KICEM Journal of Construction Engineering and Project Management
Online ISSN 2233-9582

www.jcepm.org
http://dx.doi.org/10.6106/JCEPM.2017.7.2.001

Application of Parameters-Free Adaptive Clonal Selection in Optimization of Construction Site Utilization Planning

Xi Wang[1], Abhijeet S. Deshpande, Ph.D.[2], Gabriel B. Dadi, Ph.D., P.E., LEED AP.[3]

Abstract: The Clonal Selection Algorithm (CSA) is an algorithm inspired by the human immune system mechanism. In CSA, several parameters needs to be optimized by large amount of sensitivity analysis for the optimal results. They limit the accuracy of the results due to the uncertainty and subjectivity. Adaptive Clonal Selection (ACS), a modified version of CSA, is developed as an algorithm without controls by pre-defined parameters in terms of selection process and mutation strength. In this paper, we discuss the ACS in detail and present its implementation in construction site utilization planning (CSUP). When applied to a developed model published in research literature, it proves that the ACS are capable of searching the optimal layout of temporary facilities on construction site based on the result of objective function, especially when the parameterization process is considered. Although the ACS still needs some improvements, obtaining a promising result when working on a same case study computed by Genetic Algorithm and Electimze algorithm prove its potential in solving more complex construction optimization problems in the future.

Keywords: CSUP, Optimization, Clonal Selection Algorithm, Adaptive Clonal Selection

I. INTRODUCTION

Construction Site Utilization Planning (CSUP) is the decision making process for determining the locations of temporary facilities within the boundary of a construction site by identifying spatial, functional relationships between the temporary facilities. A detailed planning of the site layout can potentially enable the construction manager to make significant improvement in productivity by minimizing travel time and cost, especially for large projects [1, 2]. Sometimes site planning is neglected in the construction phase and the constructors sometimes think that it can be done as the construction progresses [2]. Poor facility layout could potentially result in work delays, material storage; multiple handling of materials, schedule delays, loss of time and money, and unsafe working conditions [3]. Thus, CSUP is an important and indispensable part of construction engineering. The goal is to identify an optimal layout from a large number of alternative solutions so that a set of predetermined facilities are appropriately located while satisfying site specific constraints [4, 5]. The CSUP is also one of the most sophisticated problems in construction process because it involves many parameters and items that are needed to be identified and defined such as the structures to be constructed on site, facilities that serve the whole project, internal routes, and site boundaries. It is important to identify the number and size of temporary facilities to be laid out, to identify the constraints between facilities, and to determine the relative positions of these facilities [6]. Therefore, CSUP is a complex combinatorial optimization problem [7], which means it is difficult to obtain the optimal layout by hand calculations or simple optimization algorithms.

Due to the complexity of CSUP, over last two decades, a large amount of research has been focused on formulating and identifying algorithms to optimize the site layouts. The objective is to identify an optimal layout from a large number of alternative solutions so that a set of predetermined facilities are appropriately located while satisfying site specific constraints [4, 5]. According to computational complexity theory, the layout problems are known to be NP-hard (Nondeterministic polynomial time hard) combinatorial optimization problem [6]. The optimal solutions can be computed only for small or greatly restricted problems, and the process may require exponential computation time [8-10]. Most traditional optimization algorithms such as linear programming are incompetent for the complex optimization problems like CSUP. First, most classic algorithms are deterministic. Deterministic optimization methods need gradient evaluations as it uses the function values and their derivatives which works well for smooth unimodal problems unless there is some discontinuity in the objective function [11]. Also, they are not capable of ensuring the solution is global minimum because of non-randomness [12]. Therefore, researchers have developed several meta-heuristic methods to obtain feasible solutions to the layout problems. Meta-heuristics are general algorithmic frameworks, often nature-inspired, designed to solve complex optimization problems. The Meta-heuristics algorithms, such as Ant Colony Optimization (ACO), Evolutionary Algorithms (EAs), Iterative Local Search (ILS), Simulated Annealing (SA), Tabu Search (TS), and

[1] Ph.D. Student, University of Kentucky, Lexington, Kentucky, U.S.A, xi.wang@uky.edu (*Corresponding Author)
[2] Assistant Professor, University of Cincinnati, Cincinnati, Ohio, U.S.A, deshpaa@ucmail.uc.edu
[2] Assistant Professor, University of Kentucky, Lexington, Kentucky, U.S.A, gabe.dadi@uky.edu

Particle Swarm Algorithms (PSO), are emerging as a successful alternative to more classical approaches also for solving optimization problems that include in their mathematical formulation uncertain, stochastic, and dynamic information [13]. In recent decades, more evolutionary algorithms have been applied in solving complicated problems in construction. Li and Love (1998) were the first to propose the use of Genetic Algorithm (GAs) for the optimization of site layouts. They introduced a scheme for representing construction site-level facility layout problems into strings suitable for Genetic Algorithm operation. They also demonstrated the robustness of GA approach in solving layout problem. This formulation has been adopted in most of the further studies. Based on Genetic Algorithm, Hegazy and Elbeltagi (1999) have developed Evosite, which is a comprehensive system for site layout planning. The EvoSite uses a spreadsheet to represent the site and the facilities, and automates the evolution of layout solutions. Abdel-Raheem and Khalafallah (2012) used Electimize, which is one of EAs simulating the phenomenon of the flow of electrons in multi-branch electric circuits, to solve construction site layout planning optimization problem. Among all the meta-heuristic methods, Artificial Immune System (AIS), which has been inspired by the biological process of human immune systems, is also an innovative and effective optimization method. The Clonal Section Algorithm, Negative Selection Algorithm and Immune Network Algorithm are the three main algorithms developed from the principles of Artificial Immune Systems. Artificial Immune System (AIS) algorithms imitate the principles of the human immune systems and it is inspired by its features such as the distributed memory, robustness, self-organizing, and decentralized control mechanism. Dasgupta (1993) published the first study about AIS [14]. De Castro and Timmis (2002) provided the details of the main AIS models such as clonal selection, immune networks, and negative selection theories [15]. After that, Hart and Timmis (2008) state that the AIS can be applied in many areas such as numerical function optimization, combinatorial optimization, image processing, robotics, etc [16]. Timmis (2007) states the mechanism of AIS operation mainly based on the clonal selection theory [17]. The clonal selection theory in an immune system is used to explain the basic response of the adaptive immune system to an antigenic stimulus. The theory is based on the idea that only cells that are capable of recognizing an antigen will proliferate when an antigen is detected by the adaptive immune system [18]. In recent decades, the large amount of studies suggests that potential implementation of this approach and related methods in various areas such as scheduling [19-23], traveling salesman problem [24, 25], safety security [26-28], and many multi-objective optimization problems [29-33]. The AIS principle also has been cooperated with a variety of other algorithms particle 140 swarm optimization (PSO) [34-36], fuzzy function [37-39] and various other adaptive methods.

