Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), March 3 – 5, 2015

Fragmentable Group and Item Bin Packing with Compatibility Preferences

Aderemi Adeyemo
Department of Industrial and Production Engineering,
University of Ibadan,
Ibadan, Nigeria.
aderemiadeyemo1@gmail.com

Victor O. Oladokun
Department of Industrial and Production Engineering,
University of Ibadan
Ibadan, Nigeria

***Abstract*—we consider a special class of the bin packing problems having the following distinctive feature; items are grouped based on their subset of elements. Splitting of group is to be minimized while compatibility of items placed in same bins is of utmost priority.**

The problem, denoted as the fragmentable group and item bin packing with Item compatibility preferences, has a very interesting structure and models real life application ranging from industrial, educational and the financial sectors.

According to our knowledge, the FGIBPCP has never been tackled in the literature. In this paper we addressed if for the first time. We examined the theoretical properties which has some resemblance to some seen in the literature. Then we elaborate it to dense a mathematical model.

A new offline heuristic we referred to as RHEMA 1 heuristic was proposed and was used to search for solution and implemented manually for small size problem. It yielded a better result than that of an online algorithm. Further experimentation with will be carried out to analyze the performance of the solution in the future research.

Keywords— Bin packing; heuristic; group; item compatibility; subset.

I. Introduction

The bin packing problem is an interesting problem. It is a well studied problem.[17].[21] actually claimed it is most studied problems in combinatorial optimization. This can be ascribed to the fact that the bin packing problem has great generality and practical relevance has been described to represent one of the packing problem most of the most fundamental, and still active, research fields in combinatorial optimization [17].

Packing problems are very relevant to operations research and many as it has been undivided that there is an important connection between bin packing and other very important collection of research questions [24].

Many variations including dimensional diversities have been identified; 2-dimesional BPP, 3-dimesional BPP, online[13] and offline[25], dynamic bin packing[7], variable size bin packing[3,6,12,13], bin packing with uncertain volumes and capacities[24], bin packing with precedence constriants [15] etc. see[24][1]

Applications found in literature in include knapsack problems, multiple knapsacks,[14,19] project scheduling, job shop scheduling production planning [19] and supply chain management.[21]

However [17] emphasized a need for more sophisticated models that can be able to capture more application details and provide more optimization powers. [17] also indicated that the introduction of which Bin Packing with item fragmentation suit the crucial optimization needs. They also noted this variant has gained little attention in the literature. it is used to move the VLSI circuit design, POADM and CATV[17,21]. It is proved to be NP-Hard [9], [10], and both approximation results and the improved version are presented.[17]

A definition of fragmantable item packing is given in [17];

“Fragmentable object bin packing [FOBP] is the decision problem of packing N objects each of size ti, i = 1,... ,N, into m bins each of size M, M > 1 such that the capacity used up while packing each object whether whole or after fragmentation] of size k is k + 1.”

Also, in [17] it was noted that the common assumption in the packing problems is that item may not be fragmented and this has limited the attention given for this subject of item fragmentation in bin packing problem. Developing solution methods (exact and heuristics) remain an area open to research.[1,21]

Few variants of FOBP [some authors like [14] refer to it as BPPIF] have been discovered in the literature. The FIBP with size increasing fragmentation as indicated. These are different variants of BPPIF discussed in literature; one is the BPP with size increasing fragmentation [BPPSIF] where overhead units are added to the size of each fragment. Another is the BPPIF size preserving Fragmentation [BPPSPF], where the sum of fragment regulate equal so that of the original item, that is no overhead is introduced by splitting in this case. Other variations are included that in which the number of this is fixed and items are fragmentable but without additional space. This variant is useful for modelling splittable resource allocation [22][17].

Another important variation of the FOBP important in practice is the minimization bins including fragmented items, and the minimization of the fragmentations of fragmented items. [17] indicated that this version is a very BPPSPF cutting edge technology optimization.

An extension or other variants of the FOBP might be pertinent in practice. There might be some desired or necessary compatibilities between fragments that occupy same bin. Such problems is tagged -"*fragmentable group and item bin packing with Item compatibility preferences*,(FGIBPCP)". It also aims at fragment minimization and fragmentations minimization but with a consideration on compactibity of items in the same been based on their constituent elements. [see Fig 1a&1b]

To the best of our knowledge, this variant has not been tackled as a bin packing, its mathematical model nor solution method designed.

