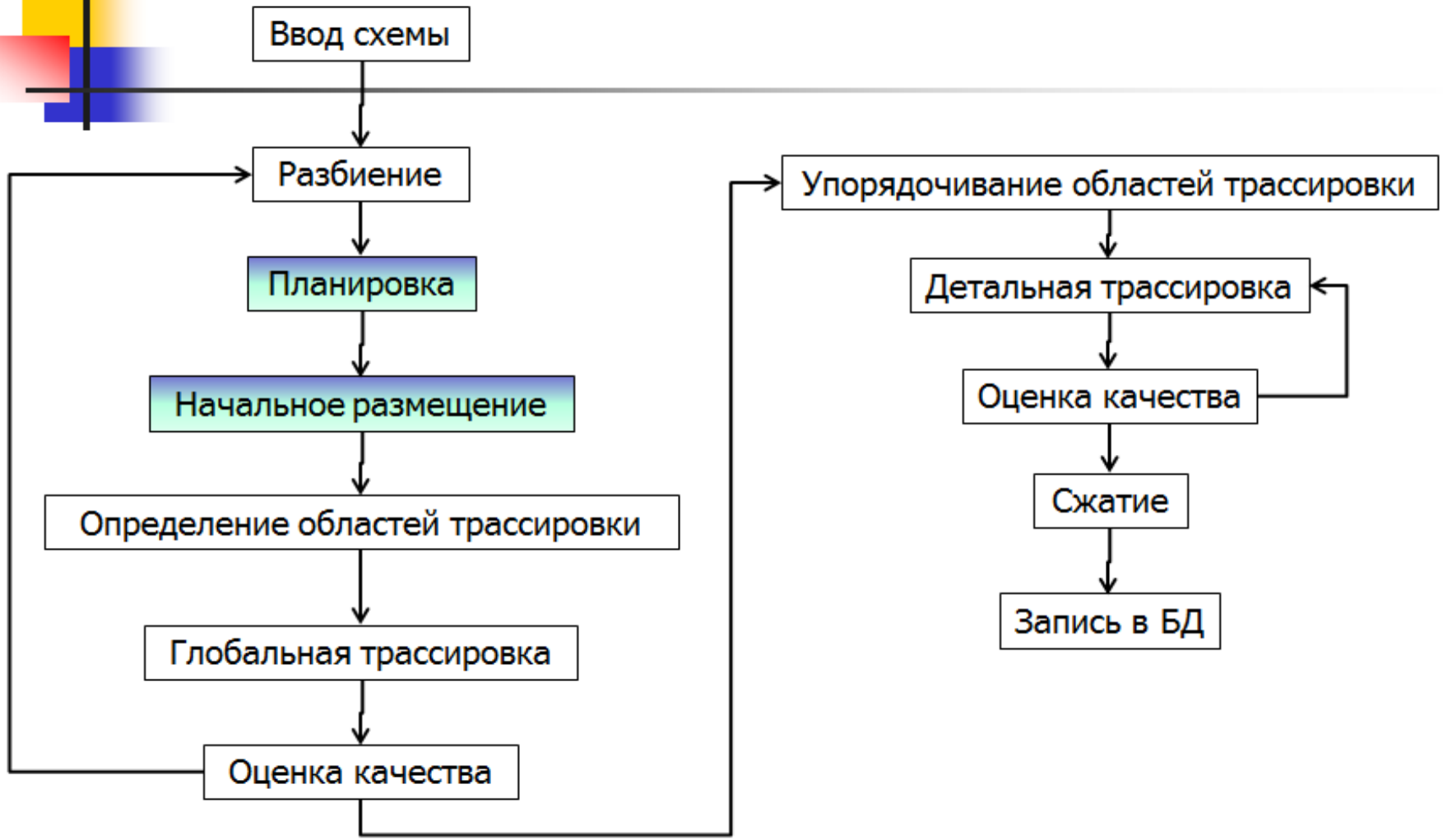
Лекции 5-6

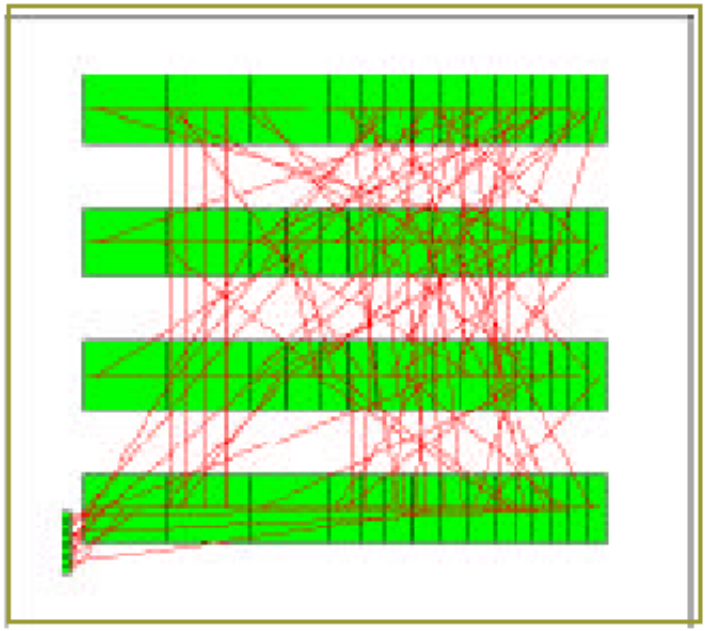
Содержание

- Сведения о КМОП технологии и методология проектирования СБИС
- Задача разбиения электрической схемы
- Задача размещения модулей СБИС
- Задача трассировки соединений

