Stochastic Simulated Annealing Algorithms using Different Annealing Schedules: An Analytical Approach

Dr. Hussein Rady
The Shorouk Academy
Higher Institute for Computers and Information Systems
e.mail : dr_hussein_rady@yahoo.com

Abstract

Simulated Annealing is one of the most important meta-heuristics or general-purpose algorithms of combinatorial optimization, due to its convergence towards high quality solutions. However, it is associated with a high computational cost and difficulties related to the parameters settings. Therefore the algorithm's convergence speed has been the subject of a largenumber of research works. Settings the parametersof the algorithm determines the generation of the new solution. One of the most important features in simulated annealing is the choice of the annealing schedule, and many attempts have been made to derive or suggest good schedulesas an optimization technique. The precise rate of cooling is an essential part of Simulated Annealing as it determines its performance. In this paper, we make a comparative study of the performance of simulated annealing using the most important annealing strategies for selecting the initial value of temperature, the cooling schedule, the number of iterations to be performed, and the stopping criterion. The analytical results among different annealing schedules are studied, analyzed and compared. The results are encouraging for application purposes.

Keywords: *Annealing schedule**,** simulated annealing, fast annealing, simulated quenching, local search, global search.*

1- Introduction

Simulated Annealing (SA) is one of the emergent calculation algorithms that solves optimization problems, and is an effective technique for solving combination optimization problems[10, 14, 22, 23]. SA is an algorithm that simulates the physical evolution of a solid from a high temperature state until it reaches a thermal equilibrium state. SA searches randomly around the neighborhood of a present searching point. The next searching point can be accepted even when the fitness value of the next point is worse than that of the present. Given an initial state, the

SA algorithm repeats these steps until the optimization state is finally reached. Therefore, it can derive the global solution.

SA is based on an analogy with the homonymous thermodynamical process. For slowly cooled thermodynamical systems (e.g., metals), nature is able to find the minimum state of energy, while the system may end in an amorphous state of higher energy if it is cooled quickly. This principle is expressed by the Boltzmann probability distribution:

$$P(S) \sim \exp(- S / kT) \qquad \rightarrow (1)$$

The energy of a solution in thermal equilibrium at a given temperature T is probabilistically distributed among all different states S. The system may switch to a new energy state with probability P, irrespective of whether it is higher or lower. Therefore, nature's minimization strategy is to allow the system sometimes to go uphill as well as downhill, so that it has a chance to escape from a local energy minimum in favor of finding a better, more global minimum. However, the lower the temperature, the less likely is a significant uphill step[24].

The principle of annealing in optimization, is transferred by introducing a control parameter, analogue of temperature, and an annealing cooling schedule that describes its gradual reduction. Assuming a large enough initial temperature and a proper schedule, SA slowly converges to the globally optimal solution[13, 24].

The stochastic simulated annealing (SSA) combines the gradient descent technique which is a probabilistic hill-climbing algorithm with a random process to find the global minimum for its energy function E[9]. Simulated annealing models the degrees of freedom as a collection of atoms slowly being cooled into their stable states with the temperature T as the controlling parameter. The energy surface defined as E(s) for a particle state s is a Boltzmann distribution function that allows changes in s to increase E, thus providing the network with a mechanism to escape from being trapped in a local minimum[2]. This is made possible since changes to s which decrease E are always accepted, whereas a move which causes an increase ΔE will be taken with the Boltzmann probability: $Pr\{\text{uphill move}\} = \exp(-\Delta E / T)$.

SA algorithm has basically three important processes: generation, acceptance criterion, and cooling. SA searches a solution with one point. The process of generation creates the next searching point from the present point. The acceptance criterion then judges the transfer of the searching point from the present point to the generated point. This acceptance criterion consists of the temperature and the function value. Usually, the Metropolis standard is used as the acceptance criterion[12, 28].

Metropolis criterion: After every challenge from a configuration u to another configuration v, we compute the cost function variation $\Delta g = g(v) - g(u)$. The transformation is accepted with a probability:

$$P (u, v) = \exp (- \Delta g / T) \qquad \rightarrow (2)$$

When $\Delta g <= 0$, then $\exp (- \Delta g / T) >= 1$ and the new configuration is accepted with a probability P(v) = 1.

If $\Delta g > 0$, we compare P (u, v) with a random number r in the interval [0,1[:

- If r < P (u, v) then the new configuration is accepted.
- Else the new configuration is rejected.

In this case, the system tries to find another configuration. If it is impossible, the last configuration is accepted and the search is stopped when the stopping criterion is reached[16].

However, SA requires huge computational cost. Specifically, SA takes much time finding the optimum solution in continuous problems like when performing SA in parallel and when performing SA with other optimization algorithms.

The remainder of this paper is organized as follows: In section 2, the SA algorithm is illustrated. Section 3 shows the characteristics of the SA algorithm. In section 4, the Annealing Schedule is studied, analyzed and discussed. The simulation results are shown in section 5. Finally section 6 concludes this paper.

2- Simulated Annealing algorithm

The simulated annealing (SA) technique is the most powerful one of the stochastic algorithms. The SA algorithm is based on the concept of attaining the lowest energy state through slow cooling (e.g., annealing of metals) and is currently used in molecular modeling. Much of its success is due to random sampling of the parameter space, based on the probabilistic Monte Carlo method[1].

SA algorithm is a general purpose optimization technique. It has been derived from the concept of metallurgy in which we have to crystallize a liquid at a required temperature. In this process the liquid are initially at a high temperature and the molecules are free to move. As the temperature goes down, the movement of the molecules is restricted and the liquid begins to solidify. If the liquid is cooled slowly enough, it forms a crystallized structure. This structure is in the minimum energy state. If the liquid is cooled down rapidly, it forms a solid which is not in the minimum energy state. Thus the main idea in simulated annealing is to cool the liquid in a controlled matter and then to rearrange the molecules if the desired output is not obtained. This rearrangement of molecules takes place based on the objective function which evaluates the energy of the molecules in the corresponding iterative algorithm[5]. SA aims to achieve global optimum by slowly converging to a final solution, making downward moves hoping to reach a global optimum solution. Given a solution S_s, we select the neighbor solution S_n and the difference is calculated using the objective function

$$\Delta f = f(S_n) - f(S_s) \qquad \rightarrow (3)$$

If the function improves the value, i.e., if ($\Delta f < 0$) , then we replace the current solution with the new one. Otherwise, i.e., if ($\Delta f \geq 0$), then the new solution is accepted with a probability factor of $p(\Delta f) = \exp(-\Delta f / T)$, where T is the temperature which is the controlling parameter. The procedure is repeated until the terminating condition is met[5].

