Pergamon

Computers Math. Applic. Vol. 27, No. 5, pp. 71–82, 1994

0898-1221/94 $6.00 + 0.00

0898-1221(94)E0007-7

Bin-Packing by Simulated Annealing

R. L. RAO AND S. S. IYENGAR*
Robotics Research Laboratory, Department of Computer Science
Louisiana State University, Baton Rouge, LA 70803, U.S.A.

(Received and accepted May 1993)

Abstract—The process of gradually settling a combinatorial system into configurations of globally minimum energy has variously been called simulated annealing, statistical cooling, and so on. In the past, very large combinatorial optimization problems have been solved using this technique. It has also been shown that this method is effective in obtaining close-to-optimal solutions for problems known to be NP complete. Further, this technique is applicable to a whole class of problems that satisfy a few requirements.

The purpose of this paper is to illustrate an efficient version of the simulated annealing method as applied to a variant of the bin-packing problem. The computational complexity of the method is linear in input size similar to various well-known heuristic methods for the problem. The solutions obtained, however, are much better than any of the heuristic methods. The particular variant of the bin-packing problem we consider has several practical applications such as static task allocation in process scheduling and batch processing. At the time of this writing, we have not yet seen a stochastic solution to the bin-packing problem in the literature.

One of the distinguishing features of our research is the high quality of solutions obtained by our method. Extensive simulation experiments we have carried out show that the solutions obtained by the stochastic method show a significant improvement over those obtained by any of the well-known heuristic methods.

Keywords—Bin-Packing, Combinatorial optimization, Global minimum, Monte Carlo methods, NP completeness, Simulated annealing, Statistical cooling.

1. INTRODUCTION

1.1. The Bin-Packing Problem

The classical definition of the bin-packing problem involves packing a list of items of (possibly) different sizes into the smallest number of bins, each of which has a given maximum capacity. Coffman *et al.* [1] is a good survey of several approximation algorithms for bin-packing that yield quick sub-optimal solutions.

The variant of the classical problem we solve, deals with a fixed number of bins each with an unlimited capacity and the objective is to pack the items into these bins so that each bin has about the same total allocation. In other words, our attempt is to find the most *equable* distribution of items to bins. Our problem differs from the classical problem in the following two ways.

1. The bin sizes are not constrained. The rationale behind this particular variation of the problem is the fact that this model is appropriate in certain real-world situations. For example, in a batch processing environment, it is sometimes necessary to complete a fixed

This work was supported in part by ONR Grant No. N000014-91-J1306.
*E-mail addresses: ramana@bit.csc.lsu.edu and iyengar@bit.csc.lsu.edu for R. L. Rao and S. S. Iyengar, respectively.

number of tasks of (possibly) different sizes, given a fixed number of processors. The primary aim here is to find an allocation that minimizes the total idle time.

2. The total number of bins is fixed. This variation is a direct consequence of variation 1 above. It is reasonable to expect, in a practical situation, that the resources available are bounded in some way. The classical problem places no restriction on the number of bins.

In the remainder of this paper, we will refer to our variant of the problem as *the bin-packing problem* for brevity; where we allude to the classical problem, we will qualify it so that it is clear which problem we are refering to.

1.2. Simulated Annealing

Simulated annealing, a general purpose combinatorial optimization technique, was first proposed by Kirkpatrick *et al.* [2] in 1983. The analogy between the states of a physical system as it approaches thermal equilibrium and intermediate stages of solution of a combinatorial problem, is exploited to yield globally-minimum-cost solutions to the problem. This technique is a generalization of the Monte Carlo method developed earlier by Metropolis *et al.* [3] and has been successfully used to solve such optimization problems as the *Traveling Salesman Problem* [4] and the *Wire Length Minimization Problem* in VLSI circuits [5].

In this technique, the solution space of the combinatorial system being optimized is explored in a controlled fashion using a control parameter (the analogue of the temperature in a physical system) so that configurations with successively better measures for the objective function are obtained. Without loss of generality, we will assume that the objective function being optimized is being minimized and when we say that some state results in a *better* objective function than some other state, what we mean is, that state has a *smaller* value for the objective function.

Simulated annealing is similar to iterative refinement in the sense that both techniques involve changing the state of the system being optimized and examining the objective function of the system. While iterative improvement does not permit state changes that increase the objective function, thus forcing systems into local minima, simulated annealing permits *uphill* moves that help the system climb out of local minima and seek out other minima thereby enhancing the chance of a *good* minimum being found. The local minima sought out by iterative refinement are analogous to the metastable states obtained in a physical system after rapid cooling from a high temperature [6], and the overall effect is that of *quenching* the system which causes defects to be frozen into the structure of the system.

