Proceedings of DETC'97
1997 ASME Design Engineering Technical Conferences
September 14–17, 1997, Sacramento, California

DETC97/DAC-3980

IMPROVING THE EFFICIENCY OF SIMULATED ANNEALING OPTIMIZATION THROUGH DETECTION OF PRODUCTIVE SEARCH

Simon Szykman
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Building 304, Room 12
Gaithersburg, MD 20899

Linda C. Schmidt and Harshawardhan Shetty
Department of Mechanical Engineering
University of Maryland
College Park, MD 20742

ABSTRACT

The popularity of simulated annealing for engineering design applications has grown in recent years, increasing the need for new techniques that improve algorithm performance. Simulated annealing is a time-consuming, iteration-intensive algorithm. One area of algorithm enhancement with high potential impact is the development of methods for improving the algorithm by reducing the amount of wasted or non-productive search. This paper presents an approach to detection of productive search based on statistical process control (SPC) concepts. The proposed Detection of Productive Search (DPS) annealing schedule is compared to three other viable schedules using a 100-city traveling salesman problem. The DPS schedule produces results on par with the best from the more traditional schedules but does so with significantly fewer iterations.

1 INTRODUCTION

Simulated annealing is a global stochastic optimization technique enjoying increased use in recent years in mechanical design. Driving the growth in simulated annealing's popularity is its ability to find optimal or near-optimal solutions to problems with ill-behaved objective functions which resist optimization by traditional gradient-based techniques. Because realistic engineering problems are often characterized by poorly-behaved objective functions, simulated annealing has a broad range of use. A sampling of recent applications of simulated annealing includes blank nesting (Jain et al., 1992), tolerance design (Zhang and Wang, 1993) truss design (Shea and Cagan, 1995), feature recognition (Dong and Vijayan, 1996), machine design (Schmidt and Cagan, 1996), assembly design (Kim and Szykman, 1996), tube routing (Szykman and Cagan, 1996), mechanism synthesis (Ullah and Kota, 1996), manufacturability improvement (Xue, 1996), and component layout (Szykman and Cagan, 1997).

In order to find globally optimal or near-optimal solutions, simulated annealing performs many iterations, tens of thousands to millions depending problem size. Because simulated annealing is a computationally intensive algorithm, there is a strong need to avoid non-productive search. Minimizing wasted search is accomplished through proper selection of various parameters that control the optimization; these parameters will be discussed further in Section 3.

Parameter values for simulated annealing are traditionally determined empirically by repeatedly running the algorithm with different values until the quality of the solutions produced by the algorithm ceases to improve. Furthermore, even when an algorithm is tuned to produce good results, it still can be wasting search. Without a measure of productive search, one is never sure that equally good results could not have been obtained more rapidly using an untried combination of parameter values.

This paper proposes an approach toward improving algorithm efficiency through detection of productive search based on statistical process control concepts. Productive search is defined as search where the objective function value is changing rapidly in general, though not necessarily with every iteration. During productive search, the average change in objective function value over multiple iterations is high. The concepts of "rapid change" and "high average" are problem dependent. They can, however, be defined in a statistical manner by examining trends in average objective function values over a series of iterations and observing their behavior with respect to means and standard deviations of those averages.

Detecting productive search provides control over the simulated annealing optimization process instead of spreading computational effort uniformly throughout the optimization. By allowing the algorithm to do more iterations when search is productive and fewer iterations when it is less productive, wasted effort is reduced and the algorithm converges more quickly. Unlike other approaches to controlling the simulated annealing optimization, this approach is generic and requires neither *a priori* knowledge about the design problem nor extensive experimentation to tune algorithm parameters for each new problem.

The next section introduces the nomenclature that will be used in the remainder of the paper. Section 3 briefly describes the simulated annealing algorithm and explains the role of the *temperature* parameter and the *annealing schedule* in the optimization. In Section 4, productive search is defined and our approach to detection of productive search is presented. Section 5 describes experimental results obtained using the Detection of Productive Search (DPS) annealing schedule, as well as comparisons with other annealing schedules. Conclusions and a discussion of areas for future work follow in Section 6.

2 NOMENCLATURE

The following definitions apply to terminology used in this paper:

- n, the batch size, or number of random samples taken from a population of interest, P,
- m, the number of samples of size n drawn from a population of interest, P,
- X_{ij}, the value of random variable "*j*" in the "*i*th" sample taken from P,
- $\overline{X}_i = \sum_{j=1}^{n} \frac{X_{ij}}{n}$, the mean of the "*i*th" sample drawn from P,
- $\overline{\overline{X}} = \sum_{i=1}^{m} \frac{\overline{X}_i}{m}$, an estimate for mean of the overall population of sample means (which is not the same population as P),
- $s_i = \sqrt{\frac{\sum_{j=1}^{n} (X_{ij} - \overline{X}_i)^2}{n-1}}$, the standard deviation of the "*i*th" sample (an estimate for standard deviation of the population, P).
- $s_{e_i} = \frac{s_i}{\sqrt{n}}$, an estimate of the standard deviation of the population of sample means (again, not the same as P). This parameter is also called the standard error.

3 BACKGROUND

Simulated annealing is a stochastic optimization technique that was introduced by Kirkpatrick et al. (1983). At the start of an optimization using simulated annealing, the algorithm begins at an initial design state. The algorithm then takes a step to a new design state by randomly perturbing the current design. The objective function value of the new state is compared to that of the previous state. If the new state is better than the previous one, it is accepted; if it is worse, it is accepted or rejected with some probability. The probability of accepting an inferior state is a function of a parameter called *temperature*. The probability is given by:

$$P_{accept} = e^{-\frac{\Delta C}{T}}, \tag{1}$$

where ΔC is the change in the objective function due to the move, and T is the current temperature.

Both the temperature and the probability of accepting inferior steps begin high. Since many inferior steps are accepted, this results in near-random exploration to find promising regions of the design space. As the optimization proceeds, the temperature decreases and fewer inferior steps are accepted, making the search more directed. As the temperature continues to decrease, algorithm behavior resembles downhill search because virtually no inferior steps are accepted. This allows the algorithm to converge to a local optimum in the current region of the design space.

