pISSN 1598-2033
eISSN 2233-5706
Journal of the Korea Institute of Building Construction, Vol. 12, No. 6
http://dx.doi.org/10.5345/JKIBC.2012.12.6.664
www.jkibc.org

# Optimization Algorithms for Site Facility Layout Problems Using Self-Organizing Maps

Park, U-Yeol[1] An, Sung-Hoon[2*]
*Department of Architecture Engineering, Andong National University, Andong-Si, Kyoungsangbuk-Do, 760-749, Korea [1]*
*Department of Architectural Engineering, Daegu University, Gyeongsan-Si, Kyoungsangbuk-Do, 706-140, Korea [2]*

Abstract

Determining the layout of temporary facilities that support construction activities at a site is an important planning activity, as layout can significantly affect cost, quality of work, safety, and other aspects of the project. The construction site layout problem involves difficult combinatorial optimization. Recently, various artificial intelligence(AI)-based algorithms have been applied to solving many complex optimization problems, including neural networks(NN), genetic algorithms(GA), and swarm intelligence(SI) which relates to the collective behavior of social systems such as honey bees and birds. This study proposes a site facility layout optimization algorithm based on self-organizing maps(SOM). Computational experiments are carried out to justify the efficiency of the proposed method and compare it with particle swarm optimization(PSO). The results show that the proposed algorithm can be efficiently employed to solve the problem of site layout.

Keywords : temporary facilities, layout, self-organizing maps, optimization

## 1. Introduction

Planning the site layout of temporary facilities in a construction project is a very important at the construction planning phase. This temporary construction plan is carried out using an heuristic method based on the planner's experience and insight. It has repeatedly been reported that a poor construction plan can cause productivity loss, frequent relocation of temporary facilities, and delay in finishing work[1].

In particular, since the lifting equipment including tower crane that transports materials and workers vertically has a direct impact on construction duration, safety management, and construction cost, studies have been actively conducted to optimize the selection of equipment and location[2,3].

The site layout problem of temporary facilities including lifting equipment is an NP–hard (Non–Polynomial hard), and the complexity of the problem increases exponentially as the number of temporary facilities is increased[4,5]. If the number of facilities is n, there are n! possible solutions. For instance, if there are 10 facilities, there are 3,628,000 possible solutions. Therefore, numerical optimization for this problem did not catch the attention of researchers due to inefficiency of calculation[6]. However AI–based algorithms including heuristic method, neural networks and genetic algorithms(GA) have been actively studied.

GA, a probabilistic detection method that mimics the theory of neutral selection and the laws of heredity in the evolutionary process of a living

Received : September 17, 2012
Revision received : October 9, 2012
Accepted : October 16, 2012
* Corresponding author : An, Sung-Hoon
[Tel: 82-53-850-6518, E-mail: shan7208@hanmail.net]

organism, is featured to detect the solution space using potential multiple populations[7]. GA can be easily applied when an object function is defined, and it has been applied in many studies[7,8,9,10,11]. However, GA has a problem of local minima[4], and there have been some studies conducted to resolve this problem. Lam et al.[12] introduced max–min ant system(MMAS), a kind of ant colony algorithm(ACO) to resolve the problem of randomly creating the populations at the early stage of GA, and Lam el al.[13] also presented an improved algorithm.

The swarm intelligence(SI) algorithm that imitates the collective behavior of bees and birds has seen wide application. The particle swarm optimization(PSO) algorithm, a representative swarm intelligent algorithm, is an algorithm introduced to the optimization process that works on the assumption that like groups of birds, bees and fishes, when a group moves, they utilize the experience that is socially shared and their own experience as well. Zhang and Wang[5] applied the PSO to the location optimization of temporary facilities, and found that it was possible to detect the optimal solution more effectively compared to GA. Lien and Cheng[14] presented a new algorithm by combining the bee algorithm(BA) and PSO, and proved that it provided a better solution than BA (simulation of food foraging of honey bees) or PSO (simulation of bird flocking). Lam et al.[4] applied the ACO (simulation of food foraging behaviors of ants), and Ling et al.[15] proposed an MMAS algorithm as a multi–object optimization method considering safety and the environment. Likewise, algorithms simulating SI provide a better solution compared to GA, but the PSO algorithm has some drawbacks, such as early convergence to local minima[14]. In addition, the optimal parameter should be set when using ACO, which is noted as another disadvantage[4]. Therefore, a new algorithm is still needed to stably provide an optimal solution for the site layout problem without being prematurely converged to a sub–optimal solution.

