

# Циклический комбинированный алгоритм для планирования назначения задач суперкомпьютера

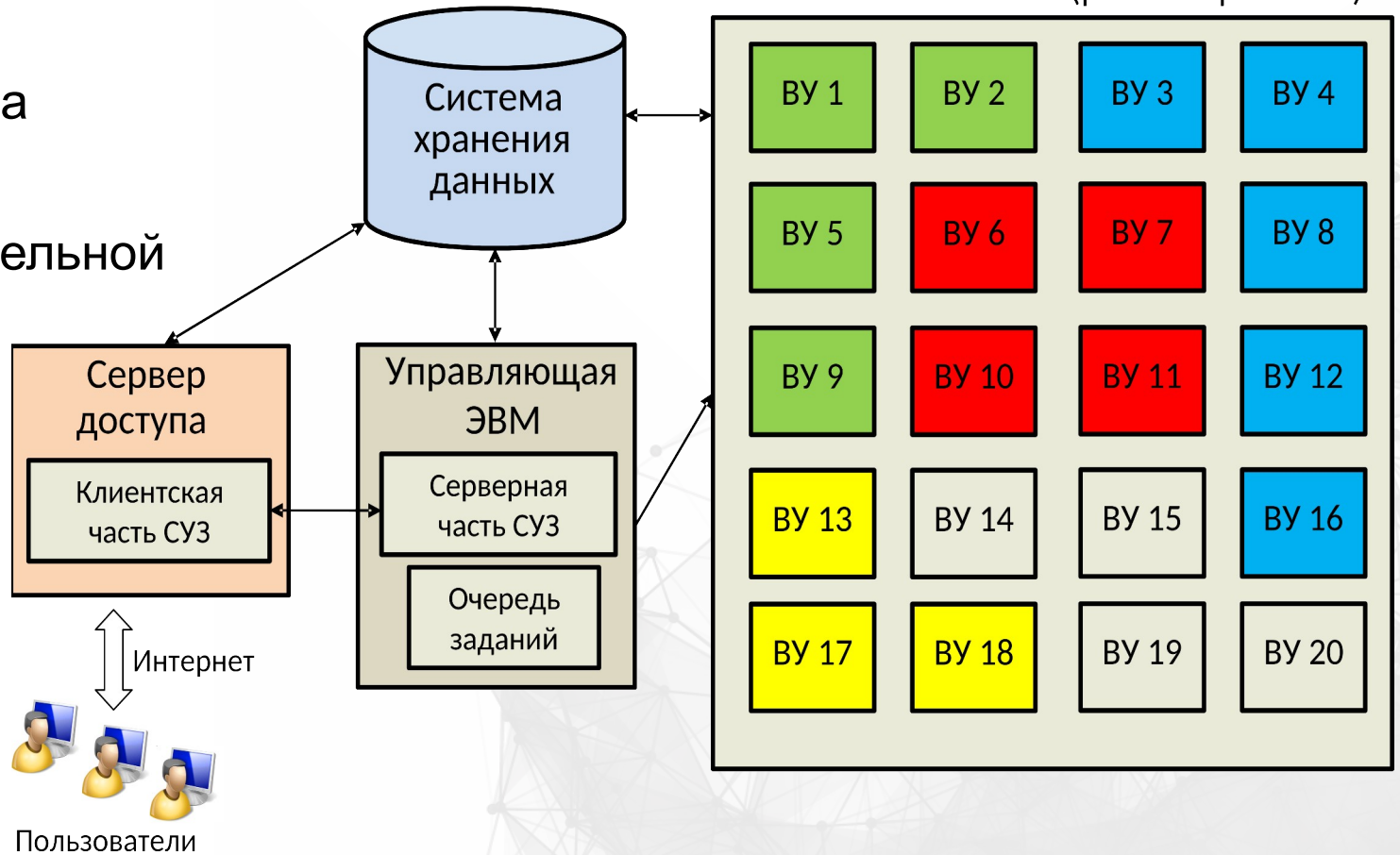
Баранов А.В, Аладышев О.С., Брагин К.А.

НИЦ "Курчатовский Институт"  
Отделение суперкомпьютерных систем и параллельных  
вычислений (МСЦ РАН)

