ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Procedia - Social and Behavioral Sciences 111 (2014) 663 – 671
Procedia
Social and Behavioral Sciences

EWGT2013 – 16th Meeting of the EURO Working Group on Transportation

Asymptotic results for the Generalized Bin Packing Problem

Mauro Maria Baldi[a,*], Teodor Gabriel Crainic[b], Guido Perboli[a,b], Roberto Tadei[a]

[a]*Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy*
[b]*CIRRELT, 2920, Chemin de la Tour, Montreal, QC, H3T 1J4, Canada*

Abstract

We present a worst case analysis for the Generalized Bin Packing Problem, a novel packing problem arising in many Transportation and Logistics settings, characterized by multiple item and bin attributes and by the joint presence of both compulsory and non-compulsory items. The contribution of this paper is twofold: we conduct a worst case analysis applied to the much richer Generalized Bin Packing Problem of two outstanding bin packing algorithms (the First Fit Decreasing and the Best Fit Decreasing algorithms) arising in Transportation and Logistics, and we propose two semi-online algorithms also arising in the fields of Transportation and Logistics. We also show how knowing part of the instance or the whole instance is not enough for computing worst case ratio bounds.

Keywords: Packing problems; Generalized Bin Packing Problem; Worst case analysis.

1. Introduction

Packing problems in Transportation and Logistics are often more complex than the ones traditionally present in the literature. In particular, it is quite normal to face situations where the nature of the problem cannot be reduced to a single packing problem. For this reason, in recent years the research community started to think to new multi-attribute and multidimensional extensions of packing problems, as already done in the VRP field (Crainic et al., 2008). One of the latest attempts in the direction of the generalization of packing problems is the so called Generalized Bin Packing Problem (GBPP). Given a set of containers, different in cost and volume, and a set of items, characterized by volume, profit and the compulsory attribute, i.e. the attribute stating if the model is obliged to load an item or whether it can decide according to an economic criterion, the GBPP aims to find the subset of bins and the subset of non-compulsory items such that all the items (all the compulsory and the chosen

* Corresponding author. Tel.: +39-011-090-7083; fax: +39-011-090-7099.
E-mail address: mauro.baldi@polito.it

doi:10.1016/j.sbspro.2014.01.100

non-compulsory ones) are accommodated into the bins and the overall cost, given by the cost for using the bins minus the profit associated to the items is minimized.

As shown in Baldi et al. (2012a), the GBPP is able to describe new operational settings in Transportation and Logistics characterized by the joint optimization of company revenues and transportation costs and by the presence of different 3PL or container types. Moreover, in multi-modal and cross-continental transportation, freight is not shipped directly from origins to destinations but calls at intermediate platforms named transshipment facilities. At these facilities freight consolidation and handling operations are performed and, usually, freight is moved to another transportation vector. However, a portion of the freight might wait to proceed its journey to destination, depending on the overall trade-off between freight profits and shipping costs. The GBPP is the first packing problem able to model this setting. Additionally, the GBPP copes with the majority of the traditional packing problems, ranging from Knapsack to Bin Packing and to different variants of Multiple Knapsack and Cutting Stock problems.

Baldi et al. (2012a, b) proposed both bounds and exact and approximate methodologies in order to address this problem. Moreover, a preliminary worst case analysis has recently been proposed (Baldi et al., 2013). The GBPP is rooted on earlier bin packing problems which are the eldest Bin Packing Problem (BPP), the Variable Sized Bin Packing Problem (VSBPP), and the most recent Variable Cost and Size Bin Packing Problem (VCSBPP). We briefly recall here these problems.

The BPP was first investigated by Ullman (1971) and Garey et al. (1972). Johnson (1973) proposed the Next Fit (NF) algorithm and proved that its performance ratio is 2. Johnson et al. (1974) showed that the First Fit (FF) and the Best Fit (BF) algorithms have both performance ratios of 17/10. Moreover, they computed a worst case ratio bound for the First Fit Decreasing (FFD) and the Best Fit Decreasing (BFD) algorithms equal to 11/9. de la Vega & Lueker (1981) presented a polynomial time approximation scheme, Seiden (2002) studied the online variant, and Crainic et al. (2007a, b) introduced fast lower bounds and conducted an asymptotic worst case analysis on BPP lower bounds. Li & Chen (2006) studied the variant where all bins have the same capacity but are characterized by a non-decreasing concave cost function of the bin utilization. The VSBPP was introduced by Friesen & Langston (1986). The authors provided one online and two offline algorithms and proved that their worst case ratios are 2, 3/2, and 4/3, respectively. Murgolo (1987) presented an approximation scheme and Kang & Park (2003) provided two offline algorithms and showed that their asymptotic worst case ratio is equal to 3/2. Crainic et al. (2011) proposed accurate bounds for the VCSBPP. For this problem, Epstein & Levin (2008, 2012) provided an APTAS and an AFPTAS.

