4

# A Multi-start Local Search Approach to the Multiple Container Loading Problem

Shigeyuki Takahara
*Kagawa Prefectural Industrial Technology Center*
*Japan*

## 1. Introduction

This paper is concerned with the multiple container loading problem with rectangular boxes of different sizes and one or more containers, which means that the boxes should be loaded into the containers so that the waste space in the containers are minimized or the total value of the boxes loaded into the containers is maximized. Several approaches have been taken to solve this problem (Ivancic et al., 1989; Mohanty et al., 1994; George, 1996; Bortfeldt, 2000; Eley, 2002; Eley, 2003; Takahara, 2005; Takahara, 2006). The aim of this paper is to propose an efficient greedy approach for the multiple container loading problem and to contribute to develop a load planning software and applications.

The problem considered in this paper is the three-dimensional packing problem, therefore it is known to NP-hard. This implies that no simple algorithm has been found so far. In this paper this problem is solved by a relatively simple method using a two-stage strategy. Namely, all boxes are numbered and for the sequence of the numbers a greedy algorithm of loading boxes into containers is considered. This greedy algorithm is based on *first-fit concept* (Johnson et al., 1974) and determines the arrangement of each box. This algorith try to load the boxes all kind of containers and select the best arrangement result. It's requires a box loading sequence and a set of orientation orders of each box type for each container type.

Each box is tried to load according to these dynamic parameters. Moreover, a static parameter overhang ratio is introduced here. This is a percentage of the bottom face area of the loading box that isn't supported with other boxes that are below. As shown by Takahara (Takahara, 2006), if this parameter is different, the arrangement given this algorithm is different in spite of using a same pair of the loading sequence and the orientation orders. Some initial pairs of solution, that is a box loading sequence and a set of orientation orders of each box type, are selected among some rules, which are given beforehand. This solution is altered by using a multi-start local search approach. The local search approach is used to find a good pair of solutions that is brought an efficient arrangement by the greedy loading algorithm. The arrangement obtained in the iterations is estimated by volume utilization or total value of the loaded boxes.

The effectiveness of the proposed approach is shown by comparing the results obtained with the approaches presented by using benchmark problems from the literature. Finally, the layout examples by the application using the proposed approach for container loading are illustrated.

Source: Advances in Greedy Algorithms, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

## 2. Problem description

The multiple container loading problem discussed in this paper is to load a given set of several boxes of varying size in one or more different containers so as to minimize the total volume of required containers or to maximize the total value of loaded boxes. This problem includes two kinds of problem, one is the problem with one container type and the other is the problem with different container types.

Before presenting the problem formulation, some notations used this paper are defined.

Let $n$ be the number of types of boxes and let $P = \{1,..., n\}$ . The box, the volume, the value per unit volume and the number of a box of type $i$ , $i \in P$ , are denoted by $b_i$, $v_i$ , $c_i$ and $m_i$, respectively. Each box of type corresponds to integer $i$ , $i \in P$ , and all permutations of the numbers $\rho = \{ \sigma : \sigma \ (b_1,..., b_i,..., b_n), \ i \in P\}$ are considered. The number of boxes is denoted by $T$ , i.e. $T = \sum_{i=1}^{n} m_i$.

Let $N$ be the number of types of containers and let $Q = \{1,..., N\}$. The container, the volume and the number of a container of type $h$ , $h \in Q$, are denoted by $C_h$, $V_h$ and $M_h$ Thus, if $N$ =1, then this problem is the one container type problem.

The boxes can be rotated. In consequence, up to six different orientations are allowed. Hence the rotation variants of a box are indexed from 1 to 6, an orientation order of each box type $i$ to load to the container type $h$ is denoted by $r_i^h = (u_{i1}^h,...,u_{i6}^h)$. If an edge (i.e. length, width or height) placed upright, the box of type $i$ can take two orientations among them. Each box of type $i$ is tried to load to the container of type $h$ according to this orientation order $r_i^h$. Here, a set of orientation orders $\lambda = \{\mu : \mu = (r_1^h,...,r_n^h), h \in Q\}$ is considered.

For each $\sigma \in \rho$ and $\mu \in \lambda$ , an algorithm that will be described below is applied, and loading positions of boxes are determined. The optimal solution is found by changing this permutation and this set of orders.

Practical constraints which have to be taken into account are load stability, weigh distribution, load bearing strength of boxes and so on. In this paper, in order to consider the load stability and vary the arrangement, overhang parameter $\gamma$ is introduced. This is a percentage of the bottom face area of the loading box that isn't supported with other boxes that are below. Therefore, a box isn't allowed to be loaded to the position in which this parameter isn't satisfied. Generally, $\gamma$ takes the value from 0% to 50%. Suppose that other constraints aren't taken into account here.

A loading algorithm $A$ and a criterion $F$ must be prepared for solving this multiple container loading problem. $A$ is greedy algorithm and determines the arrangement of loading position of each box $X$ according to the sequence $\sigma$ and the set of orientation orders $\mu$ within the range of $\gamma$ . This algorithm try to load the boxes to all kind of containers and select the container provided the best result. When all boxes are loaded by this algorithm and the arrangement is determined, the criterion $F$ can be calculated. The arrangement is estimated by volume utilization and the total value of the loaded boxes. If the result required the number of container of type $h$ is $M_h'$ and the number of the loaded box of type $i$ is $m_i'$, the volume utilization is represented by

$$U = \sum_{i=1}^{n} v_i m_i' \Big/ \sum_{h=1}^{N} V_h M_h' \tag{1}$$

Moreover, the total value of the boxes loaded into the containers is represented by

$$W = \sum_{i=1}^{n} v_i c_i \tag{2}$$

In this paper, the objective is to maximize the volume utilization and the total value of loaded boxes. Therefore, the criterion $F$ is denoted by

$$F(\sigma, \mu, \gamma) = \alpha U + \beta W \tag{3}$$

where $\alpha$, $\beta$ are constant number that depend on the problem. The optimality implies that this factor is maximized. Thus, the multiple container loading problem is denoted by

$$\max_{\sigma \in \rho, \mu \in \lambda, 0 \le \gamma \le 50} F(\sigma, \mu, \gamma) \tag{4}$$

When the calculation method for $F(\sigma, \mu, \gamma)$ is specified, the multiple container loading problem can be formulated as above combinatorial optimization problem. It naturally is adequate to use local search approach to obtain the optimal pair of the box sequence $\sigma$ and the set of orientation orders $\mu$ . It is supposed that the overhang parameter $\gamma$ is given before simulation. It should be noticed that the greedy loading algorithm and the multi-start local search can separately be considered.

