A parallel approach of Best Fit Decreasing algorithm

DIMITRIS VARSAMIS
Technological Educational Institute
of Central Macedonia Serres
Department of Informatics Engineering
Terma Magnisias, 62124 Serres
GREECE
dvarsam@teicm.gr

FOTIOS CHANLIOGLOU
Technological Educational Institute
of Central Macedonia Serres
Postgraduate program in Applied Informatics
Terma Magnisias, 62124 Serres
GREECE
dvarsam@teicm.gr

Abstract: In this work we introduce a parallel approach of Best Fit Decreasing algorithm. The Best Fit Decreasing algorithm is heuristic and is used for optimal assignment problems, for example cutting stock problem, bin packing problem etc. The above problems for optimal assignment have very large computational complexity. For this reason have developed heuristic algorithms which aim at the reduction of computational time with disadvantage on solution. The Best Fit Decreasing compute, in most times, a approach of optimal solution. The purpose of the study is twofold: (a) to split the dataset of problem with representative manner so that at every sub-problem to Best Fit Decreasing algorithm is applied and the cost to the results to be the smallest and (b) to be implemented program in Matlab that will running every sub-problem with parallel techniques with the aim of reducing computational time.

Key–Words: Linear Programming, Optimization, Bin Packing Problem, Parallel Processing

1 Introduction

Given a set of numbers (object dimensions), and a fixed bin capacity, the bin packing problem is to assign each number to a bin so that the sum of all numbers assigned to each bin does not exceed the bin capacity. An optimal solution to a bin packing problem uses the fewest number of bins possible and the less total wastage in bins. The algorithm that solves the problem of assigning objects to bins is of high computational complexity and is one of the classic NP-complete problems [1].

Approximation algorithms have been developed for the problem of assigning objects to bins is First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) [2], [3]. The above algorithms sort the data in descending order, and place each new element in the first bin to which fits (FFD), in the bin to which fits best (BFD).

Execution of the FFD and BFD algorithms requires large computational costs [4], which leads the researchers to apply computational methods with parallel and distributed processing to either FFD or BFD algorithms, in order to reduce the execution time of the algorithms. Matlab is a software tool that supports numerical computations and parallel processing [5], [7], [8]. Additionally, Matlab supports parallel processing either on a network of PCs or on a multicore PC.

In section 2, the serial BFD algorithm with an illustrative example is presented. In section 3 the data partition procedure with an illustrative example is presented. Then, the parallel BFD algorithm with an illustrative example is presented in section 4. Finally, in section 5 the implemented algorithms are tested for performance. The tests include both serial and parallel implementations of BFD algorithm using Matlab software tools and commands.

2 The serial BFD algorithm

Let the following numbers are 20 object sizes that we want to place in bins of the fixed capacity with size 10, using as less as possible. Also the wastage (unused space in bins) must be the minimum. The sizes of the objects are in descent sorting.

$$\begin{array}{cccccccc} 8 & 7 & 7 & 7 & 6 & 5 & 4 & 3 \\ 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 \end{array} \tag{1}$$

By the execution of BFD algorithm in data (serial execution), we get the following results

$bins$	1	2	3	4	5	6	7	8
	8	7	7	7	6	5	4	3
		1	1		2	3	3	2
$wastage$	0	0	0	1	0	0	1	3

We notice that the total wastage size is 5 and the number of used bins are 8.

3 Data Partition

The most simple idea to partition the data is the cycled assignment. Specifically, the cycled assignment with n elements and m groups place the first element in first group, the second element in second group,..., the m element in m group. Afterwards, the cycled assignment place the $m+1$ element in first group, the $m+2$ element in second group,..., the $m+m$ element in m group. The cycled assignment is continued until the elements finish.

3.1 Case of two partitions

For the dataset (1) the cycled assignment for $n = 16$ and $m = 2$ is applied.
Thus, we have two partitions with group A and B which is given below

8	7	7	7	6	5	4	3
A	B	A	B	A	B	A	B

3	3	3	2	2	2	1	1
A	B	A	B	A	B	A	B

Thus, the group A is

$$8 \quad 7 \quad 6 \quad 4 \quad 3 \quad 3 \quad 2 \quad 1$$

and the group B is

$$7 \quad 7 \quad 5 \quad 3 \quad 3 \quad 2 \quad 2 \quad 1$$

3.2 Case of four partitions

For the dataset (1) the cycled assignment for $n = 16$ and $m = 4$ is applied.
Thus, we have four partitions with group A, B, C and D which is given below