1.3. Motivation and Prior Work

The primary motivation for applying simulated annealing to the bin-packing problem is the observation that annealing yields remarkably good solutions to several combinatorial optimization problems known to be NP complete.[1]

Given that the solution space for the bin-packing problem is exponential in the size of the item list, it is natural to examine the feasibility of using the technique of simulated annealing to solve this problem. In fact, as the size of the item list gets larger, we can intuitively expect annealing to perform increasingly better than, say, the greedy heuristic. We have not come across an application of this technique to the bin-packing problem in the literature at the time of this writing. A secondary motivation for applying a stochastic approach to this problem is a remark in Laarhoven *et al.* [8], '... the theoretical basis of the [annealing] algorithm had reached a certain level of saturation and that major contributions were to be expected predominantly with respect to new applications.' Indeed, since then annealing has been used to solve a wide variety of optimization problems.

[1] Garey *et al.* [7] is an excellent repository of information about NP completeness.

A typical instance of the annealing experiment involves two important choices for the control parameter corresponding to the high temperature and the low temperature regimes of a physical system and a method of varying the control parameter so that the system is driven to optimality. A set of choices for these critical parameters is known as an *annealing schedule*. There is evidence in the literature that the global optimum of a combinatorial optimization problem can be found with probability one provided that the annealing schedule satisfies certain conditions [8–10].

In practical applications such as circuit placement in VLSI, the computing resources needed to obtain 'good' solutions are excessive. To remedy this, several approaches have been proposed in the literature. Greene *et al.* [6] have suggested a schedule which involves annealing without rejected moves. Their method offers a significant speed-up at the expense of increased memory usage. Lam *et al.* [5], White [11], and Huang *et al.* [12] have proposed techniques to make annealing schedules efficient.

The various approaches that attempt to remedy the massive computational time required by the annealing method, may be grouped into three broad categories—parallel annealing techniques [13], efficient annealing schedules [5,6,11,12] and controlled move generation methods. This last category of methods is usually problem-instance dependent and is not widely applicable to combinatorial optimization. The more popular techniques in the literature employ efficient annealing schedules and this paper also describes one such schedule.

2. PROBLEM FORMULATION

We define an instance of the bin-packing problem as consisting of

1. M bins, each of which has an unlimited capacity;
2. N items (sizes) $t_1, t_2, \ldots, t_N$; $0 \leq t_i \leq t_{\max}$, $1 \leq i \leq N$;
3. an objective function (also refered to as the cost or the energy function) defined as

$$C(\{a_i\}) = \sum_{j=1}^{M} (B_j - \overline{T})^2, \tag{1}$$

where B_j is the sum of the sizes of the items allocated to the j^{th} bin and $\{a_i\}$ is an allocation sequence $a_1, a_2, \ldots, a_N$; $1 \leq a_i \leq M$. An allocation sequence determines which bin each item is allocated to. Thus, every allocation sequence represents a feasible solution to the problem. The resulting distribution of items to bins is also called a configuration (state) of the problem (system). $\overline{T}$ is the total allocation at each bin that globally minimizes the cost function. Thus,

$$\overline{T} = \frac{1}{M} \sum_{i=1}^{N} t_i; \tag{2}$$

in general, though a particular instance of the problem may not render itself to such a perfect allocation scheme. Hence, there is the minimization problem.

The following lemma follows directly from the problem formulation.

LEMMA 1. *The magnitude of the objective function does not exceed* $M(M-1)\overline{T}^2$.

The maximum value for the objective function is obtained by allocating all the items in the item list to one bin and leaving the other $M-1$ bins empty. Such an allocation yields an objective function of $(M\overline{T} - \overline{T})^2 + (M-1)\overline{T}^2$ which is the same as $M(M-1)\overline{T}^2$. It is easy to see that no other allocation can yield a larger value for the objective function since moving any item from the bin to which all items are allocated now, would decrease the objective function at both the bin to which it is allocated now and the bin to which it is being moved.

```
algorithm anneal;
begin
    generate a random configuration C;
    T ← T∞;
    while T > T0 do
        repeat
            generate new configuration C'
            ΔC = C(C') − C(C);
            if ΔC < 0 or η < e^(−ΔC/T)
              C ← C';
        until thermal equilibrium is reached;
        T ← F(T);
    end do
    output C; {C is the best solution}
end.
```

Figure 1. Stochastic algorithm for bin-packing.

