bin-packingの亜種ふたつにかんする備忘録

飯田浩志*

この小品は，bin-packing問題の亜種（拡張もしくは変形）となるふたつの問題，先行制約つきbin-packingとbin-stretchingにかんする覚書になる．

キーワード：組合せ最適化，bin-packing問題

本稿は，bin-packing（ビン詰め）問題の亜種となるふたつの問題；先行制約つきbin-packingとbin-stretchingにかんするメモ書きである．

初版が出てのち改訂されることのない専門書が多い中，2012年初頭にKorteとVygenの手になる成書[8]は，第5版が刊行された．その中の第18章で取り上げられてゐるbin-packing問題は，各$w_j \in (0, 1]$なる実数列$w_1, w_2, \ldots, w_n$が与へられ（weightのw），それらを容量1のビン（bin）に凡(すべ)て詰めるにあたり，使用するビンの本数を最小にする詰め方を求めるといふ，有名な巡回セールスマン問題と同じく強*NP*困難な組合せ最適化問題である．式で書くと，項（item）jをビン$b(j)$にassignする函数を$b : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\}$として，勝手なビン$i$ $(1 \leq i \leq k)$について$\sum_{j : b(j)=i} w_j \leq 1$なる制約下で$k$を最小にする$b(\cdot)$を求めるのがbin-packingである．最小なったビン数kを最適値と呼ぶ．

このbin-packingに対してはさまざまな近似算法[*1]が提案されてゐる．近似算法とは，一般に最適な詰め方ではないがゆゑに最適値より大きい数（近

* E-mail: hiroggiida@me.com

[*1] 本稿では，アルゴリズムの意味で算法といふ語を用ゐる．

似値）を返してしまふけれど，最適値を求めるよりも格段に早く終る手法を指す．

ここに少し言葉の定義をしておき度い．bin-packing のやうな最小化問題，i.e. 何ごとか（bin-packing なら使用するビンの本数）を最小にすることを目的とする問題への近似算法 A に於いて，どんな問題例（instance）I についても，$A(I)$ と $OPT(I)$ が I に A を施して得られる値と最適値（>0 とする）をそれぞれ表すとして $A(I) \leq kOPT(I)$ が成り立つとき，$k(\geq 1)$ を A の絶対性能比と呼ぶことにする（$A(I) \geq OPT(I)$ に注意すると，絶対性能比 1 は，近似値ではなく最適値を返す算法を暗に示してゐる）．またこれに，定数 c を加えた $A(I) \leq kOPT(I)+c$ が成立するとき，k を A の漸近的性能比と呼ぶことにする．以降では，あいまいさを生じるおそれがないのであれば $A(I)$ や $OPT(I)$ の I を略すことがある．例へば前述したやうに，最小化問題への近似算法 A は常に最適値以上の値を返すわけだけれども，これを $A \geq OPT$ と書くことがある．

bin-packing への代表的な近似算法である FF と FFD に就いて，簡単に説明しておく．FF（First-Fit）は，w_1 から w_n の順に，最初のビンから次次に見ていって最初に（first）詰められる（fit する）ビンに詰め，入るビンが無ければ新たなビンを用意して詰める，といふ算法である（w_1 は必ず最初のビンに這入る）．FFD（First-Fit Decreasing）は，w_j を大きい順に整列 $w_1 \geq w_2 \geq \cdots \geq w_n$ してから FF を実行する．たとへば問題例 $\{0.5, 0.4, 0.3, 0.3, 0.3, 0.2\}$ では $OPT=2$，最初のビンに 0.5 と 0.4 が入ってしまって $FFD=3$ になる．じつはこの例は，FFD の絶対性能比が 1.5 より良く（小さく）ならないことを示してゐる．具体的には，$FFD \leq (3/2)OPT$ の等号が成り立つ例になってゐる．かうした例を，tight な例といふ[*2]．

[*2] bin-packing で $P \neq NP$ の下，いかな近似算法を以てしても絶対性能比 1.5 未満は無理[8, p. 472]．この，あらゆる近似算法を考慮した上での絶対性能比の下限（infimum）1.5 を指して，bin-packing の approximation ratio は 1.5 と云ふ[8, p. 433]．