With respect to the optimization problems, studies using self–organizing maps(SOM) have caught attention. SOM is modeled on the biological nervous systems of a living organism, particularly the information processing of the brain, and the Kohonen SOM(KSOM) algorithm has been widely applied[16]. Since its application to the traveling salesman problem, a optimization problem[17], there have been various studies performed to improve the performance of the algorithm[18,19,20]. In a construction–related study, when SOM and PSO algorithms were applied to the optimization of the order of construction works to minimize the construction time required to build a scant pile wall, SOM reduced the risk of being converged to local minima at an initialization stage, and provided a better solution compared to PSO[21], [22].

Therefore, this paper aims to propose an SOM–based algorithm for the optimization of the site layout of temporary facilities to be utilized as a support tool in decision making at a construction project planning phase.

## 2. Construction site layout planning

To optimize a site layout of temporary facilities, various variables should be considered and reflected.

First, it should be taken as dynamic problems[23]. If any of the requirements are changed during construction work, the site layout of the temporary facilities should be changed, accordingly. Studies in which the problem was considered dynamically are Ning et al.[23] and Sanad et al.[24]. Souein & Tommelein[25] presented an algorithm that calculated a solution gradually with the stage of construction by combining the heuristic method and the linear planning method. In addition, Elbeltagi et al.[26] suggested a GA model that supported a temporary construction plan needed at a different points of time in the course of the construction process in combination with a process management tool.

Second, the site layout of temporary facilities can be considered a problem of multi-object optimization. Many studies have been done to resolve the problem of single object optimization, such as in the installation cost of temporary facilities or the maintenance cost of the facilities, and the cost and frequency of transport depending on temporary facilities. However, in reality multi-object optimization is needed to deal with noise and security, as well as to minimize costs, ensure safety in the working environment, promote the efficiency of material transport, and secure the required quality[23]. El-Rayes et al.[27] presented a multi-object optimization model for minimizing transport cost and maximizing safety using GA, and Sanad et al.[24] dealt with the site layout problem using GA after reflecting safety and environmental problems in the object function. Ning et al.[15] reviewed multi-object optimization considering safety and the environment as separate object functions. In addition, Easa and Hossain[6] presented a multi-object optimization method considering physical and functional restrictions to various environments and various land shapes.

Similarly, to optimize the site layout of temporary facilities, various facets should be considered. As mentioned earlier, many studies are currently being conducted to improve the conventional algorithms. Therefore, this research focuses on developing a more effective algorithm.

In this research, we reviewed the SOM algorithm, presented an optimized algorithm appropriately modified to the site layout of temporary facilities, and verified the applicability of the algorithm through a comparative analysis of the algorithm with the previous models by applying it to a hypothetical site.

## 3. Self-Organizing Maps

### 3.1 Kohonen SOM

The SOM presented in this study is based on the unsupervised learning in a neural network presented by Kohonen[15]. SOM is modeled on biological nervous systems, particularly brain information processing, and creates a feature map by mapping multi-dimensional data into two-dimensional data. Thanks to its simple applicability, KSOM has been widely applied in optimization problems, robotics and control, function approximation, estimation and evaluation[28].

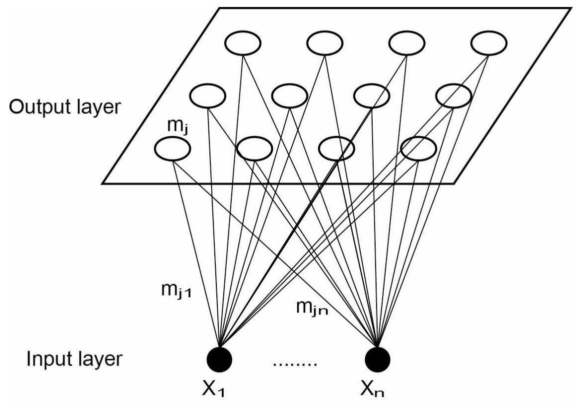


Figure 1. Typical architecture of the SOM network

As shown in Figure 1, SOM generally consists of an input layer and an output layer. The output layer is arranged in two dimensions for visual expression. Output neuron is connected with input neuron at a certain weight, and expressed with a vector $m_j(t)$ identical to one on the input layer. The size of the neuron can be changed depending on the problem concerned. The neurons can also be arranged in the hexagonal form.