The goal of the problem is the identification of an allocation sequence $\{a_i\}$ such that

$$C(\{a_i\}) \leq C(\{a_i\}'), \qquad \forall \{a_i\}'. \tag{3}$$

Thus, $\{a_i\}$ denotes an allocation sequence that yields a globally minimum value for the objective function.

3. ANNEALING ALGORITHM

The algorithm for the bin-packing problem is shown in Figure 1. The algorithm starts with a high value for the temperature parameter. The temperature is then decreased gradually until a small enough value for the temperature is reached when the algorithm is terminated.

At each temperature, the system is perturbed several times. The set of iterations carried out at each value of the temperature is called a *chain*. The number of iterations in a chain is sometimes referred to as the chain length. The chain length is typically a small multiple of the problem size. We first note that the configurations of the bin-packing problem satisfy the property of *Strong Irreducibility* [14]. This follows directly from the observation that the configuration space of the bin-packing problem is finite[2] and the fact that any configuration may be attained from any other configuration by moving a finite number of items across bins.

The algorithm first allocates items to bins randomly. The list of items is not sorted according to the sizes of the items. This allocation serves as the starting point of the annealing process and will be referred to as the *initial state* of the system. To obtain a new state from the current state, the system is perturbed by selecting one of the two methods described in Section 4.3. The value of the objective function corresponding to this new, perturbed state is calculated.

The Metropolis [3] criterion is then applied and the algorithm accepts or rejects the new state. If $\Delta C < 0$, the new state is accepted without any qualification. If $\Delta C \geq 0$, then the new state is accepted with a probability

$$P(\Delta C) = e^{-\Delta C/T}; \tag{4}$$

during simulation, this is accomplished by generating a random number η uniformly distributed in $[0.0, 1)$ and accepting the new state whenever η is smaller than $P(\Delta C)$ above. T in (4) is the control parameter analogous to the temperature in a physical system. When the temperature reaches a low enough value, T_0, the algorithm terminates after indicating the most recently accepted configuration as the best solution found.

[2]In fact, there are exactly M^N feasible solutions to the problem.

4. ANNEALING SCHEDULE

The annealing schedule is described by quantitative choices for the three parameters—the starting value of the temperature T_∞, the stopping value of the temperature T_0, and the decrement function $\mathcal{F}(t)$ which determines the profile of the temperature from the beginning till the end of the annealing process.

The annealing curve obtained from a good schedule typically displays three broad areas of interest—a high energy area characterized by high temperature, an intermediate energy area, and a low energy area characterized by low temperature. The intermediate energy area is well defined and spans a relatively small portion of the temperature axis of the curve. We will refer to the behavior of the system described by the high energy region of the curve as the *High Temperature Regime* and that described by the low energy region of the curve as the *Low Temperature Regime.* These are further described in the following sections.

During the actual simulation, we first carry out an exploratory search of the configuration space where we assume that the temperature is infinite and accept each generated configuration. From this data we obtain fundamental statistical quantities about the problem. In particular, we are interested in the average value of the cost $\langle C(T) \rangle$ and the standard deviation σ cf the the density of states distribution. The density of states information may later be used to select an appropriate starting value for the temperature parameter.

4.1. High Temperature Regime

This region of the annealing curve (and the corresponding behavior of the system) is marked by the acceptance of most generated states. The value of the temperature parameter is so high that the Metropolis criterion is always satisfied. Thus, the average energy in this regime is very high. Just how high the starting temperature must be, for a good annealing schedule, is usually determined by monitoring the acceptance ratio at each temperature. The acceptance ratio (the fraction of generated states that are accepted) is arbitrarily fixed at some high value such as 0.9 and the temperature is increased to a value where the acceptance ratio is high enough.

While this serves as a problem-independent method of fixing the starting value of the temperature, often it yields a temperature value that is too high thus yielding an annealing schedule that is wasteful of computational resources. For the bin-packing problem, Lemma 1 gives the theoretical maximum for the objective function. If the control parameter is just high enough to accept the configuration with this maximum energy, then it follows that the temperature is high enough to accept any configuration. This is the technique we use to arrive at the high temperature limit for the schedule.

