Нижние оценки Гилмора и Гомори

Имеется неограниченное число контейнеров единичной вместимости.

Для каждой заготовки $i \in L$ задана длина $0 < w_i < 1$ и их количество $n_i \geq 1$.

Требуется упаковать заготовки в минимальное число контейнеров.

Рассмотрим различные варианты упаковки одного контейнера.

Пусть матрица (a_{ij}) задает число заготовок i-го типа в j-м варианте.

Переменные задачи:

$x_j \geq 0$, целые — число контейнеров, упакованных по j-му варианту.

$$\min \sum_{j \in J} x_j$$

$$\text{при условиях: } \sum_{j \in J} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \text{ целые}, \; j \in J$$

Нижняя оценка

$$H = \min \sum_{j \in J} x_j$$

$$\sum_{j \in J} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \qquad j \in J.$$

Решая задачу линейного программирования, получаем нижнюю оценку H.

Но задача имеет гигантскую размерность!

Множество J имеет экспоненциальную мощность.

Метод генерации столбцов

Выберем подмножество $J' \subset J$ так, чтобы следующая подзадача ЛП(J') имела решение:

$$\min \sum_{j \in J'} x_j$$

$$\sum_{j \in J'} a_{ij} x_j \ge n_i, \quad i \in L;$$

$$x_j \ge 0, \qquad j \in J'.$$

Это легко сделать с помощью любой жадной эвристики: *NF*, *FF*, *BF*,…

Пусть x_j^* — оптимальное решение для J' и $\lambda_i^* \geq 0$ — оптимальное решение соответствующей двойственной задачи. Рассмотрим двойственную задачу для исходного множества J:

$$\max \sum_{i\in L} n_i \lambda_i$$

при ограничениях
$$\sum_{i\in L} a_{ij}\lambda_i \leq 1, \quad j \in J;$$

$$\lambda_i \geq 0, \qquad i \in L.$$

Если для всех $j \in J$ справедливо

$$\sum_{i\in L} a_{ij}\lambda_i^* \leq 1 \qquad (*)$$

то вектор $\bar{x}_j = \begin{cases} x_j^*, & j \in J' \\ 0, & j \in J \setminus J' \end{cases}$ — оптимальное решение задачи линейного программирования для всего множества J и $H = \sum_{j\in J'} x_j^*$.

Проблема: как проверить (∗), не просматривая все множество J, и что делать, если условие не выполняется.

Рассмотрим задачу о рюкзаке:

$$\alpha = \max \sum_{i \in L} \lambda_i^* y_i$$

при ограничениях: $$\sum_{i \in I} w_i y_i \le 1 \qquad \text{(вместимость контейнеров)}$$

$$y_i \ge 0, \text{ целые, } i \in L.$$

Оптимальное решение этой задачи дает нам новый вариант упаковки контейнера.

Если $\alpha \le 1$, то (∗) выполнено!

Если $\alpha > 1$, то получили вариант упаковки, который следует добавить в множество J' (нашли ведущий столбец в симплекс–таблице).

Общая схема метода:

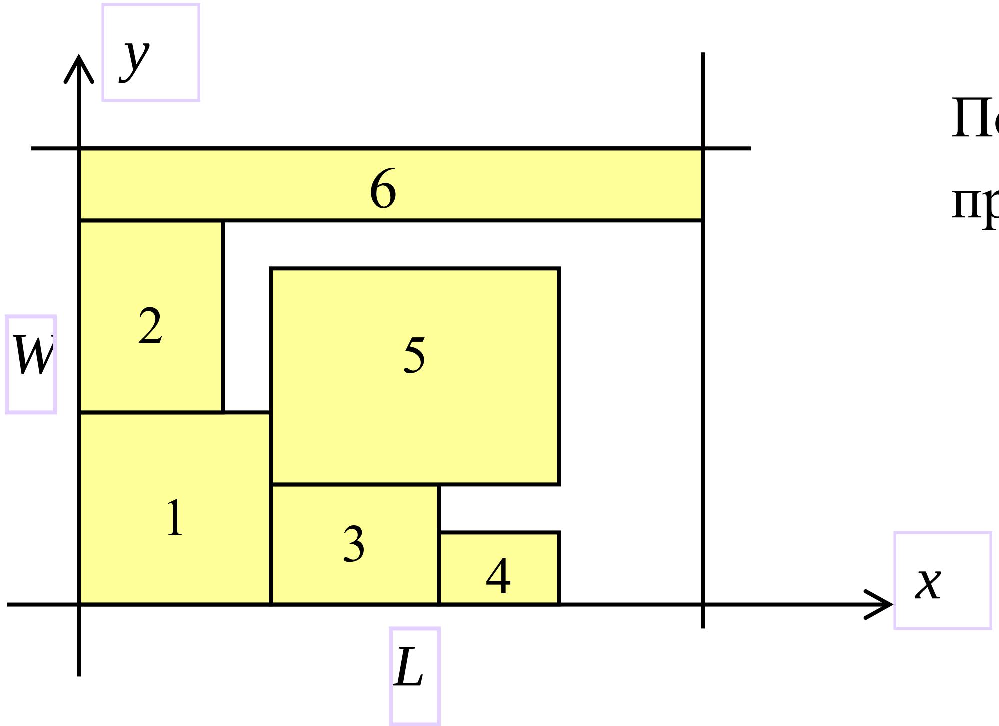
1. Выбрать подмножество $J' \subset J$
2. Решить задачу ЛП(J'), получить x_j^*, λ_i^*.
3. Решить задачу о рюкзаке, получить α.
4. Если $\alpha \le 1$, то $H = \sum_{j \in J'} x_j^*$, STOP.
5. Добавить в J' новый вариант упаковки и вернуться на шаг 2

Оценка $H = \sum_{j \in J'} x_j^*$ является трудоемкой, но доминирует другие по точности. Решение $x_j = \lceil x_j^* \rceil, j \in J$, часто оказывается оптимальным.

Задачи двумерной упаковки

Дано: n прямоугольных предметов с размерами $w_i \times l_i$, $i = 1,\ldots, n$.

Найти: упаковку в положительном ортанте с минимальной площадью окаймляющего прямоугольника



Повороты запрещены. Стороны предметов параллельны осям координат.