Our aim is to investigate a typical case of this variant whose motivation came from a real life problem. First, we examine the theoretical properties some of which are peculiar to other variants examined in the literature. These theoretical findings are discussed in Section 2. In section 3, we discuss the compact mathematical programming formulations for the problem and presented the resulting model. In section 4, we describe the main ingredients of our algorithms and the resulting heuristic method was used to solve a small sample problem and examined the solution compared to results of another solution method. Finally we summarize our results and collect some brief but beneficial results.

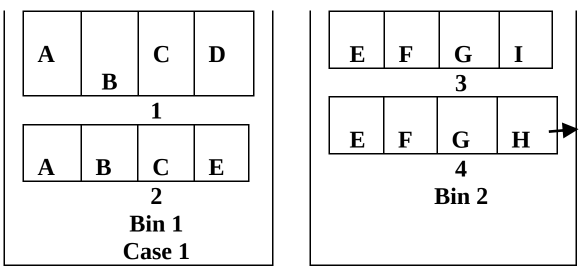

Fig 1b. Example of a case of preference in arrangement of items into bins. Case 1

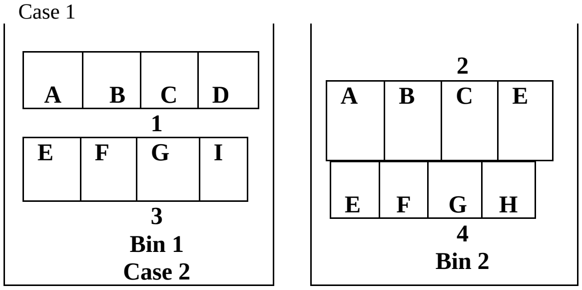

Fig 1b. Example of a case of preference in arrangement of items into bins. Case 1 is preferred to case 2 as items 1 and two have elements A,B,C in common unlike 2..

II. MODELLING

General Problem Statement

The scheduling problem therefore involves the process of assigning items i ϵ I classified to a set of groups j ϵ J based on the specific elements k ϵ K in their subsets to bins or time slots with its main objective to minimize the scattering of elements across the bins, that is, minimizing total number of element's appearance in the bins. This is to be achieved without violating any necessary constraints.

Assumptions.

1. Each Item's constituent element are known and predetermined and no item has in its set, any element appearing twice.
2. The number of elements in each item's group is the same.
3. Each bin has the same capacity.

In this model, the following indices are used:

Let,

i be the item index i=1,2,...I

j be the item group index j=1,2,...G

k be the element index k=1,2,…K

p be the bin index p= 1,2,…P.

The following symbols are used for parameters and variables.

S = set of all elements available.

I = Total Number of items

K= Total number of elements that must be assigned because they are contained in at least an item. (k= 1,2,….K). H is a subset of S.

S_{kj}- set of elements k contained by each items group j

G= Total number of groups

P = Total number of bins

$K_{p\,=}$ Maximum capacity of bin p

N_s= Number of elements in each item

The assumptions show that N_{ij} and S_{jk} is known or given.

N_{ik}= Number of items i that N_i has element k

S_{jk}= 1, if item group j has element k in its subset

= 0, otherwise.

Variables

Let, $x_{kp\,=}$1, if element h is scheduled for bin p

= 0, otherwise

$y_{ikp\,=}$1, if items i has element k in its subset and it is assigned to bin p,

= 0, otherwise

N_{ikp}= Number of items i that has element k and are assigned to bin p i.e number of times element k appear in a bin.

N_{ip}= Number of items i that and are assigned to bin p.

N_{ik}= Number of items i that has element k

N_{ikp}= Number of items i that has element k and are assigned to bin p i.e number of times element k appear in a bin.

N_{ijp}= Number of items i that belongs to group j and are assigned to bin p.

Model

Using the above symbols, parameters and variables, the mathematical representation of the model goes thus

Minimize:

$$\sum_{k=1}^{K} \sum_{p=1}^{P} x_{kp} \quad (1)$$

Subject to:

$$y_{ikp} \;/N_{ip} \le x_{kp} \qquad \forall i \epsilon I, \forall\ p \epsilon P\ \forall\ k \epsilon K \quad (2)$$

$$\sum_{i=1}^{I}\ , N_{ikp} = N_{kp} \qquad \forall\ p \epsilon P, \forall\ k \epsilon K \quad (3)$$

$$\sum_{p=1}^{P}\ x_{kp} \ge 1, \qquad \forall\ k \epsilon K, \quad (4)$$

$$\sum_{j=1}^{G}\ . N_{ij} S_{jk} = N_{ik} \qquad \forall i \epsilon I, \forall\ k \epsilon K \quad (5)$$

