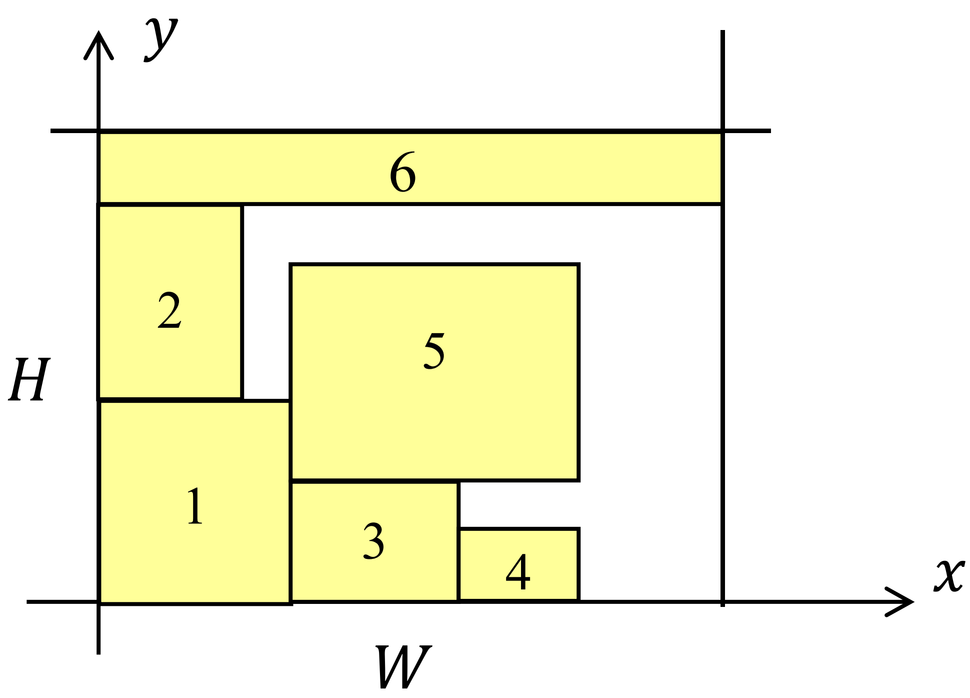
Задачи двумерной упаковки

Упаковка прямоугольников

Дано: n прямоугольных предметов с размерами $w_i \times h_i$, $i \in L = \{1, \ldots, n\}$.

Найти: упаковку с минимальной площадью окаймляющего прямоугольника



Стороны предметов параллельны осям координат.

$$\min WH$$

Задача является NP-трудной даже при фиксированном W.

Пример гильотинной упаковки

Математическая модель

(x_i, y_i) — координаты левого нижнего угла i-го прямоугольника

$l_{ij} = 1$, если i-й прямоугольник левее j-го, иначе 0

$b_{ij} = 1$, если i-й прямоугольник ниже j-го, иначе 0

$r_i = 1$, если i-й прямоугольник повернут на $90°$ иначе 0

$B \geq \sum_{i \in L}(w_i + h_i)$ — достаточно большое число.

$$\min W \cdot H$$

при ограничениях: $x_i + w_i(1 - r_i) + h_i r_i \leq W, \;\; i \in L;$

$$y_i + h_i(1 - r_i) + w_i r_i \leq H, \;\; i \in L;$$

$$x_i + w_i(1 - r_i) + h_i r_i \leq x_j + \left(1 - l_{ij}\right)B, \;\; i, j \in L;$$

$$y_i + h_i(1 - r_i) + w_i r_i \leq y_j + \left(1 - b_{ij}\right)B, \;\; i, j \in L;$$

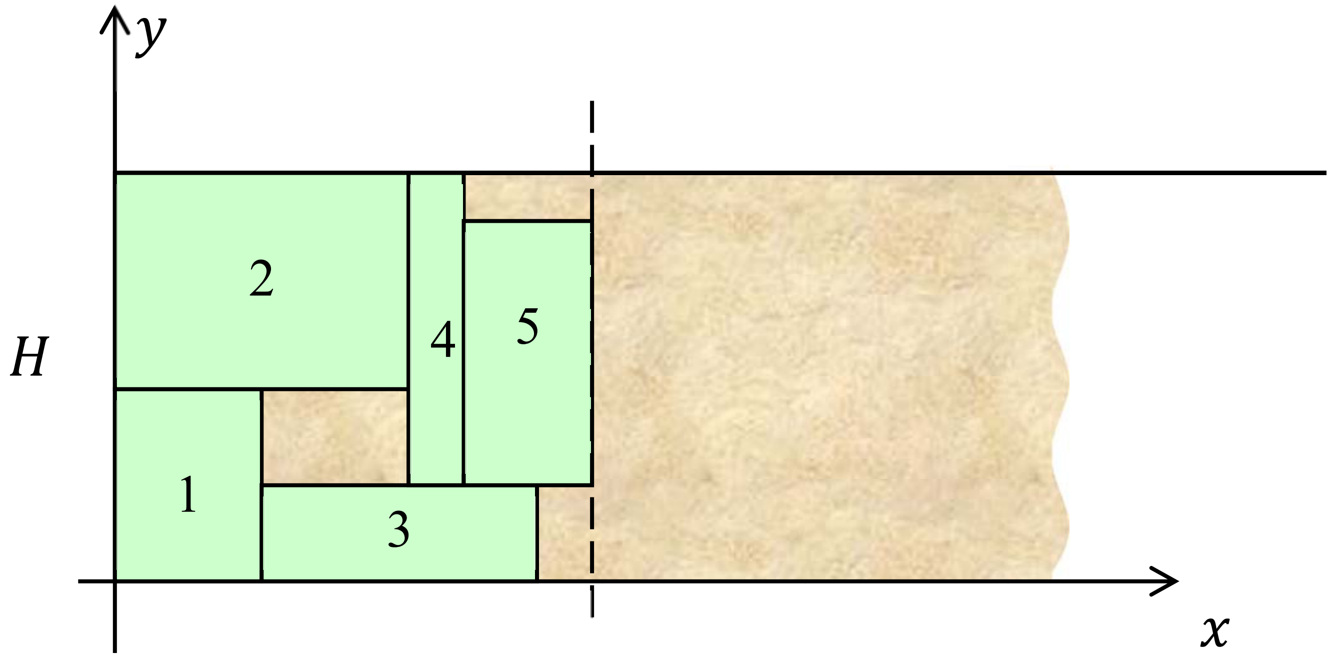
$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1, \;\; i, j \in L.$$

Задача упаковки в полосу

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

полубесконечная полоса шириной $H.$

Найти: упаковку предметов с минимальной длиной занимаемой полосы.