An SOM learning is carried out in the winner–takes–all manner, and it uses the Euclidian distance $\| \mathrm{x} - m_j \|$. To do this, a winning neuron $j^*$ which has the minimum distance is found.

$$j^* = \arg\min \| \mathrm{x}(t) - \mathrm{m_j}(t) \| \quad \text{-----} \quad (1)$$

The connection weight of the winning neuron that satisfies Eq. (1) and neighboring neurons can be modified using Eq. (2).

$$m_j(t+1) = m_j(t) + \alpha(t) h_{j^*j}(t)[\mathrm{x}(t) - \mathrm{m_j}(t)] \quad \text{--} \quad (2)$$

$h_{j^*j}(t)$ in Eq. (2) is a topological neighborhood function and is used in determining a certain neighbor distance within which learning is designed to take place. Generally, a Gaussian function is used, as indicated in Eq. (3), and the value grows with the passage of time. That is, at the beginning of learning, neuron information is searched in a broad area, but as time passes, it is switched to search in a more specific area.

$$h_{j^*j}(t) = \exp\left(-\frac{d_{j^*j}}{2\delta^2(t)}\right) \quad \text{-----} \quad (3)$$

Here, $d_{j^*j}$ is a topological distance from the winning neuron $j^*$.

In addition, $\alpha(t)$ in Eq.(2) is a learning rate, and the value was set to be high at the beginning of learning but becomes smaller with the passage of time.

In SOM, learning continues to be repeated as long as the winning neuron and the neighboring neuron do not satisfy certain termination criteria, or the learning can be repeated by predetermining the number of iterations.

### 3.2. SOM optimization algorithm

As explained in the previous chapter, the basic SOM algorithm is used to transform patterns of arbitrary dimensionality into the response of one– or two–dimensional arrays of neurons, and it is appropriate for classification problems like clustering[16]. However, to apply it to an optimization problem, the basic algorithm must be modified. In this study, an SOM optimization(SOMO) algorithm was composed based on the research by Su and Zhao[29] and Milano et al.[20]. The algorithm of Su and Zaho[29] is found to have drawn a better result than the PSO algorithm in determining the order of installation works of a scant pile wall. Based on the previous studies, fitness value was determined and the mutation operator and connection weight of a learning algorithm were modified and applied in this study.

#### 3.2.1 Determination of fitness value

In an existing SOM, the neuron that has the minimum Euclidian distance was selected as the winning neuron. However, in an optimization problem an object function should be optimized, and the method of selecting the winning neuron should be modified. First of all, to calculate the fitness value, the connection weight of individual neurons should be modified in a manner appropriate for the layout optimization of temporary facilities. The priority–based presentation method has been generally applied in previous studies related to optimization problems [5,21,22]. For this reason, the priority–based presentation method was also applied to this study.

As illustrated in Figure 2, the connection weight between input neuron and output neuron was set between 0 and 1, and considered as representing the priority value of individual facility. That is, input neurons $\mathrm{X}_1$, $\mathrm{X}_2$ ,were considered as temporary facilities, and $\mathrm{X}_2$is placed before $\mathrm{X}_1$ if $\mathrm{m}_{j1}$ is smaller than $m_{j2}$ by comparing the connection weights. After placing each facility

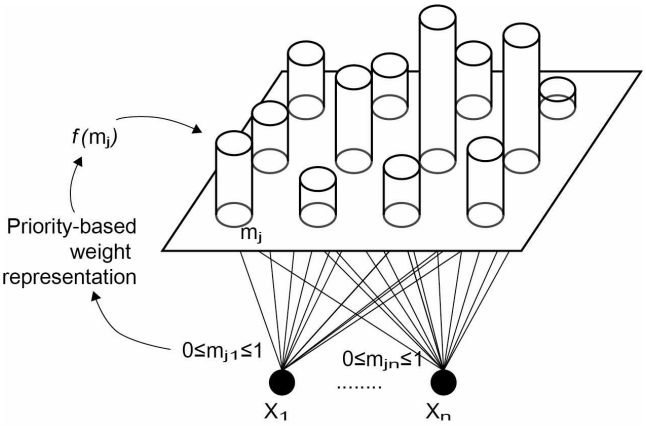

Figure 2. Priority-based weight representation

according to priority, the fitness value is calculated.

