A STOCHASTIC APPROACH TO THE BIN-PACKING PROBLEM

Ramana L. Rao **S.S. Iyengar**

Robotics Research Laboratory, Louisiana State University

keywords: Bin-Packing, Simulated Annealing, Combinatorial Optimization, Global Minimum, Statistical Cooling

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. 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.

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.

Introduction

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. [2] 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, we are interested in 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 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 availability of physical resources is bounded in some way. The classical problem places no restrictions 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.

Simulated Annealing

Simulated Annealing, a general purpose combinatorial optimization technique, was first proposed by Kirkpatrick et al. [8] in 1983. This technique is a generalization of the Monte Carlo method developed earlier by Metropolis et al. [10] and has been successfully used to solve optimization problems such as the *Traveling Salesman Problem* [3] and the *Wire Length Minimization Problem* in VLSI circuits [9].

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.

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].

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 [1], efficient annealing schedules [5, 7, 9, 11] 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 describes one such schedule.

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) $t_1, \ldots, t_N; 0 \leq t_i \leq t_{max}, 1 \leq i \leq N$
3. An objective function (also refered to as the cost or the energy function) defined as

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

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

[1] Garey et al. [4] is an excellent repository of information about NP-Completeness

$a_1, a_2, \ldots, a_N$; $1 \leq a_i \leq M$. An allocation sequence determines which bin each item is allocated to. Thus, each 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 minimizes the cost function. Thus,

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

however, a particular instance of the problem may not render itself to such a perfect allocation scheme.

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 value 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 it is currently allocated, would decrease the objective function value both at the bin to which it is currently allocated and at the bin to which it is being moved.

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

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

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

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 algorithm first allocates items to bins randomly. To obtain a new state from the current state, the system is perturbed by selecting one of the two move generation methods described later. The value of the objective function corresponding to this new state is calculated.

The Metropolis [10] criterion is then applied and the algorithm accepts or rejects the new state. If $\Delta C < 0$, the new state is accepted. If $\Delta C \geq 0$ then the new state is accepted with a probability

$$P(\Delta C) = e^{-\frac{\Delta C}{T}}. \qquad (4)$$

T in equation 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 and the most recently accepted configuration is the best solution found.

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 start to end of the annealing process.

During 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 system. In particular, we are interested in the average value of the cost $< C(T) >$ and the standard deviation σ of the the density of states distribution.

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 must the starting temperature be, for a good annealing schedule, is usually determined by monitoring the acceptance ratio at each temperature. 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.

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 [2] of the maximum energy configuration. The energy of this configuration is given by

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

and the difference in energy, ΔC, between this configuration and the maximum energy configuration is given (from Lemma 1) by $\Delta C = 2\,t_k\,(M\,\overline{T} - t_k)$.

An uphill move from a configuration with energy given by equation 5 will be accepted only if $e^{-\frac{\Delta C}{T}} \leq 1/N$ (only one out of N possible moves results in the maximum energy configuration). This gives the high temperature condition as

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

The annealing algorithm yielded good annealing curves with this high temperature condition. Based on the condition proposed by White [11], Huang et al. [7] 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.

[2] *moves* are described in a later section

Low Temperature Regime

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 stopping crioterion 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 smallest size [3]. 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 $e^{\frac{-\Delta C}{T}} \leq 1/N$ [4]. 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 good 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.

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.

1. Relocation of a single randomly selected item from the bin to which it is currently allocated to a randomly selected bin
2. Randomly selecting two items currently allocated to two different bins and exchanging their positions

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

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. The optimal cooling rate is hard to determine, although there have been schedules in the literature [9] using dynamically determined temperature decrements. Typically, the temperature is decremented according to a logarithmic scheme [7]. The idea is that in the absence of a good guideline, our best bet is to ensure that the average cost decreases smoothly.

We will use a temperature decrement function of the form $\mathcal{F}(T) = \gamma\, T$ where γ lies in the interval [0.9, 1.0). In our simulation experiments good solutions were obtained with a γ value of about 0.95.

[3] there may be more than one item with this size
[4] for the equilibrium condition to be satisfied

Simulation Results

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

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.