From all the research illustrated above, the obtained results of AIS/CSA are reported to perform better in comparisons and confirm the validity of this developed approach. AIS have been proved as a useful and efficient addition to the plethora of population-based algorithms. However, AIS has seldom been applied to solve the facility layout problem, particularly in the construction field. Ulutas and Islier (2009) proposed that CSA could be employed to solve the dynamic facility layout problem. The aim is to minimize the sum of handling and re-layout costs by devising an individual layout for each distinctive production period. In that study, the performance of CSA was examined by using test problems from the literature [40]. CSA was found to provide better solution for large sizes problem than existing results obtained by other methods. Ulutas and Sadan (2012) used CSA to solve the unequal area facility layout problem with flexible bay structure. The overall result stated that CSA with flexible bay structure representation was successful in 95 percent of the test problem when compared the best so far flexible bay structure results within short computation time [41]. Wang X. et al. (2016) verified the advantage of the CSA in computation capacity by comparing results obtained by GAs, which have been applied in a benchmark case study of many publications [56].

However, the CSA has several drawbacks that may negatively impact the performance when dealing with CSUP optimization problems: multiple evaluations of objective function potentially reduce the performance efficiency of the algorithm; and a binary-valued representation forces a predefined degree of accuracy on the computation results. The most important is the CSA has several parameters that need optimizing by sensitivity analysis because different combinations of parameters can have significant influence on the results. In CSA, parameters need to be defined before running the algorithm. The optimal values of parameters can be found after numerous comparisons between computed results. The accuracy is unreliable because the pre-definitions could yield great errors. Instead, the algorithm could be more objective and powerful if performing a deterministic optimization in the region where the global optimum appears to be rather than mapping out the parameter space to search for the best combination of parameters [55]. Motivated by these problems, this paper aims to revisit the CSA in general and introduce Adaptive Clonal Selection (ACS), which is a novel computation strategy to make the CSA parameter-free when looking for the optimal construction layout. The main objective of developing ACS is to overcome the limitations of the CSA mentioned above and provide the construction industry with a reliable optimization tool that is able to solve various construction problems especially layout planning. This paper briefly discusses the biological process underlying the algorithms and presents the ACS procedures in detail. Also, this paper presents a case study to validate the advantages of applying ACS in CSUP optimization problems compared with the CSA.

II. ADAPTIVE CLONAL SELECTION (ACS)

A. Clonal Selection Algorithm (CSA)

Clonal selection is one of the most efficient mechanisms of the immune system because evolution of cells in the adaptive immune system occurs based on two basic principles. First, only cells that recognize the antigens are selected to proliferate. Second, in the cloning process, the reaction with antigen is strengthened by the hypermutation, also called affinity maturation, which is a result of random genetic re-combinations. Inspired by this learning and memory capacity of clonal selection mechanism, researchers developed the CSA as a computational tool to solve complex engineering optimization problems. When applying it to optimize construction layouts, the solution steps are as follows:

1) Randomly generate a size of population of P antibodies. Each antibody represents one layout.

2) A subset, N, from the layout population P, is selected based on the affinities, which is objective function value.

3) Clone each selected layout and the number of clones should be proportional to its selection probability.

4) After cloning, each layout is strengthened by hypermutation. The mutation rate according to the Equation (1),

$$\alpha = e^{(-\rho * f)} \qquad (1) \qquad [43]$$

Where α represents the mutation rate, and f is the normalized affinity value. The ρ is a parameter that controls the smoothness of the inverse exponential. After mutation, if there is no improvement in the objective function value, the parent antibodies survive rather than the mutated one.

5) After cloning and hypermutation, a percent of layouts having higher objective function values in new generation are replaced by randomly generated new ones to improve the diversity of population.

6) Convergence and Termination: When the results no longer have changes after a number of generations, the layout with lowest objective function value is selected as the best solution to the optimization problem. If not, the algorithm iterates starting from step 2.

B. Comparisons with the CSA

In the procedures of applying the CSA in CSUP optimization described above, the CSA has some drawbacks in terms of processing parameters. First, how to decide the size of subset N in step 2? The size of N has a dominant effect on the scope of the searching space for the algorithm and the diversity of the general population. Second, how to select the value of the parameter ρ? This parameter controls the amount of mutations of each cloned layout in step 5. The hypermutation strategy is the key to reveal the inherent potential of a layout and then improve its objective function value. The mutating extent for each layout should be accurate and reasonable. Last, the number of clones created in each generation is usually larger than the initial population size, which means the number of objective function evaluations would be higher. Besides searching capacities, the performance efficiency is also a crucial factor for an optimization tool implemented in construction industry. It is non-trivial to define the best values of these parameters and their optimal combinations without performing large amount of sensitive analysis. That is the reason why the ACS is introduced because it takes the parameterization process into consideration. To be more specific, the improvements to the CSA are reflected in three aspects:

1) Self-Adaptive N Parameter in Selection: In CSA, Given a population size of layouts, the number of layouts N will be selected for cloning is pre-defined. In ACS, it will be adjusted by the Equation (2)

$$N_{n+1} = N_n + \rho_n * N\,(0, 1) \qquad (2)$$

In Equation (2), the number of selected layouts in *n+1* generation depends on the mutation control parameter of the previous generation. The N can be adjusted by summing a Gaussian random variations of the mutation parameter and the number of selected layouts in generation n [46]. If the mutation success rate is higher, the mutation control parameter ρ will become smaller. Thus, the N in the *n+1* generation decreases because the algorithm prefers smaller changes. Large values of N may not help find better layouts when the population is not far beyond the optimal solution. Therefore, this equation created a self-adaption process between two generations. The initial value of N is equal to 1, which does not need tuning and will not be involved any specific run. It is possible to make the algorithm get into local optima and reach the convergence too soon if the initial value of N is large.