Although Baldi et al. (2013) performed a worst case analysis for the GBPP, this study is not complete because only online algorithms were considered. In this paper, we extend the work of Baldi et al. (2013) by also considering semi-online and offline settings. These two settings often arise in Transportation and Logistics where orders (i.e., the items) are not known a priori but arrive along time to a shipping company. In many circumstances, shipping companies do not immediately ship the already received freight (coming from customers and leading to final destinations) but decide to wait in order to receive more freight to dispatch with a unique shipment. The semi-online setting of the GBPP also arises in freight transportation, and in particular among freight forwarders and carriers, where freight is shipped through means of transport with cadenced departure times.

We show that, being the GBPP a richer setting due to the presence of multiple attributes and of both compulsory and non-compulsory items, it is impossible to guarantee a worst case ratio.

This paper is organized as follows. In Section 2, we present in a more formal way the problem and the nomenclature. In Section 3, we present our worst case analysis results, while conclusions and future developments are discussed in Section 4.

2. Problem setting

In the GBPP, a set of items with volume and profit has to be loaded into proper bins characterized by capacity and cost. Moreover, items can be compulsory and non-compulsory. Whilst compulsory items must be taken, non-compulsory items might not be loaded if this is beneficial. Aim of the GBPP is to minimize the overall cost, given by the difference between the costs of the selected bins and the profits of the taken non-compulsory items.

We name I the set of items and J the set of bins. We also name I^C and I^{NC} the set of compulsory and non-compulsory items respectively. Clearly, $I^C \cup I^{NC} = I$ and $I^C \cap I^{NC} = \emptyset$. Let w_i and p_i be respectively the volume and the profit of item $i \in I$ and W_j and C_j respectively the capacity and the cost of bin $j \in J$. As for the VCSBPP, bins are classified into bin types i.e., bins belonging to the same bin type have the same capacity and cost. We name T the set of bin types and W_t and C_t the capacity and the cost of bins belonging to bin type $t \in T$. Finally, the indicator function $\sigma: J \rightarrow T$ reveals the type $\sigma(j) \in T$ of bin $j \in J$.

3. Worst case results

Let Π be a minimization problem, $\imath \in \Pi$ be an instance of the problem Π, A be an algorithm, $A(\imath)$ the value of the solution yielded by the algorithm A when applied to instance $\imath \in \Pi$, and $OPT(\imath)$ be the optimum of instance $\imath \in \Pi$, then the asymptotic worst-case ratio is the smallest positive R such that for any instance $\imath$ of problem Π we have that

$$A(\imath) \leq R \cdot OPT(\imath) + O(1) \qquad (1)$$

Similarly, the absolute worst-case ratio is the smallest positive ρ such that the for any instance $\imath$ of problem Π we have that

$$A(\imath) \leq \rho \cdot OPT(\imath) \qquad (2)$$

We prove a lemma to which we will refer within the proofs of the following theorems. In particular, we show that, due to the presence of non-compulsory items, there are instances where the optimum and the value yielded by an algorithm differ by a linear term which, in principle, can be arbitrarily large. In Lemma 1, we show that it is impossible to guarantee both an absolute and an asymptotic worst case ratio in these circumstances.

Lemma 1

Given a minimization problem P and an algorithm A, let $\imath(m, n) \in P$ be an instance of the problem with m and n non-negative integers such that $A(\imath(m, n)) = \alpha m$ and $OPT(\imath(m, n)) = \beta m - \gamma n$, with $\alpha, \beta, \gamma > 0$. Then, it is impossible to compute the asymptotic and the absolute worst case ratio for algorithm A.