If t_k is the item with the largest size in the item list, then the configuration that allocates t_k alone to a bin and all the other items to another bin has the property that it is within one move[3] of the maximum energy configuration. The energy of this configuration is given by

$$C = (M\overline{T} - t_k - \overline{T})^2 + (M-2)\overline{T}^2 + (t_k - \overline{T})^2 = M(M-1)\overline{T}^2 - 2t_k(M\overline{T} - t_k), \qquad (5)$$

and the difference in energy, ΔC, between this configuration and the maximum energy configuration is given (from Lemma 1) by

$$\Delta C = 2t_k(M\overline{T} - t_k).$$

Assuming without loss of generality that there exists only one configuration with the maximum energy, an uphill move from a configuration with energy given by equation (5) will be accepted only if $e^{-\Delta C/T} \leq 1/N$ (only one out of N possible moves results in the maximum energy configuration). This gives the high temperature condition as

$$T_\infty \simeq \frac{2t_k(M\overline{T} - t_k)}{\ln(N)}. \qquad (6)$$

[3]Moves are described in Section 4.3.

Needless to say, a T_∞ choice larger than that suggested by equation (6) does not yield any better a solution.

The annealing algorithm yielded good annealing curves with this high temperature condition. Based on the condition proposed by White [11], Huang *et al.* [12] suggest using a high temperature limit of the form $T_\infty = \kappa\sigma$ where σ is the standard deviation of the cost distribution (obtainable from the density of states graph) and κ may be calculated assuming a Gaussian cost distribution and selecting a temperature that is high enough to accept a configuration that is within a few standard deviations from the current configuration with an arbitrarily fixed probability. We have found no significant improvement in the quality of solutions obtained by fixing the high temperature limit in this way.

4.2. Low Temperature Regime

This region of the annealing curve is characterized by acceptance of new states mainly if they lead to a better value for the objective function. The Metropolis criterion is dominated by the change in cost and not by the acceptance probability due to the low value of the temperature. The average energies in this regime are close to the global minimum at the extreme low temperature end of the curve. Again, it is easy to see that there must exist one or more configurations that bound the objective function on the lower side.

Several annealing schedules in the literature recommend that the annealing process be stopped when there is no appreciable change in the quality of the solution across a few chains of computation. While in general, this is a good guideline, it is possible that the problem instance has several degenerate low energy states. In such a situation, at low temperatures, a configuration might repeat a few times in succession without necessarily being the global minimum. Our method of determining when the annealing process should be stopped takes into account the lowest temperature scale of the system.

The smallest change in the objective function can be estimated easily. The smallest value of the objective function is zero (theoretically). Let t_i be an item in the item list with the least size.[4] Thus, the smallest ΔC for any perturbation will involve moving this item from the bin to which it is allocated in a perfect allocation to any other bin. This yields $\Delta C = 2\,{t_i}^2$. Consider an allocation where $M - 2$ bins each have a total allocation of $\overline{T}$ and the remaining two bins have a total allocation of $\overline{T} - t_i$ and $\overline{T} + t_i$. Of a total of N possible moves at this configuration, only one goes downhill to the perfect allocation. Thus, it must be the case that $e^{-\Delta C/T} \leq 1/N$ for the equilibrium condition to be satisfied. This gives us the low temperature limit as

$$T_0 \simeq \frac{2\,t_i^2}{\ln(N)}. \tag{7}$$

Of course, a smaller choice for T_0 will work just as well though this would waste computational time since no new configurations would be accepted once the perfect allocation is reached and the temperature is not higher than the limit given by equation (7).

4.3. Move Generation

A *move* in the annealing process denotes the generation of a candidate configuration for the system. This new configuration may or may not be accepted as the next state of the system depending upon the control parameter and the random number η in the Metropolis criterion. Typically, a move is generated by modifying the current state of the system in some way. During the actual simulation, we have noticed that two different types of moves are effective, namely,

[4] There may be more than one item with this size.

