Криптоанализ

Павленко Артём // Университет ИТМО

Сириус
Образовательный центр

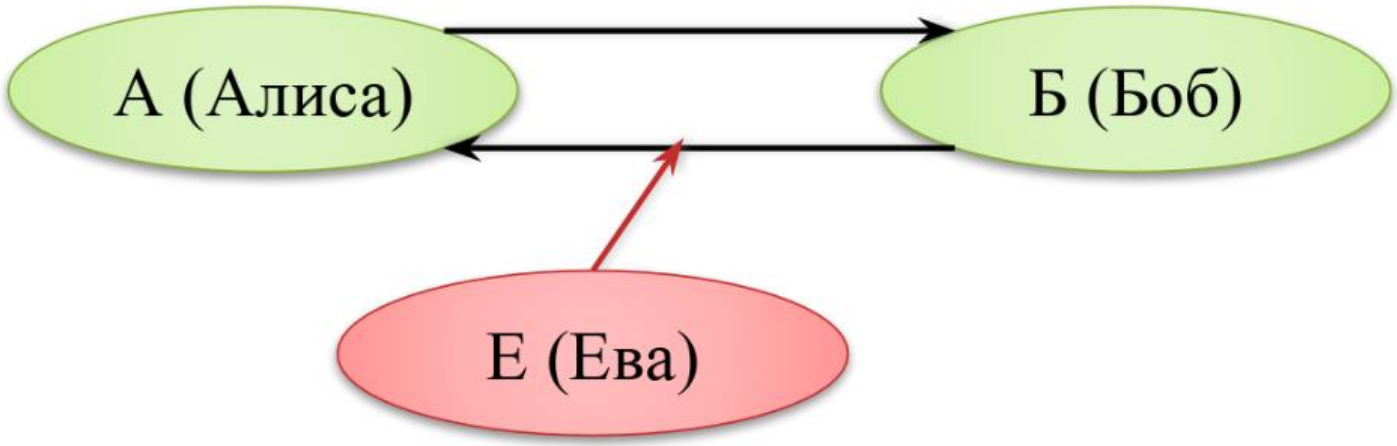
ITMOre than a UNIVERSITY

Вступление

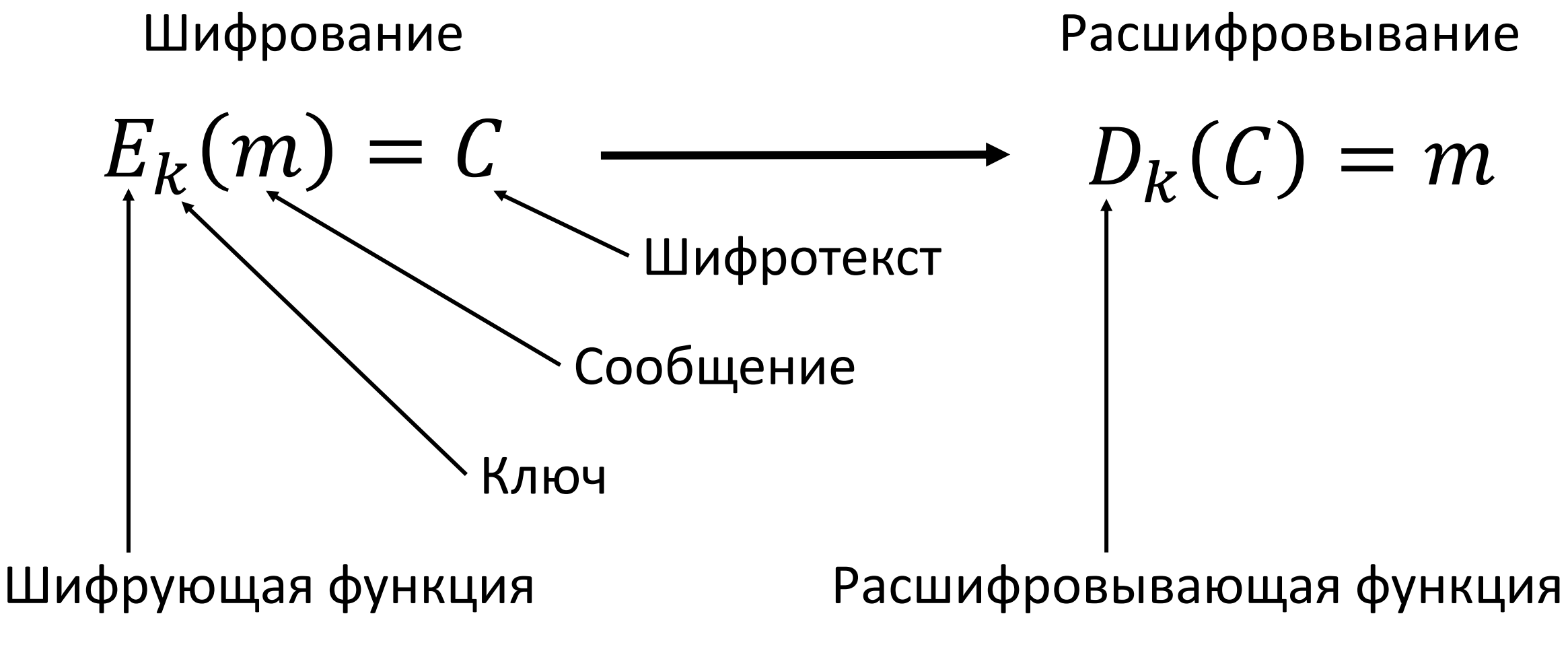
- Современный мир и информационная безопасность
  - Банковские системы
  - Ресурсы интернета (https, TLS, …)
  - Беспроводные технологии (Bluetooth, Wi-Fi, NFC, …)
  - Сотовая связь (2G, …, 5G)

Передача сообщения

- Алиса хочет отправить сообщение Бобу
- Однако Ева может их подслушивать



Процесс шифрования



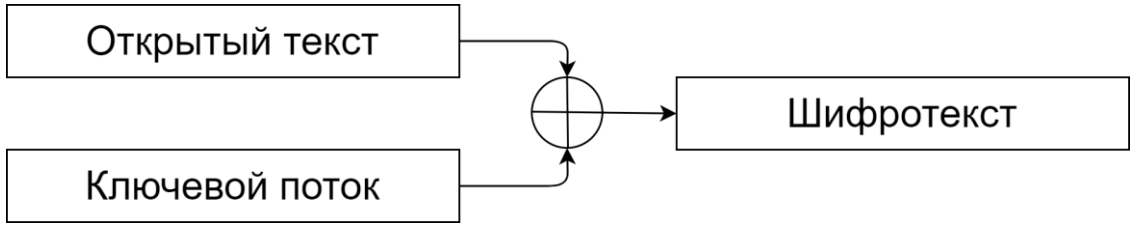
Способы шифрования

- Симметричное шифрование
  - **один** ключ для шифрования и расшифровки
  - **высокая** скорость шифрования и расшифровывания

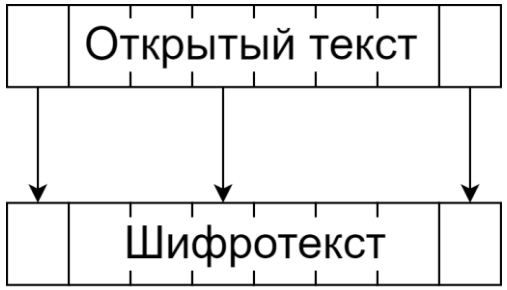
- Ассиметричное шифрование
  - **открытый и закрытый** ключи
  - **ресурсоемкий** процесс шифрования и расшифровывания

Симметричное шифрование

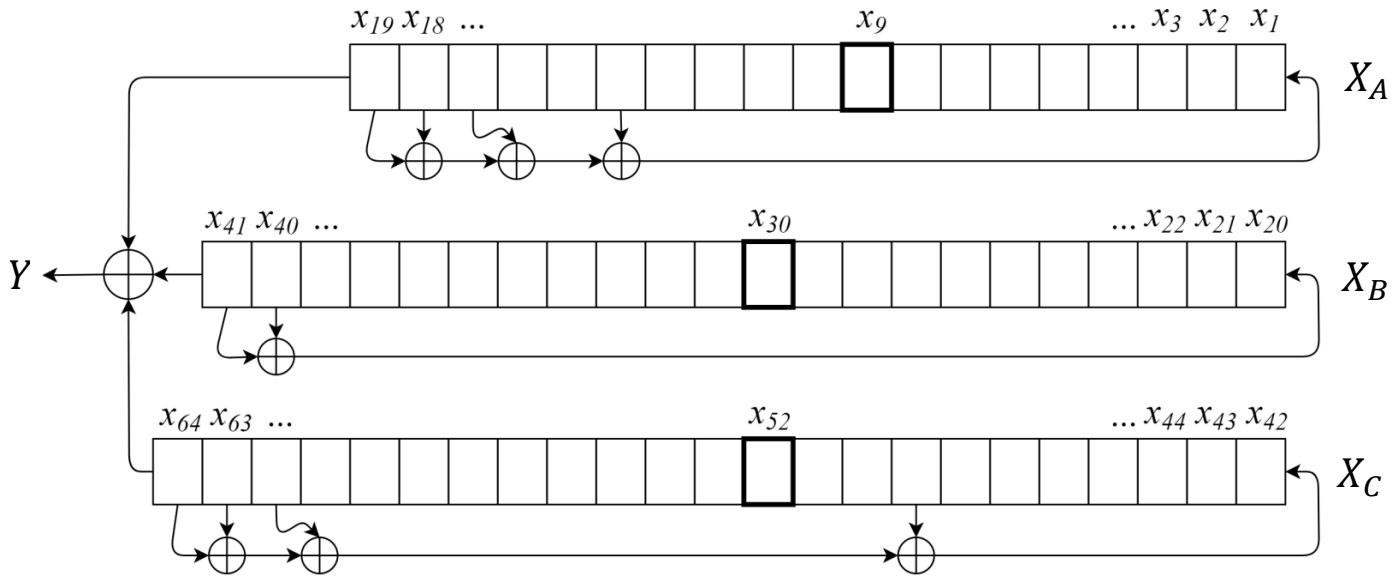
- Поточные алгоритмы
  - Легче аппаратно реализовать
  - Быстрее работают



- Блочные алгоритмы
  - Более гибкие, могут использоваться как поточные



Поточный алгоритм A5/1*



Секретный ключ X
Ключевой поток Y

$X = X_A \cup X_B \cup X_C$
$X_A = \{x_1, x_2, \ldots x_{19}\}$
$X_B = \{x_{20}, x_{21}, \ldots x_{41}\}$
$X_C = \{x_{42}, x_{43}, \ldots x_{64}\}$

*Использовался в протоколе сотовой связи 2G

Криптоанализ

- наука о методах дешифровки зашифрованной информации без предназначенного для этого ключа

Применение:

