

Летняя школа 2016 по параллельному программированию

Поиск максимума функции плотности множества точек на прямой:

- метод имитации отжига
- метод ABC

Студент: Садыкова А.А.

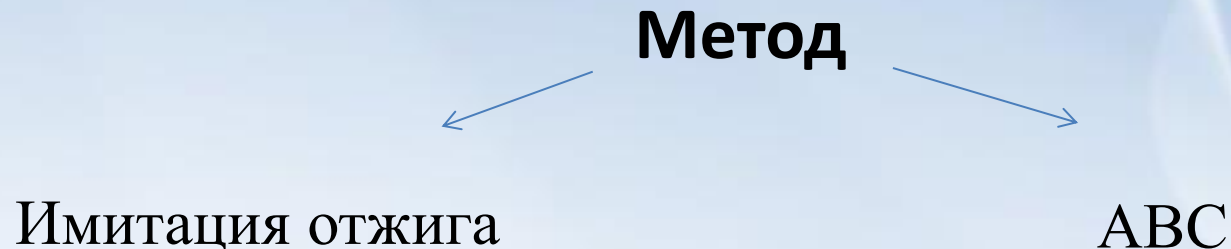
Руководители: Ахмед-Заки Д.Ж.
Городничев М.А.
Перепелкин В.А.

План

- **Введение**
- **Цель**
- **Теоретическая часть**
- **Реализация задачи**
- **Тестирование и результаты**
- **Планы**

Введение

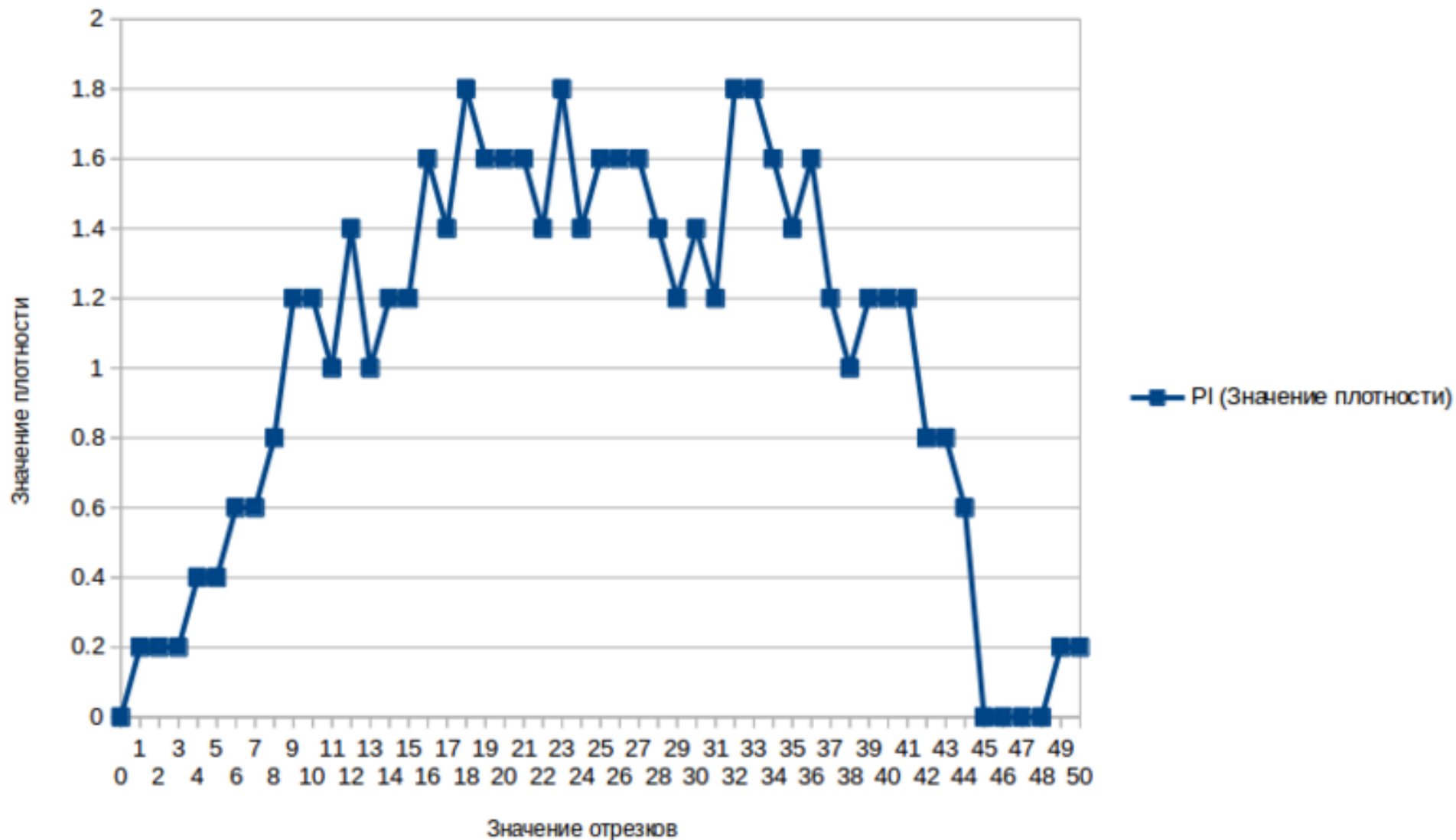
Оптимизация — нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного пространства.



