CONSTRUCTION-SITE LAYOUT USING ANNEALED NEURAL NETWORK

By I-Cheng Yeh[1]

ABSTRACT: Construction-site layout is an important construction planning activity. The impact of good layout practices on money and timesaving becomes more obvious on larger construction projects. In this study, we formulate the problem as a combinatorial optimization problem. Construction-site layout is delimited as the design problem of arranging a set of predetermined facilities on a set of predetermined sites, while satisfying a set of constraints and optimizing an objective. In this paper, the annealed neural network model, which merges many features of simulated annealing and the Hopfield neural network is employed to solve the problem, and a program written in C, called SitePlan, is built on a personal computer to implement the algorithm. In addition, a strategy to set a reasonable initial temperature in the simulated annealing procedure is proposed, the effects of various parameters in annealed neural network are examined, and two case studies are used to illustrate the practical applications and to demonstrate this model's efficiency in solving the construction-site layout problem.

INTRODUCTION

Construction-site layout is an important planning activity; the outcome of this is a detailed drawing of the locations and areas reserved for the temporary support facilities. Temporary support facilities are diverse in nature but have the same general function, to support construction activities. They range from simple lay down areas to warehouses, fabrication shops, maintenance shops, batch plants, job offices, and labor residence facilities, depending on the size and location of the project. The impact of good layout practices on money and timesaving becomes more obvious on the larger projects (Hamiani and Popescu 1988).

The nature of the problem is such that no well-defined method can guarantee a solution or can be taught. At the most, guidelines point out the issues field managers must consider while laying out their project sites (Handa and Lang 1988, 1989; Rad and James 1983). The layout problem have been treated by operation research (Seehof and Evans 1967; Moore 1980) and artificial intelligence (Hamiani and Popescu 1988; Tommelein et al. 1991).

In this study, we formulate the problem as a discrete combinatorial optimization problem. Construction-site layout is delimited as the design problem of arranging a set of predetermined facilities on a set of predetermined sites, while satisfying a set of layout constraints and optimizing layout objectives. For n facilities, the number of possible alternatives, that is the number of feasible configurations, is $n!$ This is a huge number, even for a small n. For 10 facilities, the number of possible alternatives is already well over 3,628,000. For 15 facilities we are already in the 12-digit numbers. In practical applications, a project with $n = 15$ is still small.

The subject of combinatorial optimization consists of a large set of problems central to many disciplines in science and engineering. Research in this area is aimed at finding optimal values for functions of many independent variables representing different parts of the system. The function usually referred to as the cost function represents in some sense the goodness of the system configuration and, hence, depends on the complex interaction among parts of the system. Due to their interacting, integer-valued variables, discrete combinatorial optimization problems are difficult for most stand optimization techniques, such as linear programming, integer programming, etc. These problems have been typically solved by using problem-specific heuristics (Rad and James 1983), branch-and-bound searching (Lawler and Wood 1966), or by relaxing one or more constraints and solving the simplified problem (Fisher 1981).

Artificial neural networks are a family of massively parallel architecture that solve difficult problems via the cooperation of highly interconnected but simple computing elements (or artificial neurons) (Fahlman and Hinton 1987; Wasserman 1989; Yeh et al. 1993). The simplicity of the neurons makes it promising to build them in large numbers to achieve high computing speeds through massive parallelism. Hopfield neural networks (Hopfield 1982; Hopfield and Tank 1985), a kind of artificial neural networks, have been proposed as a model of computation to solve a wide variety of discrete combinatorial optimization problems. These networks contain many simple computing elements, which cooperatively traverse the energy surface defined by the energy function to find a local or global minimum. In spite of its novel perspective, however, the Hopfield approach to the solution of these optimization problems is limited by its inability to ferret out global minimum with a single invocation of the algorithm. It is effectively a steepest-descent procedure that settles at the nearest local minimum. A global minimum is ensured only by initializing the algorithm at a sufficient number of different starting points. In addition, Hopfield neural networks have several parameters that need to be selected and, often, are carefully tuned for a network to produce a sensible computation.

Simulated annealing (SA) (Kirkpatric et al. 1983; Otten and Ginneken 1990) was proposed as a general technique for attempting to solve combinational optimization problems. Simulated annealing is a probabilistic hill-climbing search algorithm, which finds a global minimum of energy function by combining gradient descent with a random process. This combination allows, under certain conditions, changes to state that actually increase the energy function, thus providing SA with a mechanism to escape from local minimum. Although changes to state that decrease the energy function are always accepted, a move that causes an increase will be taken with a Boltzmann probability. Though, SA has been effective in many practical problems such as the traveling salesman problem (TSP), in terms of the quality of solutions, it requires unacceptably large computing times. Further, the simulated annealing procedure is inherently sequential, and efforts to make it parallel have not met with more than moderate success (Van den Bout and Miller 1990).

A new neural network model, named annealed neural network, which merges many features of simulated annealing and Hopfield neural networks was proposed (Van den Bout

[1]Assoc. Prof., Dept. of Civ. Engrg., Chung-Hua Polytechnic Inst., 30 Tung Shiang, Hsin Chu, Taiwan, 30067, Republic of China.

Note. Discussion open until December 1, 1995. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on April 14, 1993. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 9, No. 3, July, 1995. ©ASCE, ISSN 0887-3801/95/0003-0201–0208/$2.00 + $.25 per page. Paper No. 5988.

J. Comput. Civ. Eng. 1995.9:201-208.

and Miller 1990). Annealed neural network exhibits the rapid convergence of the neural network, while preserving the solution quality afforded by simulated annealing. The annealed neural networks have been proposed as a model of combinational optimization problem such as the TSP and graph partitioning problem (Van den Bout and Miller 1990).