## 3. Proposed approach

### 3.1 Greedy loading algorithm

"*Wall-building approach* (George & Robinson, 1980)" and "*Stack-building approach* (Gilmore & Gomory, 1965)" are well-known as heuristic procedure for loading boxes in container.

In this paper, *first-fit concept* is used as basic idea. Therefore, the proposed greedy loading algorithm consecutively loads the type of boxes, starting from $b_1$ to the last $b_n$. Since the total number of boxes is $T$, the box $b'_k(k=1,..,T)$ is loaded to the position $p_k$ in the container. Therefore, the arrangement of boxes $X$ is denoted by

$$X(\sigma, \mu, \gamma) = \{p_k : k = 1,...,T\} \tag{5}$$

If the box $b'_k(k=1,..,T)$ is loaded in the container $C_h^i$, $h \in Q$, $i = 1,..., M_h$ ,that means $p_h \in C_h^i$. The loading position $p_k$ is selected from a set of potential loading areas $S_k$, which consists of the container floor and the top surface of the boxes that have been already loaded. Therefore, the set of potential loading areas here is

$$S_k = (s_1,...,s_{g_k}) \tag{6}$$

where $g_k$ is the number of potential loading areas after loading k-1 boxes. The potential loading area is defined by a base point position, the length and the width. The base point is a point near the lower far left corner of the container. For example, in case of $k$ =1, the number of potential loading areas is 1 and $s_1$ is the container floor.

In order to solve this multiple container loading problem, the method can be divided by two parts. The first part is single container part, and the other part is different container part.

The single container part algorithm is to solve single container loading problem.

The single container part algorithm SCA uses the following steps.

[Algorithm SCA]

Step 1: Set the sequence $\sigma = (b_1,..., b_n)$, the set of orientation orders $\mu = (r_1^h,...,r_n^h)$ and the overhang parameter $\gamma$ for loading boxes;

decide the set of initial potential loading area $S_1$;
set $i$ = 1, $k$ = 1, and $j$ = 1 as index for the current box type, the current box, and the current orientation, respectively.

Step 2: If $i \leq n$ , then take the box of type $b_i$ for loading, else stop.
Step 3: If all boxes of type $b_i$ are already loaded, then set $i = i$ +1, $j$ =1 and go to Step 2.
Step 4: Scan the set of loading area $S_k$ and find the loading position.
If a position that can be loaded is not found, then go to Step 6.
Step 5: Load the box on the selected position by Step 4.
Set $k = k$ +1 and update the set of loading area $S_k$.
If the boxes of type $b_i$ are remained, then go to Step 2, else go to Step 7.
Step 6: If $j < 6$ , then $j = j$ +1 and go to Step 4.
Step 7: If $k \leq T$ , then set $i = i$ +1, $j$ =1 and go to Step 2, else stop.

This algorithm uses the orientation order of each box. The box of type $b_i$ is arranged in the orientation of $\mu_{b_i j}$ in Step 3. If the current orientation is not permitted, skip Step 3 and go to Step 5 to take next orientation that is determined by $r_{b_i}$. Each box has a reference point in any of the six orientations. This point is set to the lower far left corner of the box and is used for scanning the potential loading areas in Step 4. A position that the box is loaded means a position in which the reference point of the box is put. For scanning each potential loading area, the reference point is set on the base point and is moved to the direction of the width and the length. The potential loading area is examined by the order of $S_k$, that is, from $s_1$ to $s_{g_k}$. The position in which the box can be loaded is where it doesn't overlap with other boxes that are already loaded and the overhang parameter of this box is γ or less.

The update procedure of the set of potential loading area in Step 5 is as follows:

1. Update current area
   If the box is loaded to the base point of the selected loading area, the area is deleted. When it is loaded to other positions, the area changes the dimension of the length or the width, and remains.
2. Create new loading area
   New loading areas, that is, a top face of the box, a right area of the box and a front area of the box, are generated if the container space and the loading area exist.
3. Combine area
   If the height of the base point of new loading area is same as the height of the base point of an existing loading area, and the new area is adjacent to the existing area, they are combined into one area.
4. Sort area
   In order to determine the order by which the loading area is examined, the existing loading areas are rearranged.

Generally, lower and far position in the container is selected as the loading position at first. In this paper, therefore the loading areas are sorted into the far lower left order of the base point position. Namely, the loading priority is given in the order of a far position, a lower position and a left position.

This algorithm is denoted by SCA( $\sigma$, $\mu$, $\gamma$, $h$, $i$), where $h \in Q$, $i$ = 1,...,$M_h$ , and the arrangement result of SCA( $\sigma$, $\mu$, $\gamma$, $h$, $i$) is denoted by $R_h^i$.

$$R_h^i = \{p_k : p_k \in C_h^i, k = 1,..., T\} \tag{7}$$

The criterion of this result $R_h^i$ is represented by

$$F'(R_h^i) = \alpha \sum_{k=1}^{T} v'_k / V_h + \beta \sum_{k=1}^{T} v'_k c'_k \ (p_k \in C_h^i) \tag{8}$$

where $v'_k$ is the volume of the box $b'_k$, and $c'_k$ is the value of the box $b'_k$. This is a partial criterion value.

The different container part algorithm is to try to load the boxes to all kind of left containers so as to determine the type of containers that is used. Therefore the $q$ -th best arrangement $R^q$ is denoted by

$$F'(R^q) = \max_{h=1,\ldots,N} F'(R_l^{M'_{hq}+1}) \ (M'_{hq} = 0,\ldots,M_h - 1) \tag{9}$$

where $M'_{hq}$ is the number of required containers of type $h$ before he $q$ -th arrangement. The different container part algorithm DCA uses the following steps.