The stochastic character of the SA algorithm provides one of its main advantages: it is no longer necessary to make choices concerning the starting point. In fact, the initial set of parameters is generated randomly in order to avoid any bias in the choice of the subsequent search trajectory. On the other hand, since the sampling of the parameter space must be adequate, the time necessary to complete a single SA run is longer than that required by a single run of a deterministic algorithm[1].

The algorithm is explained below:

Start with randomized states throughout the network, $s_i(1)$, and select a high initial "temperature" $T(1)$. Next, choose a node i randomly. Suppose its state is $s_i = +1$. Calculate the system energy in this configuration, E_a; next recalculate the energy, E_b, for a candidate new state $s_i = -1$. If this candidate state has a lower energy, accept this change in state. If however the energy is *higher*, accept this change with a probability equal to

$$e^{-\Delta E_{ab}/T} , \qquad \rightarrow (4)$$

where $\Delta E_{ab} = E_b - E_a$. This occasional acceptance of a state that is energetically less favorable is crucial to the success of simulated annealing, and is a marked distinction from the naive gradient descent and the greedy approach. The key benefit is that it allows the system to jump out of unacceptable local energy minima. For example, at very high temperatures, every configuration has a Boltzmann factor $e^{-E/T} \approx e^0$ roughly equal. After normalization by the partition function, then, every configuration is roughly equally likely. This implies every node is equally likely to be in either of its two states.

The algorithm continues *polling* (selecting and testing) the nodes randomly several times and setting their states in this way. Next, the algorithm lowers the temperature and repeats the polling. Now, according to Equation (4), there will be a slightly smaller probability that a candidate higher energy state will be accepted. Next the algorithm polls all the nodes until each node has been visited several times. Then the temperature is lowered further, the polling repeated, and so forth.

At very low temperatures, the probability that an energetically less favorable state will be accepted is small, and thus the search becomes more like a greedy algorithm. Simulated annealing terminates when the temperature is very low (near zero). If this cooling has been sufficiently slow, the system then has a high probability of being in a low energy state — hopefully the global energy minimum[29].

We let N_i denote the set of nodes connected with non-zero weights to node i. In a fully connected net, N_i would include the complete set of $N - 1$ remaining nodes. Further, we let Rand[0, 1) denote a randomly selected positive real number less than 1. With this notation, the pseudocode for the randomized or stochastic simulated annealing algorithm presented in page 355 by Duda/Hart/stork [29]:

Stochastic simulated annealing algorithm

```
begin initialize $T(k)$, $k_{max}$, $s_i(1)$, $w_{ij}$ for $i, j = 1, . . . , N$
    $k \leftarrow 0$
    do $k \leftarrow k + 1$
        do select node $i$ randomly; suppose its state is $s_i$
            $E_a \leftarrow -½ \sum^{\mathrm{Ni}}_{\mathrm{j}} w_{ij}s_is_j$
            $E_b \leftarrow -E_a$
            if $E_b < E_a$
                then $s_i \leftarrow -s_i$
                    else if $e^{-(E_b - E_a)/T(k)} >$ Rand[0, 1)
                        then $s_i \leftarrow -s_i$
        until all nodes polled several times
    until $k = k_{max}$ or stopping criterion met
return $E$, $s_i$, for $i = 1, . . . , N$
end
```

3- Characteristics of the SA algorithm

SA optimization algorithm is based on the concept of annealing in metallurgy, a technique involving heating and controlled cooling of a material. At first, bounds for the parameters to be optimized are imposed, and an initial model is generated accordingly. By analogy with the annealing physical process, each step of the SA replaces the current solution by a random “nearby” solution, chosen with a probability that depends on the cost function values and on a global parameter referred to as Temperature. The latter is decreased during the process following a predefined cooling scheme. The dependency is such that the current solution changes almost randomly when the temperature is large, but as the temperature goes to zero solutions with lower cost are favored[7]. Some of the characteristics discussed in this section are: 1) Temperature, 2) Temperature initial value, 3) Stopping criterion, 4) Temperature decrease, 5) Markov chain. These characteristics are briefly discussed below[16]:

Temperature: It is a control parameter that is sufficiently high, in order to skip high level gates of energy and sufficiently low to be attracted to the deepest minimum. The variation law of temperature is also important in order to test a maximum number of configurations and therefore to find the global minimum.

Temperature initial value: In [16], T_0 must be chosen so that the acceptance probability of the worst configuration would be equal to P_r = 80% after the maximal increase of the cost function $\Delta g+$ is fixed by the user. T_0 is obtained by the following expression:

$$T_0 = \Delta g+ / \ln (P_r) \qquad \rightarrow (5)$$

The following formula is also used in [16] to compute T_0:

$$T_0 = r^{*}. \max \Delta g \text{ such that } r >> 1 , r \sim 10 \qquad \rightarrow (6)$$

In these two expressions, it is too difficult to compute or estimate the value of $\Delta g+$ and max Δg for a real big dimension problem.

A third formula proposed that T_0 should be selected so that system transitions are all accepted at the beginning of the research algorithm. That means:

$$\exp(- \Delta g / T_0) \sim 1 \qquad \rightarrow (7)$$

Stopping criterion: The temperature decrease is stopped:

- By fixing the number of T_k variations for which the algorithm is run
- When two consecutive configurations are identical.
- When the temperature T_k is less than a given fraction of T_0 :

$$T_k < T_{ratio} \text{ with } T_{ratio} = 10^{-6} \text{ for example.} \qquad \rightarrow (8)$$

Temperature decrease: the temperature change from T_k to T_{k+1} is determined by static stability detection. The search is realized by iterating the Markov chain, which is generally configurations number tested at T_k .

Markov chain: The Markov chain is the set of finite random states composed of the probabilities set associated with every configuration visited at temperature T_k. When T_k is constant, the probability is homogeneous. If the transition number tends towards the infinite, the most probable state appears very often, we obtain the statistic stability at this temperature.

Finally, we conclude that: Like in other meta-heuristic, the convergence and efficiency of simulated annealing algorithm depend on [16]:

- A good choice of the neighborhood function.
- Method tram diversity.
- A good choice of algorithm parameters.
- The search space size.

4- The Annealing Schedules

Annealing is an operation in metal processing[15]. Metal is heated up very strongly and then cooled slowly to get a very pure crystal structure with minimum energy so that the number of fractures and irregularities becomes minimal. First, the high temperature accelerates the movement of the particles. During the cooling time the particles can find an optimal place within the crystal structure. While the temperature is lowered the particles subsequently lose the energy they were supplied with in the first stage of the process. Because of a thermodynamic, temperature-dependent random component some of them can reach a higher energy level compared to their previous energy level. These local energy fluctuations allow particles to leave local minima and reach a state of lower energy [8].