# Система коллективного пользования

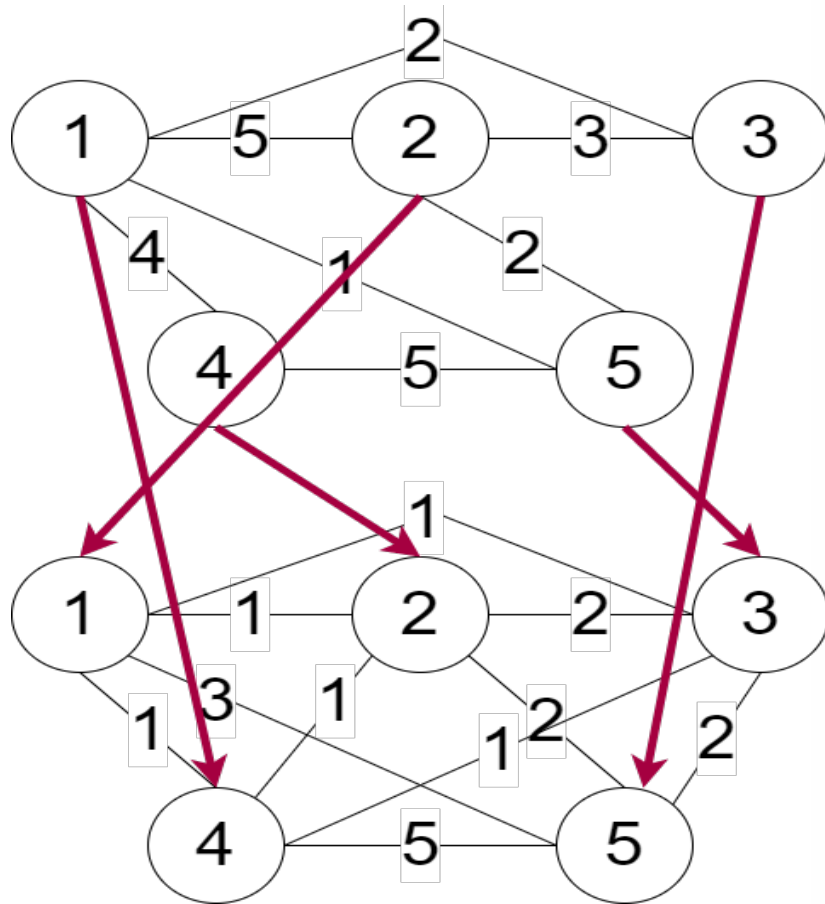
**Задание (Job)** – специальный информационный объект:

- параллельная программа
- требования к ресурсам
- входные данные параллельной программы