Цель

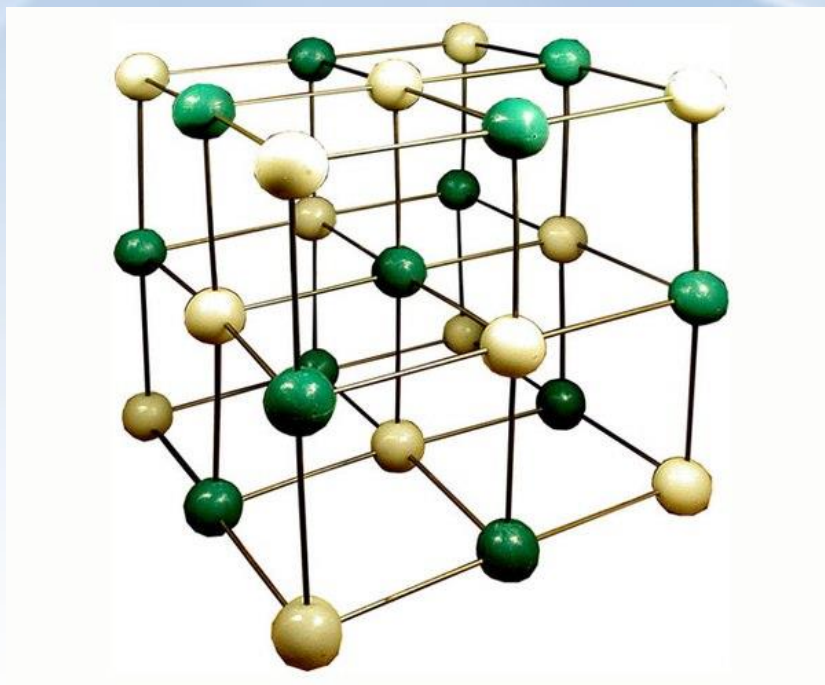
- **Реализация методов:**
 - **Имитация отжига**
 - **ABC**
- **Определение оптимального варианта;**

График функции плотности



Теоретическая часть

Алгоритм имитации отжига (Simulated annealing) — общий алгоритмический метод решения задачи глобальной оптимизации.



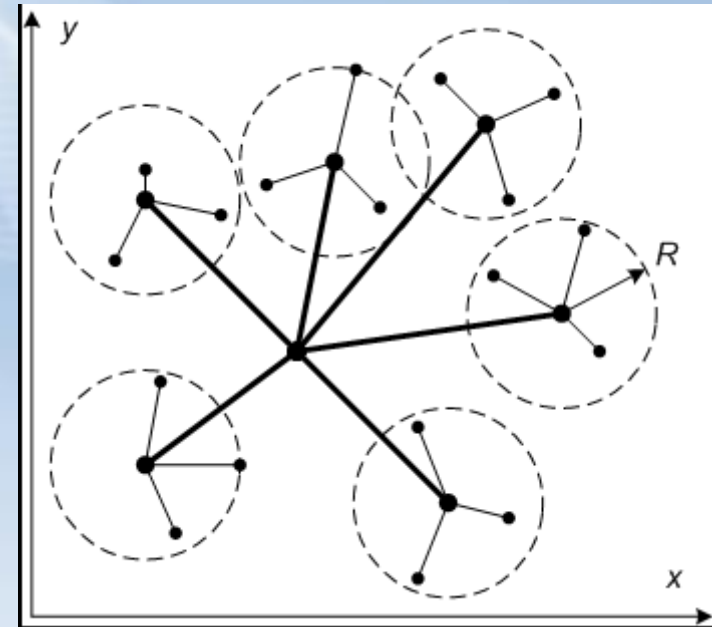
Алгоритм для метода имитации отжига:

- На входе: минимальная температура t_{min} , начальная температура t_{max}
- Задаём произвольное первое состояние s_1
- $t_1 = t_{max}$
- Пока $t_i > t_{min}$
 - $s_c = F(s_{i-1})$
 - $\Delta E = E(s_c) - E(s_{i-1})$
 - Если $\Delta E \leq 0$, тогда $s_i = s_c$
 - Если $\Delta E > 0$ переход осуществляется с вероятностью $P(\Delta E) = e^{-\Delta E/t_i}$
 - Понижаем температуру $t_{i+1} = T(i)$
- Возвращаем последнее состояние s

Алгоритм пчелиной колонии (*artificial bee colony, ABC*) — один из полиномиальных эвристических алгоритмов для решения оптимизационных задач, основан на имитации поведения колонии медоносных пчел при сборе нектара в природе. Предложен Д. Карабога в 2005 г.

Основной целью работы пчелиной колонии в природе является разведка пространства вокруг улья в целях поиска нектара с последующим его сбором.

- Первоначально источники питания производятся для всех работающих пчел
- ПОВТОРЕНИЕ
 - Каждая пчела (Employed bee, EB) ищет источник, а затем оценивает его количество нектара и танцует в улье
 - Каждый зритель (Onlooker Bee, OB) наблюдает танец пчел (EB) и выбирает одну из своих источников в зависимости от танцев, а затем идет к этому источнику. После оценивает количество нектара.
 - Покинутые источники питания будут определены и заменены на новые источники питания, обнаруженные разведчиками (Scout bee, SB).
 - Найденный лучший источник пищи сохраняется в памяти.
- ПОКА (требование)

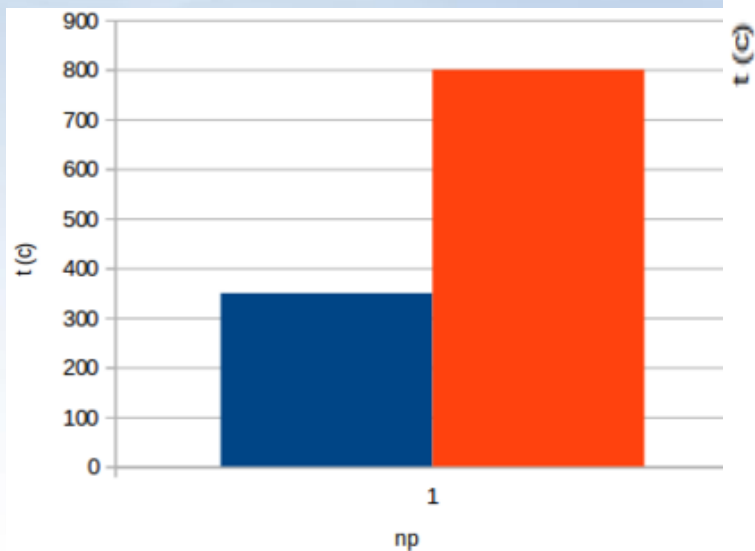


Схематичное изображение стратегии разведки двумерного пространства (жирные линии — вылеты разведчиков, тонкие линии — уточнение решений рабочими пчелами)

