Chapter 4

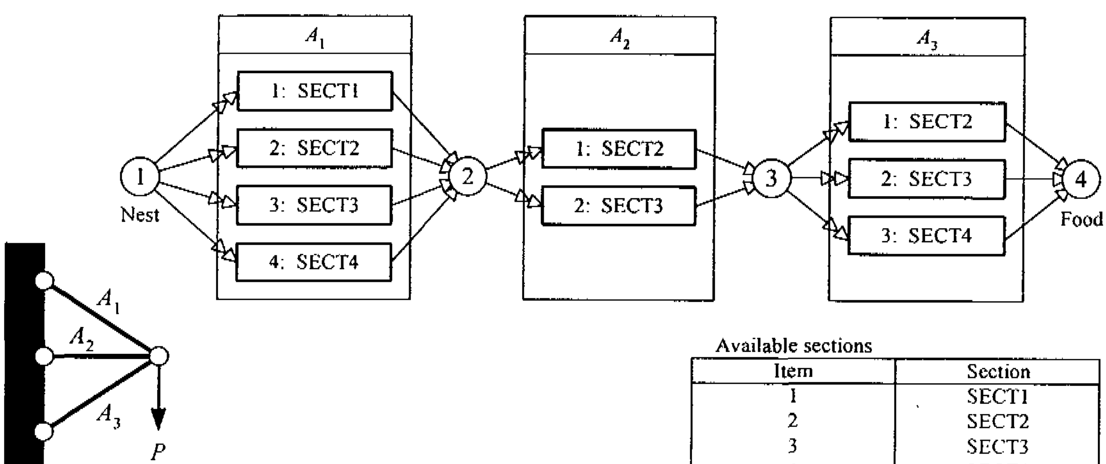
An ACO Algorithm for Sizing Optimization of Structures

4.1 Rules of Artificial Ants

Some concepts the previous chapter are transformed into the ACO algorithm for solving structural optimization problems. To understand the concept of the ACO for sizing optimization of structures, consider Figure 4.1 that shows a sizing optimization problem of a truss structure with three members. Since the truss has three members, there are three design variables, i.e. A_1, A_2 and A_3, which represent the sections of the three members. Assume that there are four available choices of sections for design variable A_1 while there are, respectively, two and three choices for design variable A_2 and A_3. As a result, the optimization becomes a problem of finding the combination of these available sections that results in an optimal structure. The problem is a combinatorial optimization problem and can be thought of as a foraging problem of an ant colony. As shown in Figure 4.1, an artificial nest and an artificial food source can be established. In the figure, node 1 represents the nest and node 4 represents the food source. The ants will have to move from node 1 to node 4 by passing all other nodes in between. Between each pair of nodes, there are sub-paths, representing available sections for each design variable. The partial walk of the ants between nodes 1 and 2 represents the selection for design variable A_1, and the partial walks between the subsequent nodes are for the subsequent design variables.

For the ACO to work, artificial ants will have to make many artificial tours and they must obey the following simple rules:

1) Ants will probabilistically select paths with higher levels of pheromone. In other words, paths with higher pheromone levels will have higher chance to be selected by ants.
2) The amount of pheromone laid by an ant on the path which it has walked depends upon the quality of the path. If the path is of high quality, the ant that has walked the path will lay a large amount of pheromone on the path. For



Available sections

Item	Section
1	SECT1
2	SECT2
3	SECT3
4	SECT4

Figure 4.1 Example of structural sizing optimization by the ACO approach

structural sizing optimization, a path is considered high quality if it represents an admissible structure with low weight.

These two rules, though simple, are enough for the colony to perform its task.

The first rule can be implemented by setting the probability of a sub-path being selected by an ant in tour t as

$$p(A_i^a,t)=\frac{\tau(A_i^a,t)}{\sum_{k=1}^{S_i}\tau(A_i^k,t)}. \tag{4.1}$$

Here, $p(A_i^a,t)$ denotes the aforementioned probability where A_i^a represents the a^{th} sub-path for design variable i. In addition, S_i denotes the total number of available sub-paths for design variable i. Finally, $\tau(A_i^a,t)$ denotes the amount of pheromone of sub-path A_i^a in tour t.