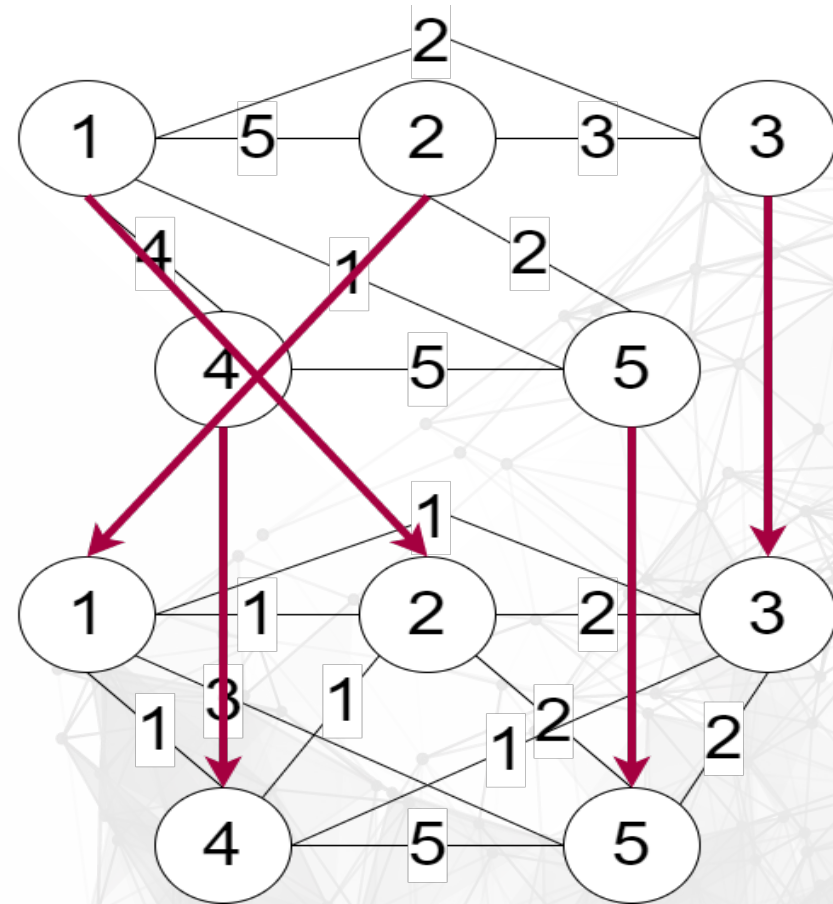




# Точность отображения



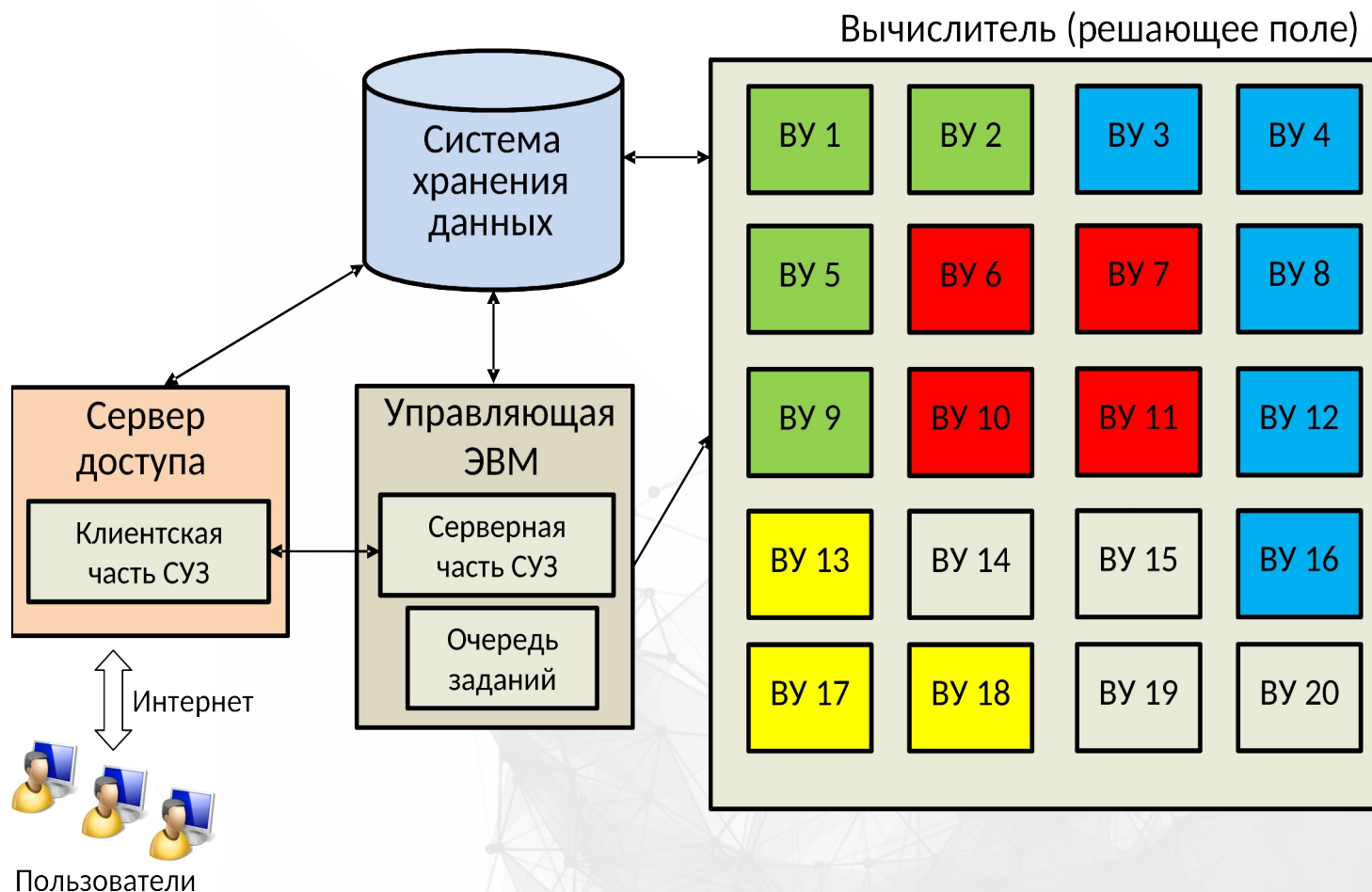
Значение  $F(X)$ : 74



Значение  $F(X)$ : 98

# Отображение в режиме коллективного пользования

- Отображение находится для набора свободных в данный момент узлов
- Конфигурация выбранных узлов заранее неизвестна
- Решение задачи о назначении вычисляется на этом же наборе узлов
- Время нахождения решения не должно превышать системные таймауты





# Параллельный генетический алгоритм с симуляцией отжига - PGSA

–Баранов А.В., Метод и алгоритмы осуществления оптимального отображения параллельной программы на структуру многопроцессорного вычислителя. 2000

–Ş. M. Türkkahraman and D. Öz, An Improved Hybrid Genetic Algorithm for the Quadratic Assignment Problem. 2021

–А.В. Баранов, Е.А. Киселев, П.Н. Телегин, А.А. Сорокин, Программное средство GraphHunter поиска отображения параллельной программы на структуру суперкомпьютерной системы. 2022

–Baranov, A.V., Kiselev, E.A., Shabanov, B.M. et al. Comparison of Three Job Mapping Algorithms for Supercomputer Resource Managers. Lobachevskii J Math. 2023