2007 年に Dósa [3] は，1990 年から暫く動かなかった，FFD の漸近的性能比を表す式 $\mathrm{FFD} \leq (11/9)\mathrm{OPT}+1$ 中の定数項 1 が，じつは 6/9 で tight になることを示した．tight な例は，$\{1/2+\epsilon, 1/4+\epsilon, 1/4-2\epsilon\}\times 4+\{1/4+2\epsilon, 1/4+2\epsilon, 1/4-2\epsilon, 1/4-2\epsilon\}\times 2$ である．これが $\mathrm{OPT}=6$，$\mathrm{FFD}=8$ $(=(11/9)\mathrm{OPT}+6/9)$ を与へる．ここに $\epsilon>0$ は，非常に小さい実数を表す．

他方 2010 年に Xia と Tan [11] は，Simchi-Levi [10] による FF の絶対性能比 7/4 を 12/7 に改良した [8, p. 475]．この 12/7 の証明には，漸近的性能比を表す式 $\mathrm{FF} \leq (17/10)\mathrm{OPT}+7/10$ が使われてゐる[*3]．こちらも，Xia と Tan [11] が導いた結果である（だから，文献 [11] のタイトルが bounds, と複数形になってゐる）．定数項 7/10 は，1976 年の 9/10 以来じつに 34 年ぶりの改良とのこと [11, p. 1669]．ただし tight とは示されておらず，FFD の漸近的性能比を表す式中の定数項は 2/3 で終結した一方，FF のそれ 7/10 は未決である．さらに FF にかんしては，絶対性能比についての議論も終ってない．Xia と Tan [11] の推測どおり 1.7 であると証明されるか，はたまた $\mathrm{OPT}=7$, $\mathrm{FF}=12$ を与へる問題例が見つかるか．云ふまでもなく 1.7 と 12/7 の間で決着する，といふ可能性もある．

以上のやうな bin-packing それ自体への研究も然る事ながら bin-packing の亜種（拡張あるいは変形）となる問題も提案されてゐる．たとへば，各ビンに就いて一ヶだけならビンをはみ出しても可とか，online（後述）で各ビンの中が大きい順になるやうに詰める問題などがある（それぞれ，Ceselli と Righini [1]，Dósa 他 [4] 等を参照され度い）．以下二節では，最近たまたま目にした先行制約つき bin-packing と bin-stretching それぞれにかんする文献についてのメモ書きを残しておくことにする．

[*3] 例へば Li と Chen [9, p. 300] でも触れられてゐるやうに，FF の漸近的性能比は 1.7 より良くならないことが知られてゐる．

先行制約つき bin-packing に提案された下界に就いて

Dell'Amico 他[2]は，先行制約つき bin-packing を exact に解く手法を提案するうえで，最適値の下界（見積もり）をいくつか考案してゐる．なかでも L_3 と L_4 に就いては，最悪の場合の性能比がそれぞれ 1/3, 1/2 であることを示してゐる．本節では其れら下界ふたつに就いて少し書きとめておき度い．

まづ，本節で扱ふのは使用するビン数の最小化問題であって，必要最低限のビン数を示す最適値より大きくない値が下界であり，その見積もりを表してゐる(0−1 ナップサック問題のやうな最大化問題なら上界にあたる)．精度の良い下界は，分枝限定法を適用するにあたって展開するノード数を減らすために重要な役割を果たす．ただ，下界の精度が良いに越したことはないけれど，一般に下界を得るまでにかかる時間と精度とのあいだには tradeoff の関係がある．拙稿[5]で示したやうに，精度が多少悪い見積もりであっても，それが素早く求まるのであれば，全体として見れば実用になることもある．

bin-packing は，いずれも非負重量[*4] $w_1, w_2, \ldots, w_n$ をもつ n ヶの項を容量 C のビン（通常は $C=1$）に凡て詰めるにあたり，使ふビンの本数を最小にする問題であった．さらに先行制約のある bin-packing は，特定のふたつの項を詰めるにあたって，先行する方 (predecessor) を strict に番号の小さいビンに詰めなければならない(同じビンでは不可)．先行制約は，項を節点とする，サイクルをもたない有向グラフで表すことができる．図 1 に例を示す．0−1 ナップサック問題に先行制約があると，例へば図 1 なら，項 j を詰めるにあたって項 i がナップサックに入っていなくてはならない；つまり，項 i 抜きで項 j を詰めることはできない[7, 13.2 節]．bin-packing での図 1 は，先述のとおり項 i を詰めるビンの番号が項 j を詰めるビンのそれよりも確か