This paper describes an effort that applies the annealed neural network to generate a construction-site layout alternative. In this paper, the second section introduces the construction-site layout problem. The third section describes the annealed neural network model. The fourth section describes the model using the annealed neural network to solve the problem. A simple example is used to illustrate the algorithm in the fifth section. Two case studies illustrate the practical applications and demonstrate this model's efficiency in solving the construction-site layout problem in the sixth section. In the seventh section, the effects of three control parameters of the model are experimented on. The summary and conclusions are given in the final section.

This paper is novel in at least the following three aspects: (1) the annealed neural network is applied to solve the construction-site layout problem; (2) a strategy to set the reasonable initial temperature in the simulated annealing procedure is proposed; and (3) the effects of various parameters in the annealed neural network are examined.

CONSTRUCTION-SITE LAYOUT PROBLEM

In the present work, the formulation of the construction-site layout problem is that a set of facilities needs to be located on the site, while optimizing layout objectives and satisfying a set of layout constraints. The layout objectives represent the goodness of the functional interactions of the facility with the world outside the site, the work area, and other facilities. Some layout objectives considered in this paper include the adjacency of objects, distance between objects, availability of space for object location, positions of objects in relation to others, and view of objects from others. The layout constraints measure the feasibility of the layout, i.e., each site should be assigned with one and only one facility, and each facility should be assigned on one and only one site. Then, the problem is formulated as follows:

$$\min F = \sum_x \sum_i \delta_{xi} C_{xi} + \sum_x \sum_i \sum_y \sum_j \delta_{xi} \delta_{yj} A_{ij} D_{xy} \quad (1a)$$

$$\text{subject } \delta_{yi} = 0 \quad \text{if} \quad \delta_{xi} = 1 \quad \text{and} \quad y \neq x \quad (1b)$$

$$\delta_{xj} = 0 \quad \text{if} \quad \delta_{xi} = 1 \quad \text{and} \quad j \neq i \quad (1c)$$

where F = cost function; δ_{xi} = permutation matrix variable (is 1 if facility X is assigned on site i); C_{xi} = construction cost of assigning facility X on site i; A_{ij} = site neighboring index, if site i neighbors site j; A_{ij} = 1, if site i does not neighbor site j, A_{ij} = 0, and A_{ii} = 0; and D_{xy} = interactive cost of assigning facility X on the site neighboring facility Y, and D_{xx} = 0.

In the cost function, the first term accounts for the total construction cost of assigning a facility on a site and the second term represents the total interactive cost between facilities.

In practical applications, if the number of predetermined sites is greater than the number of predetermined facilities, this problem can be solved by adding several fictitious facilities to make both numbers equal, and by assigning their construction cost and interactive cost was 0. Besides, if a predetermined site i is very unsuitable to a predetermined facility X, based on size or shape, this problem can be solved by assigning a very high construction cost C_{xi}. Therefore, (1) is rather practical for the construction-site layout problem.

Each site layout alternative can also be represented by $n \times n$ permutation matrix whose rows and columns are labeled by facilities and sites, respectively. The permutation matrix has only one 1 in each row and each column, with the remaining elements being 0. Fig. 1 shows a permutation matrix in which facility B is assigned on site 1, facility A is assigned on site 2, facility D is assigned on site 3, and facility C is assigned on site 4.

	Site			
Faci	**1**	**2**	**3**	**4**
A	0	1	0	0
B	1	0	0	0
C	0	0	0	1
D	0	0	1	0

FIG. 1. Permutation Matrix with Four Facilities (or Sites)

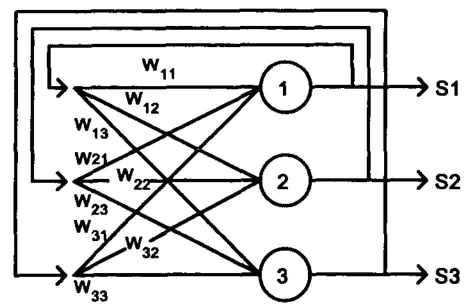

FIG. 2. Architectures of Hopfield Neural Network

ANNEALED NEURAL NETWORK

Hopfield Neural Network

The architecture of the Hopfield neural network is shown in Fig. 2. In the Hopfield network state variable S_i is 1 if neuron i is on, and 0 otherwise. Hopfield identified an energy function for networks of symmetrically connected binary threshold units (Hopfield 1982). The energy of any state of the network is given by

$$E = -\frac{1}{2} \sum_i \sum_j S_i S_j W_{ij} + \sum_i S_i \theta_i \quad (2)$$

where W_{ij} = weight on the connection between neuron i and neuron j; S_i = state variable (1 if neuron i is on, and 0 otherwise); and θ_i = threshold for neuron i.

Hopfield's energy function can be interpreted in the following way: A connection between units i and j with a positive weight represents the constraint that if one of these units is on, the other should also be on. A negative connection weight says that if one of the units is on, the other should not be on.

Given Hopfield's quadratic definition of energy, each unit i can determine the difference between the global energy of the network when it is off and the global energy when it is on, given the current states of the other units. This energy gap is simply

$$\Delta E_i = E|(S_i = \text{off}) - E|(S_i = \text{on}) = E|(S_i = 0) - E|(S_i = 1) = \sum_j S_j W_{ij} - \theta_i \quad (3)$$

If the energy gap is positive, the unit should turn on (or stay on) to minimize the global energy. Otherwise it should turn off (or stay off). In other words, to minimize energy it should behave exactly like a binary threshold unit. Using the

J. Comput. Civ. Eng. 1995.9:201-208.

aforementioned procedure iterately, the energy function will gradually reduce and reach its minimum value. In other words, the Hopfield network employs the method of steepest descent to find an optimum or near-optimum solution in the solution space.