8	7	7	7	6	5	4	3
A	B	C	D	A	B	C	D

3	3	3	2	2	2	1	1
A	B	C	D	A	B	C	D

Thus, the group A is

$$8 \quad 6 \quad 3 \quad 2$$

the group B is

$$7 \quad 5 \quad 3 \quad 2$$

the group C is

$$7 \quad 4 \quad 3 \quad 1$$

and the group D is

$$7 \quad 3 \quad 2 \quad 1$$

4 The Parallel BFD algorithm

In parallel execution we apply the serial BFD Algorithm in each group. The advantage of this method is the reduction of execution time. The disadvantage is the worst results instead to serial execution of the BFD algorithm.

4.1 Case of two partitions

For the dataset (1) with two partitions we apply the BFD algorithm in group A and B.
For group A that is

$$8 \quad 7 \quad 6 \quad 4 \quad 3 \quad 3 \quad 2 \quad 1$$

we get the following results

$bins$	1	2	3	4	5
	8	7	6	4	3
		1	2	3	
$wastage$	0	0	0	1	5

For group B that is

$$7 \quad 7 \quad 5 \quad 3 \quad 3 \quad 2 \quad 2 \quad 1$$

we get the following results

$bins$	1	2	3	4
	7	7	5	3
	1		3	2
				2
$wastage$	0	1	0	1

The wastage (unused space in bins) in group A is 6 and the number of used bins are 5. The wastage (unused space in bins) in group B is 2 and the number of used bins are 4.

Thus, in parallel execution of BFD algorithm in two partitions we have total wastage 8 and the number of used bins are 9.

4.2 Case of four partitions

For the dataset (1) with four partitions we apply the BFD algorithm in group A, B, C and D.
For group A that is

$$8 \quad 6 \quad 3 \quad 2$$

we get the following results

$bins$	1	2	3
	8	6	3
		2	
$wastage$	0	0	5

For group B that is

$$7 \quad 5 \quad 3 \quad 2$$

we get the following results

$bins$	1	2	3
	7	5	2
		3	
$wastage$	1	0	6

For group C that is

$$7 \quad 4 \quad 3 \quad 1$$

we get the following results

$bins$	1	2
	7	4
	1	3
$wastage$	0	1

For group D that is

$$7 \quad 3 \quad 2 \quad 1$$

we get the following results

$bins$	1	2
	7	3
	1	2
$wastage$	0	3

The wastage (unused space in bins) in group A is 5 and the number of used bins are 3. The wastage (unused space in bins) in group B is 7 and the number of used bins are 3. The wastage (unused space in bins) in group C is 1 and the number of used bins are 3. The wastage (unused space in bins) in group D is 3 and the number of used bins are 2.

Thus, in parallel execution of BFD in four partitions we have total wastage 16 and the number of used bins are 10.

4.3 Total results

The total results are shown in Table 1

Consequently, we see that as the number of groups in which data is divided increases, both the wastage and the number of used bins increase. This is dues to the pattern of partition of the groups and to the individual statistical values that have the data. The more representative the partition of the groups, the less variations in the results will be. It is important to note that the volume of data used to analyze the example is very small. Experimental measurements respect to execution time, wastage and number of used bins will then be presented.

5 Performance tests and Results

The performance tests are implemented in an efficient computing system with the following characteristics: CPU Intel Xeon E5640 64x 2.67GHz (multicore) and RAM 16GB. Additionally, for the accuracy of the performance tests, the execution time of the tests are calculated with the formula is given by

$$Time = \frac{t_1 + t_2 + t_3 + \ldots + t_{12} - t_{\max} - t_{\min}}{10}$$

where t_i with $i = 1, 2, \ldots, 12$ are the execution time of each run with the same data and parameters, $t_{\max}$ is maximum execution time of t_i and $t_{\min}$ is the minimum execution of t_i, respectively [9].

The BFD algorithm is implemented in Matlab and the parameters of the BFD algorithm are the following:

- mat, the matrix of dataset.
  Each object of dataset is a random integer number between 1 and 100. The corresponding data are created using the build-in function of Matlab `randi()`.
- C the length (size) of bin.

  case 1 $C = 100$, the length of elements are less or equal to C

  case 2 $C = 200$, the length of elements are less or equal to $\frac{C}{2}$

The performance tests are implemented respect to the number of objects in dataset (n) and respect to the number of partitions-cores (p). In particular, the values of parameters are $n = 2^{14}$, 2^{16}, 2^{18} and $p = 1$, 2, 4, 8.