An *annealing schedule* determines the initial temperature, how many iterations are done at each temperature, how and when the temperature decreases, and sets the algorithm termination condition which determines how many temperatures are used. The "best" values for annealing schedule parameters vary from one problem to the next. There is no unique fixed annealing schedule that will lead to good performance for a variety of problems.

In simulated annealing, wasted search can stem from three possible causes:

- *Extended random search* is caused by starting the initial temperature too high. Since the probability of accepting inferior steps (i.e., steps away from a local optimum) is a function of temperature, this can lead to completely random search early on rather than search which tends to converge towards optimal solutions.
- *Excessive search at a temperature* occurs when too many iterations are performed at a given temperature. Because of the non-zero probability of accepting uphill steps at any temperature, there is a limit to how close to an optimum the algorithm can converge at a given temperature. Beyond that point, additional search at a temperature results in non-productive search, even if the algorithm has not converged to an optimum.
- *Delayed termination* is a result of performing search over too many temperatures. Once the algorithm has converged to an optimum or once the probability of accepting inferior steps becomes sufficiently small, the improvements made by search at additional temperatures are small enough that allowing the algorithm to continue does not significantly improve the objective function values. At this point, the algorithm should be terminated.

Of these three causes of wasted search, extended random search and delayed termination are the easiest to avoid. A good starting temperature can be determined either using a small amount of trial-and-error experimentation with different starting values, or it can be calculated as a function of a target initial acceptance probability and the standard deviation obtained by generating a sample of random solutions from the design space (White, 1984). Determining an appropriate termination condition to avoid delayed termination can be done simply in a variety of ways, such as by looking for a certain number of successive rejections, or a certain number of iterations without accepting any inferior steps.

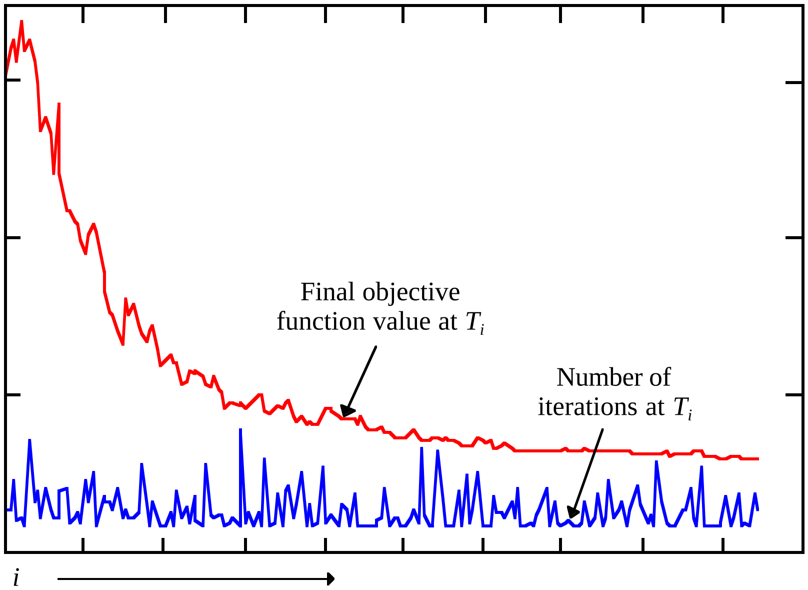

Figure 1. Illustration of Inefficient Search.

Dealing with excessive search at a temperature is more problematic. It is difficult to detect when productive search ends since (with the exception of final convergence) objective function values tend to fluctuate up and down due to the stochastic nature of the algorithm. The problem is compounded by the fact that the "right" number of iterations at one temperature may be too many or too few at another. This makes it impossible to determine the ideal number of iterations at every single temperature in advance. Too few iterations at each temperature can lead to convergence to poor local optima, so the typical approach is to err on the conservative side by performing a larger number of iterations than needed; wasted search is preferable to bad results.

This concept of wasted search is illustrated in Figure 1, which contains data from an "inefficient" optimization. The X axis represents T_i, the "ith" temperature in the optimization. The figure shows two curves, the upper curve being a series of final objective function values at the end of each successive temperature for the entire run, and the lower curve showing the number of iterations performed at each temperature, both plotted as a function of T_i. The top curve shows the objective function values starting out high and following a generally decreasing trend with a few increases representing uphill moves (i.e., moves away from a local minimum) that simulated annealing can make to escape from poor local optima. Clearly, the most productive search occurs in the first 15% of the temperatures, followed by a moderately productive phase for the next 40% to 50% of the temperatures. The final portion of the optimization is the least productive but still noticeably decreases the objective function value.

The relative magnitude of the values on the Y axis for the two curves is unimportant. What is important is the appearance of the lower curve (number of iterations) at different stages of the optimization. The greatest number of iterations would ideally be done at temperatures where search is most productive, and the fewest where search is least productive. For the inefficient search shown in the figure, there is no relationship between search productivity and the number of iterations done at a temperature. The dips in the lower curve in the early stages of the optimization represent lost opportunities; more iterations could have resulted in decreasing the objective function more rapidly early on, causing fewer iterations to be required later. The peaks in the latter stages do not have a significant impact on the objective function, indicating that more search was done than was necessary.

Excessive search at a temperature has the greatest potential for negative impact on the efficiency of the algorithm. Performing too many iterations at each temperature compounds wasted computational effort throughout the entire optimization. Due to the large number of iterations required by the simulated annealing algorithm, even small improvements in the efficiency of the algorithm can lead to significant improvements in run time. The following section describes an approach to increasing the efficiency of simulated annealing through detection of productive search.

4 IMPROVING SEARCH EFFECTIVENESS

An analogy can be drawn between the idea of productive search and concepts from statistical process control theory. Specifically, there is an inverse relationship between detecting productive search at a given temperature during a simulated annealing process and detecting *statistical control* for the process at that temperature. A manufacturing process in a state of statistical control will exhibit only random variation in process variables. Statistical process control principles are used to signal when control of the process achieved and to monitor the state of a process to indicate when statistical control is lost.