To map a problem onto a neural network, we need to perform the following procedure:

1. Choose a representation scheme that allows the state variables S of the neurons to be decoded into a solution to the problem.
2. Choose an energy function E whose minimum value corresponds with the best solutions to the problem to be mapped.
3. Derive connectivities W and input thresholds θ from the energy function; these should appropriately represent the instance of the specific problem to be solved.
4. Set up initial state variables S and run the iteration procedure until the energy function is stationary, then the set of state variables is the optimum or near-optimum solution to the problem.

In the preceding four steps, step 2 is the most important one in mapping the problem onto the Hopfield network, and will play a critical role in deciding whether a specific problem can be solved easily by using neural networks.

There is no direct method to map constrained optimization problems onto a neural network except through the addition of terms in the energy function, which penalize a violation of the constraints. In such a case, one has to represent the energy function as a weighted sum of the problem function to be minimized and the penalty for violating constraints.

Simulated Annealing

Simulated annealing is a search technique that has been applied to a number of optimization problems. The idea is that we escape from high local minimum by adding a random component to the decision process of each unit. In most cases, the unit still takes a step downhill, but occasionally it will take a step uphill instead. More precisely, the probability that the unit takes a step uphill is a Boltzmann probability

$$P = \exp\left(-\frac{\Delta E}{T}\right) \tag{4}$$

where $\Delta E = E^{k+1} - E^k$, i.e., the energy of the $k + 1$th iteration solution minus the kth one; and T = temperature, a scaling factor that controls the amount of random perturbations in the simulated annealing process. It is analogous to a temperature for random thermal perturbations.

The simulated annealing algorithm can be written as follows:

```
set initial solution Ω
set a starting temperature T;
repeat
  choose a solution Ω' in the neighborhood of Ω;
  calculate energy gap ΔE = E(Ω') − E(Ω);
  if ΔE < 0 then Ω = Ω';
  else
    choose a random number β between 0 and 1;
    if β < exp(−ΔE/T) then Ω = Ω';
  calculate new energy function E;
  lower temperature T;
until energy function is stationary
```

At any given T, once the system has reached thermal equilibrium, the relative probability of being in high energy state A versus low energy state B (see Fig. 3) obeys a Boltzmann distribution (Fahlman and Hinton 1987)

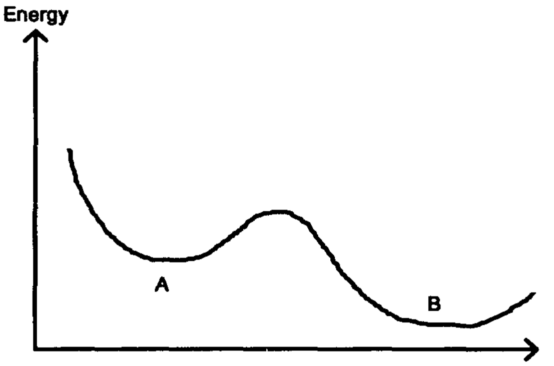

FIG. 3. Energy Surface with Local Minimum at A, Global Minimum at B

$$\frac{P(A)}{P(B)} = \exp\left[-\frac{E(A) - E(B)}{T}\right] \tag{5}$$

Thermal equilibrium does not mean that the system has settled into a particular stable state. It means that the probability distribution over states has settled down, even though the states are still changing. The best way to reach thermal equilibrium at a given temperature T is generally to start at a higher temperature, which makes it easy for the system to cross energy barriers but gives it little preference for the lower energy states. Then we gradually increase the preference for low energy states by reducing T. If T is reduced slowly enough, there is a high probability of ending up in the best global state, or if not there, in a state not much worse than the best. This process of slow cooling is analogous to the slow annealing of a metal to crystallize it in its lowest energy state (Fahlman and Hinton 1987).

Annealed Neural Network

The annealed neural network (Van den Bout and Miller 1990) combines characteristics of the simulated annealing algorithm and the Hopfield neural network. The annealed neural network exhibits the rapid convergence of the neural network while preserving the solution quality afforded by simulated annealing.

The annealed neural network algorithm can be written as follows:

```
set initial state variables S on network;
set a starting temperature T;
calculate initial energy function E;
repeat
  for i = 1 to n do
    calculate energy gap ΔE_i using (3);
    calculate new state variable S_i = P_i using (4);
  continue
  calculate new energy function E using (2);
  lower temperature T;
until energy function is stationary
```

CONSTRUCTION-SITE LAYOUT USING ANNEALED NEURAL NETWORK

Energy Function

The energy function of the construction-site layout problem must satisfy two requirements: (1) It must favor solutions with minimum cost; and (2) it must be low for only those solutions that satisfied the layout constraints.

J. Comput. Civ. Eng. 1995.9:201-208.

The first requirement, favoring low cost, is satisfied by adding a term to the energy function as follows:

$$E_1 = \sum_x \sum_i V_{xi} C_{xi} + \sum_x \sum_i \sum_y \sum_j V_{xi} V_{yj} A_{ij} D_{xy} \quad (6)$$

where V_{xi} = state variable, a probability of assigning facility X on site i.

Each state variable is looked on as the probability of finding facility X, currently assigned to site i of the alternative.