[Algorithm DCA]

Step 1: Set the sequence $\sigma$ = ($b_1$,..., $b_n$), the set of orientation orders $\mu = (r_1^h,\ldots,r_n^h)$ and the overhang parameter $\gamma$ for loading boxes;
set $h$ = 1 and $q$ =1 as index for the current container type and number of required containers.

Step 2: If $h \le N$ , then take the container of type $h \in Q$, else go to Step 5 .

Step 3: If the containers of type $h$ aren't remained, then set $h = h + 1$ and go to Step 2.

Step 4: Solve SCA( $\sigma$, $\mu$, $\gamma$, $h$, $M'_{lq}$ +1).
Set $h = h + 1$ and go to Step 2.

Step 5: Select the best result $R^q$ .
If all boxes are loaded, then set $q = q_{\max}$ and stop,
else if $q < \sum_{i=1}^{N} M_i$ set $h$ = 1, $q = q$ +1 and go to Step 2.
else stop.

This algorithm is denoted by DCA( $\sigma$, $\mu$, $\gamma$) and determines the container that loads the boxes one by one. This is also greedy algorithm and $q_{\max}$ is the number of required containers of this multiple container loading problem. The final arrangement result is represented by

$$X(\sigma,\mu,\gamma) = \{R^q : q = 1,..,q_{\max}\} \tag{10}$$

Therefore, the criterion of the result $X$ ( $\sigma$, $\mu$, $\gamma$) is $F$( $\sigma$, $\mu$, $\gamma$).

### 3.2 Multi-start local search

The multi-start local search procedure is used to obtain an optimal pair of the box sequence $\sigma$ and the set of orientation orders $\mu$. This procedure has two phases. First phase is decision of initial solutions using heuristics, and second phase is optimization of this solution using local search. Let *msn* be the number of initial solution and let *lsn* be the iteration number in local search. This procedure follows the original local search for each initial solution without any interaction among them and the random function is different at each local search.

At first phase, a good pair as an initial solution is selected. Therefore a heuristic rule that is related to decision of the box sequence $\sigma$ is prepared. Another solution $\mu$ is selected at random.

At second phase, an optimal pair is found by using local search. The neighborhood search is a fundamental concept of local search. For a given $\sigma$, the neighborhood is denoted by $N_1(\sigma)$. Here, for $\sigma = (b_1,...,b_n)$ ,

$$\mathrm{N}_1(\sigma) = \{\tau : \tau = (b_1,..,b_{k-1},b_l,b_{k+1},..,b_{l-1},b_k,b_{l+1},..,b_n), |l-k| \le nsb\} \tag{11}$$

for all combinations of $1 \le k,l \le n$ , $k \ne l$ and $nsb$ is a neighborhood size. That is, two numbers $b_k$ ,$b_l$ of a permutation $\sigma$ are taken, and they are exchanged. If $nsb = 1$ , this neighborhood only swap two adjacent numbers. For a given $\mu$, the neighborhood is denoted by $N_2(\mu)$. Here, for $\mu = (r_1^h,..., r_n^h)$,

$$\begin{aligned} &\mathrm{N}_2(\mu) = \{\upsilon : \upsilon = (r_1^h,...,r_i^h,...,r_n^h), h \in Q\} \\ &r_i^h = (u_{i1}^h,..,u_{i(k-1)}^h,u_{il}^h,u_{i(k+1)}^h,..,u_{i(l-1)}^h,u_{ik}^h,u_{i(l+1)}^h,..,u_{i6}^h) \end{aligned} \tag{12}$$

for all combinations of $1 \le i \le n$ , $1 \le h \le N$ , $1 \le k,l \le 6$ , $k \ne l$ . Thus, one order $r_i^h$ of a set of orders $\mu$ is taken; two numbers $u_{ik}^h, u_{il}^h$ of the selected order are taken, and they are exchanged. In this neighborhood, suppose that a neighborhood size $nsr$ is the number of the exchange operation.

The local search in the neighborhood consists of generating a fixed number of pair of solutions $\tau_1,...,\tau_{lsn}$ in $N_1(\sigma)$ and $\upsilon_1,...,\upsilon_{lsn}$ in $N_2(\mu)$, and finding the best pair of solution $\hat{\tau}$ and $\hat{\upsilon}$:

$$F(\hat{\tau},\hat{\upsilon},\gamma) = \max_{k=1,...,lsn} F(\tau_k,\upsilon_k,\gamma) \tag{13}$$

Assume that the pair of $j$ -th initial solutions are denoted by $\sigma_j$, $\mu_j$, $\gamma_j$, the optimal solution is found by

$$F(\hat{\sigma},\hat{\mu},\hat{\gamma}) = \max_{j=1,...,msn} F(\hat{\sigma}_j,\hat{\mu}_j,\gamma_j) \tag{14}$$

The method of multi-start local search MSLS are describes as follows.

[Algorithm MSLS]

Step 1: Set $j$ =1, $i$ =1.

Step 2: If $j \le msn$ , then set random seed $rs_j$ and decide the sequence $\sigma_j^i$, the set of orientation orders $\mu_j^i$ and the overhang parameter $\gamma_j$, else stop.

Step 3: Set $\sigma^* = \sigma_j^i$, $\mu^* = \mu_j^i$.

Step 4: If $i \le lsn$ , then solve DCA( $\sigma_j^i$, $\mu_j^i$, $\gamma_j$ ), else go to Step 7.

Step 5: If $F(\sigma^*, \mu^*, \gamma_j) < F(\sigma_j^i, \mu_j^i, \gamma_j)$ , then set $\sigma^* = \sigma_j^i$, $\mu^* = \mu_j^i$.

Step 6: Set $i = i + 1$and select $\sigma_j^i \in N_1(\sigma^*)$ and $\mu_j^i \in N_2(\mu^*)$.

Go back to Step4.