1. relocation of a single randomly selected item from the bin to which it is currently allocated to a randomly selected bin, the associated change in energy being given by

$$\begin{aligned}\Delta C &= (B_{a_i} - t_i - \overline{T})^2 + (B_j + t_i - \overline{T})^2 - (B_{a_i} - \overline{T})^2 - (B_j - \overline{T})^2 \\ &= 2\,t_i\,(B_j - B_{a_i} + t_i);\end{aligned} \tag{8}$$

2. randomly selecting two items currently allocated to two different bins and exchanging their positions, the associated change in energy in this case being

$$\begin{aligned}\Delta C &= (B_{a_i} - t_i + t_j - \overline{T})^2 + (B_j + t_i - t_j - \overline{T})^2 - (B_{a_i} - \overline{T})^2 - (B_j - \overline{T})^2 \\ &= 2\,(t_j - t_i)\,(B_{a_i} - B_{a_j} - t_i + t_j).\end{aligned} \tag{9}$$

At high temperatures, $e^{-\Delta C/T} \simeq 1$ and state changes involving relocation of items with large sizes are likely to be accepted. At low temperatures, $e^{-\Delta C/T} \simeq 0$ and generated configurations are likely to be accepted only when they have a smaller energy.

Typically, the choice of an acceptance criterion such as the Metropolis criterion means that the acceptance ratio[5] tends to become very small at low temperatures. To counteract this effect, our annealing schedule uses both kinds of move generation strategies above. At high temperature, we use moves of Type 1 above which coarsely optimize allocation, while at low temperatures we switch to moves of Type 2 which make finer adjustments to the objective function. We have also tested schedules which employ both kinds of moves all the time, increasing the proportion of Type 2 moves applied at lower temperatures. The simulation results indicate that this method of hybrid move generation yields only slightly better solutions when compared to schedules that employ only one kind of move generation.

4.4. Temperature Decrement

The rate at which the control parameter is varied has a profound impact on the quality of the final solution obtained by annealing. Too slow a rate wastes computational time while too fast a cooling rate quenches the system and yields local minima. The optimal cooling rate is hard to determine although there have been schedules in the literature [5] using dynamically determined temperature decrements. Typically, the temperature is decremented according to a logarithmic scheme [12]. The idea is that in the absence of a good guideline, our best bet is to ensure that the average cost decreases smoothly, thus, increasing the chance of obtaining a good annealing curve.

We will use a temperature decrement function of the form

$$\mathcal{F}(T) = \gamma T, \tag{10}$$

where γ lies in the interval $[0.9, 1.0)$. The closer it is to 0.9, the faster is the rate of cooling and the closer it is to 1.0, the slower is the cooling rate. In our simulation experiments, we have obtained very good solutions with a γ value of about 0.95.

5. SIMULATION RESULTS

We have applied the algorithm described in Section 3 to the bin-packing problem and have run extensive simulation experiments. In this section, we present a representative cross section of our results.

[5]This is the fraction of generated configurations that are accepted according to the Metropolis criterion.

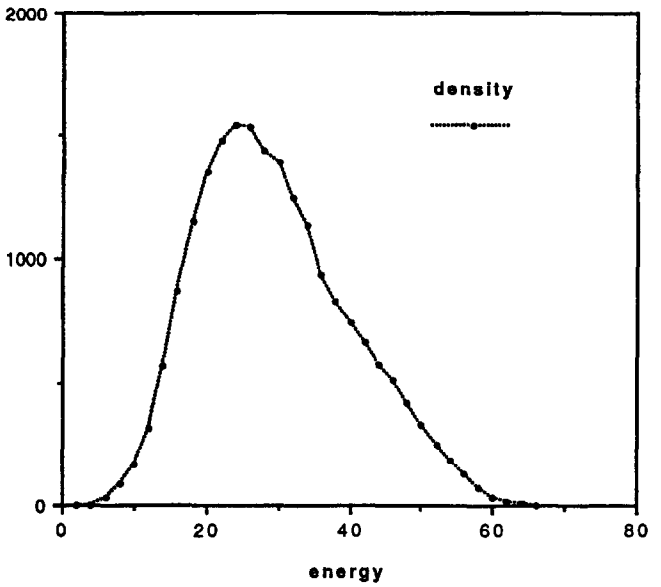


Figure 2. The density of states distribution.

5.1. Density of States

The first task is to obtain the distribution of the density of states. The density of states is a graph of the number of configurations plotted against their energy ranges. We obtain this data before the annealing process by accepting every new generated state (infinite temperature assumption) for a fixed number of iterations and counting the number of states with their energies in a particular interval. A graph of the density of states curve is shown in Figure 2. From the curve it is apparent, as expected, that the system is finite. Further, we notice that the curve is unimodal and shows Gaussian characteristics.