The second requirement is satisfied by adding a term to the energy function as follows:

$$E_2 = \sum_x \sum_i \sum_{y \neq x} V_{xi} V_{yi} \quad (7)$$

Therefore, the total energy function can be given as follows:

$$E = E_1 + \lambda \cdot E_2 \quad (8)$$

where λ = penalty factor.

The first term accounts for the total alternative cost, while the second term maintains feasibility by acting as a repulsive force that discourages two facilities from occupying the same site on an alternative basis.

The setting for the penalty factor determines the importance of the penalty term of the energy function. Setting it to a small value emphasizes the term concerned with cost and leads to low cost, which is, unfortunately, invalid. Alternatively, setting a large value makes the penalty so stiff that the neural network will converge to any feasible solution regardless of its total cost.

If two facilities try to occupy the same site, the penalty term in the energy function should be incurred to prevent this type of a site layout alternative. This is ensured by setting the penalty factor slightly larger than the maximum construction cost gap plus the maximum interactive cost:

$$\lambda = \zeta \cdot (\Delta C_{\max} + D_{\max}) \quad (9)$$

where ζ = penalty coefficient, $\zeta > 0$. Generally, ζ should be approximately equal to 1; $\Delta C_{\max}$ = maximum construction cost gap = max C_{xi} − min C_{xi}; and $D_{\max}$ = maximum interactive cost.

By increasing the value of the penalty coefficient, that is by making the second term in (8) relatively more important, we can increase the rate of success, but then the quality of the solutions will not be as good.

Therefore, the total energy function can be written as follows:

$$E = \sum_x \sum_i V_{xi} C_{xi} + \sum_x \sum_i \sum_y \sum_j V_{xi} V_{yj} A_{ij} D_{xy} + \zeta \cdot (\Delta C_{\max} + D_{\max}) \cdot \sum_x \sum_i \sum_{y \neq x} V_{xi} V_{yi} \quad (10)$$

The energy gap can be calculated as follows:

$$\Delta E_{xi} = E|(V_{xi} = 1) - E|(V_{xi} = 0) = C_{xi} + \sum_y \sum_j A_{ij} D_{xy} V_{yj} + \zeta \cdot (\Delta C_{\max} + D_{\max}) \sum_{y \neq x} V_{yi} \quad (11)$$

The probability of assigning facility X on site i, V_{xi}, can be calculated by the Boltzmann distribution and the following normalization operation (Van den Bout and Miller 1990);

$$V_{xi} = \frac{\exp(-\Delta E_{xi}/T)}{\sum_j \exp(-\Delta E_{xj}/T)} \quad (12)$$

The normalization operation in (12) guarantees that each facility will be assigned to one site only, so there will be n facilities assigned to the n sites.

Setting Reasonable Initial Temperature

At a very high temperature, each facility will be assigned equally across each site. As the temperature is reduced, the facilities will begin to coagulate in particular sites, which will hopefully minimize the energy function. In general, the cooling procedure is proposed as follows (Kirkpatric et al. 1983):

$$T^{k+1} = \alpha \cdot T^k \quad (13)$$

where α = temperature cool coefficient, $\alpha < 1$. Generally, $\alpha = 0.9 - 0.99$.

The initial temperature is very important for convergence speed. At a too-high initial temperature, convergence will be very slow, while at a too-low initial temperature, convergence will be very quick, but the quality of the solutions will not be as good. In this paper, a formula to determine a reasonable initial temperature for facility X, T_x is proposed as follows:

$$\max_{i=1}^{n} \left[\frac{\exp(-\Delta E_{xi}/T_x)}{\sum_{j=1} \exp(-\Delta E_{xj}/T_x)} \right] = P_{\max} \quad (14)$$

where $P_{\max}$ = maximum initial probability in a normalization set; and

$$P_{\max} = \eta \cdot P_{avg} \quad (15)$$

where η = maximum initial probability coefficient, $1 < \eta < n$; P_{avg} = average probability, $P_{avg} = 1/n$; and n = number of neurons in a normalization set, i.e., the number of facilities (sites).

The reason is that the reasonable initial temperature should be able to raise the maximum initial probability in a normalization set to a reasonable value, for example, from double to triple average probability, i.e., $\eta = 2$–3.

Site Layout with Simulated Annealing Procedure

The construction-site layout procedure using the annealed neural network is summarized as follows:

```
set initial state variables, V_xi, on network;
set a starting temperature T using (14);
calculate initial energy function E using (10);
repeat
  for X = 1 to n do
    for i = 1 to n do
      calculate ΔE_xi using (11);
    continue
    for i = 1 to n do
      calculate V_xi using (12);
    continue
  continue
  calculate new energy function E using (10);
  calculate new temperature T using (13);
until energy function is stationary
```

ILLUSTRATIVE EXAMPLE

In this section, a simple example illustrates the aforementioned algorithm. The example contains four facilities, their datas, the construction cost matrix, site neighboring index matrix, and interactive cost matrix, which are shown in (16)–(18)

$$\mathbf{C} = \begin{bmatrix} 4 & 14 & 8 & 10 \\ 10 & 10 & 8 & 6 \\ 35 & 10 & 6 & 26 \\ 19 & 12 & 10 & 6 \end{bmatrix} \quad (16)$$

J. Comput. Civ. Eng. 1995.9:201-208.

$$
\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{17}
$$

$$
\mathbf{D} = \begin{bmatrix} 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 \\ 8 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \end{bmatrix} \tag{18}
$$

The direct evaluation with all 24 feasible alternatives yields an average alternative cost ($\pm$standard deviation) of 69.83 $\pm$ 20.20 with the minimum being 26.0.