UGR-Metaheuristics



GraphHunter

Среднее отклонение от идеального решения составляет 18.64%

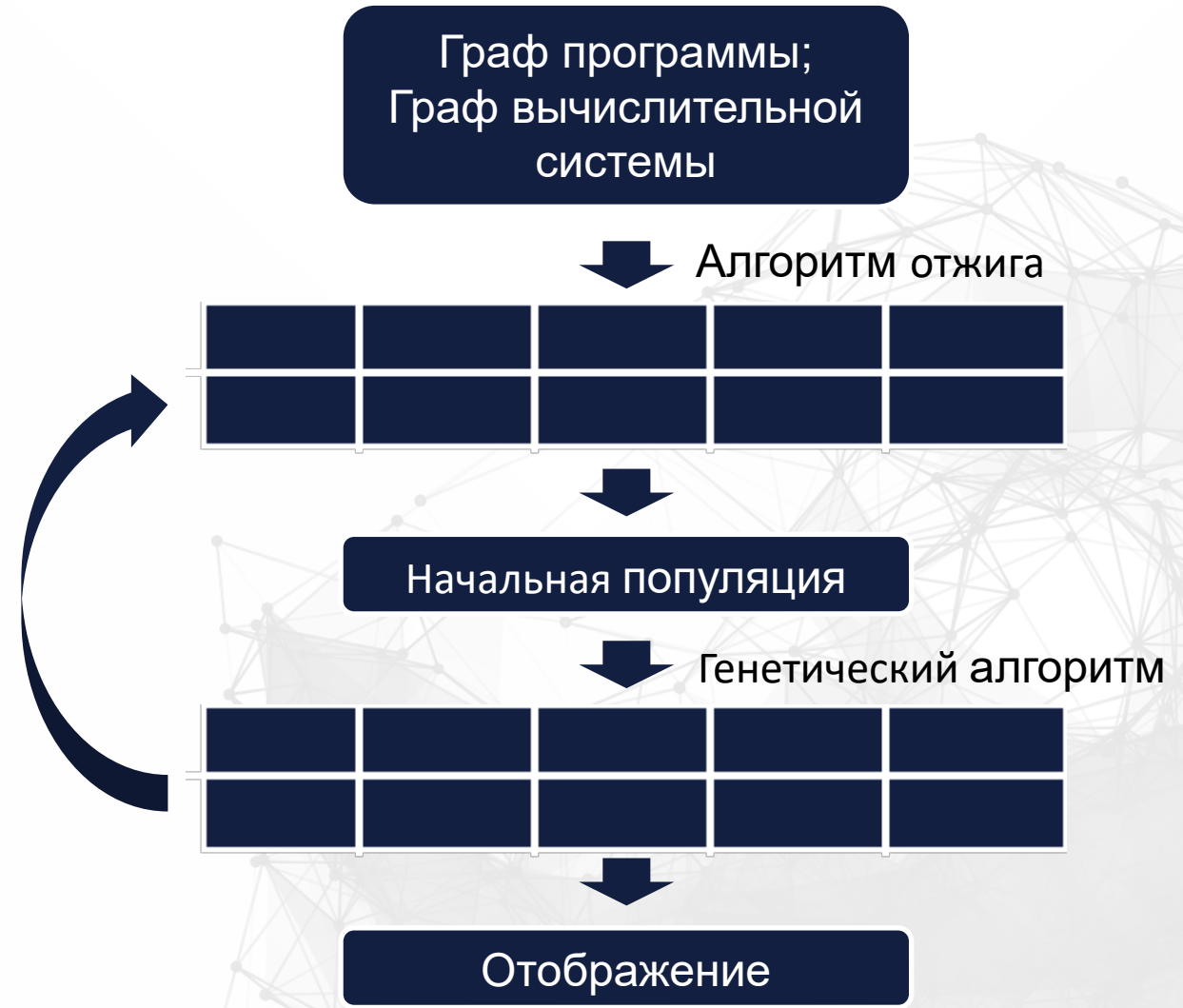


# Циклический параллельный генетический алгоритм с симуляцией отжига - CPGSA

**Входные данные:** граф программы и граф вычислительной системы;

**Выходные данные:** Отображение графа программы на граф вычислительной системы;

В результате **защивания** алгоритма, решение, получаемое в результате генетического алгоритма представляется начальным решением для алгоритма отжига. После прохождения заданного числа итераций решение генетического алгоритма считается оптимальным и выводится как конечное решение.



# Сравниваемые алгоритмы и тестовые данные

**Использовались тестовые данные из набора:**

<http://mistic.heig-vd.ch/taillard/ Problemes.dir/qap.dir/qap.html>.

**Описание наборов см. в работе:**

DOI. Drezner, Z., Hahn, P.M. & Taillard, É.D. Recent Advances for the Quadratic Assignment

Problem with Special Emphasis on Instances that are Difficult for Meta-Heuristic Methods. Ann Oper Res 139, 65–94 (2005).

<https://doi.org/10.1007/s10479-005-3444-z>

Выбранные  
наборы графов:

–tai27

–tai45

–tai75

–tai125

–tai175

Для сравнения использовались алгоритмы:

–Параллельный алгоритм имитации отжига - **Отжига**

–Параллельный алгоритм генетического отбора - **Генетический**

–Параллельный генетический алгоритм с симуляцией отжига - **PGSA**

–Циклический параллельный генетический алгоритм с симуляцией отжига - **CPGS**

–Оптимальное решение - **Оптимальный**



# Вычислительная система для проведения экспериментов

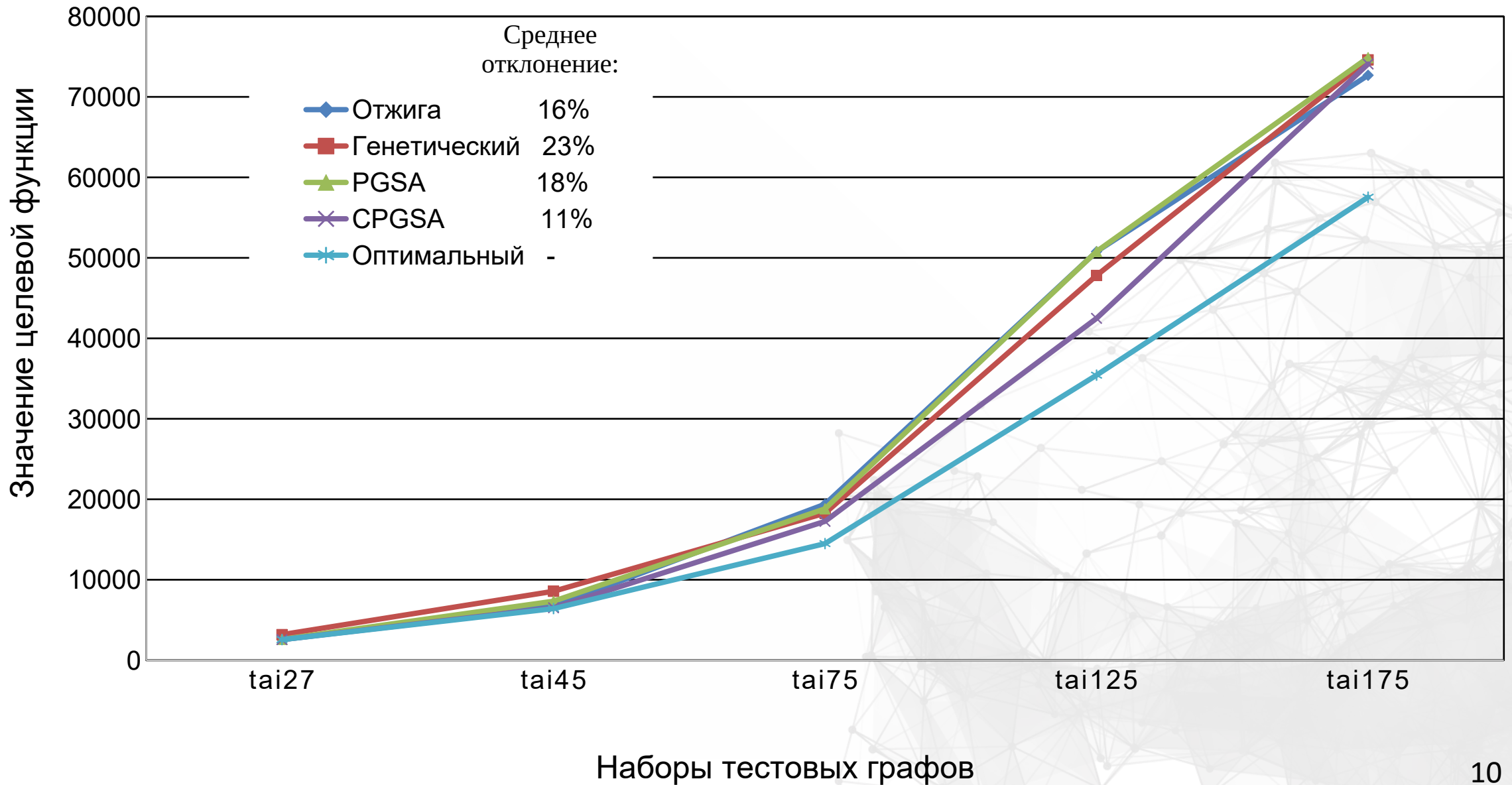


Эксперименты проводились на разделе Broadwell суперкомпьютера МВС-10П ОП, содержащим 136 узлов с характеристиками:

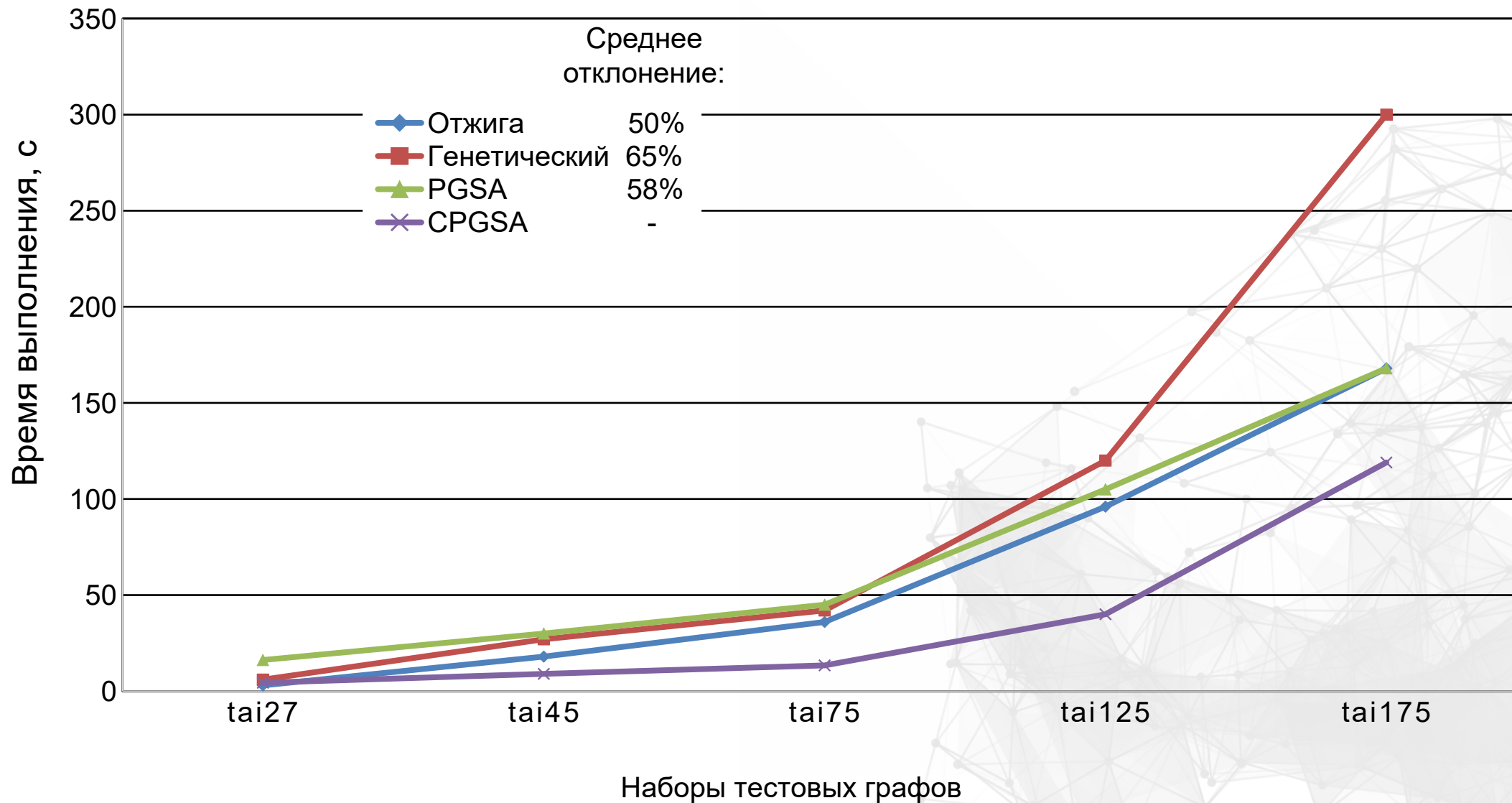
- 2 Intel Xeon E5-2697Av4 процессора;
- 32 физических, 64 виртуальных ядра на каждом узле;
- 128 GB of RAM;
- Intel Omni-Path interconnect.