Statistical control is desirable for a manufacturing process but not for a stochastic search technique. During a simulated annealing process, a period of search marked only by random variation in the current solution state's objective function evaluation (i.e., *a process in control*) is *non-productive* because there is no bulk improvement in the solution value. Periods of productive search in a simulated annealing algorithm are indicated by marked variation in objective function evaluations over a range of successive iterations. This variation may be either hill climbing or descent (as can be seen in Figure 1); either behavior indicates that the algorithm is progressively moving through the space to find a region of improving solutions.

When a simulated annealing process is found to be in control, a change in algorithm parameters should be triggered so that a state of productive search can again be achieved. The Detection of Productive Search (DPS) annealing schedule proposed in this paper uses statistical process control principles to signal when the search at a temperature becomes unproductive and reduction in annealing temperature is needed. The DPS annealing schedule is introduced in this section, following a description of the statistical process control methods adapted for the DPS approach.

4.1 SPC's $\overline{X}$ Charting

Statistical process control (SPC) is an analysis method for the investigation of process variation and corresponding causes. One very common set of SPC tools are control charts, developed by Dr. W. A. Shewhart (DeVor et al., 1992; Wetherill and Brown, 1991). They are designed to detect process variation caused by events other than the random variation expected in a well-behaved process. The Shewhart $\overline{X}$ Chart is used to detect shifts in the average level of a process. To do so, random samples of size n are taken from the process, key parameter measurements are made (X_{ij}) and the means of the sample parameter measurements ($\overline{X}_i$) are plotted on a chart.

The Shewhart $\overline{X}$ Chart plotting area is divided by *action limits* into regions signaling *in-control* and *out-of-control* behavior. Action limit lines are typically set at $\overline{\overline{X}} \pm 3s_e$. These limits are calculated based on a series of samples and bound the region in which a process in statistical control is expected to operate. If a process is producing a target average value, $\overline{X}$, and exhibiting only naturally occurring variation, there is a 99.74% probability of enclosing sample mean values ($\overline{X}_i$) inside the action limits. If this process then produces a sample mean outside of the action limits, the process is assumed to be out of statistical control and the operators are instructed to look for other causes that must be corrected to bring the process back on target. An $\overline{X}$ Chart also typically includes *warning limit* lines set at $\overline{\overline{X}} \pm 2s_e$. An on-target, in-control process has a 95.44% probability of producing sample means within the warning limit lines. These lines bound a range of sample mean values that may signal the beginning of out-of-control behavior. This behavior is signaled when two consecutive sample means fall outside of the same warning limit line.

Non-random patterns of plotted sample means on an $\overline{X}$ Chart also signal out-of-control behavior even if they fall inside the established action limits. Such patterns include: cycles, repeating means (usually 6 or 7) on the same side of the target average value $\overline{\overline{X}}$, or a extended series of plotted points (usually 6 or 7) either increasing or decreasing. A process is considered to be in statistical control while there are no out-of-control indications.

4.2 Applying $\overline{X}$ Charting Concepts to the Detection of Productive Search at a Given Temperature

Applying the $\overline{X}$ Chart technique to the detection of productive search at a given simulated annealing temperature involves three steps and several associated parameters shown in parentheses and defined below:

1. Implementing a batch sampling process (sampling frequency, *n*).
2. Calculating and updating action and/or warning limits data ($\overline{X}, \overline{\overline{X}}, s_e, m$).
3. Checking for out-of-control behavior.

Step 1: Sampling process. The frequency of sampling is set to 1, meaning that the value of the objective function for every iteration becomes a point in the current batch. A batch size, *n*, of 15 (determined empirically) is used to calculate each sample mean.

Step 2: Maintaining action and/or warning limits. Routines for the calculation of standard statistics $\overline{X}$, $\overline{\overline{X}}$, and s_e are implemented. The number of samples means, *m*, used to calculate each value of $\overline{\overline{X}}$ and s_e is 10. Because the optimization is progressively minimizing an objective function, mean, standard deviation, and standard error values drop with time. Therefore these values and the action and/or warning limits must be updated periodically. Our scheme for detection of productive search uses a "moving window" meaning that whenever $\overline{\overline{X}}$ and s_e are updated, they are calculated based on the most recent 10 sample means. Note that at the beginning of each temperature, a full window of 10 observations is collected to calculate values of $\overline{\overline{X}}$ and s_e for the initial action and/or warning limits, rather than using the most recent means which were from a different temperature.

Step 3: Testing for out-of-control behavior. Not all of the tests for out-of-control behavior used in SPC $\overline{X}$ Charts are well-suited for our purposes. For example, in SPC applications, one sample mean outside of the action limits is a clear signal of an out-of-control process. Experiments were performed to determine which of the traditional SPC tests were appropriate. The two which were found to be most successful are:

- Two out of three successive points outside of warning limits represented by $\overline{\overline{X}} \pm 2s_e$, and
- A run of six consecutive points up or down (even if inside the warning limits).

Once warning limits are calculated, the process is tested for out-of-control behavior at the end of each batch of iterations. The batching and testing repeats until an out-of-control signal is found according to either of the two tests, or until 10 batches are tested without signaling an out-of-control condition. If an out-of-control condition is found, new values for $\overline{\overline{X}}$ and s_e are calculated for the most recent 10 batches. The warning limits are updated, a new window is started, and search continues at the current temperature. Otherwise, the process is considered to be in control, indicating that search is non-productive. In this case, the temperature is reduced, beginning the testing for non-productive search again at the new temperature.

The DPS process parameters of sampling frequency, n, and m were set at levels that may be considered high for statistical sampling in general practice. In manufacturing process control, because evaluation (i.e., inspection) of every part can be prohibitively expensive, random sampling as well as smaller batch sizes and numbers of batches are often used instead of 100% inspection. With a numerical optimization, the current design is already evaluated by the objective function each iteration. Because calculation of statistics involves a minor amount of computation in comparison, the approach used here bases statistics on every iteration.

