2 | p |
| VPS | 100 | 0.05-0.1 | 0.3 | 0.3 | 0.15-0.25 |
| | Pop. size | α | w1 | w2 | p |
| EVPS | 100 | 0.05-0.1 | 0.3 | 0.3 | 0.15-0.25 |
| | Pop. size | inertia weight | c1 | c2 | |
| PSO | 100 | 0.4-0.9 | 2 | 2 | |

Table 9 The Comparison Result of Algorithms for single TCLP.

| | Previous Works | | | Present Work | | | | |
|------------|----------------|------------|-------------|--------------|---------|---------|---------|---------|
| | PSO [12] | BB-BC [12] | EBB-BC [12] | PSO | VPS | EVPS | CBO | ECBO |
| Best cost | 59536.1 | 59555.8 | 59460.6 | 59474.4 | 59519.5 | 59310.8 | 59307.8 | 59307.8 |
| Mean cost | 59828.7 | 59842.6 | 59463.8 | 59622 | 59563.6 | 59373.8 | 59371.5 | 59346.6 |
| St. Dev. | 106.6 | 134.1 | 7.5 | 64.4 | 18 | 12.8 | 41.7 | 31 |
| Worst cost | 60670.2 | 60278.7 | 59469.8 | 59746.3 | 59594 | 59379.9 | 59542.9 | 59385.4 |

| | | | | | | | | | | | | | | | | | |
|----------------|----|----|----|-----|-----|-----|----|-----|----|----|----|-----|-----|-----|----|----|----|
| Cell number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Representation | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 800 | 0 | 0 | 0 | 0 | 0 | 800 | 0 | 0 | 0 |
| Cell number | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| Representation | 0 | 0 | 0 | 0 | 300 | 400 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 600 | 0 | 0 | 0 |
| Cell number | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 |
| Representation | 0 | 0 | 0 | 500 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 600 | 0 | 0 | 0 | 0 | 0 |
| Cell number | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 |
| Representation | 0 | 0 | 0 | 700 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cell number | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | | | |
| Representation | 0 | 0 | 0 | 800 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 700 | 2 | | | |

Fig. 11 Definition of the solution candidates for single crane case

Table 10 Best results of ECBO for single tower crane layout problem

| | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | Actual supply | Limit supply | Supply degree |
|-------------------|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|--------------|---------------|
| C2 | S1 | 0 | 0 | 0 | 0 | 0 | 600 | 700 | 0 | 0 | 1300 | 1500 | 86.7% |
| | S2 | 0 | 0 | 0 | 0 | 500 | 0 | 0 | 0 | 0 | 500 | 1000 | 50% |
| | S3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1500 | 0% |
| | S4 | 100 | 0 | 300 | 600 | 0 | 0 | 0 | 0 | 0 | 1000 | 1000 | 100% |
| | S5 | 0 | 800 | 400 | 0 | 0 | 0 | 0 | 0 | 0 | 1200 | 1500 | 80% |
| | S6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 | 0% |
| | S7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1500 | 0% |
| | S8 | 800 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 1000 | 1000 | 100% |
| | S9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 800 | 700 | 1500 | 1500 | 100% |
| Total | | | | | | | | | | | 6500 | 11500 | 57% |
| Actual demand | | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | |
| Limit demand [11] | | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | |

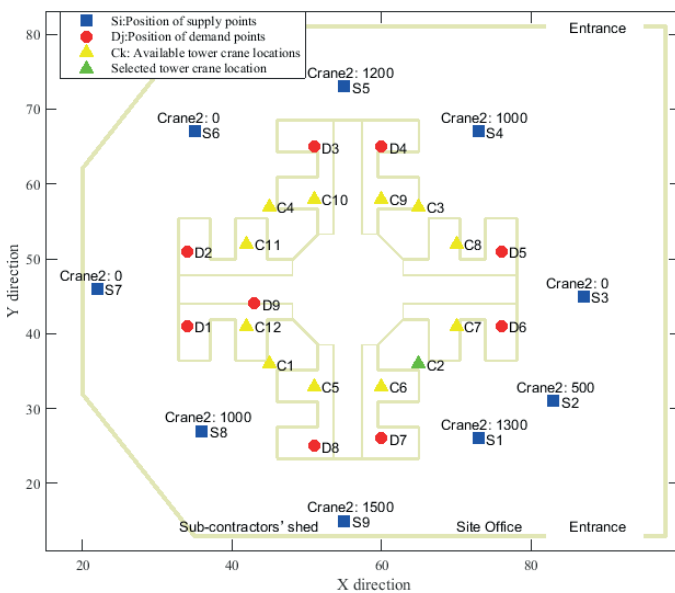


Fig. 12 ECBO Best single tower crane layout design

The encoding of single tower crane location is shown in Fig. 11. In this string, cells represent the material quantities into the demand point 1 from supply points 1 through 9. Similarly, cells represent the material quantities into the demand point 2 from supply points 1 through 9. And cells 19 through 81 are decoded in the same way. The number placed in cell 82

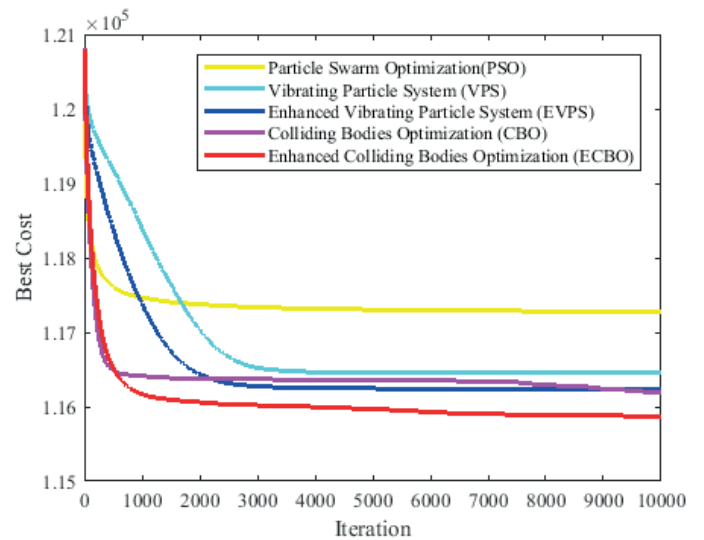


Fig. 13 Average cost convergence curves of multi-tower crane layout problem

represents the number of the selected tower crane location. The optimal material quantity and location alternative for the single tower crane Fig. 12 (decoded matrix) shows, with the best tower crane location shown at C2. Table 10 shows the ECBO optimal design for demand and supply points material quantities. As can be seen from Fig. 12 and Table 10, supply points S1, S2, S4, S5, S8, and S9 are closest to C2 and have allocated

| | | | | | | | | | | | | | | | | | |
|----------------|----|----|----|-----|-----|-----|-----|-----|----|----|----|-----|-----|-----|-----|-----|----|
| Cell number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Representation | 0 | 0 | 0 | 27 | 0 | 4 | 450 | 416 | 0 | 0 | 0 | 0 | 2 | 257 | 244 | 296 | 0 |
| Cell number | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| Representation | 0 | 0 | 0 | 0 | 213 | 234 | 249 | 0 | 1 | 1 | 0 | 0 | 30 | 257 | 256 | 0 | 0 |
| Cell number | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 |
| Representation | 0 | 0 | 1 | 246 | 252 | 0 | 0 | 0 | 0 | 0 | 1 | 260 | 249 | 91 | 0 | 0 | 0 |
| Cell number | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 |
| Representation | 0 | 0 | 0 | 416 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 72 | 4 | 0 | 0 | 0 |
| Cell number | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | | | |
| Representation | 0 | 0 | 0 | 303 | 0 | 0 | 377 | 0 | 0 | 0 | 1 | 78 | 443 | 2 | | | |