5.1 Results with $C = 100$

The results in terms of wastage are shown in table 2, in terms of number of used bins are shown in table 3 and in terms of execution time are shown in table 4, respectively.

From the Table 2 we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{14}$

$$RW^2_{2^{14}} = \frac{1136 - 1036}{1036} = 0,097 = 9,7\%$$

$$RW^4_{2^{14}} = \frac{1236 - 1036}{1036} = 0,193 = 19,3\%$$

$$RW^8_{2^{14}} = \frac{1236 - 1036}{1036} = 0,193 = 19,3\%$$

also, we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{16}$

$$RW^2_{2^{16}} = \frac{4319 - 4219}{4219} = 0,024 = 2,4\%$$

Table 1: Comparison of Serial and Parallel Execution

	Total Wastage	Total number of used Bins
Serial Execution of BFD	5	8
Parallel Execution of BFD (2 partitions)	8	9
Parallel Execution of BFD (4 partitions)	16	10

Table 2: The wastage of BFD vs the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	1036	1136	1236	1236
2^{16}	4219	4319	4319	4319
2^{18}	14320	14420	14420	14620

$$RW^4_{2^{16}} = \frac{4319 - 4219}{4219} = 0,024 = 2,4\%$$

$$RW^8_{2^{16}} = \frac{4319 - 4219}{4219} = 0,024 = 2,4\%$$

and we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{18}$

$$RW^2_{2^{18}} = \frac{14420 - 14320}{14320} = 0,007 = 0,7\%$$

$$RW^4_{2^{18}} = \frac{14420 - 14320}{14320} = 0,007 = 0,7\%$$

$$RW^8_{2^{18}} = \frac{14620 - 14320}{14320} = 0,021 = 2,1\%$$

Table 3: The number of used bins of BFD vs the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	8227	8228	8229	8229
2^{16}	33041	33042	33042	33042
2^{18}	132897	132898	132898	132900

From the Table 3 we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{14}$

$$RB^2_{2^{14}} = \frac{8228 - 8227}{8227} = 0,0001 = 0,01\%$$

$$RB^4_{2^{14}} = \frac{8229 - 8227}{8227} = 0,0002 = 0,02\%$$

$$RB^8_{2^{14}} = \frac{8229 - 8227}{8227} = 0,0002 = 0,02\%$$

also, we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{16}$

$$RB^2_{2^{16}} = \frac{33042 - 33041}{33041} = 0,00003 = 0,003\%$$

$$RB^4_{2^{16}} = \frac{33042 - 33041}{33041} = 0,00003 = 0,003\%$$

$$RB^8_{2^{16}} = \frac{33042 - 33041}{33041} = 0,00003 = 0,003\%$$

and we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{18}$

$$RB^2_{2^{18}} = \frac{132898 - 132897}{132897} = 0,000008 = 0,0008\%$$

$$RB^4_{2^{18}} = \frac{132898 - 132897}{132897} = 0,000008 = 0,0008\%$$

$$RB^8_{2^{18}} = \frac{132900 - 132897}{132897} = 0,000024 = 0,0024\%$$

Table 4: The execution times of BFD the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	1,2308	0,4359	0,3633	0,6056
2^{16}	13,6948	3,6948	1,3277	0,9903
2^{18}	204,9967	53,2644	17,0844	7,3111

From the Table 4 we calculate the relative execution time (RT) respect to execution time with $p = 1$ and $n = 2^{14}$

$$RT^2_{2^{14}} = \frac{0,4359 - 1,2308}{1,2308} = -0,646 = -64,6\%$$

$$RT^4_{2^{14}} = \frac{0,3633 - 1,2308}{1,2308} = -0,705 = -70,5\%$$

$$RT^8_{2^{14}} = \frac{0,6056 - 1,2308}{1,2308} = -0,501 = -50,1\%$$

also, we calculate the relative number of used bins (RT) respect to number of used bins with $p = 1$ and $n = 2^{16}$

$$RT^2_{2^{16}} = \frac{3,6948 - 13,6948}{13,6948} = -0,73 = -73\%$$

$$RT^4_{2^{16}} = \frac{1,3277 - 13,6948}{13,6948} = -0,903 = -90,3\%$$

$$RT^8_{2^{16}} = \frac{0,9903 - 13,6948}{13,6948} = -0,928 = -92,8\%$$

and we calculate the relative number of used bins (RT) respect to number of used bins with $p = 1$ and $n = 2^{18}$