4.3 Statistical Validity of the Approach

The $\overline{X}$ Chart techniques described in Section 4.1 here are firmly grounded in statistical theory, but their rigor depends upon four assumptions:

- The sample group sizes are equal.
- All sample groups are weighted equally.
- The parameter of interest is, at least approximately, Normally distributed.
- The sample observations are independent of each other.

The first two assumptions are satisfied since the size and weighting of samples are easy to fix within sampling method. Although other proposed annealing schedules have relied on the third assumption (e.g., Huang et al., 1986; Lam and Delosme, 1988); experiments have shown that this assumption is generally not true for the simulated annealing optimization process (Schmidt and Cagan, 1996). However, unlike other annealing schedules, the DPS schedule makes use of the standard $\overline{X}$ Chart practice of batching observations and working with the means of batch means rather than means of individual values. This mitigates the problem of non-normality because of the Central Limit Theorem (Devore, 1987). One of the implications of the theorem is that the *means* of random samples of a population are always Normally distributed regardless of the original distribution.

Counter to intuition, the most important $\overline{X}$ Chart assumptions are those concerning the nature and handling of the samples (Wetherill and Brown, 1991). It can be correctly argued that a simulated annealing process will not produce objective function evaluations of consecutive states that are independent of each other. During simulated annealing, the probability of reaching a state having a given objective function value relies on the value of the current state. Any series of values of consecutive states will exhibit positive autocorrelation.

The risk of using $\overline{X}$ Charts on autocorrelated data is that the autocorrelation introduces additional variation that might be misinterpreted as the presence of variation due to assignable causes in a process that really is in statistical control. The chart is likely to lead to false alarms, particularly with the use of standard warning limits (Wetherill and Brown, 1991). In standard SPC uses, false alarms will send the process operator on wild goose chases looking for phantom assignable causes. In this DPS application, a tendency to trigger an out-of-control condition means the search will act as if it were productive when it may not be, extending the search at the current temperature longer than may be necessary. Continuing search when faced with ambiguous search signals is a more conservative course of action since additional search will not negatively impact the quality of solutions whereas insufficient search can. As a result, using $\overline{X}$ Chart strategies with autocorrelated data does not impair the DPS approach. Alternatively, statistical process control methods tailored for autocorrelated data can be modified for the DPS application and these are mentioned in our comments on areas for future work.

5 APPLICATION OF DETECTION OF PRODUCTIVE SEARCH TO SIMULATED ANNEALING

The key to the DPS approach algorithm efficiency improvement is detecting the end of productive search to reduce excessive iterations at a given temperature. To test the value of the DPS annealing schedule, it was applied to a traditionally challenging optimization problem. The DPS schedule results were then compared to those from three other annealing schedules

5.1 The Traveling Salesman Problem

The traveling salesman problem (TSP) is often used to test the effectiveness of combinatorial optimization algorithms because it is an NP-complete problem. In the TSP, the task is to determine the minimum-length closed tour by which a salesman can travel to each of a number of cities, stopping at each city only once, and returning to the start city. The problem used in this analysis is a 100-city TSP published by Krolak et al. (1971). Problem input is the Cartesian coordinate location of each of the cities. A valid solution to the problem is a list of cities in the order in which the salesman will travel. The objective function value or "cost" of any solution is the Euclidean length of the closed tour. The best solution found by Krolak et al. is a tour of length 21,282. It was generated by a hybrid, heuristic and computational technique which was not automated but required iterative interaction between the human and computer. The globally optimal solution to the problem is not known.

The simulated annealing algorithms tested here all operate using the same procedure to generate potential tours. An initial tour is randomly selected and becomes the initial starting point for the simulated annealing algorithm. All subsequent tours are produced by perturbing the previous accepted solution. Here a perturbation consists of randomly selecting two cities in the tour and switching their positions.

5.2 Annealing Schedules Examined: GC, CR, EC and DPS

Four different annealing schedules (described further below) were applied to the TSP in a variety of scenarios designed to compare their effectiveness:

- GC: Geometric Cooling
- CR: Consecutive Rejection
- EC: Equilibrium Condition,
- DPS: Detection of Productive Search.

The GC annealing schedule is a similar to the one used in the original implementation of the simulated annealing algorithm (Kirkpatrick et al., 1983). The schedule named EC is based on the adaptive annealing schedule proposed by Huang et al.[1] (1986). Annealing schedule parameters with the exception

[1]This annealing schedule is a statistically-based schedule which originated in the electrical engineering field for applications to VLSI circuit layout. The approach assumes that objective function values are Normally distributed (which generally is not the case) and sets up target values for the number of points that should fall inside and outside of a band that is centered about a mean objective function value and who's width is a specified number of

of the number of iterations at each temperature, were standardized as much as possible and were set for each schedule as follows:

Initial Temperature. The initial temperature for all schedules is calculated using the following equation (White, 1984):

$$T_i = -\frac{3\sigma}{\ln P}, \tag{2}$$

where P is the desired probability of accepting a state that is more than three standard deviations worse than the current one, and σ is the population standard deviation of objective function values. A value of 0.9 was used for P. To approximate σ for the population, a sample of randomly-generated tours was created and the standard deviation was calculated to be about 20,000. The average length of the randomly-generated tours was about 170,000.

Temperature Reduction. In the GC schedules, the temperature is reduced geometrically; a new temperature is calculated by multiplying the current temperature by a constant:

$$T_{new} = T_{current} * M, \tag{3}$$

where M is the temperature reduction multiplier. Values of .8, .9 and .986 were used.