In metallurgy[7] and material science, annealing is a heat treatment of a material with the goal of altering some its properties like its hardness. Metal crystals have small defects like dislocations of ions which weaken the overall structure. By heating the metal, the energy of the ions is increased and, thus, their diffusion rate is increased. Then, the dislocations can be destroyed and the structure of the crystal is reformed as the material cools down and approaches its equilibrium state. When annealing a metal, the initial temperature must not be too low and the cooling must be done sufficiently slowly so as to avoid the system getting stuck in a meta-stable, non-crystalline, state representing a local minimum of energy.

In physics, the method for allowing a system such as many magnets or atoms in an alloy to find a low-energy configuration is based on *annealing*[26]. In physical annealing the system is heated, thereby conferring randomness to each component (magnet). As a result, each variable can temporarily assume a value that is energetically *un*favorable and the full system explores configurations that have high energy. Annealing proceeds by gradually lowering the temperature of the system — ultimately toward zero and thus no randomness — so as to allow the system to relax into a low-energy configuration. Such annealing is effective because even at moderately high temperatures, the system slightly favors regions in the configuration space that are overall lower in energy, and hence are more likely to contain the global minimum. As the temperature is lowered, the system has increased probability of finding the optimum configuration [29].

The act of annealing a metal so that it adopts the state of minimum energy can be thought of as a minimizing optimization problem, and as such it is interesting to use a similar technique for computational minimization. It is appealing to call the function of computational optimization using simulated annealing minimizing the energy.

Simulated annealing is a relatively straight forward algorithm which includes metropolis Monte Caro method[17]. The metropolis Monte Carlo algorithm is well suited for simulated annealing, since only energetically feasible states will be sampled at any given temperature. The simulated annealing algorithm is therefore a metropolis Monte Caro simulation that starts at a high temperature. The temperature is slowly reduced so that the search space becomes smaller for the metropolis simulation, and when the temperature is low enough the system will hopefully have settled into the most favorable state [8].

SA algorithm makes use of a number of parameters. The values of these parameters must be finely tuned; otherwise, inferior results are obtained frequently. The most important issue is the initialization of the temperature and the determination of the rate at which it should decrease [3]. On the other hand, setting the parameters of the SA-based algorithm determines the generation of the new solution. The precise rate of cooling is an essential part of SA as it determines if performance. A high cooling rate leads to poor results because of lack of enough representative states, while a low cooling rate requires high computation time.. First we show the general annealing method, then, the most interesting components of the annealing schedules.

4.1 General annealing method

Given a parameterized function of temperature in the following form

$$T(t) = \sum_{i=0}^{n-1} c_i \, f_i(t), \qquad \rightarrow (9)$$

Where constants c_i are the parameters being sought, and $f_i(t)$ functions are arbitrary continuous functions.

The derivative of equation (9) is expressed as:

$$\frac{dT(t)}{dt} = \sum_{i=0}^{n-1} c_i \frac{d f_i(t)}{dt}, \qquad \rightarrow (10)$$

In order to determine all parameters, preconditions must be set, such as

$$T(t = t_j) = T_j,\ j = 0,1,2,\ldots,n-1 \qquad \rightarrow (\ 11\ )$$

Where T_j is an arbitrary temperature value. Note that all T_j values are recommended to fall between T_{max} and T_{min}.

For simplicity, vector abbreviations are used. Let vector **t** = $\{T_0, T_1, T_2, \ldots.., T_{n-1}\}$ denote all the temperature values set by boundary conditions in (11). Let **c** = $\{c_0, c_1,\ldots, c_{n-1}\}$ denote the parameter vector, and let **F** = $\{f_i(t_j)\}$ be the matrix of function values at $t_0,t_1,t_2,\ldots,t_{n-1}$.

Parameters can be determined by solving the following linear equation system:

$$\mathbf{T = Fc} \qquad \rightarrow (\ 12\ )$$

Thus if **F** can be inverted and $\mathbf{F}^{-1}$ denotes the inverse

$$\mathbf{c = F^{-1}\ t\ .} \qquad \rightarrow (\ 13\ )$$

Equation (10) is to be used to get an iterative annealing function:

$$T(t = 0) = T_{max,}$$

$$T(t_{k+1}) = T(t_k) + \sum_{i=0}^{n-1} c_i\ \frac{df(t_k)}{dt} \qquad \rightarrow (\ 14\ )$$

Equation (14) gives the general schedule model.

4.2 The Annealing Schedules

The following choices must be made for any implementation of SA and they constitute the annealing schedule: the initial value of temperature (T), the cooling schedule, the number of iterations to be performed at each temperature, and the stopping criterion to terminate the algorithm.

4.2.1 Initial value of temperature (T)

Initial temperature is chosen such that it can capture the entire solution space. One choice is a very high initial temperature as it increases the solution space. However, at a high initial temperature, SA performs a large number of iterations, which may be without giving better results. Therefore, the initial temperature is chosen by experimentation depending upon the nature of the problem. The range of change, $\Delta\ f_0$ in the value of the objective function with different moves is determined. The initial value of temperature should be considerably larger than the largest $\Delta\ f_0$ encountered. One of the methods to select the initial temperature based on the initial acceptance ratio χ_0 , and the average increase in the objective function $\Delta\ f_0$ can be given by:

$$T = -\frac{\Delta f_0}{\ln(\chi_0)} \qquad \rightarrow (\ 15\ )$$

Where χ_0 is defined as the number of accepted bad moves divided by the number of attempted bad moves. Another method is defining χ_0 as the number of accepted moves divided by the number of attempted moves.

4.2.2 Cooling schedule

Cooling schedule determines the functional form of the change in temperature required in SA[21]. The earliest annealing schedules have been based on the analogy with physical annealing. Therefore, they set initial temperature high enough to accept all transitions, which means heating up substances till all the molecules are randomly arranged in the liquid. A proportional temperature is used, that is, $T(i + 1) = \alpha\, T(i)$, where α is a constant known as the cooling factor and it varies from 0.80 to 0.99. Finally, the temperature becomes very small and it does not search any smaller energy level. This is called the frozen state[27].

Three important cooling schedules are: logarithmic, Cauchy, and exponential. SA converges to the global minimum of the cost function if the temperature change is governed by a logarithmic schedule in which the temperature T(i) is given by $T(i) = T_0/\log i$. This schedule requires the move to be drawn from a Gaussian distribution. A faster schedule is the Cauchy schedule in which $T(i) = T_0 / i$. This schedule converges to the global minimum when moves are drawn from a Cauchy distribution. It is sometimes called 'fast simulated annealing'[25]. The fastest schedule is the exponential or geometric schedule in which $T(i) = T_0/\exp(- C_i)$ where C is constant[27].