Эксперименты проводились на 64 ядрах

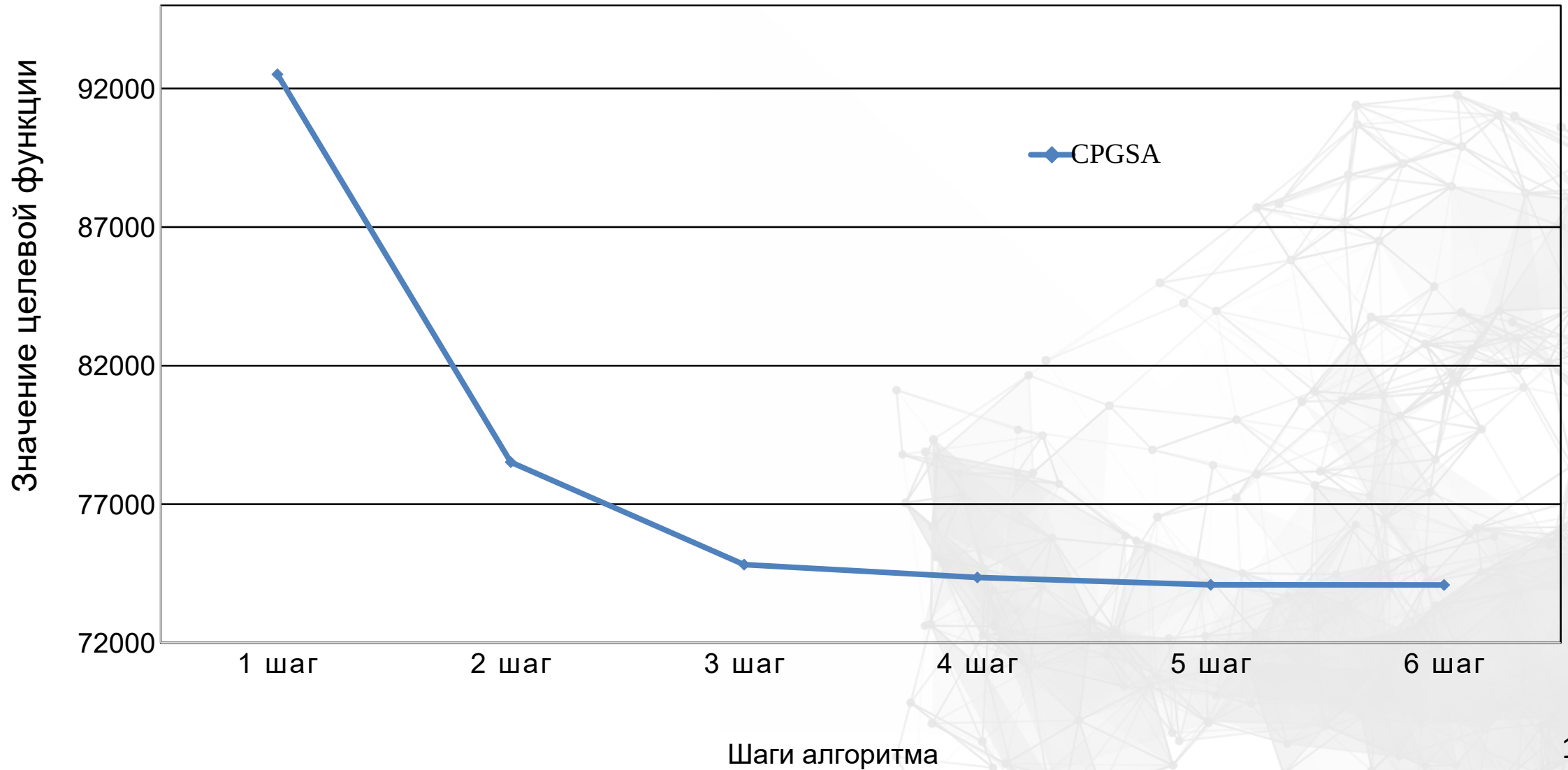
# Зависимость значения целевой функции от размера графа