The problem is resolved by the annealed neural network with the following parameters (the effects of parameters are examined in the following section):

- Penalty coefficient $\zeta = 1.0$
- Temperature cool coefficient $\alpha = 0.95$
- Maximum initial probability coefficient $\eta = 2.5$

The initial state variables on the network are produced through the following random formula

$$
V_{xi} = \frac{r + 0.5}{n} \tag{19}
$$

where n = number of facilities; and r = an uniform random number in (0, 1).

The initial state variable matrix is shown as

$$
\mathbf{V} = \begin{bmatrix} 0.16 & 0.16 & 0.17 & 0.26 \\ 0.15 & 0.26 & 0.22 & 0.15 \\ 0.23 & 0.13 & 0.30 & 0.14 \\ 0.37 & 0.13 & 0.33 & 0.34 \end{bmatrix} \tag{20}
$$

The computation procedure is described as follows:

1. Initial temperature: the initial temperature can be derived by (14). $T_a = 4.07$; $T_b = 7.16$, $T_c = 17.5$, and $T_d = 18.6$.
2. Energy gap: the energy gap of facility A can be derived by (11). $\Delta E_{a1} = 31.6$, $\Delta E_{a2} = 42.6$, $\Delta E_{a3} = 34.5$, and $\Delta E_{a4} = 45.2$.
3. State variables: the state variables of facility A can be derived by (12). $V_{a1} = 0.62$, $V_{a2} = 0.03$, $V_{a3} = 0.16$, and $V_{a4} = 0.17$.
4. Similarly, the energy gap and state variables of the other three facilities can be derived. Then, the state variable matrix is shown as

$$
\mathbf{V} = \begin{bmatrix} 0.62 & 0.04 & 0.31 & 0.02 \\ 0.02 & 0.62 & 0.12 & 0.24 \\ 0.06 & 0.19 & 0.62 & 0.13 \\ 0.12 & 0.17 & 0.09 & 0.62 \end{bmatrix} \tag{21}
$$

5. After 30 iterations, convergence is achieved and the final state variable matrix is shown as

$$
\mathbf{V} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.96 & 0.01 & 0.04 \\ 0.02 & 0.05 & 0.89 & 0.03 \\ 0.04 & 0.08 & 0.07 & 0.81 \end{bmatrix} \tag{22}
$$

Using (1), the cost function of the final alternative is 26.0 and it is equal to the best alternative.

The aforementioned procedure is implemented many times with random initial state variables. The solutions obtained by the annealed neural network method, based on 20 valid solutions, have an average alternative cost ($\pm$standard deviation) of 38.1 $\pm$ 18.0 with the minimum being 26.0, and a convergence success rate of 100%.

CASE STUDY

In this section, a C program called SitePlan is built on a personal computer to implement the annealed neural network algorithm, and two case studies are employed to demonstrate SitePlan in practical applications.

Case 1

In this case, there are two eight-story buildings on a campus; the location description of the case including 12 sites is shown in Fig. 4. The case contains the following 12 facilities:

1. Reinforcing steel shop, 1
2. Reinforcing steel shop, 2
3. Carpentry shop, 1
4. Carpentry shop, 2
5. Falsework shop, 1
6. Falsework shop, 2
7. Concrete batch plant, 1
8. Concrete batch plant, 2
9. Job office
10. Labor residence
11. Electricity equipment and water-supply shop
12. Warehouse

Their datas, construction cost matrix, site neighboring index matrix, and interactive cost matrix (the unit of all costs in the test case is $1,000) are

$$
\mathbf{C} = \begin{bmatrix}
35 & 35 & 30 & 30 & 35 & 15 & 10 & 15 & 6 & 6 & 7 & 10 \\
35 & 30 & 9 & 9 & 13 & 30 & 30 & 35 & 15 & 18 & 12 & 7 \\
18 & 15 & 15 & 15 & 15 & 8 & 14 & 10 & 8 & 10 & 15 & 15 \\
13 & 7 & 12 & 18 & 18 & 15 & 15 & 15 & 15 & 8 & 8 & 12 \\
18 & 15 & 15 & 20 & 15 & 8 & 10 & 8 & 8 & 7 & 15 & 15 \\
14 & 8 & 10 & 17 & 12 & 15 & 15 & 15 & 15 & 8 & 7 & 9 \\
32 & 35 & 15 & 15 & 15 & 10 & 9 & 13 & 7 & 10 & 15 & 15 \\
31 & 30 & 9 & 8 & 15 & 18 & 15 & 16 & 15 & 15 & 15 & 15 \\
39 & 35 & 13 & 8 & 8 & 15 & 18 & 15 & 8 & 18 & 9 & 18 \\
18 & 8 & 8 & 8 & 15 & 10 & 15 & 15 & 13 & 15 & 15 & 15 \\
7 & 10 & 8 & 19 & 15 & 10 & 10 & 8 & 15 & 10 & 6 & 15 \\
9 & 10 & 6 & 7 & 7 & 7 & 15 & 15 & 18 & 15 & 15 & 12
\end{bmatrix} \tag{23}
$$