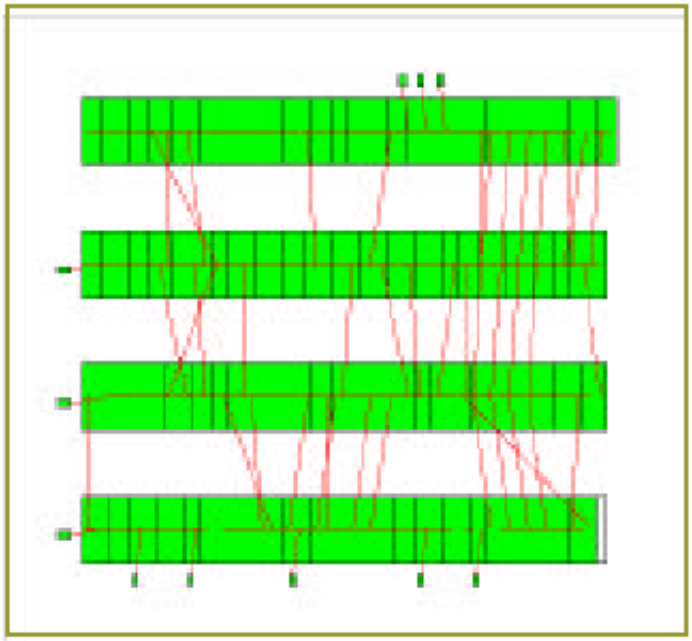
Маршрут проектирования топологии СБИС



Проблема размещения

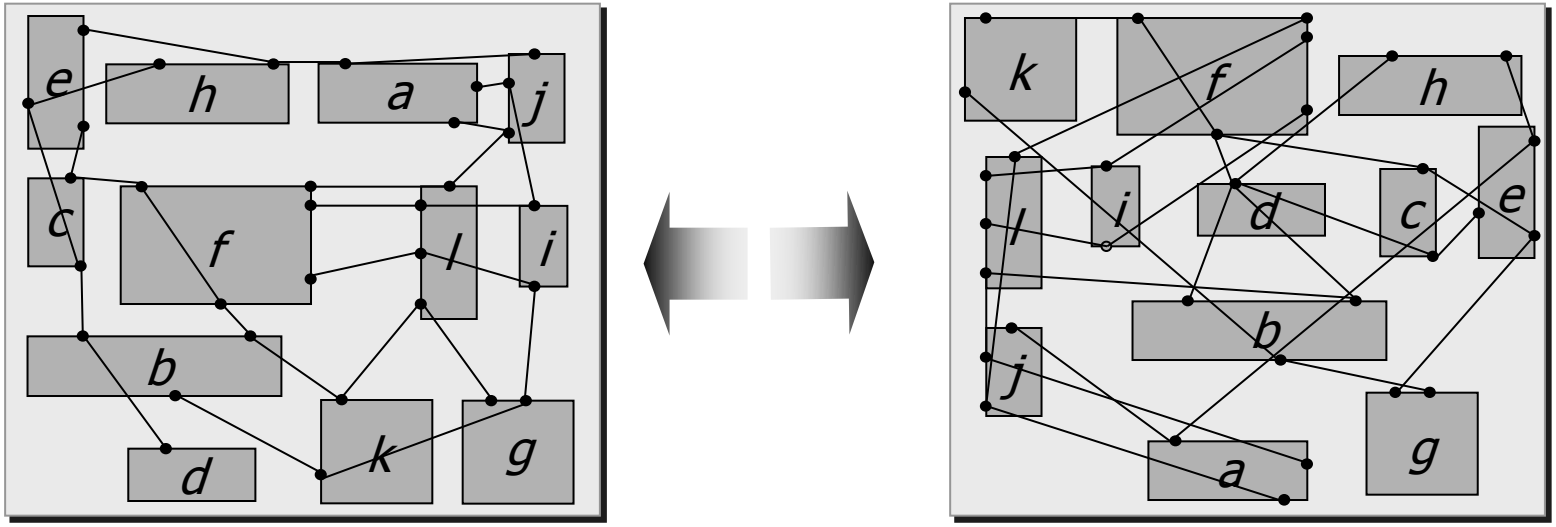


A bad placement



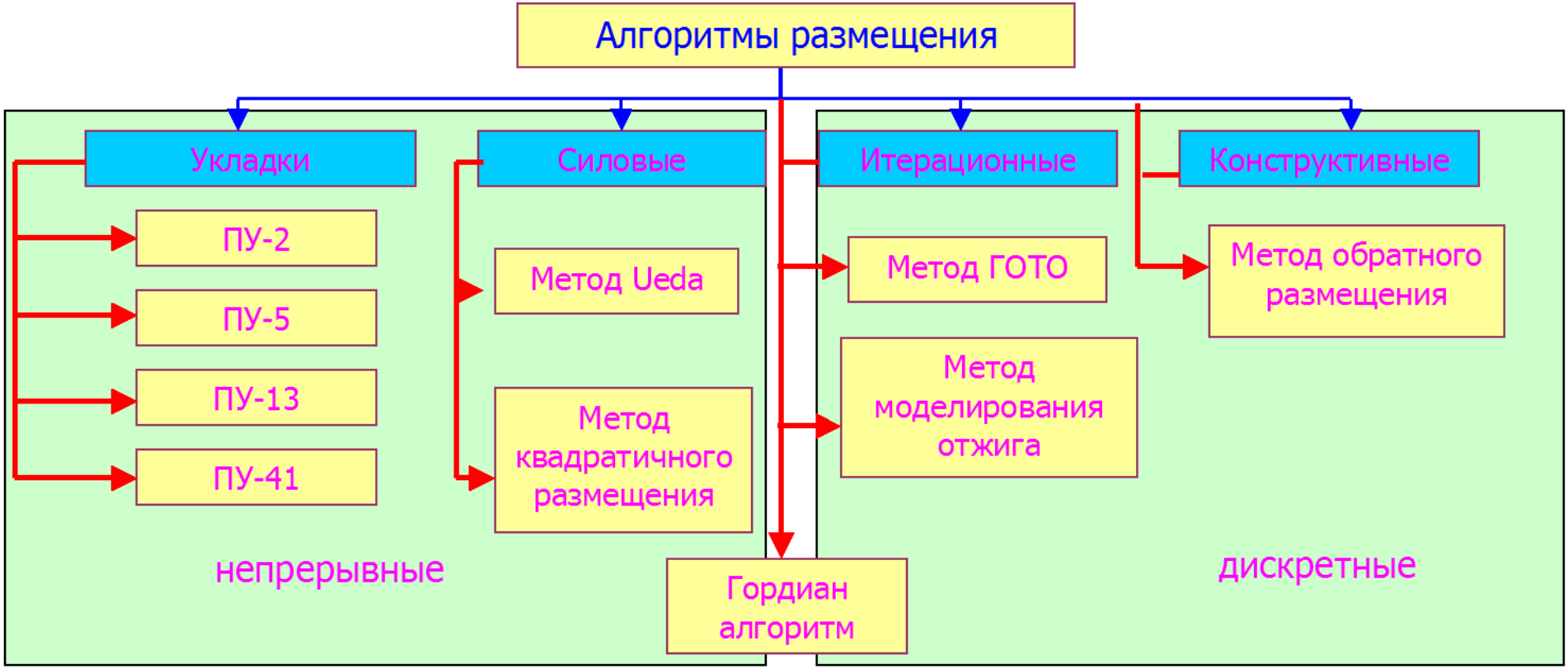
A good placement

Суммарная длина соединений (Total Wirelength)



© 2011 Springer Verlag

Классификация алгоритмов размещения

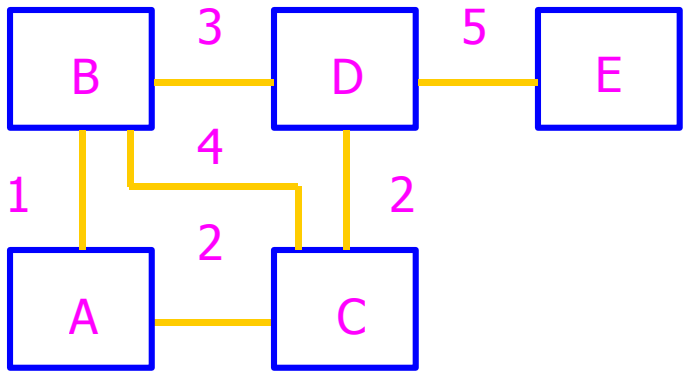


Дискретные методы размещения

- Формулировка задачи
- Обратное размещение
- Метод ветвей и границ

Метод обратного размещения

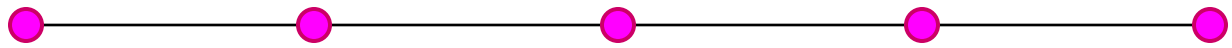
- Классификация взаимодействий между блоками



$$C = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 4 & 3 & 0 \\ 2 & 4 & 0 & 2 & 0 \\ 0 & 3 & 2 & 0 & 5 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix}$$

Метод обратного размещения

- Матрица расстояний:



$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix} \begin{matrix} 10 \\ 7 \\ 6 \\ 7 \\ 10 \end{matrix}$$

Суммы элементов строк

D^+

Метод обратного размещения

Алгоритм

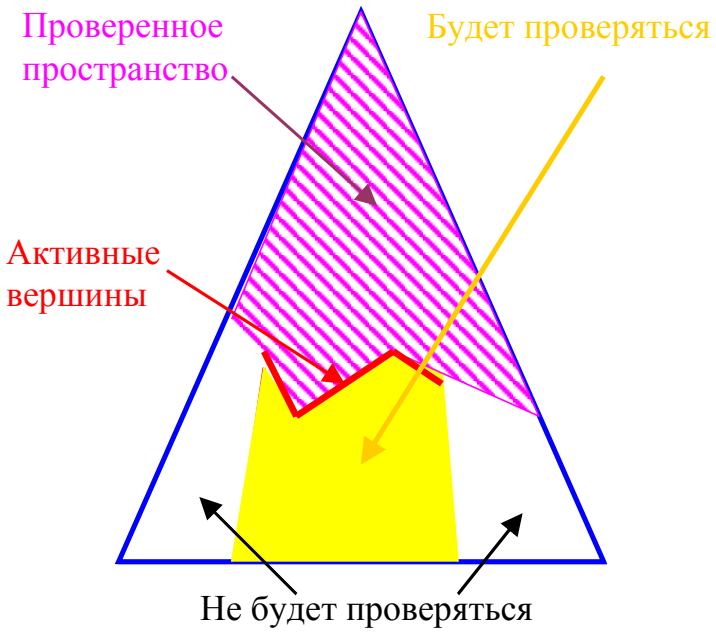
1. Упорядочить E^+ по убыванию
2. Упорядочить D^+ по убыванию
3. Разместить элемент ***i*** на позицию с индексом ***i***.

Метод ветвей и границ

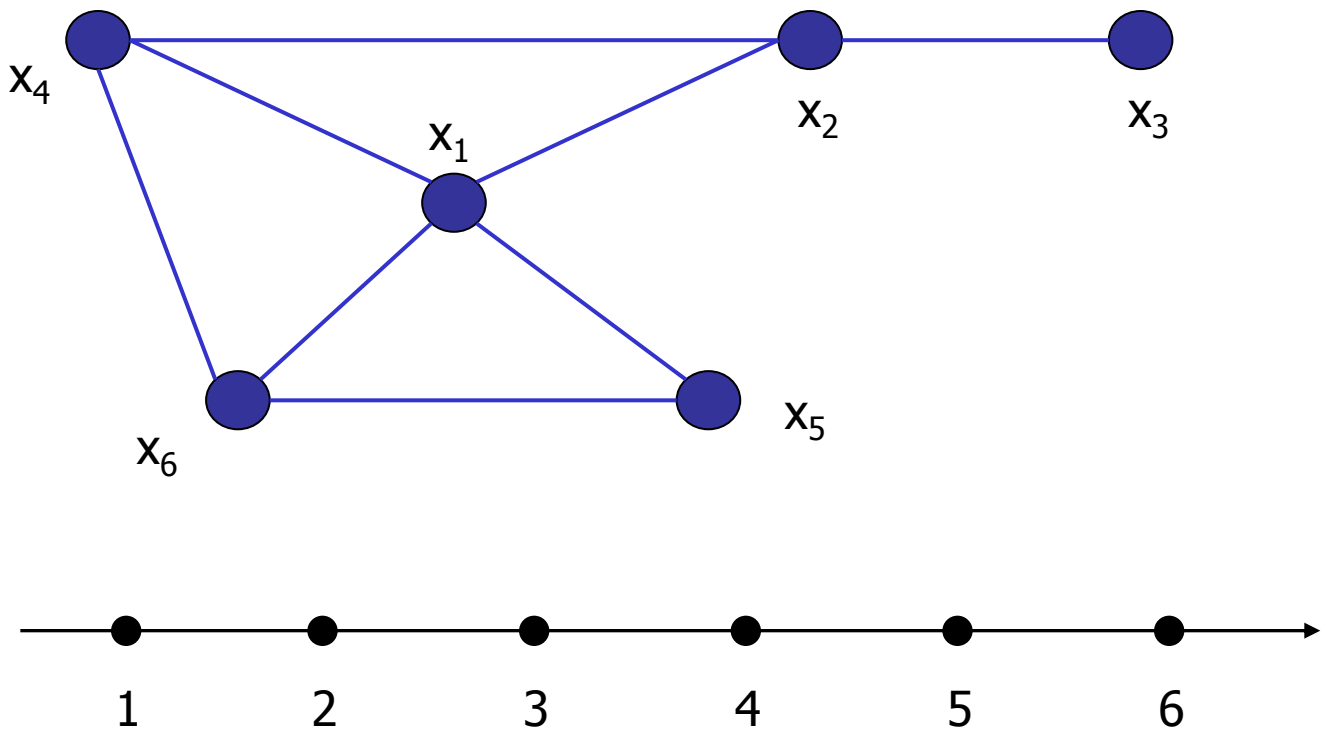
- Определение: Конфигурационное пространство задачи минимизации является древовидным (tree-structured) если можно определить дерево T, с вершинами z которого ассоциируются подмножества конфигураций S_z, такие, что:
  1. Если r – корень, то S_r содержит оптимальное решение
  2. Если z – внутренняя вершина T и $z1,\ldots,zk$ – дочерние вершины z, то $\cup S_{zi} = S_z$
  3. Если z – вершина T, то существует нижняя граница $L_z \leq c(s)$ для $\forall s \in S_z$
  4. Если z – вершина T, то существует верхняя граница такая, что $U_z = \infty$ или $U_z = c(s_z)$, если z – лист, то $L_z \leq U_z$