Step 7: If $j$ =1, then set $\hat{\sigma} = \sigma^*$ , $\hat{\mu} = \mu^*$ and $\hat{\gamma} = \gamma_j$;

else if $F(\hat{\sigma},\hat{\mu},\hat{\gamma}) < F(\sigma^*, \mu^*, \gamma_j)$, then $\hat{\sigma} = \sigma^*$, $\hat{\mu} = \mu^*$ and $\hat{\gamma} = \gamma_j$ .

Step 8: Set $j = j + 1$, $i$ =1 and go to Step 2.

In this algorithm, $\sigma_j^1$ and $\mu_j^1$ ( $j$ =1,...,$msn$) are initial solutions and optimal arrangement result is represented by $X(\hat{\sigma},\hat{\mu},\hat{\gamma})$. The $j$-th local search uses the random seed $rs_j$ to

determine the initial solution and the next solution in the neighborhood. The move strategy of this local search is first admissible move strategy as shown by this algorithm.

## 4. Computational experiments

### 4.1 Configuration

The tested method of multi-start local search procedure is shown below. At first phase, only one rule is required. The following rules are used to decide an initial solution of $\sigma_j^1$.

(R1) Sorting in decreasing order of the box volume $v_i$, $i \in P$.
(R2) Sorting in decreasing order of the box value $c_i$, $i \in P$.

These rules are used properly by problem type. If the object of the problem is to maximize the volume utilization, the rule R1 is used, and if the object of the problem is to maximize the total value of the loaded boxes, the rule R2 is used. Therefore the initial solution of box sequence is all same at any $j$, $j = 1,..., msn$.
Another initial solution of $\mu_j^1$, which is the order of orientation $r_i^h = (u_{i1}^h,...,u_{i6}^h)$ for each box of type $i$ are selected using a random function initialized random seed $rs_j$. The random function used here is linear congruential method.
The static parameter overhang ratio $\gamma$ is important factor as shown by Takahara (Takahara, 2006). However, $\gamma$ is fixed here for the simplification.
In order to show the effectiveness of the approach above described, the test cases from the literature were taken. Three kinds of test cases, that is the bin packing type multiple container loading problem with one container type, the bin packing type multiple container loading problem with different container type and the knapsack type multiple container loading problem, are taken.
As the bin packing type multiple container loading problem with one container type, the 47 examples of Ivancic et al. (Ivancic et al., 1989) are used here. The objective of the problems of Ivancic et al. is to find a minimal number of required containers load all given boxes. Each problem consists of two to five box types where only one container type is available. The problems of Ivancic et al. are denoted by IMM01 to IMM47.
As the bin packing type multiple container loading problem with different container type, the 17 examples of Ivancic et al. (Ivancic et al., 1989) are used here. The objective of these problems is to find the arrangement to maximize the volume utilization to load all given boxes. The problems of these are denoted by IMM2-01 to IMM2-17.
In these two bin packing type problems, the constant numbers are $\alpha = 1$ and $\beta = 0$ in the criterion (3) and the rule R1 is used to determine the initial solution of box sequence.
As the knapsack type multiple container loading problem, the 16 examples of Mohanty et al. (Mohanty et al., 1994) are used here. The objective of this problem is to maximize the total value of the loaded boxes. The problems are denoted by MMI01 to MMI16. In this knapsack type problem, the constant numbers are $\alpha = 0$ and $\beta = 1$ in the criterion (3) and the rule R2 is used to determine the initial solution of box sequence.
The algorithm was implemented in C using MS Visual C++ 6.0. The results of this approach were calculated on Xeon PC with a frequency 3.0GHz and 3GB memory.

### 4.2 Comparison with other methods

In order to show effectiveness of the approach above described, this approach MSLS has compared with other approaches. The following approaches were included the results of three kinds of test problems:

- IV_1989, a heuristic approach (Ivancic et al., 1989);
- MO_1994, a heuristic approach (Mohanty et al., 1994);
- B&R_1995, a heuristic approach (Bischoff & Ratcliff, 1995);
- BO_2000, a heuristic approach (Bortfeldt, 2000);
- EL_2002, a tree search approach (Eley, 2002);
- EL_2003, a bottleneck approach (Eley, 2003);
- TA_2006, a meta-heuristic approach (Takahara, 2006);

Table 1 presents the results for the 47 IMM problem classes. The parameters that were used in MSLS are $\gamma = 50$ , $msn = 5$ , $lsn = 500$ , $nsb = 1$ and $nsr = 3$ . The last column shows the total number of required containers. The results show that MSLS could be obtained the minimum number of total required containers than any other approach. But the result of the proposed approach is same as the result of the SA-Combined of TA_2006. This is because the greedy loading algorithm SCA is almost same as the loading algorithm that uses in TA_2006. However, the performance of computational time of MSLS has been improved from that of TA_2006 by 50%.

Table 2 shows the results for the 17 examples of three dimensional bin packing problem with different container types. The parameters that were used in MSLS are $\gamma = 50$ , $msn = 10$ , $lsn = 1000$ , $nsb = 1$ and $nsr = 3$ . The proposed approach obtained second highest average of volume utilization. For the test cases of IMM2-04 and IMM2-17, best arrangements ware found among these four methods. Fig.1 shows the best results found by the proposed approach.

Table 3 shows the results for the 16 MMI examples of knapsack type problem. The parameters that were used in MSLS are $\gamma = 50$ , $msn = 10$ , $lsn = 1000$ , $nsb = 2$ and $nsr = 6$ . The proposed approach obtained third highest average of volume utilization. For the test cases of MMI08 and MMI15, best arrangements ware found among these four methods. Fig.2 shows the best results found by the proposed approach.