$$\sum_{p=1}^{P} = N_{ij} \quad \forall i \epsilon I, \forall j \epsilon G \quad (6)$$

$$\sum_{j=1}^{G} \sum_{p=1}^{P} N_{jp} = \mathrm{I} \quad \forall j \epsilon G, \forall p \epsilon P \quad (7)$$

$$\sum_{i=1}^{I} \sum_{j=1}^{G} N_{ijp} = K_p. \forall p \epsilon P \quad (8)$$

$$x_{kp}, y_{ikp}, S_{jk} \in (0,1),$$

$$N_{ip}, N_{ijp}, N_{ij}\ N_{jp}, N_{ikp} N_{ik} = \mathbb{Z}\ \forall i \epsilon I, \forall j \epsilon G, \forall p \epsilon P, \forall k \epsilon K, \quad (9)$$

Variable N_{ikp} as earlier described is the number of items scheduled for bin p that possess an element k in its set. Constraints (3) and (9) shows N_{ikp} is modeled as an integer to be included between 0 and N_{ip}, that is, the sum of the fragments of items having element k scheduled to different bins must be equal to the total number of items that possess the elements.

As soon as any item offering element k is assigned to bin p, even if it is just one item, X_{kp} is equal to 1. Otherwise X_{kp} is equal to 0. So X_{kp} is a bivalent variable, equal to 0 or 1 as shown in constraint (9) that detects scheduling of element k to bin p.

Constraint (2) forces variable X_{kp} to equal to 1 as soon as variable N_{ip} is shortly greater than 0. Constraint(4) ensures each element; $k \in Sj$ must be scheduled to at least one bin, whereas (5) shows the relationship between items total no of items offering a particular elements and the total no of items in a group. It indicates that every item that has element k is assigned to exactly a group.

Constraints (6) and (7) shows that every item i must be assigned to exactly a bin p. i.e. sum of items that belongs to a group fragmented to different bins p must be equal to the total number of items in that group. Also (7) shows that sum of all items fragments in different bins must be equal to the number of items in an items group.

Constraints (8) shows bin capacity constraints must be respected i.e. no bin can take more than its capacity, K_p.

Properties

The structural properties of the problem in order to reduce the search place for an optimal solution based on the assumptions earlier stated. The two variables we are looking for are;

With the earlier stated assumption the following propositions observations and with proofs in[17], [21], will aid the heuristic in getting a good solution.

Property1. In this case where all the bins have the same capacity K, if the sum of the sizes of two items 1 and 2 is equal to K, there is at least one optimal solution in which these items 1 and 2 are placed in the same bin. So an exact cover is preferred in assigning items to bins

Property 2. if an item Group g ЄG has weight N_g > K, then an optimal solution can always be obtained by assigning a fragment K/N_g to a single bin, completely filling its capacity, and by considering a residual instance having |K| - 1 bins, and in which N_g is reduced by K.

Property 3. Even in the case where all the bins have the same capacity C, if the sum of the sizes of three items j, k and l is equal to this capacity C, it could happen that there is no optimal solution where these three items are both placed in the same bin.

Proof [21]Let us consider the following instance:

• we have four bins of capacity 7,

• we have ten items, one of size 5, two of size 1 and seven of size 3.

The solution where the item of size 5 and the two items of size 1 are placed in the same bin gives 12 fragments because two cuts are needed for the seven items of size 3 in the three remaining bins. The optimal solution is composed of 11 fragments with only one cut: two bins with items 3+3+1 and one cut for the remaining items in the two last bins.

Property 4. [21] The value of the optimal solution is at most equal to N + M − 1 (with M the number of bins and N the number of items). This property is essential to setting the upper bound in searching for solution

Proof. An obvious heuristic consists in sequentially placing the items in any order into the bins also taken in any order. If the current item cannot be entirely put in the current bin, it is cut in two fragments, the first one to complete the current bin and the second starts the next bin. This heuristic provides a solution (and thus an upper bound) where there are at most M − 1 cuts. We then have at most N + M − 1 fragments.

Property 5. if an item Group g ЄG has weight N_{ij} > K, then an optimal solution can always be obtained by assigning a fragment K_p/N_{ij} to a single bin, completely filling its capacity, and by considering a residual instance having |K| - 1 bins, and in which N_{ij} is reduced by K.