$$
\mathbf{A} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \tag{24}
$$

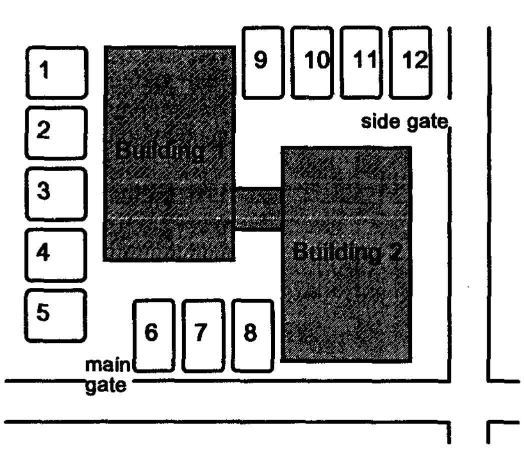

FIG. 4. Location Description of Test Case

J. Comput. Civ. Eng. 1995.9:201-208.

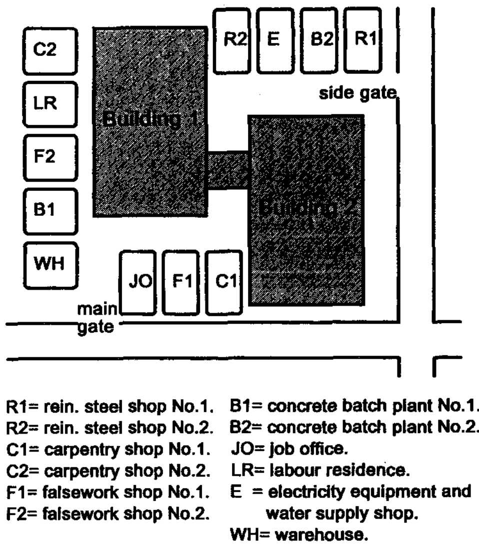

FIG. 5. Site Layout Alternative of Test Case 1 by Annealed Neural Network

$$
\mathbf{D} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{25}
$$

The direct evaluation with 1,000 feasible alternatives yields an average alternative cost ($\pm$standard deviation) of 321.3 $\pm$ 138.8, with the minimum being 139.0.

The evaluation through a greedy algorithmic method (refer to Appendix I) yields 509.0 (construction cost = 109, interactive cost = 400).

The problem is resolved by the annealed neural network with the aforementioned parameters. The solution, based on 20 experiments, has an average alternative cost ($\pm$standard deviation) of 110.1 $\pm$ 10.7 with the minimum being 93.0, and a convergence success rate of 100%. Each experiment takes about 13 s on a 80386 PC. Based on alternative cost, it is concluded that the annealed neural network is superior to the direct evaluation with 1,000 feasible alternatives and a greedy algorithmic method. One of the site layout alternatives by the annealed neural network is shown in Fig. 5.

Case 2

This case is similar to case 1, except the site neighboring index matrix and interactive cost matrix (the unit of all costs in the test case is \$1,000) are shown in the following:

$$
\mathbf{A} = \begin{bmatrix}
0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 \\
1 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \\
0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0.5 \\
0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 \\
1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0
\end{bmatrix} \tag{26}
$$

$$
\mathbf{D} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 100 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 100 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 100 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 100 & 0 \\
100 & 100 & 0 & 0 & 0 & 0 & 100 & 100 & 0 & -50 & -50 & -50 \\
100 & 100 & 0 & 0 & 0 & 0 & 100 & 100 & -50 & 0 & 0 & 0 \\
100 & 100 & 0 & 0 & 0 & 0 & 100 & 100 & -50 & 0 & 0 & 100 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -50 & 0 & 100 & 0
\end{bmatrix} \tag{27}
$$

The direct evaluation with 1,000 feasible alternatives yields an average alternative cost ($\pm$standard deviation) of 804.1 $\pm$ 261.5, with the minimum being 17.0.

The evaluation with a greedy algorithmic method (refer to Appendix I) yielded 1045.0 (construction cost = 95, interactive cost = 950).

The problem is resolved by the annealed neural network with the aforementioned parameters. The solution, based on 20 experiments, has an average alternative cost ($\pm$standard deviation) of -22.1 ± 101.2 with the minimum being -133.0, and a convergence success rate of 100%. Based on alternative cost, it is concluded that the annealed neural network is superior to the direct evaluation with 1,000 feasible alternatives and a greedy algorithmic method. A site layout alternative by the annealed neural network is shown in Fig. 6.

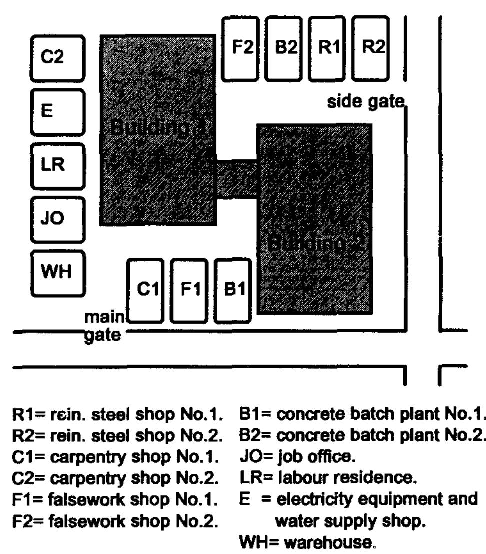

FIG. 6. Site Layout Alternative of Test Case 2 by Annealed Neural Network

TABLE 1. Comparison of Results with Various Penalty Coefficient of Case 1

Penalty coefficient (1)	Alternative Cost ($1,000) Average (2)	Alternative Cost ($1,000) Standard deviation (3)	Alternative Cost ($1,000) Minimum (4)	Convergence success rate (percent) (5)
0.0	—	—	—	0
0.125	—	—	—	0
0.25	95.2	3.0	92.0	25
0.5	106.7	8.1	95.0	100
0.75	108.2	9.4	97.0	100
1.0	110.1	10.7	93.0	100
1.5	145.2	58.1	100.0	100
2.0	136.8	48.5	105.0	100
5.0	208.3	111.3	119.0	100

TABLE 2. Comparison of Results with Various Temperature Cool Coefficient of Case 1