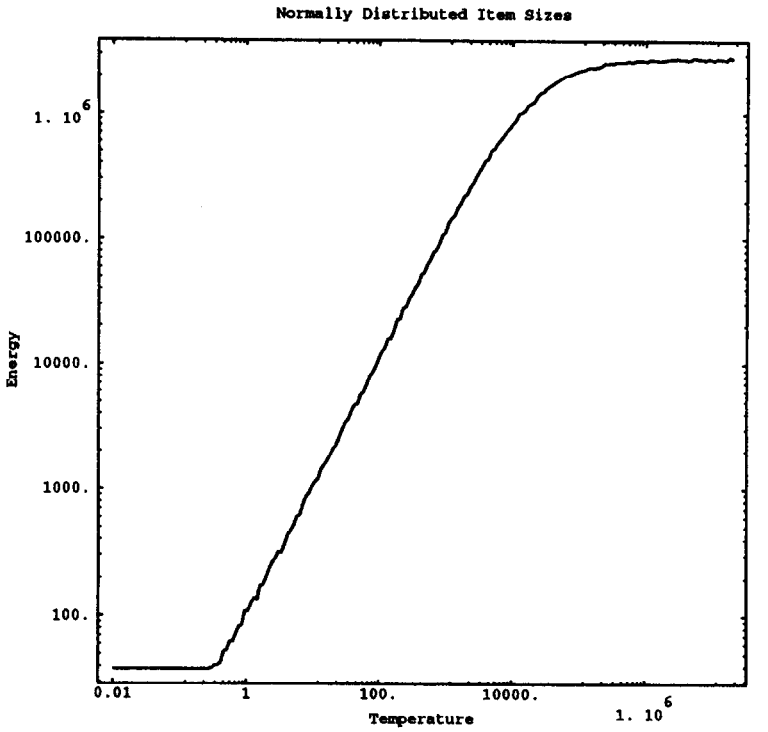
5.2. Annealing Curve

In analyzing annealing schedules, it is useful to examine the graph of the average energy at a fixed temperature at various temperatures during the annealing experiment. This graph of $\langle C(T) \rangle$ vs. T is called an annealing curve and contains important information about the experiment. The most important feature of this curve is that it is intimately related to the fundamental statistical quantities of the system and hence provides valuable information about the condition of the system at various temperatures. 'Good' annealing curves are marked by well-defined regions corresponding to the high, intermediate, and low temperature ranges. Annealing curves, representative of the experiments conducted, appear in Figure 3.

5.3. Comparison Studies

Several heuristic methods have been used to solve the bin-packing problem. We have chosen four candidate heuristic methods and compared their performance with that of the annealing algorithm. The four well-known methods we tested are

1. **LPF**: attempts to pack the Largest Piece First; from a priority queue built out of a sorted (nonincreasing) item list, the next item is allocated to the bin with the least total allocation so far;
2. **SPF**: identical to LPF except that the item list is sorted in nondecreasing order;
3. **FFI**: attempts to pack the next item on the sorted (nondecreasing) item list into the first bin in which it will fit;
4. **FFD**: identical to FFI except that the item list is sorted in nonincreasing order.



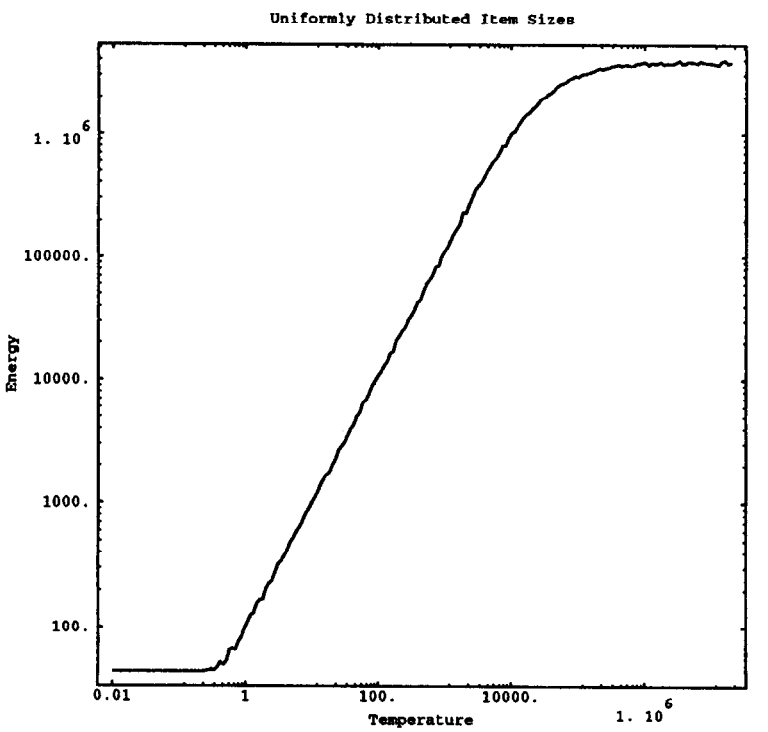


Figure 3. Annealing curves.