A proportional temperature cooling schedule does not lead to equilibrium at a low temperature. Therefore, there is a need for a small number of transitions to be sufficient to reach the thermal equilibrium. However, recently few annealing schedules use information about the cost function obtained during the annealing run itself. Such a schedule is called an adaptive cooling schedule. An adaptive cooling schedule tries to keep the annealing temperature close to the equilibrium as well as reducing the number of transitions to reach equilibrium. It adjusts the rate of temperature decrease based on the past history of the run[27]. One of the adaptive cooling schedules uses the following formula:

$$T_{i+1} = T_i - \frac{1}{M_k}\frac{T_k^3}{\sigma^2(T_i)} \qquad \rightarrow \quad (16)$$

Where σ^2 is the variance of the objective function at equilibrium and M_k is given by

$$M_k = \frac{f_{\max} + T_i \ln(1+\delta)}{\sigma^2(T_i)\ln(1+\delta)} T_i \qquad \rightarrow (17)$$

Where f_{max} is an estimated maximum value of the objective function.

A second cooling schedule uses the following expression:

$$T_{i+1} = \frac{T_i}{1+(\frac{\ln(1+\delta)T_i}{3\sigma T_i})} \qquad \rightarrow (18)$$

Where δ is a small real number.

A third cooling schedule uses:

$$T_{i+1} = T_i \exp\left(- \frac{\lambda T_i}{\sigma(T_i)} \right) \qquad \rightarrow (19)$$

And a fourth cooling schedule uses:

$$T_{i+1} = T_i \left(1 - T_i \frac{\Delta(T_i)}{\sigma^2(T_i)} \right) \qquad \rightarrow (20)$$

Where λ is a quality factor and $\sigma(T_i)$ is the standard deviation of the temperature.

Other cooling schedules make a more direct appeal to the theoretical results on asymptotic convergence: An annealing schedule where there is only a single iteration at each temperature. A heuristic argument to derive a temperature function is of the form:

$$T_{i+1} = \frac{T_i}{1+BT_i} \qquad \rightarrow (22)$$

Where B is a constant.

4.2.3 Number of iterations

The number of iterations at each temperature is chosen so that the system is sufficiently close to the stationary distribution at that temperature which is known as ‘quasi-equilibrium’. Enough number of iterations at each temperature should be performed if temperature is decreased periodically. If lessthe number of iterations performed is too small, not all represented states will be searched and the algorithm will not be able to reach the global optimum. The value of the number of iterations depends on the nature of the problem[27].

4.2.4 Stopping criterion

Various stopping criteria have been developed with time. Here are a few examples: (1) Both the total number of iterations and the number of iterations at each temperature are given. This criterion leads to longer computation time without much update in *f* and sometimes it may lead to a local minimum due to an insufficient number of iterations. The number of iterations used by an algorithm depends on the complexity of a problem, which may not be known beforehand. (2) A minimum value of temperature and the number of iterations to move at each temperature are given. This idea relies on the fact that the chance of improvement in a solution is rare once the temperature is close to zero. At very low temperature, moves will be trapped in the neighbourhood of the current solution. (3) Both the number of iterations to move at each temperature and a predefined number of iterations to get a better solution are given[27].

5- Simulation Results

Before we analyze and compare the most famous annealing schedules used in the SA algorithm, we state the following points:

1. SA method resembles the cooling process of molten metals through annealing.
2. At high temperature, the atoms in the molten metal can move freely with respect to each other but as the temperature is reduced, the movement of atoms gets restricted.
3. The atoms start to get ordered and finally form crystals having the minimum possible energy.
4. The formation of a crystal mainly depends on the cooling rate.
5. If the temperature is reduced at a very fast rate, the crystalline state may not be achieved at all; instead the system may end up in a polycrystalline state, which has higher energy level than the crystalline state.
6. Therefore, in order to achieve the absolute minimum energy state, the temperature needs to be reduced at a low rate[19].

Annealing schedule is an essential part of SA as it determines the performance of the SA algorithm. In this paper, we investigate many of the parameters settings and annealing schedules to enhance the performance of this algorithm. These are shown below:

1- $T(k) = \alpha\ T(k-1)$ where $\alpha < 1$ → (S1)
Experience has shown that α should be between 0.8 and 0.99, with better results being found in the higher end of the range. Of course, the higher the value of α, the longer it will take to decrement the temperature to the stopping criterion.

2- $T(k) = (1-\alpha)\ T(k-1)\ ,\ 0 < \alpha < 1$ → (S2)
It is a linear annealing function which is the simplest form of the polynomial annealing functions interpolated between the points determined by the start temperature at the beginning of the annealing process, the ending temperature which at the end of the process, the maximum temperature, and the minimum temperature. Note that equations (S2) and (S1) are both linear functions and that both depend on the choice of α.

3- $T(k) = T(0) / k$ → (S3)
This is known as the Cauchy annealing. The Cauchy schedule is a faster schedule in which equation (S3) converges to the global minimum when moves are drawn from a Cauchy distribution. It is sometimes called 'fast simulated annealing'. It was noted that the Cauchy distribution has a "fatter" tail than the Gaussian form of the Boltzmann distribution, permitting easier access to test the local minima in the search for the desired global minimum.

4- $T(k) = T(0) / \ln(k+1)$ → (S4)
This is known as Boltzmann annealing (BA). It has the credit for being the first simulated annealing which is generally given to a Monte Carlo importance-sampling technique for doing large-dimensional path integrals arising in statistical physics problems. This method was generalized to fit non-convex cost functions arising in a variety of problems.

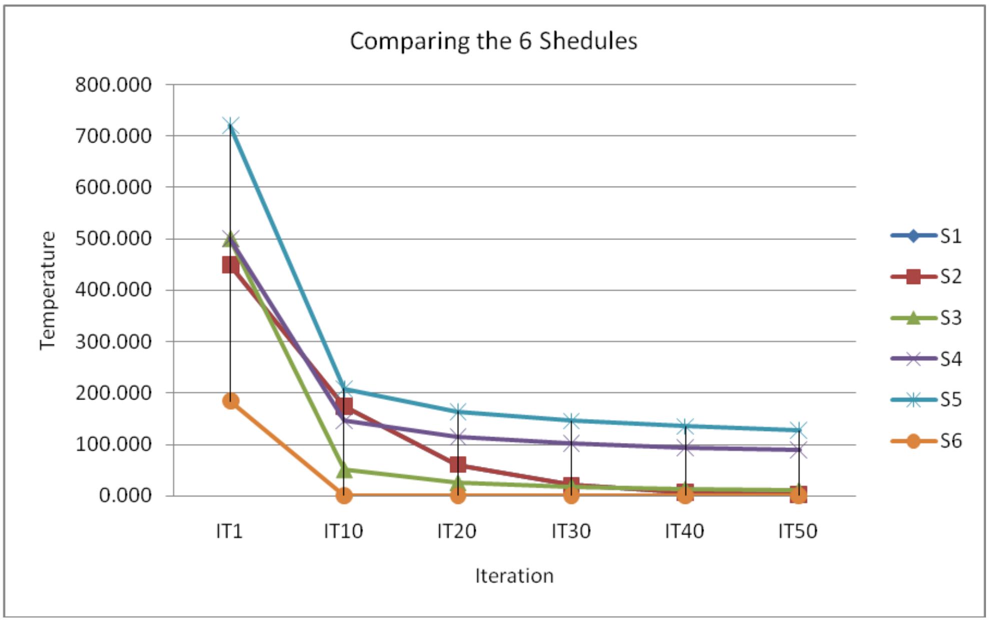
5- $T(k) = T(0) / \log(1+k)$ → (S5)
This is known as the logarithmic annealing schedule. Using this schedule SA converges to the global minimum of the cost function if temperature change is governed by a logarithmic schedule in which the temperature

T(k) at step k is given by equation (S5). This schedule requires the move to be drawn from a Gaussian distribution.