Proof

If there exists an asymptotic worst case ratio then, according to (1), we have to find proper R and $O(1)$ such that

$$\alpha m \leq R(\beta m - \gamma n) + O(1). \qquad (3)$$

If we consider the particular instance $\iota(0, n)$, then (3) becomes

$$0 \leq -R\gamma n + O(1). \quad (4)$$

Since n can be arbitrarily large and $\gamma > 0$, then, independently of the constant $O(1)$, it must be $R \leq 0$. Vice versa, considering instance $\iota(m, 0)$, (3) becomes

$$\alpha m \leq R\beta m + O(1). \quad (5)$$

Since m can be arbitrarily large and $\beta > 0$, then, independently of the constant $O(1)$, it must be $R \geq \alpha/\beta$. Since, by hypothesis, both α and β are positive, then their ratio is a positive number. Hence, requiring $R \geq \alpha/\beta$ contradicts the previous requirement that $R \leq 0$. Therefore, it is impossible to compute the asymptotic worst case ratio. Since such a result holds independently of constant $O(1)$ and according to (2) the absolute worst case ratio is the particular case when $O(1) = 0$, then it is also impossible to compute the absolute worst case ratio.

□

Lemma 1 will be used in Theorem 1 to prove that no worst case ratio can be computed for the offline FFD and BFD algorithms. These are well known heuristics, initially conceived for the BPP (Garey, 1972) and recently adapted to the GBPP by Baldi et al. (2012a). To make this article self contained, we briefly recall here the FFD and the BFD heuristics adapted by Baldi et al. (2012a) to the GBPP. Their pseudo-code is reported in Table 1. Both heuristics work with three lists: the *Sorted Items List* (*SIL*), the *Sorted Bins List* (*SBL*), and the list of selected bins (*S*). The *SIL* and the *SBL* lists are built according to a given sorting criterion. Baldi et al. (2012a) defined four sorting criteria. For each of them, the compulsory items are sorted at the top of the *SIL* list by non-increasing volume. The four strategies are:

- **Sorting 1.** Bins: Non-decreasing $C_{\sigma(j)}/ W_{\sigma(j)}$ and non-decreasing volumes $W_{\sigma(j)}$; Non-compulsory items: Non-increasing p_i/w_i and non-increasing volumes w_i

- **Sorting 2.** Bins: Non-decreasing $C_{\sigma(j)}/ W_{\sigma(j)}$ and non-decreasing volumes $W_{\sigma(j)}$; Non-compulsory items: Non-increasing volumes w_i and non-increasing p_i/w_i

  B
- **Sorting 3.** Bins: Non-decreasing $C_{\sigma(j)}/ W_{\sigma(j)}$ and non-increasing volumes $W_{\sigma(j)}$; Non-compulsory items: Non-increasing p_i/w_i and non-increasing volumes w_i

- **Sorting 4.** Bins: Non-decreasing $C_{\sigma(j)}/ W_{\sigma(j)}$ and non-increasing volumes $W_{\sigma(j)}$; Non-compulsory items: Non-increasing volumes w_i and non-increasing p_i/w_i.

Finally, *S* is the list of the selected bins making up the solution yielded by the heuristics. Initially, all the items in *SIL* are unpacked and the list must be scanned. For each item *i* encountered when scanning *SIL*, both heuristics try to accommodate *i* into the already selected bins in *S*. The differences between the two heuristics is how such a bin, if exists, is determined. The FFD heuristics assigns item *i* to the *first* bin (if exists) among those in *S* able to accommodate it. The BFD heuristics assigns item *i* to the *best* bin (if exists) among those in *S* able to accommodate it. The best bin is the one with the least residual space after placing item *i*. If item *i* cannot be placed among the already selected bins in *S* (and this happens when none of them has enough residual space to accommodate the item), then a new bin *b* must be selected from the list of available bins *SBL* and added to *S*. If *i* is a compulsory item, then *b* is the first bin in *SBL\S* able to contain it. Once bin *b* has been selected from list *SBL* and added to list *S*, the compulsory item *i* is loaded into bin *b*. If item *i* is a non-compulsory item, then it is loaded into a new bin only if it is profitable, that is, if there exists, among the bins in *SBL\S*, a bin *b* such that the profit of item i plus the profits of the succeeding non-compulsory items in *SIL* able to be accommodated into bin b together with item i is greater than the cost of bin b. The search of bin b is performed by the *PROFITABLE* function, which pseudo-code is reported in Table 2. If there exists a profitable bin b for the non-compulsory item i, then bin b is added to the list of the selected bins S and item i is loaded into bin b. After the whole *SIL* list has been scanned, the *POST-OPTIMIZATION* function (which pseudo-code is reported in Table 3) checks whether, for any selected bin $j \in S$ making up the final solution, there exists, among the non-selected bins in $SBL \setminus S$, a more convenient bin k. This happens when bin k can accommodate all the items which have been loaded into bin j and its cost is less than the cost of bin j. If such a swap is possible, then items into bin j are moved to bin k, with a loss (recall that we are dealing with a minimization problem) of $C_{\sigma(j)}$ - $C_{\sigma(k)}$.