Метод ветвей и границ (продолжение)

begin
  $u \leftarrow \infty$, CurrentBest $\leftarrow \varnothing$, вставить (r,L_r) в D
  repeat выделить следующий (z,L_z) из D
    $l \leftarrow \infty$
    for z' дочерней вершины z *do*
      вычислить $L_{z'}$ and $U_{z'}$
      if $U_{z'}<u$ *then*
        CurrentBest $\leftarrow s_{z'}$, $u \leftarrow U_{z'}$
        удалить все $(z'',L_{z''})$ такие, что $L_{z''}\geq u$ из D
      if $L_{z'}<u$ *then*
      вставить $(z',L_{z'})$ в D
      $l \leftarrow \min(l,L_{z'})$
  until $D = \varnothing$
end



Решение задачи линейного размещения методом ветвей и границ



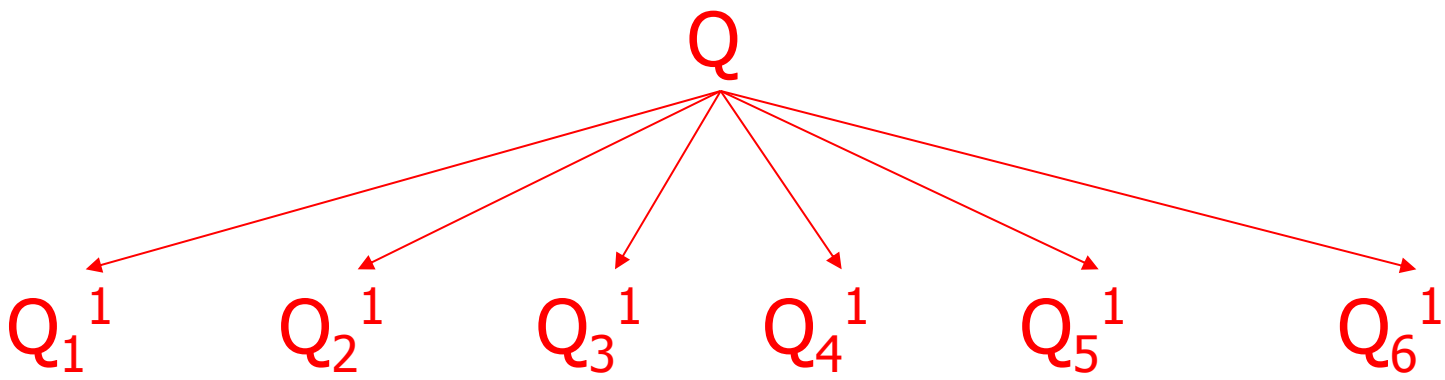
Правило ветвления, нижняя оценка, целевая функция

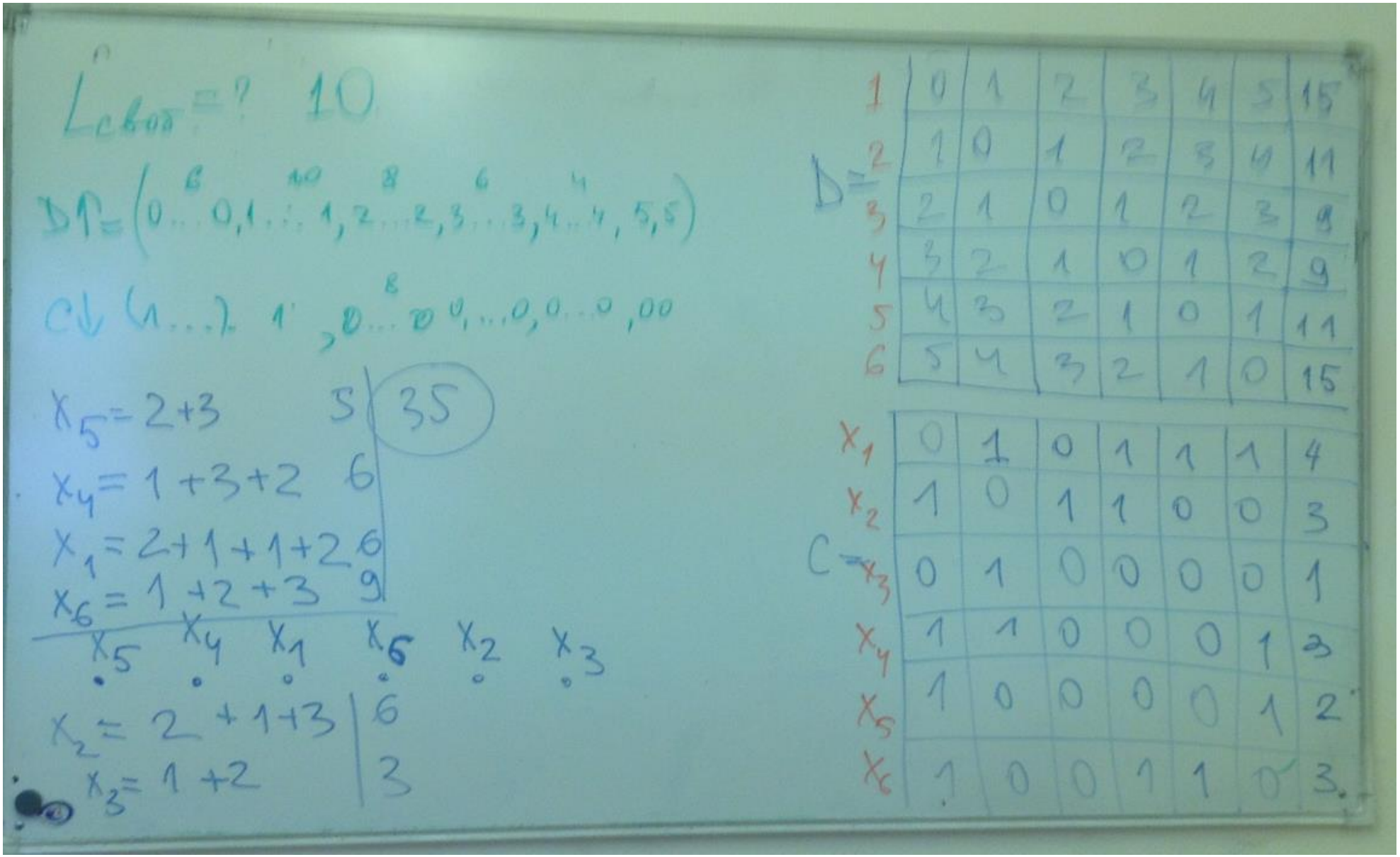
$$F(p) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} r_{p(i)p(j)} = \frac{1}{2}(\vec{C}\,\vec{R})$$

$$(\vec{A}\,\vec{B}) \geq (increase(\vec{A})decrease(\vec{B}))$$

$$L = L_{фикс} + L_{своб} + L_{фс}$$

Ветвление




Lсвоб = ? 10
D↑ = (0...0,1...1,2...2,3...3,4...4,5,5)
C↓ (1...) 1 , 0...0 0,...0,0...0,00
x5 = 2+3 5 35
x4 = 1+3+2 6
x1 = 2+1+1+2 6
x6 = 1+2+3 9
x5 x4 x1 x6 x2 x3
x2 = 2+1+3 6
x3 = 1+2 3
D =
1 0 1 2 3 4 5 15
2 1 0 1 2 3 4 11
3 2 1 0 1 2 3 9
4 3 2 1 0 1 2 9
5 4 3 2 1 0 1 11
6 5 4 3 2 1 0 15
C =
x1 0 1 0 1 1 1 4
x2 1 0 1 1 0 0 3
x3 0 1 0 0 0 0 1
x4 1 1 0 0 0 1 3
x5 1 0 0 0 0 1 2
x6 1 0 0 1 1 0 3

Непрерывные методы размещения

- Формулировка задачи
- Методы планировки
- Силовые алгоритмы

Гильотинные алгоритмы ПУ-2

1. Определение размеров поля для размещения блоков:

$$L_x = L_y = \sqrt{\sum S_i}$$

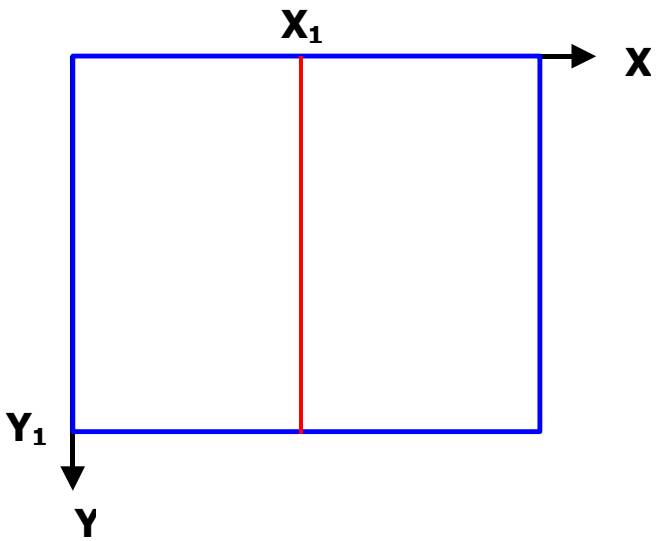
2. Собственно размещение:

$S_1 = x_1 L_y$

$S_2 = (L_x - x_1) L_y$

$L_x = L$

$0 < x_1 < L_x$



Негильотинные алгоритмы ПУ-5

$S_1=x_2y_1$

$S_2=(L_x-x_2)y_2$

$S_3=(L_x-x_1)(L_y-y_2)$

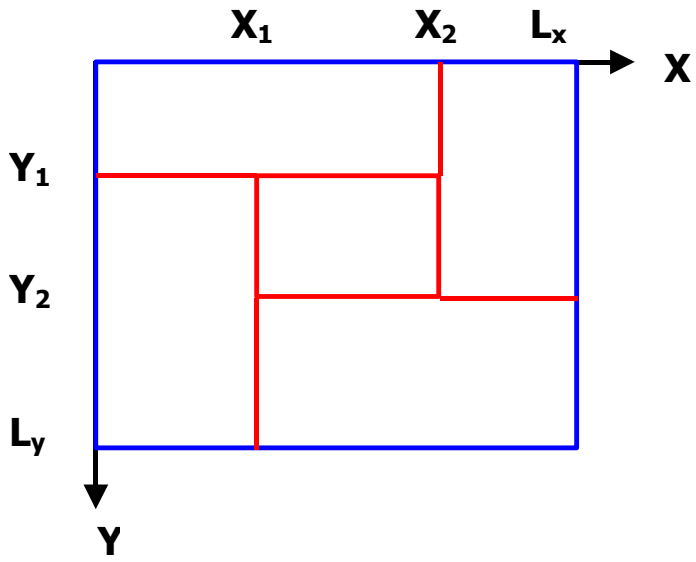
$S_4=x_1(L_y-y_1)$

$S_5=(x_2-x_1)(y_2-y_1)$

$L_x=L$

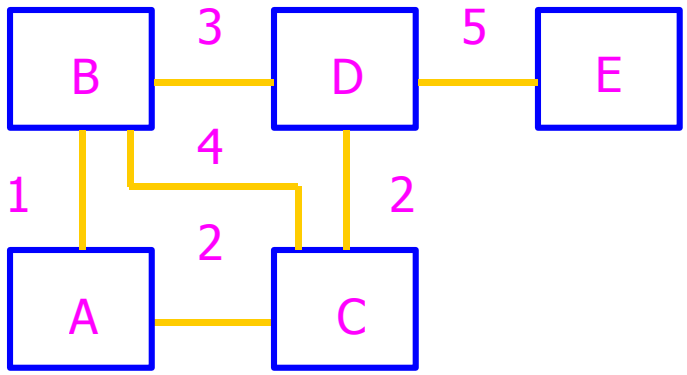
$0<x_1< x_2< L_x$

$0<y_1< y_2< L_x$



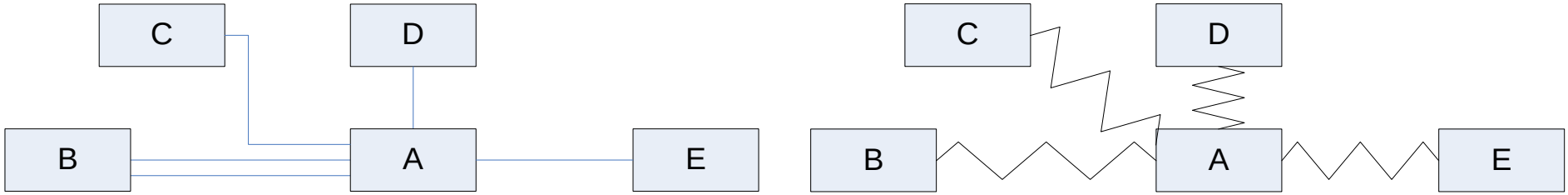
Силовой алгоритм размещения

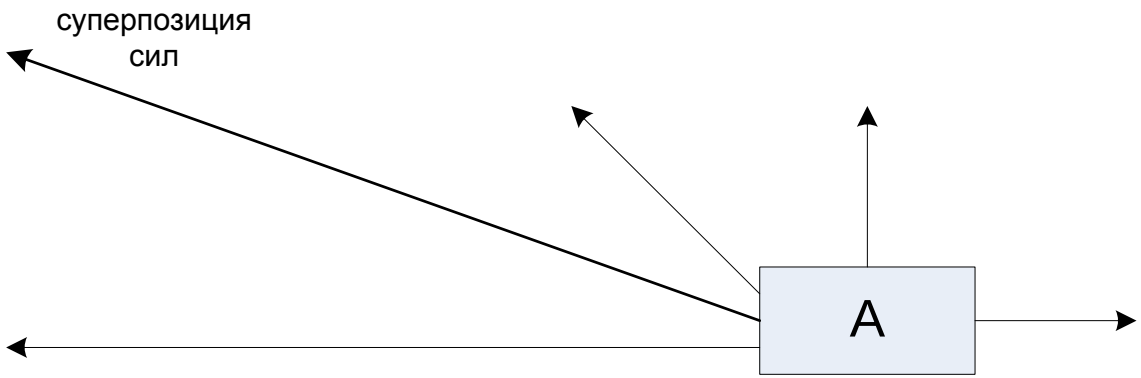
- Классификация взаимодействий между блоками



$$C = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 4 & 3 & 0 \\ 2 & 4 & 0 & 2 & 0 \\ 0 & 3 & 2 & 0 & 5 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix}$$

Силовой алгоритм. Идея – суперпозиция сил.





Силовой алгоритм размещения

- Вычисление расширенной матрицы связей

$$e_{ij} = \begin{cases} \min(p_{ij}), i \neq j \\ 0, i = j \end{cases}$$

$$p_{12}(a,b) = \frac{1}{c_{ab}} = 1$$

$$p_{12}(a,b,c) = \frac{1}{c_{ac}} + \frac{1}{c_{cb}} = \frac{1}{2} + \frac{1}{4} = 0.75$$

$$p_{12}(a,c,d,b) = \frac{1}{c_{ac}} + \frac{1}{c_{cd}} + \frac{1}{c_{db}} = 1.33$$

$$E = \begin{bmatrix} 0.00 & 0.75 & 0.50 & 1.08 & 1.28 \\ 0.75 & 0.00 & 0.25 & 0.33 & 0.53 \\ 0.50 & 0.25 & 0.00 & 0.50 & 0.70 \\ 1.08 & 0.33 & 0.50 & 0.00 & 0.20 \\ 1.28 & 0.53 & 0.70 & 0.20 & 0.00 \end{bmatrix}$$

Силовой алгоритм размещения

- Вычисление матрицы отношений между блоками

$$\overline{E} = \frac{1}{n^2 - n} \sum_{i=1, i \neq j}^{n} \sum_{j=1}^{n} e_{ij}$$

$$r_{ij} = \begin{cases} e_{ij} - \overline{E}, i \neq j \\ 0, i = j \end{cases}$$

$$R = \begin{bmatrix} 0.00 & 0.14 & -0.11 & 0.47 & 0.67 \\ 0.14 & 0.00 & -0.36 & -0.28 & -0.08 \\ -0.11 & -0.36 & 0.00 & -0.11 & 0.09 \\ 0.47 & -0.28 & -0.11 & 0.00 & -0.41 \\ 0.67 & -0.08 & 0.09 & -0.41 & 0.00 \end{bmatrix}$$