Fig. 14 Definition of the solution candidates for multi crane case (cell 1 to 82)

| | | | | | | | | | | | | | | | | | |
|----------------|----|----|----|-----|----|----|----|----|----|----|----|----|-----|----|----|----|----|
| Cell number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Representation | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Cell number | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| Representation | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 57 | 0 | 0 |
| Cell number | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 |
| Representation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cell number | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 |
| Representation | 0 | 0 | 0 | 282 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 369 | 1 | 0 | 0 | 0 |
| Cell number | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | | | |
| Representation | 0 | 0 | 2 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | | | |

Fig. 15 Definition of the solution candidates for multi crane case (cell 83 to 164)

Table 11 Best results of EBB-BC for single tower crane layout problem [12].

| | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | Actual supply | Limit supply | Supply degree |
|---------------|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|--------------|---------------|
| C2 | S1 | 73 | 0 | 0 | 0 | 0 | 250 | 250 | 250 | 250 | 1073 | 1500 | 71.5% |
| | S2 | 0 | 0 | 0 | 0 | 250 | 250 | 250 | 250 | 0 | 1000 | 1000 | 100% |
| | S3 | 0 | 0 | 0 | 250 | 250 | 100 | 150 | 26 | 250 | 1026 | 1500 | 68.4% |
| | S4 | 250 | 250 | 250 | 250 | 0 | 0 | 0 | 0 | 0 | 100 | 1000 | 100% |
| | S5 | 5 | 250 | 249 | 0 | 0 | 0 | 0 | 0 | 0 | 504 | 1500 | 33.6% |
| | S6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 | 0% |
| | S7 | 107 | 50 | 33 | 16 | 0 | 0 | 0 | 8 | 0 | 214 | 1500 | 14.3% |
| | S8 | 215 | 100 | 67 | 33 | 0 | 0 | 0 | 16 | 200 | 631 | 1000 | 63.1% |
| | S9 | 250 | 150 | 101 | 51 | 0 | 0 | 50 | 250 | 200 | 1052 | 1500 | 70.1% |
| Total | | | | | | | | | | | | 11500 | 57% |
| Actual demand | | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | |
| Limit demand | | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | |

Table 12 The Comparison result of algorithms for multi-TCLP.

| | Previous Works | | | Present Work | | | | |
|------------|----------------|------------|-------------|--------------|----------|----------|----------|----------|
| | PSO [12] | BB-BC [12] | EBB-BC [12] | PSO | VPS | EVPS | CBO | ECBO |
| Best cost | 115702.5 | 114236.3 | 113887.9 | 117063.9 | 116287.7 | 116128 | 116029.2 | 115807.6 |
| Mean cost | 116494.4 | 114501.6 | 113919.3 | 117275.4 | 116449.1 | 116238.8 | 116192.7 | 115876.9 |
| St. Dev. | 331.71 | 163.9 | 45.04 | 128.2 | 131.7 | 77.78 | 91.7 | 36 |
| Worst cost | 117143.1 | 114965.6 | 114037.6 | 117626 | 116678.9 | 116485.3 | 116392.7 | 115937.8 |

capacities of 1300, 500, 1000, 1200, 1000, 1500, respectively. Table 11 shows the best design of Ref. [12] (EBB-BC). C2 was the best tower crane location identified in Ref. [12]. Furthermore, the results demonstrate that ECBO not only optimized the tower crane location but also minimizes operating costs in line with demand and supply point capacity requirements.

5.2 Results and discussion for multi tower crane layout problem

The statistical results of the five meta-heuristics are listed in Table 12 and cost convergence curves are shown in Fig. 13.