As an example, consider a partial walk between nodes 1 and 2 in the example in Figure 4.1. The partial walk between nodes 1 and 2 is the selection for design variable A_1. Between nodes 1 and 2, there are four available sub-paths; i.e. S_1=4. The probability of sub-path SECT3 (A_1^3) being selected by an ant, for example, can be written as

$$p(A_1^3,t)=\frac{\tau(A_1^3,t)}{\sum_{k=1}^{4}\tau(A_1^k,t)}. \tag{4.2}$$

For the first tour where there is still no pheromone on any sub-paths, a random selection can be used.

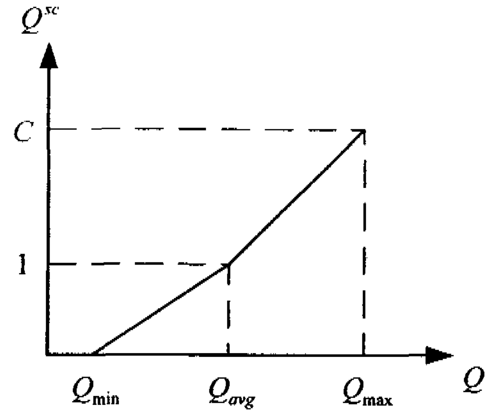
To be able to implement the second rule of ants, it is necessary to define a parameter that represents the quality of the path. Since each path actually represents a design, the objective function value and the degree of constraint violation of each design must be considered in order to evaluate the quality of the design or path. For sizing optimization of structures, the following form of a quality function Q may be employed, i.e.

$$Q(\mathbf{A})=Q_o(\mathbf{A})-\lambda E_r(\mathbf{A}). \tag{4.3}$$

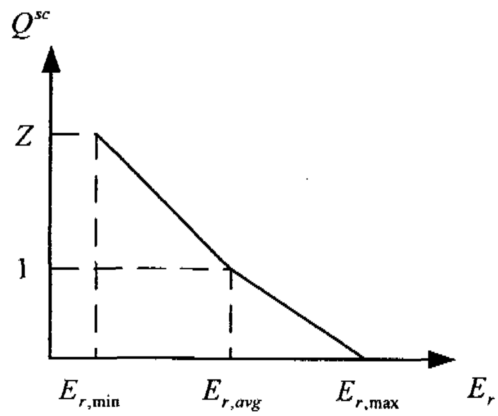
Here, $\mathbf{A}$ denotes a design vector defined as

$$\mathbf{A}=[A_1,A_2,\ldots,A_n]^T, \tag{4.4}$$

where A_i represents the i^{th} design variable and n denotes the number of design variables. Moreover, $Q_o(\mathbf{A})$ is a function representing the basic quality value calculated from the



(A) Some designs selected by ants are admissible.



(B) All designs selected by ants are inadmissible.

Figure 4.2 Bilinear scaling for the quality function

objective function while $E_r(\mathbf{A})$ is a non-negative function representing the degree of constraint violation. In addition, λ denotes a user-defined positive constant.

In order to have further control over the optimization, the quality function in Equation 4.3 will not be used directly. Rather, it will be scaled into a certain range. In this study, the bilinear scaling techniques shown in Figure 4.2 will be used. In the figure, the scaling scheme A is used when, in the tour under consideration, there are some admissible designs selected by the ants, and the scaling scheme B is used when all designs selected by the ants are inadmissible. In the figure, the subscripts *min*, *max* and *avg* denote the minimum, maximum and average values, respectively. In addition, C and Z are user-defined constants. The scaled quality function Q^{sc} will be used in the subsequence calculation instead of Q.

By employing the general procedure for ACO algorithms (see, for example, Gutjahr, 2000), the pheromone-trail-laying algorithm can be constructed. To this end, denote the design vector $\mathbf{A}$ selected by ant Ant_j during tour t as $\mathbf{A}(Ant_j, t)$ and define a function $\Delta\tau$ as

$$\Delta\tau(A_i^a, Ant_j, t) = \begin{cases} Q^{sc}[\mathbf{A}(Ant_j, t)]/n & \text{: if } Ant_j \text{ has traversed } A_i^a \text{ during tour } t, \\ 0 & \text{: otherwise.} \end{cases} \quad (4.5)$$

Note again that n denotes the number of the design variables.

