SOLVING BIN PACKING PROBLEM USING SIMULATED ANNEALING

[1]EMRULLAH SONUC, [2]BAHA SEN, [3]SAFAK BAYIR

[1,2,3]Karabuk University , Yildirim Beyazit University, Karabuk University
E-mail: [1]esonuc@karabuk.edu.tr, [2]bsen@ybu.edu.tr, [3]safakbayir@karabuk.edu.tr

Abstract- Bin packing problem (BPP) is an NP-Hard combinatorial optimization problem. Several exact methods are presented in literature for finding optimal solutions of BPP. Also heuristic and metaheuristic methods are used for one dimensional BPP. In this paper, a proposed method using First Fit Decreasing (FFD) and Simulated Annealing (SA) algorithm is presented for solving one dimensional BPP. At first, FFD runs to place the objects for achieving max capacity of each bin. After this process, SA runs for searching neighbor solutions to obtain best result for objective function. In term of quality solutions, the proposed method is capable of delivering good quality results for BPP datasets.

Index Terms- Bin packing problem, combinatorial optimization, heuristic algorithms, simulated annealing.

I. INTRODUCTION

The Bin Packing Problem is an NP-Hard combinatorial optimization problem. Exact methods and heuristics are used to solve these types of problems. As exact methods, Branch and bound [1] or dynamic programming are used to find optimal solutions. However the run-time performance of traditional exact methods is a problem if the size of problem increases. In order to deal with these problems heuristic methods have been developed. These methods provide a optimal or near-optimal solutions in a reasonable time. Example of some heuristic and metaheuristic algorithms such as First Fit Heuristic, Best Fit Heuristic, Tabu Search [2], Weight Annealing [3] and Variable Neighbourhood Search [4]. In this paper, a method using First Fit Decreasing Heuristic and Simulated Annealing has been developed for solving bin packing problem. This paper is structured as follows. 2nd part gives definition and formulation of Bin Packing Problem. Proposed method is presented in 3rd part. Computational results are shown in 4th part and conclusion is presented in 5th part.

II. BIN PACKING PROBLEM

The Bin Packing Problem (BPP) can be described as follows. Given a set of n objects and n bins, weight of each object is shown as w_i, capacity of each bin is shown as C and C is a positive integer, BPP can be defined as follows:

$$\min \sum_{i=1}^{n} b_i \qquad (1)$$

$$s.t. \sum_{j=1}^{n} w_j a_{ij} \leq C_{b_i} \qquad a_i \in N, \quad i \in \{1, \ldots, n\}, \qquad (2)$$

$$\sum_{i=1}^{n} a_{ij} = 1, \qquad j \in N, \qquad (3)$$

$$b_i = 0 \; or \; 1, \qquad i \in N, \qquad (4)$$

$$a_{ij} = 0 \; or \; 1, \qquad i \in N, \qquad j \in N, \qquad (5)$$

where

$$b_i = \begin{cases} 1 & if \; bin \; i \; is \; used, \\ 0 & otherwise, \end{cases}$$

$$a_{ij} = \begin{cases} 1 & if \; object \; j \; is \; assigned \; to \; bin \; i \;, \\ 0 & otherwise. \end{cases}$$

It will be assumed that weight of each object is less than capacity of each bin:

$$w_j \leq C, \qquad j \in N. \qquad (6)$$

III. PROPOSED METHOD

First Fit Decreasing Heuristic (FFD) is an approximation algorithm and works on greedy strategy. Objects are sorted in decreasing order for FFD. Then place the next object in first bin where it will fit. If the object does not fit any available bin, create a new bin and place the object in it [5]. The pseudocode of FFD is as follows:

```
Begin
Sort objects in descending order by weight.
for All objects i = 1,  . . . , n do
for All bins j = 1,  . . . do
if Object i fits in bin j then
Place object i in bin j.
Break and continue with the next object.
end if
end for
if Object i does not fit in any open bin then
Create new bin and place object i.
end if
```

end for

End.