Критерий сильной связности: $r_{ij} < 0$

Силовой алгоритм размещения

- Вычисление сил притяжения и отталкивания между блоками

$$F^{+} = \begin{cases} k(r-a), r > a \\ 0, 0 \le r \le a \end{cases}$$

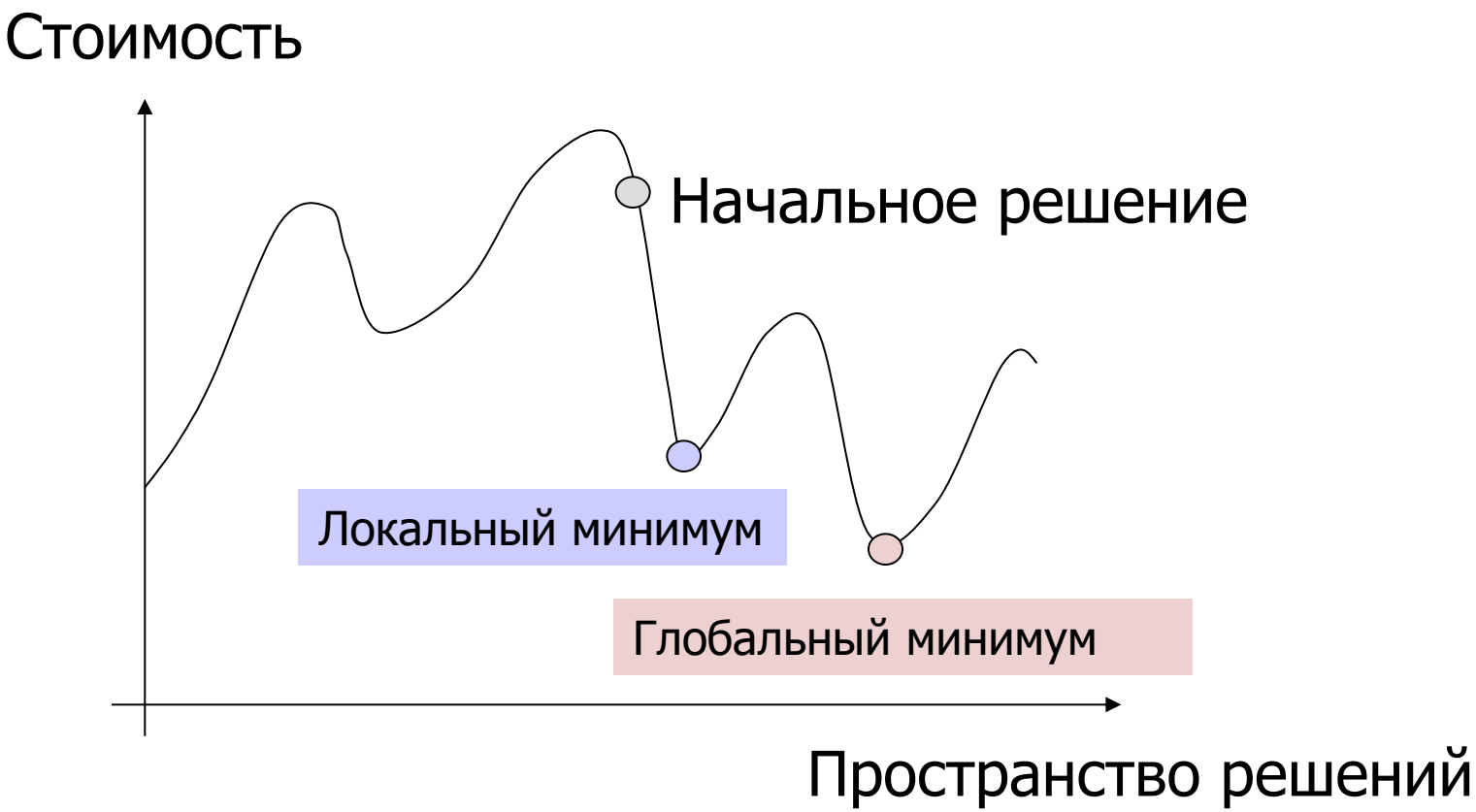
$$F^{-} = \begin{cases} -k(r-a), 0 < r < a \\ 0, r \ge a \end{cases}$$

Simulated Annealing (Моделирование отжига)

- Это итерационный алгоритм.
- В отличие от жадных алгоритмов, SA может принимать в качестве кандидата решение с худшим значением целевой функции.

25

Моделирование отжига



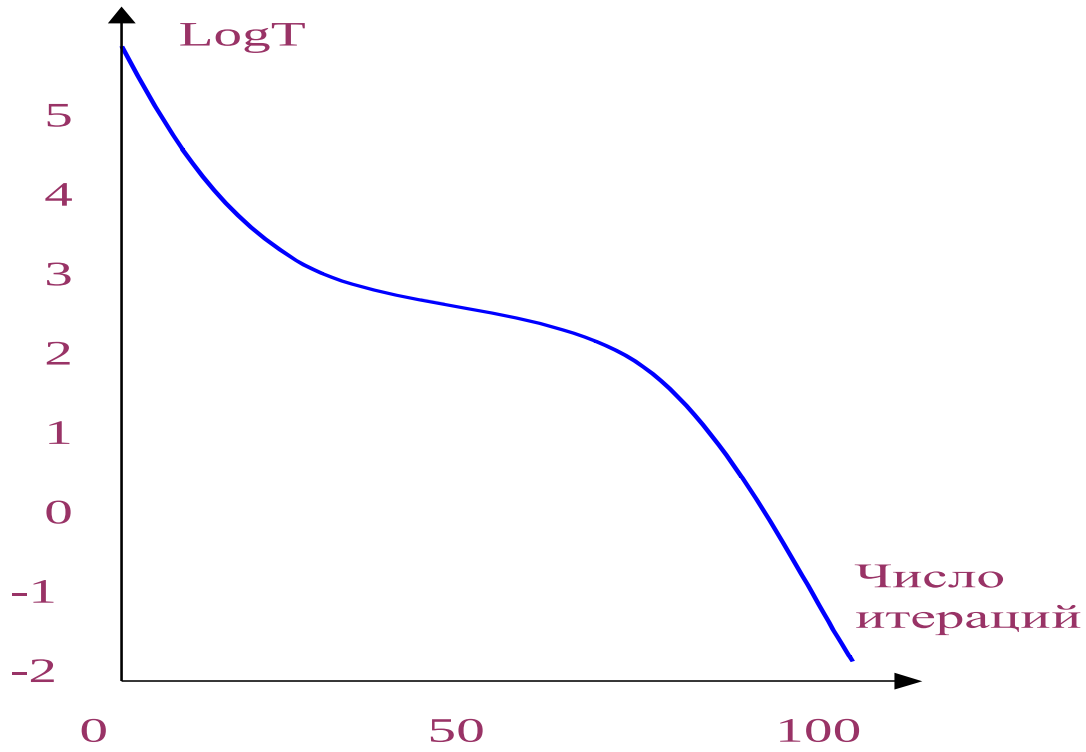
Моделирование отжига

- Generate an initial solution S_{init}, and evaluate its cost.
- Generate a new solution S_{new} by performing a random walk
- S_{new} is accepted or rejected based on the temperature T
  - Higher T means a higher probability to accept S_{new} if $COST(S_{new}) > COST(S_{init})$
  - T slowly decreases to form the final solution
- Boltzmann acceptance criterion, where r is a random number [0,1)

$$e^{\frac{COST(S_{init})-COST(S_{new})}{T}} > r$$

27

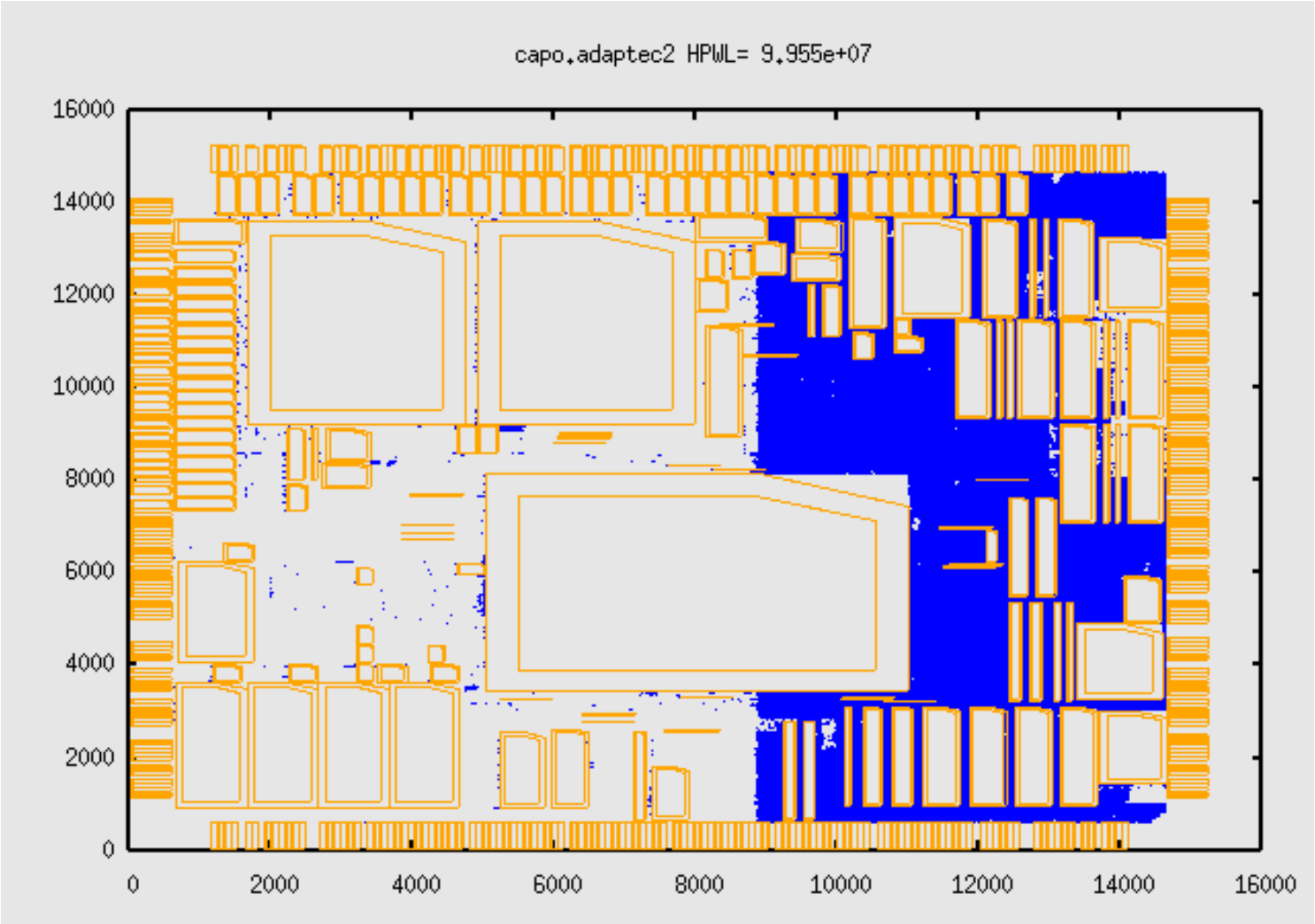
Моделирование отжига, дополнение



Современное состояние проблемы

- Классификация алгоритмов
- Примеры

- Дихотомические – Capo
  - Jarrod A. Roy, David A. Papa, Saurabh N. Adya, Hayward H. Chan, Aaron N. Ng, James F. Lu, Igor L. Markov "Capo: Robust and scalable open-source min-cut floorplacer", ISPD'05, 2005, p.224-226