Table 1. Main procedure of FFD and BFD heuristics

$S := \varnothing$
for all $i \in SIL$ **do**
Identify the bin $b \in S$ into which item i can be loaded
• FFD: the first bin with enough empty volume to accommodate item i
• BFD: the bin with the minimum free volume after loading item i
if b exists **then**
Load item i into bin b
else
if $i \in I^C$ **then**
Identify the first bin $b \in SBL \setminus S$ such that $w_i \leq W_{\sigma(b)}$
Load item i into bin b
$S := S \cup \{b\}$
else
Identify the bin $b \in SBL \setminus S$ such that *PROFITABLE(i, b)* returns *TRUE*
if b exists **then**
Load item i into bin b
$S := S \cup \{b\}$
else
reject item i
POST-OPTIMIZATION

Table 2. *PROFITABLE(i, b)*

SIL_i: sublist of SIL starting from the item i
Load item i into bin b and initialize the bin profit $P_b := p_i$
for all $l \in SIL_i$ **do**
if item l can be loaded into bin b **then**
Load item l into bin b and update the bin profit $P_b := P_b + p_l$
if $P_b > C_{\sigma(b)}$ **then**

```
    return TRUE
else
    return FALSE
```

Table 3. *POST-OPTIMIZATION*

for all $j \in S$ **do**
  for all $k \in SBL \backslash S$ **do**
    $U_j := \sum_{i \text{ loaded into } j} w_i$
    if $W_{\sigma(k)} \geq U_j$ and $C_{\sigma(k)} < C_{\sigma(j)}$ **then**
      Move all the items from j to k
      $S := S \backslash \{j\} \cup \{k\}$

In Theorem 1, we prove a very strong result: it is impossible to guarantee asymptotic and absolute worst case ratios for the FFD and BFD heuristics when applied to the GBPP.

Theorem 1

It is impossible to compute the asymptotic and the absolute worst case ratios for the FFD and BFD heuristics when applied to the GBPP.

Proof

Consider instance $\imath(m, n)$ consisting of two bin types ($T = \{1, 2\}$) with $W_1 = W$, $C_1 = C$, $W_2 = 72/75\ W$, $C_2 = 49/50\ C$, and the set of items I is split into four subsets, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$, with $|\mathcal{A}| = m$, $|\mathcal{B}| = 4n$, $|\mathcal{C}| = 6n$, and $|\mathcal{D}| = 2n$. An item which belongs to subset $X \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$ is called a type X item. Let type $\mathcal{A}$ items be compulsory with $w_{\mathcal{A}} = 72/75\ W$, type $\mathcal{B}$ items be non-compulsory with $w_{\mathcal{B}} = 18/75\ W$, $p_{\mathcal{B}} = 22/100\ C$, and type $\mathcal{C}$ items be non-compulsory with $w_{\mathcal{C}} = 25/75\ W$, $p_{\mathcal{C}} = 32/100\ C$, and let type $\mathcal{D}$ items be non-compulsory with $w_{\mathcal{D}} = 39/75\ W$ and $p_{\mathcal{D}} = 57/100\ C$. It is easy to verify that the optimal solution consists of m type 2 bins each containing one type $\mathcal{A}$ compulsory item, and $2n$ type 1 bins each containing two type $\mathcal{B}$ non-compulsory items and one type $\mathcal{D}$ non-compulsory item:

$$OPT(\imath(m, n)) = mC_2 + 2n(C_1 - 2p_{\mathcal{B}} - p_{\mathcal{D}}) = 49/50\ mC - 1/50\ nC \qquad (6)$$

