

## 積み込み、積み下ろし順序を考慮した 3次元パッキング問題に関する研究

小畑尚輝<sup>†</sup> 太田秀典<sup>†</sup> 中森眞理雄<sup>†</sup>

荷物の積み込み順序と積み下ろし順序が与えられたコンテナ積み付け問題について考える。ただし、充填率をあげるため、荷物をコンテナに積み込む（積み下ろす）際に、コンテナ内の邪魔な荷物を一旦取り出し、目的の荷物を積み込んでから（積み下ろしてから）、取り出した荷物を再びコンテナに積み込むことを許容する。荷物の取り出し回数が増えると作業時間や労力が増大するため、より少ない取り出し回数で荷物の積み込み（積み下ろし）が行われるべきである。本論文では荷物の取り出し回数が最小となる荷物配置を求めるアルゴリズムを提案する。

### A Study on Three Dimensional Packing Problem Considering the Order of Loading and Unloading

NAOKI KOBATAKE<sup>†</sup> HIDENORI OHTA<sup>†</sup>  
MARIO NAKAMORI<sup>†</sup>

In this paper, we discuss the packing problem where the order of items loading and unloading is supplied. To obtain the high packing ratio, we permit to unpack the item if the item blockades other items loading (unloading), and after the loading (unloading), the blockading item will be re-packed into the container. A large number of unpacking and re-packing makes the cost increase. So this paper propose an algorithm to pack the items into the container with minimized number of unpacking and re-packing.

#### 1. はじめに

トラック、コンテナ船等による荷物輸送において、より多くの荷物をコンテナに積み込むことは、輸送コスト削減の観点から非常に重要である。そのため、コンテナへの荷物の積み付けは、大きな直方体容器に小さな直方体オブジェクトをなるべく密に配置する“直方体パッキング問題”へと帰着させて考えることが多い。直方体パッキング問題に対しては、これまで様々な解法が提案されており、現在も盛んに研究が行われている[1 - 5]。

ところで、現実の積載計画を立てる場合には、荷物の積み込み順序、積み下ろし順序を考慮しなければならないことがある。例えば、船やトラックでいくつかの集積所を順番に訪れ、各集積所で荷物の積み込みや積み下ろしを行うことを考える。このとき、荷物の積み込み順序、積み下ろし順序は、集積所に立ち寄る順序に従うことになり、集積所に立ち寄る順序を変更しない限り、これらを変更することは出来ない。積み込み順序、あるいは積み下ろし順序のどちらかのみが指定された場合には、積み込み順序（もしくは指定された積み下ろし順序の逆順）に従って荷物をコンテナの奥から順番に配置することが考えられる。しかしながら、この方法で得られた配置では、密な荷物配置を得ることができない場合があり、与えられた荷物の全てがコンテナ内に収まらない恐れがある。

そこで本論文では、荷物の積み込み、積み下ろしの度に、既にコンテナに積んである荷物のいくつかを一旦コンテナから取り出し、目的の荷物を積み込んでから（積み下ろしてから）、取り出した荷物を再びコンテナに積み込むことを許容した積み付け問題を考える。

無論、荷物の取り出し回数が増えると、作業時間や労力がかかるほか、荷物破損等のリスクが増大するので、なるべく少ない取り出し回数で荷物の積み込み、積み下ろしを済ませることが求められる。本論文は、少ない取り出し回数で、与えられた荷物の積み込み、積み下ろしが可能となる荷物配置を求める効果的なアルゴリズムを提案する。

#### 2. 問題設定

コンテナへの積み込みと、コンテナからの積み下ろしを考慮した荷物の配置を考える。コンテナと荷物は直方体形状であり、コンテナは3次元ユークリッド空間上の $x$ 軸、 $y$ 軸、 $z$ 軸に沿って置かれている。なお、本論文では $x$ 方向を右方向、 $y$ 方向を手

<sup>†</sup> 東京農工大学  
Tokyo University of Agriculture and Technology

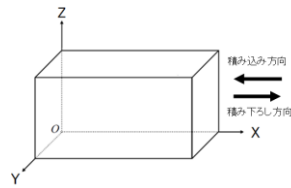


図 1. コンテナと荷物の積み込み, 積み下ろし方向

前方向,  $z$  方向を上方向とする. コンテナへ積み込まれた荷物は, その重心等を考慮に入れないこととして, 重なる領域さえなければどのような配置でも可能なことになる.