For the other schedules (CR, EC, and DPS), the new temperature is calculated as a function of the current temperature using the adaptive method suggested by Huang et al. (1986):

$$T_{new} = T_{current} * e^{-\left(\frac{0.7 \ T_{current}}{\sigma}\right)}. \tag{4}$$

Number of Iterations at a Temperature. Each of annealing the schedules run at a given temperature until their temperature reduction criteria are met:

- GC: Reaching a fixed number of allowable iterations at a temperature.
- CR: Achieving a specified number (either 50 or 100) of consecutive rejections of new states at a temperature, (referred to as the CR-50 and the CR-100 schedules).
- EC: Satisfying the "equilibrium condition" proposed by Huang et al. (1986).
- DPS: Lack of detection of productive search as described in Section 4.2.

In contrast to the other schedules, the number of iterations at a temperature is fixed *a priori* for the GC annealing schedule. In order to "tune" the algorithm, an initial set of runs was done with a 500 iteration limit to determine a good value for the multiplier, M. Of the three values, the value of .986 produced the best results so additional sets of runs were done using that value.

Although the CR, EC and DPS annealing schedules have temperature reduction criteria, in any simulated annealing algorithm, it is good practice to include an iteration limit for each temperature that automatically causes a temperature reduction when exceeded. This limit is often reached in the early portion of the simulated annealing process because the high initial temperature results in high a probability of accepting uphill steps, which reduces the probability of convergence. Thus, with any of these three annealing schedules a temperature reduction can also be triggered by exceeding the iteration limit.

For all four annealing schedules, a set of runs was performed with three different maximum iteration limits at every temperature: 1000, 2500 and 5000.

Algorithm Termination Condition. What differentiates simulated annealing from downhill search techniques is the probability of accepting inferior steps. Once the temperature becomes sufficiently low, the probability of accepting inferior steps becomes so small that the algorithm behaves just as a downhill search would, performing a purely local search. At this point the simulated annealing algorithm is no longer effective for global optimization and should be halted. The termination condition used in all annealing algorithm scenarios ends the algorithm optimization when three consecutive temperatures pass during which the algorithm accepts no inferior solutions.

5.3 Optimization Results

Simulated annealing is a stochastic search technique likely to produce different solutions to the same problem each time it is run. A set of ten runs seeking to find the minimum tour length for the traveling salesman problem was generated for the GC, CR, EC and DPS annealing algorithms described in Section 5.2. Performance parameters of interest in determining algorithm efficiency are: converged tour length, overall number of iterations and average iterations per temperature. Mean performance parameters are calculated for each annealing run set and displayed in Table 1. These results are elaborated upon with accompanying figures below and discussed in greater detail in Section 5.4.

The best solution to the posed TSP found in our experimentation was a tour valued at length 23,276. It was generated during a GC-.986 (2500 iteration limit) annealing schedule run that converged after 1,752,500 iterations. The DPS schedule best solution was a tour of length 24,423 generated during a run of 344,277 iterations. Given that the average randomly-generated tour has a length of about 170,000, these values are on par with one another. Neither tour is a good as the published TSP solution of 21,282 cited earlier, but recall that the published solution required iterative iteration between a human and a computer. These experiments indicate that the annealing schedules are operating properly and that the results can be used to make valid comparisons between the schedules.

Figure 2 displays means and 95% confidence intervals for the mean tour length for each of the annealing schedules. For example, there is a 95% probability that another set of 10 runs of the DPS annealing schedule using a maximum iteration limit of 2500 would have a mean converged value in the range from about 26,000 to 29,000. As can be seen in Figure 2, the three schedules EC, DPS and GC-.986 produced comparable results. Overlapping confidence intervals indicate that there is statistically significant difference in results.

The shortest tour lengths were obtained from the GC-.986 with 2500 and 5000 iteration limits (see Table 1 and Figure 2). The figure exhibits a trend that as the iteration limit increases, the quality of final results improves, as would be expected. For

standard deviations. As the simulated annealing algorithm iterates, counters keep track of whether points are inside or outside of the band. Whenever the "outside" target is met, both counters are reset and iteration continues. The temperature is reduced when the "inside" counter reaches its target. Further details are omitted for reasons of brevity but can be found in the referenced paper.

Table 1. Mean Performance Data for 10 Runs of Annealing Schedules on the Traveling Salesman Problem.

Annealing Run Scenario	Iteration Limit per Temperature	10-Run Mean Values of Algorithm Performance Parameters		
		Converged Tour Length	Number of Iterations	Iterations per Temperature
GC-.8	500	42,205	25,200	500
GC-.9	500	38,536	49,450	500
GC-.986	500	30,917	327,650	500
GC-.986	1000	28,965	676,500	1000
GC-.986	2500	25,693	1,803,500	2500
GC-.986	5000	25,471	3,794,500	5000
CR-50	1000	42,695	56,847	543
CR-50	2500	43,218	116,340	1109
CR-50	5000	42,934	202,758	2203
CR-100	1000	34,261	100,787	696
CR-100	2500	33,540	197,280	1364
CR-100	5000	32,777	340,585	2505
EC	1000	28,521	427,462	896
EC	2500	27,744	2,100,283	2252
EC	5000	27,079	6,354,698	4480
DPS	1000	28,280	289,921	750
DPS	2500	27,611	414,381	1113
DPS	5000	27,482	483,868	1422