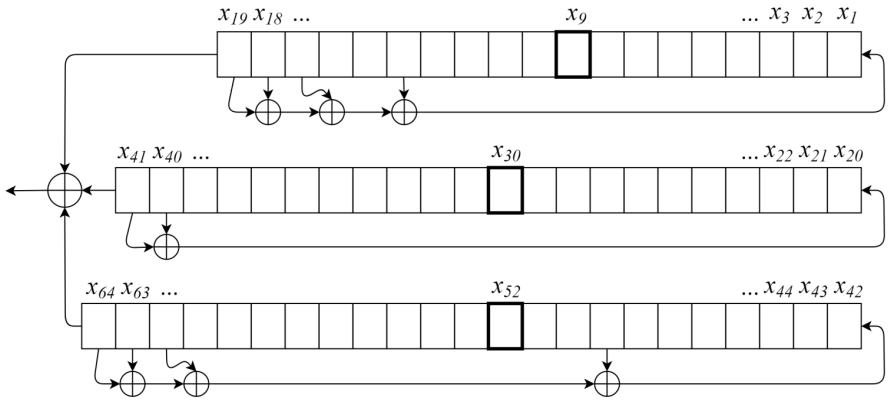
- Поиск уязвимостей в алгоритмах шифрования
- Обоснование их криптографической стойкости

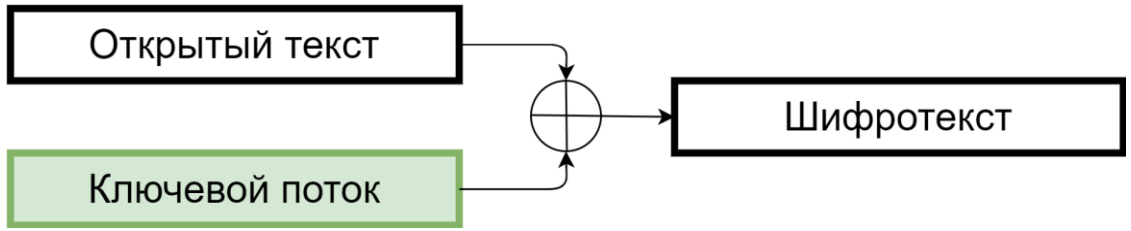
Методы криптоанализа

- На основе шифротекста
- На основе открытых текстов и соответствующих шифротекстов
- На основе подобранного открытого текста
- На основе адаптивно подобранного открытого текста

Методы криптоанализа

- На основе шифротекста
- **На основе открытых текстов и соответствующих шифротекстов**



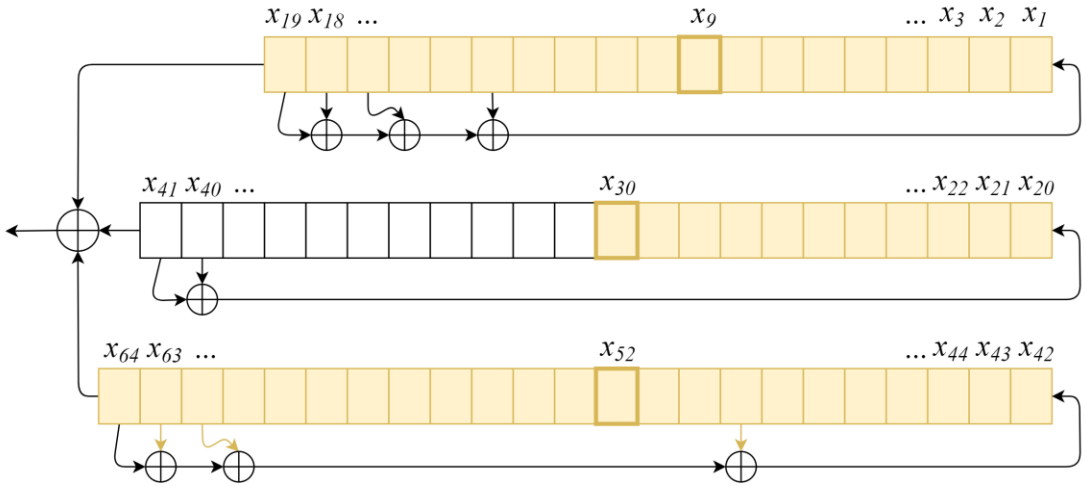


- На основе подобранного открытого текста
- На основе адаптивно подобранного открытого текста

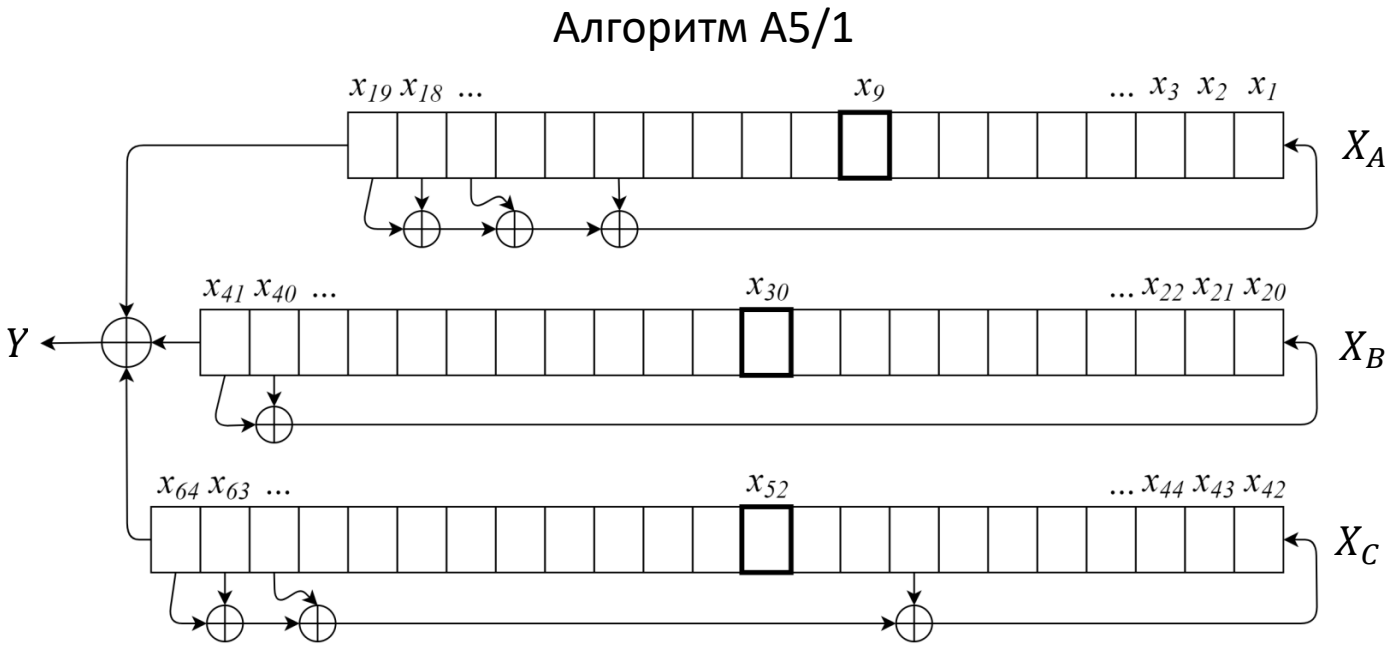
Криптографическая атака

- Результат применения методов криптоанализа
  - Асимптотическая сложность (элементарные операции)
  - Временная сложность (секунды)

Алгоритм A5/1	
Полный перебор	Атака Андерсона (1994 г.)
$O(2^{64})$	$O(2^{53})$



Задача криптоанализа (1)



Секретный ключ X

Ключевой поток Y

$X = X_A \cup X_B \cup X_C$

$X_A = \{x_1, x_2, \ldots x_{19}\}$

$X_B = \{x_{20}, x_{21}, \ldots x_{41}\}$

$X_C = \{x_{42}, x_{43}, \ldots x_{64}\}$

$$f_{A5/1}\colon \{0,1\}^{64} \to \{0,1\}^{m}$$

Задача криптоанализа (2)

$$f_{A5/1}: \{0,1\}^{64} \to \{0,1\}^{m}$$

$$f_{A5/1}(x) = y$$

Задача обращения криптографической функции:
зная ключевой поток **y**, восстановить секретный ключ **x**

$$f_{A5/1}^{-1}(y) = x$$

Алгебраический криптоанализ

- Метод предполагает сведение задачи к полиномиальной системе уравнений

SAT-based криптоанализ

- Boolean SATisfiability первая известная NP-полная задача
- Большое число применимых SAT-решателей
- Ежегодные SAT Competitions

Транслятор

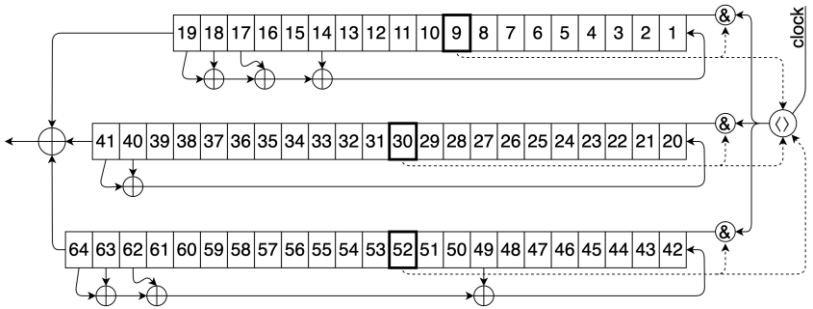
- Позволяет преобразовать исходную программу, представленную на одном языке, в программу на другом языке

Нас интересует трансляция: $* \rightarrow SAT$

- Transalg
- Cryptol
- URSA

Сведение к SAT

Алгоритм A5/1



вручную ⇒

Программа для Transalg

```
__in bit XA[19];
__in bit XB[22];
__in bit XC[23];
__out bit Y[128];

bit shift_rslosA() {
  bit x0 = XA[18]^XA[17]^XA[16]^XA[13];
  for(int j = 18; j > 0; j=j-1) {
    XA[j] = XA[j-1];
  }
  XA[0] = x0;
}
...

bit majority(bit A, bit B, bit C) {
    return A&B|A&C|B&C;
}

void main() {
  int b1 = 8;
  int b2 = 10;
  int b3 = 10;
  bit maj;
  for(int i = 0; i < 128; i=i+1) {
    maj = majority(XA[b1], XB[b2], XC[b3]);
    if(!(maj^XA[b1])) shift_rslosA();
    if(!(maj^XB[b2])) shift_rslosB();
    if(!(maj^XC[b3])) shift_rslosC();
    Y[i] = XA[18]^XB[21]^XC[22];
  }
}
```

автоматически ⇒

SAT-формула

```
p cnf 8425 38262
c input variables 64
c literals count 128374
65 9 30 0
65 9 52 0
-65 9 -30 -52 0
65 -9 -52 0
-65 -9 30 52 0
65 -9 -30 0
66 -19 65 0
66 -18 -65 0
-66 19 65 0
-66 18 -65 0
67 -18 65 0
67 -17 -65 0
-67 18 65 0
-67 17 -65 0
68 -17 65 0
68 -16 -65 0
-68 17 65 0
-68 16 -65 0
69 -16 65 0
69 -15 -65 0
-69 16 65 0
... ...
-8425 -8293 8295 -8297 0
-8425 -8293 -8295 8297 0
```