As described in the research scope, this study aims to develop an algorithm, and total distance is used in the optimization function as in previous studies[7,14].

$$j^{*}=\arg\min TD=\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\delta_{ki}f_{ki}d_{ij} \quad --- (4)$$

subject to $\sum_{k=1}^{n}\delta_{ki}=1,\ i=1,2,\ldots,n$

Here, the number of temporary facilities is expressed as $n$, permutation matrix as $\delta_{ki}$, travel frequency as $f_{ij}$, distance between facilities as $d_{ij}$.

Eq. (5) is a permutation equation. For example, when there are three temporary facilities to be placed (1,2,3) at the places (B,C,A), the equation can be expressed as follows.

$$\begin{array}{cc} & A,B,C \\ \begin{array}{c}1\\2\\3\end{array} & \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{pmatrix}\end{array} \quad ----(5)$$

### 3.2.2 Mutation operator

Upon initialization, the SOMO algorithm generally randomizes the value using random values. However, the initial connection weight greatly affects the convergence of the network and the mapping direction[28]. In addition, in an optimization problem, to get the optimal solution for the object function, it needs to search a broad area. Therefore, based on the previous study[30], a mutation operator is used to introduce an evolution algorithm that searches the solution space.

When the mutation operator is selected in pure random search manner, efficiency cannot be secured, and a simplex that consists of part of the SOM network is used to select highly probable candidates. Figure 3 shows the concept of subdivision of a 2D SOM into simplexes. The network composed of a 3x4 permutation matrix is divided by triangles into 12 simplexes.

A simplex is randomly selected and a connection weight is created randomly within the range of connection weight of the three neurons in this algorithm. Through this process, when a next-generation mutation is selected, an optimal solution can be searched based on the neuron learned.

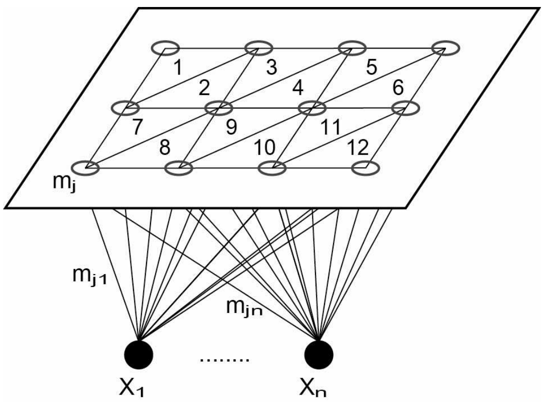

Figure 3. Subdivision of a 2D SOM into simplexes

### 3.2.3 Learning by connection weight

In an optimization problem, training data required for the neutral network learning cannot be applied, since there is no input vector. Therefore, to resolve this kind of problem, the result value is calculated by substituting the connection weight created by the mutation operator in the objective function, and then compared with the previous value. At this time, if the value is higher than the previous one, it is considered as the winning neuron, and learning is determined.

The connection weight at every point of time is renewed using the following equation.

$$m_j(t+1) = m_j(t) + \alpha h_{j^*j}[\mathrm{m}_{j^*}(t) - \mathrm{m}_j(t)] \quad ---(6)$$

$$\text{Here, } h_{j^*j} = \begin{cases} 1, j \in N_w \\ 0, otherwise \end{cases} \quad ------ (7)$$

$\alpha$ is a learning rate, which is fixed at a value within the normal range, or becomes smaller according to the number of iterations. Here, the learning rate was set at 0.2 based on the previous study[30].

Eq. (7) is used to determine the neighbor distance $h_{j^*j}$. $N_w$ can be defined as the nearest neighbor group. The neurons linked to the selected simplex at a given time are set as the group. Therefore, the connection weight is modified based on the neurons directly linked to the given simplex.

#### 3.2.4 SOMO adaptation procedure

The application order of the SOMO algorithm is shown in Figure 4. To initialize the program, output neuron is determined and the connection weight linking output and input neurons is initialized. Then, to determine fitness value, the connection weight is created using a mutation operator, and the connection weight obtained is substituted in the function to get the fitness value.

Using a new mutation operator, the connect weight is created to calculate the fitness value. The fitness value is compared with one obtained previously.