$$L \times M \to \min$$

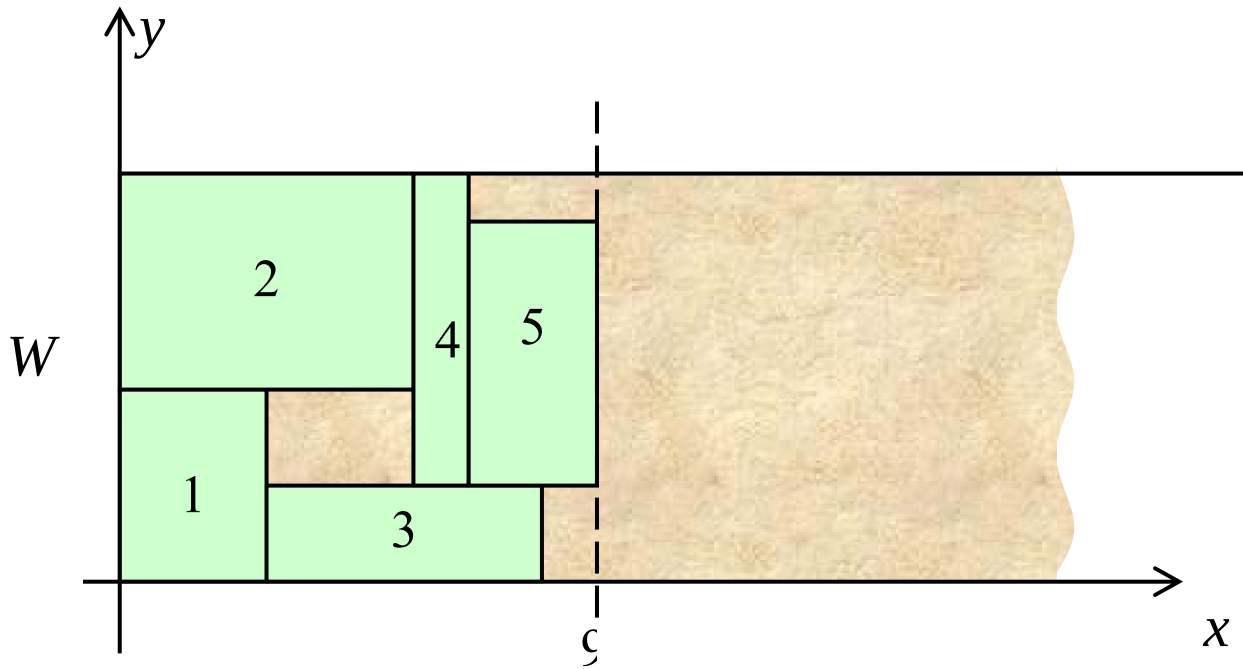
Задача является NP-трудной.

Пример гильотинной упаковки

Задача упаковки в полосу

Дано: n прямоугольных предметов с размерами $w_i \times l_i$, $i = 1,\ldots, n$, и полубесконечная полоса шириной W.

Найти: упаковку предметов с минимальной длиной занимаемой полосы.

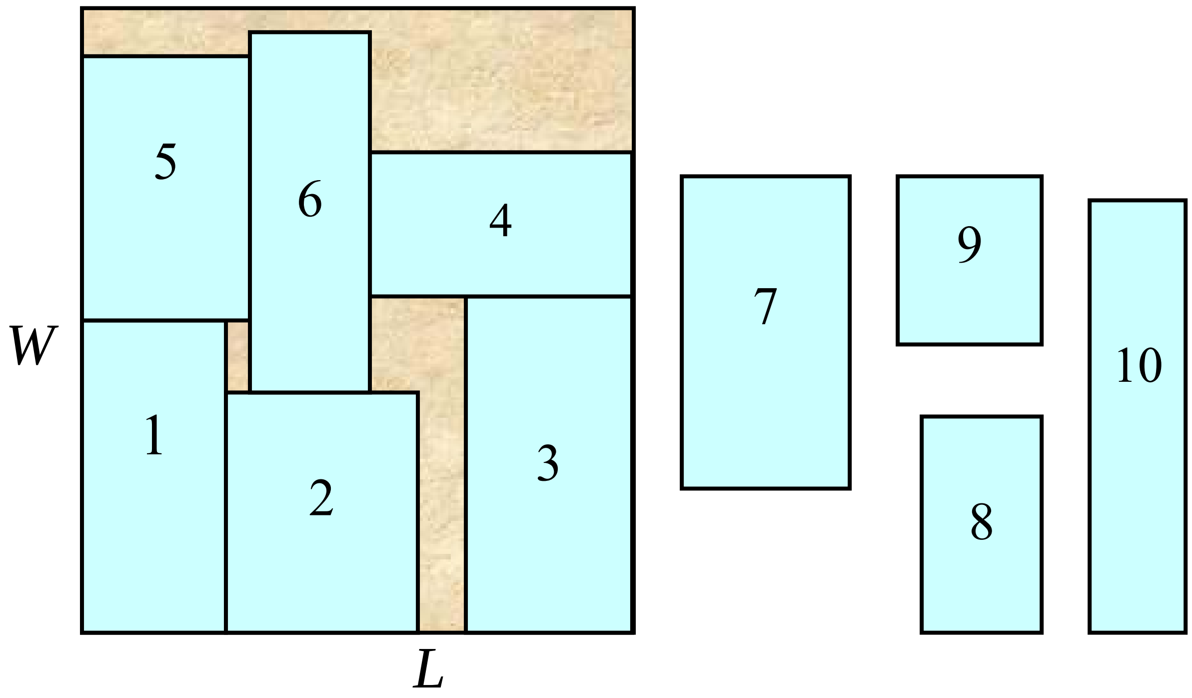


При $l_i=1$, $i = 1,\ldots, n$, получаем одномерную задачу упаковки в контейнеры.

Задача о рюкзаке для прямоугольников

Дано: n прямоугольных предметов с размерами $w_i \times l_i$ и весами c_i, $i = 1,\ldots, n$, и большой прямоугольник $W \times L$.

Найти: подмножество предметов максимального веса, которые можно вырезать из большого прямоугольника.



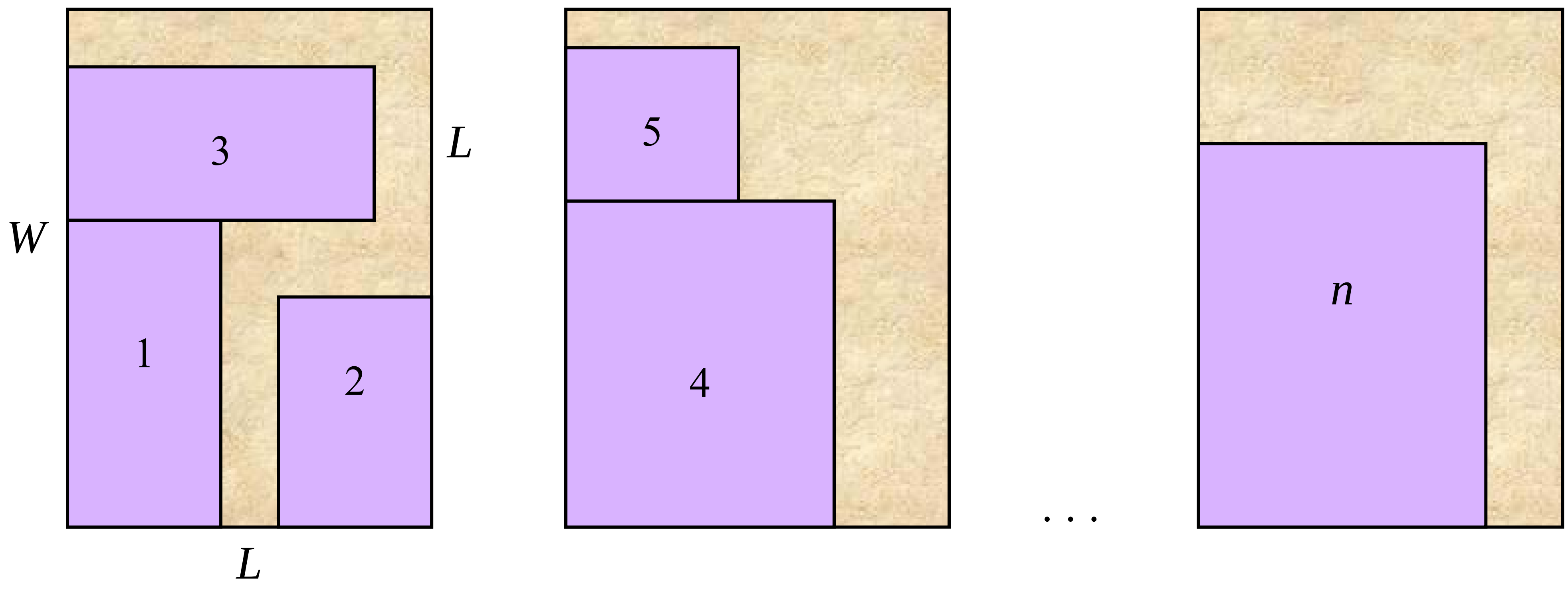
При $l_i = L$, $i = 1,\ldots, n$,

получаем известную задачу о рюкзаке.