### 4.3 Effect of neighborhood size

The neighborhood sizes, that is *nsb* and *nsr* , are key parameters of the proposed approach. *nsb* is the box sequence element ranges that can be exchanged. *nsr* is the number of iterations in which the orientation order elements are exchanged. In order to show the effect of these parameters, the following experiments have been done. The other parameters that were used here are $\gamma = 50$ , $msn = 10$ , $lsn = 1000$ . Table 4 shows the Ivancic et al. with different container types test problems results. The value in this table is volume utilization. If the neighborhood size *nsb* grows, the effect of the neighborhood size *nsr* becomes small. In the case of $nsr = 3$ or $nsr = 4$ , the better results are obtained. Table 5 shows Mohanty et al. test problems results. The value is the percentage of bounds. If the neighborhood size *nsb* becomes small, the effect of the neighborhood size *nsr* becomes small. In the case of $nsr = 5$ or $nsr = 6$ , the better results are obtained.

| | IV_1989 | B&R_1995 | BO_2000 | EL_2002 | EL_2003 | TA_2006 | MSLS |
|---|---|---|---|---|---|---|---|
| | Number of Containers | Number of Containers | Number of Containers | Number of Containers | Number of Containers | Number of Containers | Number of Containers |
| IMM01 | 26 | 27 | 25 | 26 | 25 | 25 | 25 |
| IMM02 | 11 | 11 | 10 | 10 | 10 | 10 | 10 |
| IMM03 | 20 | 21 | 20 | 22 | 20 | 20 | 20 |
| IMM04 | 27 | 29 | 28 | 30 | 26 | 26 | 26 |
| IMM05 | 65 | 61 | 51 | 51 | 51 | 51 | 51 |
| IMM06 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| IMM07 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| IMM08 | 5 | 4 | 4 | 4 | 4 | 4 | 4 |
| IMM09 | 19 | 19 | 19 | 19 | 19 | 19 | 19 |
| IMM10 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |
| IMM11 | 18 | 19 | 18 | 18 | 17 | 16 | 16 |
| IMM12 | 55 | 55 | 53 | 53 | 53 | 53 | 53 |
| IMM13 | 27 | 25 | 25 | 25 | 25 | 25 | 25 |
| IMM14 | 28 | 27 | 28 | 27 | 27 | 27 | 27 |
| IMM15 | 11 | 11 | 11 | 12 | 11 | 11 | 11 |
| IMM16 | 34 | 28 | 26 | 26 | 26 | 26 | 26 |
| IMM17 | 8 | 8 | 7 | 7 | 7 | 7 | 7 |
| IMM18 | 3 | 3 | 2 | 1 | 2 | 2 | 2 |
| IMM19 | 3 | 3 | 3 | 2 | 3 | 3 | 3 |
| IMM20 | 5 | 5 | 5 | 2 | 5 | 5 | 5 |
| IMM21 | 24 | 24 | 21 | 26 | 20 | 20 | 20 |
| IMM22 | 10 | 11 | 9 | 9 | 8 | 9 | 9 |
| IMM23 | 21 | 22 | 20 | 21 | 20 | 20 | 20 |
| IMM24 | 6 | 6 | 6 | 6 | 6 | 5 | 5 |
| IMM25 | 6 | 5 | 5 | 5 | 5 | 5 | 5 |
| IMM26 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| IMM27 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| IMM28 | 10 | 11 | 10 | 10 | 10 | 10 | 10 |
| IMM29 | 18 | 17 | 17 | 18 | 17 | 17 | 17 |
| IMM30 | 24 | 24 | 22 | 23 | 22 | 22 | 22 |
| IMM31 | 13 | 13 | 13 | 14 | 13 | 13 | 13 |
| IMM32 | 5 | 4 | 4 | 4 | 4 | 4 | 4 |
| IMM33 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| IMM34 | 9 | 9 | 8 | 9 | 8 | 8 | 8 |
| IMM35 | 3 | 3 | 2 | 2 | 2 | 2 | 2 |
| IMM36 | 18 | 19 | 14 | 14 | 14 | 14 | 14 |
| IMM37 | 26 | 27 | 23 | 23 | 23 | 23 | 23 |
| IMM38 | 50 | 56 | 45 | 45 | 45 | 45 | 45 |
| IMM39 | 16 | 16 | 15 | 15 | 15 | 15 | 15 |
| IMM40 | 9 | 10 | 9 | 9 | 8 | 9 | 9 |
| IMM41 | 16 | 16 | 15 | 15 | 15 | 15 | 15 |
| IMM42 | 4 | 5 | 4 | 4 | 4 | 4 | 4 |
| IMM43 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| IMM44 | 4 | 4 | 3 | 4 | 4 | 3 | 3 |
| IMM45 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| IMM46 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| IMM47 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
| Total | 763 | 763 | 705 | 716 | 699 | 698 | 698 |

Table 1. Results obtained for the problems from Ivancic et al. with one container type

| | IV_1989 | | BO_2000 | | EL_2003 | | MSLS | |
|---|---|---|---|---|---|---|---|---|
| | Volume utilization [%] | Number of containers for every type | Volume utilization [%] | Number of containers for every type | Volume utilization [%] | Number of containers for every type | Volume utilization [%] | Number of containers for every type |
| IMM2-01 | 71.8 | 26/0 | 74.7 | 25/0 | 74.7 | 25/0 | 74.7 | 25/0 |
| IMM2-02 | 87.3 | 2/5 | 87.3 | 2/5 | 88.5 | 1/7 | 88.5 | 1/7 |
| IMM2-03 | 74.3 | 1/8 | 92.2 | 2/0 | 92.2 | 2/0 | 92.2 | 2/0 |
| IMM2-04 | 84.1 | 1/14 | 89 | 0/15 | 89 | 0/15 | 89.5 | 2/11 |
| IMM2-05 | 97.6 | 7/13/6 | 95.1 | 1/19/11 | 99.9 | 2/13/17 | 99.7 | 7/15/1 |
| IMM2-06 | 97.6 | 4/6/1 | 99.7 | 4/1/2 | 99.7 | 7/4/0 | 99.6 | 7/1/0 |
| IMM2-07 | 85.8 | 10/1/7 | 86.8 | 16/0/2 | 87.4 | 2/0/14 | 87.1 | 9/0/8 |
| IMM2-08 | 95.8 | 3/0/26 | 97.9 | 7/0/23 | 99.4 | 3/0/25 | 98.7 | 1/1/25 |
| IMM2-09 | 92.2 | 7/6/1 | 96.6 | 8/5/0 | 96.6 | 6/9/0 | 96.6 | 7/7/0 |
| IMM2-10 | 90.6 | 1/0/2 | 90.6 | 1/0/2 | 90.6 | 1/0/2 | 90.6 | 1/0/2 |
| IMM2-11 | 81.2 | 3/3/11 | 85.9 | 0/4/10 | 88.4 | 9/2/5 | 87.7 | 1/6/4 |
| IMM2-12 | 75 | 5/1/0 | 90.2 | 2/1/1 | 93.5 | 3/0/1 | 92.7 | 5/0/0 |
| IMM2-13 | 85.3 | 9/1/5 | 87.8 | 14/1/1 | 86.3 | 13/1/2 | 85 | 5/11/2 |
| IMM2-14 | 88.7 | 0/2/4 | 94 | 2/1/1 | 94 | 2/1/1 | 94 | 2/1/1 |
| IMM2-15 | 76.3 | 3/2/11 | 79.1 | 3/3/10 | 79.2 | 2/3/11 | 78.7 | 3/1/11 |
| IMM2-16 | 82.7 | 4/0/0 | 91.6 | 2/1/0 | 91.6 | 2/1/0 | 82.9 | 1/1/1 |
| IMM2-17 | 77.1 | 26/0 | 84.7 | 0/0/3 | 84.7 | 0/0/3 | 91.6 | 0/1/1 |
| Average | 84.9 | | 89.6 | | 90.3 | | 90.0 | |