Table 13 Best results of ECBO for multi-tower crane layout problem

| | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | Actual supply | Limit supply | Supply degree | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|---------------|--------------|---------------|-------|------|
| C2 | S1 | 0 | 0 | 0 | 0 | 1 | 260 | 416 | 72 | 0 | 749 | 750 | 99.5% | | |
| | S2 | 0 | 0 | 0 | 0 | 246 | 249 | 0 | 4 | 0 | 499 | 500 | 100% | | |
| | S3 | 0 | 0 | 0 | 30 | 252 | 91 | 0 | 0 | 0 | 377 | 750 | 750 | 50% | |
| | S4 | 27 | 2 | 213 | 257 | 0 | 0 | 0 | 0 | 0 | 0 | 499 | 500 | 49.9% | |
| | S5 | 0 | 257 | 234 | 256 | 0 | 0 | 0 | 0 | 0 | 0 | 747 | 750 | 49.8% | |
| | S6 | 4 | 244 | 249 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 497 | 500 | 49.7% | |
| | S7 | 450 | 296 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 747 | 750 | 49.8% | |
| | S8 | 416 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 78 | 495 | 500 | 49.5% | |
| | S9 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 303 | 443 | 750 | 750 | 50% | |
| C3 | S1 | 0 | 0 | 1 | 0 | 0 | 0 | 282 | 369 | 0 | 652 | 750 | 43.5% | | |
| | S2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 500 | 500 | 0.1% | |
| | S3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 750 | 750 | 0% | |
| | S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 500 | 500 | 0% | |
| | S5 | 0 | 0 | 1 | 57 | 0 | 0 | 0 | 0 | 0 | 0 | 58 | 750 | 750 | 3.9% |
| | S6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 500 | 500 | 0% |
| | S7 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 750 | 750 | 0.2% |
| | S8 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 4 | 500 | 500 | 0.4% |
| | S9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 49 | 0 | 49 | 750 | 750 | 3.3% |
| Total | | | | | | | | | | | 11500 | 57% | | | |
| Actual demand | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | | | | |
| Limit demand | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | | | | |

Table 14 Best results of EBB-BC for multi tower crane layout problem [12].

| | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | Actual supply | Limit supply | Supply degree | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|---------------|--------------|---------------|------|
| C3 | S1 | 245 | 250 | 0 | 0 | 0 | 0 | 242 | 250 | 0 | 987 | 750 | 131.6% | |
| | S2 | 250 | 0 | 0 | 0 | 0 | 0 | 250 | 0 | 0 | 500 | 500 | 100% | |
| | S3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 250 | 250 | 750 | 33.3% | |
| | S4 | 0 | 0 | 67 | 250 | 250 | 0 | 0 | 0 | 0 | 567 | 500 | 113.4% | |
| | S5 | 0 | 0 | 184 | 46 | 0 | 250 | 0 | 0 | 0 | 480 | 750 | 64% | |
| | S6 | 0 | 172 | 0 | 0 | 0 | 0 | 208 | 209 | 0 | 589 | 500 | 117.8% | |
| | S7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 750 | 750 | 0% |
| | S8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 500 | 500 | 0% |
| | S9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 750 | 750 | 0% |
| C6 | S1 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 250 | 0 | 350 | 750 | 46.7 | |
| | S2 | 50 | 0 | 0 | 0 | 250 | 0 | 0 | 0 | 200 | 500 | 500 | 100% | |
| | S3 | 249 | 244 | 250 | 250 | 0 | 0 | 0 | 0 | 0 | 993 | 750 | 132.4% | |
| | S4 | 0 | 134 | 199 | 0 | 0 | 0 | 0 | 0 | 0 | 333 | 500 | 66.6% | |
| | S5 | 0 | 0 | 0 | 54 | 0 | 0 | 0 | 0 | 0 | 54 | 750 | 750 | 7.2% |
| | S6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 500 | 500 | 0% |
| | S7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 64 | 64 | 750 | 750 | 8.5% |
| | S8 | 106 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 136 | 242 | 500 | 48.4% | |
| | S9 | 0 | 0 | 0 | 0 | 0 | 250 | 0 | 91 | 250 | 591 | 750 | 78.8% | |
| Total | | | | | | | | | | | 11500 | 57% | | |
| Actual demand | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | | | |
| Limit demand | 900 | 800 | 700 | 600 | 500 | 600 | 700 | 800 | 900 | 6500 | | | | |