Задача упаковки в контейнеры для прямоугольников

Дано: n прямоугольных предметов с размерами $w_i \times l_i$, $i = 1,\ldots, n$, и неограниченное число одинаковых заготовок размером $W \times L$.

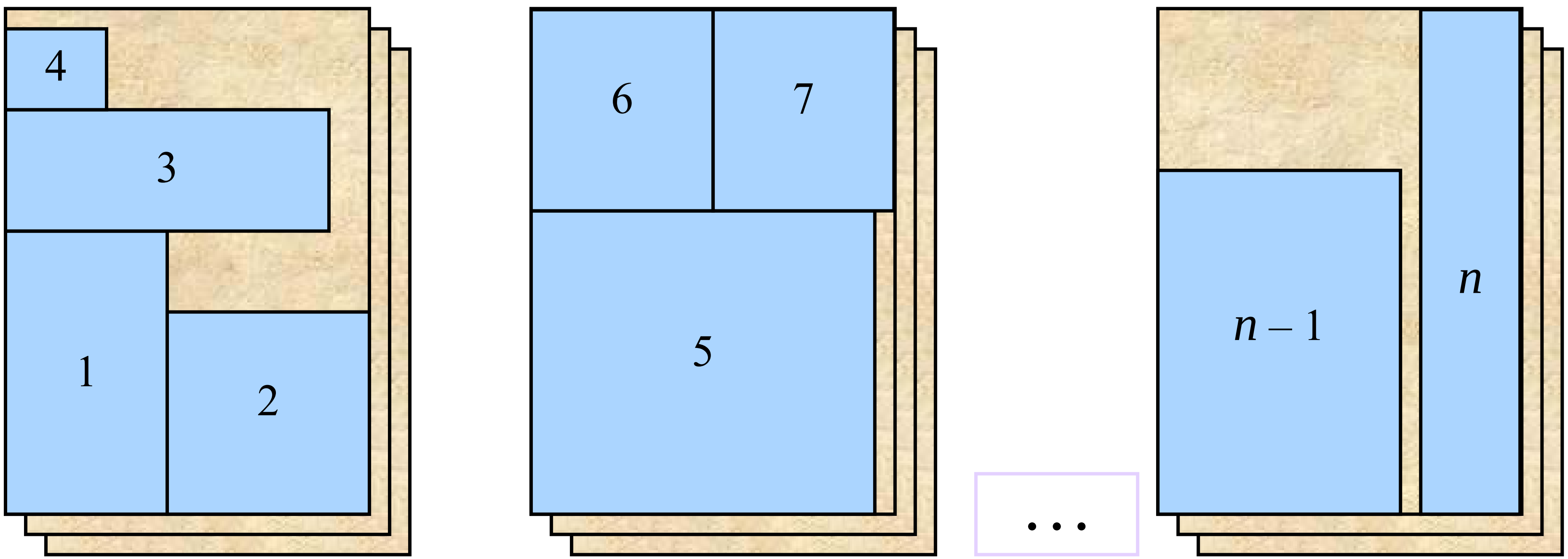
Найти: упаковку предметов с минимальным числом используемых заготовок.



Задача прямоугольного раскроя для серийного производства

Дано: n прямоугольных предметов с размерами $w_i \times l_i$ и потребностью k_i штук $i = 1,\ldots, n$. Известны размеры заготовок $W \times L$.

Найти: план раскроя заготовок, позволяющий получить все предметы в требуемых количествах и использующий минимальное число заготовок.

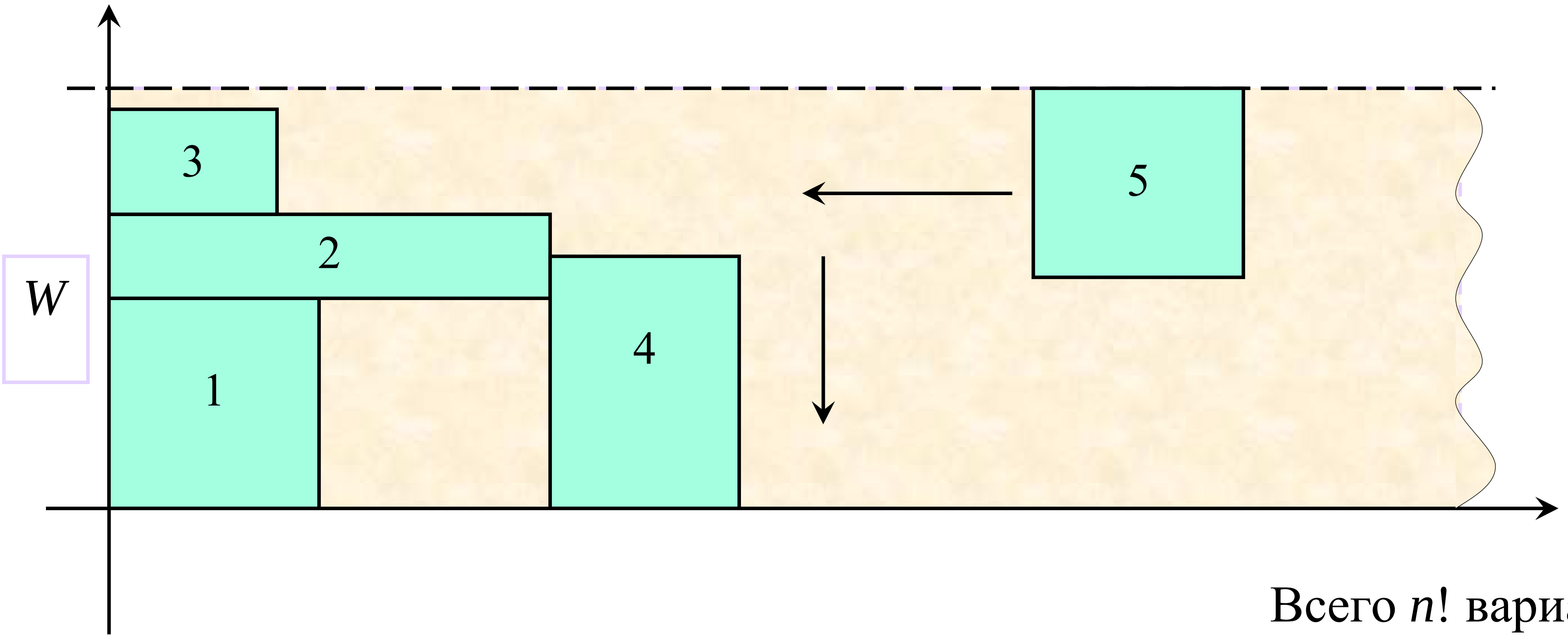


Кодировки решений

Список предметов $L = \{i_1, i_2, \dots , i_n\}$.