2) Self-Adaptive ρ Parameter in Mutation: In Equation (1), the ρ is a parameter that controls the strength of mutation, as shown in FIGURE I.

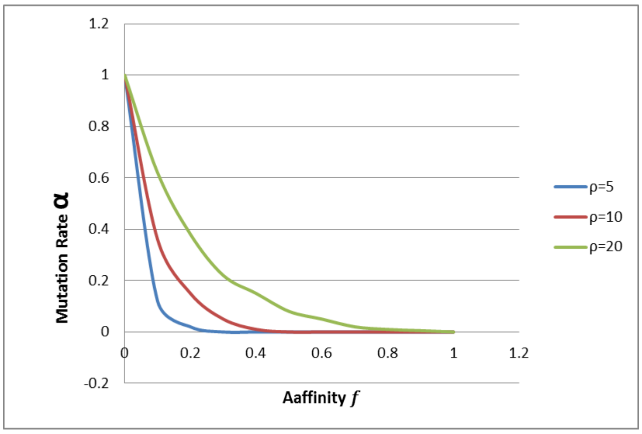

FIGURE I
THE RELATION BETWEEN NORMALIZED AFFINITY AND ITS MUTATION RATE FOR DIFFERENT VALUE OF ρ

The mutation rate heavily depends on the selection of ρ. To avoid subjectivity in the selection process, large amounts of sensitive analysis is necessary to identify the optimal value. Therefore, without the parameter ρ, the algorithm would become more efficient, which is important for complex optimization problems like CSUP. The method of removing the ρ parameters of ACS is based on the self-adaption principles and 1/5th-rule of Evolution Strategy (ES). In ES, tuning the mutation strength in such a way that the mutation success rate is about 1/5 in order to obtain optimal performance of the algorithm [44]. Successful mutation means the affinity (objective function value) of a layout is improved after mutation. Otherwise, the mutation fails and the parent one will stay for the next generation.

The implementation of the 1/5th-rule can be described in three steps:

Step1. Perform the algorithm for a number G of one generation, and keep the ρ constant during this generation.

Step2. Determine the success rate P of one generation by the Equation (3)

$$P = G_S / G \quad (3)$$

G_S: The number of clones obtained by successful mutation in one generation
G: The total number of clones in one generation.

Step3. Change the ρ according to the Equation (4)

$$\rho = \begin{cases} \rho \, / \, 1.3\,, & if\ P > 1/5 \\ \rho * 1.3, & if\ P < 1/5 \\ \rho = 1.3, & if\ P = 1/5 \end{cases} \quad (4)$$

In Equation (4), although the adjustment value 1.3 is also a parameter, it has been demonstrated to be the most reasonable and general value for adjusting the strategy parameter such as mutation strength in ES [45]. By following this rule and principle, the individuals will learn the optimal mutation strength during the evolution process.

3) The number of objective function evaluation: In the CSA, the number of clones created in each generation is usually larger than the initial population size, which may cause a mass of redundant computation for objective function evaluation. It is important to effectively control the total number of clones to ensure the efficient performance of the algorithm. In ACS, the number of clones for one layout is equal to the product of its selection probability and the number of layouts in the initial population. The total number of the clones reproduced by selected layouts is equal to the total number of individuals in the initial population. Thus, the population size keeps balanced before and after cloning. Otherwise, the layouts selected at the first run are always superior in the later generation, which discourages the global search and contributes to the local optima. The following section will go into more detail about the adaptive process of the ACS from a practical perspective based on a benchmark case study.

C. Case Study

Over the last two decades, a wide variety of algorithms were developed to solve site layout optimization problems such as Genetic Algorithm [3-4, 47], Ant Colony Optimization [2, 48-49], Neural Network [50, 51], etc. This paper aims to demonstrate the effectiveness and improvements of adopting ACS in solving CSUP optimization problems by conducting a comparison with other published methodologies. Rodriguez-Ramos (1982) employed CRAFT quantitative techniques in site layout planning [54]. Hegazy and Elbetagi (1999) used Genetic Algorithm to solve the same problem [2]. Recently, Abdel-Raheem and Ahmed Khalafallah (2012) applied the Electimize algorithm and obtained a better result [52]. The case studies from these three publications were selected as a benchmark because they used the same CSUP problems.

1). Site and Facility Representation (Encoding) Scheme: In the previous model, the construction site is a two-dimensional grid composed of 15×17 squares of equal areas. The area of each grid is the greatest common divisor of the areas of all temporary facilities. In this model, the area of each unit grid is 100 ft^2. The location reference is used to represent the location of any facility on the site and it is calculated as Equation (5)

Location reference = (row position - 1) × total columns + column position (5)