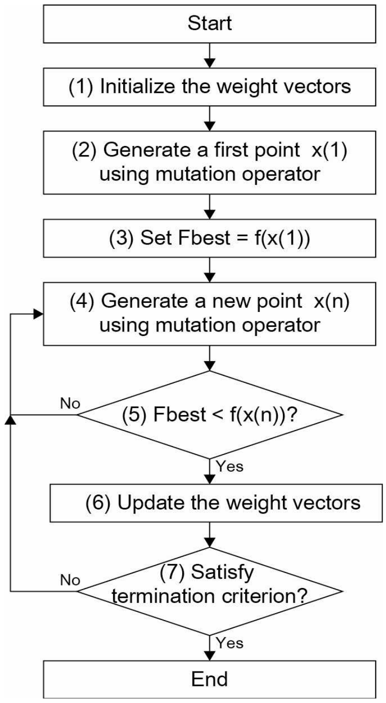


Figure 4. SOM optimization flowchart

In an optimization problem, if the fitness value is smaller than the previous one, a new mutation is created. Otherwise, the connection weight of the neurons directly connected to the simplex is modified. Using Eq. (6), the connection weight modification equation, connection weight is modified to make the neuron closest to the winning neuron exactly mimic the winning neuron.

The process continues to be repeated, and the optimization function value is converged within a certain range or forcibly terminated after predetermined number of iterations.

## 4. Case study

### 4.1 Hypothetical site

To verify the algorithm suggested in this study, the algorithm was applied to a hypothetical site in a previous study[14]. The 7-story reinforced concrete structure is a school building, and is placed as shown in Figure 5.

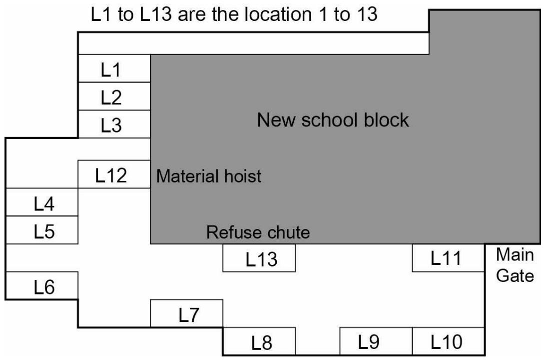


Figure 5. Hypothetical construction site layout

As indicated in Table 1, a total of 9 facilities need to be placed optimally: site office, debris storage area, reinforcement bending/storage yard, carpentry workshop and store, labor hut, materials storage area, main gate, materials hoist, and refuse chute. The relative location of the entrance affects the site layout of temporary facilities, and is very important from the perspective of materials transport and approach. However, the location of the entry is determined before construction, and is rarely changed. Therefore, the main entrance is treated as special facilities fixed at the location determined in advance[7]. Here, the main gate, material hoist, and refuse chute are treated as fixed, as illustrated in Figure 5.

To calculate the fitness value using Eq. (4), the values of travel distance between facilities and travel frequency are required. Here, as in the previous study, the values set in advance were used in this study to calculate the fitness value as shown in Tables 2 and 3. The building used in the case is one used in a previous study, and it is considered that the validity of the algorithm can be sufficiently verified.

### 4.2 Analysis of result

To examine the validity of the SOMO algorithm proposed in this study, the basic PSO algorithm used in the previous study[5] was applied. In the previous study, the basic PSO algorithm was modified, but here the basic PSO algorithm was not modified in order to perform a comparison. In the PSO, the result value can vary depending on the parameters created at the beginning, including population size. The larger the population, the wider the solution space. For this reason, 100 larger than the value applied in the previous study was used, and other parameters are shown in Table 4.

Table 1. Facilities used in the case study

| Facility No. | Facility Name | Note |
|---|---|---|
| A | Site office | – |
| B | Debris storage area | – |
| C | Reinforcement bending/storage yard | – |
| D | Carpentry workshop and store | – |
| E | Labor hut | – |
| F | Materials storage area | – |
| G | Main gate | Fixed |
| H | Materials hoist | Fixed |
| I | Refuse chute | Fixed |