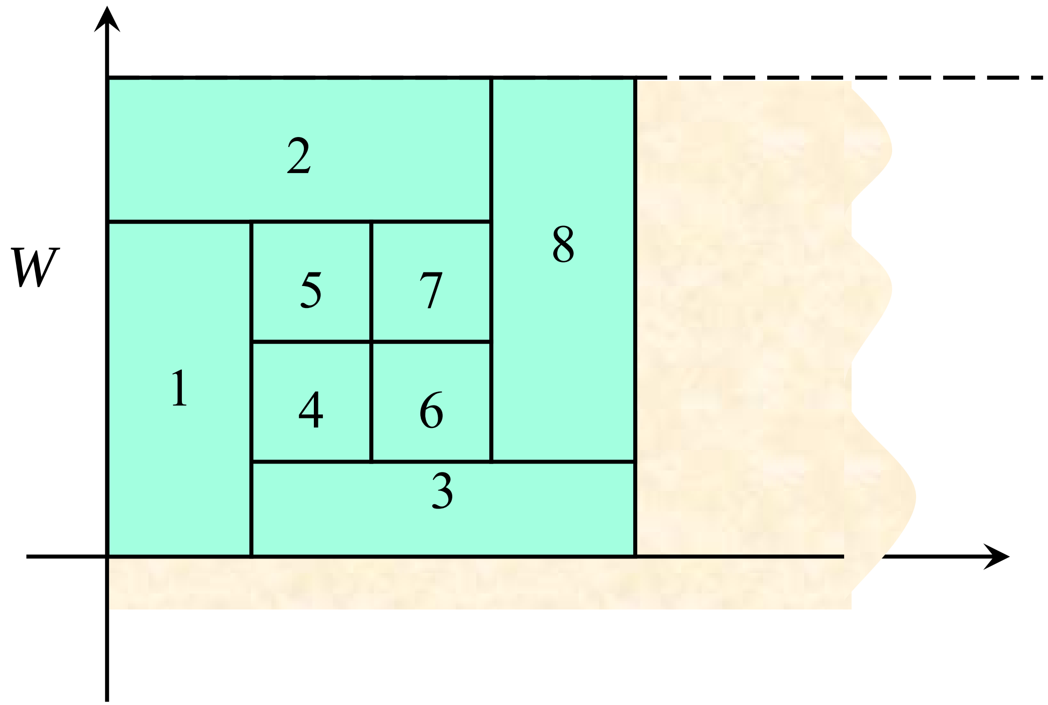
Декодирующая процедура: до упора влево, до упора вниз, до упора влево, …

Если нельзя сдвинуть предмет влево или вправо, то переходим к следующему предмету.



Всего $n!$ вариантов.

Контрпример



Оптимальное решение, которое нельзя получить никаким списком.

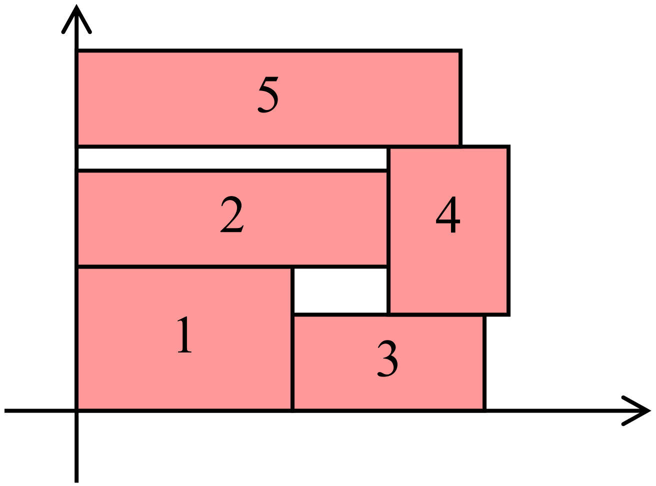
Определение. Кодировка решений называется **допустимой**, если она удовлетворяет следующим требованиям:

1. множество всех кодов конечно;
2. каждому коду соответствует допустимое решение;
3. вычисление целевой функции для каждого кода осуществляется с полиномиальной трудоемкостью;
4. решение, соответствующее коду с наименьшим значением целевой функции, является оптимальным решением исходной задачи.

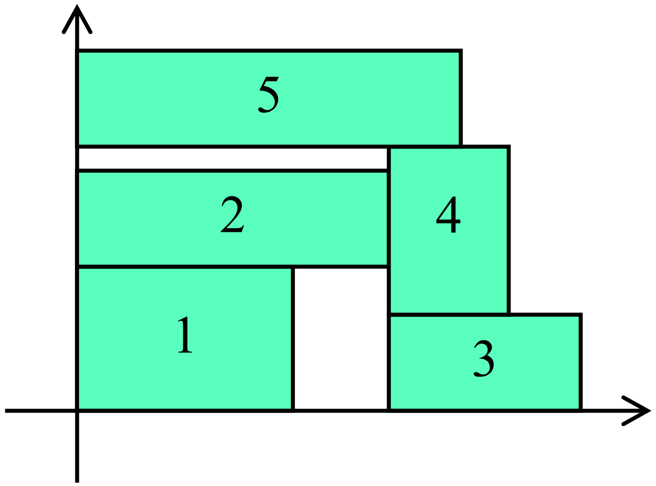
Пример. Каждому предмету сопоставляется два числа (x, y) — координаты левого нижнего угла.

Кодировка «O – tree»

Определение. Решение называется ***L*–компактным,** если ни один предмет нельзя сместить влево при условии, что остальные предметы остаются неподвижными. Аналогично определяются *B*–, *V*– и *R*–компактные решения для смещения вниз, вверх и вправо соответственно. Решение называется ***LB*–компактным**, если оно является *L*–компактным и *B*–компактным одновременно.