Temperature cool coefficient (1)	Alternative Cost ($1,000) Average (2)	Standard deviation (3)	Minimum (4)	Convergence success rate (percent) (5)
0.125	116.4	9.3	104.0	90
0.25	115.7	8.5	104.0	100
0.5	114.7	12.2	93.0	90
0.75	115.3	7.2	104.0	100
0.9	113.5	13.0	99.0	100
0.95	110.1	10.7	93.0	100
0.975	114.3	12.1	94.0	100
1.0	116.0	9.6	94.0	100

TABLE 3. Comparison of Results with Various Maximum Initial Probability Coefficient of Case 1

Maximum initial probability coefficient (1)	Alternative Cost ($1,000) Average (2)	Standard deviation (3)	Minimum (4)	Convergence success rate (percent) (5)
1.25	—	—	—	0
1.5	—	—	—	0
1.75	99.6	3.5	94.0	80
2.0	110.0	7.3	98.0	100
2.5	110.1	10.7	93.0	100
3.0	119.6	11.4	106.0	100
5.0	124.8	12.3	111.0	100

TABLE 4. Comparison of Results with Various Penalty Coefficient of Case 2

Penalty coefficient (1)	Alternative Cost ($1,000) Average (2)	Standard deviation (3)	Minimum (4)	Convergence success rate (percent) (5)
0.0	—	—	—	0
0.125	—	—	—	0
0.25	−131.3	8.6	−142.0	40
0.5	−23.4	60.2	−137.0	80
0.75	−17.4	61.2	−132.0	90
1.0	−22.1	101.2	−133.0	100
1.5	−19.5	59.0	−122.0	100
2.0	61.7	135.3	−104.0	100
5.0	329.6	256.8	32.0	100

TABLE 5. Comparison of Results with Various Temperature Cool Coefficient of Case 2

Temperature cool coefficient (1)	Alternative Cost ($1,000) Average (2)	Standard deviation (3)	Minimum (4)	Convergence success rate (percent) (5)
0.125	1.3	76.1	−111.0	100
0.25	7.7	85.8	−129.0	100
0.5	−22.3	73.2	−128.0	100
0.75	−21.5	65.8	−126.0	100
0.9	−32.2	101.8	−127.0	100
0.95	−22.1	101.2	−133.0	100
0.975	−20.6	62.7	−130.0	95
1.0	−19.5	59.0	−122.0	100

TABLE 6. Comparison of Results with Various Maximum Initial Probability Coefficient of Case 2

Maximum initial probability coefficient (1)	Alternative Cost ($1,000) Average (2)	Standard deviation (3)	Minimum (4)	Convergence success rate (percent) (5)
1.25	—	—	—	0
1.5	−89.1	31.7	−123.0	25
1.75	−79.6	40.0	−135.0	80
2.0	−52.2	56.5	−130.0	90
2.5	−22.1	101.2	−133.0	100
3.0	−6.9	64.6	−128.0	100
5.0	76.7	76.3	−49.0	100

EFFECTS OF PARAMETERS

For comparison purposes, the aforementioned implementation is reimplemented by the cases applying various parameters including penalty coefficient, temperature cool coefficient, and maximum initial probability coefficient. Their results, based on 20 experiments, are shown in Tables 1–6. From the results, some observations can be given as follows:

Penalty coefficient (refer to Tables 1 and 4): The results show that the average alternative cost and minimum alternative cost decrease monotonically with the decrease of the penalty coefficient. But the convergence success rate also decreases montonically with the decrease of the penalty coefficient. Therefore, a penalty coefficient, from 0.5 to 1.0, is suggested to obtain a low alternative cost and high convergence success rate.

Temperature cool coefficient (refer to Tables 2 and 5): The results show that the convergence success rate is almost independent of the temperature cool coefficient. The results also show that a critical range, about 0.9–0.975, of the temperature cool coefficient exists for minimizing the average alternative cost and minimum alternative cost. Therefore, a temperature cool coefficient, from 0.9 to 0.975, is suggested to get a low alternative cost and high convergence success rate.

Maximum initial probability coefficient (refer to Tables 3 and 6): The results show that the average alternative cost and minimum alternative cost decreases monotonically with the decrease of the maximum initial probability coefficient. But the convergence success rate also decreases monotonically with the decrease of the maximum initial probability coefficient. Therefore, a maximum initial probability coefficient, from 1.75 to 2.5, is suggested for a low alternative cost and high convergence success rate.

SUMMARY AND CONCLUSIONS

In this study, construction-site layout is delimited as a combinatorial optimization problem that consists of arranging a set of predetermined facilities on a set of predetermined sites while satisfying a set of constraints. The annealed neural network model, which merges many features of the simulated annealing and Hopfield neural networks, was employed to solve the problem. A strategy to set a reasonable initial temperature in the simulated annealing procedure was proposed, and the effects of various parameters in the annealed neural network were examined. Two case studies illustrated the practical applications and demonstrated this model's efficiency in solving the problem.

The main limitation of this approach is that the sites must be predeterminated, but the problem can be solved by assigning all available sites as predeterminated sites, and adding several fictitious facilities to make the number of predeter-

J. Comput. Civ. Eng. 1995.9:201-208.

mined sites equal the number of facilities (including fictitious facilities).

In practical applications, the predetermination of the construction cost and interactive cost is not easy, and professional experience is necessary. Because a human being needs years of study or practice to become an expert, building a knowledge-based expert system (Yeh et al. 1991, 1992) is valuable for storing and employing the expertise. This problem will be considered in future research.