[*4] 重量 0 の項は，先行制約がある場合には意味を持たせることができる．例へば，ふたつの項がそれぞれ格納されるビンの間に少なくともひとつビンを挟みたい（隣にはしたくない）といふ向きには，先行制約を表すグラフに於いて当該二項を結ぶ枝の中間に重量 0 のダミー項を設けるとよい．

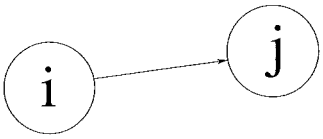

図1 項 i は項 j に先行するという制約

に小さくあること，i.e. $b(i) < b(j)$ を要請する．

Dell'Amico 他[2]は，先行制約つき bin-packing への最適解を求める手法を提案した．その手法は分枝限定法に基づくが，その分枝限定法を実装するにあたって幾つかの下界を考案してゐる．こと L_3 と L_4 に就いては，最悪の場合の性能比がそれぞれ 1/3, 1/2 であることを，しかも tight であることをも示してゐる(tight であることから，求められた L_3, L_4 それぞれの性能比の導き方に改善の余地なし)．このことについて以下，すこしメモっておく．

以降 Dell'Amico 他 [2] に倣(なら)い，とある問題のいかなる例 I についても，その最適値 $\mathrm{OPT}(I)$ と下界 $L(I)(\leq \mathrm{OPT}(I))$ について $\epsilon \leq L(I)/\mathrm{OPT}(I)$ が成り立つとき，この $\epsilon(\leq 1)$ を最悪の場合の性能比，または単に性能比と呼ぶことにする．たとへば性能比 $\epsilon = 1/2$ の下界なら，最適値の半分より悪い(小さい) 見積もりは出さない，と云ふこと．これに加へて $\epsilon' > \epsilon$ のとき，$\epsilon' > L(I)/\mathrm{OPT}(I)$ を満足する問題例 I の存在を必ず示せるなら，ϵ は tight といふ[*5] (もうこれより大きくできないギリ)．

さて，最適値の見積もりには，元の問題における制約条件を緩(ゆる)めた問題を考える．制約条件を緩和した結果，元の問題よりも実行可能解 (制約条件を凡て満たす解)が増えるから最適値が向上するであらう，といふ理屈である．

一般的な緩和の枠組みとして線形緩和，ラグランジュ緩和，代理緩和等あるけれど，先行制約を投げることで緩和する，つまり通常の bin-packing と看做(みな)すことで得られる下界が $L_1 := \lceil \sum_j w_j/C \rceil$ (Korte と Vygen [8, p. 473])

[*5] [2, p. 1493] では，if for any $\epsilon' < \epsilon$，と書いたあるものの，不等号が逆の誤植だらう．

で[*6], 逆に先行制約のみ考慮してビンの容量を忘れる, つまり先行制約を表す有効グラフに於いて, 最長パスを構成する節点数が L_2 である. たとへば先行制約に推移的なものがなく, 図1のやうな二項間の関係で凡てが完結するとき $L_2=2$ である (図2参照). これら下界ふたつに就いては, 最適値 (必要最低限のビン数) がいくらでも大きくできる ($=n$) にも拘(かかわ)らず常に1を返す最悪の例をそれぞれに対して簡単に構成できるけれど, $L_3:=\max\{L_1, L_2\}$ としただけで性能比 1/3 が出てゐるのは面白い.

定理1が, 下界 L_3 の性能比は 1/3 だと主張してゐる. さらに証明の最後には, 項数 $n\to\infty$ のときに L_3/OPT が上から 1/3 に限りなく近づく例題がある.