Table 2. Results obtained for the problems from Ivancic et al. with different container types

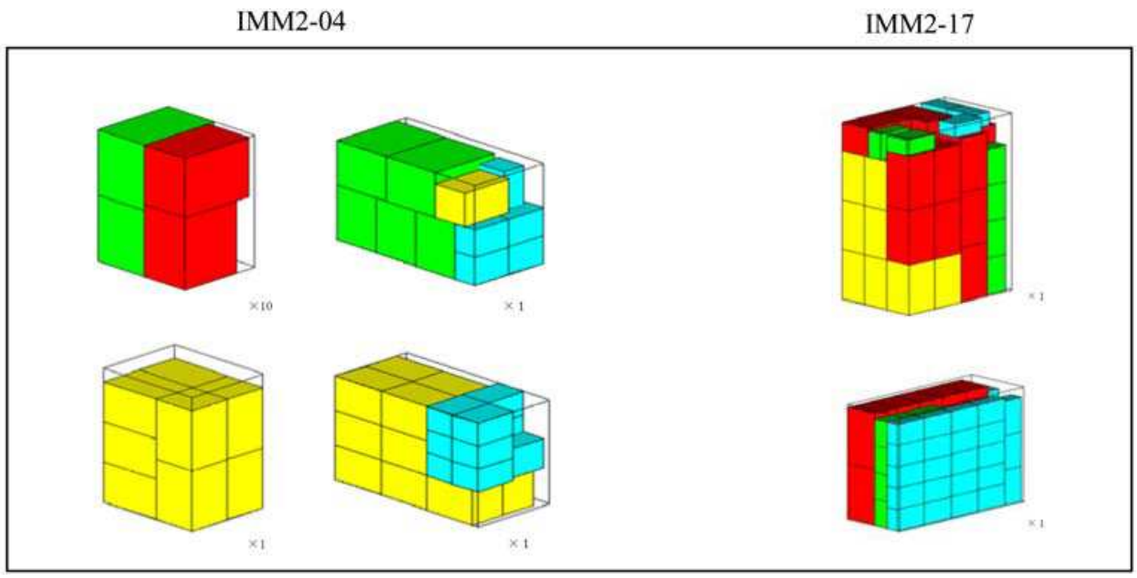



Fig. 1. Layout results of Ivancic et al. with different container types test problems

| | Bound | MO_1994 | | BO_2000 | | EL_2003 | | MSLS | |
|---|---|---|---|---|---|---|---|---|---|
| | | Absolute | In % of bound | Absolute | In % of bound | Absolute | In % of bound | Absolute | In % of bound |
| MMI01 | 11112.0 | 8640.0 | 77.7 | 8640.0 | 77.7 | 8640.0 | 77.7 | 8640.0 | 77.7 |
| MMI02 | 86016.0 | 83494.4 | 97.1 | 85120.0 | 99.0 | 85376.0 | 99.3 | 84224.0 | 97.9 |
| MMI03 | 53500.0 | 53262.5 | 99.6 | 53262.5 | 99.6 | 53262.5 | 99.6 | 52350.0 | 97.9 |
| MMI04 | 2720640.0 | 2333440.0 | 85.8 | 2333440.0 | 85.8 | 2307840.0 | 84.8 | 23334400.0 | 85.8 |
| MMI05 | 653750.0 | 495500.0 | 75.8 | 581250.0 | 88.9 | 583750.0 | 89.3 | 579250.0 | 88.6 |
| MMI06 | 143424.0 | 138240.0 | 96.4 | 139584.0 | 97.3 | 141216.0 | 98.5 | 137952.0 | 96.2 |
| MMI07 | 20203.2 | 16668.0 | 82.5 | 17409.0 | 86.2 | 17004.0 | 84.2 | 17262.0 | 85.4 |
| MMI08 | 77986.8 | 65741.0 | 84.3 | 68645.6 | 88.0 | 69121.2 | 88.6 | 69747.2 | 89.4 |
| MMI09 | 139356.0 | 119772.0 | 85.9 | 128952.0 | 92.5 | 133632.0 | 95.9 | 128556.0 | 92.3 |
| MMI10 | 15360.0 | 15360.0 | 100.0 | 15360.0 | 100.0 | 15360.0 | 100.0 | 15360.0 | 100.0 |
| IMMI11 | 68353.2 | 49995.0 | 73.1 | 53202.8 | 77.8 | 52873.6 | 77.4 | 53202.8 | 77.8 |
| MMI12 | 24964.0 | 23529.0 | 94.3 | 24235.2 | 97.1 | 23673.0 | 94.8 | 23990.4 | 96.1 |
| MMI13 | 36556.8 | 36556.8 | 100.0 | 36556.8 | 100.0 | 36556.8 | 100.0 | 36556.8 | 100.0 |
| MMI14 | 71552.0 | 56492.8 | 78.9 | 65316.8 | 91.3 | 68723.2 | 96.0 | 68723.2 | 96.0 |
| MMI15 | 42922.8 | 37558.8 | 87.5 | 39727.2 | 92.6 | 39382.2 | 91.8 | 40590.0 | 94.6 |
| MMI16 | 666829.6 | 556458.0 | 83.5 | 595770.0 | 89.3 | 591535.0 | 88.7 | 571290.0 | 85.7 |
| Average | | | 87.7 | | 91.4 | | 91.7 | | 91.3 |

Table 3. Results obtained for the problems from Mohanty et al.