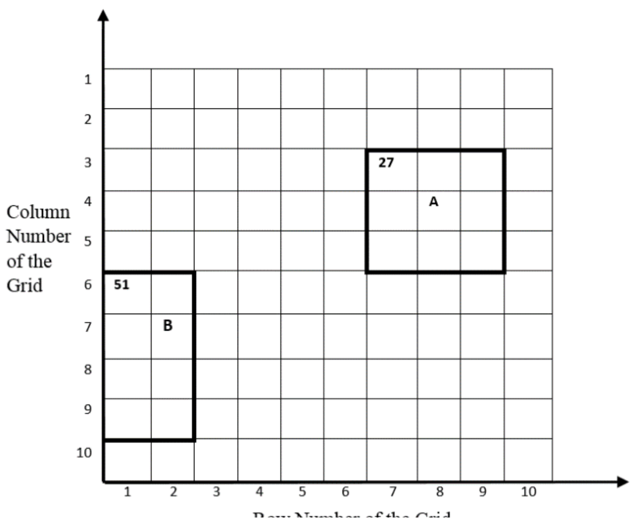

FIGURE II
LOCATION REFERENCE OF FACILITIES ON A SITE

The location reference represents the starting position at which a temporary facility is to be located within the boundary of the construction site, as shown in FIGURE II, the location reference of Facility A is (3-1)×10+7=27. Also, a facility is located on the site as a number of unit grids. For instance, the Facility A, the area of it is 900 ft^2, can be represented as nine grid units on the site, 10 ft^2 for each grid. Although there are three alternative methods for placing a temporary facility on the site and they are horizontal, vertical and rectangular, the arrangement of each facility is predefined in the published example. All facilities are rectangles instead of irregular shapes. Therefore, the arrangement methods remain unchanged during computations.

Similar to many other optimization algorithms, the ACS also requires a representation scheme to encode the layout so that the algorithms can be applicable to this specific optimization problem. According to the biological immune theory, an antibody is produced through combination of different gene components randomly selected from different gene libraries. An antibody is alike an attribute string of predetermined length. The elements of a string are genes randomly generated from a pre-defined population. The length of gene represents the total number of temporary facilities. Each element is the location reference of a temporary facility. In this case, there are total 17 facilities on a construction site, and 9 of them are fixed facilities, as shown in TABLE I.

TABLE I
FACILITIES INFORMATION IN APPLICATION EXAMPLE

	Name	Size (ft^2)	Arrangement (0=H, 1=V, 2=R)	Facility Type (1=fixed, 0=not fixed)
Fixed Facility	Project I	1,200	0	1
	Project J	600	0	1
	Project K	1,200	0	1
	Project L	800	0	1
	Project M	600	0	1
	Project N	1,500	0	1
	Project O	1,600	0	1
	Project P	1,200	0	1
	Project Q	1,200	0	1
Flexible Facility	Field office	600	0	0
	Warehouse	600	1	0
	Reinforcing bars shop 1	600	0	0
	Reinforcing bars shop 2	600	0	0
	Excavated material	800	0	0
	Patch plant	800	0	0
	Subcontractor's office	600	1	0
	Formwork shop	800	0	0

Note: H=horizontal; V=vertical; R=rectangular

2). The Objective Function: The objective function in algorithm is used to evaluate the affinity or goodness of a layout (antibody). The objective function value is also the criteria of selection and mutation strength in the whole process. Although various criteria could be adopted to evaluate a candidate layout solution, minimization of costs related to the transportation between facilities is always selected. The travel distance is what most directly affects the transportation cost. Li and Love (1998) used genetic algorithm to minimize the total traveling distance of site personnel between facilities [4]. Hegazy and Elbeltagi (1999) constructed an objective function to arrive at the optimum layout that results in the least travel distance. Lien and Cheng (2012) applied a particle bee algorithm to minimize the total travel distance between site facilities [53]. The total traveling distance is calculated by multiplying the desired proximity weight by the distance between facilities. It is one of the most objective and reasonable references that can be used to reflect the consumption on a site according to literature. The desired proximity weight reflects the closeness relationship between facilities. These relationships reflect a certain preference in having a close or far distance between each facility, and they are determined by using the proximity weights based on prior publications [2] [52]. A high proximate weight means two facilities share a high level of interaction [2]. A symmetrical preference matrix is constructed based on a pair-wise assessment, as shown in TABLE II and TABLE III. The actual distance between two facilities is the Euclidean distance between the facilities centroid positions, as shown in FIGURE III.

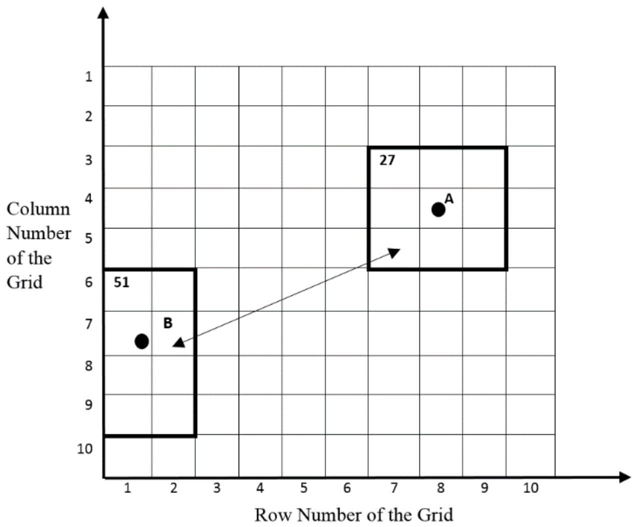

FIGURE III
ACTUAL DISTANCE BETWEEN FACILITIES

The algorithm aims to minimizing the total distance in order to find out the optimal layout. Due to this paper's goal is to prove the efficiency of ACS by comparing with

5

three other published methodologies in the same case study, the objective function in this paper is based on those prior publications. Rodriguez-Ramos (1982), Hegazy and Elbetagi (1999), and Abdel-Raheem and Ahmed Khalafallah (2012) give the objective function as following Equation (6):

$$Total\ travel\ distance = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} f_{ij} \qquad (6)$$

n: Total number of facilities
d_{ij}: Actual distance between facility i and facility j.
f_{ij} : Desired proximity between facility i and facility j.