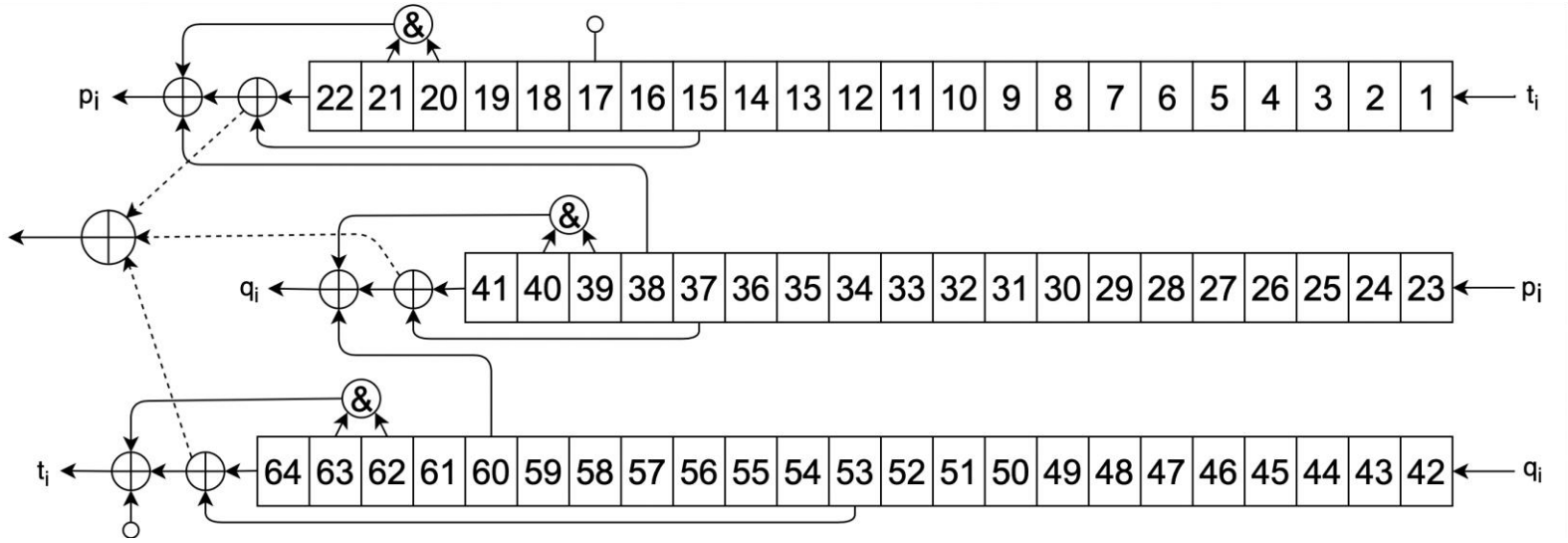
SAT-решатели

TL: ограничение времени работы SAT-решателя

Пример взлома алгоритма Trivium-Toy 64

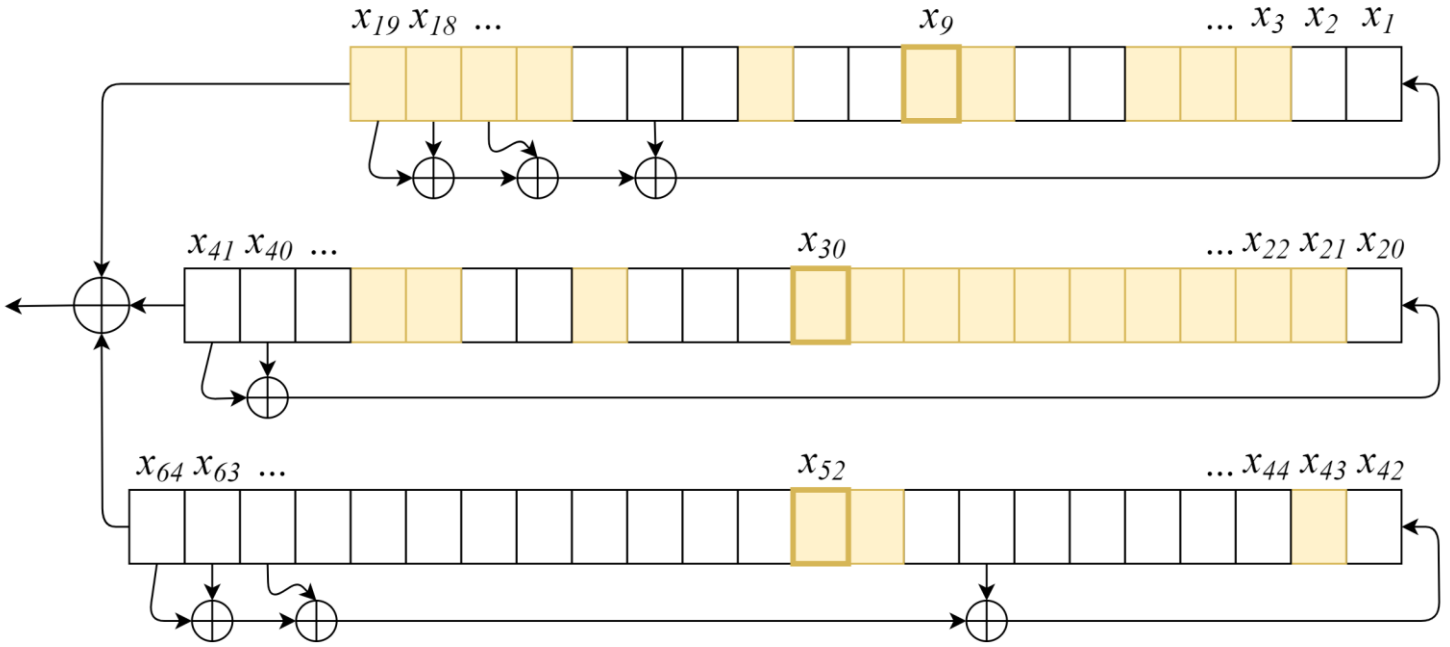
CPU:
AMD Opteron 6276
@ 2.3 GHz x32

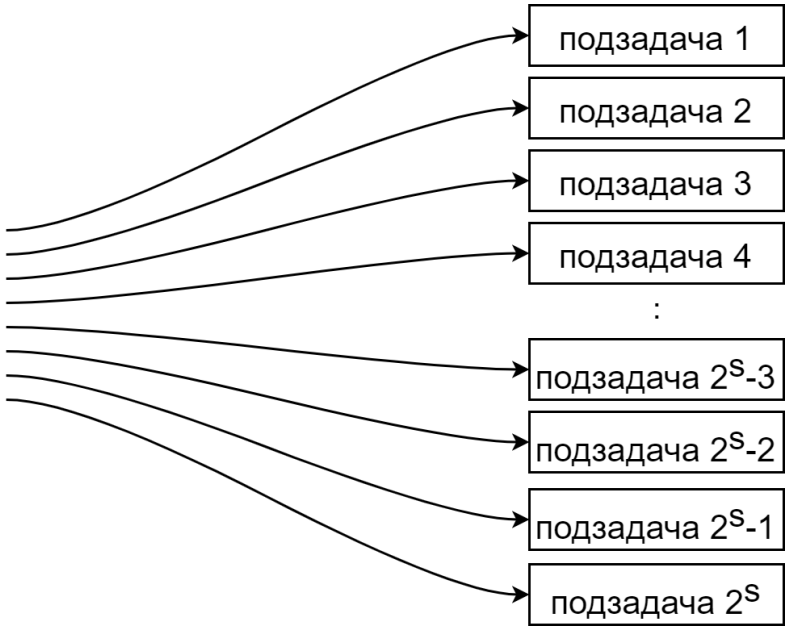
Ограничение времени:
7 дней



	PLingeling	Treengeling	Guess-and-determine атака
Задача 1	прервана	прервана	2д 6ч
Задача 2	прервана	3д 2ч	3д 19ч
Задача 3	прервана	4д 10ч	15ч
Задача 4	прервана	прервана	1д 21ч
Задача 5	прервана	прервана	4д 3ч

Guess-and-Determine атака



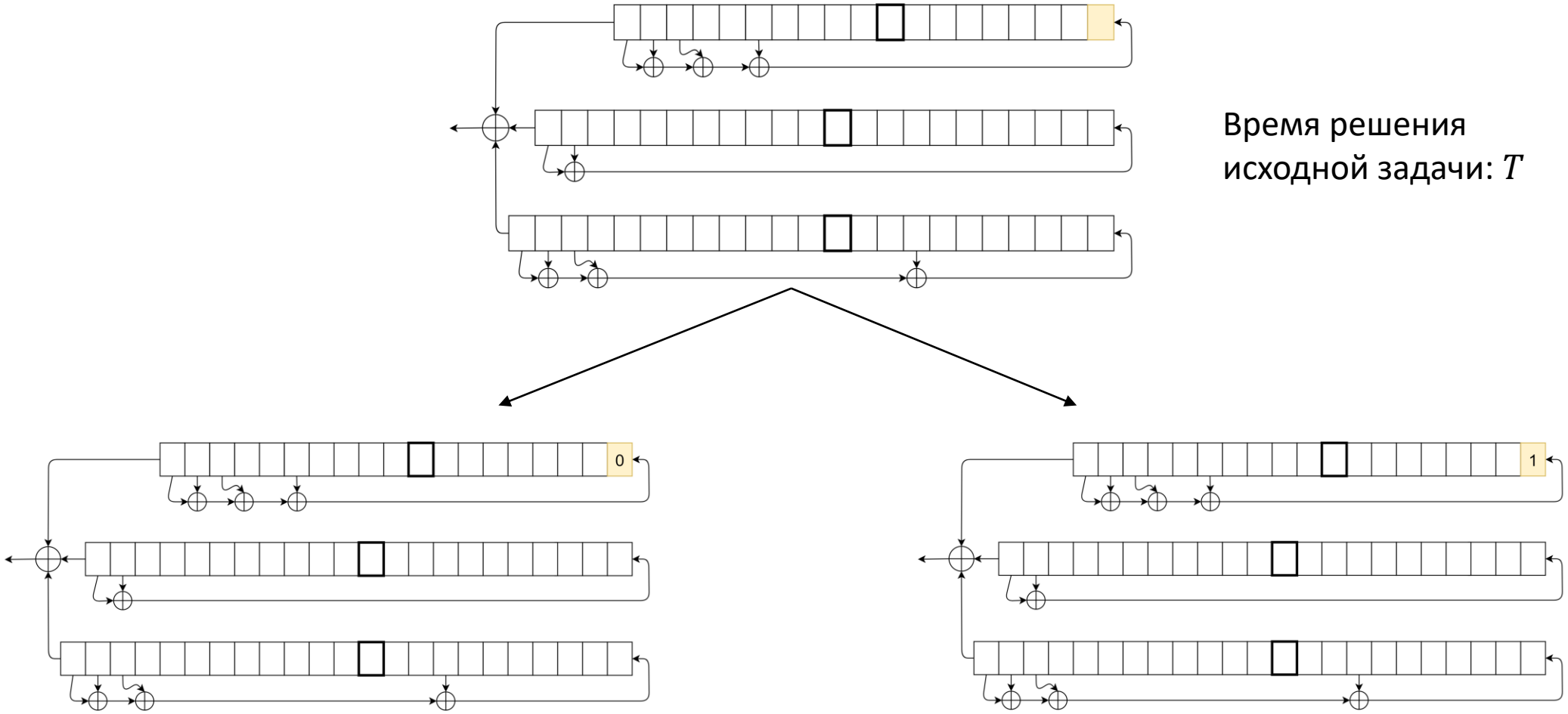


$s = |B|$

B = { x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, …, x_{19}, x_{21}, …, x_{30}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} }