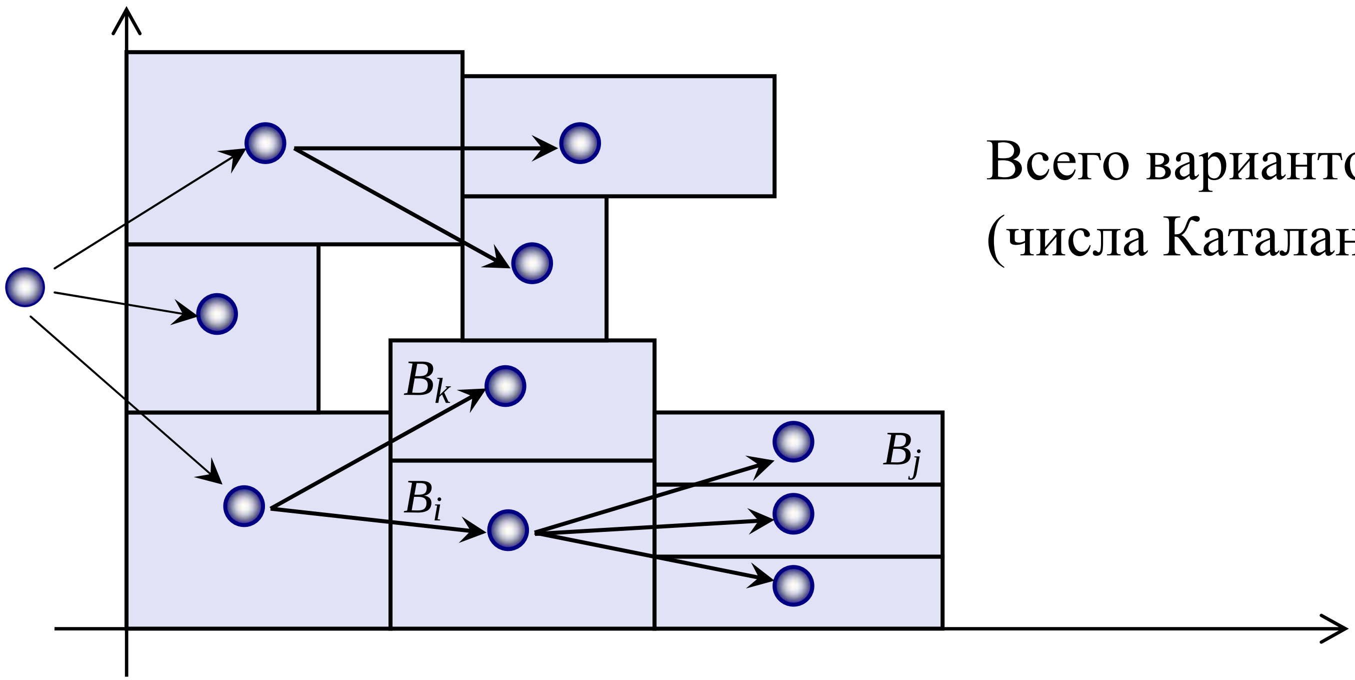
LB–компактное



B–, но не *L*–компактное

Представление *LB*–компактных решений

Корень дерева — левая граница упаковки. Ориентация дуг от корня к листьям. Предмет B_i связан дугой с B_j, если левая сторона B_j касается правой стороны B_i. Если для B_j имеется несколько таких предметов (B_i, B_k), то дуга идет только от нижнего предмета.

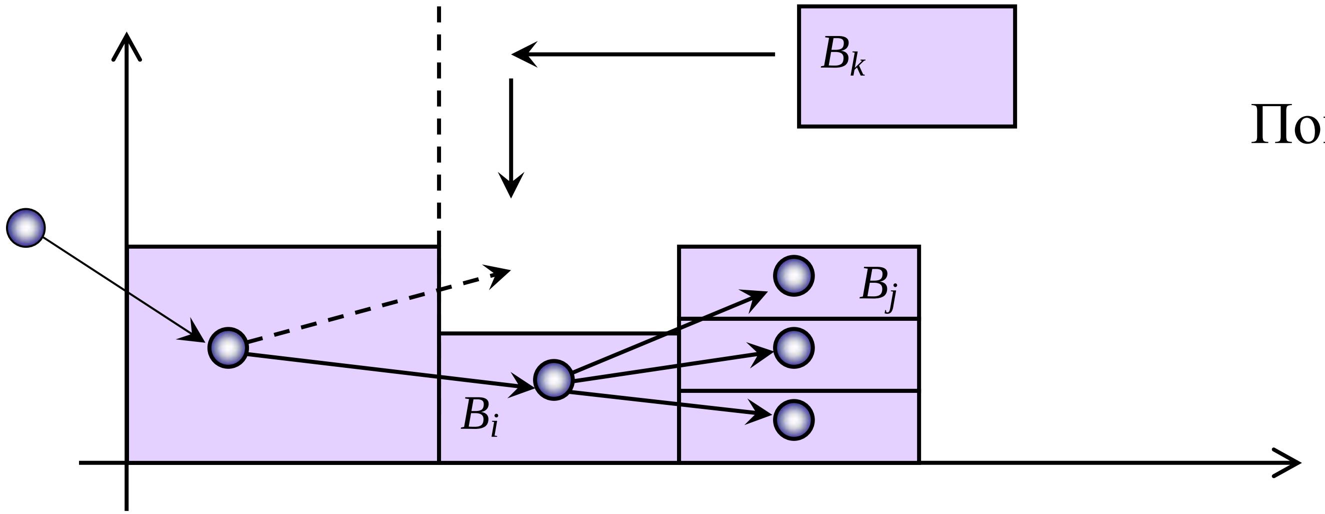


Всего вариантов $n!\cdot C_{2n+1}^{n}/(2n+1)$
(числа Каталана)

Процедура декодирования

Дано: ориентированное корневое дерево, каждой вершине кроме корня приписан предмет (метка).

Найти: площадь окаймляющего прямоугольника.



Поиск в глубину

Упражнение. Найти алгоритм декодирования с T = $O(n)$

Соседние упаковки

Для данного корневого ориентированного дерева с помеченными вершинами определим две операции:

- две некорневые вершины меняются метками (предметами);
- отрываем лист дерева и приклеиваем к другой вершине.

Каждая операция порождает новую упаковку. Таких упаковок $O(n^2)$.

Назовем их **соседними** для данной.

Множество соседних упаковок называется **окрестностью.**

Алгоритм локального спуска

S — упаковка

$N(S)$ — окрестность для S.

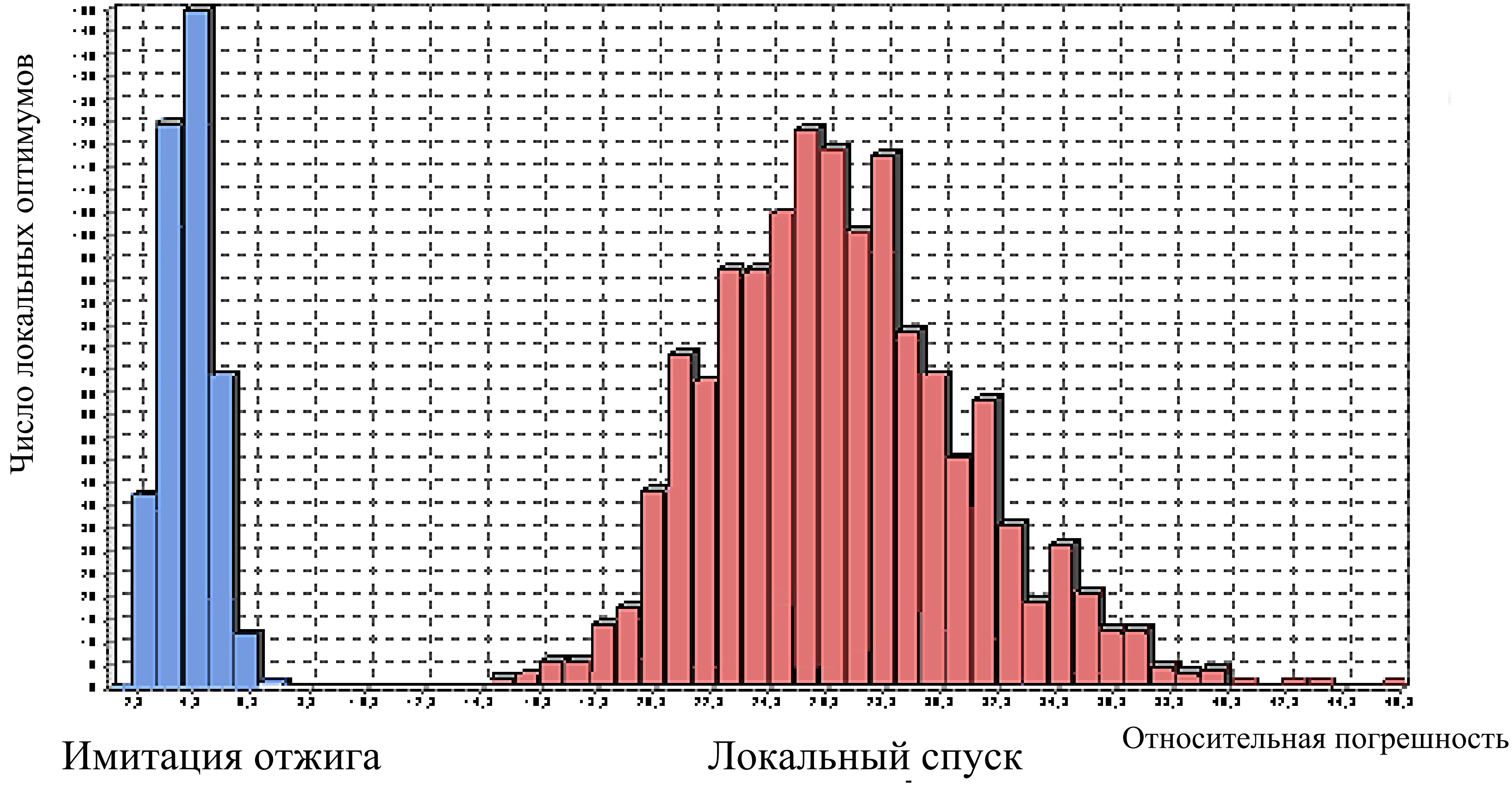
$F(S)$ — значение целевой функции для S.