荷物にはそれぞれ積み込み時刻, 積み下ろし時刻が与えられており, これらに従い, 1 つずつ順番に荷物の積み込み, 積み下ろしを行う. 荷物の取り出し口はコンテナの右面に位置しており, どの荷物も取り出し口を通して積み込みと積み下ろしを行うことにする. 図 1 にコンテナと荷物の積み込み方向, 積み下ろし方向について示す.

荷物の積み込み (積み下ろし) 過程で, 既にコンテナに配置されている荷物が邪魔で, 決められた位置に目的の荷物を移動できないことが考えられる. このときは, 邪魔な荷物 (以下, 邪魔荷物と呼ぶ) を一旦コンテナから積み降ろし, 目的の荷物を積み込んでから (積み下ろしてから), 邪魔荷物を再びコンテナに積み込むことにする. 邪魔荷物の積み下ろしのべ回数を“無駄な取り出し回数”と呼ぶことにする. なお, 本論文では簡単のため, 積み下ろした邪魔荷物は, 全て元の位置に積み直されることにする.

荷物の積み込みの際に邪魔な荷物は, 積み込む荷物を  $a$  としたとき, 荷物  $a$  の配置位置と取り出し口の間にある荷物や, 荷物  $a$  の配置位置の真上に置かれている荷物である. 本論文では, 以下の 2 つの条件を両方満たす荷物  $b$  は荷物を  $a$  を積み込む際の邪魔荷物とみなす:

- (i) 配置後の荷物  $a$  の手前面が荷物  $b$  の奥面より手前にあるか, 配置後の荷物  $a$  の奥面が荷物  $b$  の手前面よりも奥にある;
- (ii) 以下の 2 つの条件の内, 少なくとも片方を満たす:
  - ・配置後の荷物  $a$  の右面が荷物  $b$  の左面より左にあり, 配置後の荷物  $a$  の下面が荷物  $b$  の上面よりも下にある;
  - ・配置後の荷物  $a$  の上面が荷物  $b$  の下面より下にあり, 配置後の荷物  $a$  の左面が荷物  $b$  の右面よりも左にある.

荷物を積み下ろすときの邪魔荷物も, 積み込むときの邪魔荷物と同様に定義できる. 図 2 (a) に示した配置に荷物  $b$  を破線で示した位置に配置することを考えると, 荷物  $a$  は邪魔荷物となるので, 無駄な取り出し回数は 1 である. 図 2 (b) に示した配置から

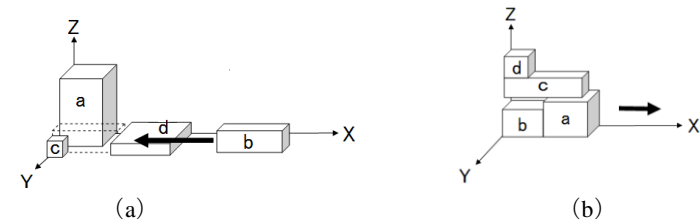


図 2. 積み込み, 積み下ろしにおける邪魔荷物

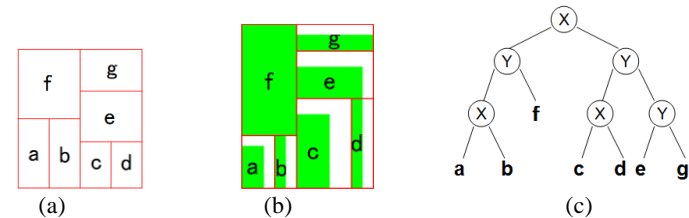


図 3. 2次元スライス構造, スライス構造に基づく矩形パッキング, 対応する2次元スライス木

荷物  $a$  を取り出すことを考えると, 荷物  $c$  は邪魔荷物である. そこで荷物  $c$  を取り出すことを考えるが, 荷物  $c$  を取り出すには荷物  $d$  が邪魔荷物となる. 結果, 荷物  $a$  を取り出すのに必要な, 無駄な取り出し回数は 2 となる.

本論文の目的は無駄な取り出し回数の総和が最小となる荷物の配置を求めることである.