ACKNOWLEDGMENT

This research was funded by the National Science Council, Taiwan, Republic of China (ROC), under contract NSC 82-0113-E-216-025-T.

APPENDIX I. ALGORITHM TO SOLVE CONSTRUCTION-SITE LAYOUT PROBLEM

1. Form a queue FACILITY consisting of all facilities, and form a queue SITE consisting of all sites
2. Until the FACILITY is empty
   • Calculate the minimum possible cost MINCOST, for each facility in FACILITY

$$\mathrm{MINCOST}_x = \min_i \left(C_{xi} + \sum_y \sum_j A_{ij} D_{xy} \right) \tag{28}$$

   where x = a facility in FACILITY; i = a site in SITE; y = a facility not in FACILITY; and j = a site not in SITE.
   • Select the facility x in FACILITY with

$$\mathrm{MINCOST}_x = \min_y \mathrm{MINCOST}_y \tag{29}$$

   where y = a facility in FACILITY
   • Select the site i with $\mathrm{MINCOST}_x$.
   • Assign the facility x to site i
   • Remove the facility x from FACILITY
   • Remove the site i from SITE

APPENDIX II. REFERENCES

Fahlman, S. E., and Hinton, G. E. (1987). "Connectionist architectures for artificial intelligence." *Computer*, 100–109.

Fisher, M. L. (1981). "The Lagrangian relaxation method for solving integer programming problems." *Mgmt. Sci.*, 27(1), 1–18.

Hamiani, A., and Popescu, G. (1988). "CONSITE: a knowledge-based expert system for site layout." *Computing in civil engineering: microcomputers to supercomputers*, K. M. Will, ed., ASCE, New York, N.Y., 248–256.

Handa, V., and Lang, B. (1988). "Construction site planning." *Constr. Canada*, 30(3), 43–49.

Handa, V., and Lang, B. (1989). "Construction site efficiency." *Constr. Canada*, 31(1), 40–48.

Hopfield, J. (1982). "Neural networks and physical systems with emergent collective computational abilities." *Proc.*, Nat. Acad. of Sci., Vol. 79, 2554–2558.

Hopfield, J., and Tank, D. (1985). "Neural computations of decisions in optimization problems." *Biological Cybernetics*, Vol. 52, 141–152.

Kirkpatric, S., Gelatt, C., and Vecchi, M. (1983). "Optimization by simulated annealing." *Science*, 220(4598), 671–680.

Lawler, E. L., and Wood, D. E. (1966). "Branch-and-bound methods: a survey." *Operations Res.*, 14(4), 679–719.

Moore, J. (1980). "Computer methods in facility layout." *Industrial Engrg.*, 19(Sep.), 82–93.

Otten, R. H. J. M., and Van Ginneken, L. P. P. P. (1990). *The annealing algorithm*, Kluwer Academic Publishers, London, England.

Rad, P. F., and James, B. M. (1983). "The layout of temporary construction facilities." *Cost Engrg.*, 25(2), 19–27.

Seehof, J. M., and Evans, W. O. (1967). "Automated layout design program." *J. Industrial Engrg.*, Vol. 18, 690–695.

Tommelein, I. D., Levitt, R. E., and Confrey, T. (1991). "SightPlan expertments: alternate strategies for site layout design." *J. Comp. in Civ. Engrg.*, ASCE, 5(1), 42–63.

Van den Bout, D. E., and Miller, T. K. (1990). "Graph partitioning using annealed neural networks." *IEEE Trans., on Neural Networks*, 1(2), 192–203.

Wasserman, P. D. (1989). *Neural computing: theory and practice*. Van Nostrand Reinhold, New York, N.Y.

Yeh, Y.-C., Kuo, Y.-H., and Hsu, D. S. (1991). "Expert system for diagnosing damage of prestressed concrete pile during construction process." *J. Constr. Engrg. Mgmt.*, ASCE, 117(1), 13–26.

Yeh, Y.-C., Kuo, Y.-H., and Hsu, D.-S. (1992). "Building KBES for diagnosing PC pile with inductive learning." *J. Comp. in Civ. Engrg.*, ASCE, 6(2), 200–219.

Yeh, Y.-C., Kuo, Y.-H., and Hsu, D. S. (1992). "Building a KBES for diagnosing PC pile with artificial neural network." *J. Comp. in Civ. Engrg.*, ASCE, 7(1), 71–93.

APPENDIX III. NOTATION

The following symbols are used in this paper:

A_{ij} = site neighboring index, if site i neighbors site j, A_{ij} = 1, if site i does not neighbor site j, A_{ij} = 0, and A_{ii} = 0;
C_{xi} = construction cost of assigning facility X on site i;
D_{xy} = interactive cost of assigning facility X on the site neighboring facility Y, and $D_{xx} = 0$;
D_{max} = maximum of interactive cost;
E = energy function;
F = cost function;
n = number of facilities (sites);
P = probability;
P_{avg} = average probability;
P_{max} = maximum initial probability, $P_{max} = \eta \cdot P_{avg}$;
S_i = state variable (1 if neuron i is on and 0 otherwise);
T = simulated annealing temperature;
V_{xi} = state variable, a probability of assigning facility X on site i;
W_{ij} = weight on the connection between neuron i and neuron j;
α = temperature cool coefficient;
ΔC_{max} = maximum gap of construction cost;
ΔE = energy gap;
δ_{xi} = permutation matrix variable (1 if element i is assigned on position j);
ζ = penalty coefficient;
η = maximum initial probability coefficient;
θ_i = threshold for neuron i; and
λ = penalty factor.

Subscripts

i, j = site indices; and
x, y = facility indices.