証明中で構成される実行可能解 H が使うビン数は最適値以上につき, 最終的に, 最適値が $3L_3$ 以下であると云ってゐる. また, 式 (14) の前に, 先行制約がない通常の bin-packing における最適解ではロード (詰められた項の重量の総和) がビンの半分未満 (実際には半分以下[*7]) であるやうなビンは

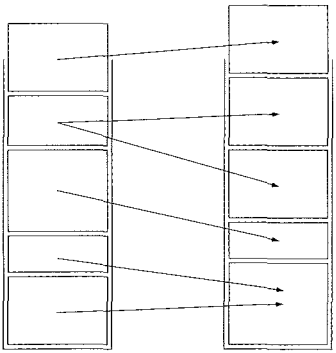

図2 L_2 はビンの容量を気にしない

[*6] 天井 (ceiling) 函数 $\lceil\ \rceil$ は, 小数以下を切り上げる. $\lceil 2.3\rceil=3$.

[*7] だから, 細かいことなんだけど, 半分以下しか入ってないかも知れないビン一本の中味のことを加味せずとも (14) は厳密に不等号で成立し, 最終的に $1/3<L_3$/OPT ではあるまひか.

高高ひとつとある．といふのは，もしそんなビンがふたつあったら，合はせてひとつにしてビンを減らせるからである．

下界 L_4 は，先行制約を表すグラフにおける最長パスを与へる項群 P の元それぞれがひとつずつ収められてゐる L_2 個のビンに就いて，残りの項全体の集合 Q から，先行制約があって同じビンには入れられない項の組を考慮しつつ最初から全部は入らない項を排除しつつ，その一部でも構わないのでその L_2 個のビン全体に向かって重量合計 f を流し込み(最大流問題)，結果として余る $\sum_{j\in Q} w_j - f$ については L_1 と同じ考え方 (11) をあて嵌める，といふものである．

定理2で，L_4 の性能比が 1/2 であることが示される．証明は定理1のそれと同じ初期解 H (ビン数 L_2) から始まり，各ビン i には最長パスを与える項群 P の元 $p(i)$ がひとつずつ収まってゐる．

証明の途中，$R(i)$ を通常のbin-packingとして解いたとき (ビン数 l_i)，項 $t(i)$ がもっともロードの少ないビンに入らない場合 (B) について：もし $w_{t(i)}$ が $C/2$ 以下であれば，それが入らない解であるなら，どのビンのロードも $C/2$ を超えてゐるし；$C/2$ より大きければ，$t(i)$ が $R(i)\cup\{t(i)\}$ の最軽量であることから，$R(i)$ に属する元の重量はどれも $C/2$ を超えてゐることになる[*8]．よって，どっちの場合であっても $(C/2)l_i < \sum_{j\in R(i)} w_j$ がしたがう．

最終的に定理1の証明と同様，ここで構成される実行可能解 H の使うビン数 z_H が $z_H \le 2L_4$ を満たすから，最適値が $2L_4$ 以下 —— つまり，$1/2 \le L_4/\mathrm{OPT}$[*9]．

以下では，話を簡単にするために慣例に倣って $C=1$ とする．定理2の証明の最後に，1/2 が tight であることを示すために，L_4/OPT を上からいくら

[*8] ほんとうに些細なことながら [2, p. 1496] では，$t(i)$ が $R(i)\cup\{t(i)\}$ の中で最小のビン (bin) とあるけれど，最小の項 (item) のミスプリ．

[*9] 小さいことで恐縮だが，解 H を構成する各ビン i に於いて，$C - w_{p(i)} < w_{t(i)} + \sum_{j\in S(i)} w_j$ でなければ $S(i)$ の最適性に反する —— $S(i)$ に加えてはもう何も入らないはず．したがって $f < \sum_{i=1}^{l_2} (w_{t(i)} + \sum_{j\in S(i)} w_j)$ であり，最終的に $1/2 < L_4/\mathrm{OPT}$ ではないか．