$$\sum_{i=1}^{2^s} t_i \ll T_{BruteForce}$$

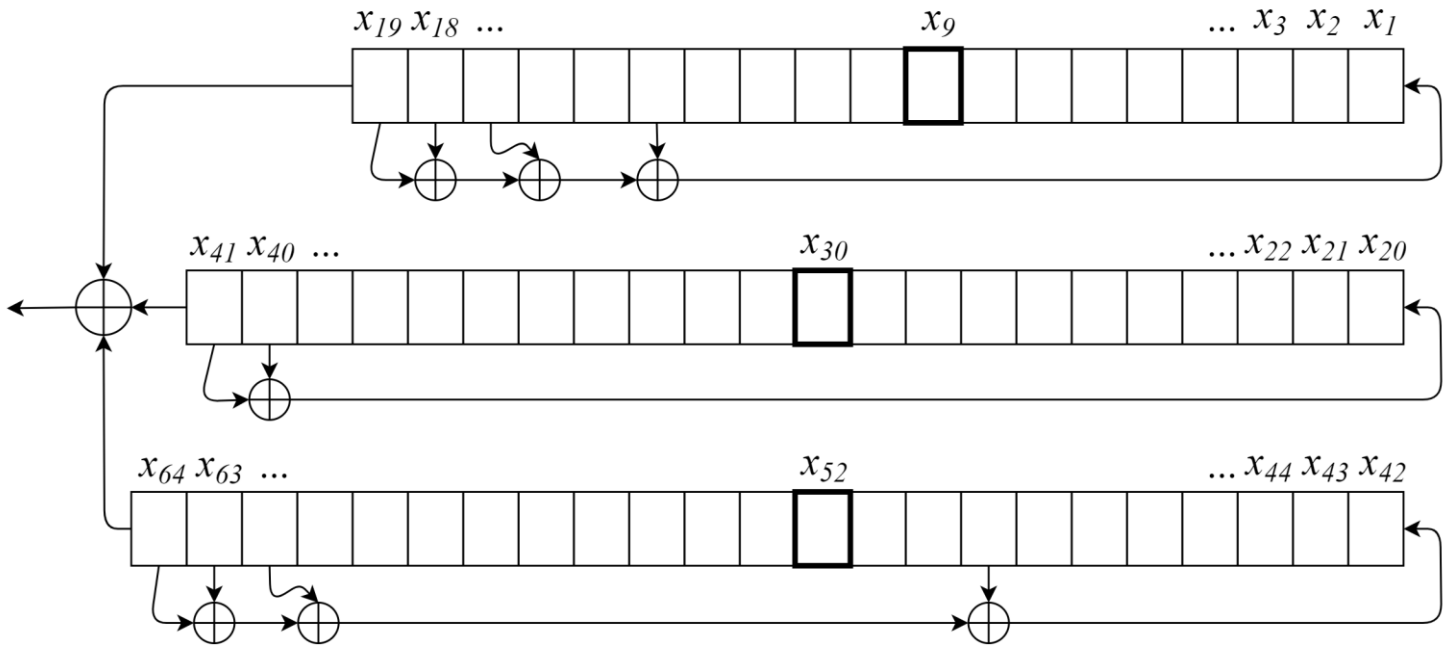
Guess-and-Determine. Декомпозиция



Время решения исходной задачи: T

Суммарное время решения: $T' = t_1 + t_2$

Как построить эффективную декомпозицию?



B = ?

$$\Phi(B) = \sum_{i=1}^{2^s} t_i$$

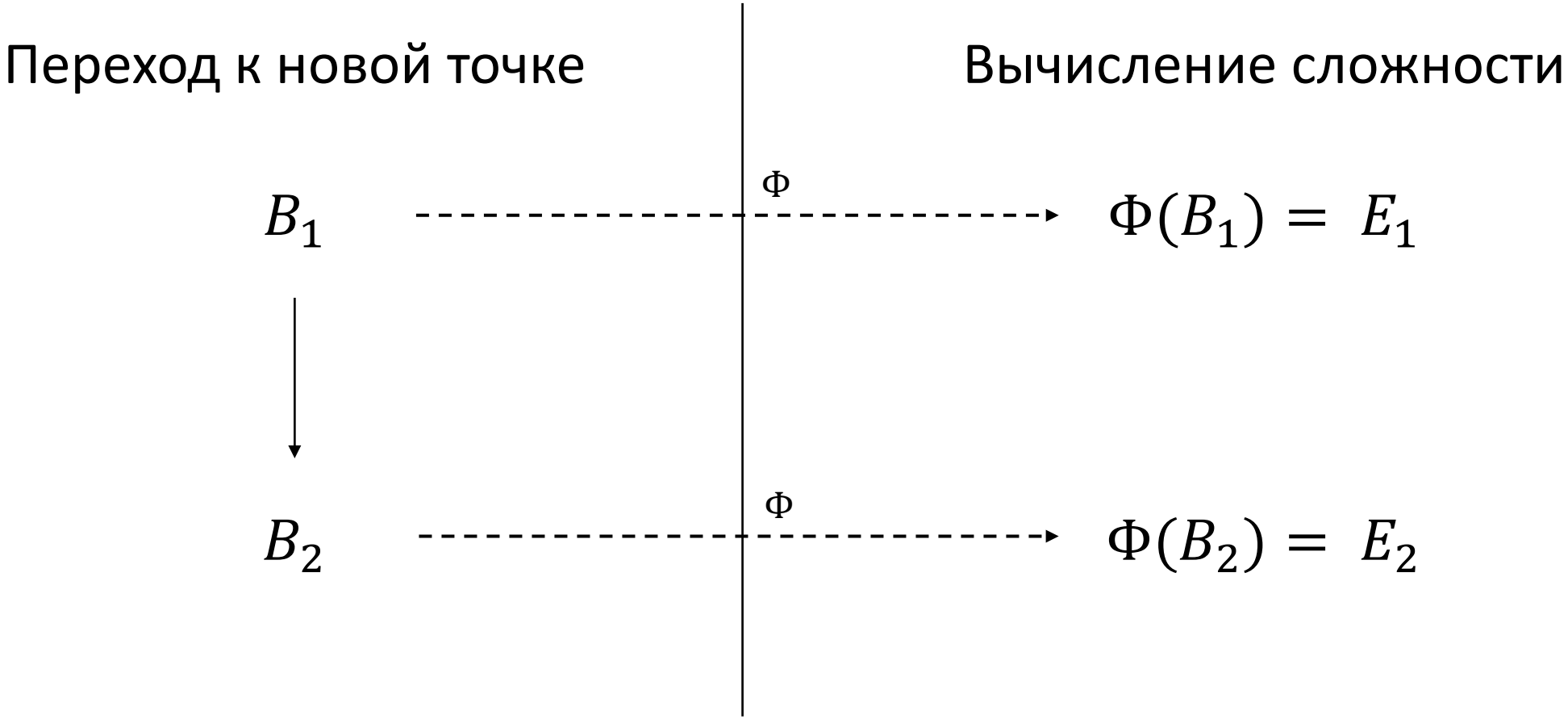
Метаэвристические алгоритмы

- Итеративные практические алгоритмы поиска решения задачи, о которой мало что известно

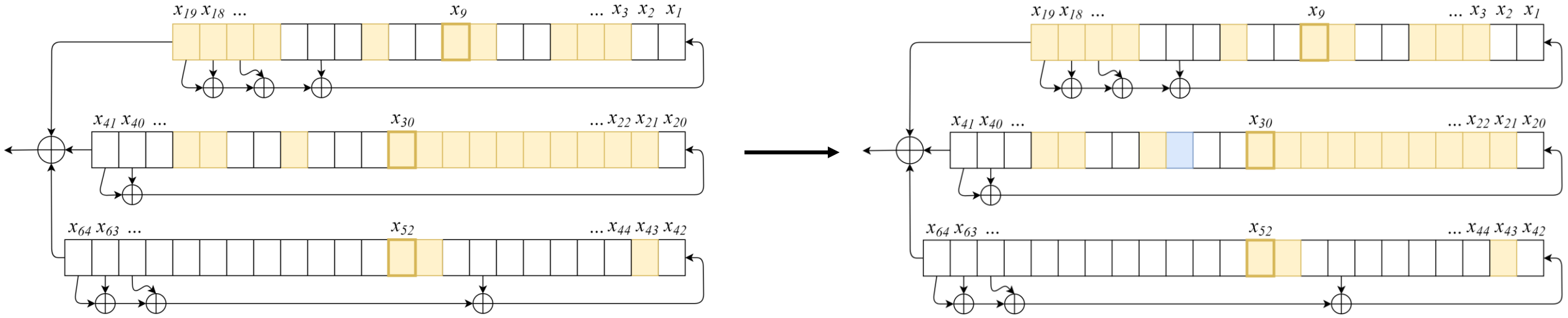
Алгоритмы:

- Локальный поиск
- Поиск с запретами
- Имитация отжига
- Эволюционные алгоритмы

Необходимые операции



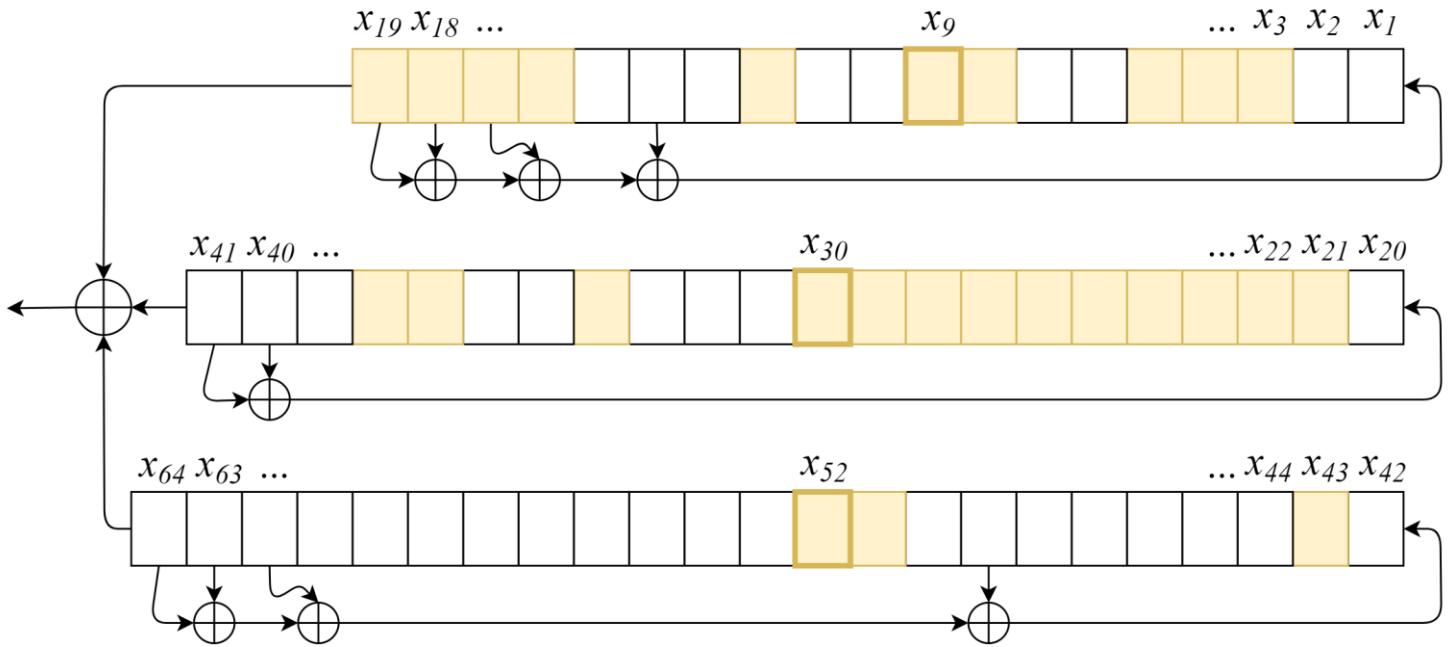
Изменение точки

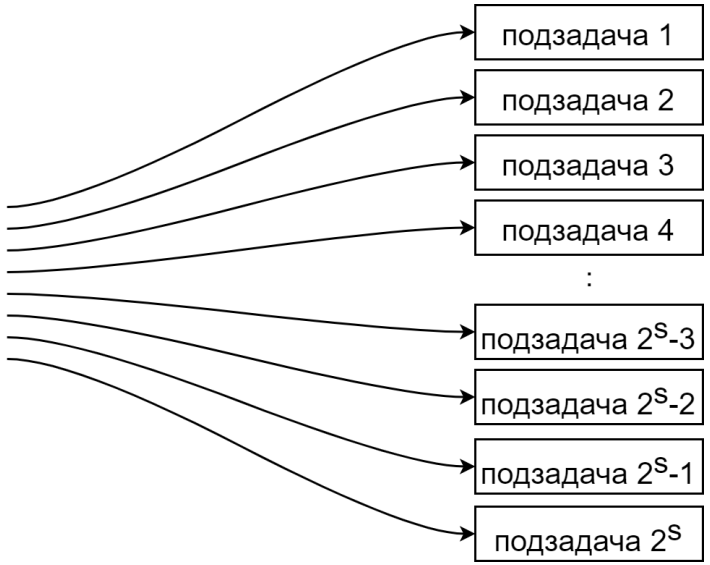


$B_1 = \{ x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, \ldots, x_{19}, x_{21}, \ldots, x_{30}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} \}$

$B_2 = \{ x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, \ldots, x_{19}, x_{21}, \ldots, x_{30}, \mathbf{x_{33}}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} \}$

Вычисление сложности

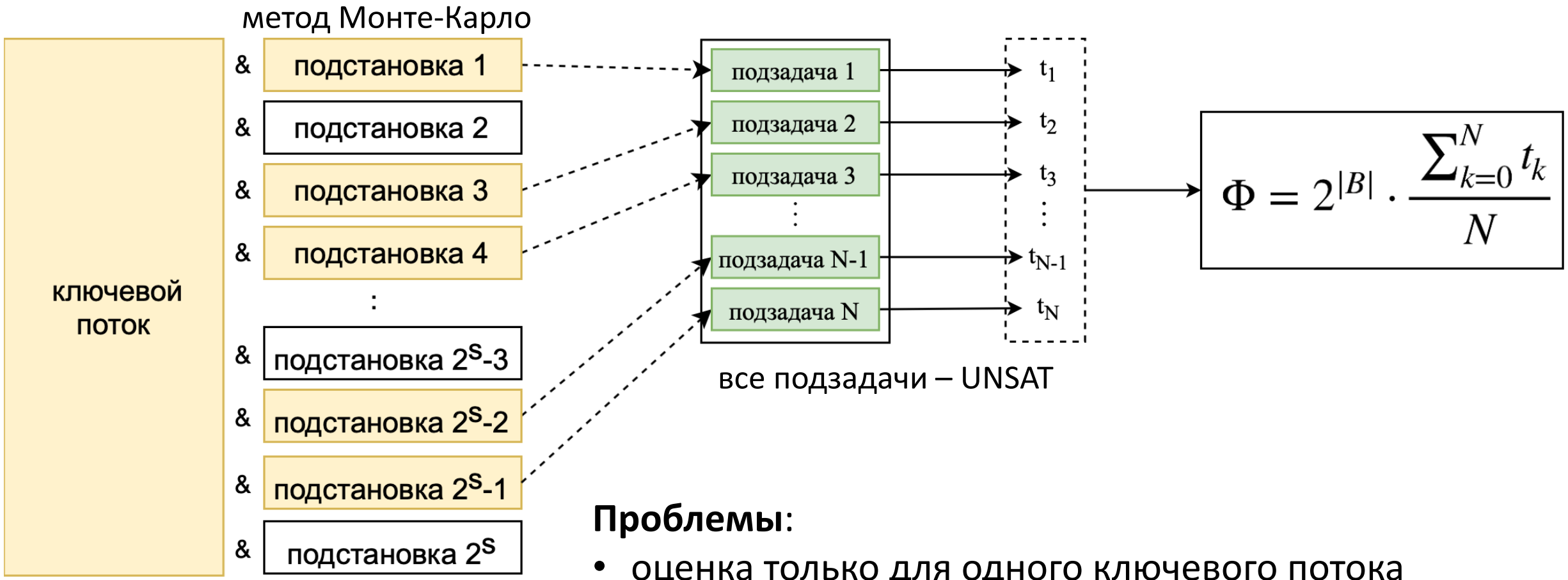




B = { x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, ..., x_{19}, x_{21}, ..., x_{30}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} }

$$T_B = \sum_{i=0}^{N} t_i \text{, где } N = 2^s \text{ и } s = |B|$$

Оценивание декомпозиции. UNSAT-иммунность

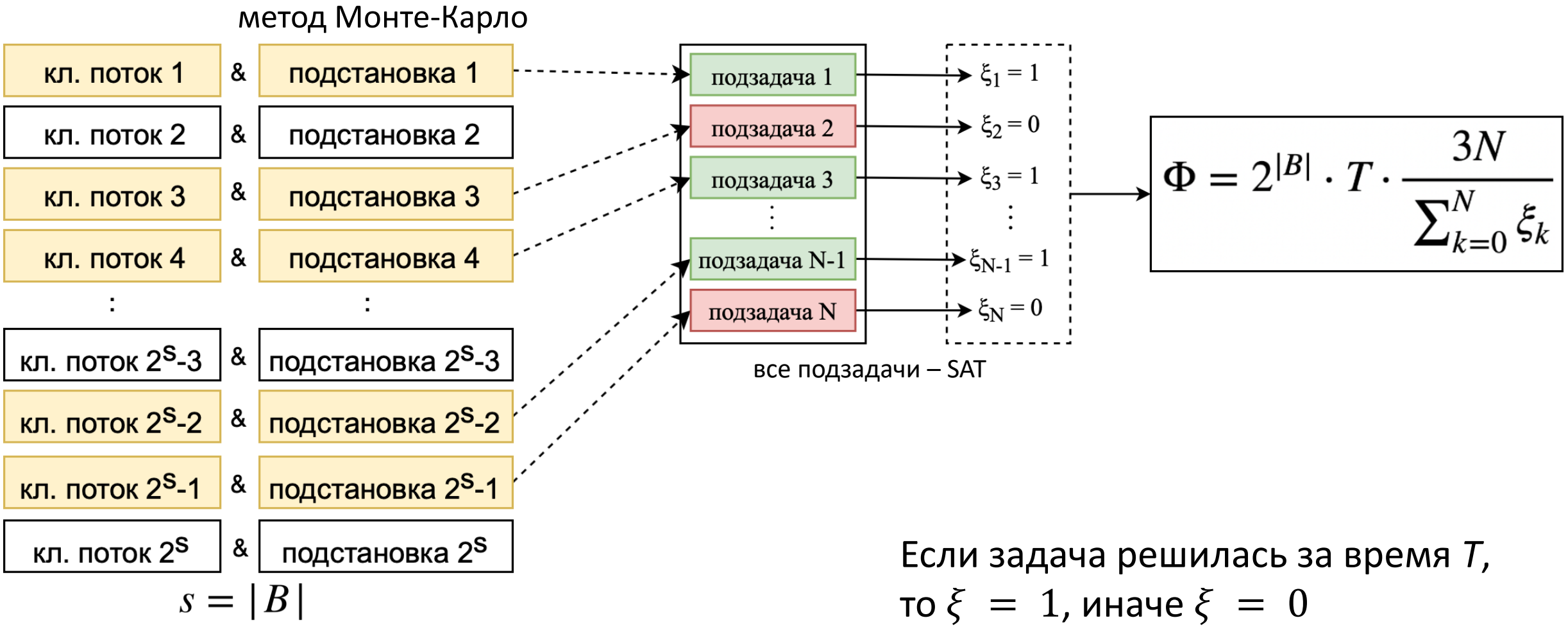


$s = |B|$

Проблемы:

- оценка только для одного ключевого потока
- время решения задач имеет плохое распределение