6. T(k) = T(0) / exp(k) → (S6)

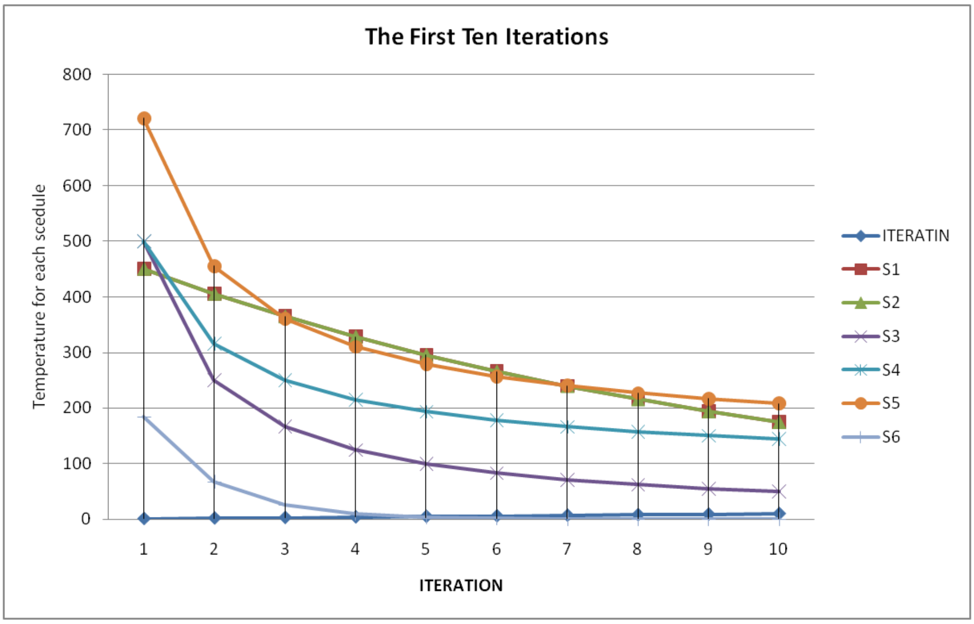
   This is known as the exponential or geometric annealing schedule. This schedule is the fastest schedule as shown in the results. There is no rigorous proof of the convergence of this schedule to the global optimum although good heuristic arguments for its convergence have been made for a system in which annealing state variables are bounded.

All the simulated data are performed on the same data sets and under the same circumstances. Figure (1) shows a comparison between the six schedules taking the iterations shown in the figure. The fast schedule to converge is (S6). (S3) comes second. The third and fourth schedules to converge are (S1) and (S2). Their relative arrangement depends on the choice of the value α. The following schedule is (S4) and the slowest one is (S5).



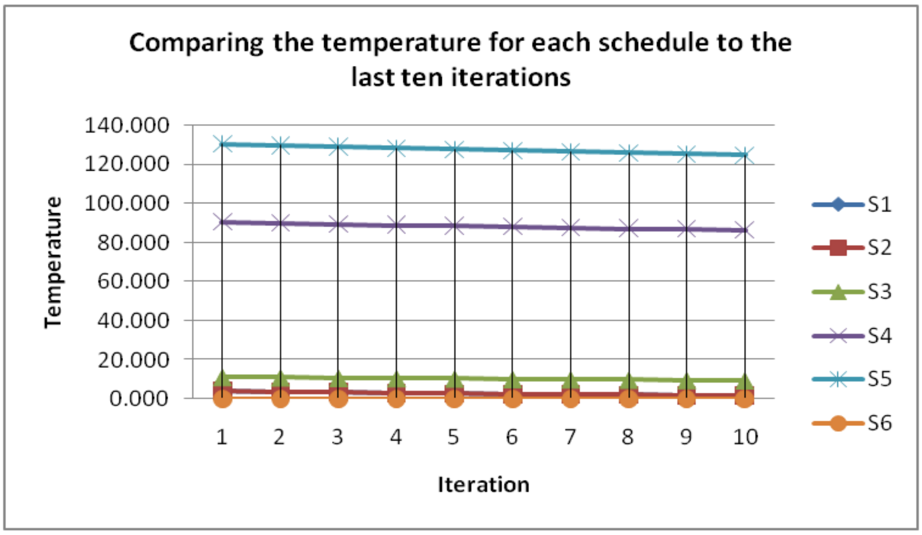
Figure(1): comparing the six schedules for the iterations shown.

Figure(2) shows a comparison between the temperature in all the six schedules in the first ten iterations. A great variation among the cooling schedules occurs during the first three iterations. After that a natural movement occurs in each schedule. Schedule (S5) has the largest temperature at the first and tenth iterations. On the other hand, schedule (S6) has the lowest temperature at the first and tenth iterations.



Figure(2) : comparing the temperature for each schedule to the first ten iterations.

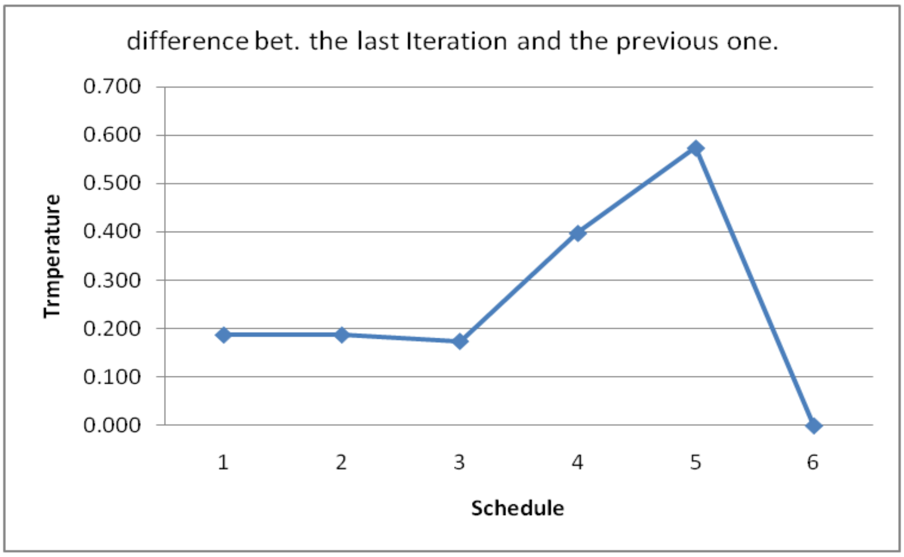
In figure (3), we can see that the annealing process of the schedule (S5) is better than the annealing process of the schedule (S4). (S3) Follows these two schedules. This is followed by the schedules (S1) and (S2). The last one is the annealing schedule (S6). On the other hand the schedule (S6) is the fastest one. (S1) and (S2) represent a fast annealing following (S6). These are followed by (S3). The annealing schedule (S4) is slower . The slowest annealing schedule is (S5).



Figure(3): Comparing the temperature schedule for the last ten iterations.

The fastest annealing schedule (S6) is called Quenching schedule, and the simulated annealing in this case is called simulated quenching. In order to guarantee an optimal solution a simulated annealing algorithm is preferred than a simulated quenching algorithm. But for some problems, we can do this with a faster annealing schedule, and in many cases when we do not need an optimal solution, a faster annealing may be more efficient. In natural cooling (or annealing), we can “find” the lowest energy state-crystallization. However, if we hasten the process by rapid cooling as shown in schedule (S6), the lowest energy state of amorphous will not occur.

Figure(4) shows the temperature difference between the last iteration and the previous one. The figure indicates that the difference in schedule (S5) is more than the difference in schedule (S4). Following these is (S1) and (S2) which are nearly on the same level. The temperature difference in schedule (S3) is even lower. The difference between the temperatures of the last two iterations of schedule (S6) reaches zero



Figure(4): the temperature difference between the last two iterations

Figure (5) shows the temperature difference (the behavior of the cooling schedule) between the first two iterations as follows:

In schedule (S1), the temperature difference = 450 – 405 = 45.

In schedule (S2), the temperature difference = 450 – 405 = 45.

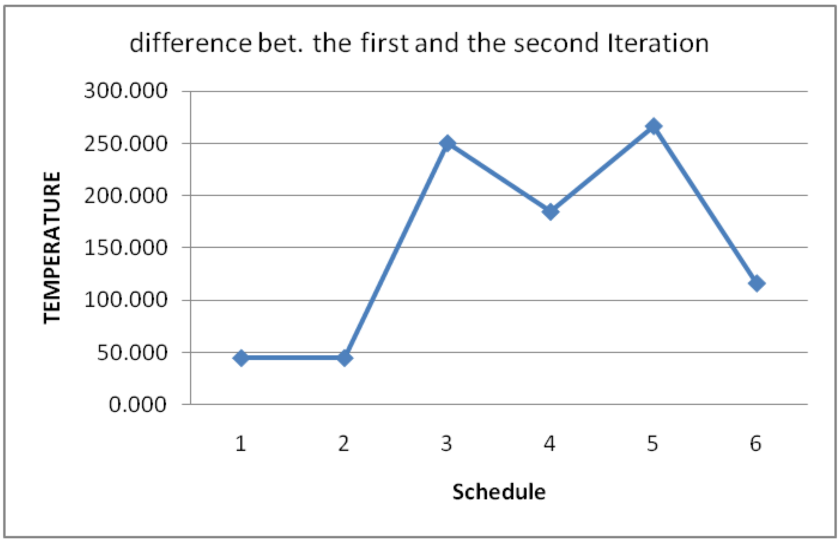
In schedule (S3), the temperature difference = 500 – 250 = 250.

In schedule (S4), the temperature difference = 500 – 315 = 185.

In schedule (S5), the temperature difference = 721 – 455 = 26 6.

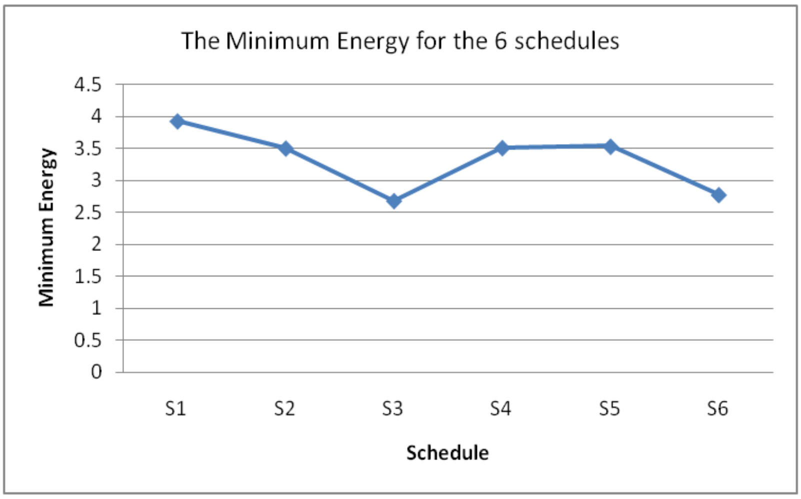
In schedule (S6), the temperature difference = 184 – 67 = 117.

These values indicate that the cooling schedule (S5) cools fastest in the beginning of the annealing process.. Schedule (5) is followed by schedule (S3) then (S4) then (S6). Finally (S1) and (S2) are on the same level.



Figure(5): the temperature difference between the first and the second iterations for the six schedules.

Figure(6) shows the behavior of the minimum energy reached by the six schedules. The minimum energy of schedule (S1) is the largest. The minimum energy of schedule (S3) is the lowest. Table (2) shows the numerical values of the minimum energy of all six schedules.



Figure(6) the Minimum Energy for the six Schedules.

Table (1) shows the temperature for each schedule at iteration (1) and at iteration (14) when the temperature reached 0 in schedule (S6).. In this table two temperature values are considered. In iteration (1) the value of the temperature using schedule (S5) = 721.348 while the starting temperature (atiteration zero) = 500. This means that an uphill occurs in this iteration. The second consideration is the temperature = 0.000 for schedule (S6), at iteration (14). No improvement can happen after that.

Table(1): The first iteration's temperature until temperature = 0.000 for S6.

IT	S1	S2	S3	S4	S5	S6
1	450.000	450.000	500.000	500.000	**721.348**	183.940
14	114.384	114.384	35.714	127.979	184.635	**0.000**

Table(2): The minimum energy for the six schedules

Schedule	S1	S2	S3	S4	S5	S6
M.E	**3.92476**	3.498277	**2.671506**	3.505392	3.528039	2.766086

Table(3) shows the temperature average and standard deviation of all iterations. Schedule (S6) has the lowest average temperature. This means that a fastest annealing occurs (simulated quenching). Schedule (S5) has the largest average temperature. This means that the annealing process is better than the others.