### 3. スライス木によるパッキング表現

#### 3.1 2次元スライス構造[4]

矩形領域 (全体矩形) を水平線分 (水平分割線) または垂直線分 (垂直分割線) によって 2 つの矩形領域に分割する操作を再帰的に行うことで得られる構造は 2 次元スライス構造と呼ばれる. この分割によって得られる小矩形領域 (部屋) に 1 対 1 で矩形を割り当てることによって矩形パッキングが得られる. 図 3 (a) に 2 次元スライス構造の例を, 図 3 (b) に図 3 (a) のスライス構造に基づいた矩形パッキングを示す.

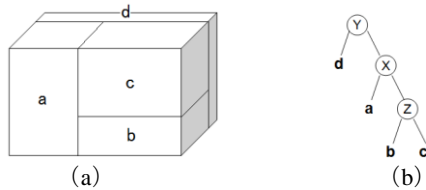


図 4. 3次元スライス構造, 対応する3次元スライス木

スライス構造はスライス木と呼ばれる2分木で表現可能である. スライス木の内部節点は分割線と, 外部節点は部屋とそれぞれ対応する. 垂直分割線に対応する内部節点は“X 節点”で表され, 垂直分割線の左側の矩形領域は X 節点の左部分木に, 右側の矩形領域は右部分木にそれぞれ対応する. 同様に, 水平分割線は“Y 節点”で表され, 水平分割線の下側の矩形領域は左部分木に, 上側の矩形領域は右部分木にそれぞれ対応する. 図 3 (c) に図 3 (a) のスライス構造を表現するスライス木を示す.

### 3.2 3次元スライス構造[5]

2次元スライス構造は3次元スライス構造へと容易に拡張できる. 6つの面がそれぞれ  $x$  軸,  $y$  軸,  $z$  軸と垂直である直方体領域 (全体直方体) をいずれかの軸に垂直な平面 (分割面) によって2つの直方体領域に分割する操作を再帰的に行うことで得られる構造が3次元スライス構造とする. 分割によって得られる小直方体領域 (部屋) に1対1で直方体 (荷物) を割り当てることで直方体パッキングが得られる. 図 4 (a) に3次元スライス構造の例を, 図 4 (b) に図 4 (a) のスライス構造に基づいた直方体パッキングを示す.

3次元スライス構造もまた, スライス木で表現可能である. スライス木の内部節点は分割面と, 外部節点は部屋とそれぞれ対応する.  $x$  軸 ( $y$  軸/ $z$  軸) に垂直な分割面に対応する内部節点は“X 節点 (Y 節点/Z 節点)”で表され, 分割面の左側 (手前/下側) の直方体領域は X 節点 (Y 節点/Z 節点) の左部分木に, 分割面の右側 (奥/上側) の直方体領域は右部分木にそれぞれ対応する. 図 4 (c) に図 4 (a) のスライス構造を表現するスライス木を示す.

3次元スライス木において, それぞれの部分木はスライス構造内の直方体領域と対応している. 外部節点に対応する直方体領域 (部屋) には, その部屋に対応する荷物以上の大きさが必要である. ここで, 任意の X 節点  $P$  の左の部分木  $L$  に対応する直方体領域の各辺の長さを  $X_L, Y_L, Z_L$ , 右の部分木  $R$  に対応する直方体領域の各辺の

長さを  $X_R, Y_R, Z_R$  としたとき,  $P$  を根とする部分木に対応する直方体領域が必要とする直方体領域の各辺の長さ  $X_P, Y_P, Z_P$  は,

$$X_P = X_L + X_R, Y_P = \max(Y_L, Y_R), Z_P = \max(Z_L, Z_R)$$

となる. Y 節点, Z 節点を根とする部分木に対応する直方体領域についても同様に必要な直方体領域の大きさが求まる. この性質から, スライス木を後順で走査することでスライス木に対応する荷物配置を  $O(n)$  時間でデコード可能である. なお,  $n$  は荷物数である.

全体直方体の各辺の長さがコンテナの各辺の長さよりも短いとき, スライス木で表現された荷物の配置はコンテナ内に全て収まることになる.