Applying any among the four sorting rules listed before, FFD and BFD use type 1 bins and pack first all the type $\mathcal{A}$ compulsory items, then all the type $\mathcal{D}$ non-compulsory items, all the type $\mathcal{C}$ non-compulsory items, and finally all the type $\mathcal{B}$ non-compulsory items. After packing all the type $\mathcal{A}$ compulsory items, each into type 1 bin, type $\mathcal{D}$ items must be accommodated. Since only one type $\mathcal{D}$ item can be accommodated into one type 1 bin, the *PROFITABLE* procedure scans the succeeding items: those of type $\mathcal{C}$. Only one type $\mathcal{C}$ item can be accommodated with one type $\mathcal{D}$ item into one type 1 bin, say b. The level of bin b is $w_{\mathcal{C}} + w_{\mathcal{D}} = 64/75\ W$ and its residual space is not enough to load any type $\mathcal{B}$ item. The overall profit of bin b is $P_b = p_{\mathcal{C}} + p_{\mathcal{D}} = 89/100\ C < C_1$. Therefore, all type $\mathcal{D}$ items will be rejected from algorithms FFD and BFD. When scanning type $\mathcal{C}$ items, at most three of them can be accommodated into one type 1 bin because $3w_{\mathcal{C}} = W = W_1$ but $3p_{\mathcal{C}} = 96/100\ C < C_1$; therefore the *PROFITABLE* procedure rejects all the type $\mathcal{C}$ items in the list *SIL* with two type $\mathcal{C}$ succeeding items. In fact, this is not the case for the next to last and for the last type $\mathcal{C}$ items in the list *SIL*. More precisely, when the next to last type $\mathcal{C}$ compulsory item must be accommodated, the *PROFITABLE* procedure computes the overall profit P_b taking into account that there are two more type $\mathcal{C}$ items and then type $\mathcal{B}$ items follow in the list. Loading two type $\mathcal{C}$ items into one type *1* bin, there is room only for one more type $\mathcal{B}$ item (because $2w_{\mathcal{C}} + w_{\mathcal{B}} = 68/75\ W < W_1$, but $2w_{\mathcal{C}} + 2\ w_{\mathcal{B}} = 86/75\ W > W_1$). The overall profit P_b is then $2p_{\mathcal{C}} + p_{\mathcal{B}} = 86/100\ C < C_1$; therefore even the next to last type $\mathcal{C}$ bin will be discarded. Even the last type $\mathcal{C}$ item will be discarded by the *PROFITABLE* procedure because it can be loaded with at most two type $\mathcal{B}$ items, but $p_{\mathcal{C}} + 2p_{\mathcal{B}} = 76/100\ C < C_1$. Finally, all the type $\mathcal{B}$ items will be discarded because at most *4* of them can be loaded into one type *1* bin but $4p_{\mathcal{B}} = 88/100\ C < C_1$. Therefore,

$$FFD(\imath(m, n)) = BFD(\imath(m, n)) = mC_1 = mC \tag{7}$$

The theorem holds applying Lemma 1 with $\alpha = C$, $\beta = 49/50\ C$, and $\gamma = C/50$.

□

We propose two semi-online settings where items arriving online are not directly packed into bins but are placed into a buffer. In the first setting, the buffer has a capacity of $k > 1$ items. When the buffer is full, the items in the buffer are loaded into the already open bins or, if necessary, into new bins. For this setting, we propose two semi-online algorithms named First Fit with Buffer and Rejections (FFBR) and Best Fit with Buffer and Rejections (BFBR). In the FFBR, each item is accommodated according to its arrival order and to a First Fit policy, i.e., it is loaded (if possible) into the first open bin able to contain it, otherwise a new bin is selected from a list of bins sorted according to a given criterion. In the BFBR, each item is accommodated according to its arrival order and to a Best Fit policy, i.e., it is loaded (if possible) into the best open bin able to contain it, otherwise a new bin is selected from a list of bins sorted according to a given criterion. As for the aforementioned BFD algorithm, the best bin is that bin with the least residual space after placing an item. At the end of the process, bins with an overall profit less than their cost will be rejected. In Theorem 2, we prove that even retaining some items into a buffer is not enough to guarantee an asymptotic and an absolute worst case ratio. In the second setting, the buffer has a time capacity of τ. When a time interval of length τ has elapsed, the items arrived within this interval are loaded into the already open bins or, if necessary, into new bins. At the end of the process, bins with an overall profit less than their cost will be rejected. We name First Fit with Time Buffer and Rejections (FFTBR) and Best Fit with Time Buffer and Rejections (BFTBR) the natural extension of algorithms FFBR and BFBR to this setting.