Оценивание декомпозиции. SAT-иммунность



Если задача решилась за время *T*,
то $\xi = 1$, иначе $\xi = 0$

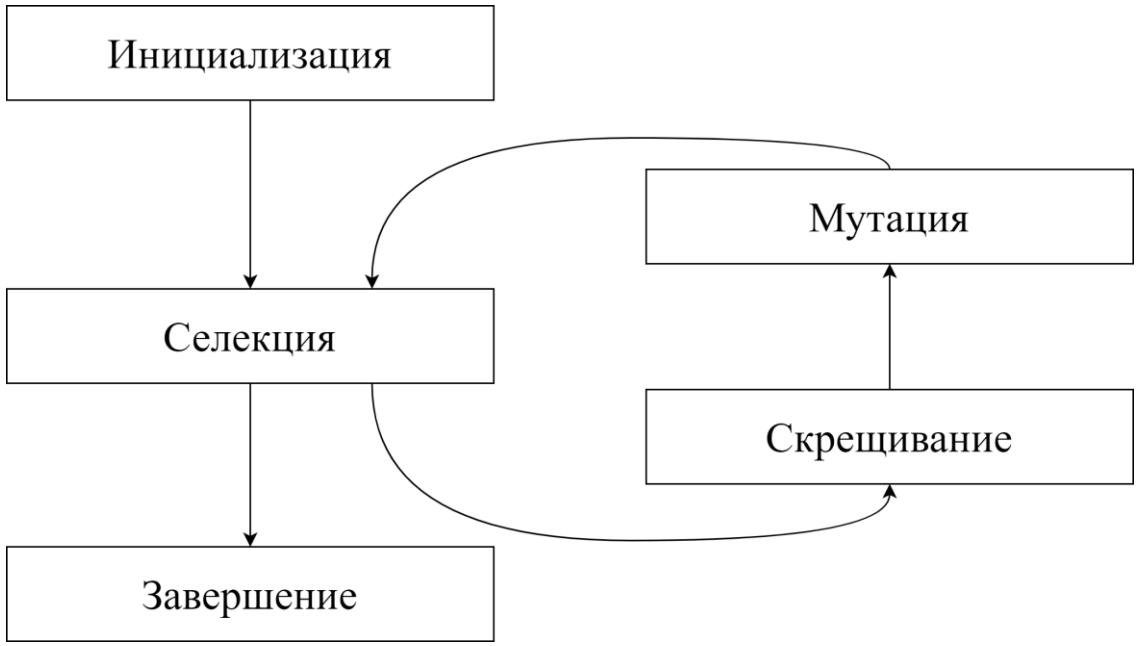
*IBS resistant function: [*Semenov A., Zaikin O., Otpuschennikov I., Kochemazov S., Ignatiev A.* On cryptographic attacks using backdoors for SAT // The Thirty-Second AAAI Conference on Artificial Intelligence. – IEEE, 2018. – P. 6641-6648]

Промежуточный итог

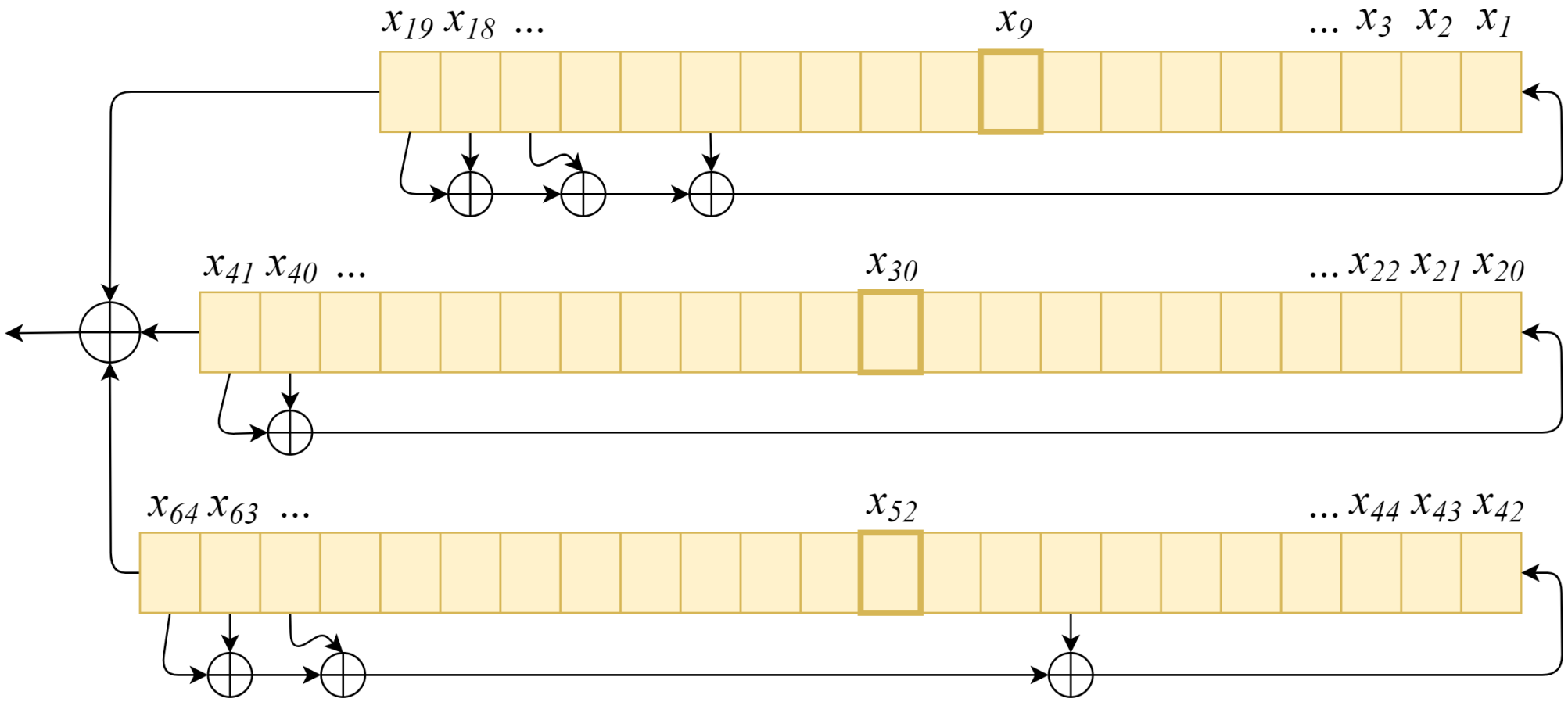
- Обеспечение информационной безопасности методами **криптографии**
- Криптоанализ, как метод проверки **криптостойкости** систем, посредством построения криптографических атак
- Сведение задачи криптоанализа к **SAT** для построения атак
- Для ускорения атак строятся **декомпозиционные множества**
- Выбор **эффективной** декомпозиции является трудной задачей
- Были рассмотрены **быстрые** методы **оценки** исследуемого декомпозиционного множества

Эволюционные алгоритмы

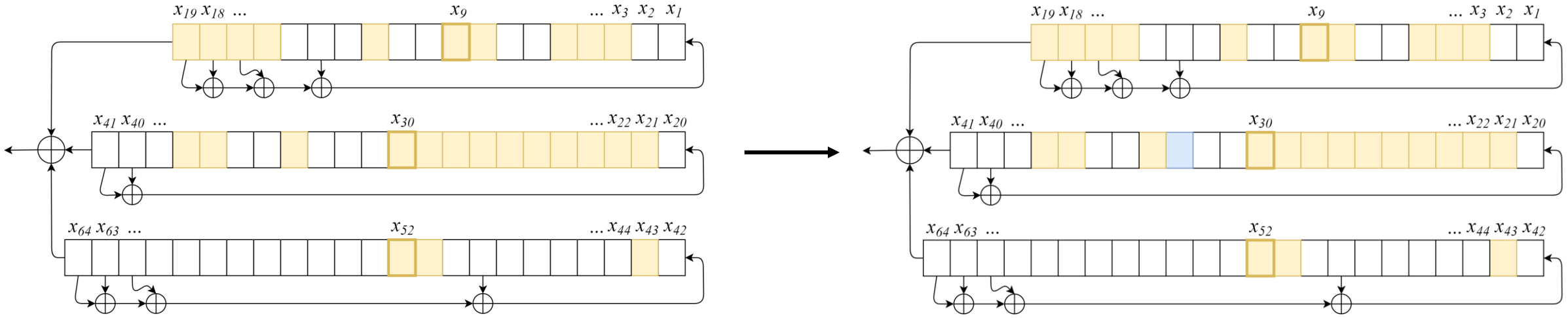
- Эволюционные стратегии
  - Мутация
  - Селекция
- Генетические алгоритмы
  - Мутация
  - Скрещивание
  - Селекция