Property 6. In the case where all the bins have the same capacity K_p, if an item i has a size greater than K_p, there is at least one optimal solution in which this item i completely fills $\lfloor N_i/K_p \rfloor$ bins. The reduced problem contains N or N −1 (if the previous fraction is integer) items and $(M - \lfloor N_i/K_p \rfloor)$ bins. By repeating this operation on all the items whose size is greater than C, we obtain an equivalent problem in which the sizes of the items are all strictly lower than C.

Observation 1: Each solution minimizing the overall number of fragmentations also minimizes the overall number of fragments. Just like in this structure, the number of bins that will give the optimum solution can be guessed intelligently thereby forming a reasonable lower bound for number of bins p, where I is the total no of items divided by the capacity assuming bins are of same capacity is the same

$$P = I/C \quad (10)$$

P approximated to larger whole value

These observations and propositions were useful in developing the Heuristics.

III. ALGORITHM AND HEURISTICS

Algorithm

Solving a big size problem of the FGIBPCP is complex and time consuming. It is not treated in this paper. We propose a search heuristic called RHEMA-1 heuristic which reduces fragments, fragmentation and still seek to maximize compatibility arrangement of items in the bins to be different The idea is from a man called Paul [23] who seek problem conflicting things at once in order to gain some people to believe the son of God. RHEMA-1 uses the formulated model for implementation.

RHEMA-1 search Algorithm

A search heuristic called RHEMA-1 heuristic was proposed. It seeks to reduce fragments, fragmentation and still seek to maximize compatibility arrangement of items in the bins. RHEMA-1 is an improvement on the Best Fit Decreasing Item Fragmentation (BFD_f) algorithm [16].

RHEMA 1 Heuristics: we sort the items (Groups) in non-increasing size order, and we pack them without fragmentations using a Best-Fit BPP heuristic, until the sum of residual capacity in the bins is at least equal to the sum of the weights of unpacked items; then we exploit the residual capacity with a NF_f . We complete the solution by breaking ties between items of the same size giving preference to compatibility of the elements that make up the groups. This regroups the items into the bins to give an improved solution.

Result computing algorithm

We proposed a result computing algorithm which is required because of the nature and complexity of the model and the model and also because the approach is manual.

Step 1: create a table with P+2 rows and K+2 columns. Input the bins p = (1…P) on the first column starting from 2^{nd} row, and elements k= (1…k) on 1st row. Fill on the second column the schedule result of search algorithm i.e. the item groups that are scheduled for each bin.

Step 2: Match each bin p with the elements k and initialize making all cells zero.

Step 3: Every cell in intersection of elements k found in bin p from the bin table is turned 1.

Step 4: Do these for all the cells. No duplication one a cell is turned 1.

Step 5: Add the total number of all the '1's across each row. The resulting solution gives each $\sum x_{kp}$- for each element k. the resulting total after adding across the column is the minimal value of the solution.

Step 56 compute total values for each row of elements. The resulting total is the number of times element k is rescheduled.

The solution from this algorithm ensures that the constraints of the model are satisfied.

University Entrance Examination Scheduling Problem.

The university Entrance examination considered is a one-time exam where candidates sit for all the subjects they register for at once.

Candidates and parents sometimes have doubted the credibility of some of these processes due the variability of question types for different sets of items offering same elements. This has primarily been due to insufficient capacity both in space and manpower of organizing institutions to attend to the demands posed by qualified intending candidates *at once* which has led to scheduling of exams to more than a period or day. In an attempt to minimize cheating, repetition of questions of same papers which are scheduled for more than a day has to be avoided. Consequentially, this has sometimes led to unwarranted reduction of candidates allowed to write exams and/or testing candidates with different set of questions. It has however reduced the fairness of testing students.

The scheduling problem therefore involves the process of assigning candidates classified to a set of groups based on the specific subjects they offer to periods or time slots with its main objective to minimize the total number of question types while satisfying a given number of constraints.

We modeled using the following indices and some others

Students - Item i
Student exam set - Group j, e.g. j=1;(A,B,C,D)
Subjects - Element k
Exam period or day - Bin p
Exam Centre capacity - K
Total no of student – C

All of this and many others form corresponding notations for variables and parameters already discussed in section (II).