$$RT^2_{2^{18}} = \frac{53,2644 - 204,9967}{204,9967} = -0,741 = -74,1\%$$

$$RT^4_{2^{18}} = \frac{17,0844 - 204,9967}{204,9967} = -0,917 = -91,7\%$$

$$RT^8_{2^{18}} = \frac{7,3111 - 204,9967}{204,9967} = -0,964 = -96,4\%$$

5.2 Results with $C = 200$

The results in terms of wastage are shown in table 5, in terms of number of used bins are shown in table 6 and in terms of execution time are shown in table 7, respectively.

Table 5: The wastage of BFD vs the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	136	336	536	736
2^{16}	119	119	319	919
2^{18}	20	220	620	820

From the Table 5 we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{14}$

$$RW^2_{2^{14}} = \frac{336 - 136}{136} = 1,471 = 147,1\%$$

$$RW^4_{2^{14}} = \frac{536 - 136}{136} = 2,941 = 294,1\%$$

$$RW^8_{2^{14}} = \frac{736 - 136}{136} = 4,412 = 441,2\%$$

also, we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{16}$

$$RW^2_{2^{16}} = \frac{119 - 119}{119} = 0 = 0\%$$

$$RW^4_{2^{16}} = \frac{319 - 119}{119} = 1,681 = 168,1\%$$

$$RW^8_{2^{16}} = \frac{919 - 119}{119} = 6,723 = 672,3\%$$

and we calculate the relative wastage (RW) respect to wastage with $p = 1$ and $n = 2^{18}$

$$RW^2_{2^{18}} = \frac{220 - 20}{20} = 10 = 1000\%$$

$$RW^4_{2^{18}} = \frac{620 - 20}{20} = 30 = 3000\%$$

$$RW^8_{2^{18}} = \frac{820 - 20}{20} = 40 = 4000\%$$

Table 6: The number of used bins of BFD vs the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	4109	4110	4111	4112
2^{16}	16500	16500	16501	16504
2^{18}	66377	66378	66380	66381

From the Table 6 we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{14}$

$$RB^2_{2^{14}} = \frac{4110 - 4109}{4109} = 0,0002 = 0,02\%$$

$$RB^4_{2^{14}} = \frac{4111 - 4109}{4109} = 0,0004 = 0,04\%$$

$$RB^8_{2^{14}} = \frac{4112 - 4109}{4109} = 0,0006 = 0,06\%$$

also, we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{16}$

$$RB^2_{2^{16}} = \frac{16500 - 16500}{16500} = 0 = 0\%$$

$$RB^4_{2^{16}} = \frac{16501 - 16500}{16500} = 0,00006 = 0,006\%$$

$$RB^8_{2^{16}} = \frac{16504 - 16500}{16500} = 0,00024 = 0,024\%$$

and we calculate the relative number of used bins (RB) respect to number of used bins with $p = 1$ and $n = 2^{18}$

$$RB^2_{2^{18}} = \frac{66378 - 66377}{66377} = 0,00002 = 0,002\%$$

$$RB^4_{2^{18}} = \frac{66380 - 66377}{66377} = 0,00006 = 0,006\%$$

$$RB^8_{2^{18}} = \frac{66381 - 66377}{66377} = 0,00008 = 0,008\%$$

Table 7: The execution times of BFD the number of partitions (p) and the number of objects in dataset n

n \ p	1	2	4	8
2^{14}	0,6323	0,3257	0,3387	0,5871
2^{16}	6,7229	1,9058	0,8281	0,8148
2^{18}	97,8441	25,2299	8,0345	3,5785

From the Table 7 we calculate the relative execution time (RT) respect to execution time with $p = 1$ and $n = 2^{14}$

$$RT^2_{2^{14}} = \frac{0,3257 - 0,6323}{0,6323} = -0,485 = -48,5\%$$

$$RT^4_{2^{14}} = \frac{0,3387 - 0,6323}{0,6323} = -0,464 = -46,4\%$$

$$RT^8_{2^{14}} = \frac{0,5871 - 0,6323}{0,6323} = -0,0715 = -7,15\%$$

also, we calculate the relative number of used bins (RT) respect to number of used bins with $p = 1$ and $n = 2^{16}$