In Theorem 3, we prove that even for algorithms FFTBR and BFTBR it is impossible to guarantee asymptotic and absolute worst case ratios.

Theorem 2

It is impossible to compute the asymptotic and absolute worst case ratio for algorithms FFBR and BFBR when applied to the GBPP.

Proof

Given the size $k > 1$ of the buffer, consider instance $\imath^k(m, n)$ consisting of one bin type with capacity W and cost C, and where the set of items I is split into three subsets, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, with $|\mathcal{A}| = 2mk$, $|\mathcal{B}| = 2nk$, and $|\mathcal{C}| = 2nk(k - 1)$. Let type $\mathcal{A}$ items be compulsory with $w_{\mathcal{A}} = W$, let type $\mathcal{B}$ items be non-compulsory with $w_{\mathcal{B}} = W/k$, $p_{\mathcal{B}} = C/k + \varepsilon$, and let type $\mathcal{C}$ items be non-compulsory with $w_{\mathcal{C}} = W/k$, and $p_{\mathcal{C}} = C/k - 2\varepsilon/(k - 1)$, with $\varepsilon > 0$ small enough, and $C > 2k\varepsilon$. Note that at most k type $\mathcal{B}$ items can be accommodated into one bin. Also, at most k type $\mathcal{C}$ items can be accommodated into one bin. We have that $kp_{\mathcal{B}} = C + k\varepsilon > C$ and $kp_{\mathcal{C}} = C - 2k\varepsilon/(k - 1) < C$. Consequently, the optimum consists of $2mk$ bins each containing one type $\mathcal{A}$ compulsory item and $2n$ bins each containing k type $\mathcal{B}$ non-compulsory items:

$$OPT(\imath^k(m, n)) = 2mkC + 2n(C - kp_B) = 2mkC - 2nk\varepsilon \tag{8}$$

Consider the following sequence of items of instance $\imath^k(m, n)$ for algorithm FFBR:

$$\overbrace{\underbrace{i_{\mathcal{A}} \; \dots \; i_{\mathcal{A}}}_{k \text{ times}} \; \dots \; \underbrace{i_{\mathcal{A}} \; \dots \; i_{\mathcal{A}}}_{k \text{ times}}}^{2m \text{ times}} \quad \overbrace{i_{\mathcal{B}} \; \underbrace{i_{\mathcal{C}} \; \dots \; i_{\mathcal{C}}}_{k-1 \text{ times}} \; \dots \; i_{\mathcal{B}} \; \underbrace{i_{\mathcal{C}} \; \dots \; i_{\mathcal{C}}}_{k-1 \text{ times}}}^{2nk \text{ times}} \tag{9}$$

Then, applying FFBR or BFBR to sequence (9), we have $2mk$ bins each containing one type $\mathcal{A}$ compulsory item and $2nk$ bins each containing one type $\mathcal{B}$ non-compulsory item and $k - 1$ type $\mathcal{C}$ non-compulsory items. However, bins containing non-compulsory items will be rejected because $p_{\mathcal{B}} + (k - 1)p_{\mathcal{C}} = C - \varepsilon < C = C_I$. Therefore we have:

$$FFBR(\imath^k(m, n)) = BFBR(\imath^k(m, n)) = 2mkC \tag{10}$$

and the theorem holds applying Lemma 1 with $\alpha = 2kC$, $\beta = 2kC$, and $\gamma = 2k$

□

In Theorem 3, we prove that the same conclusion holds even for algorithm FFTBR and BFTBR.

Theorem 3

It is impossible to compute the asymptotic and absolute worst case ratio for algorithms FFTBR and BFTBR when applied to the GBPP.