## 4. 提案手法

3次元スライス木を拡張し, 荷物の積み込み, 積み下ろしを考慮したパッキング表現を行い, この表現方法を用いて無駄な取り出し回数を最小化した荷物の配置を求める. 荷物配置の探索は2段階にて行われ, Phase 1 では全ての荷物をコンテナに収めた配置の探索を, Phase 2 では Phase 1 で得られた配置を初期解として, 無駄な取り出し回数の最小化をそれぞれ行う.

### 4.1 積み込み, 積み下ろしを考慮した直方体パッキングの表現方法

本論文で扱う問題は荷物の積み込み, 積み下ろしを考慮している. 任意の2つの荷物について, 片方の荷物を積み込むよりも早くにもう片方の荷物を積み下ろす場合, 2つの荷物は同時にはコンテナ内に存在しないことになるので, これらは互いの位置を考慮せずに, 同じ場所に配置可能である. 複数の荷物を同じ場所へ配置することを表現するために, スライス木の内部節点として“T 節点”を新たに導入する. なお, T 節点の左の部分木上にある任意の荷物と, 右の部分木上にある任意の荷物は, 同時にコンテナ内に存在しないことにする. 図 5 (a) に従来の3次元スライス木を, 図 5 (b) に図 5 (a) の3次元スライス木に T 節点を導入した3次元スライス木を示す. 外部節点の下の数字は, 対応する荷物の積み込み時刻, 積み下ろし時刻である. また, 図 5 (b) の3次元スライス木に対応する荷物配置を図 6 に示す.

荷物の積み込み時刻, 積み下ろし時刻が与えられているとき, 従来の3次元スライス木の指定された内部節点が T 節点に変更可能かどうかは, その節点の左の部分木,

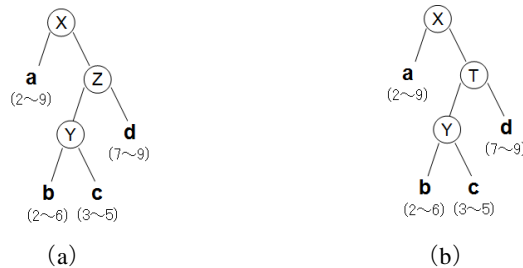


図 5. 従来の 3 次元スライス木, T 節点を導入した 3 次元スライス木

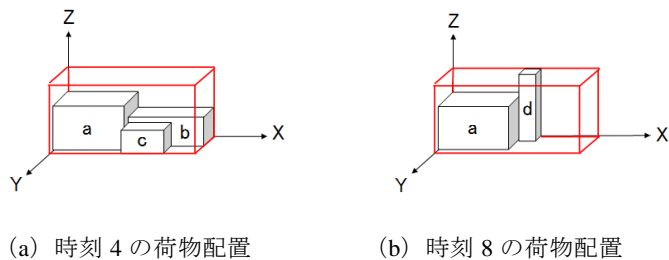


図 6. 図 5 (b) に示した T 節点を導入した 3 次元スライス木と対応する配置

右の部分木にある荷物の最も早い積み込み時刻, 最も遅い積み下ろし時刻をそれぞれ調べることで判定可能である. よって, 3 次元スライス木を後順で走査することにより,  $O(n)$  時間で T 節点に変更可能な内部節点を全て列挙することが可能である.

任意の T 節点  $P$  の左の部分木  $L$  に対応する直方体領域の各辺の長さを  $X_L, Y_L, Z_L$ , 右の部分木  $R$  に対応する直方体領域の各辺の長さを  $X_R, Y_R, Z_R$  としたとき,  $P$  を根とする部分木に対応する直方体領域が必要とする各辺の長さ  $X_P, Y_P, Z_P$  は,

$$X_P = \max(X_L, X_R), Y_P = \max(Y_L, Y_R), Z_P = \max(Z_L, Z_R)$$

となる.