Let

$$V(t) = \sum_{j=1}^{M} Q^{sc}[\mathbf{A}(Ant_j, t)], \quad (4.6)$$

where M denotes the number of the ants. It can be seen that $V(t)$ is actually the summation of the value of $\Delta\tau$ in tour t.

Next, let

$$\Delta\overline{\tau}(A_i^a, t) = \frac{\sum_{j=1}^{M} \Delta\tau(A_i^a, Ant_j, t)}{V(t)}. \quad (4.7)$$

Finally, define the pheromone-updating scheme as

$$\begin{aligned} &\tau(A_i^k, t) = 0 && t = 1, \\ &\tau(A_i^k, t+1) = \Delta\overline{\tau}(A_i^k, t) && t = 1, \\ &\tau(A_i^k, t+1) = (1-\rho)\tau(A_i^k, t) + \rho\Delta\overline{\tau}(A_i^k, t) && t \geq 2, \end{aligned} \quad (4.8)$$

where ρ denotes the evaporation factor. This factor is used to control the evaporation rate. It can be seen from Equation 4.8 that, from the second tour, the evaporation is implemented by the term $(1-\rho)\tau(A_i^k, t)$ while the pheromone laying is implemented by the term $\rho\Delta\overline{\tau}(A_i^k, t)$. It can also be seen from Equations 4.5-4.8 that, after the first tour, the sum of pheromone values of all sub-paths always remains equal to one.

In the calculation, identification of the best obtained design of all tours is naturally required. For a design to be acceptable, it must be at least admissible. Therefore, the quality function defined in Equation 4.3 cannot directly be used for the purpose of finding the best admissible design and a new rule of comparison must be employed. In the algorithm, finding the best admissible design is actually equivalent to finding the ant which selects that design. To find the best ant from all available ants in the calculation, the following rule of comparison is defined; i.e. Ant_i is considered to be better than Ant_j when

1) Ant_i selects an admissible design while Ant_j selects an inadmissible design, or

2) Both Ant_i and Ant_j select admissible designs but Ant_i's design has a larger Q_o, or

3) Both Ant_i and Ant_j select inadmissible designs but Ant_i's design has a smaller E_r.

Finally, the complete algorithm can be summarized as

```
Tour=1;
All_Ants_Select_Paths(Random);
Calculate_Paths_Quality();
Find_The_Best_Ant_of_The_Tour();
Update_The_Best_Ant_of_All_Tours();
All_Ants_Lay_Pheromone();
For Tour=2 to N
{
        All_Ants_Select_Paths(Pheromone_Based);
        Calculate_Paths_Quality();
        Find_The_Best_Ant_of_The_Tour();
        Update_The_Best_Ant_of_All_Tours();
        Pheromone_Evaporation();
        All_Ants_Lay_Pheromone();
}
```

4.2 A Greedy Heuristic in the ACO Algorithm

In order to possibly improve the quality of the ACO algorithm, a greedy heuristic can be, to a desirable degree, incorporated into the algorithm. From Equation 4.1, it can be seen that ants select their paths based only on the level of pheromone. To introduce the greedy characteristic to their search, the probability function in Equation 4.1 is modified as

$$p(A_i^a, t) = \frac{[\tau(A_i^a, t)]^\alpha [\eta(A_i^a)]^\beta}{\sum_{k=1}^{S_i} \left\{ [\tau(A_i^k, t)]^\alpha [\eta(A_i^k)]^\beta \right\}}, \tag{4.9}$$

where $\eta(A_i^a)$ is a greedy function. In this study, this function is taken as the inverse of the corresponding sectional area of sub-path A_i^a, i.e.

$$\eta(A_i^a) = \frac{1}{Area(A_i^a)}. \tag{4.10}$$

In addition, α and β are non-negative parameters that control the relative importance of the pheromone level and the sectional area in the evaluation of the probability. When β is not zero, ants are not only attracted to sub-paths with high levels of pheromone but also sub-paths with small corresponding sectional sizes.