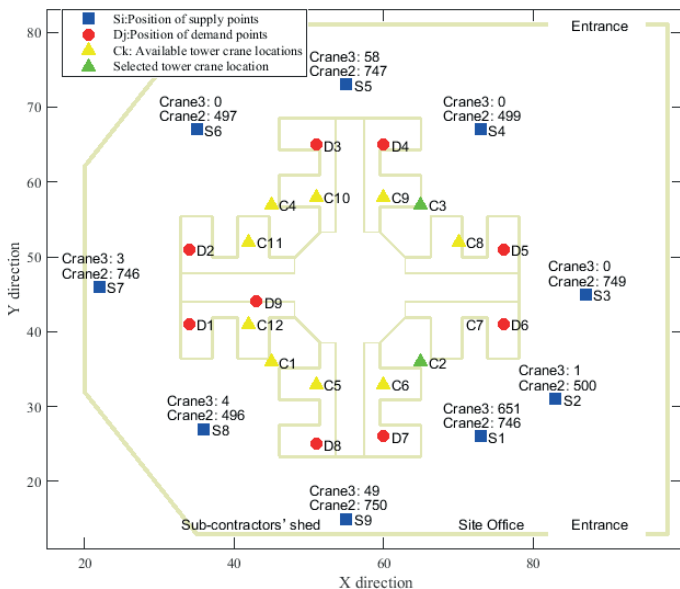


Fig. 16 ECBO Best multi-tower crane layout design

It shows that ECBO performs better than other algorithms and gives better solutions. As can be seen from Table 12, the mean costs are 115876.9, 116192.7, 116238.8, 116449.1, and 117275.4 for ECBO, CBO, EVPS, VPS, and PSO, respectively. Also, the best cost of the ECBO (115807.6) is better than other utilized algorithms. Moreover, the performance of both of ECBO, and EVPS are better than their standard versions (CBO, and VPS). Although Ref. [12] produced better costs, these result are not real since it neglected the influence of supply capacity constraint (See Eq. 14, and Table 14).

The encoding of multi-tower crane are shown in Figs. 14 and 15. For simplicity in understanding, the solution code of each crane is shown as a separated string. In fact, these two strings are connected. Fig. 16 (decoded matrix) shows the optimal material quantity and location alternative for the multi-tower crane, with the best tower crane locations shown at C2, and C3. In this strings cells in represent the material quantities into the demand point 1 from supply points 1 through 9 by first crane. Similarly, cells represent the material quantities into the demand point 1 from supply points 1 through 9. And other cells are decoded in the same way. The numbers placed in cells 82 and 164 represent the number of the selected tower crane locations.

Table 13 shows the best ECBO optimal design for demand and supply points material quantities. As seen in Fig. 16 and Table 13, supply points S1 through S9 are closest to C2 and have allocated capacities of 749, 499, 750, 499, 747, 497, 747, 495, 750, respectively. Also, supply points S1, S2, S5, S7, S8, and S9 are closest to C3 and have allocated capacities of 651, 1, 58, 3, 4, 49, respectively. Table 14 shows the best design of Ref. [12] (EBB-BC). C3, and C6 are the best tower crane location identified in Ref. [12]. Furthermore, the results demonstrate that ECBO not only optimizes the tower crane location but also minimizes operating costs in line with demand and supply point capacity requirements.

6 Conclusions

This paper presented a performance comparison of four newly developed meta-heuristic algorithms CBO, ECBO, VPS, and EVPS in terms of their effectiveness for resolving a practical TCL problem. These algorithms are inspired by physical laws. Results show that ECBO performs better in comparison with other algorithms for solving both case studies. The ECBO optimized both of crane location and operating costs. Also, the ECBO, and EVPS algorithms have better performance than their standard versions. Future works can execute hybridized algorithms to solve these problems.

