Abstract. By separating the search control and the solution updating of the commonly used simulated annealing technique, we propose a revised version of the simulated annealing method which produces better solutions and can reduce the computation time. We also use it to improve the performance of the Boltzmann machine. Furthermore, we present a simple combinatorial optimization model for solving the attributed graph matching problem of e.g. computer vision and give two algorithms to solve the model, one using our improved simulated annealing method directly, the other using it via the Boltzmann machine. Computer simulations have been conducted on the model using both the revised and the original simulated annealing and the Boltzmann machine. The advantages of our revised methods are shown by the results.

1. Introduction. Simulated Annealing (SA) has been widely used to solve various combinatorial optimization problems such as TSP, VLSI design [1,2] as well as clustering and attributed graph matching [3]. It has also been used in the Boltzmann Machine (BM) [4, 5-7].

By separating the Metropolis Sampling (MS) process which is a major part of the SA process, into a search control process and a solution updating process, one of the present authors proposed an Improved Simulated Annealing (ISA) method [8,9] which will be reviewed and further analyzed in this paper. This method can be guaranteed to always yield a better solution than SA. It is useful especially in the following cases, which are often encountered in actual applications:

(1) The time spent on each MS process is not long enough to let the process reach the equilibrium state.
(2) The speed of annealing is too fast.
(3) The temperature specified for stopping the annealing process is not low enough.

In these cases, SA usually finds a bad solution, but ISA can still obtain a better solution. In addition, ISA also has a simple but effective way to decide when a MS process can be finished to start another MS process, and when the whole annealing process can be stopped in such a way that the time cost is reduced but the solution is still satisfactory.

Furthermore, in this paper we will use the basic idea of ISA to improve the performance of BM. Advantages similar to the ones given above are again obtained.

Our work is motivated by the computer vision problem. Attributed graphs have turned out to be very useful data structures when used for image representation and understanding in computer vision systems [10-12]. They have also been successfully
used to handle optimal task assignments in distributed computer systems [13]. In [3], one of the authors proposed a way to use SA to implement attributed graph matching. This paper will give another solution: a general attributed graph matching problem is turned into a model of combinatorial optimization which is different from that in [3]. With this model, we can either directly use SA or use symmetrically interconnected neural networks via the BM to obtain the solution of attributed graph matching.

In sec. 2, the commonly used SA algorithm is analyzed. In sec. 3, the ISA is given, its advantages are discussed, and it is applied on the BM. In sec. 4, a general combinatorial optimization model for attributed graph matching is presented and solved by ISA and the revised BM. Finally, in sec. 5, the advantages of ISA are shown through computer simulations with performance comparisons on both the optimality of solution and the time costs.

2. Analysis of Simulated Annealing. In problem-solving with SA, each combinatorial state $s_i$ is regarded as a configuration state of a physical system, the objective function $E(s_i)$ as the system energy, and a parameter $T$ is used to imitate temperature. For a given $T$, MS is used to simulate the thermodynamical equilibrium at which the Boltzmann distribution

$$f(s_i) = \frac{e^{-E(s_i)}}{\sum_i e^{-E(s_i)}}$$

(1)

can properly describe the probability of the energy at each state. As $T$ gradually decreases, $f(s_i)$ becomes sharp around the state of global minimum energy and will be fixed at the state as $T \rightarrow 0$, i.e., the global optimization solution can be obtained.

Generally, the commonly used SA could be described as follows:

**Initialization:** Generate a random state $s$ as the present solution, and initialize $E(s)$ and $T = T^{(0)}$;

*step 1:* Randomly make a small perturbation $\Delta s$ to get a new state $s + \Delta s$ with the energy increment $\Delta E = E(s + \Delta s) - E(s)$;

*step 2:* If $\Delta E < 0$ goto step 3, otherwise, generate a random number $\xi$ by sampling a uniform distribution over $[0,1]$ and if $e^{-\Delta E/T} \leq \xi$, goto step 1;

*step 3:* $s + \Delta s$ replaces $s$ as the new present solution, and $E := E + \Delta E$;

*step 4:* Check whether the MS at the present $T$ reached its equilibrium; if not, goto step 1;

*step 5:* Reduce $T$ into $T' < T$ by some means (e.g., $T := \lambda T$). Check whether the annealing process has terminated (e.g., $T < T_{\text{min}}$); if yes, the present $s$ with its $E(s)$ is taken as the final solution, stop; otherwise, goto step 1.

There steps 1,2,3,4 implement the MS process and step 5 imitates the annealing procedure. The effectiveness of SA depends on: (1) Whether each MS process reaches its equilibrium, i.e., how long each MS process should take at each $T$. (2) Whether $T^{(0)}$ is high enough, $T_{\text{min}}$ is low enough, and $T$ decreases slowly enough. A low $T^{(0)}$, high $T_{\text{min}}$, sharply decreasing $T$ and short time implementation for each MS process will lead to low computer cost, but the solution is not so good. In contrast, very high $T^{(0)}$, very low $T_{\text{min}}$, and slowly decreasing $T$ will always result in an optimal or near-optimal solution, but usually the cost is substantially high. Although there are several