Bins	Previous Results				Our Method
	LPF	SPF	FFI	FFD	SA
10	2.4	8114	8810	2.4	2.4
20	3.2	16563	26055	3.2	3.2
40	9.6	32068	32187	21.6	9.6
60	19.7	49126	55696	49.7	13.7
80	62.8	63107	63739	80.8	14.8
100	361.0	—	—	95.0	29.0
120	77.9	—	—	181.9	25.9
140	964.1	—	—	66.1	36.1
160	326.4	—	—	282.4	34.4
180	673.9	—	—	193.9	47.9
200	4967.5	—	—	101.5	47.5
300	10654.3	—	—	354.3	86.3

Figure 4. Performance comparison (uniform distribution).

Bins	Previous Results				Our Method
	LPF	SPF	FFI	FFD	SA
10	100.9	6962	5750	20.9	0.9
20	211.0	21341	23963	25.0	6.9
40	1074.9	34758	42180	130.9	8.9
60	1053.2	48569	56731	197.3	19.3
80	480.3	48274	71894	162.3	22.3
100	1079.2	—	—	381.2	27.2
120	1484.1	—	—	688.1	26.1
140	4584.7	—	—	778.7	38.7
160	7186.1	—	—	1052.1	40.1
180	5458.9	—	—	719.0	35.0
200	4967.5	—	—	101.5	47.5
300	14742.3	—	—	982.3	42.3

Figure 5. Performance comparison (normal distribution).

The following figures summarize the results of the simulation experiment involving 1000 items and bins varying in number from 10 to 300. In all the experiments, the stochastic method yielded a solution that was at least as good as, and in most cases much better than, the solution obtained by applying the other methods.

Figure 4 shows the tables pertaining to the experiment involving item sizes generated randomly according to a uniform probability distribution. The blank cells in rows 6 through 12 indicate that the values for the corresponding techniques were too large to warrant inclusion in the table.

Figure 5 shows the results of the simulation experiment where the item sizes were generated according to the Gaussian (normal) probability distribution. It is worth noting that the stochastic technique works slightly better with normally distributed item sizes than with uniformly distributed sizes.

Figures 6 and 7 show a graphical comparison between the results obtained by the best heuristic method and that obtained by our method. It is interesting to note that while the heuristic method has an erratic behavior, our method performs consistently well as the number of bins in the simulation experiment increases.

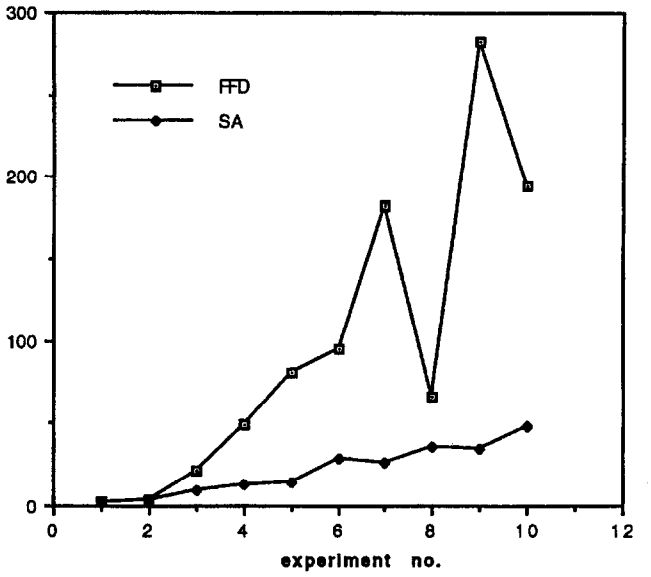


Figure 6. Performance comparison (uniform distribution).