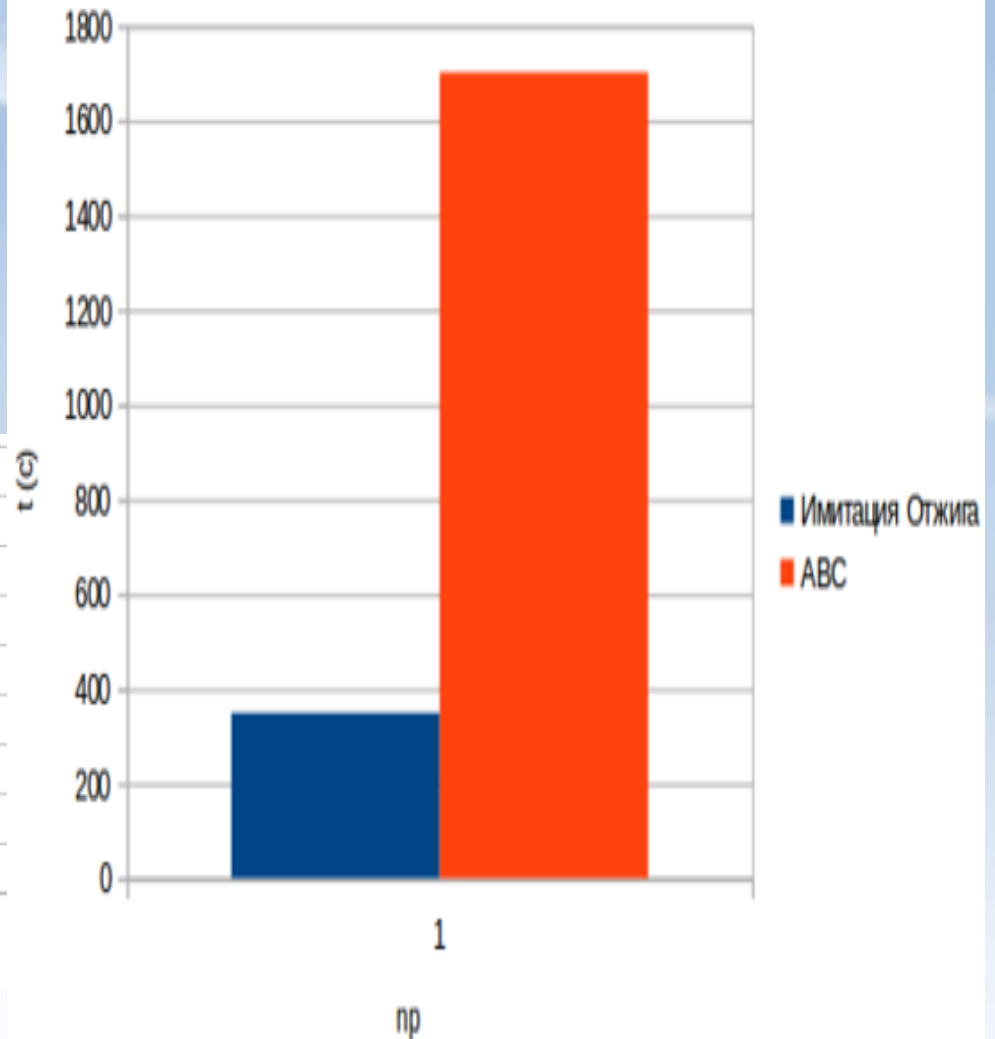


Тестирование

Последовательная реализация: время выполнения

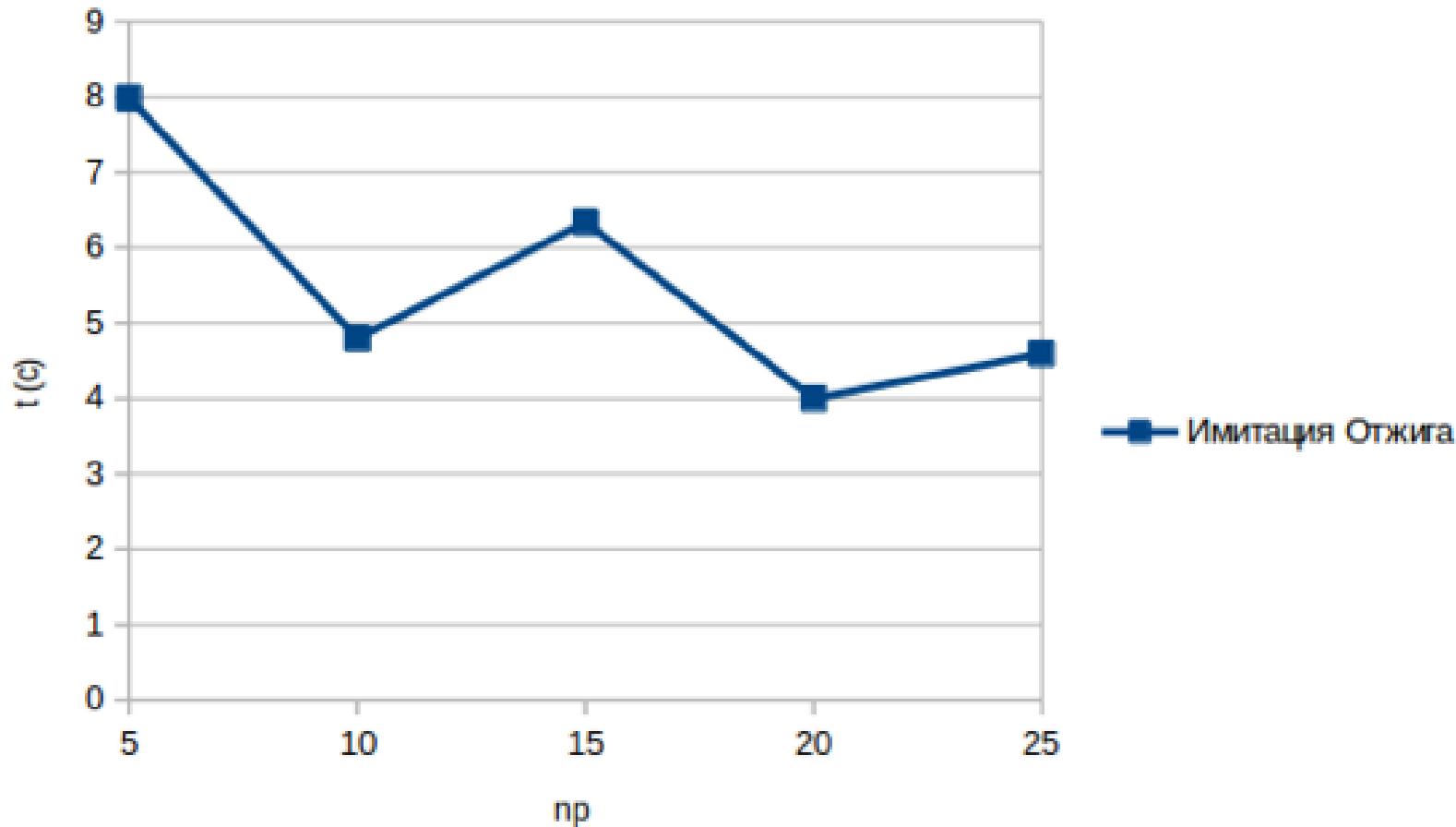


Cycle_number_of_ABC= 100