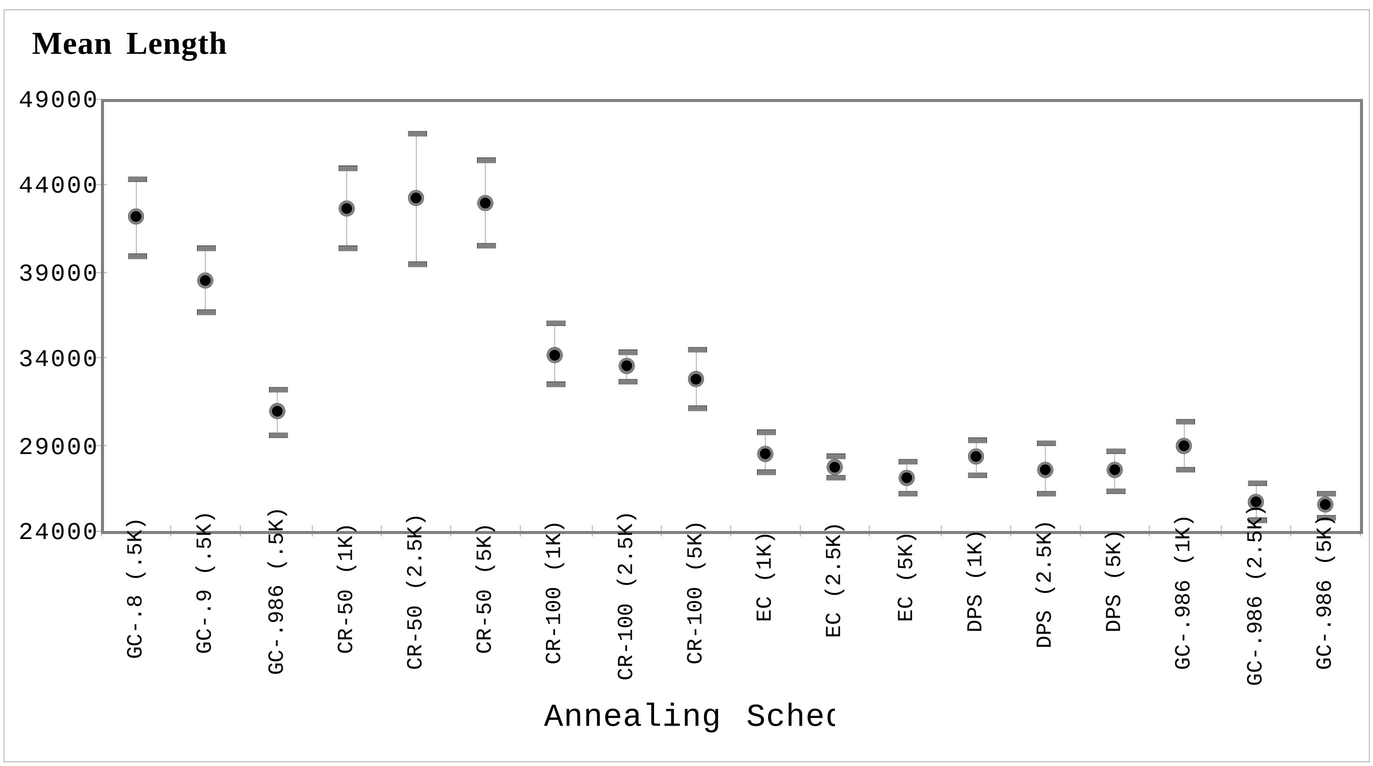

Figure 2. 95% Confidence Interval for Mean TSP Tour Lengths.

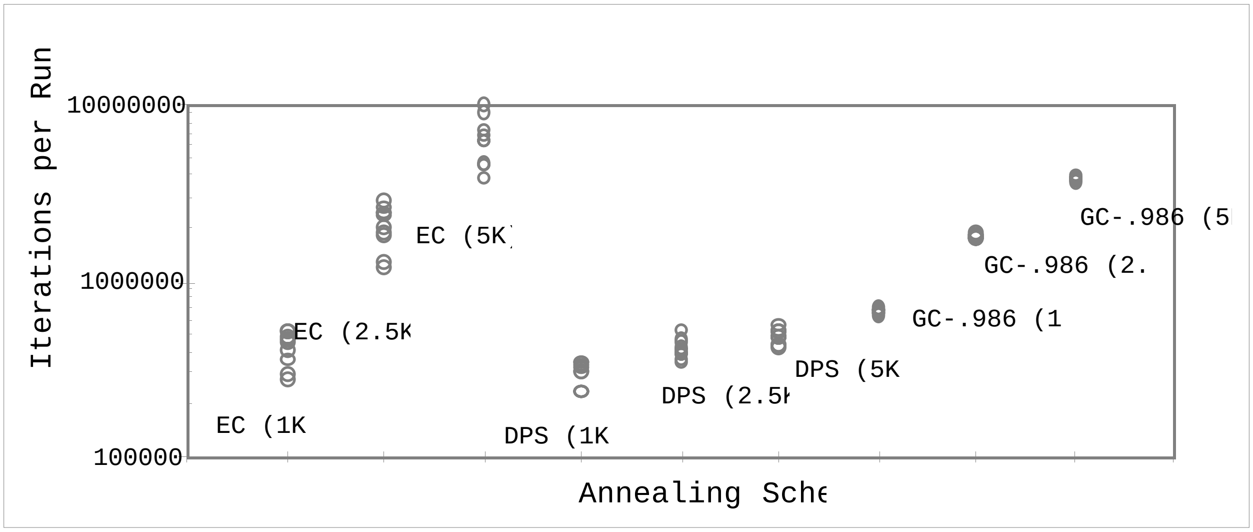

Figure 3. Iterations Until Convergence for the Three Best Annealing Schedules (DPS, GC-.986, and EC). Note: Y Axis is on a Log Scale.

the remainder of this section, discussions will focus on the results of the EC, DPS and GC-.986 schedules, and not the poorer performing schedules.

Figure 3 plots the number of algorithm iterations to convergence for each of the ten runs using the EC, DPS and GC-.986 schedules (with the Y axis on a log scale). On average, the DPS required fewer iterations than the EC or GC-.986 schedules to converge regardless of the iteration limit. In some cases, the effect on run length was tremendous. The quality of solutions for the EC and DPS schedules was about the same, but the DPS schedule performed better with regards to number of iterations. For the 5000 iteration limit, the tuned GC-.986 schedule had better final tour lengths than the DPS schedule, but at the cost of many iterations: that schedule required almost 8 times as many iterations (the EC schedule required over 13 times as many).

Figure 4 contains a plot of the average number of iterations per temperature for each EC, DPS and GC-.986 schedule run. The GC schedule will always use 100% of the maximum allowable iterations per temperature. The EC schedule can reduce the temperature before reaching the iteration limit. Table 1 shows that the EC schedule uses about 90% of the allowable iterations, irrespective of the iteration limit. The performance of the DPS schedule is noticeably different, using 75% of the allowable iterations for the 1000 iteration limit algorithm, all the way down to less than 30% for the 5000 iteration limit algorithm. The following section contains a more qualitative discussion of some of these results.