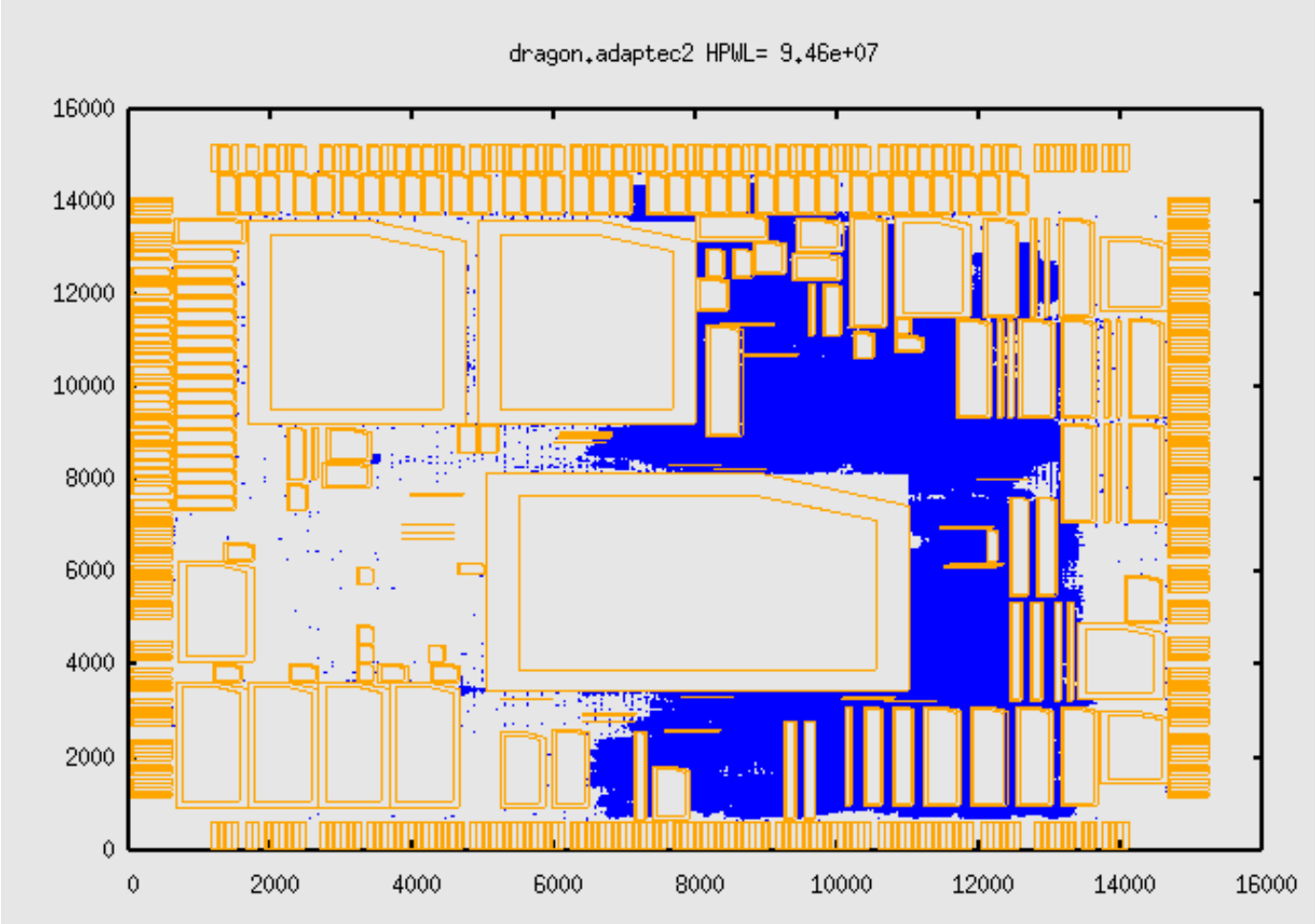


Capo – хорошо известная академическая программа размещения разработанный в Калифорнийском университете (UCLA – University of California, Los Angeles). Исходные тексты Capo доступны в Интернете.

- Capo реализует рекурсивный алгоритм бисекции (min-cut recursive approach). Этот алгоритм основан на подходе “разделяй и властвуй” (divide and conquer). Идея алгоритма состоит в том, что схема и площадь кристалла разбиваются на две части, такие, что количество соединений между этими частями минимально. Затем задача решается для каждой из частей отдельно аналогичным способом, т.е. каждая часть так же разбивается на две части, и т.д. Рекурсивное разбиение заканчивается тогда, когда размер каждой части станет достаточно малым и на этом уровне задача решается точно для каждой из полученных частей.
- Для задачи разбиения на две части с минимальным количеством соединений не существует эффективных алгоритмов решающих задачу точно, поэтому используются различные эвристические алгоритмы, причем Capo использует различные эвристики, в зависимости от размеров разбиваемой части: на верхнем уровне применяются алгоритмы, которые работают быстрее, но решают задачу хуже, чем алгоритмы, используемые на нижнем уровне

Моделирования отжига – Dragon

- Taraneh Taghavi, Xiaojian Yang, Bo-Kyung Choi “Dragon2005: large-scale mixed-size placement tool”, ISPD’05, 2005, p.245-247



Dragon разработан в северо-западном университете (Northwestern University) и доступен в Интернете

- Dragon использует рекурсивное деление на четыре части с минимальным разбиением (recursive min-cut quadrisection), и затем, для детального размещения используется моделирование отжига.
- Первая стадия аналогична алгоритму используемому в Capo: на первой стадии схема и площадь кристалла разбиваются на четыре части, каждая из которых потом также разбивается на четыре части, и т.д. до тех пор, пока для полученных малых частей нельзя будет решить задачу точно