TABLE II
PROXIMITY WEIGHT MATRIX BETWEEN FLEXIBLE FACILITIES

Facility	Field office	Warehouse	Rein. shop1	Rein. shop2	Exc. material	Patch Plant	Sub office	Form shop
Field office	0	10	2	4	0	2	0	2
Warehouse	10	0	4	4	0	0	0	20
Reinforcing bars shop 1	2	4	0	0	0	0	0	0
Reinforcing bars shop 2	4	4	0	0	0	0	0	0
Excavated material	0	0	0	0	0	0	0	0
Patch plant	2	0	0	0	0	0	0	0
Subcontractor's office	0	0	0	0	0	0	0	0
Formwork shop	2	20	0	0	0	0	0	0
Project I	2	4	20	0	0	100	6	20
Project J	2	6	20	0	0	200	6	20
Project K	6	4	20	0	0	100	6	20
Project L	10	2	0	20	0	400	6	20
Project M	2	4	0	40	0	300	6	20
Project N	2	6	0	60	0	100	6	20
Project O	20	20	20	0	0	500	6	60
Project P	4	4	20	0	0	400	6	20
Project Q	2	2	0	20	0	300	6	20

TABLE III
PROXIMITY WEIGHT MATRIX BETWEEN FIXED FACILITIES

Facility	I	J	K	L	M	N	O	P	Q
Field office	2	2	6	10	2	2	20	4	2
Warehouse	4	6	4	2	4	6	20	4	2
Reinforcing bars shop 1	20	20	20	0	0	0	20	20	0
Reinforcing bars shop 2	0	0	0	20	40	60	0	0	20
Excavated material	0	0	0	0	0	0	0	0	0
Patch plant	100	200	100	400	300	100	500	400	300
Subcontractor's office	6	6	6	6	6	6	6	6	6
Formwork shop	20	20	20	20	20	20	60	20	20
Project I	0	0	0	0	0	0	0	0	0
Project J	0	0	0	0	0	0	0	0	0
Project K	0	0	0	0	0	0	0	0	0
Project L	0	0	0	0	0	0	0	0	0
Project M	0	0	0	0	0	0	0	0	0
Project N	0	0	0	0	0	0	0	0	0
Project O	0	0	0	0	0	0	0	0	0
Project P	0	0	0	0	0	0	0	0	0
Project Q	0	0	0	0	0	0	0	0	0

3). Solution Steps of the ACS in the case study

Step 1. Population initialization: Randomly generate a size of population of P layouts. Each layout is represented as a string. A layout string includes 17 location references. The position of fixed facilities remain the same for the whole construction process.

Step 2. Affinity Evaluation: Calculate the total traveling distance of each layout using the objective function. The affinity should be the inverse of the objective function value as in Equation (7) since the objective of site layout planning is to minimize distances. Thus, the optimal layout with highest affinity has the lowest total traveling distance.

$$Affinity = \frac{1}{Total\ Distance} \qquad (7)$$

Step 3. Clonal Selection: The number of antibodies selected N begins at 1 in the first generation, and the Roulette Wheel selection strategy is adopted in this step. Roulette wheel selection is one of commonly used selection operators in evolutionary computing. The basic idea of the selection process is to stochastically select from one generation to create a new population for the next generation. The selection principle is that the layouts with higher affinity have a higher survival probability. The probability, P_i, is determined by the affinity of each layout. If f_i is the affinity of an individual i in the population, its probability P_i is as shown in (8),

$$P_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \qquad (8)$$

The procedures of the Roulette Wheel selection strategy are as follows:

According to the probabilities of N individuals, a circular wheel is divided into N sections and the area of each section is proportional to its selection probability. Using an integer

i between 1 and N denote the sequence number of the sections.

Randomly generate a number m between 0 and 1 as the selection point.

If $p_1+p_2+...+p_{i-1} < m < p_1+p_2+...+p_i$, the i^{th} layout will be selected. Although layouts with high affinity have high probabilities to be selected, it is still possible that low affinity layouts will also be selected. A layout with low affinity it may become a higher quality candidate after hypermutation so that diversity can be improved.

Step 4. Cloning: Clone the layouts that are selected from the initial population. The initial number of clones for the first selected layout is also equal to 1. After that, the number of clones for each selected layout should be proportional to its selection probability. The number of clones for one layout is equal to the product of its selection probability and the total number of selected layouts (N) in the initial population, as shown in TABLE IV.

TABLE IV
THE METHOD OF CALCULATING THE NUMBER OF CLONES

Layout	Selection Probability	Number of Clones
Layout 1	P_1	$N * P_1$
Layout 2	P_2	$N * P_2$
Layout 3	P_3	$N * P_3$
Layout 4	P_4	$N * P_4$
:	:	:
:	:	:
:	:	:
Layout N	P_N	$N * P_n$

Step 5. Hypermutation: After cloning, each cloned layout is strengthened further by the hypemutation. The mutation is inversely proportional to the affinity of the layout. According to Equation 1, the parameter ρ is to control the smoothness of the inverse exponential, and its initial value is equal to 1. The mutation strategy is random replacement of facility positions. For example, if the number of mutation for one layout is 3, then the positions of three randomly selected facilities, which are not fixed facilities, will be replaced randomly by other available positions. In CSA, the affinity maturation is performed through two mutation procedures, inverse mutation and pair-wise interchange mutation. Both of them aim to inter-exchanging the positions between two or more facilities rather than replacement by random locations. It is not difficult to understand that random replacement strategy is more helpful to avoid the limitations of local optima.

Step 6. Affinity Evaluation and Receptor Editing: After cloning and hypermutation, all the layouts undergoes different changes in objective function value. Layouts need to be reorganized based on their updated values. The algorithm will eliminate the non-improved layouts. The rest will replace their parents to be a member of new population. The population size remain unchanged before and after cloning. Then, a small proportion of layouts that having low affinities in the population are replaced by randomly generated new layouts. This procedure helps the algorithm escape from local optima on an affinity landscape. The newcomers are crucial for the diversity in the population. They increase the probability of searching potential better results in a larger population. However, the proportion of new layouts should be within a reasonable range. Unlimited randomness makes the algorithm inefficient, which may lead to a large increase of difficulties in finding the optimal solution. At the same time, the parameters need updates for next generation based on the mutation success rate.