Инициализация



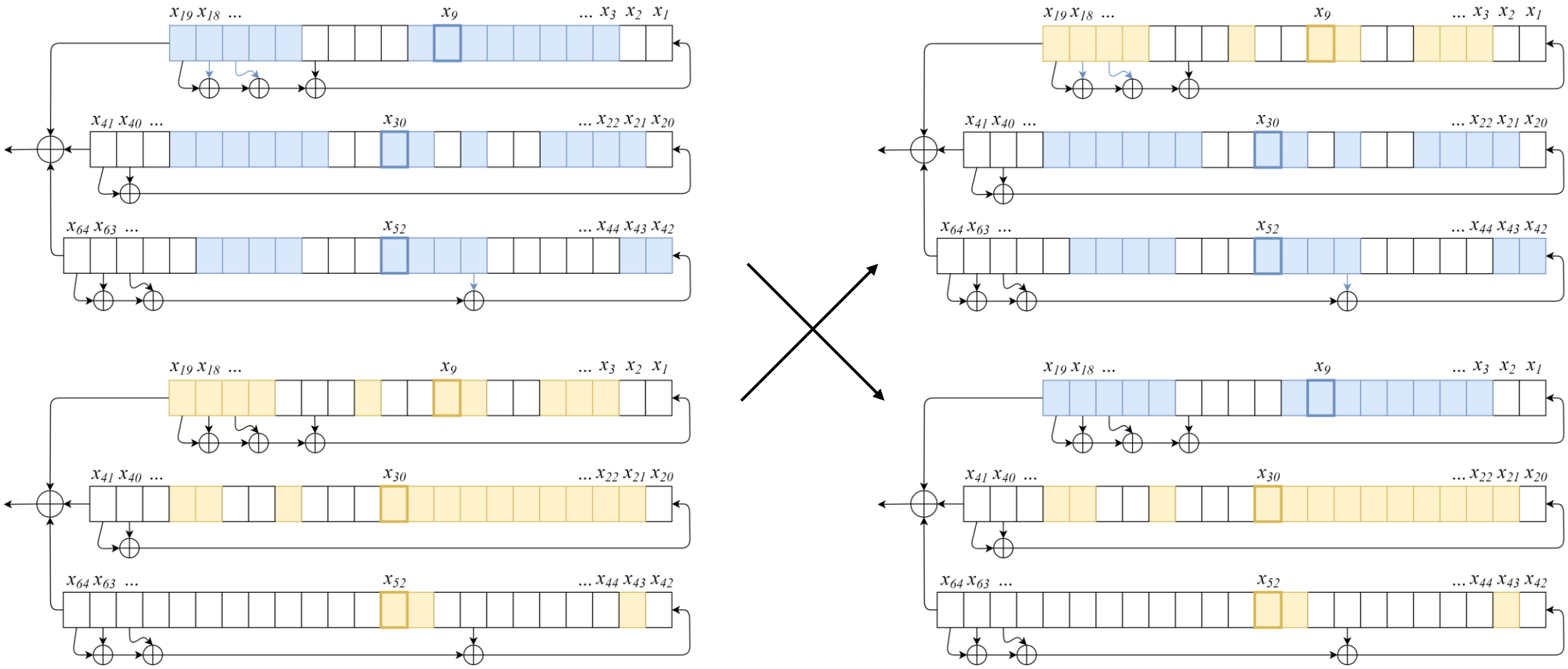
Мутация



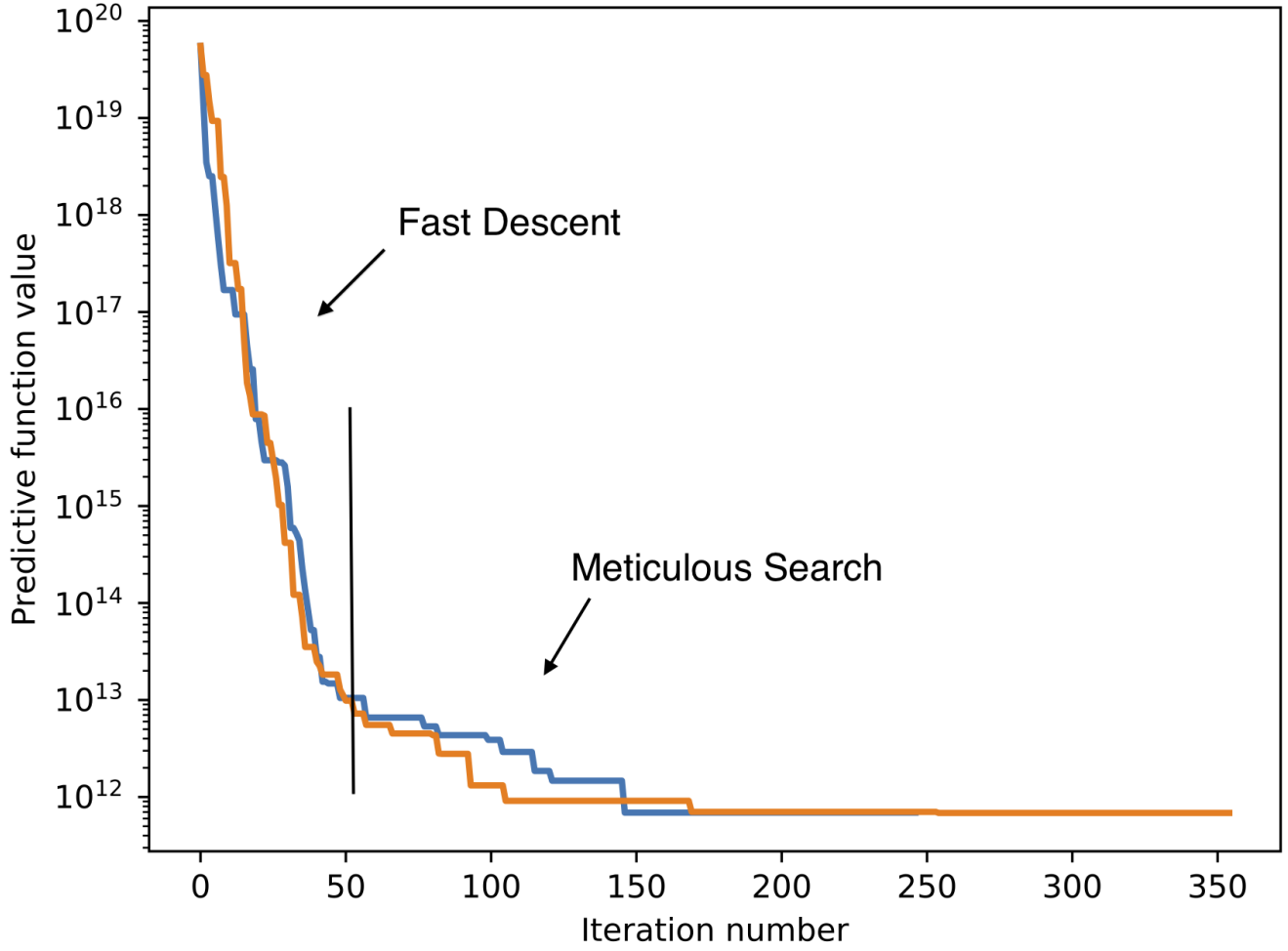
$B_1 = \{ x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, \ldots, x_{19}, x_{21}, \ldots, x_{30}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} \}$

$B_2 = \{ x_3, x_4, x_5, x_8, x_9, x_{12}, x_{16}, \ldots, x_{19}, x_{21}, \ldots, x_{30}, \mathbf{x_{33}}, x_{34}, x_{37}, x_{38}, x_{43}, x_{51}, x_{52} \}$

Скрещивание



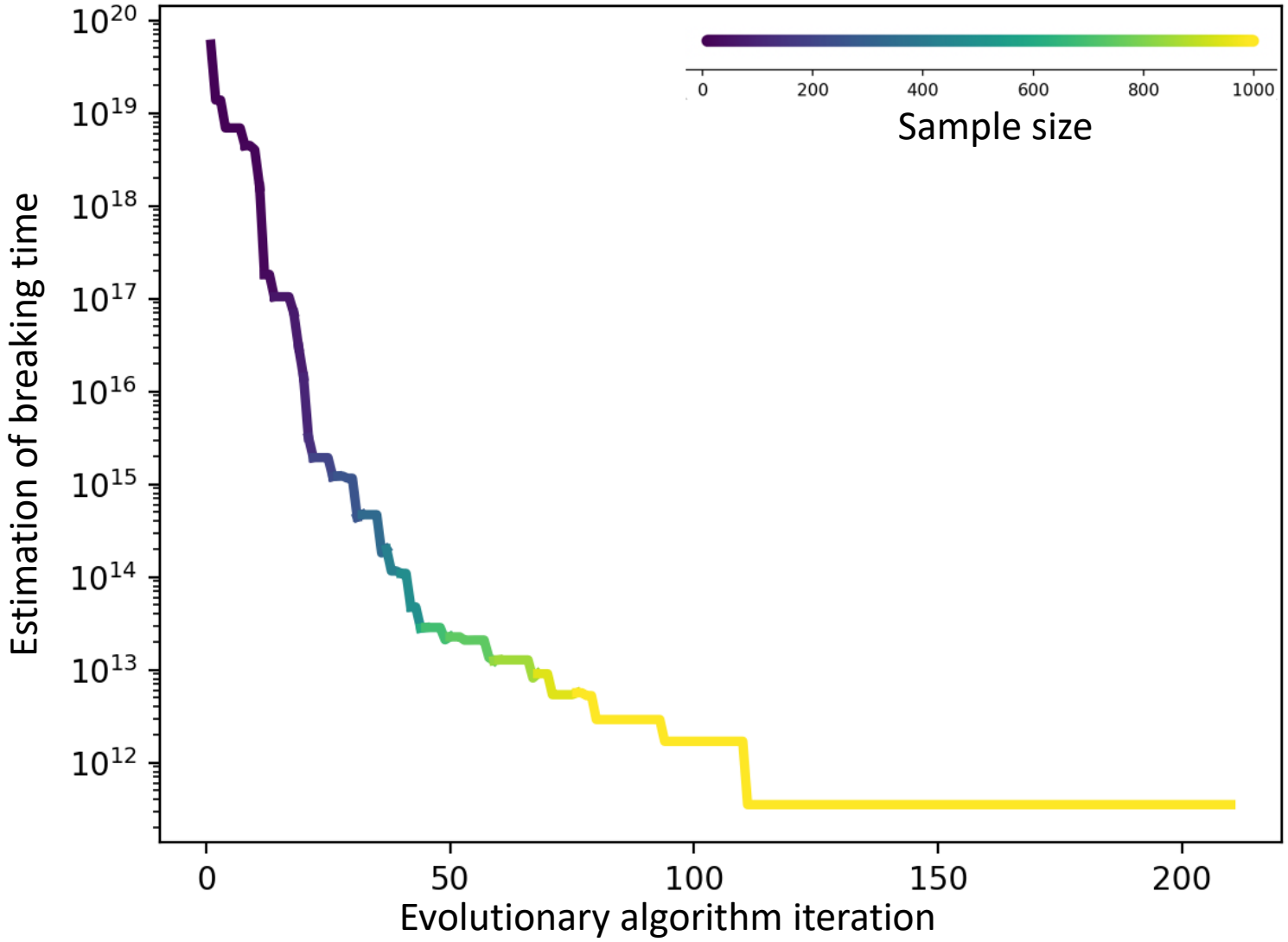
Пример процесса оптимизации



Алгоритм: A5/1

Стратегия: (1+1)

Адаптивная стратегия



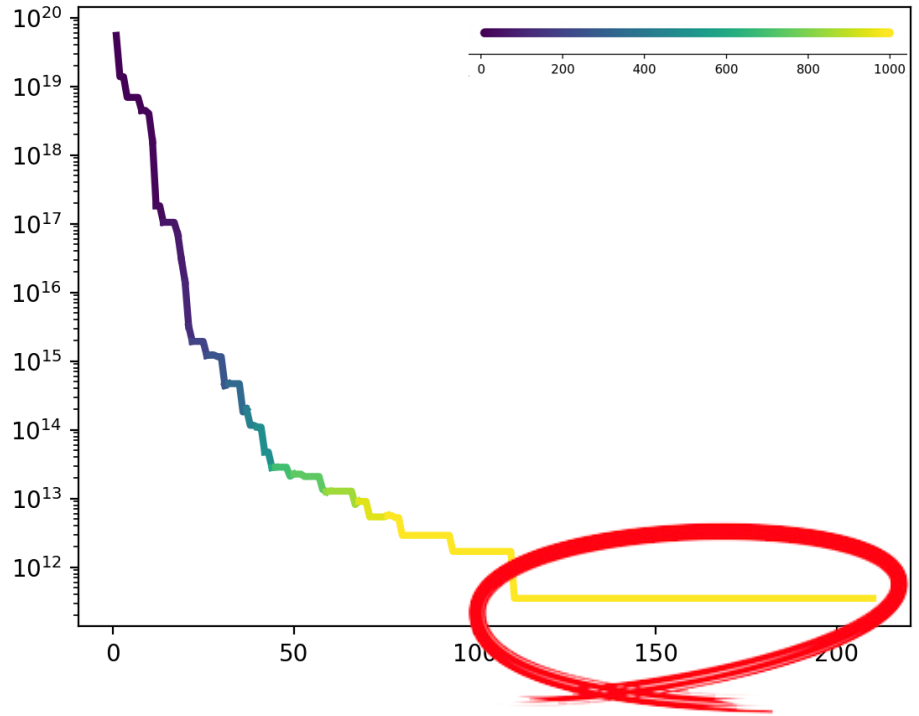
- Алгоритм начинает оптимизацию с размером выборки Монте-Карло $N = 10$
- Размер выборки N постепенно растет до 1000 задач с уменьшением значения оценочной функции

Результаты экспериментов

	Tabu Search		(1+1)-EA		GA	
	|B|	Оценка времен, сек	|B|	Оценка времени, сек	|B|	Оценка времени, сек
Trivium-Toy 64/75	17	4.30e+07	17	**3.19e+07**	22	5.36+07
Trivium-Toy 96/100	34	3.14e+12	33	1.28e+13	40	**2.09+12**
Bivium 177/200	40	4.29e+12	32	2.60e+12	39	**1.49+12**
ASG 72/76	8	**5601.33**	9	5604.8	8	6155.19
ASG 96/112	14	3.95e+06	13	6.76e+06	16	**3.72e+06**
ASG 192/200	47	**1.14e+16**	47	2.27e+18	44	2.84e+17