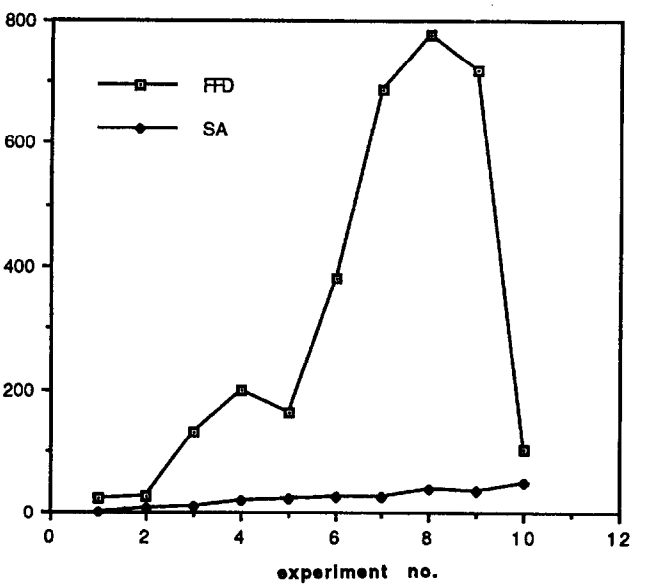


Figure 7. Performance comparison (normal distribution).

6. CONCLUDING REMARKS

We have presented an efficient annealing schedule for the solution of a variant of the classical bin-packing problem. The stochastic method yields solutions that are, in almost all cases, much better than heuristic methods for bin-packing. The solutions obtained by the stochastic method are seldom *worse* than one of the heuristic methods. Further, the solutions obtained with the stochastic method are *stable*, which means that the quality of the solution is consistently good unlike solutions obtained by heuristic methods whose performance tends to be problem-instance-dependent and consequently, erratic.

Possible future extensions to this work include dynamic thermal equilibrium detection and parallel annealing techniques. Another area for future work could be the exploration of more efficient move generation methods to counteract the low acceptance ratio of generated states in the low temperature regime.

REFERENCES

1. E.G. Coffman Jr., M.R. Garey and D.S. Johnson, Approximation algorithms for bin-packing—an updated survey, In *Algorithm Design for Computer System Design*, (Edited by G. Ausiello, M. Lucertini and P. Serafini), Springer-Verlag, 49–106 (1984).
2. S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671–680 (1983).
3. M. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087–1092 (1953).
4. V. Černy, A thermodynamic approach to the traveling salesman problem: An efficient simulation algorithm, *J. Optimization Theory and Applications* **45**, 41–51.
5. J. Lam and J-M. Delosme, Performance of a new annealing schedule, *Proc. 25th ACM/IEEE Design Automation Conference*, 306–311 (1988).
6. J.W. Greene and K.J. Supowit, Simulated annealing without rejected moves, *Proc. IEEE Int. Conf. on Computer Design*, 658–663, October (1984).
7. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., San Francisco (1979).
8. P.J.M. Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Co., Dordecht, Holland (1987).
9. D. Mitra, F. Romeo and A. Sangiovanni-Vincentelli, Convergence and finite-time behavior of simulated annealing, *Proc. 24th Conf. on Decision and Control*, Ft. Lauderdale, 761–767, December (1985).
10. F. Romeo and A. Sangiovanni-Vincentelli, Probabilistic hill climbing algorithms: Properties and applications, *1985 Chapel Hill Conf. on VLSI*, (Edited by H. Fuchs), Computer Sciences Press, Chapel Hill, NC (1985).
11. S. White, Concepts of scale in simulated annealing, *Proc. IEEE Int. Conf. on Computer Design*, 646–651, October (1984).
12. M.D. Huang, F. Romeo and A. Sangiovanni-Vincentelli, An efficient general schedule for simulated annealing, *Proc. IEEE Int. Conf. on Computer Aided Design*, 381–384 (1986).
13. P. Banerjee and M. Jones, A parallel simulated annealing algorithm for standard cell placement on a hypercube computer, *Proc. IEEE Int. Conf. on Computer-Aided Design*, 34–37 (1986).
14. B. Hajek, A tutorial survey of theory and applications of simulated annealing, *Proc. 24th Conf. on Decision and Control*, 755–760, Ft. Lauderdale, December (1985).
15. A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA (1974).
16. S. Nahar, S. Sahni and E. Shragowitz, Simulated annealing and combinatorial optimization, *Proc. of the 23rd Design Automation Conf.*, 293–299 (1986).
17. R.H.J.M. Otten and L.P.P.P. van Ginneken, Stop criteria in simulated annealing, *Proc. IEEE Int. Conf. on Computer Design*, 549–552, October (1988).