でも 1/2 に近づけることのできる例題として，先行制約は皆無で，凡ての項の重量が $1/2+\epsilon$ なる例（ϵ は，十分小さい正の数）が提示してある．そこで項数 n は偶数であり，先行制約がないから例えば $P=\{1\}$ として，どのふたつの項も同じビンには入らないので Q から P に伸びる枝はなく $f=0$ であり，$L_2=1$ と合はせて $L_4=1+\lceil\sum_{j\in Q}w_j\rceil=\lceil 1+(1/2+\epsilon)(n-1)\rceil=\lceil n/2+1/2+(n-1)\epsilon\rceil=n/2+1$．ゆゑに $L_4=L_1=n/2+1$ を得る．思ふに，この例に $n/2+1$ より大きい下界を返す簡便な手はないものだらうか．それが L_1 に替はって L_4 を良くするやも知れない．

bin-stretching にかんするメモ

本節は，bin-stretching への Kellerer と Kotov [6] が提案した算法——引伸し係数これまでの最小値 1.6 を塗り替へる $11/7=1.\dot{5}7142\dot{8}$ を実現する——にかんするメモになる．

用意された容量 1 のビン m ヶに凡てを納め得ることがあらかじめ分かってゐる，それぞれが重さを持つ項の列が online で次次にやってくる端からビン m ヶに詰めていくことを考える．通常，ビンの容量がそのままで online の格納は無理だから[*10]，ビンの容量を $\beta(\geq 1)$ に引き伸すとして，この引き伸す係数（stretching factor）β の最小化を目的とするのが bin-stretching である．$m=1$ ならもちろんビンを伸ばす必要はなく $\beta=1$．

Kellerer と Kotov [6] が提案した，bin stretching への引伸し係数 11/7 を実現する online 算法は，ふたつの段階（phase）から構成される：段階 1 [6, p. 345 の図 1] では，段階 2 で使うバッチの列を構成するための準備を行う．

[*10] 例へば sequences $\{0.4, 0.4, 0.5, 0.6\}$ と $\{0.4, 0.4, 0.7\}$ では，ふたつ目の 0.4 を格納すべきビンが異なるが，online 算法は今の項の後にどんな列が続いてくるか知らない．online 算法は offline 算法（開始前に全情報が開示される）のように振る舞えるけれども逆は無理．前出の FF は online 算法だけど，FFD はさうでない [8, p. 477]．

具体的に段階1は $3eB \leq msB(=mB+sB) \leq 3eB+3$ が成立した時点で終了する．ここに $eB, sB, mB, \ell B$ それぞれ空ビン，小ビン，中ビン，大ビンの総数を指す．各ビンはその中にある項の総重量別に次のやうに呼び分けられる：

その名称は	中味の総重量 $w(B)$ が
空ビン	0
小ビン	$0 < w(B) \leq 4/7$
中ビン	$4/7 < w(B) \leq 11/14$
大ビン	$11/14 < w(B) \leq 1$
超ビン	$w(B) > 1$

加へて各項の名称も，その重量がこの区分けに準じたものになる．段階1終了時に $sB > 0$ のとき，段階2では基本的には三つの小ビンと一つの空ビンからなるバッチと呼ばれる四つ組 $\boldsymbol{B}$ の列 $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_{k'}$ を構成し，小項 (small item, $\leq 4/7$) と中項 (medium item；$(4/7, 11/14]$) を前のバッチから，大項 (large item；$(11/14, 1]$) を後ろのバッチから FF で詰めていく．どのビンにも入らないとき，今のバッチを閉じて次のバッチを開ける．段階1の終了条件に関聯して，$\lambda = msB - 3eB$ と定義される．$\lambda = 0$ なら最後のバッチ $k' = k\ (:= eB)$ はビン四つ；でなければ最後のバッチ $k' = k+1$ は空ビンなしでビン数 $\lambda\ (1 \leq \lambda \leq 3)$．また，大項は小ビンに入る ($\leq 1+4/7 = 11/7$；ついでに逆で，小項は大ビンに入る)．

段階1で特徴的な点として，

- 小項を集めても中ビンにはならない．小項が (online で) 来たとき，大項の入るビン (large item bin，大ビンか超ビン) に入れる (結果，大ビンが超ビンになるかも) か，小ビンに入れて小ビンのまま[*11] か，

[*11] tiny 項 ($\leq 2/7$) を tiny ビンに入れても小ビンのままだから tiny ビンは高高ひとつ (補題1(c))．

空ビンに詰めて小ビンを作るかの三通りしかない.