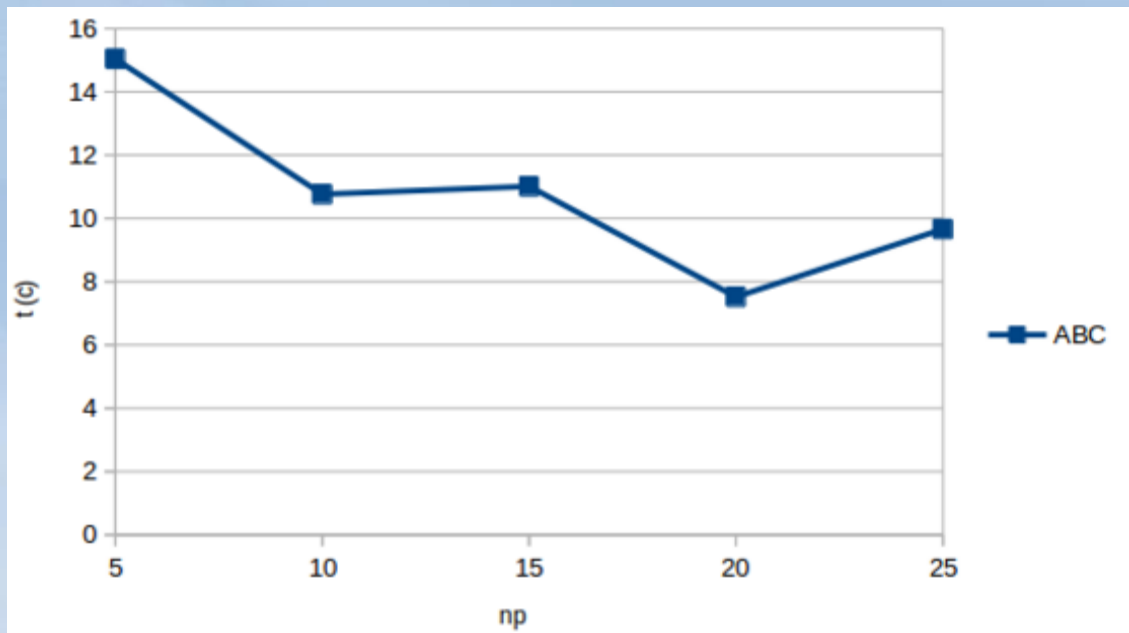


Cycle_number_of_ABC= 10000

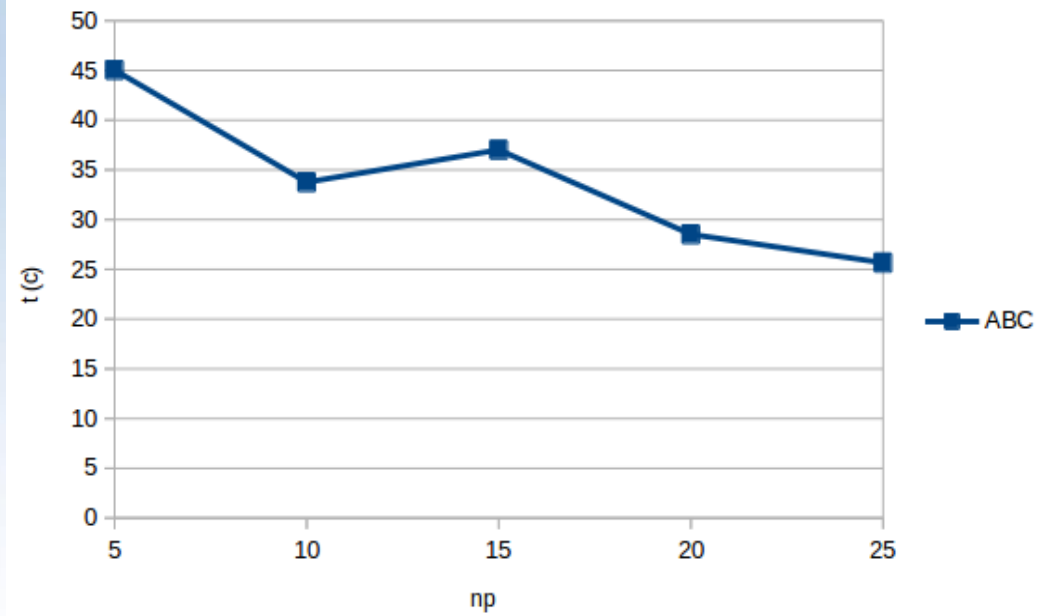
Параллельная реализация: время выполнения