Annealing Curves

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 $< C(T) >$ vs. T is called an *annealing curve* and contains important information about the experiment. '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.

Comparison Studies

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

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

The following figures summarize the results of the simulation experiment involving 1000 items and bins varying in number from 10 till 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.

Figures 4 and 5 show the tables pertaining to the experiment involving item sizes generated randomly according to a uniform and the normal probability distribution respectively. 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.

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 worth noting that, while the heuristic method has an erratic behavior, our method performs consistently well as the number of bins in the simulation experiment increases.

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.

```
algorithm anneal;
begin
    generate a random configuration C;
    $T \leftarrow T_\infty$;
    while $T > T_0$ do
        repeat
            generate new configuration C'
            $\Delta C = \mathcal{C}(C') - \mathcal{C}(C)$;
            if $\Delta C < 0$ or $\eta < e^{-\frac{\Delta C}{T}}$
                $C \leftarrow C'$;
        until thermal equilibrium is reached;
        $T \leftarrow \mathcal{F}(T)$;
    end do
    output C; {C is the best solution}
end.
```

Figure 1: Stochastic Algorithm for Bin-Packing

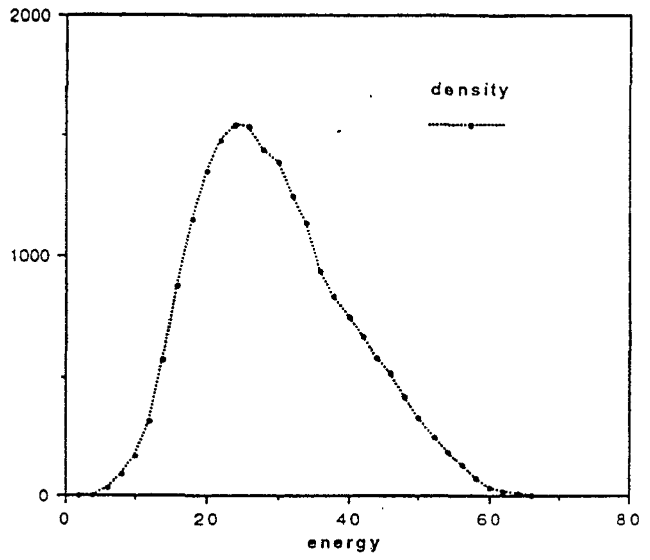

Figure 2: The density of states distribution

References

[1] Banerjee, P., and Jones, M., "A Parallel Simulated Annealing Algorithm for Standard Cell Placement on a Hypercube Computer," *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1986, pp. 34–37.

[2] Coffman Jr., E.G., Garey, M.R., and Johnson, D.S., "Approximation Algorithms for Bin-Packing — An Updated Survey," *Algorithm Design for Computer System Design*, G. Ausiello, M. Lucertini, and P. Serafini Ed., Springer-Verlag, 1984, pp. 49–106.

[3] Černy, V., "A Thermodynamic Approach to the Traveling Salesman Problem: an efficient simulation algorithm," *J. Optimization Theory and Applications*, 45, pp. 41–51.

[4] Garey, M.R., and Johnson, D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness," W.H. Freeman and Co., San Francisco, 1979.

[5] Greene, J.W., and Supowit, K.J., "Simulated Annealing without Rejected Moves," *Proc. IEEE Int. Conf. on Computer Design*, October 1984, pp. 658–663.

[6] Hajek, B., "A Tutorial Survey of Theory and Applications of Simulated Annealing," *Proc. 24th Conf. on Decision and Control*, Ft. Lauderdale, December, 1985, pp. 755–760.

[7] Huang, M.D., Romeo, F., and Sangiovanni-Vincentelli A., "An efficient General Schedule for Simulated Annealing," *Proc. IEEE Int. Conf. on Computer Aided Design*, 1986, pp. 381–384.

[8] Kirkpatrick S., Gelatt Jr., C.D., Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, 220, 1983, pp. 671–680.

[9] Lam, J., and Delosme, J-M., "Performance of a New Annealing Schedule," *Proc. 25th ACM/IEEE Design Automation Conference*, 1988, pp. 306–311.

[10] Metropolis, M., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, 21, 1953, pp. 1087–1092.