Table 2. Travel distance between facilities

| Distance | | Location | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| L | 1 | 0 | 1 | 2 | 6 | 7 | 9 | 12 | 14 | 16 | 17 | 13 | 4 | 9 |
| o | 2 | 1 | 0 | 1 | 5 | 6 | 8 | 11 | 13 | 15 | 16 | 12 | 3 | 8 |
| c | 3 | 2 | 1 | 0 | 4 | 5 | 7 | 10 | 12 | 14 | 15 | 11 | 2 | 7 |
| a | 4 | 6 | 5 | 4 | 0 | 1 | 3 | 7 | 9 | 11 | 12 | 9 | 2 | 5 |
| t | 5 | 7 | 6 | 5 | 1 | 0 | 2 | 6 | 8 | 10 | 11 | 8 | 3 | 4 |
| i | 6 | 9 | 8 | 7 | 3 | 2 | 0 | 3 | 5 | 7 | 8 | 8 | 5 | 4 |
| o | 7 | 12 | 11 | 10 | 7 | 6 | 3 | 0 | 2 | 4 | 5 | 7 | 6 | 3 |
| n | 8 | 14 | 13 | 12 | 9 | 8 | 5 | 2 | 0 | 2 | 3 | 5 | 8 | 3 |
| | 9 | 16 | 15 | 14 | 11 | 10 | 7 | 4 | 2 | 0 | 1 | 3 | 11 | 6 |
| | 10 | 17 | 16 | 15 | 12 | 11 | 8 | 5 | 3 | 1 | 0 | 2 | 12 | 7 |
| | 11 | 13 | 12 | 11 | 9 | 8 | 8 | 7 | 5 | 3 | 2 | 0 | 9 | 5 |
| | 12 | 4 | 3 | 2 | 2 | 3 | 5 | 6 | 8 | 11 | 12 | 9 | 0 | 4 |
| | 13 | 9 | 8 | 7 | 5 | 4 | 4 | 3 | 3 | 6 | 7 | 5 | 4 | 0 |

Table 3. Frequency of trips between facilities

| Frequency | | Facility | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Facility | 1 | 0 | 3.11 | 4.79 | 4.94 | 5.15 | 5.41 | 6.34 | 3.48 | 2.55 |
| | 2 | 3.11 | 0 | 3.69 | 3.71 | 3.7 | 3.36 | 4.42 | 3.07 | 5.85 |
| | 3 | 4.79 | 3.69 | 0 | 4.27 | 4 | 4.4 | 5.65 | 6.26 | 2.03 |
| | 4 | 4.94 | 3.71 | 4.27 | 0 | 4.51 | 4.58 | 5.14 | 6.2 | 2.24 |
| | 5 | 5.15 | 3.7 | 4 | 4.51 | 0 | 4.99 | 4.39 | 4.13 | 2.48 |
| | 6 | 5.41 | 3.36 | 4.4 | 4.58 | 4.99 | 0 | 5.24 | 6.2 | 2.65 |
| | 7 | 6.34 | 4.42 | 5.65 | 5.14 | 4.39 | 5.24 | 0 | 4.62 | 3.75 |
| | 8 | 3.48 | 3.07 | 6.26 | 6.2 | 4.13 | 6.2 | 4.62 | 0 | 2.37 |
| | 9 | 2.55 | 5.85 | 2.03 | 2.24 | 2.48 | 2.65 | 3.75 | 2.37 | 0 |

Table 4. The result of SOMO and PSO

| | Mean | Std. | Min | Note |
|---|---|---|---|---|
| SOM | 8440.9 | 0.45 | 843.94 | Average number of weight update = 11.6 |
| PSO | 847.72 | 3.41 | 843.94 | Population Size = 100<br>inertia weight = 1<br>learning factor = 1 |

The SOMO algorithm applied in this study was terminated when the fitness value was less than a certain value or after the learning had been repeated a certain number of iterations. Here, the algorithm was set to be terminated when a mutation operator was applied 1000000 times.

The test was repeated 30 times, and the test results are shown in Table 4. The minimum value of test results of the PSO algorithm was 843.94 in 27 times, except for 3 times. The two algorithms could not be directly compared with this test, but the standard distribution of the result values was higher in PSO, from which it can be interpreted that it poses a higher risk of being converged to local minima. On average, the connection weight was renewed 11.6 times until the SOMO algorithm reached the minimum value. The fitness value created at the early phase affected the number of iterations required to reach the optimal solution. Here, to identify the basic performance of the algorithm, the algorithm was set to be terminated at a certain number of iterations of a mutation operator. The algorithm is believed to reach the optimal solution even when terminated at the specific number of renews of connection weight or at the fitness value that satisfies a certain function value depending on the problem to be resolved.

Figure 6 shows the comparison of changes in the optimal solution between the two algorithms according to repetition. Here, the solutions of the SOMO algorithm are shown based on the number of renewals of connection weight. The PSO algorithm is basically set to mimic the best neuron, and the final result value is greatly affected by the fitness value of the population created at the beginning. Figure 6 illustrates that the fitness value of the best neuron was not improved, and converged to local minima even though the operation was repeated in the PSO algorithm.