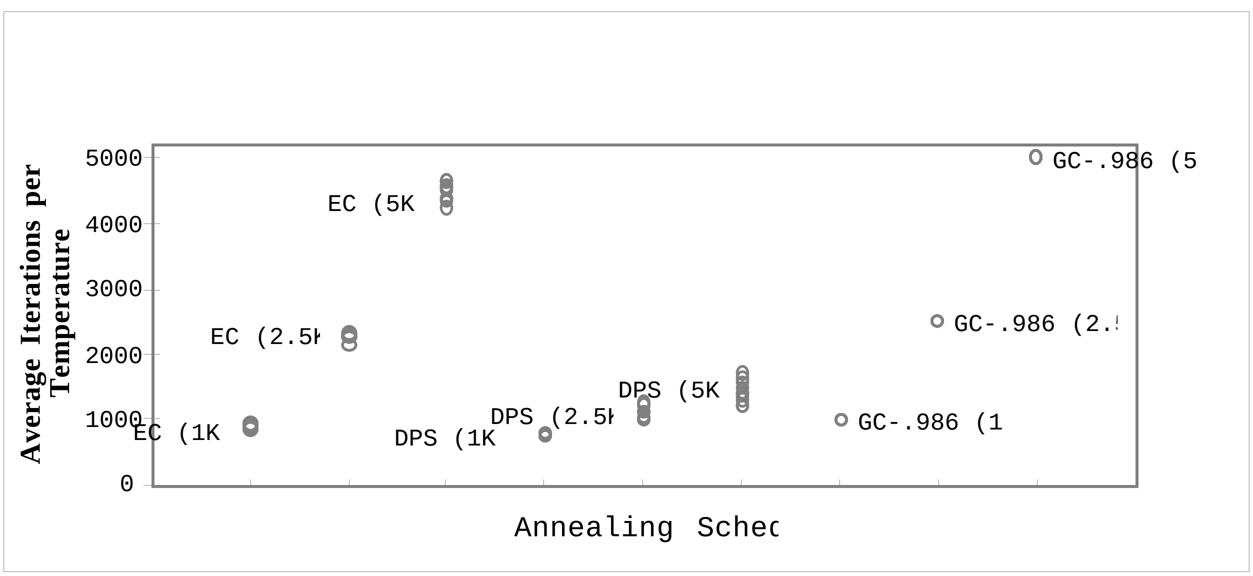

Figure 4. Average Number of Iterations Per Temperature for the Three Best Annealing Schedules (DPS, GC-.986, and EC).

5.4 Discussion on the DPS Approach to Detection of Productive Search

Iterations required for simulated annealing optimization run convergence is of primary importance because each iteration corresponds to a possibly complex objective function evaluation. Each of the schedules exhibited a trend of improved quality with increased iterations. However, the steep slopes of iteration increase in overall iterations for the EC and GC-.986 schedules are contrasted by the gentle slope displayed for the DPS schedule in Figure 3.

The ability of the DPS approach to control search can also be observed by comparing changes in the average number of iterations per temperature for each schedule (Table 1 and Figure 4). There is little to be said concerning the GC-.986 schedule; because it is a fixed schedule that runs for a constant number of iterations at each temperature. The EC schedule uses 90% of the run's maximum iteration limit. The EC schedule's dependence on the iteration limit for stopping search at a temperature, *and not on its own ability to detect the end of productive search.* In contrast, the percentage of maximum iterations used *decreases* for the DPS schedule as the maximum limit increases, indicating that the approach is very efficient in its use of additional allowable iterations. The DPS schedule can make use of additional available iterations when search is productive, but increases in the maximum iteration limit are not reflected by equivalent increases in the average iterations per temperature.

Recall that productive search is defined as search where the objective function is changing rapidly, meaning that the optimization algorithm is being effective. To illustrate the detection of productive search graphically, Figure 5 shows a plot for the DPS schedule that is similar to the plot in Figure 1; the upper curve is the objective function value at each temperature, T_i, and the lower curve is the number of iterations done at each temperature.

Contrasting Figures 1 and 5 highlights the significant effect of detection of productive search. In Figure 1, there was no relationship between the number of iterations done at a temperature and the productivity of search. In Figure 5, the relationship is quite visible; the algorithm reaches the maximum allowable iteration limit very often in the region where search is most productive. As search becomes less productive, the number of iterations drops off and the DPS schedule always decreases the temperature before the algorithm reaches the maximum allowable number of iterations. It should be noted that the peaks in number of iterations (lower curve) seen in the second half of the optimization correspond to small spurts of productive search that lead to a greater number of iterations at a temperature.

From this perspective, it becomes apparent that the fixed GC schedule, having a fixed number of iterations at each temperature, is an inherently inefficient one. If one were to imagine a plot similar to the one shown in Figure 1 for this schedule, the iteration curve would be a straight line, meaning that many iterations are done during the less-productive phases of the optimization. Although the GC-.986 schedule produced the best results for the traveling salesman problem, this was a result of sheer brute force. The DPS schedule demonstrated a many-fold reduction in computation at the expense of only a few percent in terms of quality.

In realistic engineering problems, it is generally not possible to create an objective function that encompasses every design objective. Thus, "within a few percent of an optimum" is often acceptable in optimization algorithms. Furthermore, depending on how time-consuming the objective function evaluations are, the reduced number of iterations may make the difference between being able to apply simulated annealing to a problem and not being able to.