References

- [1] Michalek, J., Choudhary, R., Papalambros, P. "Architectural layout design optimization". *Engineering Optimization*, 34(5), pp. 461–484. 2002. <https://doi.org/10.1080/03052150214016>
- [2] Sawhney, A., Mund, A. "Adaptive Probabilistic Neural Network-based Crane Type Selection System". *Journal of Construction Engineering and Management*, 128(3), pp. 265–273 2002. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2002\)128:3\(265\)](https://doi.org/10.1061/(ASCE)0733-9364(2002)128:3(265))
- [3] Huang, C., Wong, C. K., Tam, C. M. "Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming". *Automation in Construction*, 20(5), pp. 571–580. 2011. <https://doi.org/10.1016/j.autcon.2010.11.023>
- [4] Li, H., Love, P. E. D. "Site-Level Facilities Layout Using Genetic Algorithms". *Journal of Computing in Civil Engineering*, 12(4), pp. 227–231. 1998. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1998\)12:4\(227\)](https://doi.org/10.1061/(ASCE)0887-3801(1998)12:4(227))
- [5] Gharaie, E., Afshar, A., Jalali, M. R. "Site layout optimization with ACO algorithm". In: *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pp. 90–94. 2006.
- [6] Kaveh, A., Khanzadi, M., Alipour, M., Rastegar-Moghaddam, M. "Construction site layout planning problem using two new meta-heuristic algorithms". *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 40(4), pp. 263–275. 2016. <https://doi.org/10.1007/s40996-016-0041-0>
- [7] Tam, C. M., Tong, T. K. L. "GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction". *Construction Management and Economics*, 21(3), pp. 257–266. 2003. <https://doi.org/10.1080/0144619032000049665>
- [8] Zhang, P., Harris, F. C., Olomolaiye, P. O., Holt, G. D. "Location optimization for a group of tower cranes". *Journal of Construction Engineering and Management*, 125(2), pp. 115–122. 1999. [https://doi.org/10.1061/\(ASCE\)0733-9364\(1999\)125:2\(115\)](https://doi.org/10.1061/(ASCE)0733-9364(1999)125:2(115))
- [9] Tam, C. M., Tong, T. K. L., Chan, W. K. W. "Genetic Algorithm for Optimizing Supply Locations around Tower Crane". *Journal of Construction Engineering and Management*, 127(4), pp. 315–321. 2001. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2001\)127:4\(315\)](https://doi.org/10.1061/(ASCE)0733-9364(2001)127:4(315))
- [10] Kaveh, A., Vazirinia, Y. "Tower cranes and supply points locating problem using CBO, ECBO, and VPS". *International Journal of Optimization in Civil Engineering*, 7(3), pp. 393–411 2017. <http://www.iust.ac.ir/ijoc/article-1-305-en.html>
- [11] Lien, L. C., Cheng, M. Y. "Particle bee algorithm for tower crane layout with material quantity supply and demand optimization". *Automation in Construction*, 45, pp. 25–32. 2014. <https://doi.org/10.1016/j.autcon.2014.05.002>