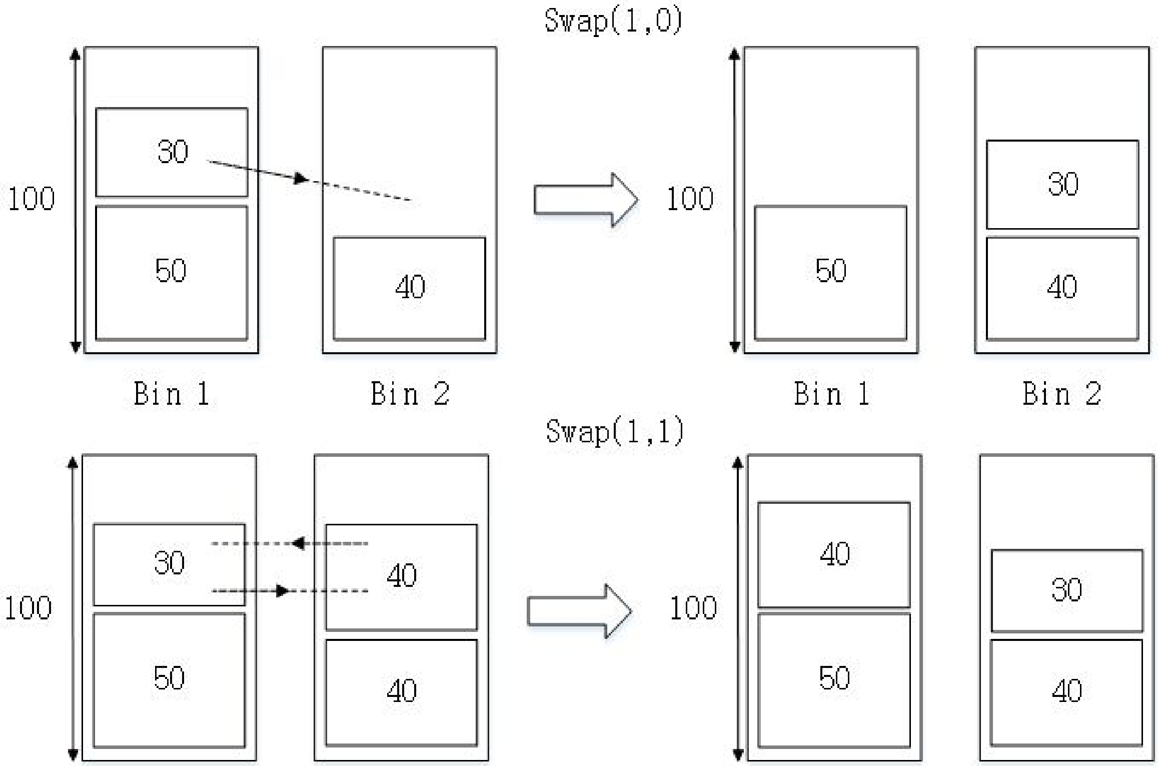

Fig.1. Swap(1,0) and Swap(1,1) operations

Kirkpatrick et al. [6] developed Simulated Annealing algorithm for economic activities in 1983. SA uses the Metropolis Criterion [7] and Boltzmann distribution to accept a new solution. For this purpose, function of acceptance probability is shown as follows:

$$P(\Delta E) = \begin{cases} 1 & if\ \Delta E < 0 \\ exp(-\Delta E/T) & otherwise \end{cases} \quad (7)$$

ΔE is difference between two neighbor solutions. The pseudocode of SA is as follows:

```
Begin
best ← Initialize first solution as best;
for ITER = 1 to ITER_target do
   while T > T_target
      next ← A randomly solution is created.
      ΔE ← E(next) – E(current);
      if (ΔE > 0) or exp(-ΔE / T) then
current ← next;
end if
   if current > best then
best ← current;
   end if
T ← cooling(T);
end while
end for
End.
```

In this paper, at first, FFD runs to place the objects for achieving max capacity of each bin. Secondly SA starts for searching new solutions. Two swapping techniques which are Swap(1,0) and Swap(1,1) are used to neighborhood searching by SA (see Fig.1).

IV. COMPUTATIONAL RESULTS

Computational tests have been carried out three different datasets (https://people.sc.fsu.edu/~jburkardt/datasets/bin_packing/bin_packing.html). Capacity of each bin is 100 and same for all datasets, weight of the objects for each dataset is shown in Table I.

SA parameter configurations are determined as T = 1000, T_{target} = 0.1, Cooling_Factor = 0.80, Iteration = 100. Objective function is used for accepting neighbor solution after swapping objects between bins and shown as follows:

$$max\ f = \sum_{i=1}^{m} \left(\sum_{j=1}^{k_i} w_j \right)^2 \quad (8)$$

where m = number of bins, k_i = number of objects in bin i and w_j = weight of object j.

TABLE I
DATASETS FOR BPP

DATASET	N	WEIGHTS
P01	9	w = {70, 60, 50, 33, 33, 33, 11, 7, 3}
P02	14	w = {99, 94, 79, 64, 50, 46, 43, 37, 32, 19, 18, 7, 6, 3}
P03	10	w = {49, 41, 34, 33, 29, 26, 26, 22, 20, 19}

TABLE II
RESULTS OF PROPOSED METHOD ON PROBLEM DATASETS

DATASET	f (FFD)	NB (FFD)	f (FFD+SA)	NB (FFD+SA)
p01	24908	4	25398	4
p02	58075	7	58083	7
p03	26513	4	29801	3

The algorithm is coded using Code Blocks as development platform with C++ programming language. Tests are performed on a CPU with Core 2 Duo 2.53 GHz running under Windows 7. Results on problems are presented in Table 2. Objective function f and number of bins (NB) are used to compare results. According to the results obtained, for all problems, f is better in results obtained using the FFD+SA. On the other hand NBs are same for P01 and P02 but SA improved quality of f function results for these problems. For the P03, NB is 4 for FFD, after SA is carried out NB is been 3. FFD+SA is obtained optimal results both f and NB for all problems.

CONCLUSION

In this paper, a proposed method is presented using FFD and SA methods for solving one dimensional BPP. In term of quality solutions, the method is capable of delivering good quality results for BPP datasets. In future, proposed method can be carried out larger datasets and the method can be parallelized to decreasing run time.

REFERENCES

[1] A. Scholl, R. Klein, and C. Jürgens, "Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem," Computers & Operations Research, vol. 24, pp. 627-645, 1997.

[2] A. C. Alvim, C. C. Ribeiro, F. Glover, and D. J. Aloise, "A hybrid improvement heuristic for the one-dimensional bin packing problem," Journal of Heuristics, vol. 10, pp. 205-229, 2004.

[3] K.-H. Loh, B. Golden, and E. Wasil, "Solving the one-dimensional bin packing problem with a weight annealing heuristic," Computers & Operations Research, vol. 35, pp. 2283-2291, 2008.

[4] K. Fleszar and K. S. Hindi, "New heuristics for one-dimensional bin-packing," Computers & Operations Research, vol. 29, pp. 821-839, 2002.

[5] I. Yildirim, "Bayesian Inference: Metropolis-Hastings Sampling," Dept. of Brain and Cognitive Sciences, Univ. of Rochester, Rochester, NY, 2012.

[6] S. Kirkpatrick and M. P. Vecchi, "Optimization by simulated annealing," science, vol. 220, pp. 671-680, 1983.

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," The journal of chemical physics, vol. 21, pp. 1087-1092, 1953.

★★★