# Зависимость времени выполнения алгоритма от размера графа



# Зависимость значения целевой функции от шагов алгоритма для набора tai175



## Параметры и их модификаторы

**Модификатор** - числовое значение в промежутке от 0 до 2, определяющее изменение каждого из параметров работы алгоритмов на каждом шагу.

Вычисление модификаторов производится **однократно** для графов заданного размера.

Для определения набора модификаторов используется **алгоритм отжига**:

1. Нахождение базового решения без учета модификаторов
2. Устанавливаем начальные значения модификаторов = 1
3. Пока температура больше минимальной:
  1. Изменяем лучшие модификаторы
  2. Находим решение с текущими модификаторами
  3. Если значение целевой функции меньше минимальной, сохраняем текущие модификаторы как лучшие
4. Выводим значение лучших модификаторов

## Параметры работы алгоритмов:

### Алгоритм отжига:

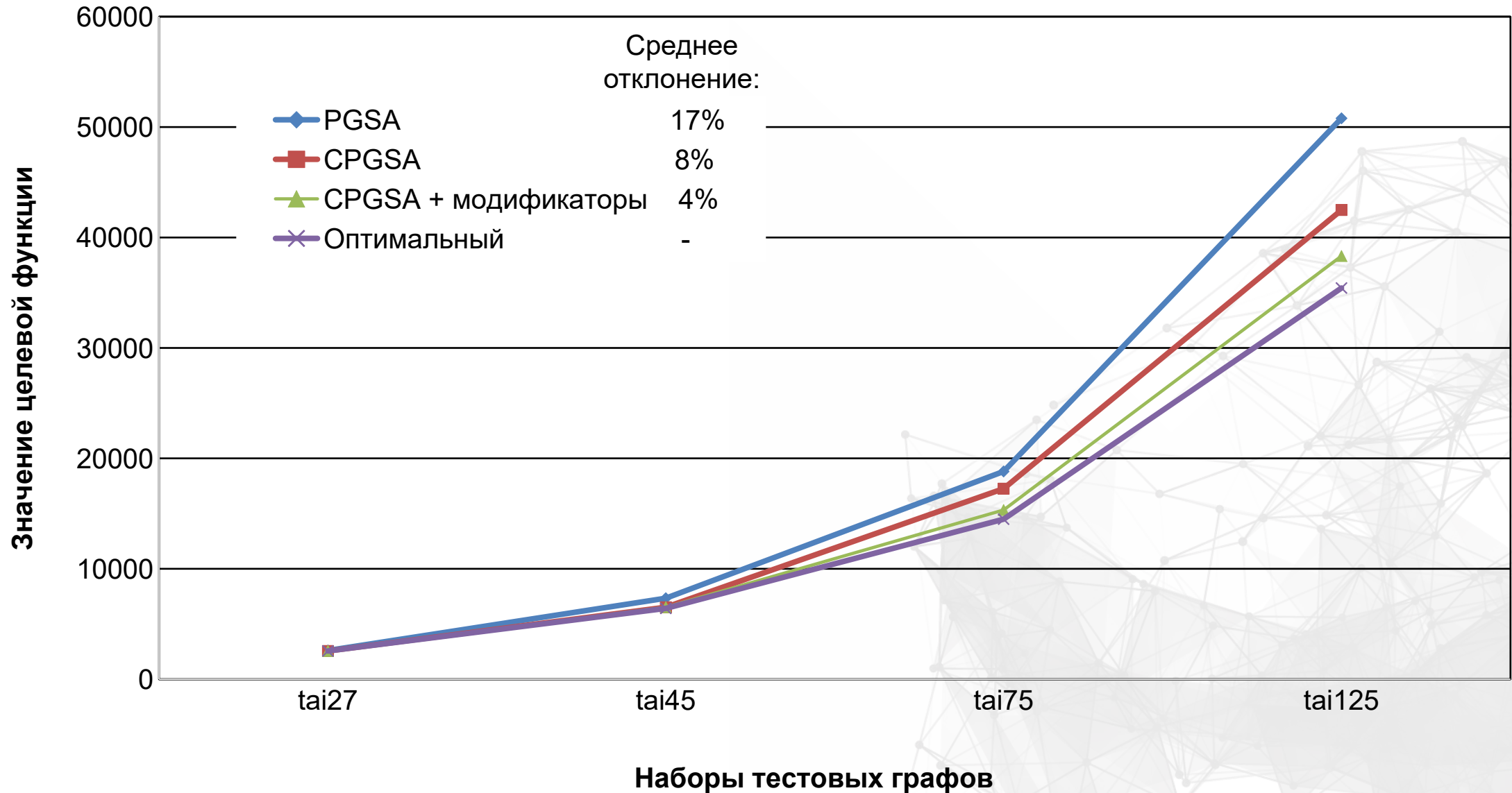
- Количество итераций при одном значении температуры
- Количество успешно принятых решений при одном значении температуры
- Конечная температура
- Коэффициент понижения температуры
- Число итераций последовательного алгоритма поиска
- Количество коммуникационных обменов