Алгоритм

1. Выбрать начальное решение S и вычислить $F(S)$.
2. Найти в окрестности $N(S)$ решение S' с минимальным значением целевой функции $F(S') = \min\{F(\bar{S}),\ \bar{S} \in N(S)\}$.
3. Если $F(S') < F(S)$, то положить $S := S'$ и вернуться на 2

   иначе STOP.

Численные эксперименты



Задача упаковки в положительном ортанте, $n = 100$.

Пороговые алгоритмы

Алгоритм имитации отжига относится к классу пороговых алгоритмов.

На каждом шаге в окрестности текущего решения i_k выбирается некоторое решение j, и если разность по целевой функции между новым и текущим решением не превосходит заданного порога t_k, то новое решение j заменяет текущее. В противном случае выбирается новое соседнее решение.

Пороговый алгоритм

1. Выбрать начальное решение $i_0 \in I$ и положить $f^* = f(i_0)$, $k = 0$.
2. Пока не выполнен критерий остановки, делать следующее:
   2.1. Случайно выбрать $j \in N(i_k)$.
   2.2. Если $f(j) - f(i_k) < t_k$ то $i_{k+1} := j$.
   2.3. Если $f^* > f(i_k)$, то $f^* := f(i_k)$.
   2.4. Положить $k := k + 1$.

Типы пороговых алгоритмов

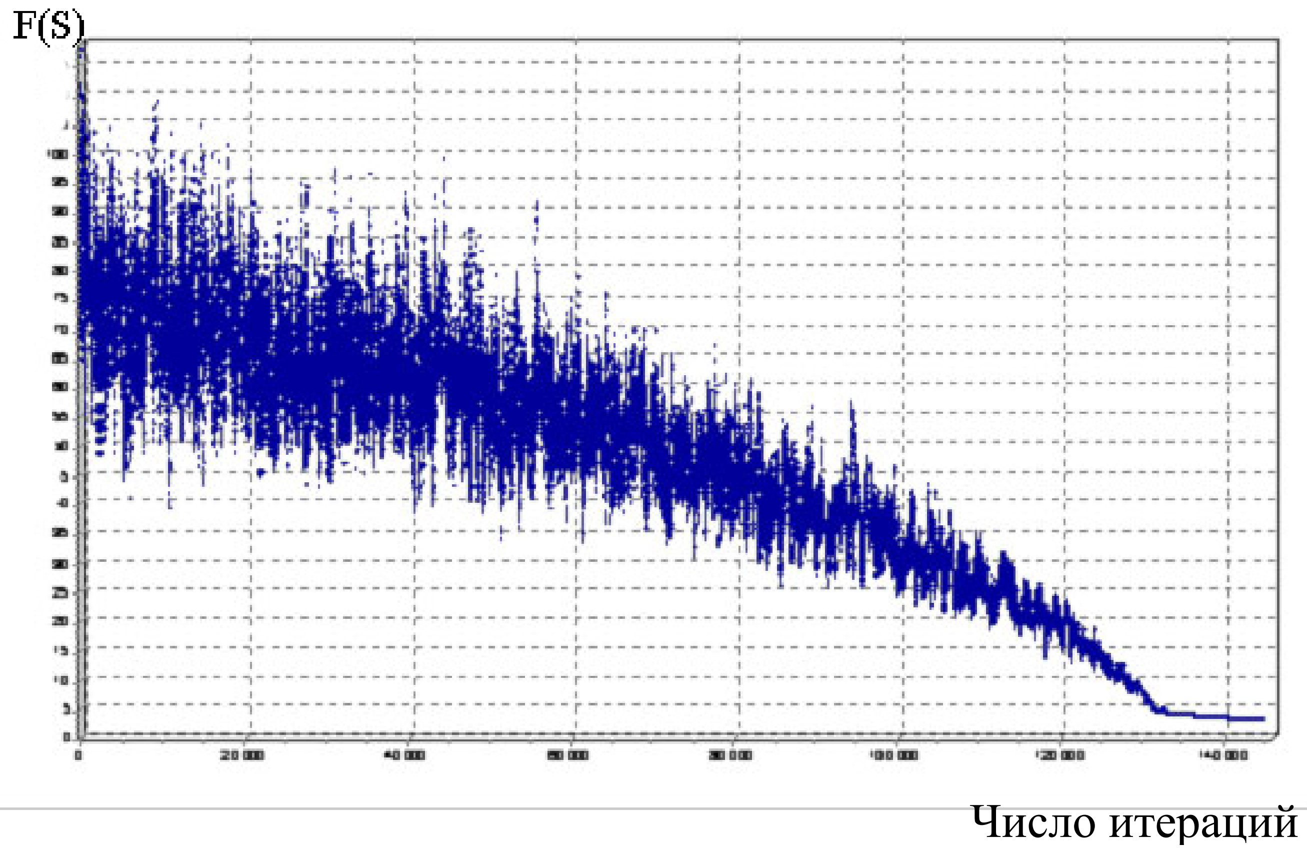
1. **Последовательное улучшение**: $t_k = 0$, $k = 0, 1, 2, \ldots$ — вариант локального спуска с монотонным улучшением по целевой функции.

2. **Пороговое улучшение**: $t_k = c_k$, $k = 0, 1, 2, \ldots$, $c_k \ge 0$; $c_k \ge c_{k+1}$ и $\lim\limits_{k\to\infty} c_k \to 0$ — вариант локального поиска, когда допускается ухудшение по целевой функции до некоторого заданного порога, и этот порог последовательно снижается до нуля.

3. **Имитация отжига**: $t_k \ge 0$, $k = 0, 1, 2, \ldots$ — случайная величина с математическим ожиданием $E(t_k) = c_k \ge 0$ — вариант локального поиска, когда
допускается произвольное ухудшение по целевой функции, но вероятность такого перехода обратно пропорциональна величине ухудшения

$$P_{ij} = \begin{cases} 1, & \text{если } f(j) \le f(i) \\ \exp\left(\dfrac{f(i) - f(j)}{c_k}\right), & \text{если } f(j) > f(i) \end{cases}.$$

Алгоритм имитации отжига



Предложен в 1983 г. физиками S. Kirpatrick,

C.D. Gelatt, M.P. Vecchi.