- [12] Herdany, A. A. "Evolutionary Big Bang - Big Crunch Algorithm for Construction Engineering Optimization Problem". MSc. dissertation, Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, 2014.
- [13] Kaveh, A., Mahdavi, V. R. "Colliding bodies optimization: a novel meta-heuristic method". *Computers & Structures*, 139, pp. 18–27. 2014. <https://doi.org/10.1016/j.compstruc.2014.04.005>
- [14] Kaveh, A., Ilchi-Ghazaan, M. "Enhanced colliding bodies optimization for design problems with continuous and discrete variables". *Advances in Engineering Software*, 77, pp. 66–75. 2014. <https://doi.org/10.1016/j.advengsoft.2014.08.003>
- [15] Kaveh, A., Ilchi-Ghazaan, M. "A new meta-heuristic algorithm : vibrating particles system". *Scientia Iranica*, 24(2), pp. 551–566. 2017. <https://doi.org/10.24200/sci.2017.2417>
- [16] Kaveh, A., Hosseini-Vaez, S. R., Hosseini, P. "Enhanced vibrating particles system algorithm for damage identification of truss structures". *Scientia Iranica*, p. 25. 2017. <https://doi.org/10.24200/sci.2017.4265>
- [17] Goldberg, D. E. "Genetic algorithms in search, optimization, and machine learning". Addison-Wesley Publishing Company, Boston, MA, USA. 1989.
- [18] Eberhart, R., Kennedy, J. "A new optimizer using particle swarm theory". In: *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, Nagoya, pp. 39-43. IEEE, 1995. <https://doi.org/10.1109/MHS.1995.494215>
- [19] Kaveh, A., Bakhshpoori, T., "Water evaporation optimization: a novel physically inspired optimization algorithm". *Computers & Structures*, 167, pp. 69–85. 2016. <https://doi.org/10.1016/j.compstruc.2016.01.008>
- [20] Atashpaz-Gargari, E., Lucas, C. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition". In: *Evolutionary computation, IEEE Congress*, pp. 4661–4667. 2007. <https://doi.org/10.1109/CEC.2007.4425083>
- [21] Kaveh, A., Talatahari, S. "A novel heuristic optimization method: charged system search". *Acta Mechanica*, 213(3–4), pp:267–289. 2010. <https://doi.org/10.1007/s00707-009-0270-4>
- [22] Erol, O. K., Eksin, I. "A new optimization method: big bang-big crunch". *Advances in Engineering Software*, 37(2), pp. 106–111. 2006. <https://doi.org/10.1016/j.advengsoft.2005.04.005>
- [23] Kaveh, A. "Advances in Metaheuristic Algorithms for Optimal Design of Structures". Springer International Publishing, 2nd edition, Switzerland. 2017. <https://doi.org/10.1007/978-3-319-05549-7>
- [24] Choi, C. W., Harris, F. C. "A model for determining optimum crane position". *Proceedings of the Institution of Civil Engineers*, 90(3), pp. 627–634. 1991. <https://doi.org/10.1680/iicep.1991.14735>
- [25] Koumoussis, V. K., Georgiou, P. G. "Genetic algorithms in discrete optimization of steel truss roofs". *Journal of Computing in Civil Engineering*, 8(3), pp. 309–325. 1994. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1994\)8:3\(309\)](https://doi.org/10.1061/(ASCE)0887-3801(1994)8:3(309))
- [26] Goldberg, D. E., Lingle, R. "Alleles loci and the travelling salesman problem". *Proceedings of first International Conference on Genetic Algorithms and Their Applications, Carnegie-Mellon Univ, Pittsburgh*, pp. 154–159. 1985.
- [27] Wright, A. H. "Genetic algorithms for real parameter optimization". *Foundations of Genetic Algorithms*, 1, pp. 205–218. 1991. <https://doi.org/10.1016/B978-0-08-050684-5.50016-1>
- [28] O'Reilly, U. M., Oppacher, F. "Program search with a hierarchical variable length representation: Genetic programming simulated annealing and hill climbing". *International Conference on Parallel Problem Solving from Nature. Springer Verlag (Lecture Notes in Computer Science (866))*, pp. 397–406. 1994. https://doi.org/10.1007/3-540-58484-6_283
- [29] Tubaileh, A. "Working time optimal planning of construction site served by a single tower crane". *Journal of Mechanical Science and Technology*, 30(6), pp. 2793–2804. 2016. <https://doi.org/10.1007/s12206-016-0346-8>
- [30] MATLAB Software, 2014. *Mathworks*, Retrieved from <http://www.mathworks.com/>