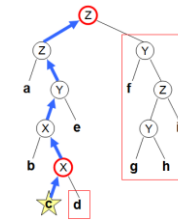


図 7 荷物  $c$  の積み込み, 積み下ろしの際の邪魔荷物を含む可能性のある部分木

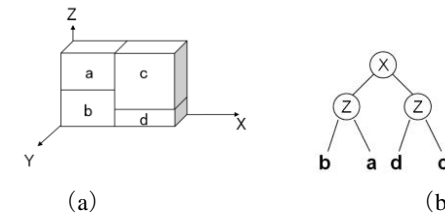


図 8 無駄な取り出し回数を多く数えてしまう配置, 対応する 3 次元スライス木

#### 4.1.1 無駄な取り出し回数の数え上げ

3 次元スライス木上で, 対応する荷物配置での無駄な取り出し回数を数え上げることを考える. 任意の荷物  $a, b$  について,  $a$  の積み込み (積み下ろし) の際に  $b$  が邪魔荷物となるための条件として, 以下の 3 つの必要条件が挙げられる:

- (i)  $a, b$  の最小共通先祖  $c$  が X 節点, Z 節点のどちらかである;
- (ii)  $a$  は  $c$  の左部分木に含まれる;
- (iii)  $a$  の積み込み時刻 (積み下ろし時刻) に,  $b$  がコンテナ内に存在する.

この 3 つの条件を満たす荷物は, 3 次元スライス木を走査して,  $O(n)$  時間で列挙することが可能である. 例えば, 図 7 に示した 3 次元スライス木に対応する荷物配置では, 荷物  $c$  の積み込み, 積み下ろしの際の邪魔荷物は, 方形で囲んだ部分木の中にのみ存在する.

任意の荷物の積み込み (積み下ろし) の際の無駄な取り出し回数は  $O(n)$  で数え上げられるので, 配置全ての無駄な取り出し回数は  $O(n^2)$  で求めることが可能である.

なお, 上に挙げた条件は必要条件であるため, 実際の無駄な取り出し回数は, より少なくなることが考えられる. 例えば, 図 8(a) に示した配置は図 8(b) のスライス木で

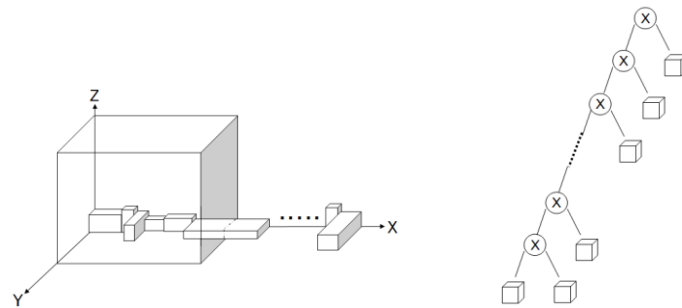


図9 荷物の初期配置と対応するスライス木

表現される. この配置から荷物  $a$  を取り出すことを考えると, 荷物  $c$  のみが邪魔荷物となるが, 本論文の手法では荷物  $d$  も無駄荷物として扱うことになっている.

#### 4.2 Phase 1

Phase 1 では無駄な取り出し回数は考慮せずに, 全ての荷物をコンテナ内に配置することを目的とする. 探索は Simulated Annealing (SA 法) によって行い, 探索のコスト関数は, 全体直方体の  $x$  幅とする. ただし, 全体直方体の  $y$  幅 ( $z$  幅) がコンテナの  $y$  幅 ( $z$  幅) よりも長いときは, その差に比例したペナルティ値を加える. 探索は従来の 3 次元スライス木で行い, 評価の際には T 節点に変更可能な節点を全て T 節点に変更する. 初期解は図 9 のように荷物を横一列に並べた状態とし, 近傍解の生成方法は以下の 4 つの方法から毎回ランダムに選択する.

- ・部分木交換: ランダムに 1 つの荷物を選択し, 荷物の積み付け向きを変更する.
- ・部分木交換: 根以外の節点からランダムに 2 つの節点を選択し, それらの節点を根とする部分木同士を交換する. ただし, 片方の部分木にもう片方の部分木が含まれていた場合は交換を行わない.
- ・荷物交換: ランダムに 2 つの外部節点を選択し, 対応する荷物を交換する.
- ・配置変換: ランダムで選んだ内部節点の種類 (X 節点, Y 節点, Z 節点) を変換する.

#### 4.3 Phase 2

Phase 2 では無駄な取り出しの回数を最小化した配置を探索する. Phase 2 もまた, Simulated Annealing (SA 法) を用いる. 探索の評価値は無駄な取り出しの回数である. ただし, 全ての荷物がコンテナ内に配置されない解は非許容解として扱う. 初期解は

Phase 1 の探索で得られた配置を用い, 近傍解の生成は, Phase 1 のときに用いた 4 つの方法に加え, 左右の部分木の入れ替えによる無駄な取り出し回数の最小化を一定の探索回数毎に行う.