В основе аналогия с поведением атомов при медленном остывании тела.

Алгоритм не останавливается в локальном минимуме и может «путешествовать» по всей допустимой области.

Цепи Маркова

Определение. Пусть O обозначает множество возможных исходов некоторого случайного процесса. Цепь Маркова есть последовательность испытаний, когда вероятность исхода в каждом испытании зависит только от результата предшествующего испытания.

Пусть $x(k)$ — случайная переменная, обозначающая результат k-го испытания. Тогда для каждой пары $i,\ j \in O$ вероятность перехода от i к j при k-ом испытании задается выражением

$$P_{ij}(k) = \mathbb{P}\{x(k) = j \mid x(k-1) = i\}.$$

Матрица $\{P_{ij}\}$ называется переходной матрицей. Цепь Маркова называется конечной, если множество исходов O конечно, и однородной, если переходные вероятности не зависят от номера шага k.

Общая схема имитации отжига

1. Выбрать начальное решение i и вычислить $F(i)$.
2. Задать начальную температуру T.
3. Пока не выполнен критерий остановки, делать следующее:

   Выполнить цикл L раз:

   Выбрать в $N(i)$ случайным образом решение j;

   Положить $\Delta = F(j) - F(i)$;

   Если $\Delta \le 0$, то $i := j$;

   Если $\Delta > 0$, то с вероятностью $e^{-\Delta/T}$ положить $i := j$;

   3.2. Понизить температуру $T := T \cdot r$.

Теорема. Пусть $c_k = c$ для всех k, и для любой пары $i, j \in I$ найдется число $p \geq 1$, и допустимые решения $l_0, l_1, \ldots, l_p \in I$ такие, что $l_0 = i$, $l_p = j$ и $l_{k+1} \in N(l_k)$, $k = 0, 1, \ldots, p–1$. Тогда для цепи Маркова, порожденной алгоритмом имитации отжига, существует единственное стационарное распределение, компоненты которого задаются формулой

$$q_i(c) = \frac{\exp(-f(i)/c)}{\sum\limits_{j \in I} \exp(f(i)/c)}, \quad i \in I,$$

а

$$\lim_{c \downarrow 0} q_i(c) = \begin{cases} 1/|I^*|, & \text{если } \; i \in I^* \\ 0, & \text{в противном случае} \end{cases}$$

Последнее равенство, по сути, означает, что с ростом числа итераций вероятность оказаться в точке глобального оптимума $i^* \in I^*$ стремится к 1, если значение порога c_k стремится к нулю, т. е.

$$\lim_{c \downarrow 0} \lim_{k \to \infty} \mathbb{P}_c\{x(k) \in I^*\} = 1.$$

Параметры алгоритма

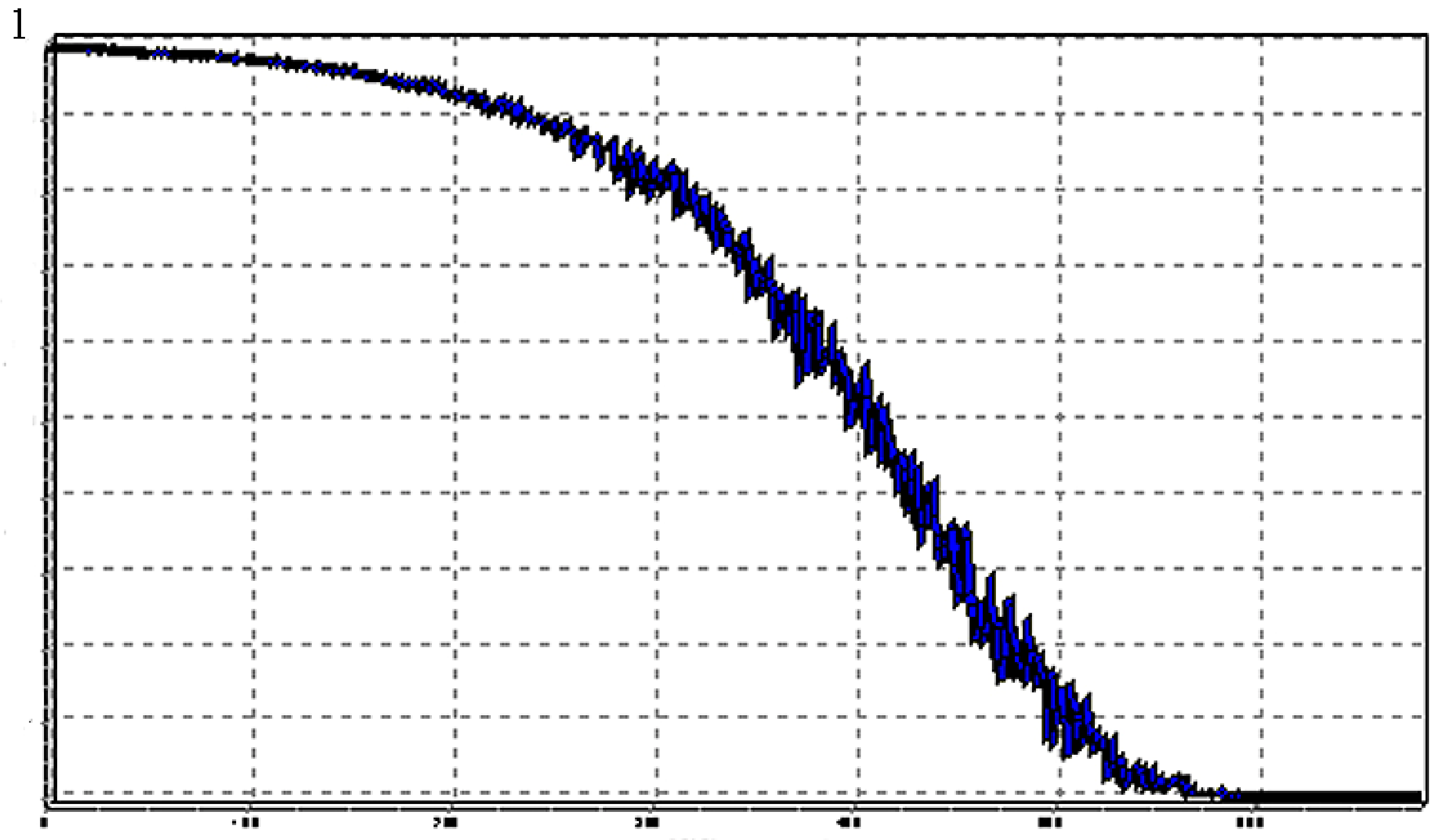
- Начальная температура T.
- Коэффициент охлаждения r.
- Число шагов L при заданной температуре
- Критерий остановки:

  – минимальное значение температуры

  – число смен температуры без изменения текущего решения S.