TABLE I. STUDENT GROUP DOING SAME SUBJECTS

Candidates group j	*Subject (Elements k)*	*No of candidates in each group(N_{ij})*
1	A,B,C,D	50
2	A,C,E,K	40
3	A,B,D,E	35
4	A,E,F,G	15
5	A,F,G,C	10
6	A,B,C,K	20
7	C,D,E,F	45
8	C,E,G,K	15
9	C,G,B,K	45

TABLE II: SHOWING SOLUTIONS GENERATED FOR AN ONLINE HEURISTIC

	EL	A	B	C	D	E	F	G	K	
X	**1**	1	1	1	1					4
Y	**2,3**	1	1	1	1	1			1	6
Z	**3,4,5**	1	1	1	1	1	1	1		7
V	**5,6,7**	1	1	1	1	1	1	1	1	8
W	**7,8,9**		1	1	1	1	1		1	6
T	**9**		1	1				1	1	4
		4	5	6	5	4	3	3	3	**35**

TABLE III: SHOWING SOLUTIONS FOR CLASSICAL BFDF

B	**EL**	A	B	C	D	E	F	G	K	
X	1	1	1	1	1					4
Y	9		1	1				1	1	4
Z	7			1	1	1	1			4
V	2	1		1		1			1	4
W	3,8	1	1	1	1	1		1	1	7
T	6,4,5	1	1	1		1	1	1	1	7
		4	4	6	4	3	2	3	4	**30**

TABLE IV: SHOWING SOLUTIONS GENERATED BY RHEMA-1 ALGORITHM

B	**EL**	A	B	C	D	E	F	G	K	
X	1	1	1	1	1					4
Y	9		1	1				1	1	4
Z	7			1	1	1	1			4
V	2	1		1		1			1	4
W	3,4	1	1		1	1	1	1		6
T	6,8,5	1	1	1		1		1	1	7
		4	4	5	4	4	2	3	3	**29**

The sample problem considered here is a case of 280 candidates applying for a computer based entrance exam in a local community which has a computer based test centre of capacity 50.

Each candidate is to write on 4 subjects relating to his/her discipline. It is discovered from the data submitted by the student that a total of 9 subjects denoted as {A,B,C,D,E,F,G,K} were being registered for by all the students. Candidates groups based on their chosen subjects and the no of such candidates is shown in Table 1. Due to lack of available space subjects have to be repeated which does not go well with both students parents and the organizing institution. The objective is to minimize the total number of repeated questions (or question type) in the entrance examination.

Solution to this is typical of the FGIPCPF model

The first is to determine the optimal possible number of or periods will be required using (10). This will give an optimal of 6 bins –denoted (X,Y,Z,V,W,T). the next step is to apply the RHEMA 1 method to solve the problem.

IV. RESULTS, ANALYSIS AND CONCLUSION

We implemented the RHEMA-1 algorithm and it gave a minimum value of 29 compared to 30 by BFD_f and 35 question papers generated by another procedure. The result shows among many things that it will be better to pair subjects [(1),(9),(2),(7),(3,4),(6,8,5)] into the same day or period

The Table IV also shows the no of subject to be done on each day (see last column on table) and no question types of each subject required (see last row of table) e.g. A- 4, F-3 to give a total of 29 question types.

An implementation on a larger problem will better reveal the modeling of the system and importance to different industries. This will validate the results and inferences generated. The F proposed have a lot of application especially in the flexible manufacturing systems, Group technology, assembly line problems, supply chain management and many other complex scheduling problem. A solution to the model will not solve many operation research and productivity improvement questions, but also open new area of researchers for future purpose.

Future research will examine a lot of other things related to this problem, and also implement a computer program search for a better solution which can also generate a software solution for emerging manufacturing technologies.

REFERENCES

[1]. E. G. Coffman Jr., J. Csirik, Performance Guarentees for One-Dimensional Bin Packing, Handbook of Approximation Algorithms and Metaheuristics, Taylor & Francis Group, 2007, Ch. 32.

[2]. H. Shachnai, and T. Tamir, Tight Bound for online class-constrained packing, Proceedings of Latin American theoretical information (LATIN), 321(1), 2004, pp. 103-123.

[3]. D.K. Friesen, M.A. Langston, Variable size bin packing, SIAM Journal on Computing, 15, 1986, pp. 222-230.

[4]. L. Epstein, A. Levin, An APTAS for generalized cost variable sized bin packing, SIAM Journal on Computing, 38(1), 2008, pp. 411-428.

[5]. M. A. Langston, Performance of heuristics for a computer resource allocation problem, SIAM Journal on Algebraic and Discrete Methods, 5 (2), 1984, pp. 154-161.

[6]. J. Kang, S. Park, Algorithms for the variable sized bin packing problem, European Journal of Operational Research, Vol. 147, 365-372, 2003.