- 中項が絡(から)むのは，それがビンに一つだけ入って他は無しか，あるいは二つ入って超ビン（$\leq 11/14+11/14=11/7$ でギリ）かの二通りだけ（補題1の (b) と (e)）．つまり中項は，他の種類の項と一緒になることがない.
- 大項を空ビンに入れる前に考慮するのは，最大の重量を持つ小ビンのみ[*12]で，先に触れたように，中ビンとは絡まない.

が挙げられよう.

補題1の (f) では『$sB=0$ or $\ell B=0$』とある．段階2における算法の説明では，Case 1 が $sB=0$ のとき，Case 2 が $\ell B=0$，$sB>0$ のときで，$sB=\ell B=0$ の場合の指示は特にない．Case 1 に含まれる特殊な場合ではあるけれど，ちょっと気になる．補題1の (f) の証明で，$sB^j=0$，$\ell B^j=1$ かつ時刻 j で来たのが小項のとき，大ビンにその小項が加へられて超ビンになったとすると，$sB^{j+1}=\ell B^{j+1}=0$ は起こり得る．では実際に $sB=\ell B=0$ のとき，定理1の証明で Case 1, i.e. $sB=0$ の場合には $eB=0$ が示されてゐる．したがって，凡てのビンのロード（入ってゐる項の総重量）が1を超えることはない[*13]ところへ $mB\leq 1$ (補題1の (c)) を加味すれば，$mB=1$ しかない（超ビンが無くてもさうだけれど，そもそも超ビンが無いなら $m=1$ で，さう云った例は最初から除外してよい）．よって，段階2で加へられる項はこの，唯一ロードが1未満の中ビンに残らず入ることになる．結局，段階1の終了時に $sB=0$ のときは，段階2ではただ FF で残りを詰めるとあるけ

[*12] なぜ小ビンの中でも最大の重量のなのかは，補題1の(f)の証明と関係してゐる.

[*13] 項全体は最初に与へられた mヶのビンに入ることが前提なので，全部のビンのロードが1を超えるのは物理的に不可能．同様に，定理1の証明で $\boldsymbol{B}_{1,\ldots,r-1,r+1,\ldots,k'}$ に属するビンのロードが平均して1を超えてゐるのだから，残るひとつのバッチ $\boldsymbol{B}_r$ に属する四つのビンのロード，i.e. 段階1で入ったのと段階2で入る予定の項の総重量が4以上のはずがない．これは，ビンが三つ以下の $r=k+1$ のときも同じ.

れど，$sB=\ell B=0$ であれば FF を持ち出すまでもないやうだ．

段階 1 終了時 $eB=sB=0$ で全部が中ビン以上である Case 1 につけ加へて，（もともと全部の項が m ヶのビンに入る前提なんだから）この場合には 1/2 を超える中項や大項が just m ヶ（結局，中項は一個以下）で，段階 2 でやってくる残りの項はみな小項（$\leq 4/7$）である．まだ算法が未終了の時点で m ヶのビン全部のロードが ≥ 1 は不可能で，どれかのビンのロードは <1．したがって，FF がこれから入れるビンについて $<1+4/7=11/7$ が，Case 1 では保証され続ける．

定理 1 の証明中最後の段落，$r=k+1$（最後に開いたバッチが列の最後尾で且つ $\lambda>0$）のとき：ビンが二つ B_1，B_2 の場合で，x_1 が B_1 に入らなかった最初の項とすると，B_2 に段階 1 で入った重量と x_1 以降にやってくる残り重量の総計は $2-w(B_1)-x_1<3/7$ 未満だから；x_1 が大項であったとしても $<3/7+1=10/7<11/7$ なので，x_1 を含めて以降全部が B_2 に入る，と云ふことなのだらう．別の見方をすると，x_1 を含んで以降の残りを凡て B_2 に詰めたとして，$w(B_1)+w(B_2)<2$ だから $w(B_2)<2-w(B_1)-x_1+x_1<3/7+1=10/7$．