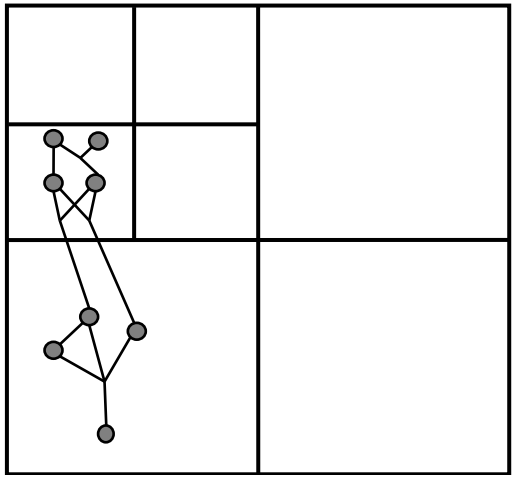
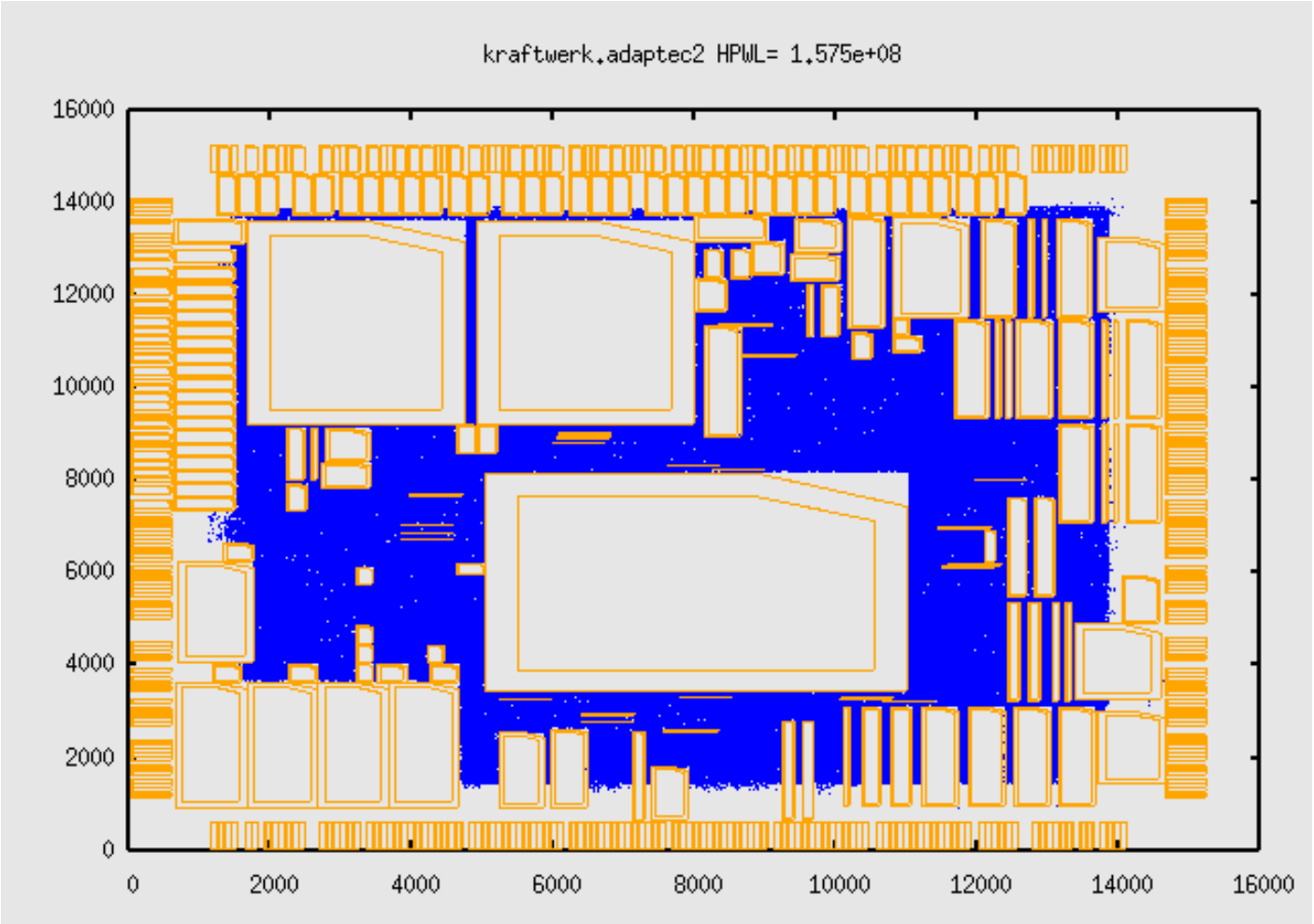
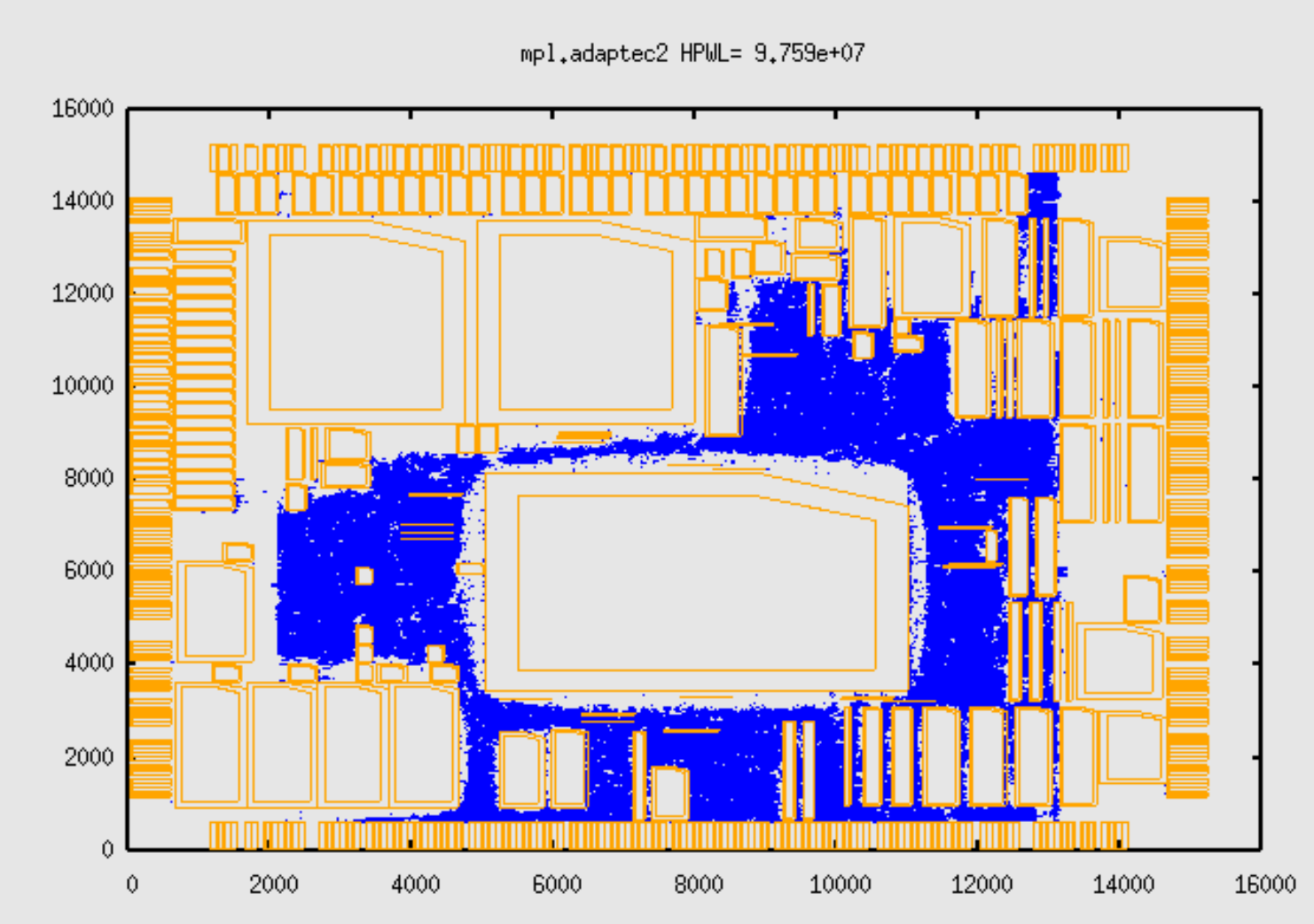


Рис. : Рекурсивное разбиение с минимальным пересечением (Dragon)

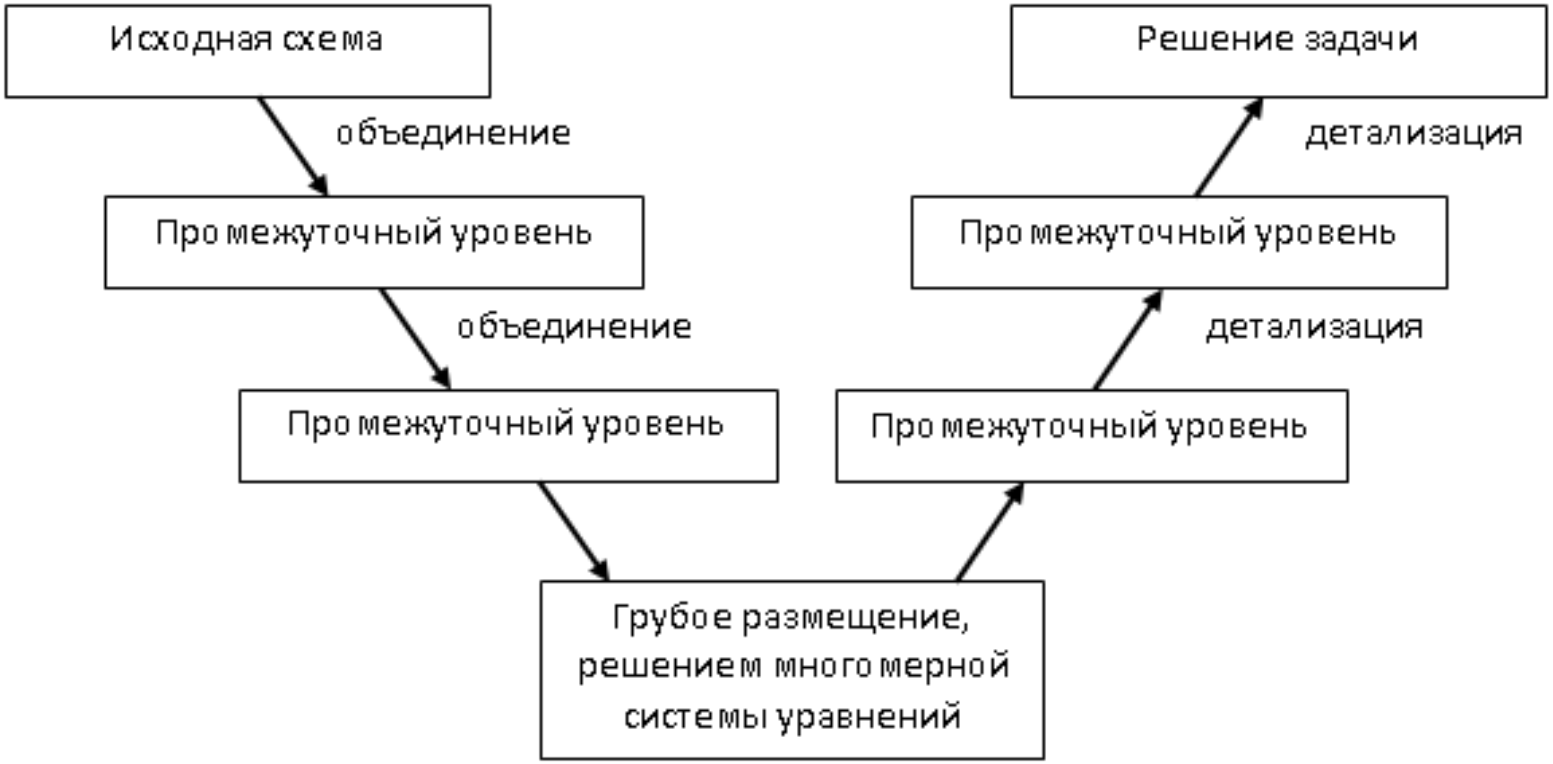
Силовые Kraftwerk



- mPL5
  - Tony Chan, Jason Cong, Kenton Sze “Multilevel generalized force-directed method for circuit placement”, ISPD’05, 2005, p.185-192



Программа размещения – mPL5 использует подход многоуровневой оптимизации



mPL5

- На первом шаге группы элементов связанных друг с другом, объединяются в один элемент, затем эта операция повторяется до тех пор, пока количество элементов не станет достаточно малым (меньше 500).
- На втором шаге производится размещение получившихся групп элементов так называемым методом квадратичного размещения (quadratic placement). Для этого производится поиск минимума функции , $\Phi(\bar{x},\bar{y}) = \sum_{i}\sum_{j}\left(a_{ij}(x_i - x_j)^2 + a_{ij}(y_i - y_j)^2\right)$
- где a_{ij} – степень связности между элементами i и j, (x_i, y_i) – положение элемента i.
- На третьем, заключительном шаге производится последовательная детализация размещения, т.е. вычисляется положение для каждой из подгрупп элементов, а затем и для каждого отдельного элемента