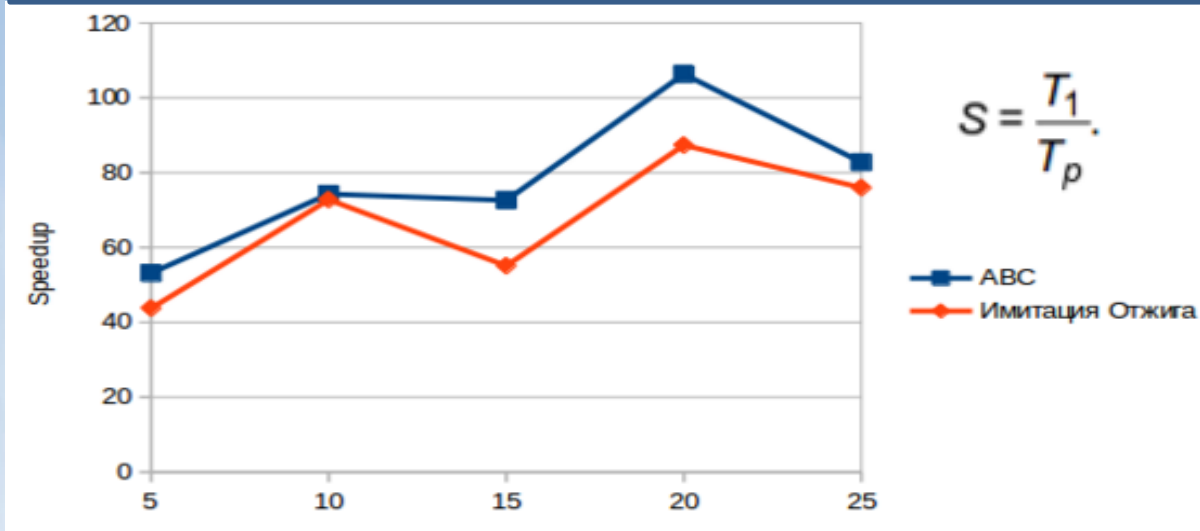
Cycle_number_of_ABC= 100



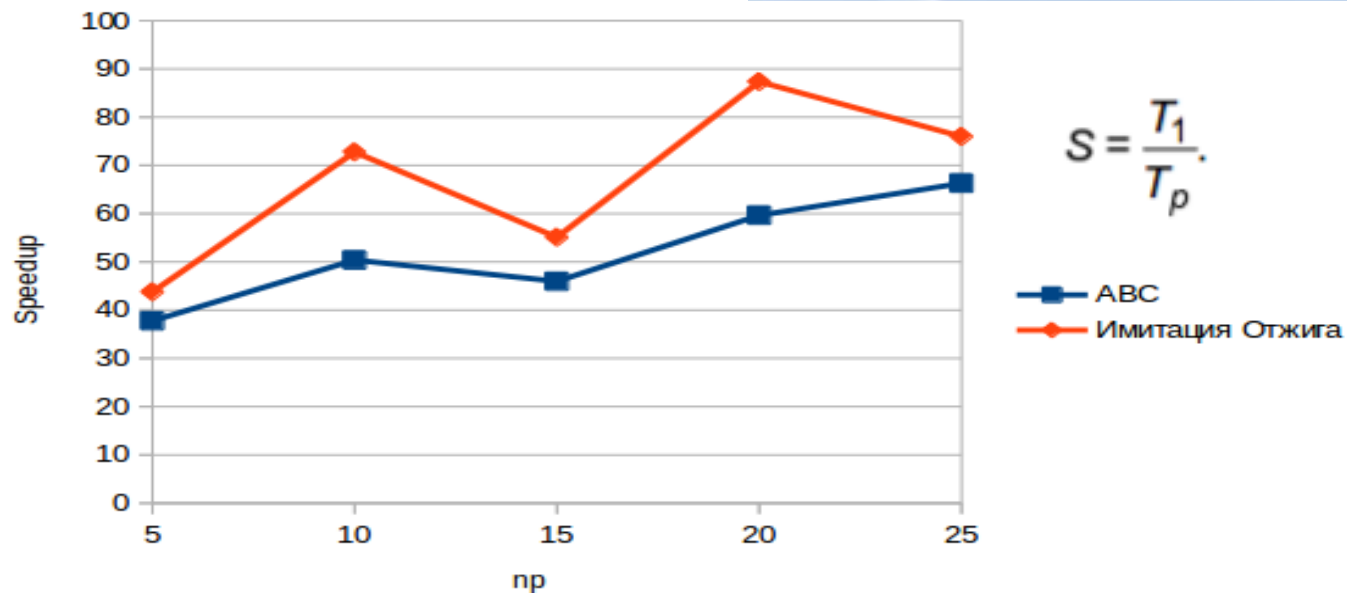
Cycle_number_of_ABC = 10000



Параллельная реализация: ускорение

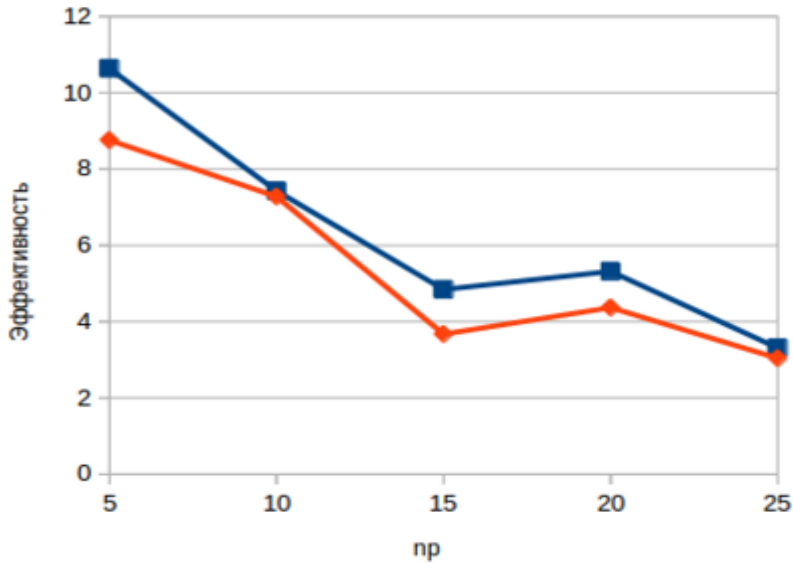


Cycle_number
_of_ABC= 100



Cycle_number_of_ABC
= 10000

Параллельная реализация: эффективность

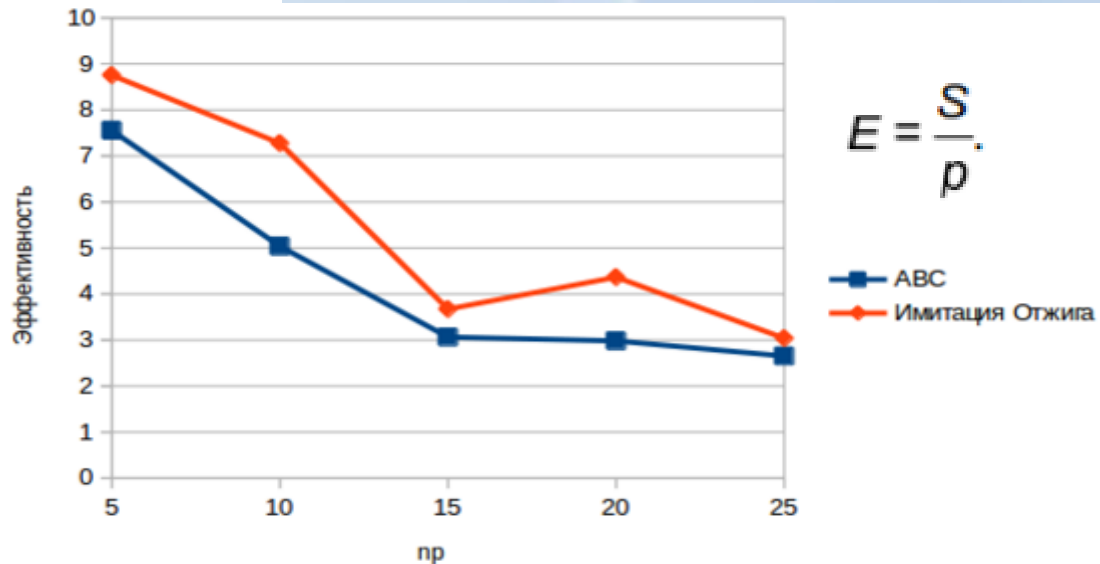


$$E = \frac{S}{p}$$

■ ABC