  – суммарное число шагов алгоритма

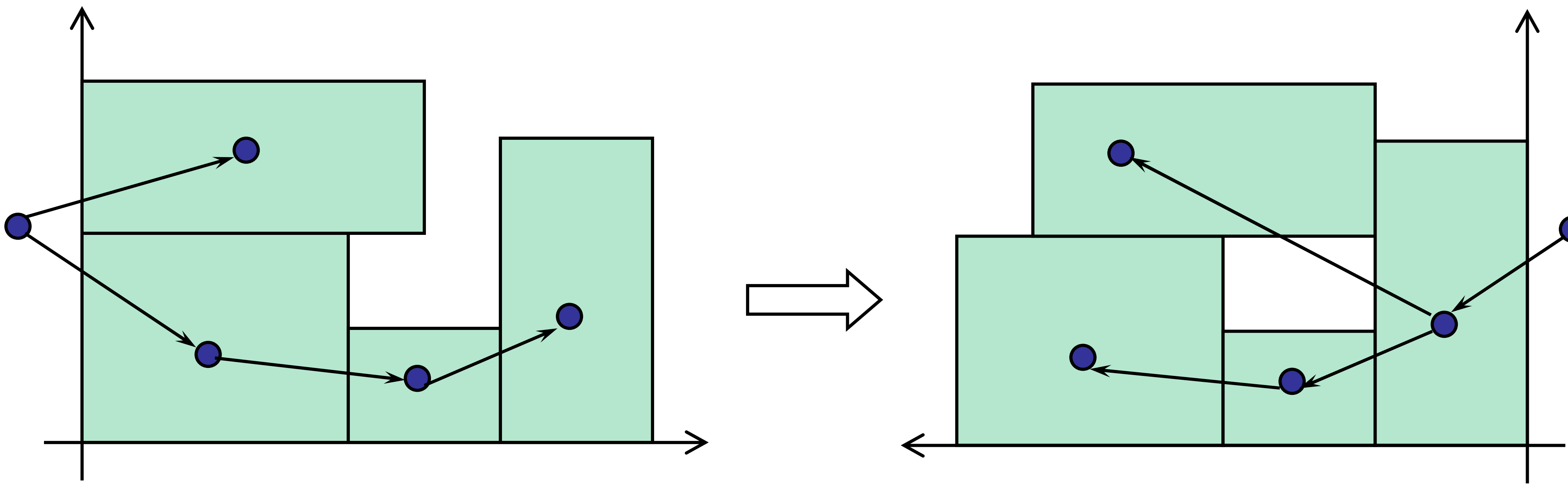
Зависимость частоты переходов в соседнее решение от температуры

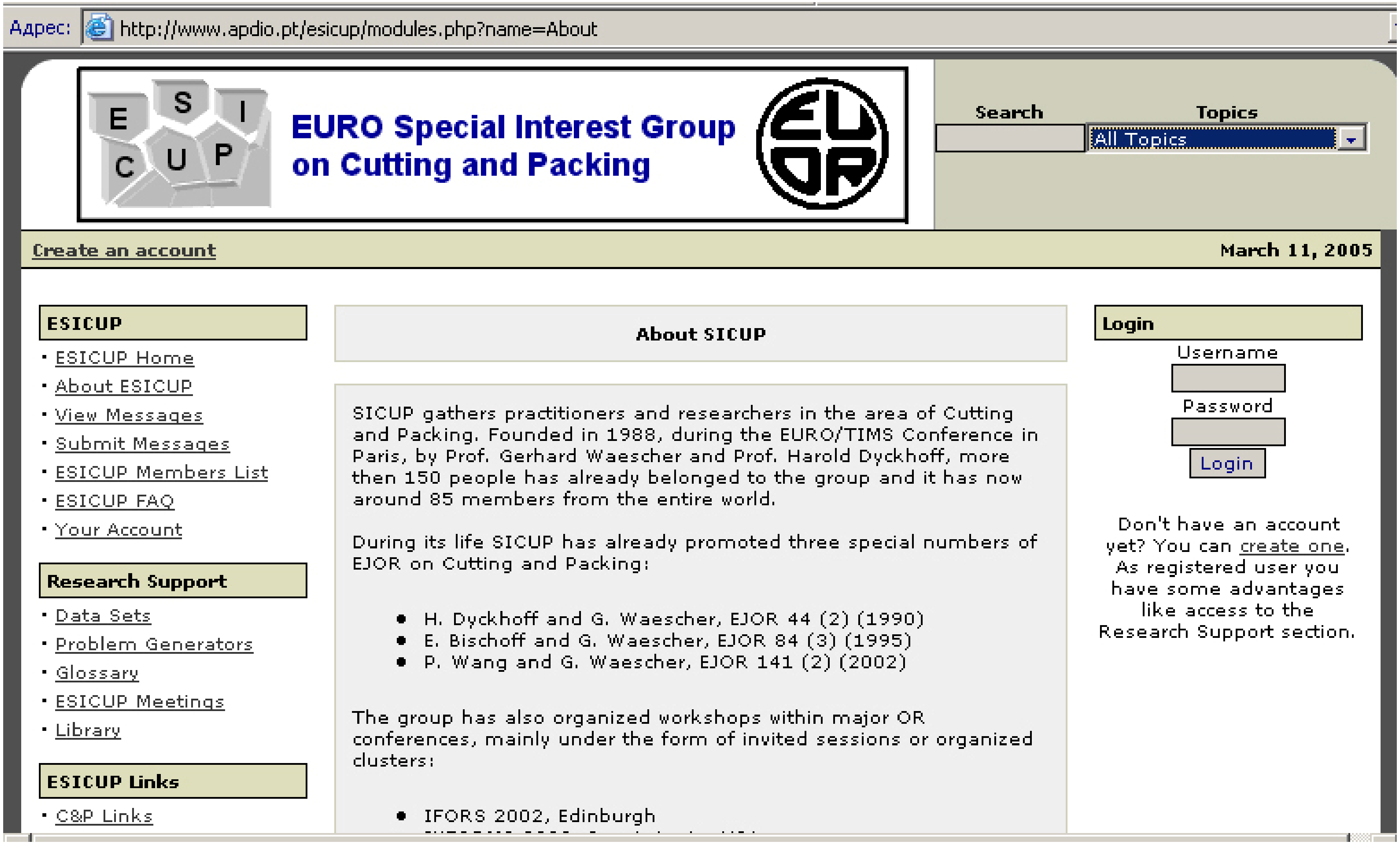


Понижение температуры

Переход от *LB*–компактных к *RB*–компактным решениям

Перестройка корневого ориентированного дерева без ухудшения целевой функции. Выполняется при каждой смене температуры.





Адрес: http://www.apdio.pt/esicup/modules.php?name=About

EURO Special Interest Group on Cutting and Packing

Search | Topics: All Topics

Create an account — March 11, 2005

ESICUP
- ESICUP Home
- About ESICUP
- View Messages
- Submit Messages
- ESICUP Members List
- ESICUP FAQ
- Your Account

Research Support
- Data Sets
- Problem Generators
- Glossary
- ESICUP Meetings
- Library

ESICUP Links
- C&P Links

About SICUP

SICUP gathers practitioners and researchers in the area of Cutting and Packing. Founded in 1988, during the EURO/TIMS Conference in Paris, by Prof. Gerhard Waescher and Prof. Harold Dyckhoff, more then 150 people has already belonged to the group and it has now around 85 members from the entire world.

During its life SICUP has already promoted three special numbers of EJOR on Cutting and Packing:

- H. Dyckhoff and G. Waescher, EJOR 44 (2) (1990)
- E. Bischoff and G. Waescher, EJOR 84 (3) (1995)
- P. Wang and G. Waescher, EJOR 141 (2) (2002)

The group has also organized workshops within major OR conferences, mainly under the form of invited sessions or organized clusters:

- IFORS 2002, Edinburgh

Login

Username

Password

Login

Don't have an account yet? You can create one. As registered user you have some advantages like access to the Research Support section.