[7]. X. Han, C. Peng, D. Ye, D. Zhang, Y. Lan, Dynamic bin packing with unit fractions items revisited, Information Processing Letters, Vol. 110, No. 23, 1049-1054, 2010.

[8]. H. Kellerer, U. Pferschy, D. Pisinger. Knapsack problems. Springer, 2004.

[9]. N. Menakerman and R. Rom. Bin Packing with Item Fragmentation. Lecture Notes In Computer Science, Proceedings of the 7th International Workshop on Algorithms and Data Structures WADS, 2125, 2001, pp. 313-324.

[10]. C. A. Mandal, P. P. Chakrabarti, and S. Ghose, Complexity of fragmentable object bin packing and an application, Computers & Mathematics with Applications, Elsevier, 35(1), 1998, pp. 91-97.

[11]. W. Xing, A bin packing problem with over-sized items. Operations Research Letters, 30, 2002, pp. 83-88.

[12]. G. Zhang, A new version of on-line variable-sized bin packing, Discrete Applied Mathematics, 72(33), 1997, pp. 193-197.

[13]. D. Ye, and G. Zhang, On-line extensible bin packing with unequal bin sizes, Proceedings of 2nd Workshop on Approximation and Online Algorithms (WAOA), LNCS, Springer, Berlin, 2909, 2004, pp. 235-247.

[14]. A. S. Fukunaga, R. E. Korf. Bin Completion Algorithms for Multicontainer Packing, Knapsack, and Covering Problems. Journal of Artificial Intelligence Research 28 (2007) 393-429.

[15]. M. Dell'Amico, J. C. Díaz Díaz, M.Iori. The Bin Packing Problem with Precedence Constraints. Operations Research Vol. 60, No. 6, November–December 2012, pp. 1491–1504.

[16]. P. Shaw. A Constraint for Bin-Packing. In Proceedings of the Tenth InternationalConference on Principles and Practice of Constraint Programming, volume 3258,Toronto, Canada, October 2004, pp 648–662.

[17]. M.Casazza, A.Ceselli. Mathematical programming algorithms for binpacking problems with item fragmentation. Computers & Operations Research, 46(2014)1–11.

[18]. P. C. Gilmore, R. E. A linear programming approach to the cutting-stock problem. Oper Res 1961:9:849-59.

[19]. S. Martello, P. Toth. Knapsack problems: algorithms and computer implementations. New York: John Wiley & Sons Inc.; 1990.

[20]. J. De Carvalho. LP models for bin packing and cutting stock problems. Eur J Oper Res 2002; 141(2):253-73.

[21]. B.LeCun, T.Mautor, F Quessette, M.Weisser.Bin Packing with Fragmentable Items: Presentation and Approximations. 15 pages. 2013. <hal-00780434>

[22]. U. Eliiyi, D. T. Eliiyi .Applications of Bin Packing Models through The Supply Chain. International Journal of Business and Management. Vol 1, No 1, 2009 ISSN: 1309-8047 (Online)

[23]. The Holy Bible John14:16, 1 Corinthians 9:20 ltd, King James Version, Standard Centre Column Reference Edition, Evangel Publishers ISBN:978-1-58712-148-7, 2003

[24]. J. Peng, and B. Zhang, Bin packing problem with uncertain volume and capacity," untitled (online), http://www.orsc.edu.cn/online/120601.pdf accessed 23/09/14.

[25]. R.. Romero and L. Burtseva, Bin packing implementation for material election in read switches production planning, Proceedings of world Congress on Engineering and Computer Science (WCECS), Vol II San Francisco, USA, 2009, ISBN:978-988-18210-2—7.(online).

BIOGRAPHY

Aderemi Adeyemo is an M.Sc student of the Department of Industrial and Production engineering, Faculty of Technology, University of Ibadan, where he also serves as a Tutorial assistant being a University Scholar. He holds a first class Honours degree in Industrial and production engineering. His research interests include applied optimization, logistics management, enterprise model development and application Integer linear Programming in Biomedical engineering.

Victor. O. Oladokun is currently a full time senior lecturer in the Department of of Industrial and production engineering, Faculty of Technology, University of Ibadan. He graduated with a B.Sc Honours in Mechanical Engineering from the Obafemi Awolowo University (OAU), Ile-Ife. He has an M.Sc. and Ph.D from the University of Ibadan. He is a member, Nigerian society of Engineers; and a member Nigerian Institution of engineering Management. His research interests include enterprise model development, Risk analysis, soft computing and Production system optimization in developing economies.