Математическая модель

$$\min\ W$$

при ограничениях: $x_i + w_i(1 - r_i) + h_i r_i \leq W,\ \ i \in L;$

$$y_i + h_i(1 - r_i) + w_i r_i \leq H,\ \ i \in L;$$

$$x_i + w_i(1 - r_i) + h_i r_i \leq x_j + \left(1 - l_{ij}\right)B,\ \ i, j \in L;$$

$$y_i + h_i(1 - r_i) + w_i r_i \leq y_j + \left(1 - b_{ij}\right)B,\ \ i, j \in L;$$

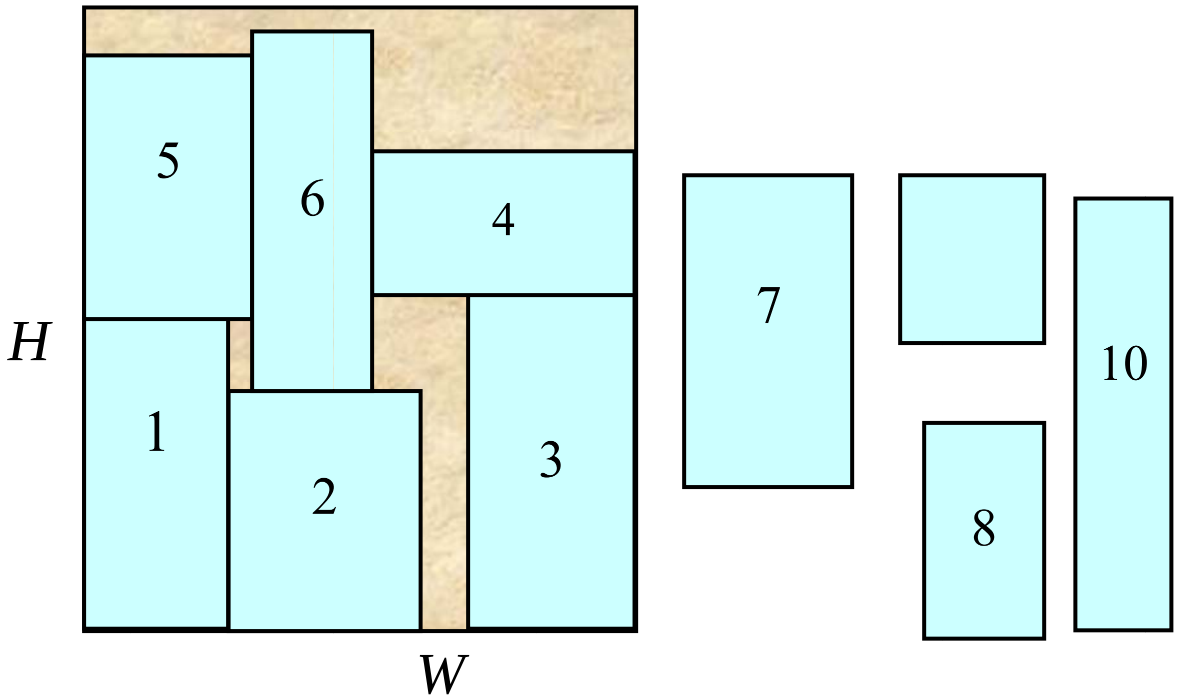
$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1,\ \ i, j \in L.$$

Задача целочисленного линейного программирования, можно применять коммерческое программное обеспечение GUROBI, CPLEX, GLPK и др.

Задача о рюкзаке для прямоугольников

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

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



При $h_i = H, i = 1, \ldots, n,$ получаем классическую задачу о рюкзаке

Математическая модель

Новые переменные:

$z_i = 1$, если i-й прямоугольник будем вырезать, иначе 0.

$$\max \sum_{i \in L} c_i z_i$$

при ограничениях:

$$x_i + w_i(1 - r_i) + h_i r_i \le W + (1 - z_i)B, \;\; i \in L;$$

$$y_i + h_i(1 - r_i) + w_i r_i \le H + (1 - z_i)B, \;\; i \in L;$$

$$x_i + w_i(1 - r_i) + h_i r_i \le x_j + \left(1 - l_{ij}\right)B, \;\; i, j \in L;$$

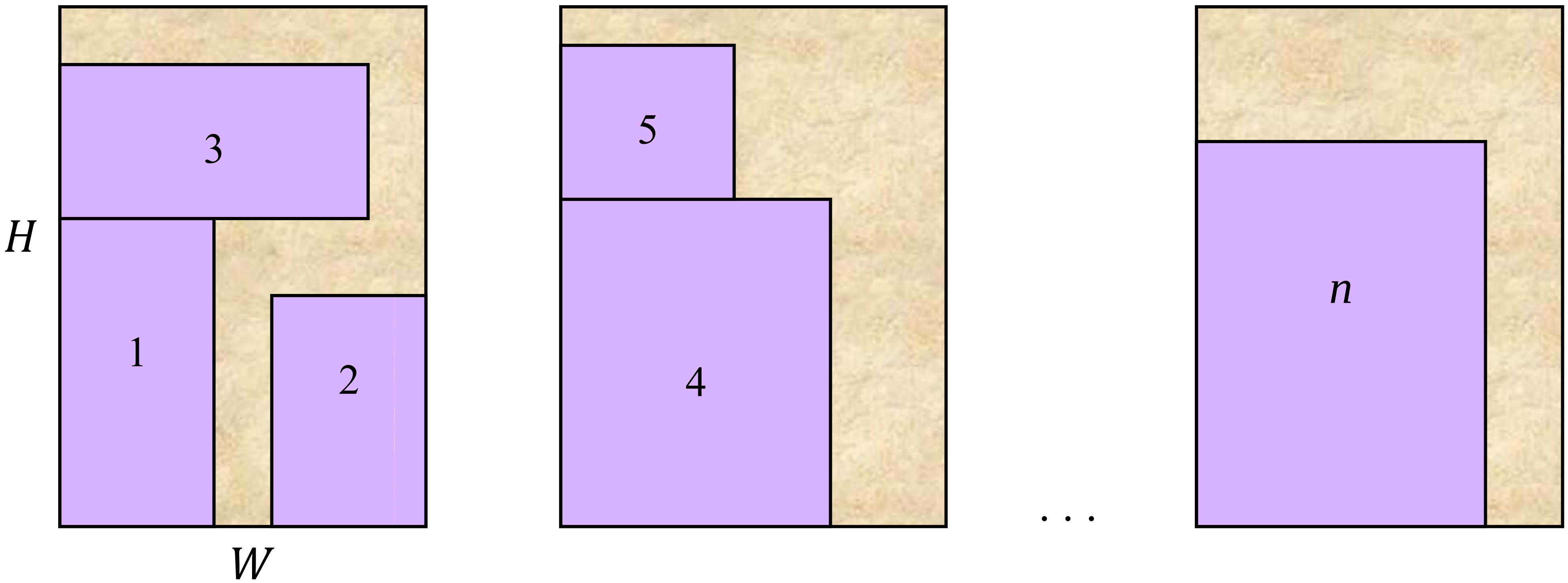
$$y_i + h_i(1 - r_i) + w_i r_i \le y_j + \left(1 - b_{ij}\right)B, \;\; i, j \in L;$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \ge 1, \;\; i, j \in L.$$

Задача упаковки в контейнеры для прямоугольников

Дано: n прямоугольных предметов с размерами $w_i \times h_i,\ i = 1, \dots, n,$
и неограниченное число одинаковых заготовок размером $H \times W$.

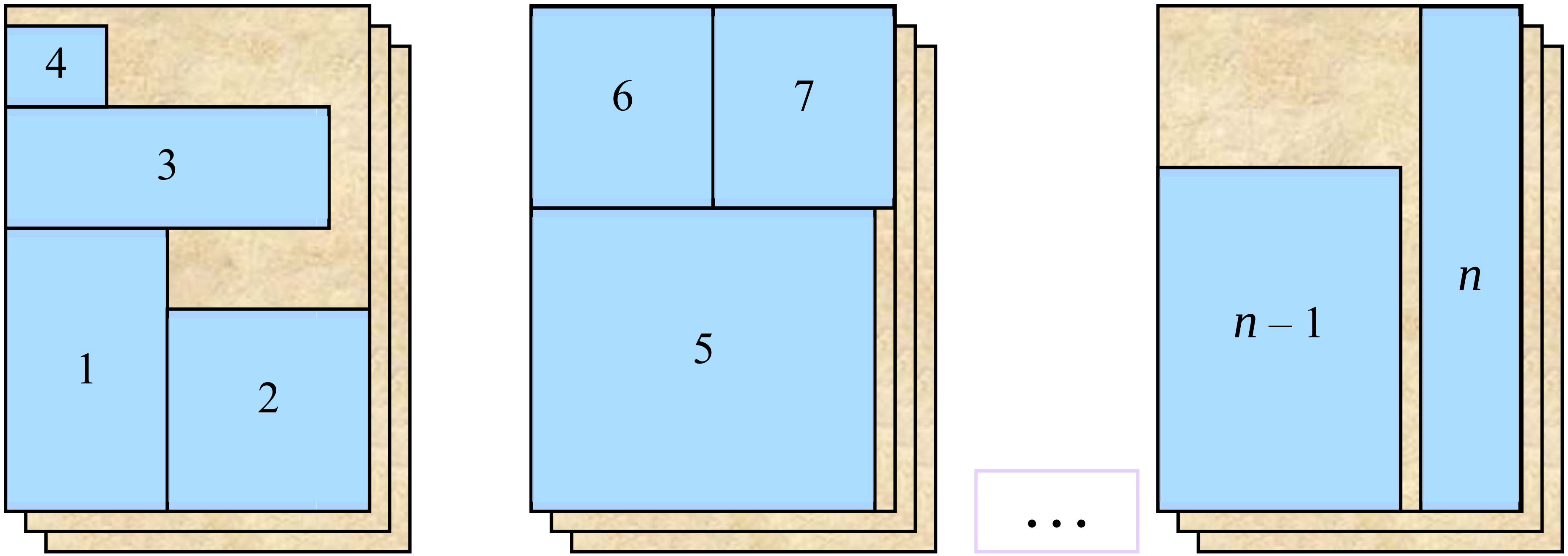
Найти: упаковку предметов с минимальным числом используемых заготовок.



Задача прямоугольного раскроя для серийного производства

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

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

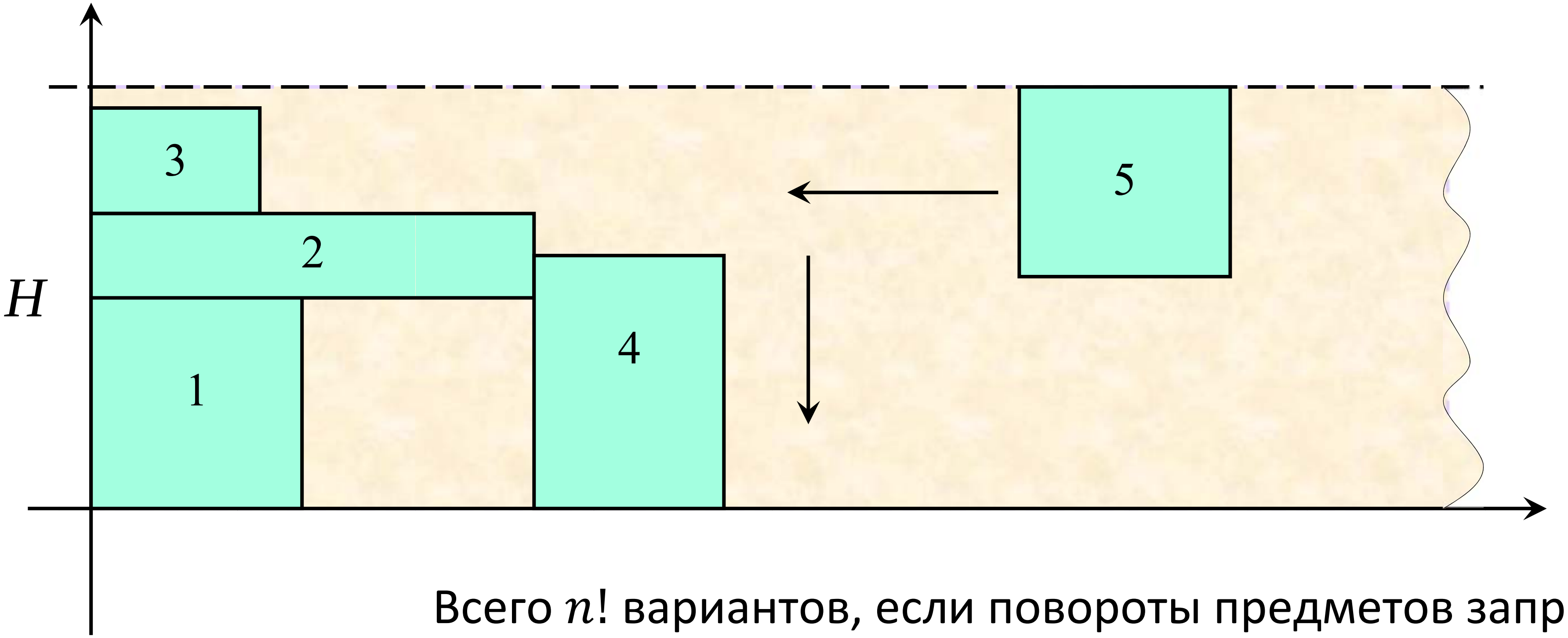


Кодировки решений

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

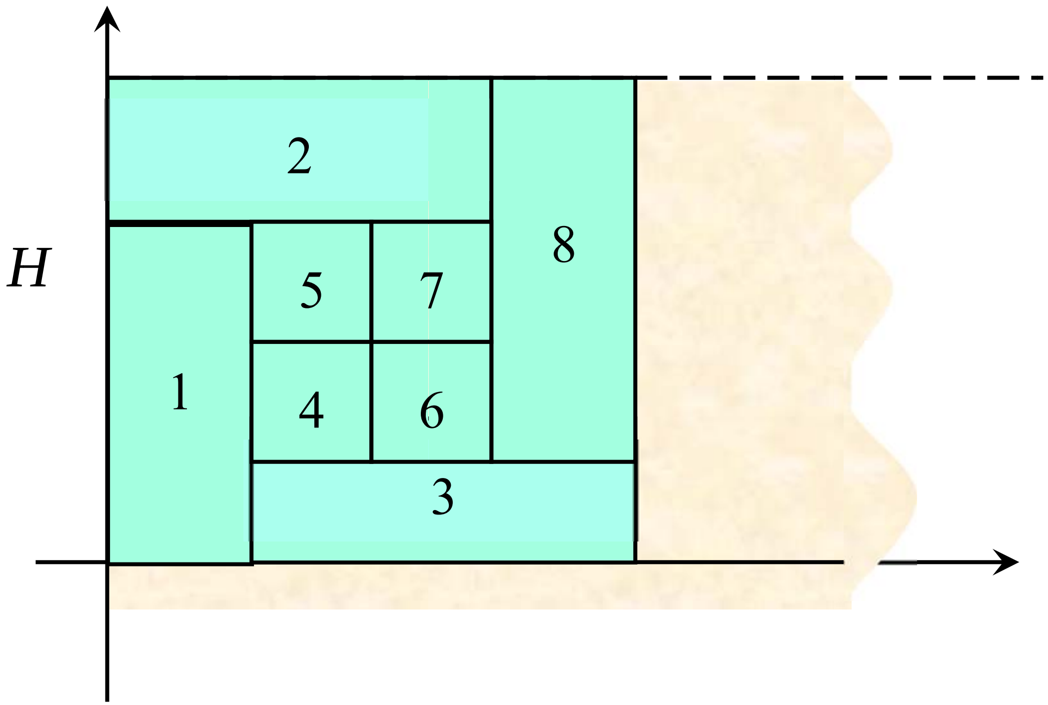
Декодирующая процедура: до упора влево, до упора вниз, до упора влево, …

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



Всего $n!$ вариантов, если повороты предметов запрещены.

Контрпример



Оптимальное решение, которое нельзя получить никаким списком.

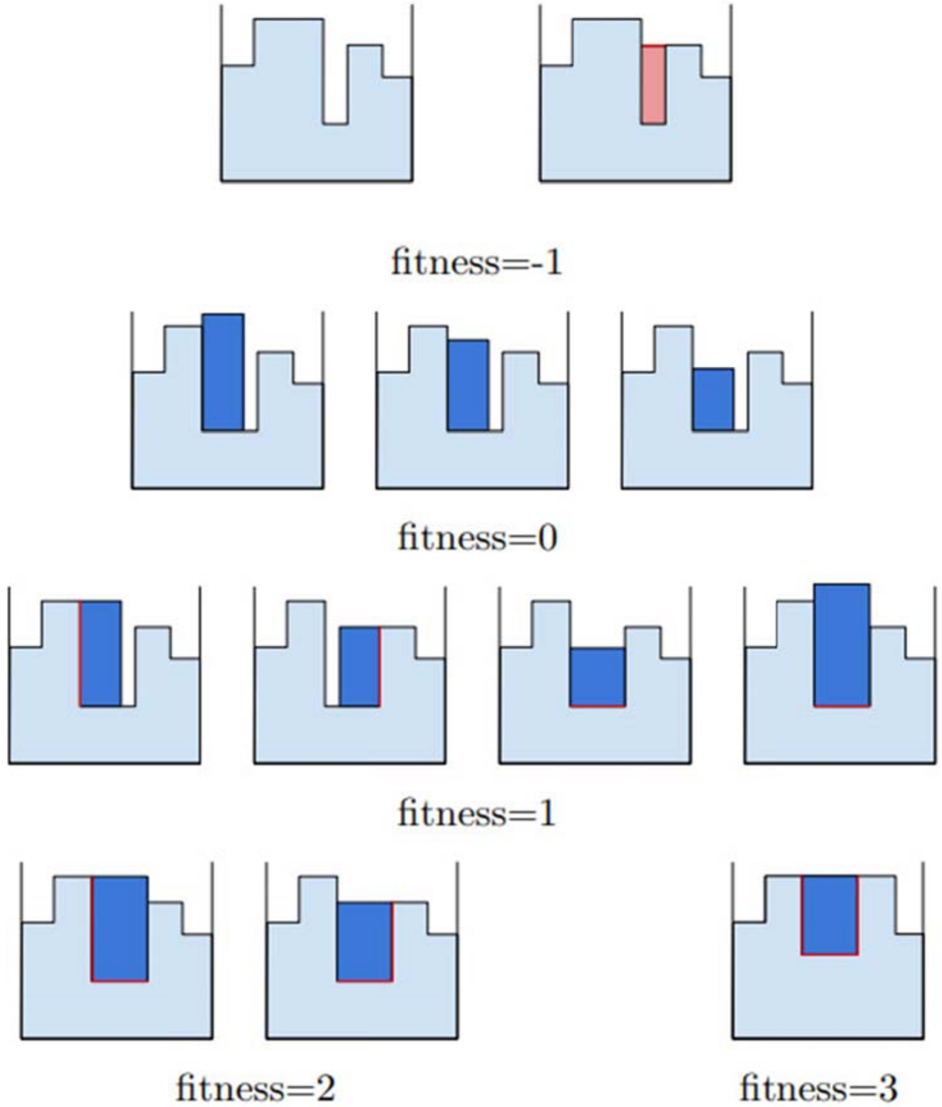
Skyline Алгоритм

Алгоритм работает со списками, но строит упаковку по новому правилу. На каждом шаге имеется подмножество уже упакованных предметов и текущий профиль упаковки. Шаг состоит в выборе одного неупакованного предмета и размещении его на профиле.

Профиль состоит из последовательности горизонтальных и вертикальных линий. Выбирается самая нижняя горизонтальная линия. Если таких несколько, то левая из них. Для нее выбирается наиболее подходящий неупакованный предмет по следующему правилу. Для каждого предмета вычисляется значение функции пригодности (*fitness function*) и берется первый в списке с максимальным значением.

Если ни один предмет не может поместиться на линии, то профиль перестраивается с генерацией минимального пустого места.

Функция пригодности – число сторон прямоугольника, в точности совпадающее со сторонами «ямы». Принимает только 5 значений {-1,0,1,2,3}.



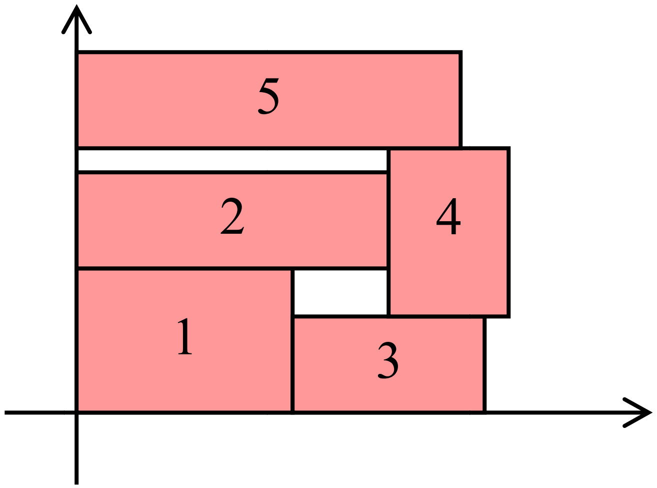
Определение. Кодировка решений называется ***допустимой,*** если она удовлетворяет следующим требованиям:

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.

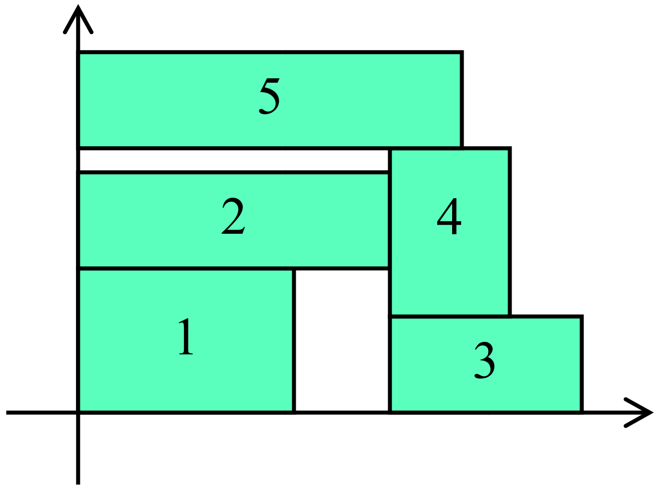
Пример. Упорядоченный список предметов с алгоритмом декодирования Skyline.

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

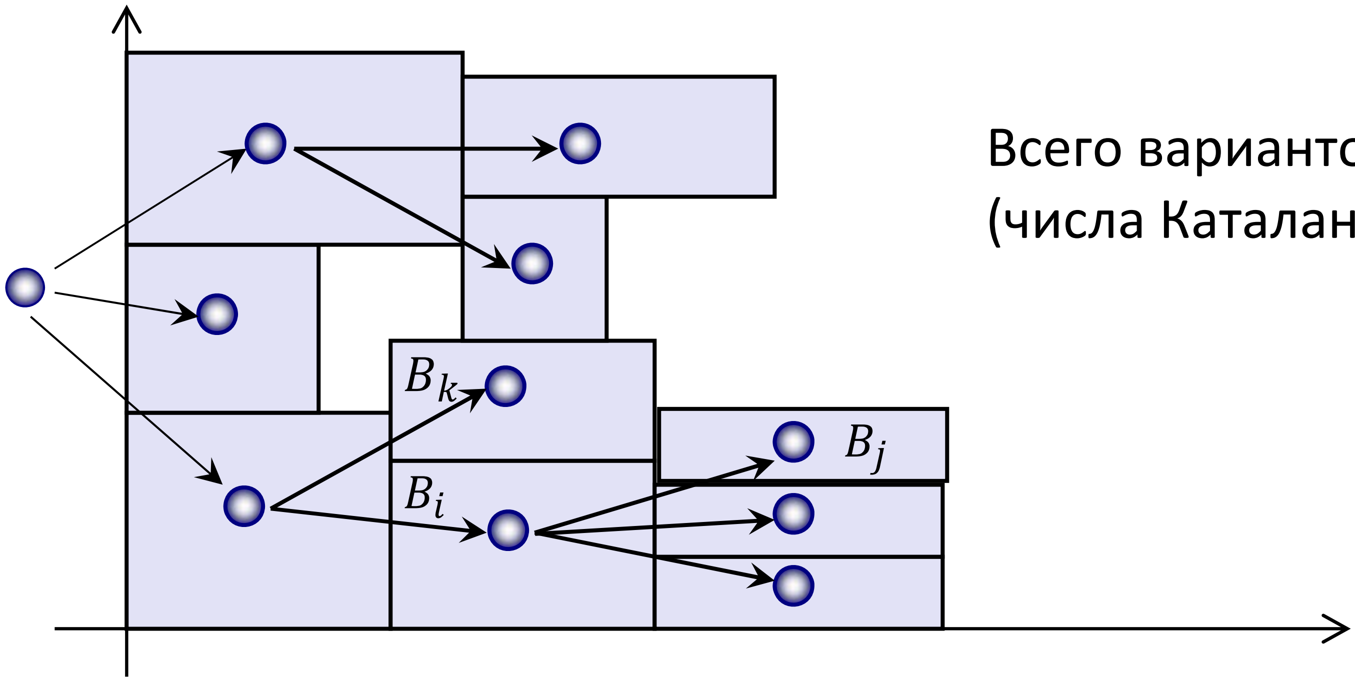


$B-$, но не $L-$компактное

Среди оптимальных решений всегда есть $LB-$ компактное решение.

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.

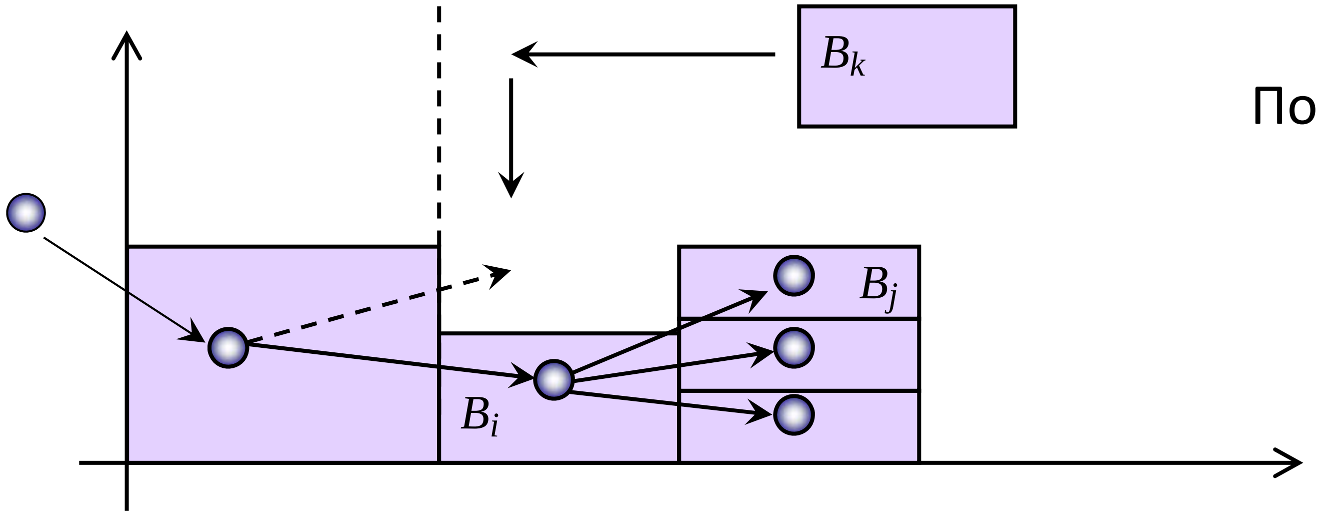


Всего вариантов $n!\, C_{2n+1}^{n}/(2n+1)$
(числа Каталана)

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 трудоемкостью $O(n)$.

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 называют **окрестностью** данной упаковки.

Стандартный алгоритм локального улучшения

1. Выбрать L_0 – начальное решение и вычислить его длину $F(L_0)$, $i := 0$.
2. Найти наилучшего соседа L_{i+1} для решения L_i :

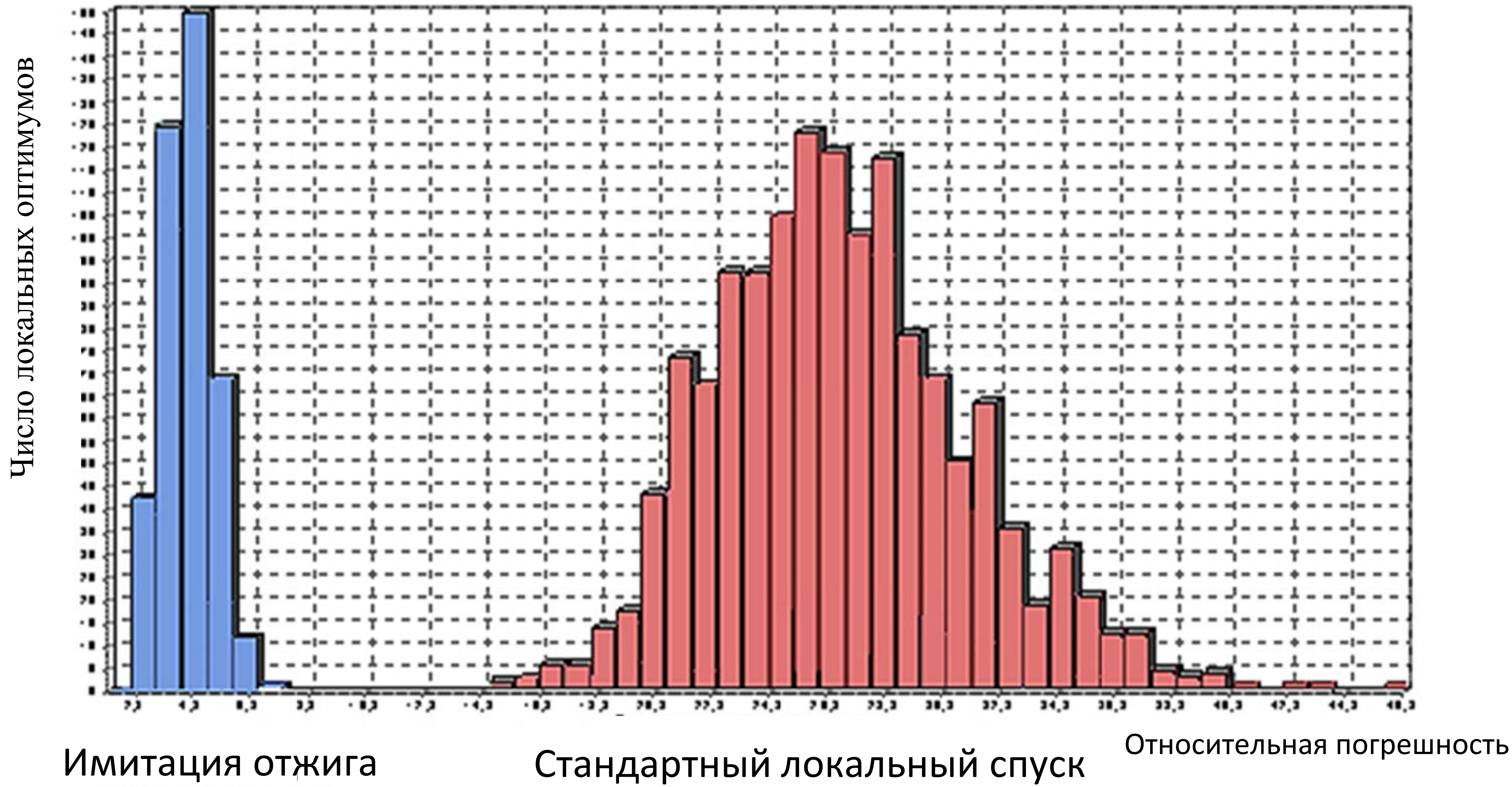
$$F(L_{i+1}) = \min\{F(L) | l \in N(L_i)\}.$$

Если $F(L_{i+1}) < F(L_i)$, то положить $i := i + 1$ и вернуться на шаг 2,

иначе STOP, L_i – локальный минимум.

Варианты: best improvement, first improvement, randomized neighborhoods, multi-start …

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 решение $j \in N(i_k)$.
   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

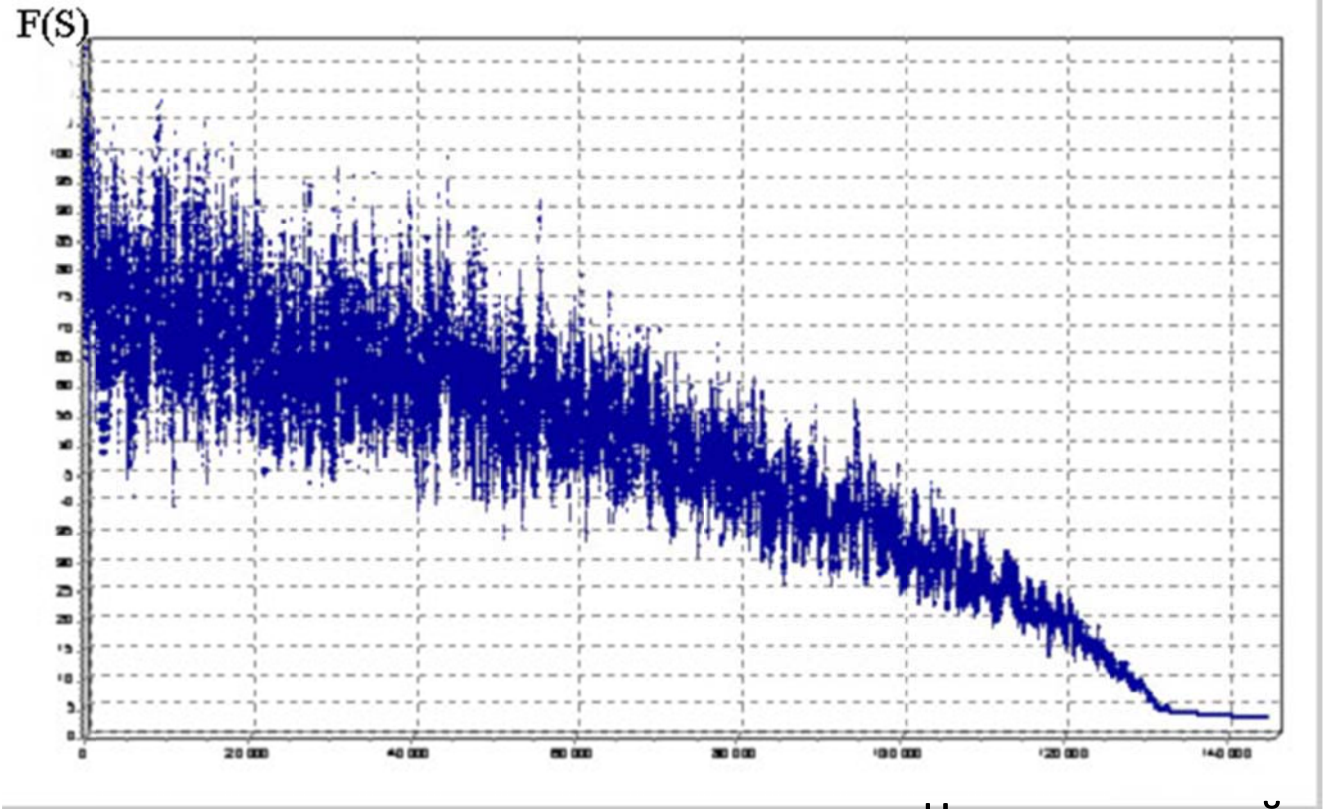
1. **Последовательное улучшение**: $t_k = 0,\ k = 0, 1, 2, \ldots$ – вариант локального спуска с монотонным улучшением по целевой функции.

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

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

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

Алгоритм имитации отжига



Число итераций

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-м 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 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:

   3.1. Выполнить цикл L раз:

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

      3.1.2. Положить $\Delta = F(j) - F(i)$;

      3.1.3. Если $\Delta \leq 0$, то $i \coloneqq j$;

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

   3.2. Понизить температуру $T \coloneqq T \cdot r$.

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

$$q_i(c) = \frac{\exp(-\frac{F(i)}{c})}{\sum_{j \in I} \exp(-\frac{F(j)}{c})}, \quad i \in I,$$

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:

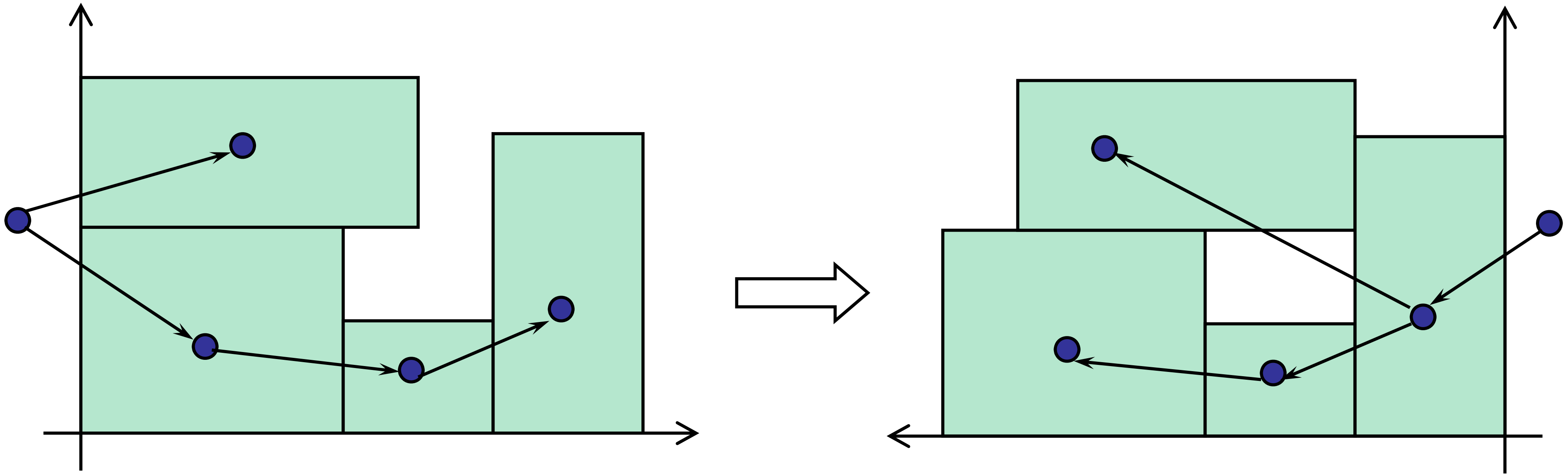
  – минимальное значение температуры

  – число смен температуры без изменения текущего решения.

  – суммарное число шагов алгоритма

Переход от LB–компактных к RB–компактным решениям

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



https://www.euro-online.org/websites/esicup/

Вопросы

- Процедура декодирования дерева в упаковку является полиномиальной (*Да или Нет?*)
- Любая упаковка прямоугольников в полосу является гильотинной (*Да или Нет?*)
- Переход от LB-компактных решений к RB-компактным решениям является полиномиальной процедурой?
- Метод имитации отжига в асимптотике всегда гарантирует получение точного решения задачи?
- Двумерная задача упаковки в контейнеры является NP-трудной?