◆ Имитация Отжига

Cycle_number_of_ABC
= 100

Cycle_number_of_ABC
= 10000



$$E = \frac{S}{p}$$

■ ABC
◆ Имитация Отжига

Заключение

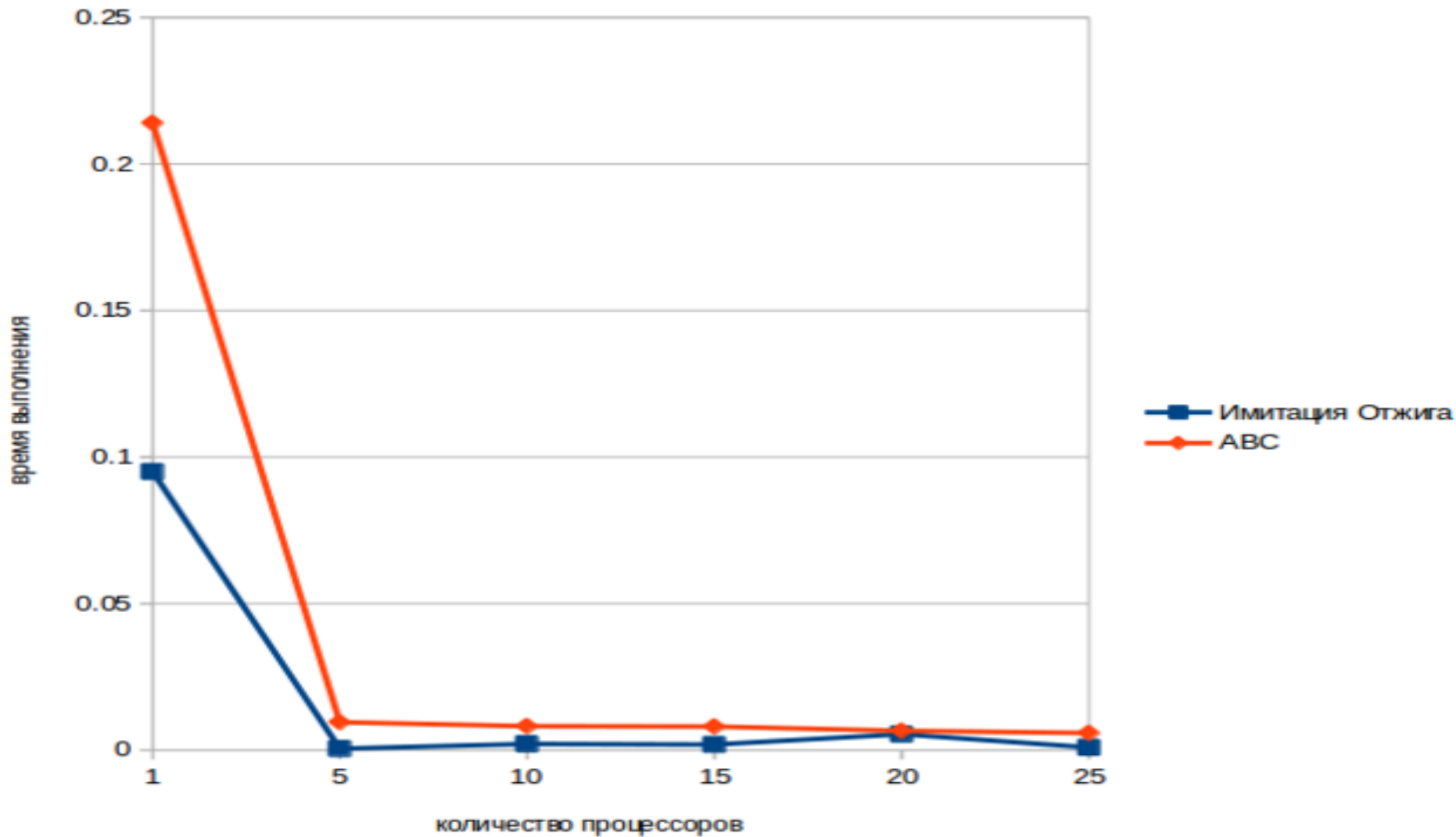
- Задача реализована на C++ (последовательно), а так же на MPI (параллельно)
- Проведены запуски и тесты для определения времени, ускорения и производительности.
- По результатам, оптимальный метод – метод имитации отжига