さらに $r=k+1$ でビンが三つのとき，最初に B_1 に fit しなかった x_1 が B_2 に入ったら，x_1 がまだどのビンにも assign されていないときの snapshot に於いて $w(B_1)+w(B_2)+w(B_3)+x_1+$(残り)$<3$ とすると，$w(B_3)+$(残り)$<w(B_2)+w(B_3)+$(残り)$<3-w(B_1)-x_1<10/7$．これから，x_1 が B_2 に入ったあと残り全部が B_3 に入る；x_1 が B_2 にも fit しなかったら[*14]，$w(B_1)+y+w(B_3)+x_1+$(残り)<3 から(残り)$<w(B_3)+$(残り)$<3-w(B_1)-y-x_1<$

[*14] 本文中『as in (b)』とあるやうに，最後に閉じられるバッチ $\boldsymbol{B}_r$ にかんする考察の (b) と同じく，B_1 に最初に入らなかった x_1 が（まだ段階 2 では手つかずの）B_2 に入らないとは，大項は小ビンに入るから，つまり B_2 が（中項 y を含む）中ビンだったということ（なおかつ x_1 は大項 —— 中ビンに中項は入るから）．存在すれば唯一の中ビンは，先頭バッチ $\boldsymbol{B}_1$ のふたつ目に配置してある（ついでにいうと，存在すれば唯一の tiny ビンは $\boldsymbol{B}_1$ のひとつ目に配置済）から，これは，バッチがひとつきり（$r=1$）でしかもビンは三つ（段階 1 の終了時に空ビン無し）と云ふことになる．段階 1 終了時に $eB(=k)=0$，$msB=\lambda=3$ はアル．

$10/7-y$．よって B_2 では y+(残り)$<10/7$ となって x_1 後の残りが全部入る，と云ふ解釈でいいのだらう，たぶん．

最後に記述されてゐる，β の下界 4/3 と，そこで示された 11/7 の差がどう云ふ風に縮まっていくのか，興味深いところである．

参考文献

[1] Alberto Ceselli and Giovanni Righini, An optimization algorithm for the ordered open-end bin-packing problem. *Oper Res* **56**(2) 425–36 (2008) http://dx.doi.org/10.1287/opre.1070.0415.

[2] Mauro Dell'Amico, José Carlos Díaz Díaz and Manuel Iori, The bin packing problem with precedence constraints. *Oper Res* **60**(6) 1491–504 (2012) http://dx.doi.org/10.1287/opre.1120.1109.

[3] György Dósa, The tight bound of first fit decreasing bin-packing algorithm is $FFD(I) \leq 11/9OPT(I)+6/9$. In: B. Chen, M. Paterson and G. Zhang (Eds.) ESCAPE 2007, LNCS 4614, pp. 1–11, Springer 2007, http://dx.doi.org/10.1007/978-3-540-74450-4_1.

[4] György Dósa, Zsolt Tuza and Deshi Ye, Bin packing with "Largest In Bottom" constraint: tighter bounds and generalizations. *J Comb Optim* (2011) http://dx.doi.org/10.1007/s10878-011-9408-0.

[5] Hiroshi Iida, A note on the max-min 0-1 knapsack problem. *J Comb Optim* **3**(1) 89–94 (1999) http://dx.doi.org/10.1023/A:1009821323279.

[6] Hans Kellerer and Vladimir Kotov, An efficient algorithm for bin stretching. *Oper Res Lett* **41**(4) 343–6 (2013) http://dx.doi.org/10.1016/j.orl.2013.03.005.

[7] Hans Kellerer, Ulrich Pferschy and David Pisinger, *Knapsack Problems.* Springer 2004.

[8] Bernhard Korte and Jens Vygen, *Combinatorial Optimization—Theory and Algorithms (Algorithms and Combinatorics, Volume 21)* 5th Edition, Springer 2012, http://dx.doi.org/10.1007/978-3-642-24488-9_18.

[9] Chung-Lun Li and Zhi-Long Chen, Bin-packing problem with concave costs of bin utilization. *Naval Res Logist* **53**(4) 298–308 (2006) http://dx.doi.org/10.1002/nav.20142.

[10] David Simchi-Levi, New worst-case results for the bin-packing problem. *Naval Res Logist* **41**(4) 579–85 (1994).

[11] Binzhou Xia and Zhiyi Tan, Tighter bounds of the First Fit algorithm for the bin-packing problem. *Discrete Applied Mathematics* **158**(15) 1668–75 (2010) http://dx.doi.org/10.1016/j.dam.2010.05.026.