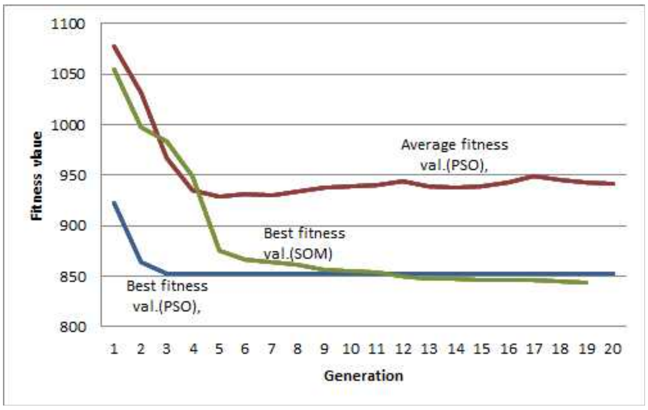


Figure 6. Performance comparison between SOMO and PSO

Unlike the PSO, the optimal value of SOMO was continuously renewed. It is believed that the mutation operation is performed based on the connection weight of the output neuron in the SOMO algorithm applied in this study. The algorithm applied in this study calculates candidate solutions based on a specific neuron of the simplex that is randomly selected when a mutation operation was performed. Therefore, as the operation is continuously repeated, the output fitness value of the SOMO algorithm is gradually improved.

## 5. Conclusion

This study aims to present an effective algorithm applicable to an optimization problem such as site layout problem of temporary facilities required at the construction planning phase. In relation to this, AI-based algorithms, including NN, GA, and PSO, have currently been studied actively, and the advantages and disadvantages of each algorithm have also been studied. In this study, an SOM-based optimization algorithm for the site layout plan of temporary facilities was presented.

The basic SOM algorithm was modified, and the method to determine fitness value, application of a mutation operator, and learning by connection weight were presented, and the algorithm was verified by applying it to a hypothetical case. Through the test, the SOMO presented in this study was verified to reduce the risk of being converged to local minima, which is known as one of the problems of the PSO algorithm. Therefore, the SOM algorithm is believed to be appropriate to use for making a decision on the site layout plan of temporary facilities based on the results of an objective analysis. In addition, it is considered applicable to other optimization problems.

The scope of this research is limited to developing an effective algorithm. The site layout plan of temporary facilities has multi-object optimization solutions that satisfy dynamic characteristics as well as diverse purposes. Therefore, the SOMO algorithm should be studied in subsequent research, to determine whether it can be effectively applied to multi-object problems or dynamic problems.

## References


1. Chun JY, Oh SJ. A Study on the Computerization of Knowledge Based Site Layout System in High-rise Building Construction Plan. Journal of the Architectural Institute of Korea(Structure & Construction Section). 2003 Jan;19(1):129-36.
2. Lee UK, Kim JY, Seo DK, Kang KI. Genetic Algorithms of Planning Transportation. Journal of the Architectural Institute of Korea(Structure & Construction Section). 2004 Dec;20(2):127-34.
3. Lee HS, Chae HD, Jang MH. Development of Tower Crane Planning Process. Journal of the Architectural Institute of Korea(Structure & Construction Section). 2002 Jun;18(6):119-26.
4. Lam KC, Ning X, NG T. The Application of the Ant Colony Optimization Algorithm to the Construction Site Layout Planning Problem. Construction Management and Economics. 2007 Apr;25(4):359-74.
5. Lien LC, Cheng MY. A Hybrid Swarm Intelligence Based Particle-Bee Algorithm for Construction Site Layout Optimization. Expert Systems with Applications. 2012 Aug;39(10):9642-50.
6. Easa SM, Hossain KMA. New Mathematical Optimization Model for Construction Site Layout. Journal Construction Engineering and Management. 2008 Aug;134(8):653-62.
7. Li H, Love PE. Site-Level Facilities Layout Using Genetic Algorithms. Journal of Computing in Civil Engineering. 1998 Oct;12(4):227-31.
8. Hagazy T, Elbeltagi EvoSite: Evolution-Based Model for Site Layout Planning. Journal of Computing in Civil Engineering. 1999 Jul;13(3):198-206.
9. Zouein PP, Hajar HH. Genetic Algorithm for Solving Site Layout Problem with Unequal-Size and Constrained Facilities. Journal of Computing in Civil Engineering. 2002 Apr;16(2):143-51.
10. Mawdesley MJ, Al-jibouri SH, Yang H. Genetic Algorithms for Construction Site Layout in Project Planning. Journal of Construction Engineering and Management. 2002 Oct;128(5):418-26.
11. Wong CK, Fung WH, Tam CM. Comparison of Using Mixed-Integer Programming and Genetic Algorithms for Construction Site Facility Layout Planning. Journal of Construction Engineering and Management. 2010 Oct;136(10):1116-28.
12. Lam KC, Ning X, Lam MCK. Conjoining MMAS to GA to Solve Construction Site Layout Planning Problem. Journal of Construction Engineering and Management. 2009 Oct;135(10):1049-57.
13. Lam KC, Tang CM, Lee WC. Application of the Entropy Technique and Genetic Algorithms to Construction Site Layout Planning of Medium-Size Projects. Construction Management and Economics. 2005 Feb;23(2):127-45.
14. Lien LC, Cheng MY. A Hybrid Swarm Intelligence Based