Планы

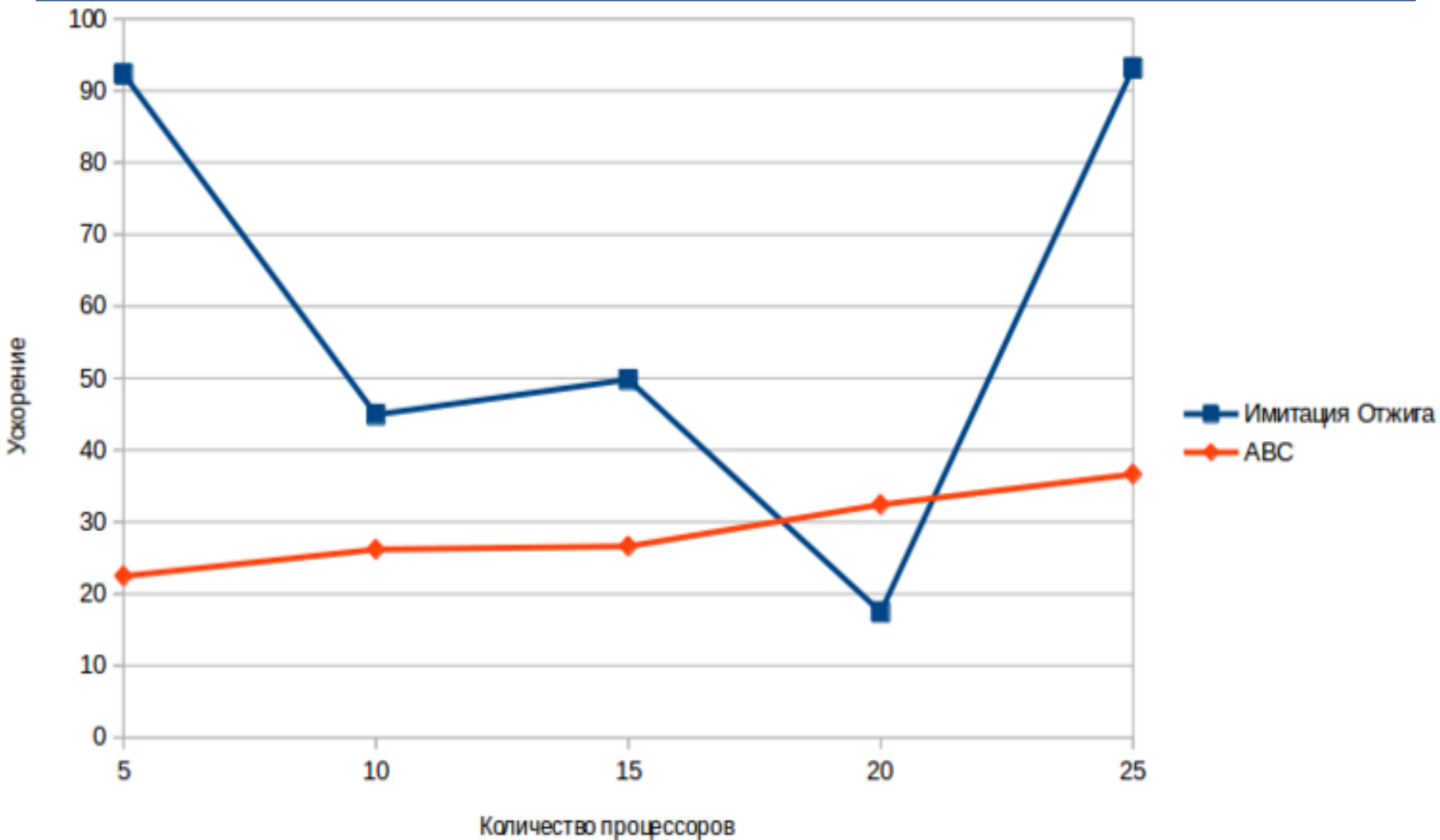
- Исследовать параметры ABC
- Сделать параллельную сортировку
- Параллельно сделать расчет плотности

Спасибо за внимание!