Proof

The theorem trivially holds considering instance $\imath^k(m, n)$ in Theorem 2 where k items fall within each time-slot of length τ.

□

4. Conclusion

In this paper we presented a worst case analysis for the Generalized Bin Packing Problem (GBPP), a fundamental packing problem arising in many Transportation and Logistics settings due to the joint presence of multiple bin and item attributes and to the tradeoff between item profits and bin costs. Our main contribution was the development of a worst case analysis of two popular algorithms among bin packing problems (the First Fit Decreasing and the Best Fit Decreasing algorithms) extended to the much richer GBPP. In fact, worst case analyses of these outstanding algorithms were conducted for the classical Bin Packing Problem and for many variants of it, but not yet for the more general GBPP.

Moreover, we also proposed two semi-online algorithms arising in freight transportation, in particular among carriers and shipping companies. Our study revealed that, in spite of a much richer problem as the GBPP (thanks to which it is possible to describe many Transportation and Logistics settings), the counterpart becomes that it is not possible to guarantee worst case ratio bounds.

Nevertheless, this study is the starting point for further research works, in order to find finite worst-case ratio bounds algorithms for the GBPP, if they exist.

Acknowledgements

This project has been partially funded by the Natural Sciences and Engineering Council of Canada (NSERC), the Fonds de recherche du Québec - Nature et technologies (FRQNT), and the Italian Ministry of Education, University, and Research, under the PRIN 2009 project "Methods and Algorithms for the Logistics Optimization".

References

Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2013). *Worst-case analysis for new online bin packing problems* (Technical Report CIRRELT-2013-11). CIRRELT, Montreal, Canada.

Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2012a). The generalized bin packing problem. *Transportation Research Part E*, 48, 1205 - 1220.

Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2012b). Branch-and-price and beam search algorithms for the variable cost and size bin packing problem with optional items. *Annals of Operations Research*, DOI 10.1007/s10479-012-1283-2.

Crainic, T. G., Perboli, G., Pezzuto, M., & Tadei, R. (2007a). Computing the asymptotic worst-case of bin packing lower bounds. *European Journal of Operational Research*, 183, 1295 - 1303.

Crainic, T. G., Perboli, G., Pezzuto, M., & Tadei, R. (2007b). New bin packing fast lower bounds. *Computers & Operations Research*, 34, 3439 - 3457.

Crainic T. G., Perboli G., & Tadei. R. (2008). Extreme-Point-based Heuristics for the Three-Dimensional Bin Packing problem. *Informs Journal On Computing*, 20, 368 - 384.

Crainic, T. G., Perboli, G., Rei, W., & Tadei, R. (2011). Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers & Operations Research*, 38, 1474 - 1482.

de la Vega, W. F. & Lueker, G. S. (1981). Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1, 349-355.

Epstein, L. & Levin, A. (2008). An APTAS for Generalized Cost Variable-Sized Bin Packing. *SIAM Journal on Computing*, 38, 411-428.

Epstein, L. & Levin, A. (2012). Bin packing with general cost structures. *Mathematical Programming*, 132, 355 – 391.

Friesen, D. K., & Langston, M. A. (1986). Variable sized bin packing, *SIAM Journal on Computing*, 15, 222 - 230.

Garey, M. R., Graham, R. L., & Ullman, J. D. (1972). Worst-case analysis of memory allocation algorithms. In *Proceedings of the fourth annual ACM symposium on Theory of computing* (pp. 143.150). New York, NY, USA.

Johnson, D. S. (1973). *Near-Optimal bin packing algorithms* (PhD thesis). Dept. of Mathematics, M.I.T., Cambridge, MA.

Johnson, D. S., Demeters, A., Hullman, J. D., Garey M. R., & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3, 299 - 325.

Kang, J., & Park, S. (2003). Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147, 365 - 372.

Li, C. L., & Chen, Z. L. (2006). Bin-packing problem with concave costs of bin utilization. *Naval Research Logistics*, 53, 298 - 308.

Murgolo, F. D. (1987). An efficient approximation scheme for variable-sized bin packing. *SIAM - Journal on Computing*, 16, 149-161.

Seiden, S. (2002). On the online bin packing problem. *Journal of the ACM*, 49, 640 - 671.

Ullman, J. D. (1971). *The performance of a memory allocation algorithm* (Tech. Rep.). Princeton University.