Проблемы адаптивной стратегии

- Необходимо подстраивать значения под каждый алгоритм отдельно
- Снижение эффективности по мере уменьшения значения оценочной функции



Статистические тесты

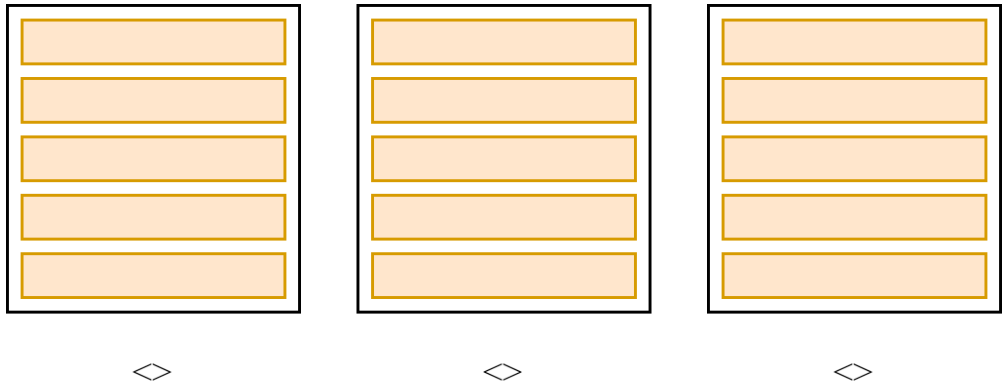
- Для проверки статистических гипотез

Статистическая гипотеза — предположение о виде распределения и свойствах случайной величины

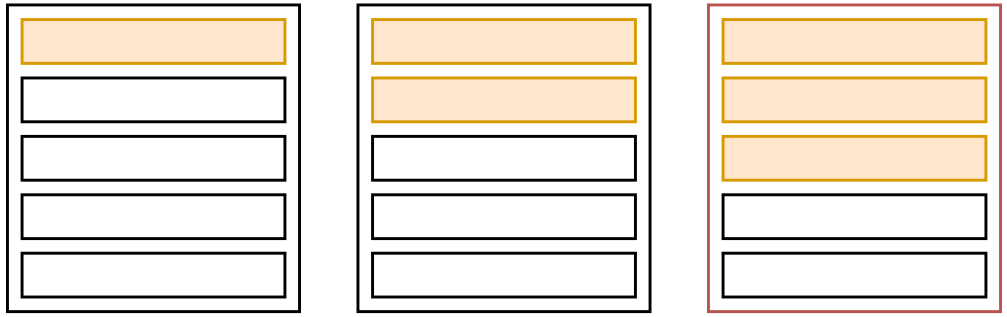
Применение статистических тестов (1)

Статистическая гипотеза: Выборки не различаются

Лучшая особь $\boldsymbol{X}$



Новая особь $\mathbf{Y}$

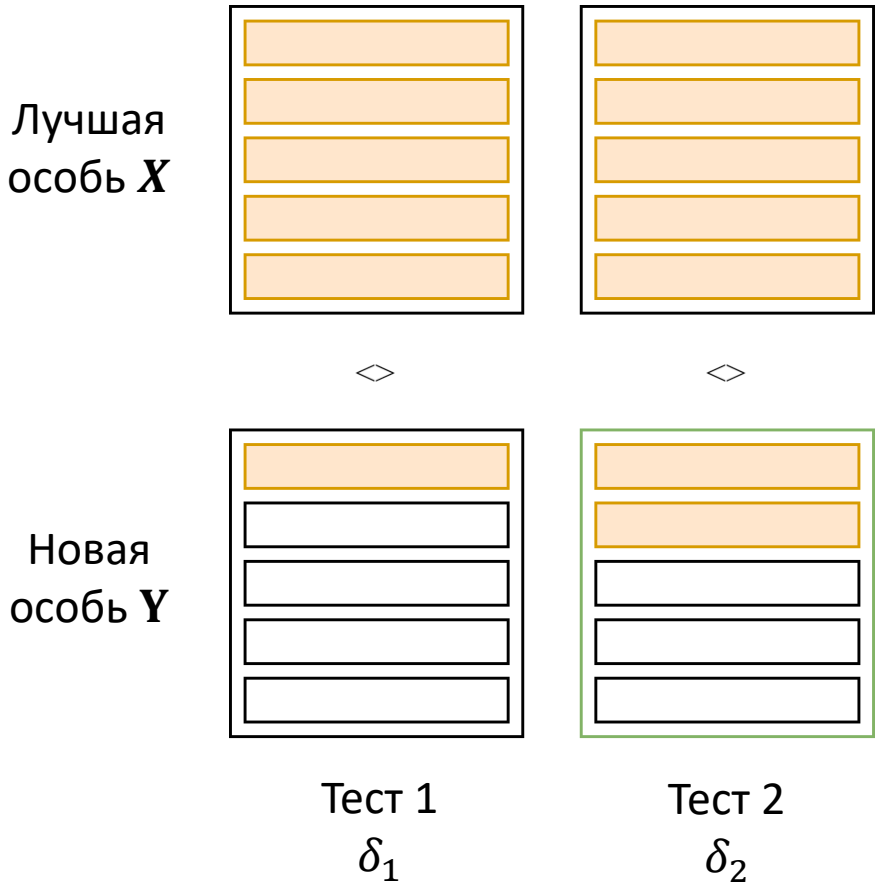


Тест 1 δ_1 | Тест 2 δ_2 | Тест 3 δ_3

$\delta_1 = 0$ — наращиваем выборку
$\delta_2 = 0$ — наращиваем выборку
$\delta_3 < 0$ — особь $\boldsymbol{Y}$ хуже, переходим к следующей

Применение статистических тестов (2)

Статистическая гипотеза: Выборки не различаются

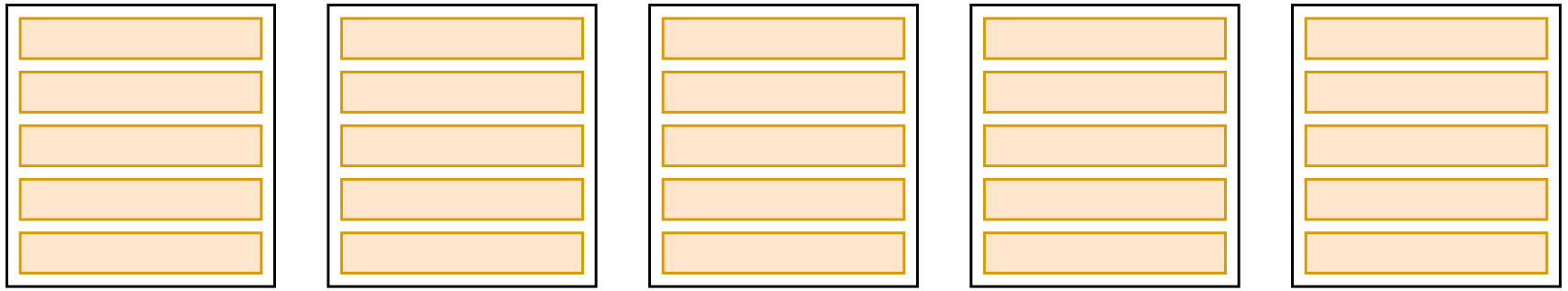


$\delta_1 = 0$ — наращиваем выборку
$\delta_2 > 0$ — особь $\boldsymbol{X}$ хуже, заменяем её новой особью $\boldsymbol{Y}$

Применение статистических тестов (3)

Статистическая гипотеза: Выборки не различаются

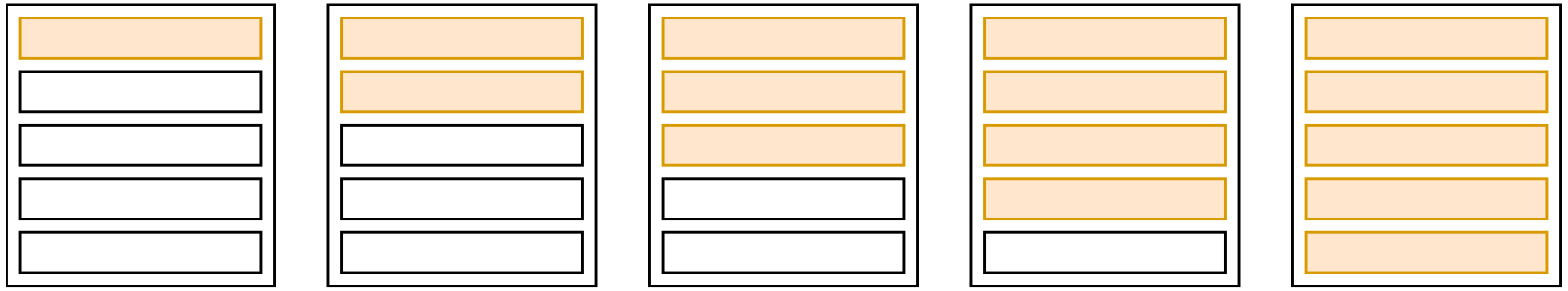
Лучшая особь $\boldsymbol{X}$



$\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = 0$

<> <> <> <> <>

Новая особь $\mathbf{Y}$



Считаем оценочную функцию и выбираем лучшую особь

Тест 1	Тест 2	Тест 3	Тест 4	Тест 5
δ_1	δ_2	δ_3	δ_4	δ_5

Число просмотренных точек

Алгоритм	**Число точек**		
	Со статистическим тестом	Без статистического теста	Во сколько раз больше
A5/1	1471	341	x4.31
Bivium	3616	2439	x1.48
Trivium 64	3398	1323	x2.57
Trivium 96	2494	1299	x1.92

Сравнение результатов

Алгоритм	Статистические тесты		Адаптивная стратегия	
	|B|	Оценка времени, сек	|B|	Оценка времени, сек
ASG 72	10	**4794.55**	9	5604.8
ASG 96	19	**1.56e+06**	16	3.72e+06
Bivium	27	**7.49e+11**	39	1.49e+12
Trivium 64	17	**2.03e+07**	21	3.17e+07
Trivium 96	35	**1.24e+12**	40	2.09e+12

Атака на Alternative Step Generator (ASG 72)

Полученная оценка: 4794.55 сек.

Число атак	Ограничение времени, сек.	Успешные атаки, %	Усредненное время, сек.	Отклонение времени, сек.
1000	1.0	64.8	1039.24	17.43
1000	2.0	100	1081.32	492.72
1000	3.0	100	1592.62	281.56
10000	**2.0**	**100**	**1084.26**	**486.00**

Из расчета на одно ядро процессора Intel®Xeon®E5-2695 v4

Выводы

- Были рассмотрены автоматизированные методы построения декомпозиционных множеств
- Некоторые построенные декомпозиционные множества лучше, чем найденные раннее, однако оценки времени взлома все еще очень велики

- Современные криптографические алгоритмы надежны!

Спасибо за внимание!

alpavlenko@itmo.ru

Сириус
Образовательный центр

ITMOre than a UNIVERSITY