Table(3): The temperature average and standard deviation for all iterations of the six schedules.

Schedule	S1	S2	S3	S4	S5	S6
Average	83.052	83.052	42.365	125.875	**181.599**	**5.389**
SD	**126.491**	126.4907	97.69085	83.85988	105.75236	**71.71881**

Table(4) shows the range between the temperatures in iteration (1) and in iteration (54). The lowest range is in schedule (S6). In fact it is lower than the value mentioned in the table. The reason is that from iteration (14) to the last iteration, the temperature equals zero. On the other hand, the largest range is associated with schedule (S5).

Table(4): The temperature range from the first iteration to the last one.

	S1	S2	S3	S4	**S5**	S6
IT1	450.000	450.000	500.000	500.000	**721.348**	183.940
IT54	1.691	1.691	9.259	86.485	**124.771**	0.000
Range	448.309	448.309	490.741	413.515	**596.576**	**183.940**

This paper gives a comparison of the performance of simulated annealing. The analytical results among different annealing schedules are illustrated as follows:

1. **Dynamics analysis.** Simulated annealing is a generic probabilistic meta-heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete. The annealing process is divided into two stages: High temperature and low temperature.

   High temperature: This stage is a classic thermodynamics process. According to metropolis criteria, the algorithm accepts the worse solution at high probability. The algorithm's dynamics behavior is a random movement in the search space. In this stage metropolis criteria act as the driving force of the algorithm movement. We can use classic thermodynamics and Brownian motion theory to describe this stage. We call it classic dynamics stage.

   Low temperature: This stage is a quantum mechanics process. The Metropolis criteria restricts the solution free movement in the search space, because it accepts the worse solution at very low probability. Crystal lattice vibration theory could describe the algorithm's dynamics behavior at low temperature. In this stage energy becomes the main factor of the system. We call it quantum dynamics stage. At high temperature the algorithm system can be analyzed by classic thermodynamics; at low temperature the algorithm system can be analyzed by quantum mechanics.

   From Figures(1,2,3) and Tables(1,4), we find that the exponential or geometric annealing schedule (S6) is the fastest to make the simulated annealing algorithm change from a classic thermodynamics process to a quantum mechanics process. Immediately following that schedule is the Cauchy annealing schedule (S3) although there is a great difference between the initial temperatures of the two schedules. The linear annealing schedules represented in (S1 and S2) come in the third stage. The Boltzmann annealing schedule (S4) comes fourth. The logarithmic annealing schedule (S5) is the last one in this regard. The rate of change from classic thermodynamics to the quantum mechanics is also different from schedule to schedule. For example, as shown in Figure(1), the rate of change (behavior) of the temperature from iteration 1 to iteration 10 differs from the behavior of temperature from iteration 10 to iteration 20 for the stated schedules. S2 tends to decrease from a certain temperature to a lower one more rapidly than S4. All the schedules tend to reach a lower temperature than S4 and S5 starting from iteration 20 towards iteration 50. Figure(3) shows that all schedules oriented the simulated annealing algorithm to the quantum mechanics process within a certain limit.

2. **Mathematical analysis.** Mathematical analysis studies functions and their generalizations. Functional analysis is the part of modern mathematical analysis in which the basic purpose is to study functions. In its most general form such a study falls into three parts: 1) the introduction and study of infinite-dimensional spaces, 2) the study of the simplest functions, and 3) the study of general functions. The paper discusses the general functions that illustrate the general annealing schedules in section (4.1). The specific annealing schedules are shown in sections (4.2) and (5).

3. **Statistical analysis.** Statistical manipulation is often necessary to order, define and/or organize raw data. Basic statistical analysis involves computing the mean and standard deviation. The mean is the average of all the values obtained. The Standard deviation is a widely used measurement of variability or diversity used in statistics and probability theory. It shows how much variation or “dispersion” there is from the “average”. A low standard deviation indicates that the data points tend to be very close to the mean, whereas a high standard deviation indicates that the data are spread out over a large range of values. In addition to expressing the variability of a population, standard deviation is commonly used to measure confidence in statistical conclusions. In science, researchers commonly report the standard deviation of experimental data, and only effects that fall far outside the range of standard deviation are considered statistically significant.

   Table(3) shows the mean value and standard deviation of the temperature for all the mentioned schedules. In this table S6 (the exponential or geometric annealing schedule) has the minimum mean value for the iterations required. This means that the process of annealing takes minimum number of iterations which indicates that this is the fastest annealing schedule (simulated quenching occurs). Also S6 has the lowest standard deviation. Following S6, S3 (the

Chauchy annealing schedule) has the second smallest mean value and standard deviation. The 3rd is the linear annealing schedules represented by S1 and S2 although their standard deviation is the highest which means that the data is spread out over a large range of values. Following these schedules we get tothe mean and standard deviation of the Boltzmann annealing schedule which are less then counterparts for the logarithmic annealing schedule.

4. **A quantitative study** is presented for the typical behavior of the simulated annealing algorithm based on the mentioned annealing schedules. The study is based on the analysis of numerical results obtained by applying the algorithm on a simulated data and are shown through the figures and tables in this section.
5. Each annealing schedule is derived from a certain **statistical distribution** that has different characteristics. The details of these characteristics are beyond the scope of this paper.

6- Discussion and Conclusions

For optimization problems where derivatives of the cost function are not available, stochastic methods like Simulated Annealing can be applied[15]. SA-based algorithms have attractive and unique features when compared with other optimization techniques. Firstly, a solution does not get trapped in a local minimum or maximum by sometimes accepting even the worse move. Secondly, configuration decisions proceed in a logical manner[27]. SA can narrow the field of search and speed up the rate of convergence continually in the optimization process. The true strength of SA lies in its ability to statistically deliver a true global optimization.

In this paper different versions of annealing schedules are used to optimize the performance of the algorithm. From the study, we state the following conclusions:

1. The exponential annealing schedule is the fastest to converge. It provides the fastest annealing schedule. It is followed by the Cauchy annealing schedule, which also provides a fast simulated annealing. The linear schedules follow. The Boltzmann annealing schedule comes fifth. The logarithmic annealing schedule is the slowest to converge
2. The annealing process of the logarithmic annealing schedule is better than the annealing process of the Boltzman annealing which is better than the annealing process of the Cauchy annealing schedule. Following the above comes the linear annealing schedules. The exponential annealing schedule has the worst annealing process.