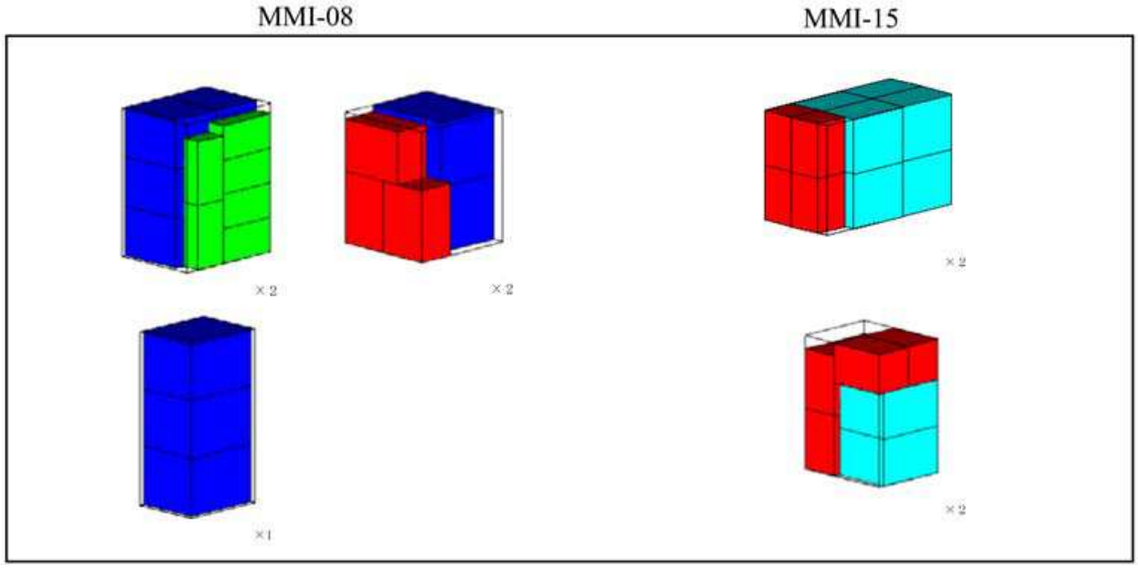



Fig. 2. Layout results of Mohanty et al. test problems

| | nsb = 1 nsr = 2 | nsb = 1 nsr = 4 | nsb = 1 nsr = 8 | nsb = 2 nsr = 2 | nsb = 2 nsr = 4 | nsb = 2 nsr = 8 | nsb = 3 nsr = 2 | nsb = 3 nsr = 4 | nsb = 3 nsr = 8 |
|---|---|---|---|---|---|---|---|---|---|
| IMM2-01 | 74.67 | 74.67 | 74.67 | 74.67 | 74.67 | 74.67 | 74.67 | 74.67 | 74.67 |
| IMM2-02 | 87.34 | 87.34 | 87.34 | 86.18 | 86.18 | 87.34 | 86.18 | 87.16 | 87.34 |
| IMM2-03 | 92.16 | 92.16 | 92.16 | 92.16 | 92.16 | 92.16 | 92.16 | 92.16 | 92.16 |
| IMM2-04 | 89.48 | 89.48 | 89.48 | 89.72 | 89.72 | 89.48 | 89.72 | 89.72 | 84.07 |
| IMM2-05 | 99.73 | 99.73 | 99.73 | 99.73 | 99.73 | 99.73 | 99.73 | 99.73 | 99.73 |
| IMM2-06 | 99.65 | 99.65 | 99.65 | 99.65 | 99.65 | 99.65 | 99.65 | 99.65 | 99.65 |
| IMM2-07 | 87.09 | 87.09 | 87.09 | 87.09 | 87.09 | 87.09 | 87.09 | 87.09 | 87.09 |
| IMM2-08 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 | 98.68 |
| IMM2-09 | 96.63 | 96.63 | 93.30 | 96.63 | 96.06 | 96.63 | 96.63 | 96.06 | 96.63 |
| IMM2-10 | 90.57 | 90.57 | 90.57 | 90.57 | 90.57 | 90.57 | 90.57 | 90.57 | 90.57 |
| IMM2-11 | 86.35 | 87.73 | 88.35 | 86.35 | 87.73 | 86.35 | 86.28 | 86.35 | 87.73 |
| IMM2-12 | 90.98 | 90.98 | 90.98 | 92.73 | 92.73 | 90.98 | 90.23 | 90.98 | 92.73 |
| IMM2-13 | 85.96 | 86.29 | 85.84 | 86.29 | 85.02 | 85.88 | 85.84 | 85.88 | 86.29 |
| IMM2-14 | 94.03 | 94.03 | 94.03 | 94.03 | 94.03 | 94.03 | 94.03 | 94.03 | 94.03 |
| IMM2-15 | 78.75 | 78.75 | 78.75 | 78.75 | 78.75 | 78.75 | 78.75 | 78.75 | 78.75 |
| IMM2-16 | 82.87 | 82.87 | 82.87 | 82.87 | 82.87 | 82.87 | 82.87 | 82.87 | 82.87 |
| IMM2-17 | 91.59 | 91.59 | 91.59 | 91.59 | 91.59 | 91.59 | 91.59 | 91.59 | 91.59 |
| Average | 89.79 | 89.89 | 89.71 | 89.86 | 89.83 | 89.79 | 89.68 | 89.70 | 89.68 |