##### 4.3.1 左右の部分木の入れ替えによる無駄な取り出し回数最小化

3 次元スライス木では, どの節点の左右の部分木を入れ替えても, 全体直方体が必要とする大きさは変化しない. これは許容解である 3 次元スライス木の任意の節点の左右の部分木を入れ替えても, 非許容解に陥ることがないことを意味している. 以下に, 与えられた 3 次元スライス木から全ての兄弟節点对の左右関係を最適化するアルゴリズムを示す. アルゴリズムの計算複雑度は  $O(n^2)$  である.

```

※入力: 3次元スライス木
※出力: 全ての兄弟節点对の左右関係を最適化した3次元スライス木
while (全ての内部節点  $i$  について) {変数  $C(i)$  を用意,  $C(i)=0$  に初期化;}
while (全ての外部節点  $e$  について) {
    while ( $e$  の全ての先祖節点  $a$  について) {
        if ( $a$  が X 節点, Z 節点のいずれはである) {
            if ( $e$  は  $a$  の左の部分木に含まれる) {
                while ( $a$  の右の部分木に含まれる全ての外部節点  $o$  について) {
                    if ( $o$  が  $e$  の積み込み時刻にコンテナ内に存在)  $C(a)++$ ;
                    if ( $o$  が  $e$  の積み下ろし時刻にコンテナ内に存在)  $C(a)++$ ;
                }
            }
        }
        else {
            while ( $a$  の左の部分木に含まれる全ての外部節点  $o$  について) {
                if ( $o$  が  $e$  の積み込み時刻にコンテナ内に存在)  $C(a)--$ ;
                if ( $o$  が  $e$  の積み下ろし時刻にコンテナ内に存在)  $C(a)--$ ;
            }
        }
    }
}
while (全ての内部節点  $i$  について) {
    if ( $C(i) > 0$ )  $i$  の左右の部分木を入れ替え;
}
    
```



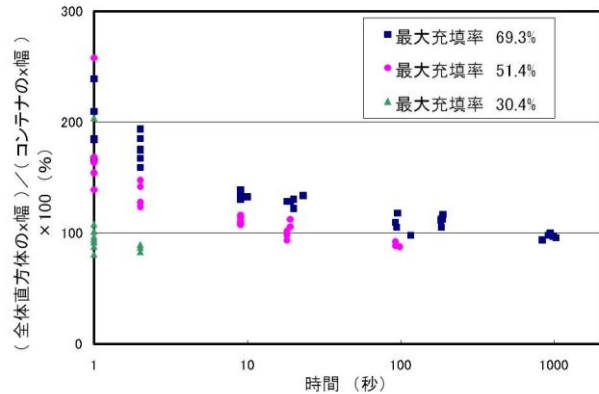


図 10 Phase 1 における探索時間と全体直方体の x 幅の関係

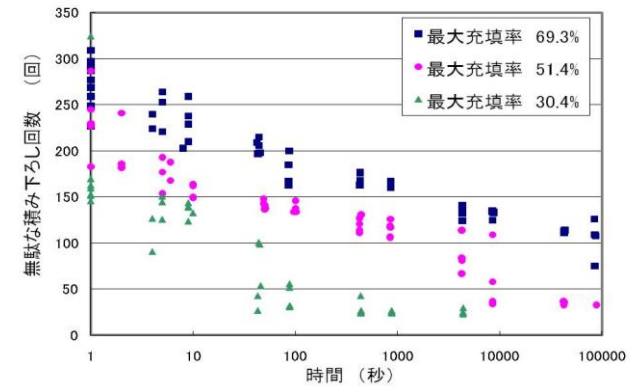


図 11 Phase 2 における探索時間と無駄な取り出し回数の関係

## 5. 計算機実験

提案手法を C 言語にて実装し，計算機実験を行った．実験には CPU が Intel Core 2 Duo T7300 2.00GHz，メモリが 2GB の計算機を用いた．