Step 7. Convergence and Termination: When the minimum objective function value of a generation does not show any improvement over the previous generation, the algorithm reaches the convergence. The layout is the best layout, then the algorithm stops; otherwise, the algorithm goes back to step 2 and the cycle continues.

4). Computational Results: The Adaptive Clonal Selection was coded in MATLAB, and all experiments were performed on a 64 bit Window PC with a 2.00 GHz CPU and 32.0 GB RAM. The Adaptive Clonal Selection (ACS) is capable of finding a new optimal solution (18,486 ft) for this case using 150 initial population size and 500 generations, as shown in FIGURE IV. The earliest result for this problem published by Rodriguez-Ramos in 1982 is 22,386 ft. After that, Hegazy and Elbetagi in 1999 obtained an improved result for this problem by using Genetic Algorithm, and the value is 22,229 ft. In 2012, by conducting another new evolutionary algorithm, which is the Electimize, in the same problem, Abdel-Raheem and

Khalafallah identified a better solution (18,713.14 ft). The improved objective function value presents the capability and effectiveness of the Adaptive Clonal Selection when it is applied to solve this particular complex construction optimization problems. The final site utilization figured by ACS is presented in FIGURE V.

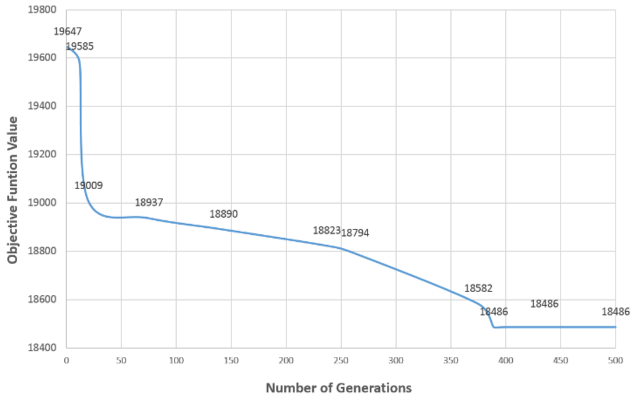

FIGURE IV
RESULT OF EXAMPLE APPLICATION BY ACS

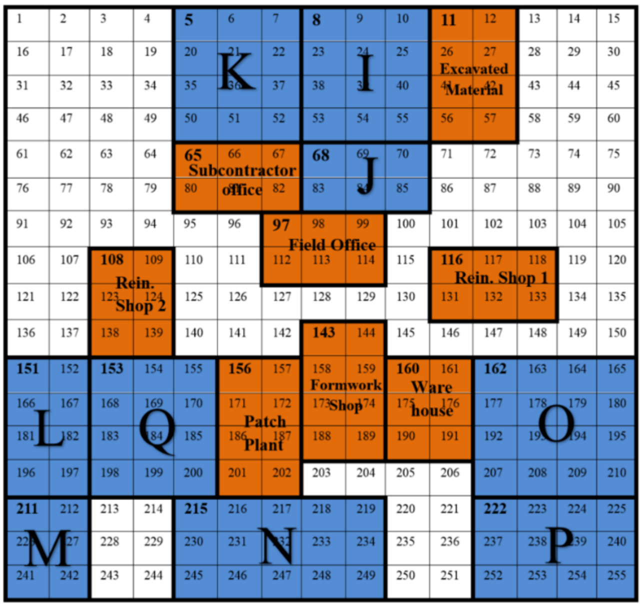

FIGURE V
FINAL SITE UTILIZATION PLANNING

5) Comparison with CSA: In order to verify the improvement of ACS, the CSA is applied to search the optimal layout planning in this case study using same initial population size (150) and the number of generations (500). Through assigning three value of ρ (5, 10 and 20) respectively and running 50 times for each, the average minimum objective function value is 20,200 (ft). FIGURE VI specifies a typical result after running the algorithm. The X axis represents the number of generation, and the Y axis represents the objective function value. According to observations, the CSA reaches the convergence after around 40 generations. The CSA computation speed is faster. The difference of running time between two algorithms is around 102s, but the CSA fails to find a better solution as the ACS does. At least in this case study, the ACS shows more robust searching capacity.

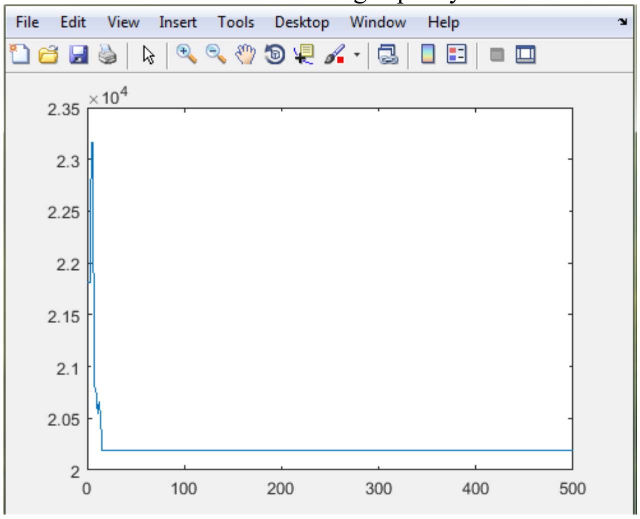

FIGURE VI
COMPUTATION RESULT BY CSA

III. Conclusion and Further Research

This paper proposes an improved Clonal Selection Algorithm (CSA), Adaptive Clonal Selection (ACS), for the construction site utilization planning (CSUP). The CSUP problem is formulated as a quadratic assignment problem with an objective function of minimizing the total traveling distance between facilities on site. Both the ACS and the CSA are applied to a case study in which feasible solutions are obtained by three different methodologies from literature. The test result preliminarily validates the feasibility and efficiency of the ACS in solving CSUP problems. The developed algorithm introduced several improvements that enhance the algorithm's capabilities and solutions in the following aspects:

1) The ACS successfully implements the Evolution Strategy (ES) to remove the parameter ρ and N. The selection of ρ and N could be essentially random and require large amount of sensitive analyses. The ES enables the algorithm to choose and update the value based on the mutation strengthen. Without external intervention, the operation process becomes more intelligent and has more rational results.