3. At the beginning, the logarithmic annealing schedule has the fastest cooling schedule. It is followed by the Cauchy annealing schedule, then the Boltzman annealing schedule. The linear schedules are the slowest at the beginning of the process.
4. The minimum energy of the Cauchy annealing schedule is the lowest. It is followed by the exponential annealing schedule, then by the linear annealing, then by the Boltzman annealing schedule, and finally by the logarithmic annealing schedule.
5. The temperature average and standard deviation of the annealing process for all the annealing schedules are calculated indicating that the exponential annealing schedule has the lowest mean value, the lowest standard deviation, and the smallest temperature range.

Using simulated annealing it has been proven that it is possible to converge to the best solution.

References

1- Czaplicki J., Cornelissen G., and Halberg F.”GOSA, a simulated annealing-based program for global optimization of nonlinear problems, also reveals transyears”. Journal of APPLIED BIOMEDICINE 4:87-94, 2006.

2- Shi H., and Li W.”Evolving Artificial Neural Networks Using Simulated Annealing-based Hybrid Genetic Algorithms”. JOURNAL OF SOFTWARE, VOL.5, NO.4, 2010.

3- Kundu S., Mahato M, Mahanty B., and Acharyya S. “Comparative Performance of Simulated Annealing and Genetic Algorithms in Solving Nurse Scheduling Problem”. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I.

4- ERDOGMUS P, OZTURK A., and TOSUN S.: “CONTINIOUS OPTIMIZATION PROBLEM SOLUTION WITH SIMULATED ANNEALING AND GENETIC ALGORITHMS”. 5th International Advanced Technologies Symposium (IATS’09), 2009, Karabuk, Turkey.

5- Nair T., and Sooda K.: “Comparison of Genetic Algorithm and Simulated Annealing Technique for Optimal Path Selection In Network Routing”. NCVN-09 october 2009 KCG College of Technology.

6- Yoshikawa M., Yamauchi H., and Terai H.: “Hybrid Architecture of Genetic Algorithm and Simulated Annealing”. Engineering Letters, 16:3, EL_16_3_11, 2008.

7- Manconi A., Tizzani P., Zeni G., Pepe S., and Solaro G.: “Simulated Annealing and Genetic Algorithm Optimization using COMSOL Multiphysics Applications to the Analysis of Ground Deformation in Active Volcanic Areas”. Proceedings of the COMSOL Conference 2009 Milan.

8- Tamilarasi A and Kumar T.: “An enhanced genetic algorithm with simulated annealing for job-shop scheduling”. International Journal of Engineering, Science and Technology Vol.2, No.1, 2010.

9- Chen D., Lee C., Park C., and Mendes P.: “Parallelizing simulated annealing algorithms based on high-performance computer”. Journal of Global Optimization, 20007.

10- Alan R., McKendall J, Shang J., and Kuppusamy S.: “Simulated annealing heuristics for the dynamic facility layout problem”. Computers & Operations Research 33 (2006).

11- Bai R., Burke E., and Kendall G.: “A Simulated Annealing Hyper-heuristic for University Course Timetabling”. PATAT 2006, pp. 345-350.

12- Chen P., and Shahandashti S.: STOCHASTIC SCHEDULING WITH MULTIPLE RESOURSE CONSTRAINTS USING A SIMULATED ANNEALING-BASED ALGORITHM”. The 25th International Symposium on Automation and Robotics in Construction. (ISARC – 2008).

13- Bertsimas D., and Tsitsiklis J.: “Simulated Annealing”. Statistical Science, Vol. 8, No. 1, 1993.

14- Bisht S.: “Hybrid Genetic-simulated Annealing for Optimal Weapon Allocation in Multilayer Defence Scenario”. Defence Science Journal, Vol 54, No. 3, 2004.

15- Brauer H., and Ziolkowski M.: “Magnet Shape Optimization Using Adaptive Simulated Annealing”. ELEC. ENERG. Vol. 19, no. 2, 2006.

16- Ghoul R., Benjelloul A., Kechida S., and Tebbikh H. “ A Scheduling Algorithm Based on Petri Nets and Simulated Annealing”. American Journal of Applied Sciences 4(5) , 2007

17- Widita R.: “Projection Correlation Optimization using Simulated Annealing in CT Image Reconstruction”. Indonesian Journal of Physics. Vol 17 No. 4 , 2006.

18- Chakrapani Y., and Rajan K.: “ Hybrid Genetic-Simulated Annealing Approach for Fractal Image Compression”. International Journal of Computational Intelligence 4; 4, 2008.

19- Venkatesan R., and Naryanasamy R.: “Application of genetic algorithm and simulated annealing for optimization of extrusion die ration and die cone angle”. 12th International Scientific Conference Achievements in Mechanical & Materials Engineering, 2003.

20- Saha S., and Bandyopadhyay S. : “ A New Multiobjective Simulated Annealing Based Clustering Technique Using Stability and Symmetry”. IEEE, 2008.

21- Andricioaei I., and Straub J. : “ Generalized simulated annealing algorithms using Tsallis statistic: Application to conformational optimization of a tetrapeptide”. Physical Review E, volume 53, Number 4, 1996.

22- Ho S., Yang S., Wong H., and Ni G.: “ A Simulated Annealing Algorithm for Multiobjective Optimizations of Electromagnetic Devices”. IEEE Transactions on Magnetics, Vol. 39, No. 3, 2003.

23- Simopoulos D., Kavatza S., Vournas C. : " Unit Commitment by an Enhanced Simulated Annealing Algorithm". Paper no. TPWRS- 00234-2005. PSCE, 2006.

24- Efstratiadis A., and Koutsoyiannis D. : "An evolutionary annealing-simplex algorithm for global optimization of water resource systems". Hydroinformatics 2002: Proceedings of the Fifth International Conference on Hydroinformatics, Cardiff, UK, 2002.

25- Ingber L. : "Adaptive simulated annealing (ASA): Lessons learned". J Control and Cybernetics, 1995.

26- Haidine A., and Lehnert R. : "Multi-Case Multi-Objective Simulated Annealing (MC-MOSA): New Approach to Adapt Simulated Annealing to Multi-objective Optimization". International Journal of Information Technology 4:3 2008.

27- Suman B., and Kumar P. : "A survey of simulated annealing as a tool for single and multiobjective optimization". Journal of the Operational Research Society 57, 2006.

28- Hiroyasu T., Mike M., and Ogura M. : "Parallel Simulated Annealing using Genetic Crossover". Proceedings of the IASTED International Conference PARALLEL AND DISTRIBUTED COMPUTING AND SYSTEMS, 2000.

29- Duda R., Hart P., and Stork D. : "Simulated Annealing" PATTERN CLASSIFICATION, Second Edition, 2001.

30- Eldos T. : " Simulated Annealing with Deterministic Decisions". Journal of Computer Science 5(12): 977-982, 2009.