荷物として，1 辺の長さが 10 から 30 のランダムな整数値である直方体 40 個を用意し，それぞれの荷物にランダムに積み込み時刻，積み下ろし時刻を与えた．コンテナは実際のトラックの荷台を参考にし，各辺の長さの比は 5:2:3 になる直方体を 3 種類用意した．それぞれ最大充填率は 69.3%，51.4%，30.4%，のべ充填率が 123.7%，91.6%，54.3% である．なお，最大充填率とは，コンテナ内の荷物の総体積が最大となる時刻の充填率であり，のべ充填率とは，コンテナ体積に対しての全ての荷物の体積和の割合である．

図 10 に Phase 1 における全体直方体の x 幅がコンテナの x 幅に占める割合と時間の関係を示す．約 1000 秒で，実験に使用したどのコンテナにも許容解が発見できている．

図 11 に Phase 2 における無駄な取り出し回数と時間の関係を示す．それぞれ，無駄な取り出し回数が 75 回，33 回，23 回の解が発見できた．これは 1 回の積み込み（積み下ろし）あたり平均 0.94 回，0.41 回，0.26 回の無駄な取り出しを行うことを意味する．図 12 に最大充填率が 69.3% となるコンテナへの配置で，無駄な取り出し回数が 75 回となる配置を示す．

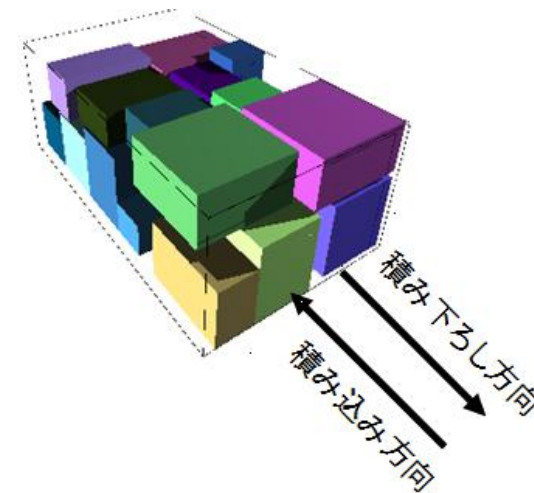


図 12 最大充填率 69.3% の時の荷物配置  
 (直方体数：23，探索時間：84721 秒，無駄な取り出し回数：75 回)

## 6. まとめ

本論文では積み込み、積み下ろし順序を考慮したコンテナ積み付け問題について扱った。今後の課題として、無駄な取り出し回数をより正確にカウントする手法の考案や、荷物を取り出して積み直す際に元にあった場所に戻さずに新たな場所に積み直すことを可能とするアルゴリズムの考案、荷物の重心等を考慮に入れたより現実的なコンテナへの積み付け要件を考慮した問題への取り組みなどが挙げられる。

## 参考文献

- 1) Hidenori Ohta, Toshinori Yamada, Chikaaki Kodama, and Kunihiro Fujiyoshi, "The O-Sequence: Representation of 3D-Dissection," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences archive, Volume E91-A Issue 8, August 2008.
- 2) S. D. Allen, E. K. Burke, and G. Kendall, "A hybrid placement strategy for the three-dimensional strip packing problem," European Journal of Operational Research, Volume 209, Issue 3, 16 March 2011, pp.219-227.
- 3) Guenther Fuellerer, Karl F. Doerner, Richard F. Hartl, and Manuel Iori, "Metaheuristics for vehicle routing problems with three-dimensional loading constraints," European Journal of Operational Research, Volume 201, Issue 3, 16 March 2010, pp.751-759.
- 4) 川島大貴, 田中勇真, 今堀慎治, 柳浦睦憲, "3次元箱詰め問題に対する構築型解法の効率的実現法," FIT2010 第9回情報科学技術フォーラム, pp.31-38.
- 5) F. K. Miyazawa, and Y. Wakabayashi, "Three-dimensional packings with rotations," Computers & Operations Research, Volume 36, Issue 10, October 2009, pp.2801-2815.
- 6) D. F. Wong, C. L. Liu, "A new algorithm for floorplan design," IEEE DAC, pp.101-107, 1986.
- 7) L. Cheng, L. Deng, and M. D. F. Wong, "Floorplan Design for 3-D ICs," Proc. SASIMI, pp.395-401, 2004.