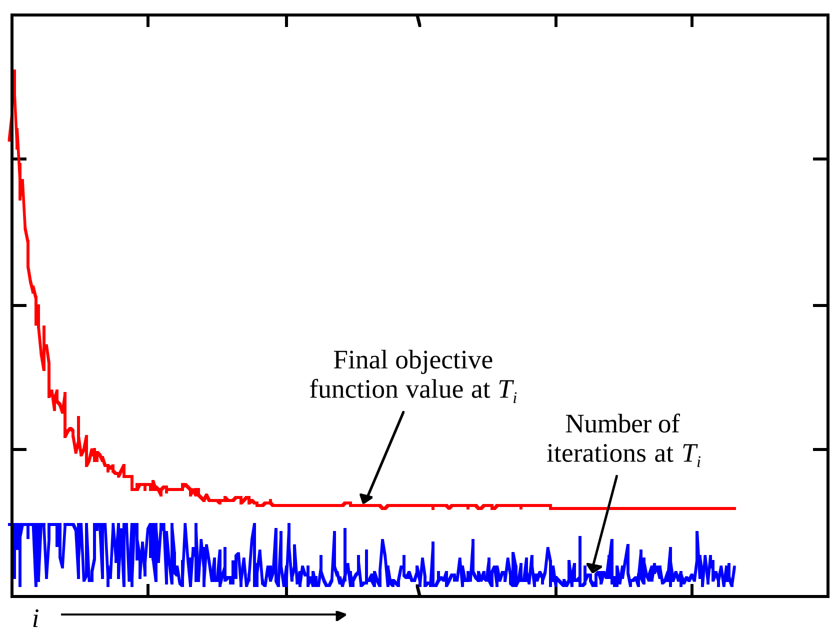

Figure 5. Illustration of Detection of Productive Search.

6 CONCLUSIONS

As simulated annealing becomes more popular in engineering design, the demand increases for improved algorithms. Increasing the efficiency of the search is a key issue for making the algorithm applicable to a broader range of computationally intensive problems. This paper proposes an approach toward increasing the algorithm efficiency by monitoring productive search and describes a new annealing schedule which reduced wasted computation and produced good results on a 100-city traveling salesman problem. The DPS approach also frees the user from the burden of tuning the schedule through experimentation, which is required for many other annealing schedules. Because the DPS schedule presented in this paper is relatively simple, it is quite likely that more sophisticated means of controlling annealing schedule parameters will lead to even better results.

Work in progress focuses on the application of the DPS schedule to an engineering design problem. Future work includes adapting SPC methods for use with autocorrelated data. Two possible methods for investigation are CuSum charting, and applying estimates of within-group variance in calculating charting limits (Wetherill and Brown, 1991). CuSum control charts seem particularly relevant because they are designed for monitoring systems having dynamic process means. Finally, the approach to detection of productive search described in this paper addresses only one of the components of the annealing schedule for the simulated annealing algorithm, that is, the number of iterations at a given temperature. The development of methods for controlling other parameters in the annealing schedule, such as the amount of temperature reduction, is another area of research that holds promise for additional improvements to the algorithm.

REFERENCES

DeVor, R. E., T.-H. Chang and J. W. Sutherland (1992), *Statistical Quality Design and Control*, Macmillan Publishing Company, New York.

Devore, J. L. (1987), *Probability and Statistics for Engineering and the Sciences*, Brooks/Cole Publishing Company, Monterey, CA.

Dong, J. and S. Vijayan (1996), "Optimal Feature Extraction Techniques – A Comparative Study," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper no. 96-DETC/CIE-1337, Irvine, CA, August.

Huang, M. D., F. Romeo and A. Sangiovanni-Vincentelli (1986), "An Efficient General Cooling Schedule for Simulated Annealing," *ICCAD-86: IEEE International Conference on Computer-Aided Design–Digest of Technical Papers,* Santa Clara, CA, November 11-13, pp 381-384.

Jain, P., P. Fenyes and R. Richter (1992), "Optimal Blank Nesting Using Simulated Annealing," *ASME Journal of Mechanical Design*, **114**:160-165.

Kim, G. J. and S. Szykman (1996), "Combining Interactive Exploration and Optimization for Assembly Design," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper no. 96-DETC/DAC-1482, Irvine, CA, August.

Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi (1983), "Optimization by Simulated Annealing," *Science*, **220**(4598):671-679.

Krolak, P., W. Felts and G. Marble (1971), "A Man-Machine Approach Toward Solving the Traveling Salesman Problem," *Communications of the ACM*, **14**(5):327-334.

Lam, J. and J.-M. Delosme (1988), "Performance of a New Annealing Schedule," *Proceedings of the 15th ACM/IEEE Design Automation Conference,* Anaheim, CA, June, pp 306-311.

Schmidt and Cagan (1996), "Configuration Design: An Integrated Approach Using Grammars," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper no. 96-DETC/DTM-1511, Irvine, CA, August.

Shea, K. and J. Cagan (1995), "A Shape Annealing Approach to Optimal Truss Design with Dynamic Grouping of Members," *Proceedings of the 1995 ASME Design Engineering Technical Conferences*, Vol. 1, DE-Vol. 82, Boston, MA, September, pp 377-383.

Szykman S. and J. Cagan (1996), "Synthesis of Optimal Nonorthogonal Routes," *ASME Journal of Mechanical Design*, **118**(3):419-424.

Szykman S. and J. Cagan (1997), "Constrained Three Dimensional Component Layout Using Simulated Annealing," *ASME Journal of Mechanical Design*, **119**(1):28-35.

Ullah, I. and S. Kota (1996), "Globally-Optimal Synthesis of Mechanisms for Path Generation Using Simulated Annealing and Powell's Method," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper no. 96-DETC/MECH-1225, Irvine, CA, August.

Xue, D. (1996), "Identification of the Optimal Product Configuration and Parameters for Improving Manufacturability," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper no. 96-DETC/DAC-1526, Irvine, CA, August.

Wetherill, G. B. and D. W. Brown (1991), *Statistical Process Control: Theory and Practice*, Chapman and Hall, New York.

White, S. R. (1984). "Concepts of Scale in Simulated Annealing," *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers*, Port Chester, NY, October, pp 646-651.

Zhang, C. and H.-P. Wang (1993), "The Discrete Tolerance Optimization Problem," *Manufacturing Review*, **6**:60-71.