2) The ACS effectively controls the working scope while enhancing the algorithm capacity. The total number of the clones reproduced by selected layouts is equal to the total number of individuals in initial population. The population is able to keep refreshed after each generation without uneven distributions.

3) In a construction site layout planning process, the manager always assigns facilities to locations based on experience. The ACS is able to help provide more layout choices and evaluations in a reasonable time. More importantly is the ACS successfully implements the grid pattern site representation scheme, which permits users to

define and specify their preferences about placing directions, shapes, or sizes of facilities.

4) The string layout representation strategy of the ACS works well in layout planning optimization. In practice, it is a simple and effective method for recording and programming.

5) The ACS stores all potential solutions to the construction site layout plan and considers the total travel distance as an objective function, which is one of the most straightforward and effective criteria to evaluate a layout.

Although the result generated by the ACS in this case study is convincing, more cases studies need to be conducted to further validate the feasibility and efficiency of the ACS in CSUP optimization. Due to the lack of references, this study is the first attempt to apply the ACS to optimize the construction site layout planning. The main objective is to preliminarily test the capacity and limitations of ACS in construction layout planning applications. In the future, the ACS should be adopted and examined in real construction conditions rather than a predetermined virtual project from literature. Such an application should help answer how to enable the algorithm to adjust and perform well when more constraints are required on site? How to ensure the operation efficiency of the ACS if the site layout plans needs to be visualized by cooperating with other algorithms or tools? Also, more comparisons with other evolutionary algorithms are necessary. Although the parameter-free ACS is capable of searching for a better solution compared with the CSA, self-evolving parameters slow the algorithm running speed because it takes longer time to do self-examinations and modifications, Therefore, how to simplify the parameters self-adaption process could be another important topic for the future research.

REFERENCES

[1] Hamiani, A., and Popescu, C., “Consite: A Knowledge-Based Expert System for Site Layout.” *Computing in Civil Engineering*, 248–256, 1988.

[2] Hegazy, T., and Elbeltagi, E., “EvoSite: Evolution-Based Model for Site Layout Planning.” *Journal of Computing in Civil Engineering*, 13(3), 198–206, 1999.

[3] Mincks, W., and Johnston, H., Construction Jobsite Management. Cengage Learning, 2010.

[4] Li, H., and Love, P., “Site-Level Facilities Layout Using Genetic Algorithms.” *Journal of Computing in Civil Engineering*, 12(4), 227–231, 1998.

[5] Yeh, I., “Construction-Site Layout Using Annealed Neural Network.” *Journal of Computing in Civil Engineering*, 9(3), 201–208, 1995.

[6] Zouein, P. P., Harmanani, H., & Hajar, A., Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. *Journal of Computing in Civil Engineering*, 16(2), 143-151, 2002.

[7] Samdani, S. A., Bhakal, L., & Singh, A. K., Site layout of temporary construction facilities using ant colony optimization. In ASCE Los Angeles Section International Committee 4th International Engineering and Construction Conference at California State University, Fullerton on July (Vol. 28), 2006.

[8] Goss, S., Aron, S., Deneubourg, J.L. and Pasteels, J.M., Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76(12), pp.579-581, 1989.

[9] Alagarsamy, K., CONSITEPLAN–A Multi-Objective Construction Site Utilization Planning Tool (Doctoral dissertation, Auburn University), 2012.

[10] Kusiak, A. and Heragu, S.S., The facility layout problem. *European Journal of operational research*, 29(3), pp.229-251, 1987.

[11] Yang, X.S., Nature-inspired metaheuristic algorithms. Luniver press, 2010.

[12] Vasconcelos, J.A., Saldanha, R.R., Krähenbühl, L. and Nicolas, A., Genetic algorithm coupled with a deterministic method for optimization in electromagnetics. Magnetics, *IEEE Transactions on*, 33(2), pp.1860-1863, 1997.

[13] Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J., A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2), 239-287, 2009

[14] DasGupta, D., An overview of artificial immune systems and their applications , Springer Berlin Heidelberg, pp. 3-21, 1993.

[15] De Castro, L.N. and Timmis, J., Artificial immune systems: a new computational intelligence approach. Springer Science & Business Media, 2002.

[16] Hart, E. and Timmis, J., Application areas of AIS: The past, the present and the future. *Applied soft computing*, 8(1), pp.191-201, 2008.

[17] Timmis, J., Artificial immune systems—today and tomorrow. *Natural computing*, 6(1), pp.1-18, 2007.

[18] Burnet, S.F.M., The clonal selection theory of acquired immunity (Vol. 3). Nashville: Vanderbilt University Press, 1959.

[19] Hsieh, Y.C., You, P.S. and Liou, C.D., A note of using effective immune based approach for the flow shop scheduling with buffers. *Applied Mathematics and Computation*, 215(5), pp.1984-1989, 2009.

[20] Tsai, J.T., Ho, W.H., Liu, T.K. and Chou, J.H., Improved immune algorithm for global numerical optimization and job-shop scheduling problems. *Applied Mathematics and Computation*, 194(2), pp.406-424, 2007.

[21] Bagheri, A., Zandieh, M., Mahdavi, I. and Yazdani, M., An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems*, 26(4), pp.533-541, 2010.

[22] Zandieh, M., Ghomi, S.F. and Husseini, S.M., An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation*, 180(1), pp.111-127, 2006.

[23] Engin, O. and Döyen, A., A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future generation computer systems*, 20(6), pp.1083-1095, 2004.

[24] Masutti, T.A. and de Castro, L.N., A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Information Sciences*, 179(10), pp.1454-1468, 2009.

[25] Keko, H., Skok, M. and Skrlec, D., September. Artificial immune systems in solving routing problems. In EUROCON 2003. Computer as a Tool. The IEEE Region 8 (Vol. 1, pp. 62-66). IEEE, 2003,

[26] Harmer, P.K., Williams, P.D., Gunsch, G.H. and Lamont, G.B., An artificial immune system architecture for computer security applications. Evolutionary computation, *IEEE transactions on*, 6(3), pp.252-280, 2002.