Результаты, полученные на тестах от 210K до 2.1M gates

Circuit	mPL5 Wirelength(WL)	mPL5 runtime(s)	Capo9.0 WL,runtime	Dragon3.01 WL,runtime	FastPlace1.0 WL,runtime	FengShui5.0 WL,runtime	mPL5-fast WL,runtime
ibm01	1.67E+06	64	1.08 , 1.90	1.02 , 16.81	1.09 , 0.10	1.08 , 2.06	1.09 , 0.23
ibm02	3.62E+06	126	1.09 , 1.82	1.02 , 7.34	1.06 , 0.13	1.02 , 1.93	1.02 , 0.26
ibm03	4.57E+06	113	1.10 , 2.56	1.05 , 8.04	1.12 , 0.13	1.03 , 2.39	1.05 , 0.28
ibm04	5.75E+06	151	1.06 , 2.47	1.00 , 10.95	1.04 , 0.13	1.05 2.16	1.04 , 0.35
ibm05	9.92E+06	158	1.02 , 2.52	0.98 , 17.36	1.05 , 0.14	1.00 , 2.29	1.05 , 0.30
ibm06	5.10E+06	200	1.11 , 2.01	0.98 , 10.44	1.04 , 0.12	1.02 , 2.12	1.03 , 0.28
ibm07	8.23E+06	259	1.11 , 2.45	1.04 , 9.61	1.08 , 0.19	1.09 , 2.32	1.09 , 0.30
ibm08	9.38E+06	389	1.05 , 1.73	0.96 , 15.47	1.02 , 0.14	n/a	1.05 , 0.38
ibm09	9.33E+06	342	1.08 , 2.30	1.07 , 16.26	1.12 , 0.17	1.06 , 1.72	1.07 , 0.28
ibm10	1.73E+07	450	1.10 , 2.42	1.04 , 10.96	1.07 , 0.20	1.07 , 1.56	1.11 , 0.30
ibm11	1.40E+07	437	1.09 , 2.70	1.03 , 7.81	1.09 , 0.19	1.04 , 2.31	1.06 , 0.30
ibm12	2.23E+07	482	1.11 , 2.48	1.03 , 11.15	1.08 , 0.20	1.07 2.03	1.06 , 0.36
ibm13	1.66E+07	596	1.10 , 2.32	1.05 , 7.73	1.11 , 0.20	1.09 , 1.80	1.07 , 0.28
ibm14	3.16E+07	1064	1.10 , 2.49	1.05 , 10.65	1.11 , 0.21	1.04 , 1.20	1.08 , 0.33
ibm15	3.85E+07	1379	1.09 , 2.41	1.04 , 11.14	1.13 , 0.23	1.07 , 2.15	1.07 , 0.30
ibm16	4.30E+07	1577	1.10 , 2.29	1.05 , 11.09	1.07 , 0.23	1.09 , 2.18	1.08 , 0.30
ibm17	6.13E+07	1705	1.09 , 2.32	1.08 , 22.22	1.08 , 0.23	1.08 , 2.17	1.09 , 0.30
ibm18	4.10E+07	1904	1.09 , 2.00	1.02 , 17.84	1.10 , 0.25	1.04 , 2.07	1.07 , 0.29
average	1.00	1.00	1.09 , 2.29	1.03 , 12.38	1.08 , 0.18	1.06 , 2.03	1.07 , 0.30

APlace: гладкая аппроксимация целевой функции

1. Квадратичная аппроксимация

$$\sum_{e \in E} \left(\sum_{v_i, v_j \in e, i<j} w_{ij} (x_i - x_j)^2 + \sum_{v_i, v_j \in e, i<j} w_{ij} (y_i - y_j)^2 \right)$$

2. log-sum-exp аппроксимация (Kahng A. “Implementation and Extensibility of an Analytical Placement”)

$$\eta \sum_{e \in E} \left(\log \sum_{v_k \in e} \exp(x_k/\eta) + \log \sum_{v_k \in e} \exp(-x_k/\eta) + \log \sum_{v_k \in e} \exp(x_k/\eta) + \log \sum_{v_k \in e} \exp(-x_k/\eta) \right)$$

3. L_p – аппроксимация (Cong J. “Multilevel Generalized Force-directed Method for Circuit Placement”)

$$\sum_{e \in E} \left(\left(\sum_{v_k \in e} x_k^p \right)^{\frac{1}{p}} - \left(\sum_{v_k \in e} x_k^{-p} \right)^{-\frac{1}{p}} + \left(\sum_{v_k \in e} y_k^p \right)^{\frac{1}{p}} - \left(\sum_{v_k \in e} y_k^{-p} \right)^{-\frac{1}{p}} \right)$$

Сглаживание стандартных функций: β - сглаживание для модуля

Ross Baldick, Andrew B. Kahng, Andrew kennings, Igor L. Markov “Function Smoothing with Applications to VLSI Layout”

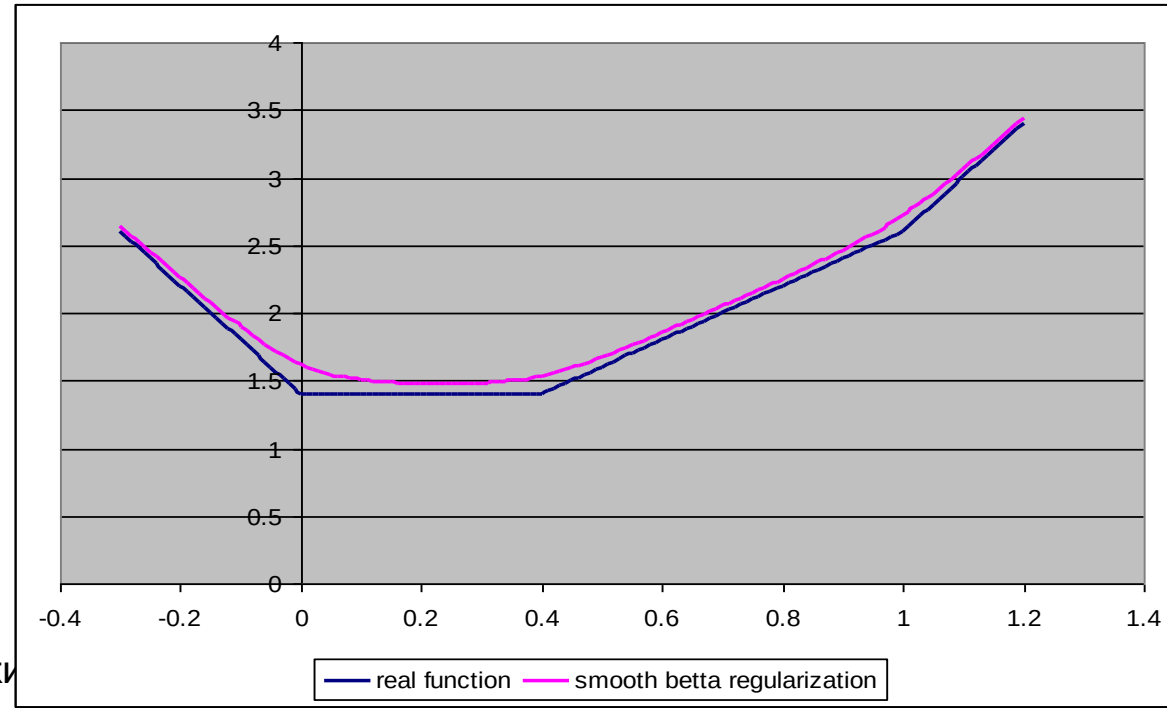
$$|f(x)| = ((f(x))^{\alpha} + \beta)^{1/\alpha}$$

$$\min(a,b) = \frac{a+b-|a-b|}{2}$$

$$\max(a,b) = \frac{a+b+|a-b|}{2}$$

Пример β-сглаживания для функции с β=0.01 и α=2

$$f(x) = |1-x| + 2|x| + |0.4-x|$$



Математически

Сравнение аппроксимаций WL

Схема	LogSumExp WL, runtime	Lp-norm WL, runtime	Quadratic WL, runtime
ibm1	1, 1	1.05, 1.71	1.73, 0.81
Ibm2	1, 1	1.02, 1.80	1.84, 1.65
Ibm3	1, 1	0.99, 1.82	1.63, 0.64
Ibm4	1, 1	0.98, 1.47	1.48, 0,47
Ibm5	1, 1	1.06, 1.74	1.49, 1.17
Ibm6	1, 1	1.03, 1.36	1.82, 0.50
Ibm7	1, 1	1.04, 1.62	1.50, 0.67
Ibm8	1, 1	1.01, 1.45	1.79, 0.72
Ibm9	1, 1	1.05, 1.52	1.65, 0.53
Ibm10	1, 1	1.03, 2.07	1.47, 0.72
Ibm11	1, 1	1.04, 1.62	1.54, 0.52
Ibm12	1, 1	1.02, 1.68	1.34, 0.67
Ibm13	1, 1	1.03, 1.72	1.69, 0.63
Ibm14	1, 1	1.03, 1.46	1.60, 0.71
Ibm15	1, 1	1.04, 1.84	1.61, 0.80
Ibm16	1, 1	1.04, 1.63	1.66, 0.79
Ibm17	1, 1	1.03, 1.73	1.42, 0.85
Ibm18	1, 1	1.07, 1.78	1.77, 0.97
среднее	1, 1	1.03, 1.67	1.61, 0.77

Cong J. “Multilevel Generalized Force-directed Method for Circuit Placement”

APlace: формулировка задачи нелинейного программирования

$$H = (V, E),$$
$$V = \{v_1, \ldots, v_N, v_{N+1}, \ldots, v_{N+P}\},$$
$$E = \{e_1, \ldots, e_m\}$$

$$l(e) = \max_{v_i, v_j \in e, i<j} | x_i - x_j | + \max_{v_i, v_j \in e, i<j} | y_i - y_j |$$

$$\sum_{\forall e} l(e) \to \min$$