### Алгоритм генетического отбора:

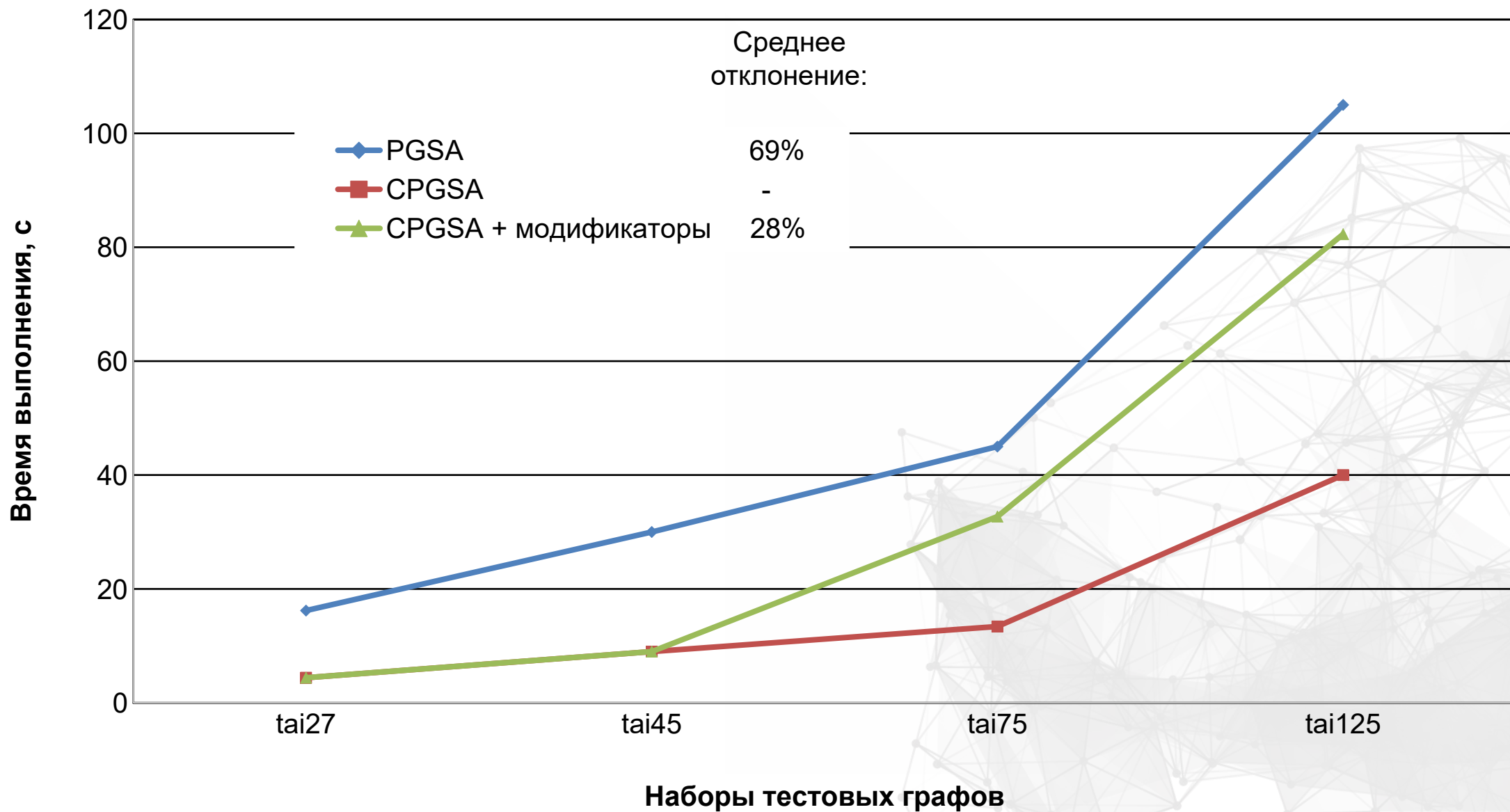
- Вероятность применения операции кроссовера
- Вероятность применения операции мутации
- Количество мигрирующих особей
- Число членов популяции
- Число коммуникационных обменов



# Зависимость значения целевой функции от размера графа



# Зависимость времени выполнения алгоритма от размера графа



# Выводы и перспективы

## Выводы:

- 1) За счет применения циклической схемы алгоритма и применения модификаторов его параметров повышены точность отображения программного графа на граф вычислительной системы и скорость поиска отображения
- 2) Для поиска оптимальных модификаторов параметров алгоритма отображения применен алгоритм моделирования отжига
- 3) Для графов средней размерности был получен алгоритм поиска отображения, близкого к оптимальному, за время, допустимое при применении алгоритма в системах коллективного пользования.

## Перспективы:

- 1) Проанализировать работу алгоритма на графах большой размерности
- 2) Рассмотреть применение нейронных сетей для поиска параметров алгоритма отображения
- 3) Дальнейшая оптимизация параметров алгоритма отображения

**Благодарю за внимание**