$$RT^2_{2^{16}} = \frac{1,9058 - 6,7229}{6,7229} = -0,717 = -71,7\%$$

$$RT^4_{2^{16}} = \frac{0,8281 - 6,7229}{6,7229} = -0,877 = -87,7\%$$

$$RT^8_{2^{16}} = \frac{0,8148 - 6,7229}{6,7229} = -0,879 = -87,9\%$$

and we calculate the relative number of used bins (RT) respect to number of used bins with $p = 1$ and $n = 2^{18}$

$$RT^2_{2^{18}} = \frac{25,2299 - 97,8441}{97,8441} = -0,742 = -74,2\%$$

$$RT^4_{2^{18}} = \frac{8,0345 - 97,8441}{97,8441} = -0,918 = -91,8\%$$

$$RT^8_{2^{18}} = \frac{3,5785 - 97,8441}{97,8441} = -0,963 = -96,3\%$$

6 Conclusion

From the results of performance tests it becomes evident that the parallel implementations of Best Fit Decreasing algorithm lead to execution time reduction. The parallel implementations of Best Fit Decreasing algorithm approach the results of corresponding serial implementation. The differences in results between serial and parallel implementation in case 1 ($C = 100$), which mean that the size of object is less or equal to the size of bin, are: (a) the relative wastage (RW) is increasing with low rate but the wastage is low to relation of the total size of objects, (b) the relative number of used bins (RB) is up to $0,02\%$ while (c) the relative execution time (RT) is decreasing from at least 50% up to 97%.Thus, the parallel implementation improves the execution time of BFD algorithm especially in large dataset, namely, $n = 2^{16}$ and $n = 2^{18}$.

The differences in results between serial and parallel implementation in case 2 ($C = 200$), which mean that the size of object is less or equal to the half size of bin, are: (a) the relative wastage (RW) is increasing significantly but the wastage is very low to relation of the total size of objects, (b) the relative number of used bins (RB) is up to $0,06\%$ while (c) the relative execution time (RT) is decreasing from at least 71% up to 97% for $n = 2^{16}$ and $n = 2^{18}$. For $n = 2^{14}$, small dataset, the relative execution time is not efficient. Thus, the parallel implementation in case of $= 200$ improves the execution time of BFD algorithm in large dataset, namely, $n = 2^{16}$ and $n = 2^{18}$.

Consequently, in case 2, where the size of object is less or equal to the half size of bin, we have less effectiveness instead to case 1 for the relative wastage and relative execution time, respectively.

It needs to be noticed that both serial and parallel Matlab implementations run on the same computing system, with the same resources. The parallel algorithms take advantage of the resources of the computing system, namely the cores of CPU.

Acknowledgements: The authors wish to acknowledge financial support provided by the Research Committee of the Technological Education Institute of Central Macedonia, under grant SAT/IC/01112017-191/11.

References:

[1] M. R. Garey, D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness,* W. H. Freeman & Co., New York, NY, USA, 1990.

[2] R. E. Korf, A New Algorithm for Optimal Bin Packing, *Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada,* (2012), 731-736.

[3] A. N. Zehmakan, Bin Packing Problem: Two Approximation Algorithms, *CoRR,* (2015), abs/1508.01376

[4] D. S. Johnson, A. J. Demers, J. D. Ullman, M. R. Garey, R. L. Graham, Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms, *SIAM J. Comput.,* **4** (1974), 299-325.

[5] P. Luszczek, Parallel Programming in MATLAB, *International Journal of High Performance Computing Applications,* **23** (2009), no. 3, 277-283.

[6] C. Moler, Parallel MATLAB: Multiple processors and multiple cores, *The MathWorks News & Notes,* (2007).

[7] G. Sharma, J. Martin, MATLAB : A Language for Parallel Computing, *International Journal of Parallel Programming,* **37** (2009), no. 1, 3-36.

[8] D. N. Varsamis, C. Talagkozis, P. A. Mastorocostas, E. Outsios, N. P. Karampetakis, The performance of the MATLAB Parallel Computing Toolbox in specific problems, *Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, Poland,* (2012), 587-593.

[9] D. N. Varsamis, P. A. Mastorocostas, A. K. Papakonstantinou, N. P. Karampetakis, A parallel searching algorithm for the insetting procedure in Matlab Parallel Toolbox, *Advanced Information Science and Applications Volume I, 18th Int. Conf. on Circuits, Systems, Communications and Computers (CSCC 2014), July 17-21, 2014, Santorini Island, Greece,* (2014), 145-150.