Table 4. Ivancic et al. with different container types test problems results for different neighborhood size

| | nsb = 1 nsr = 2 | nsb = 1 nsr = 6 | nsb = 1 nsr = 8 | nsb = 2 nsr = 2 | nsb = 2 nsr = 6 | nsb = 2 nsr = 8 | nsb = 3 nsr = 2 | nsb = 3 nsr = 6 | nsb = 3 nsr = 8 |
|---|---|---|---|---|---|---|---|---|---|
| MMI01 | 77.75 | 77.75 | 77.75 | 77.75 | 77.75 | 77.75 | 77.75 | 77.75 | 77.75 |
| MMI02 | 97.92 | 97.92 | 97.92 | 97.92 | 97.92 | 97.92 | 97.92 | 97.92 | 97.92 |
| MMI03 | 97.85 | 97.85 | 97.85 | 97.85 | 97.85 | 97.85 | 97.85 | 97.85 | 97.85 |
| MMI04 | 85.77 | 85.77 | 85.77 | 85.77 | 85.77 | 85.77 | 85.77 | 85.77 | 85.77 |
| MMI05 | 87.23 | 88.60 | 86.77 | 87.23 | 88.60 | 88.60 | 87.23 | 88.60 | 88.60 |
| MMI06 | 96.18 | 96.52 | 96.85 | 96.18 | 96.18 | 96.18 | 96.18 | 96.18 | 96.18 |
| MMI07 | 85.44 | 85.44 | 85.44 | 85.44 | 85.44 | 85.44 | 85.44 | 85.44 | 85.44 |
| MMI08 | 89.43 | 89.43 | 89.43 | 89.43 | 89.43 | 89.43 | 89.43 | 89.43 | 89.43 |
| MMI09 | 92.59 | 91.91 | 92.25 | 91.91 | 92.25 | 92.10 | 92.59 | 91.91 | 92.25 |
| MMI10 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| MMI11 | 77.84 | 77.84 | 77.84 | 77.84 | 77.84 | 77.84 | 77.84 | 77.84 | 77.84 |
| MMI12 | 93.20 | 93.62 | 94.89 | 92.64 | 96.10 | 93.91 | 92.64 | 96.10 | 93.91 |
| MMI13 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| MMI14 | 96.05 | 96.05 | 94.60 | 96.05 | 96.05 | 96.05 | 96.05 | 96.05 | 96.05 |
| MMI15 | 94.57 | 94.57 | 94.57 | 94.57 | 94.57 | 94.57 | 94.57 | 94.57 | 94.57 |
| MMI16 | 85.67 | 85.67 | 85.67 | 85.67 | 85.67 | 85.67 | 85.67 | 85.67 | 85.67 |
| Average | 91.09 | 91.18 | 91.10 | 91.02 | 91.34 | 91.19 | 91.06 | 91.32 | 91.20 |

Table 5. Mohanty et al. test problems results for different neighborhood size

## 5. Conclusion

The presented multi-start local search approach for the multiple container loading problem is suitable for solving various kind of problem, because the proposed approach is based on greedy loading algorithm and hardly uses problem-specific operators and heuristic rules. Hence it is easy to improve and manage by users. Its good performance and superiority compared with the other approaches were shown in the test results.

In this paper, only the weakly heterogeneous problems are taken. Thus, further studies include integration of greedy loading algorithm and multi-start approach, dealing with the strongly heterogeneous problems, and development of an efficient container loading software.

## 6. References

Bischoff, E. E. & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading, *Omega*, 23, pp.377-390.

Bortfeldt, A. (2000). Eine heuristik für multiple containerladeprobleme, *OR Spektrum*, 22 pp.239-262.

Eley, M. (2002). Solving container loading problems by block arrangement, *EJOR*, 141, pp.393-402.

Eley, M. (2003). A bottleneck assignment approach to the multiple container loading problem, *OR Spectrum*, 25, pp.45-60.

George, J. A. & D. F. Robinson, D. F. (1980). A heuristic for packing boxes into a container, *Computers Opns Res.*, 7, pp.147-156.

George, J. A. (1996). Multiple container packing: a case study of pipe packing, *Journal of the Operational Research Society*, 47, pp.1098-1109.

Gilmore, P. C. & Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions, *Opns Res.*, 13, pp.94-120.

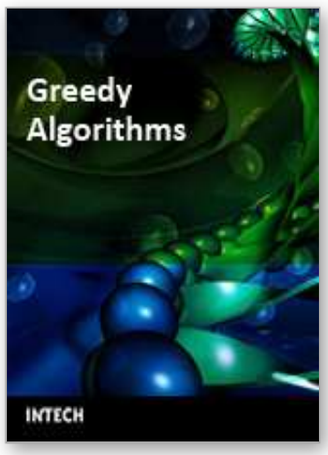
Ivancic, N.J., Mathur, K. & Mohanty, B.B. (1989). An integer-programming based heuristic approach to the three dimensional packing problem, *Journal of Manufacturing and Operation Management*, 2, pp.268-298.

Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R. & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM J. Comput.*, 3, pp.299-325.

Mohanty, B. B., Mathur, K. & Ivancic, N.J. (1994). Value considerations in three-dimensional packing – A heuristic procedure using the fractional knapsack problem, *EJOR*, 74, pp.143-151.

Takahara, S. (2005). Loading problem in multiple containers and pallets using strategic search method, *Lecture Notes in Artificial Intelligence 3558*, pp.448-456.

Takahara, S. (2006). A simple meta-heuristic approach for the multiple container loading problem, *Proceedings of 2006 IEEE International Conference on Systems, Man and Cybernetics*, pp.2328-2333, Taipei, Taiwan.




**Greedy Algorithms**

Edited by Witold Bednorz

ISBN 978-953-7619-27-5
Hard cover, 586 pages
**Publisher** InTech
**Published online** 01, November, 2008
**Published in print edition** November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Shigeyuki Takahara (2008). A Multi-start Local Search Approach to the Multiple Container Loading Problem, Greedy Algorithms, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech, Available from: http://www.intechopen.com/books/greedy_algorithms/a_multi-start_local_search_approach_to_the_multiple_container_loading_problem

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen