ФРАГМЕНТАРНЫЕ МОДЕЛИ, ГИБРИДНЫЕ МЕТАЭВРИСТИКИ И СЛОЖНЫЕ ЗАДАЧИ ТРАНСПОРТНОЙ ЛОГИСТИКИ

КОЗИН И.В.   ЗАПОРОЖСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ

ЧАСТЬ 1. “ЖАДНЫЙ” АЛГОРИТМ

Жадный алгоритм (*Greedy algorithm*) — алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Задачи, в которых жадный алгоритм не приводит к оптимальному решению

- Задача коммивояжера
- Задача минимальной раскраски графа
- Задачи раскроя
- Задачи теории расписаний
- …………………

1. МАТРОИДЫ WHITNEY H. 1935Г.

$$\text{Матроид} - \text{пара } (X, E): 1.\ \varnothing \in E$$

$$2.\ \forall A \in E, B \subseteq A \Rightarrow B \in E$$

$$3.\ \forall A, B \in E, |A| = |B| + 1\ \ \exists x \in A \setminus B\ \ B \cup \{x\} \in E$$

2. Гридоиды Korte B., Lovasz L., Schrader R. 1991г.

$$\text{Гридоид} - \text{пара } (X, E): 1.\ \varnothing \in E$$

$$2.\ \forall A \in E, A \neq \varnothing \quad \exists x \in A \quad A \setminus \{x\} \in E$$

$$3.\ \forall A, B \in E, |A| > |B|\ \ \exists x \in A \setminus B\ \ B \cup \{x\} \in E$$

3. Наследственные системы Ильев В.П. 2007г.

Наследственная система - пара (X, E):

$$1.\ \varnothing \in E$$

$$2.\ \forall A \in E, A \neq \varnothing \quad \forall B \subset A \quad A \setminus B \in E$$

ФРАГМЕНТАРНАЯ СТРУКТУРА

Фрагментарная структура (X, E):

1. $\varnothing \in E$
2. $\forall A \in E, A \neq \varnothing \quad \exists x \in A \quad A \setminus \{x\} \in E$

Пусть (X, E) - фрагментарная структура.

$\forall A \in E \quad \exists B_{0,} B_1, ..., B_n \qquad B_0 \subset B_1 \subset ... \subset B_n = A$

$B_0 = \varnothing, \quad \forall k = 1, 2, ..., n-1 \quad |B_{k+1} \setminus B_k| = 1$

ОРИЕНТИРОВАННАЯ ФРАГМЕНТАРНАЯ СТРУКТУРА

Ориентированная фрагментарная структура (X, E):

1. $\varnothing \in E$
2. $\forall A \in E, A \neq \varnothing \quad A = \{x_1, x_2, ..., x_k\} \quad \forall s \leq k \quad \{x_1, x_2, ..., x_s\} \in E$

(X, E) - фрагментарная структура

Определение_1. Элементы множества ***E*** будем называть допустимыми фрагментами.

Определение_2. Одноэлементные множества, которые являются допустимыми фрагментами, будем называть элементарными фрагментами.

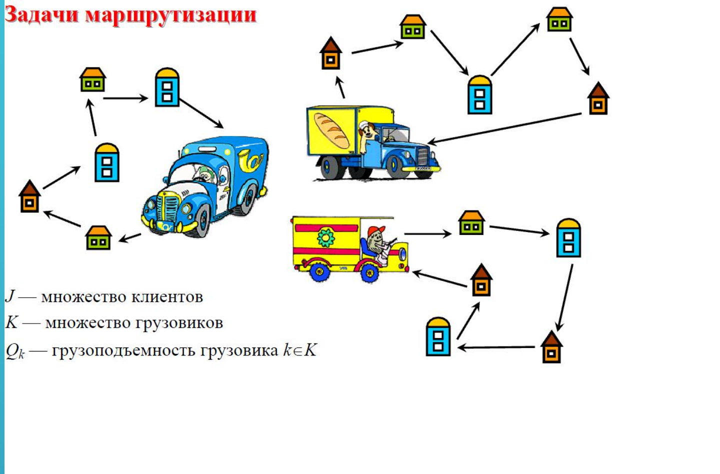
Определение_3. Фрагмент называется максимальным, если он не является подмножеством (начальной подпоследовательностью) никакого другого допустимого фрагмента.

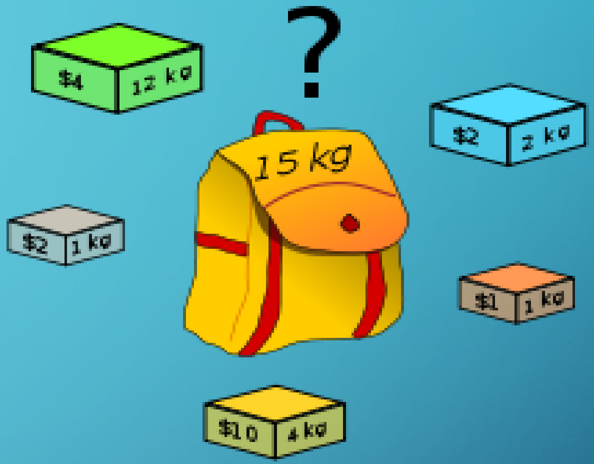
ФРАГМЕНТАРНАЯ СТРУКТУРА И ФРАГМЕНТАРНЫЙ АЛГОРИТМ

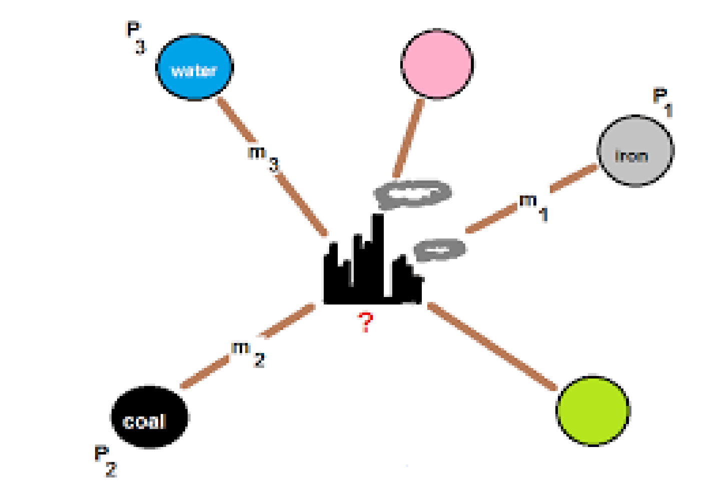
Максимальный фрагмент строится с помощью следующего жадного алгоритма:

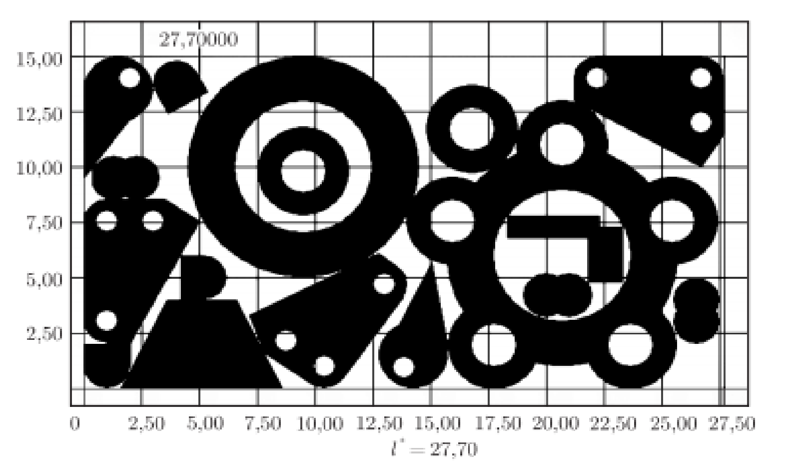
- предварительно элементы множества Y линейно упорядочиваются;
- на начальном шаге выбирается пустое множество $Y_0 = \varnothing$;
- на шаге с номером $k+1$ выбирается первый по порядку элемент $y \in Y \setminus Y_k$, такой, что $Y_k \cup \{y\} \in E$, после чего строится множество $Y_{k+1} = Y_k \cup \{\dot{y}\}$
- алгоритм заканчивает работу, если на очередном шаге не удалось найти элемент $y \in Y \setminus Y_k$ с требуемым свойством.

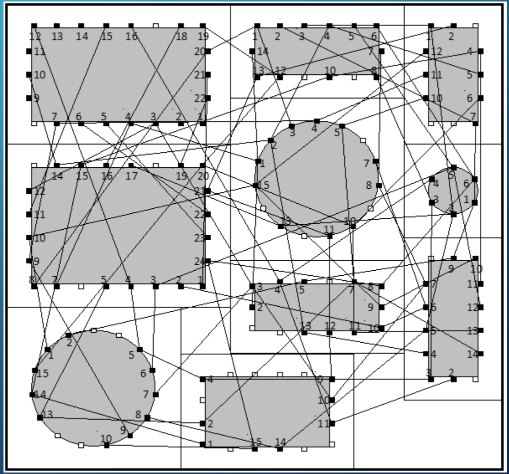
ПРИМЕРЫ ФРАГМЕНТАРНЫХ СТРУКТУР



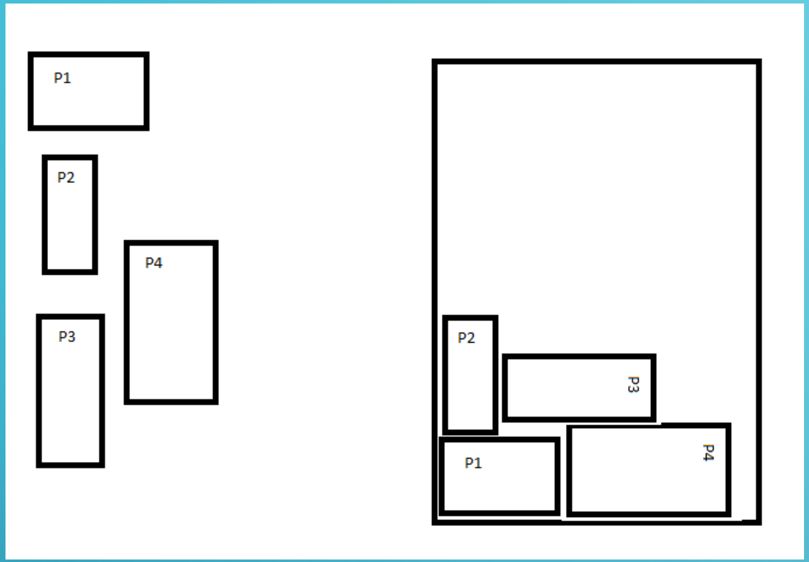


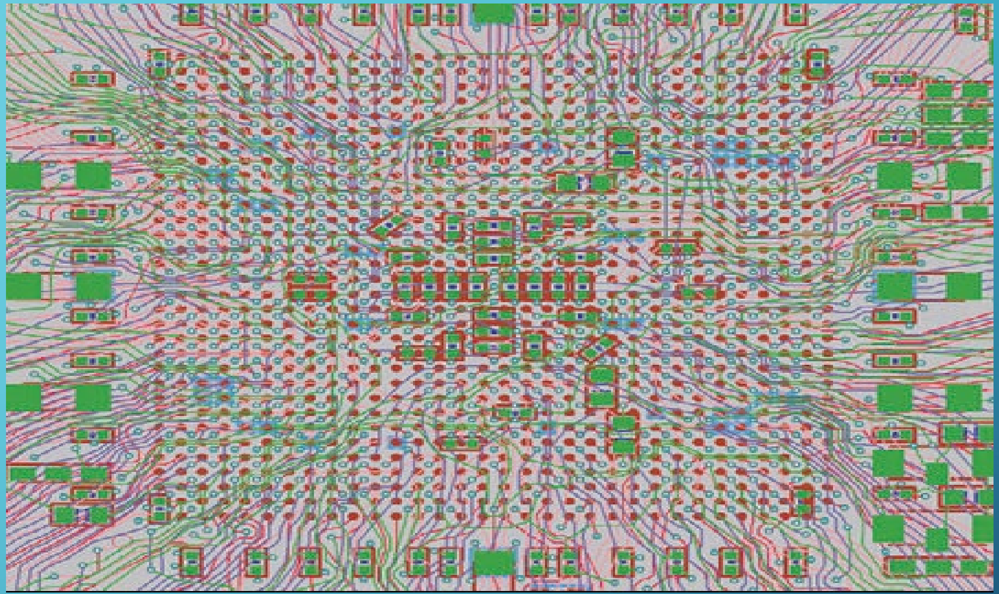


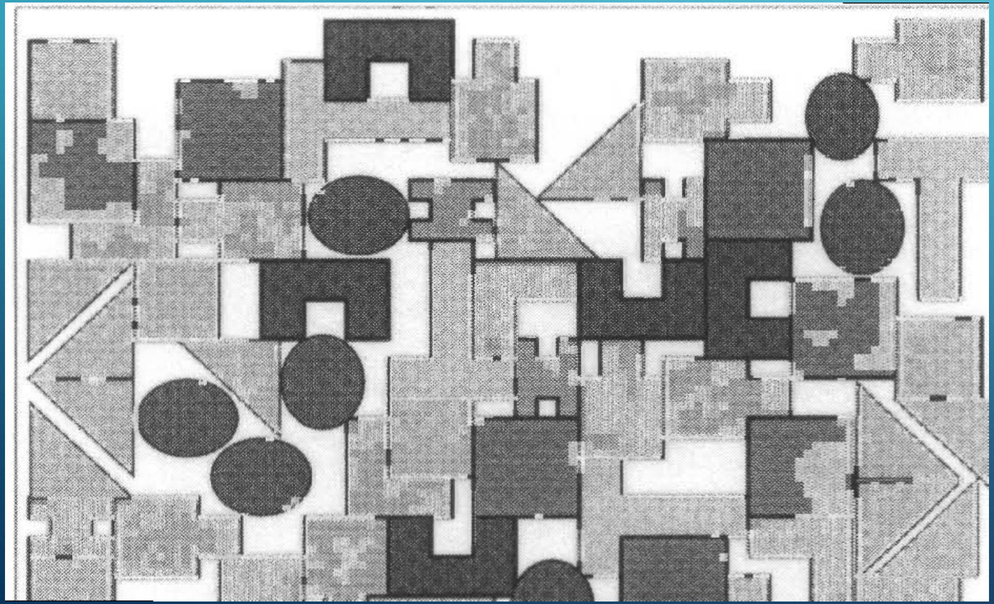




ПРИМЕРЫ ФРАГМЕНТАРНЫХ СТРУКТУР







ОПТИМИЗАЦИЯ НА ФРАГМЕНТАРНОЙ СТРУКТУРЕ

Фрагментарная структура называется взвешенной, если задана функция $F: E \to \mathrm{R}^1{}_+$ $(F : 2^X \to R^1_+)$

Весовая функция монотонная, если $\forall A, B \in E, A \subseteq B$ => $\mathrm{F}(A) \leq F(B)$

Весовая функция аддитивная, если $\forall e \in E \quad F(e) = \sum_{x \in e} F(\{x\})$

ФРАГМЕНТАРНАЯ МОДЕЛЬ

$$\begin{tikzcd} S_{\mathrm{n}} \arrow[d, "\varphi"'] \arrow[dr, "F \circ \varphi"] & \\ X \arrow[r, "F"] & \mathrm{R}^1 \end{tikzcd}$$

$$F \circ \varphi(\mathrm{s}) \longrightarrow \max\ (\min)$$

Достижимость

Определение_4. Фрагментарная модель имеет свойство достижимости 1-го типа, если любое **допустимое** решение может быть получено фрагментарным алгоритмом при некоторой перестановке элементарных фрагментов.

Определение_5. Фрагментарная модель имеет свойство достижимости 2-го типа, если любое **оптимальное** решение может быть получено фрагментарным алгоритмом при некоторой перестановке элементарных фрагментов.

Определение_6. Фрагментарная модель имеет свойство достижимости 3-го типа, если существует **оптимальное** решение которое может быть получено фрагментарным алгоритмом при некоторой перестановке элементарных фрагментов.

ЧАСТЬ 2. МЕТАЭВРИСТИКИ НА ФРАГМЕНТАРНЫХ СТРУКТУРАХ

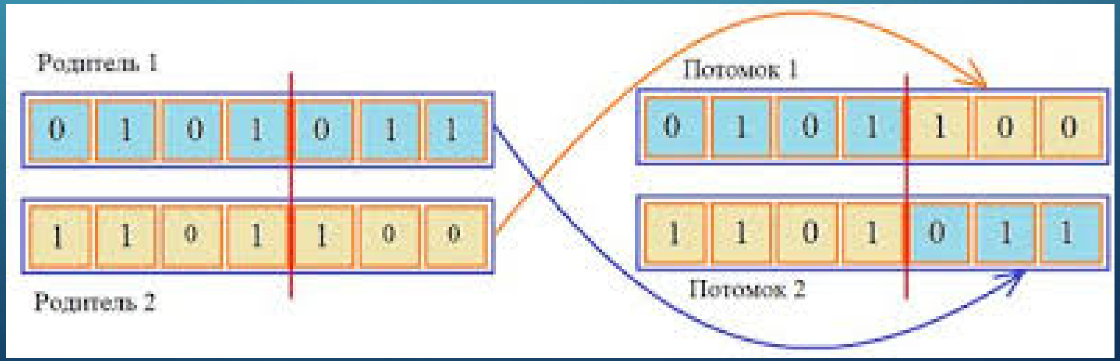
ГИПОТЕЗА БОЛЬШОЙ ДОЛИНЫ
(ИЛИ БОЛЬШОЙ ВЕРШИНЫ)

Гипотеза большой долины предполагает, что при комбинаторной оптимизации локальные оптимумы хорошего качества группируются и окружают глобальный оптимум

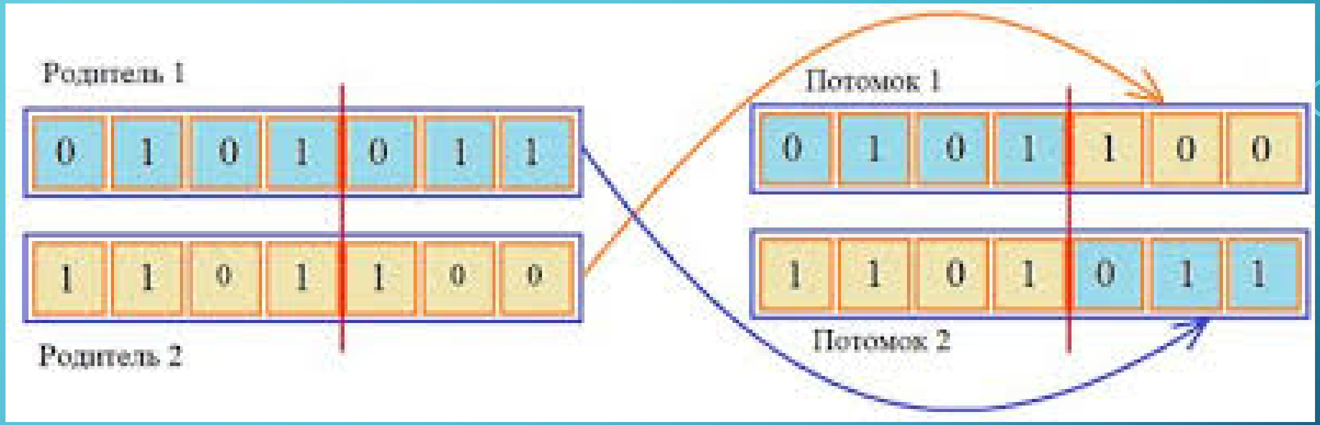
ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

1954 год - Нільс Баричелли Принстонский университет.[
1957 год - Алекс Фрезер
Начало 70-х годов Джон Холанд

1. **Теория Дарвина – борьба за существование.**
2. **Теория наследственности – признаки передаются по наследству.**
3. **Теория катастроф – необходимость мутаций.**



ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ



ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ

$$F(x) \to \max(\min) \quad x \in X$$

1 Оператор построения начальной популяции
2. Оператор селекции
3. Оператор кроссовера
4. Оператор мутации
5 Оператор отбора

Геометрическая теория эволюционных алгоритмов

Dr Alberto Moraglio

Yoon Y, Kim Y-H, Moraglio A, Moon B-R. (2007) Geometric crossovers for real-code representation, *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference*, pages 1539-1539, DOI:10.1145/1276958.1277268.

Moraglio A. (2011) Geometry of evolutionary algorithms, *Genetic and Evolutionary Computation Conference, GECCO'11 - Companion Publication*, pages 1439-1467, DOI:10.1145/2001858.2002144.

Geometric Theory of Evolutionary Algorithms

http://www.slideshare.net/AlbertoMoraglio/cec-2013-tutorial

ГЕОМЕТРИЯ МЕТАЭВРИСТИК

Метрика в базовом пространстве

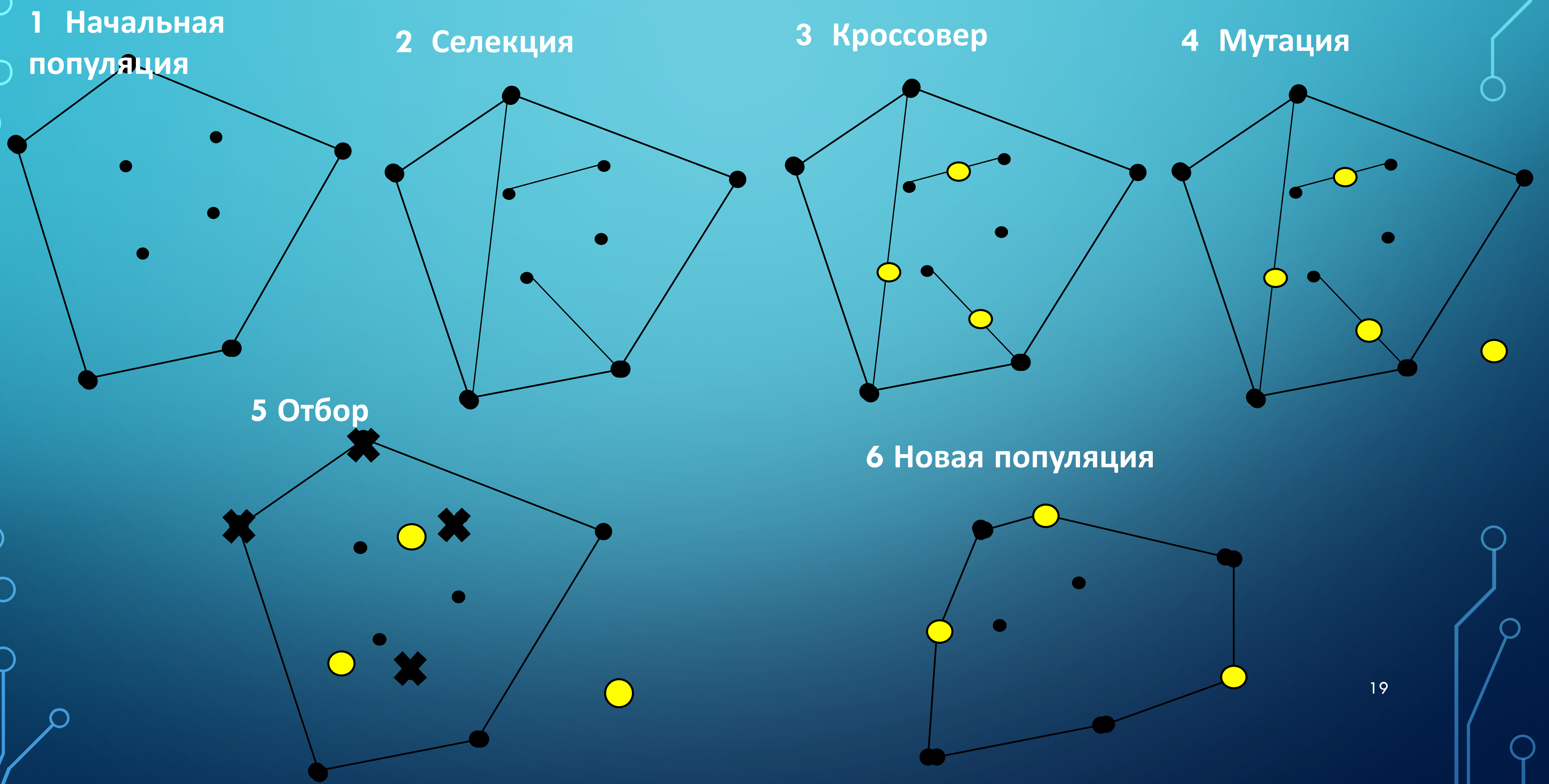
1. $\forall x, y \in X \quad \rho(x, y) \geq 0; \quad \rho(x, y) = 0 \Leftrightarrow x = y$
2. $\forall x, y \in X \quad \rho(x, y) = \rho(y, x)$
3. $\forall x, y, z \in X \quad \rho(x, y) \leq \rho(x, z) + \rho(z, y)$

Отрезок:

$$[x, y] = \{z \in X : \ \rho(x, y) = \rho(x, z) + \rho(z, y)$$

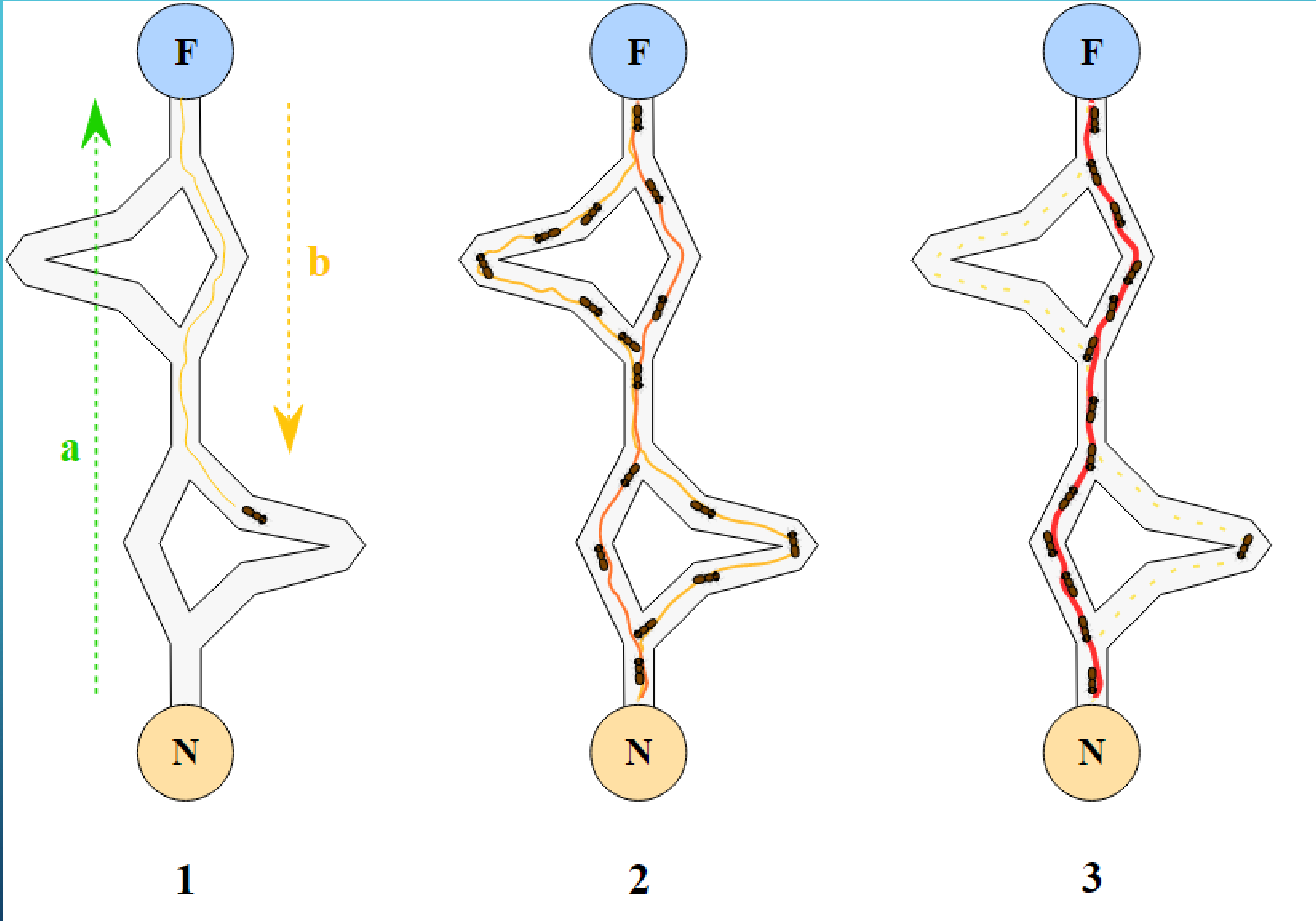
Выпуклое множество

ПРИНЦИП РАБОТЫ ЭВОЛЮЦИОННОЙ МЕТАЭВРИСТИКИ
С ГЕОМЕТРИЧЕСКИМ ОПЕРАТОРОМ КРОССОВЕРА



Алгоритм колонии муравьев

Dorigo M. Optimization, Learning, and Natural Algorithms. PhD Thesis, Dipartimento di Elettronica, Politechnico Di Milano, Italy. – 1992.– 140 p.



МУРАВЬИНЫЙ АЛГОРИТМ

Количество муравьёв задаётся равным общему числу элементов множества Y.

$J_{i,k}$ – множество фигур, доступных для выбора муравью k на шаге i.
D_{ij} – переход муравья от фигуры i к фигуре j.

Для коммуникации друг с другом муравьи используют **феромон**.

$\tau_{ij}(t)$ – количество феромона в момент времени t, которое соответствует переходу D_{ij}.

Вероятность выбора муравьём k перехода D_{ij}:

$$P_{ij,k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)}{\sum\limits_{l \in J_{i,k}} \tau_{il}(t)}, & j \in J_{i,k} \\ 0, & j \notin J_{i,k} \end{cases}$$

МУРАВЬИНЫЙ АЛГОРИТМ

$T_k(t)$ – множество всех переходов, сделанных муравьём k.

Количество феромона, которое муравей откладывает для каждого перехода:

$$\Delta\tau_{ij,k}(t) = \begin{cases} \dfrac{Q}{|T_k(t)|}, & D_{ij} \in T_k(t) \\ 0, & D_{ij} \notin T_k(t) \end{cases}$$

Дополнительно используется эффект **испарения феромона**.

Новое количество феромона определяется по формуле:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^{s} \Delta\tau_{ij,k}(t)$$

p – коэффициент испарения феромона.

s – количество всех муравьёв.

Алгоритм перемешанных прыгающих лягушек

1. Инициализация

N – максимальное число итераций

K – размер популяции лягушек

M – длина вектора позиции лягушки (длина перестановки)

Q – количество классов лягушек (мощности классов одинаковы)

D – количество лучших лягушек в классе ($D < K/Q$)

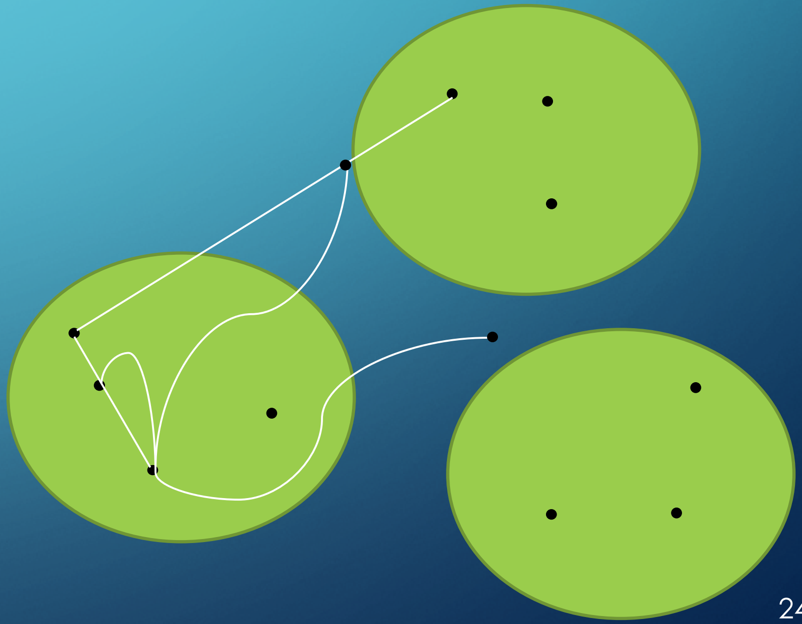
$F(x)$ – целевая функция, $x \in SK$

АЛГОРИТМ ПЕРЕМЕШАННЫХ ПРЫГАЮЩИХ ЛЯГУШЕК

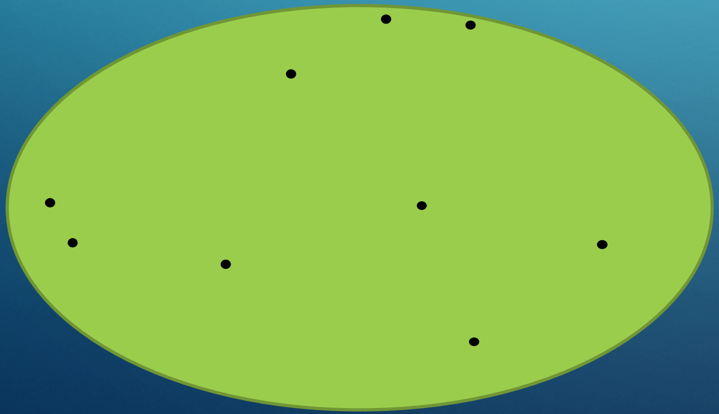
1. Начальная популяция $P^{(0)} = (x_1, x_2, \ldots, x_K)$ $F(x_j) \le F(x_{j+1})$,

2. Разбиение на классы

3. Улучшение худших позиций в классах (прыжки)



4. Перемешивание . Объединение классов




ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ДОНЕЦКАЯ АКАДЕМИЯ АВТОМОБИЛЬНОГО ТРАНСПОРТА

Ю.А. СКОБЦОВ
Е.Е. ФЕДОРОВ

МЕТАЭВРИСТИКИ

Монография

Донецк
Издательство «Ноулидж»
Донецкое отделение
2013

Часть 1
НАТУРАЛЬНЫЕ НЕПОПУЛЯЦИОННЫЕ МЕТАЭВРИСТИКИ И НЕНАТУРАЛЬНЫЕ МЕТАЭВРИСТИКИ

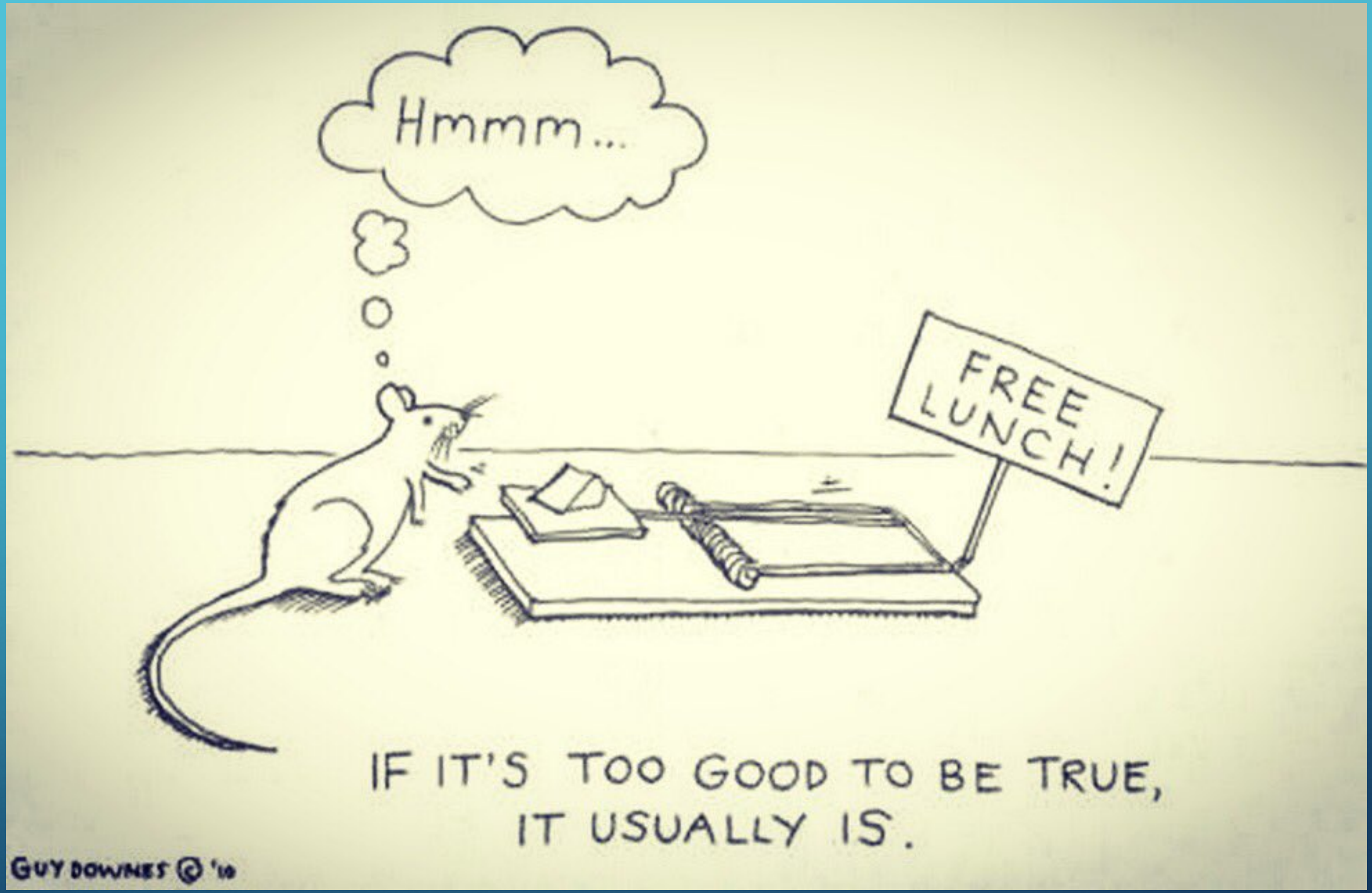
Часть 2
ДЕТЕРМИНИРОВАННЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ

Часть 3
ВЕРОЯТНОСТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ

Часть 4
РОЕВЫЕ МЕТАЭВРИСТИКИ

Часть 5
ИММУННЫЕ МЕТАЭВРИСТИКИ

- Теорема об отсутствии бесплатных обедов
- No Free lanch theorem



Часть 3. Сложные задачи транспортной логистики

Емкостный VRP Capacitated Vehicle Routing Problems

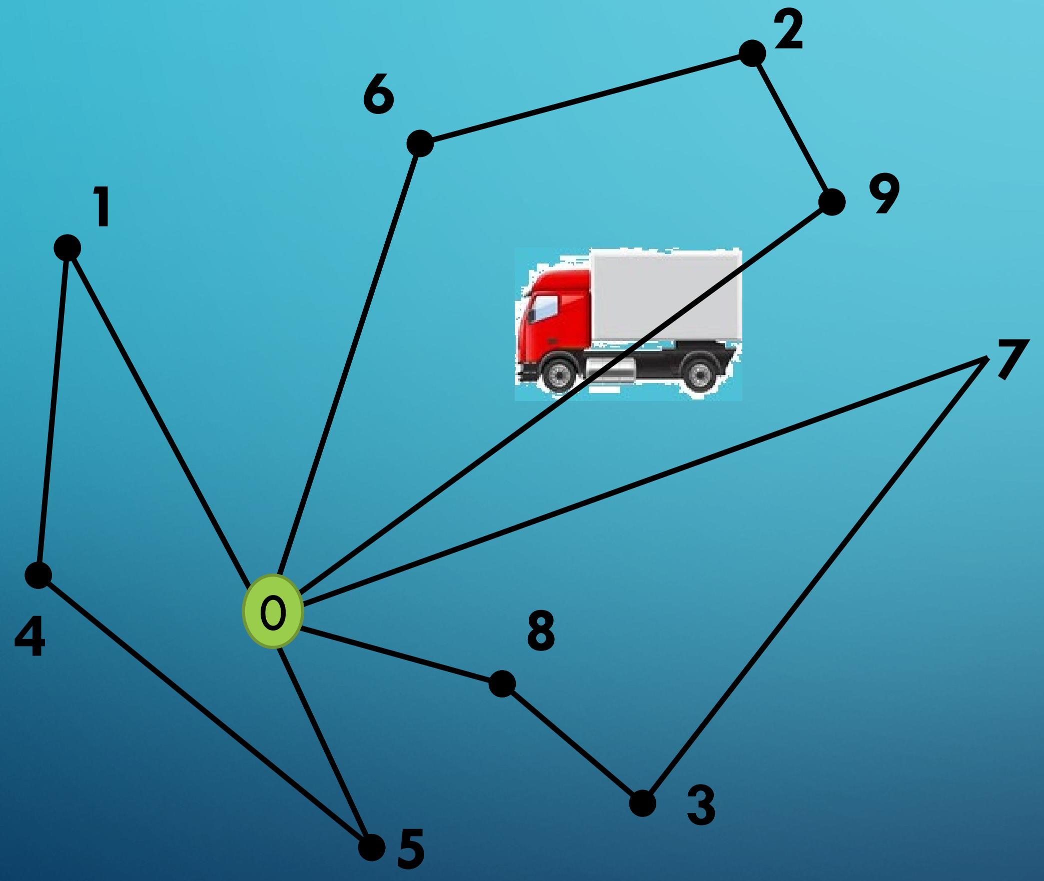
Среди вариантов VRP, CVRP является наиболее центральным и является тем, от которого происходят многие другие. Входные данные для CVRP состоят из *n* местоположений (депо и набор из $n-1$ клиентов), симметричной матрицы *D* размера $n \times n$, определяющей расстояние (или некоторые другие затраты), которое необходимо пройти между каждой парой местоположений, количество q_i, которое определяет потребность каждого клиента *i* в некотором ресурсе и максимальное количество *Q* ресурса, которое может перевозить транспортное средство.

Емкостный VRP Capacitated Vehicle Routing Problems

Допустимое решение для CVRP состоит из набора маршрутов, которые начинаются и заканчиваются в депо, так что каждый клиент посещается ровно по одному маршруту, а общая потребность клиентов, назначенных для маршрута, не превышает пропускную способность Q транспортного средства. Оптимальное решение для CVRP - это допустимое решение, которое сводит к минимуму общее комбинированное расстояние маршрутов.

Емкостный VRP Capacitated Vehicle Routing Problems

Фрагментарная структура допустимого решения



0 1 4 5 0 6 2 9 0 7 3 8 0

Фрагмент – номер клиента. Номер депо повторяется n+1 раз.

VRP с временными окнами VRP with Time Windows (VRPTW)

Как и CVRP, входные данные для VRPTW состоят из *n* местоположений (депо и набор из *n* -1 клиентов), матрицы *D,* определяющей расстояние (которое также будет временем) для перехода между каждой парой местоположений, количество q_i, которое определяет потребность каждого клиента *i в* некотором ресурсе и максимальное количество *Q* ресурса, которое может перевозить транспортное средство. Кроме того, для каждого узла *i* указывается отрезок времени s_i, обозначающий время, необходимое для обслуживания покупателя *i,* и временное окно [t_i , T_i], где $t_i < T_i$, в течение которого должна начаться доставка. Транспортному средству разрешено прибыть к месту нахождения клиента до начала временного окна, но он должен дождаться, пока окно «откроется», чтобы осуществить доставку. Доставка не может начаться после закрытия временного окна. Таким образом, задача реализации рассматривает ограничения временного окна как жесткие. Наконец, может существовать ограничение *V* на количество доступных транспортных средств.

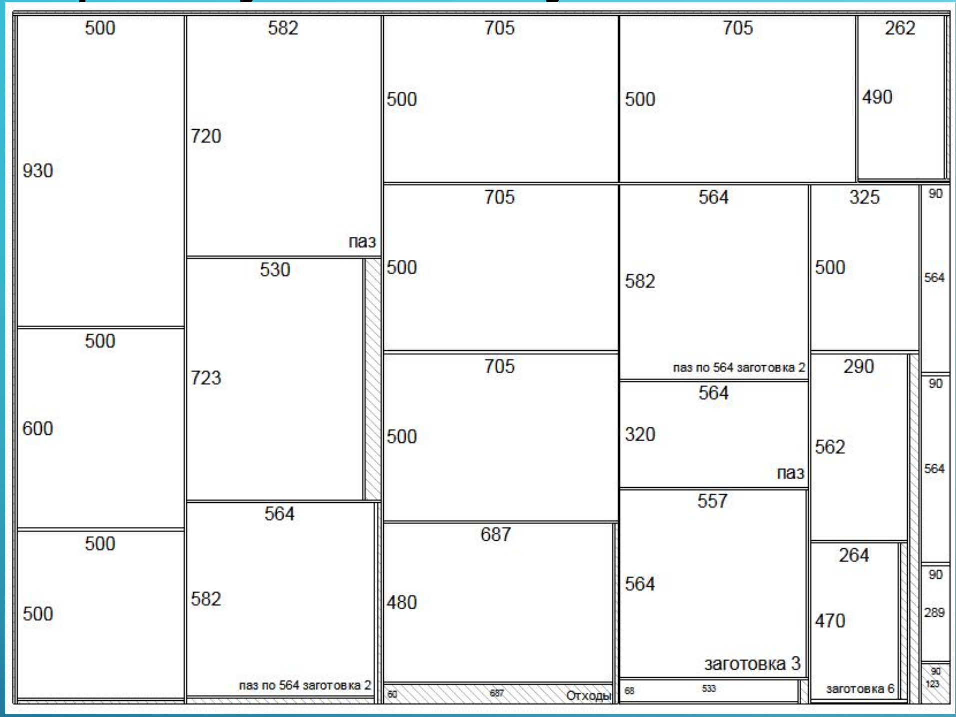
IRP проблема маршрутизации запасов Inventory Routing Problem (IRP)

Входные данные для IRP состоят из местоположений депо и набора из n клиентов $\{1,\ldots, n\}$, временного горизонта, состоящего из T периодов времени $\{1,\ldots, T\}$, и парка из M автомобилей $\{1,\ldots, M\}$. Депо находится в узле 0. Каждое транспортное средство имеет емкость Q .

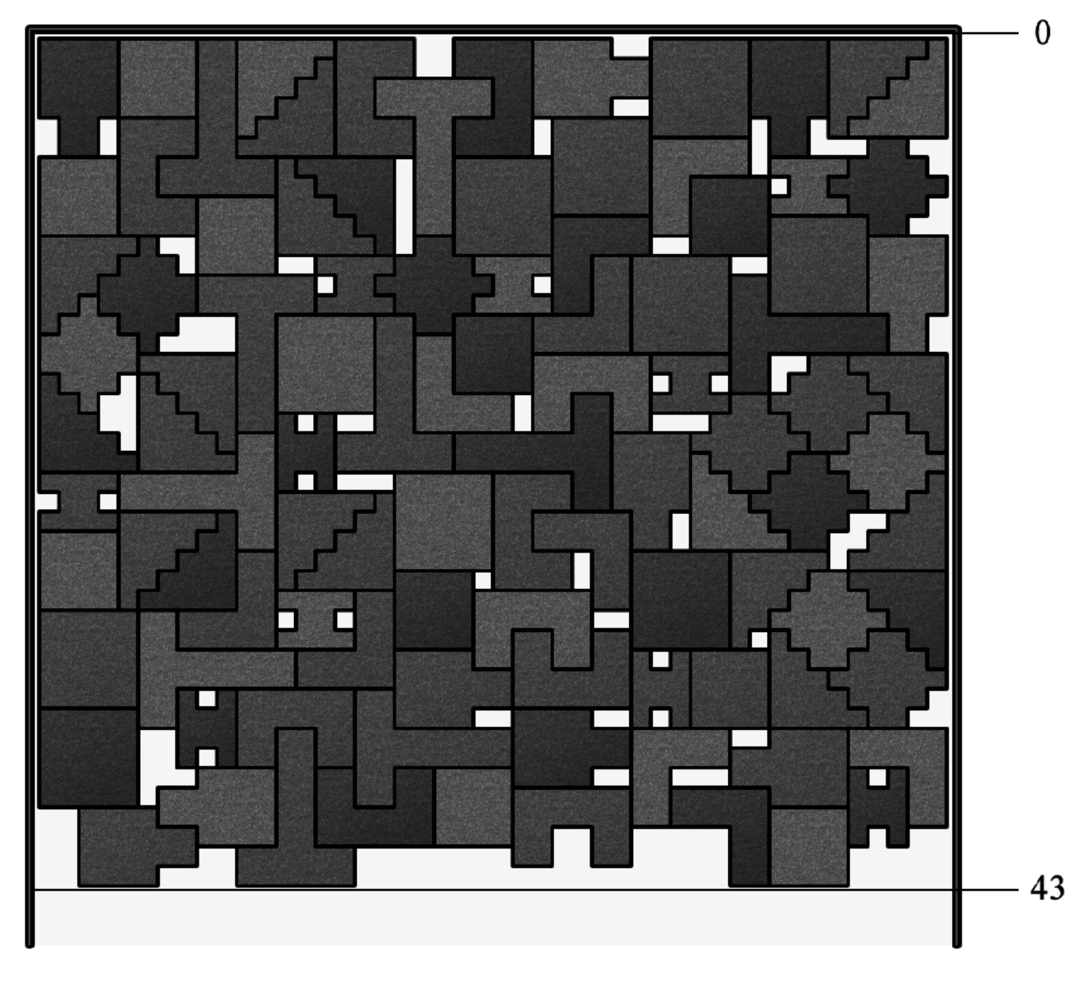
В каждый период времени t , r_{0t} единиц ресурса становятся доступными на складе, а r_{it} единиц потребляются *i-м* покупателем . Каждый узел i в $\{0,\ldots, n\}$ начинается с начального уровня запасов I_{i0} . Каждый покупатель $i \in \{1,\ldots, n\}$ должен поддерживать уровень запасов не менее L_i и не более U_i и будет нести затраты за период в размере h_i для каждой единицы запасов. Для депо не определен максимальный уровень запасов, но он не может упасть ниже 0, и существует стоимость хранения запасов за период h_0 .

Плоская упаковка

Прямоугольная упаковка

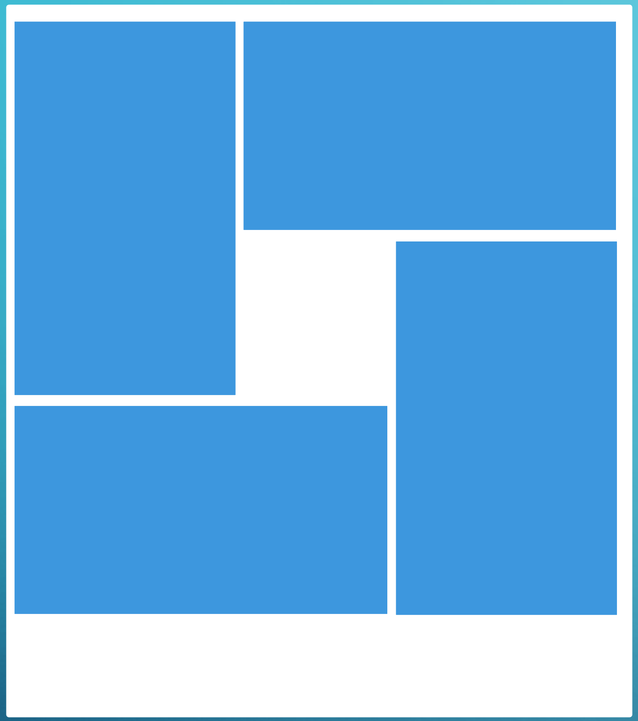


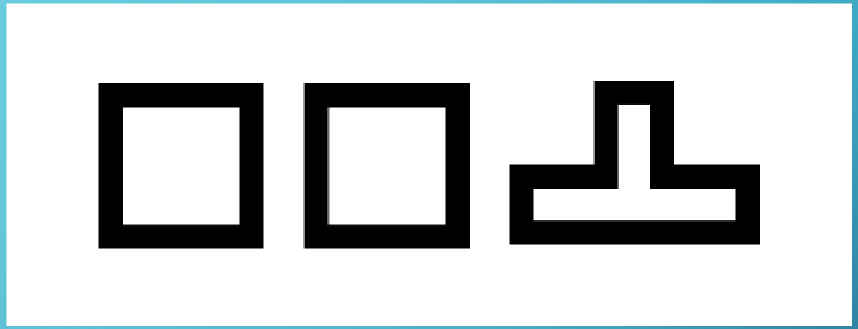
Упаковка сложных фигур

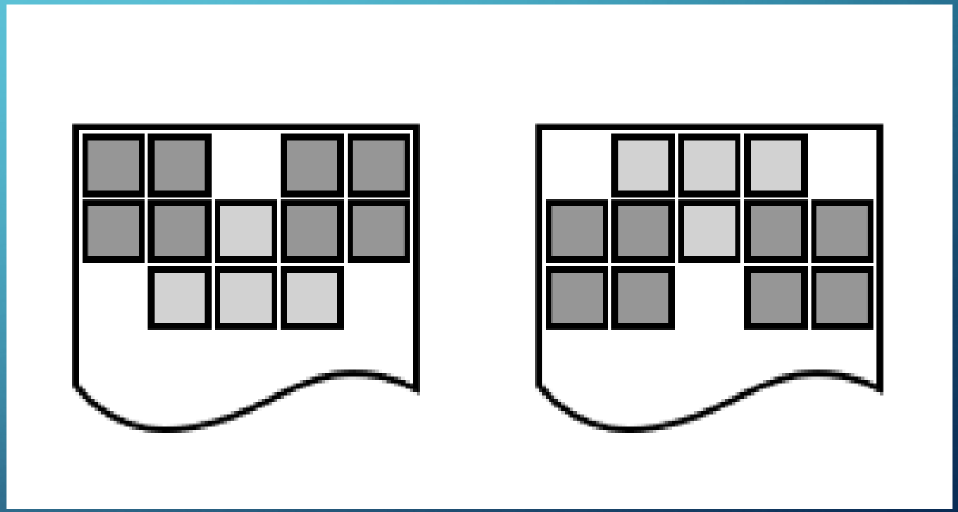


Упаковка плоских фигур

Маленькие неприятности

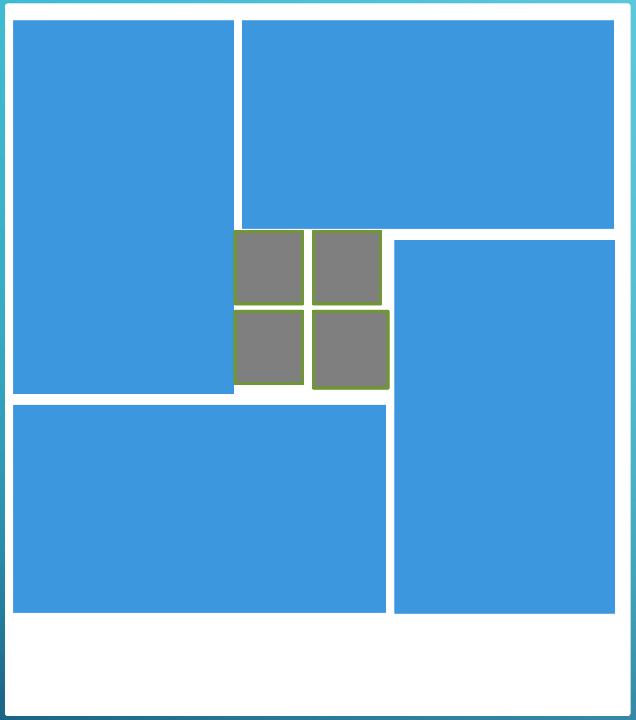


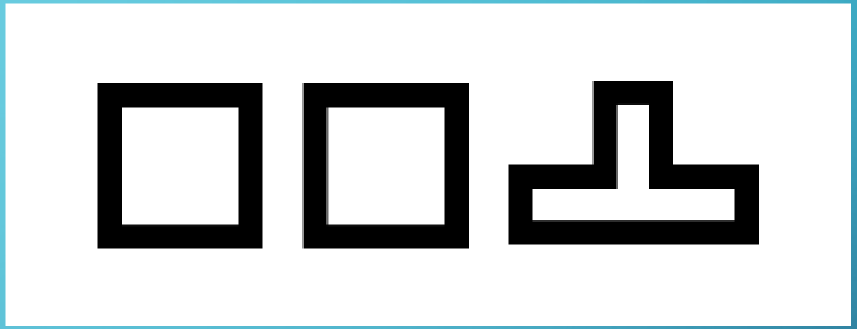


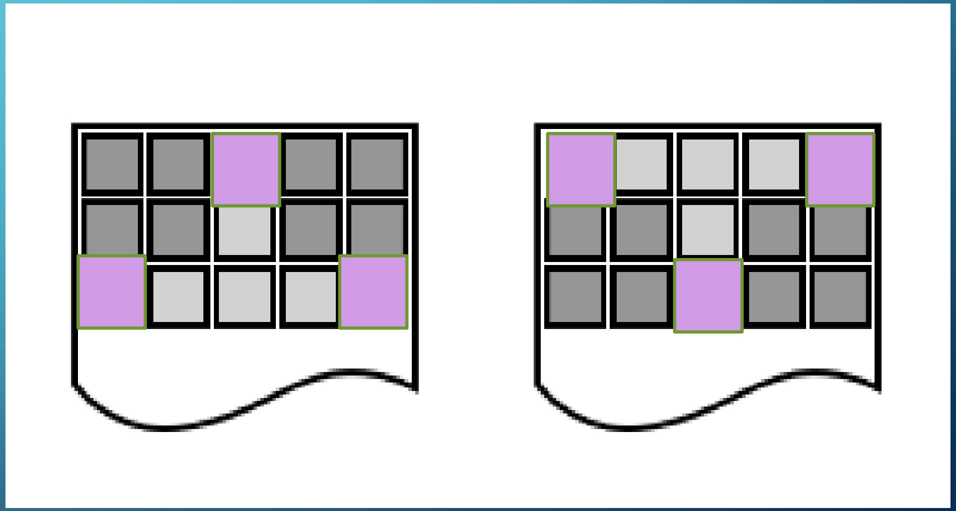


Упаковка плоских фигур

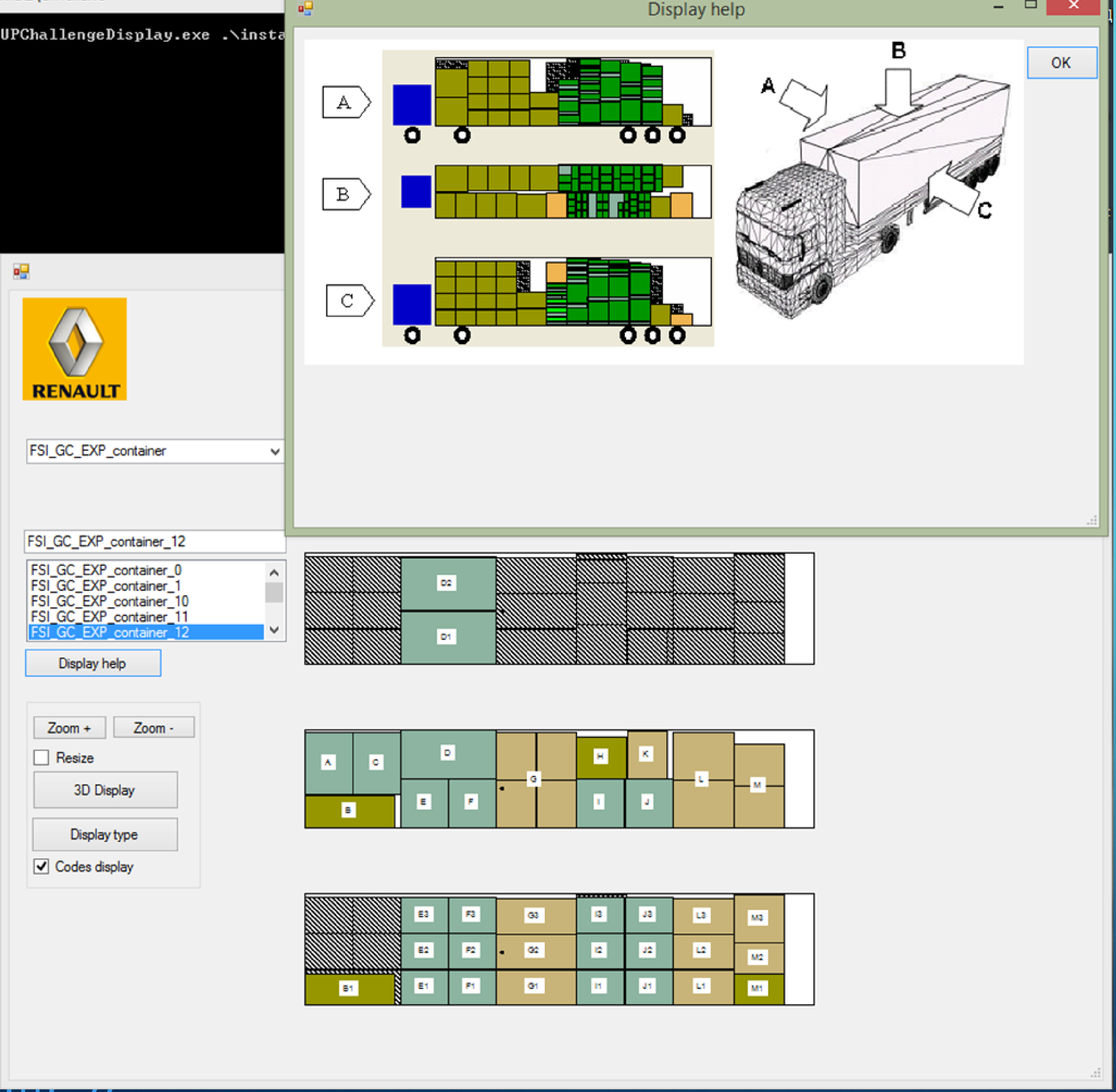
Добавление единичных квадратов







Примеры реальных загрузочных конфигураций компании Renault



ЦЕЛЕВАЯ ФУНКЦИЯ

Общий объем контейнеров, необходимых для перевозки заданного набора предметов

СПОСОБЫ УПАКОВКИ

1. предмет
2. ряд
3. слой
4. стек
5. контейнер

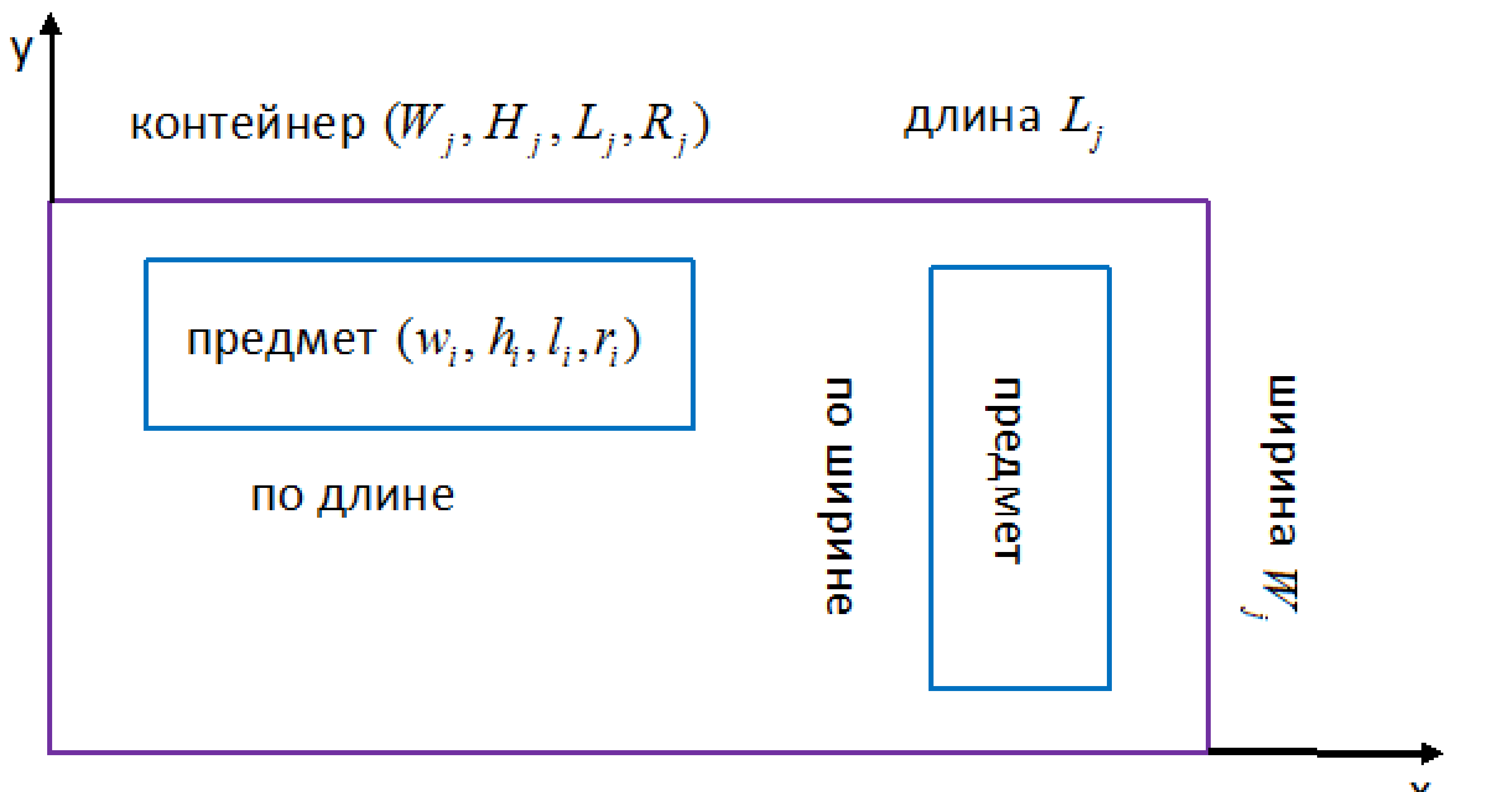


Рис. 1 Вид j-го контейнера сверху. Возможные ориентации предметов

Ряд – это непрерывная последовательность предметов одинаковой высоты, не сильно отличающихся по длине или ширине. Количество предметов в ряду ограничено.

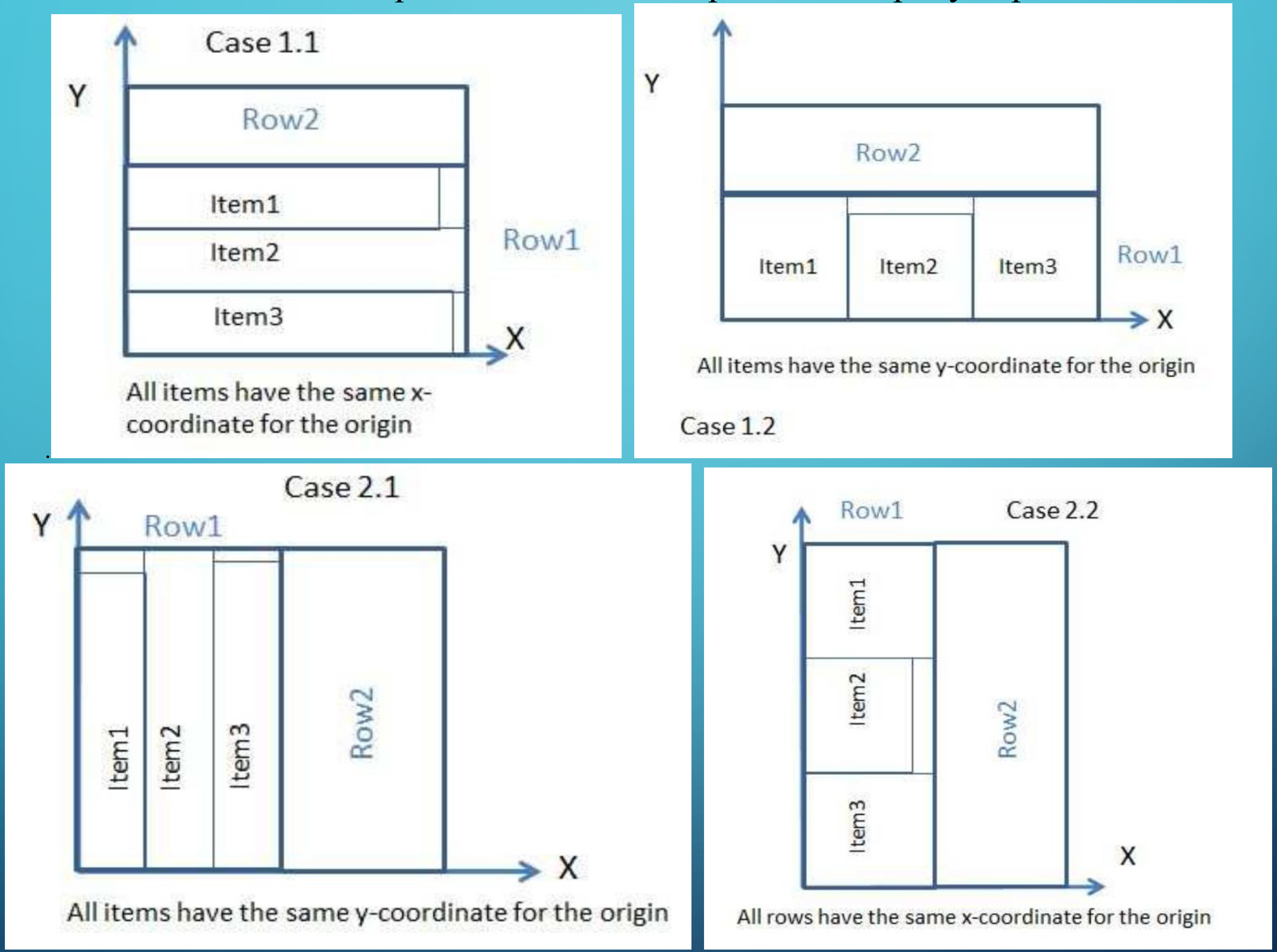


Рис. 2 Различные возможные конфигурации ряда с тремя предметами

Слой состоит из непрерывных рядов. Количество рядов в слое ограничено. Размеры рядов в направлении, перпендикулярном слою, одинаковы до заданной точности.

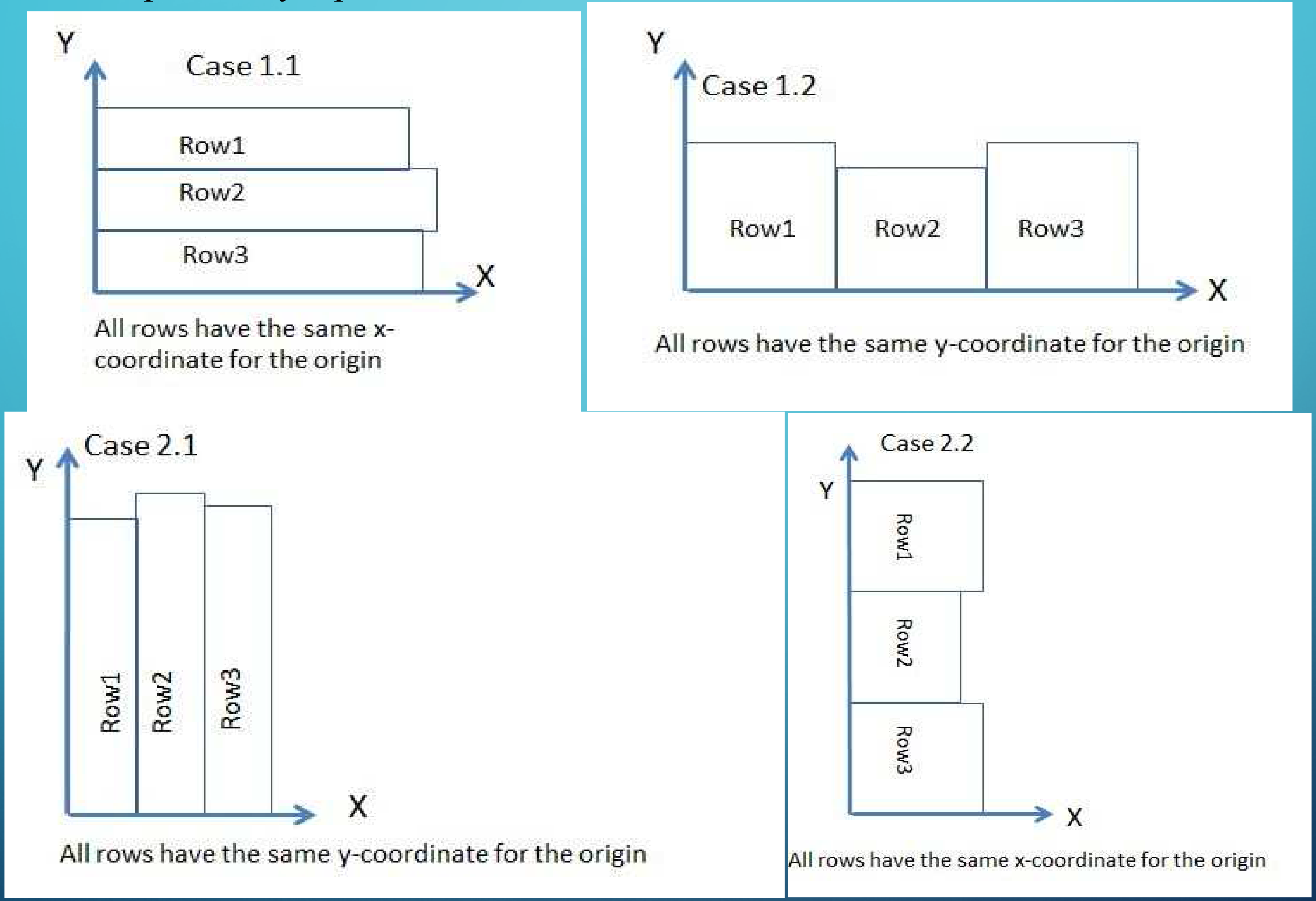


Рис. 3 Различные возможные конфигурации для слоя с тремя рядами

Ограничения

(**C1**) Одна или две возможные ориентации

(**C2**) Максимальный общий вес в бине

(**C3**) Стеки не могут перекрываться

(**C4**) Стеки находятся полностью в бинах

(**C5**) Каждый предмет упакован

(**C6**) Максимальное количество предметов, которое может быть упаковано в последний бин

(**C7**) Бин 0 имеет наименьший объем

(**C8**) Высота стека равна сумме высот его слоев

(**C9**) Максимальная общая высота

(**C10**) Слои в стеке почти одинакового размера

(**C11**) Прямоугольное место для стека вмещает в себя ортогональную проекцию его слоев

(**C12**) Металлические упаковки располагаются вместе в стеках

(**C13**) Максимальная плотность для каждого стека

(**C14**) Слои в стеке отсортированы в порядке убывания веса

(**C15**) Слои составлены из непрерывных рядов

(**C16**) Максимальное количество рядов в слое

(**C17**) Одинаковый размер рядов в слое

Ограничения

(**C18**) Все предметы в слое одинакового размера

(**C19**) Ряды выровнены по слою

(**C20**) Контур слоя – это прямоугольник, охватывающий его ряды

(**C21**) Ряды составлены из стоящих подряд без попусков предметов

(**C22**) Максимальное количество предметов в ряду

(**C23**) Одинаковый горизонтальный размер предметов в ряду

(**C24**) Предметы выровнены по ряду

(**C25**) Контур ряда – это прямоугольник, охватывающий его предметы

(**C26**) Соседние слои располагаются без пропусков в вертикальном направлении

(**C27**) Вершина стека – это вершина его самого верхнего слоя

(**C28**) Слои составленные из металлических упаковок могут содержать только один предмет

(**C29**) Максимальный вес, помещенный на предметы базового уровня.

Спасибо за внимание!