Particle-Bee Algorithm for Construction Site Layout Optimization. Expert Systems with Applications. 2012 Aug;39(10):9642-50.
15. Ning X, Lam KC, Lam MCK. Dynamic Construction Site Layout Planning Using Max-Min Ant System. Automation in Constrution. 2010 Jan;19(1):55-65.
16. Kohonen T. Self-Organizing Maps. Germany: Springer; 1995
17. Angéniol B, Vaubois GDLC, Texier JL. Self-Organizing Feature Maps and the Travelling Savelesman Problem. Neurla Networks. 1988 Apr;1(4):289-93.
18. Leung KS, Jin HD, Xu ZB. An Expanding Self-Organizing Neural Network for the Traveling Salesman Problem. Neurocomputing. 2004 Dec;62:267-92.
19. Faigl J, Kulich M, Vonásek V, Preucl. An Application of the Self-Organizing Map in the Non Euclidean Traveling Salesman Problem. Neurocomputing. 2011 Feb;74(5):671-79.
20. Faigl J. On the Performance of Self-Organizing Maps for the Non-Euclidean Traveling Salesman Problem in the Polygonal Domain. Information Sciences. 2011 Oct;181(19):4414-29.
21. Chen JH, Yang LR, Su MC. Comparison of SOM-Based Optimization and Particle Swarm Optimization for Minimizing the Construction Time of a Secant Pile Wall. Automation in Construction. 2009 Oct;18(6):844-8.
22. Chen JH, Su MC, Huang DY. Application of a SOM-Based Optimization Algorithm in Minimizing Construction Time for Secant Pile Wall. Journal of Construction Engineering and Management. 2010 Nov;136(11):1189-95.
23. Ning X, Lam KC, Lam MCK. A Decision-Making System for Construction Site Layout Planning. Automation in Construction 2011 Jul;20(4):459-73.
24. Sanad HM, Ammar MA, Ibrahim ME. Optimal Construction Site Layout Considering Safety and Environment Aspects. Journal of Construction Engineering and Management. 2008 Jul;134(7):536-44.
25. Zouein PP, Tommelein ID. Dynamic Layout Planning Using a Hybrid Incremental Solution Method. Journal of Construction Engineering and Management. 1999 Dec;125(6):400-8.
26. Elbeltagi E, Hegazy T, Hosny AH, Eldosouky A. Schedule-Dependent Evolution of Site Layout Planning. Construction Management and Economics. 2001 Nov;19(7):689-97.
27. El-Rayes K, K-halafallah A. Trade-off between Safety and Cost in Planning Construction Site Layouts. Journal of Construction Engineering and Management. 2005 Nov;131(11):1186-95.
28. Nour MA, Madey GR. Heuristic and Optimization Approaches to Extending the Kohonen Self Organizing Algorithm. European Journal of Operational Research. 1996 Sep;93(2):428-48.
29. Su MC, Zhao YX. A Variant of the SOM Algorithm and its Interpretation in the Viewpoint of Social Influence and Learning. Neural Computing and Applications. 2009 Nov;18(8):1043-55.
30. Milano M, Koumoutsakos P, Schmidhuber J. Self-Ogranizing Nets for Optimization. IEEE Transactions on Neural Netwroks. 2004 May;15(3):758-65.