[11] White, S., "Concepts of Scale in Simulated Annealing," *Proc. IEEE Int. Conf. on Computer Design*, October 1984, pp. 646–651.

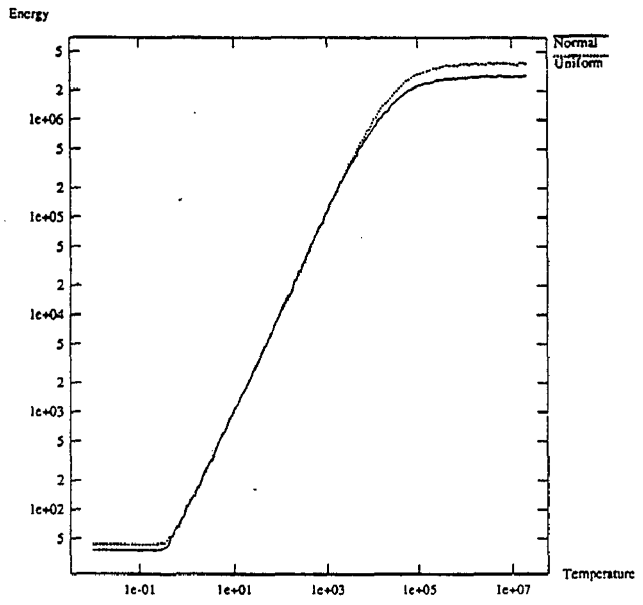

Figure 3: Annealing Curves

Bins	Previous Results				
	LPF	SPF	FFI	FFD	ANNEAL
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)

Bins	Previous Results				
	LPF	SPF	FFI	FFD	ANNEAL
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)

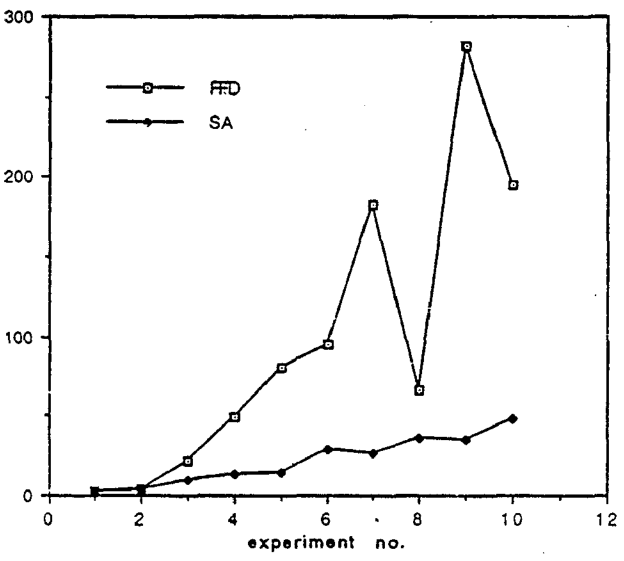

Figure 6: Performance Comparison (Uniform)

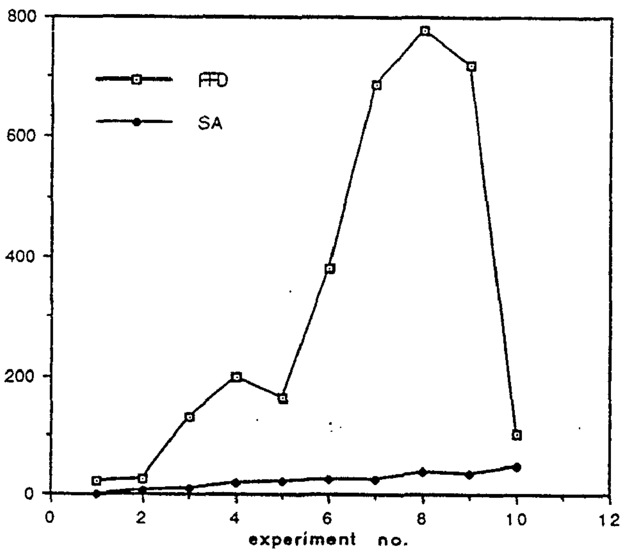

Figure 7: Performance Comparison (Normal)