[27] Mazhar, N. and Farooq, M., July. A sense of danger: dendritic cells inspired artificial immune system for manet security. In Proceedings of the 10th annual conference on Genetic and evolutionary computation , *ACM*, pp. 63-70, 2008.

[28] Le Boudec, J.Y. and Sarafijanović, S., An artificial immune system approach to misbehavior detection in mobile ad hoc networks. In Biologically Inspired Approaches to Advanced Information Technology (pp. 396-411). Springer Berlin Heidelberg, 2004.

[29] Coello, C.A.C. and Cortés, N.C., Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2), pp.163-190, 2005.

[30] Tan, K.C., Goh, C.K., Mamun, A.A. and Ei, E.Z., An evolutionary artificial immune system for multi-objective optimization. European *Journal of Operational Research*, 187(2), pp.371-392, 2008.

[31] Freschi, F. and Repetto, M., Multiobjective optimization by a modified artificial immune system algorithm. In Artificial Immune Systems (pp. 248-261). Springer Berlin Heidelberg, 2005.

[32] de Franca, F.O., Von Zuben, F.J. and de Castro, L.N., June. An artificial immune network for multimodal function optimization on dynamic environments. In Proceedings of the 7th annual conference on Genetic and evolutionary computation, *ACM*, pp. 289-296, 2005,

[33] Aydin, I., Karakose, M. and Akin, E., A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Applied Soft Computing*, 11(1), pp.120-129, 2011.

[34] Yap, D.F., Koh, S.P., Tiong, S.K. and Prajindra, S.K., Particle swarm based artificial immune system for multimodal function optimization and engineering application problem. *Trends in Applied Sciences Research*, 6(3), p.282, 2011.

[35] Yildiz, A.R. and Solanki, K.N., Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach. *The International Journal of Advanced Manufacturing Technology*, 59(1-4), pp.367-376, 2012.

[36] Ibrahim, A.A., Mohamed, A., Shareef, H. and Ghoshal, S.P., June. Optimal power quality monitor placement in power systems based on particle swarm optimization and artificial immune system. In Data Mining and Optimization (DMO), 2011 3rd Conference on (pp. 141-145). IEEE, 2011.

[37] Kuo, R.J., Tseng, W.L., Tien, F.C. and Liao, T.W., Application of an artificial immune system-based fuzzy neural network to a RFID-based positioning system. *Computers & Industrial Engineering*, 63(4), pp.943-956, 2012.

[38] Chikh, M.A., Saidi, M. and Settouti, N., Diagnosis of diabetes diseases using an artificial immune recognition system2 (AIRS2) with fuzzy k-nearest neighbor. *Journal of medical systems*, 36(5), pp.2721-2729, 2012.

[39] Shamshirband, S., Anuar, N.B., Kiah, M.L.M., Rohani, V.A., Petković, D., Misra, S. and Khan, A.N., Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. *Journal of Network and Computer Applications*, 42, pp.102-117, 2014.

[40] Ulutas, B.H. and Islier, A.A., A clonal selection algorithm for dynamic facility layout problems. *Journal of Manufacturing Systems*, 28(4), pp.123-131, 2009.

[41] Ulutas, B.H. and Kulturel-Konak, S., An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), pp.5384-5395, 2012.

[42] M. Burnet, "Auto-immune Disease," *British Medical Journal*, 2(5153), pp. 645–650, 1959, 1959.

[43] Medzhitov, R., and Janeway Jr, C. A., "Innate immune recognition and control of adaptive immune responses." *Seminars in Immunology*, 10(5), 351–353, 1998.

[44] Rechenberg, I., Evolution Strategy: Optimization of Technical systems by means of biological evolution. *Fromman-Holzboog*, Stuttgart, 104, 1973.

[45] Rechenberg, I., Evolution strategy. Computational intelligence: *Imitating life*, 1, 147-159, 1994.

[46] Beyer, H. G., and Schwefel, H. P., "Evolution Strategies, A comprehensive introduction." *National Computing*, 1, 3-52, 2002.

[47] Mawdesley, M., Al-jibouri, S., and Yang, H., "Genetic Algorithms for Construction Site Layout in Project Planning." *Journal of Construction Engineering and Management*, 128(5), 418–426, 2002.

[48] Lee, H., "Integrating Simulation and Ant Colony Optimization to Improve the Service Facility Layout in a Station." *Journal of Computing in Civil Engineering*, 26(2), 259–269, 2012.

[49] Lam, K., Ning, X., and Ng, T., "The application of the ant colony optimization algorithm to the construction site layout planning problem." *Construction Management and Economics*, 25(4), 359–374, 2007.

[50] Zhang, J. P., Liu, L. H., and J, R., "Hybrid intelligence utilization for construction site layout." *Automation in Construction*, 11(5), 511–519, 2002.

[51] Tsuchiya, K., Bharitkar, S., and Takefuji, Y., "A neural network approach to facility layout problems." *European Journal of Operational Research*, 89(3), 556–563, 1996.

[52] Abdel-Raheem, M., and Khalafallah, A., "Application of Electimize in Solving the Construction Site Layout Planning Optimization Problem." *Construction Research Congress*, ASCE, 2012.

[53] Lien, L. C., & Cheng, M. Y., A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Systems with Applications*, 39(10), 9642-9650, 2012.

[54] Rodriguez-Ramos, W. E., ‘‘Quantitative techniques for construction site layout planning,’’ PhD thesis, University of Florida, Gainesville, Fla, 1982.

[55] Garrett, S. M., "Parameter-free, adaptive clonal selection," *Evolutionary Computation*, (1), 1052-1058, 2004.

[56] Wang, X., Deshpande, A. S., Dadi, G. B., & Salman, B., Application of Clonal Selection Algorithm in Construction Site Utilization Planning Optimization. *Procedia Engineering*, (145), 267-273, 2016.