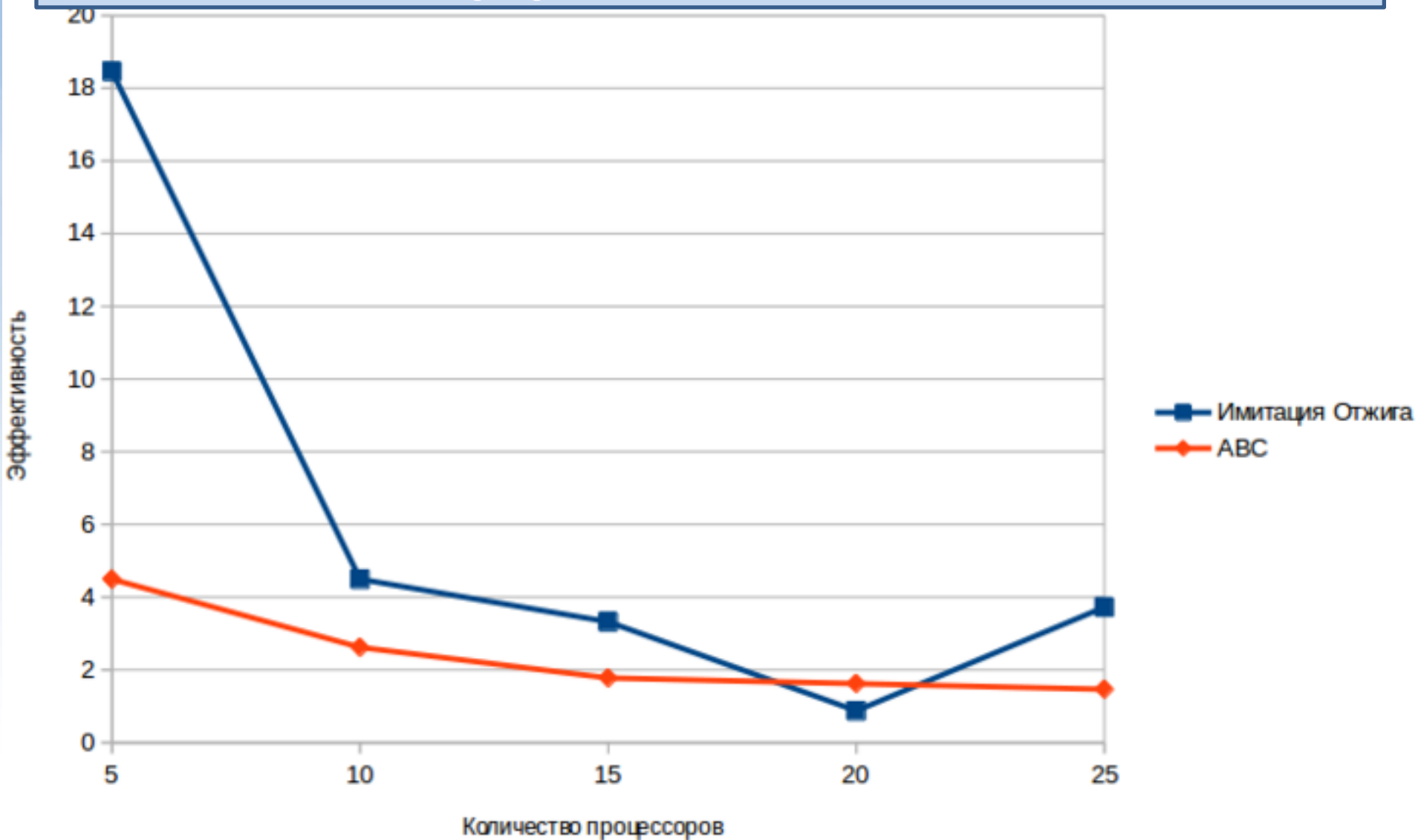
Параллельная реализация: время выполнения



Параллельная реализация: ускорение



Параллельная реализация: эффективность



Эвристический алгоритм — это алгоритм решения задачи, правильность которого для всех возможных случаев не доказана, но про который известно, что он даёт достаточно хорошее решение в большинстве случаев. В действительности может быть даже известно (то есть доказано) то, что эвристический алгоритм формально неверен. Его всё равно можно применять, если при этом он даёт неверный результат только в отдельных, достаточно редких и хорошо выделяемых случаях, или же даёт неточный, но всё же приемлемый результат.

В теории **алгоритмов** классом P (от англ. **polynomial**) называют множество задач, для которых существуют «быстрые» **алгоритмы** решения (время работы которых полиномиально зависит от размера входных данных). Класс P включён в более широкие классы сложности **алгоритмов**.

Let $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$ represent the i^{th} solution in the swarm, Where n is the dimension size. Each employed bee X_i generates a new candidate solution V_i in the neighborhood of its present position as equation below:

$$V_{i_k} = X_{i_k} + \Phi_{i_k} \times (X_{i_k} - X_{j_k})$$

Where X_j is a randomly selected candidate solution ($i \neq j$), k is a random dimension index selected from the set $\{1, 2, \dots, n\}$, and Φ_{i_k} is a random number within $[-1, 1]$. Once the new candidate solution V_i is generated, a greedy selection is used. If the fitness value of V_i is better than that of its parent X_i , then update X_i with V_i ; otherwise keep X_i unchanged. After all employed bees complete the search process; they share the information of their food sources with the onlooker bees through waggle dances. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. This probabilistic selection is really a roulette wheel selection mechanism which is described as equation below:

$$P_i = \frac{fit_i}{\sum_j fit_j}$$

Where fit_i is the fitness value of the i^{th} solution in the swarm. As seen, the better the solution i , the higher the probability of the i^{th} food source selected. If a position cannot be improved over a predefined number (called limit) of cycles, then the food source is abandoned. Assume that the abandoned source is X_i , and then the scout bee discovers a new food source to be replaced with i^{th} as equation below:

$$X_{i_k} = lb_j + rand(0, 1) \times (ub_j - lb_j)$$

Where $rand(0, 1)$ is a random number within $[0, 1]$ based on a normal distribution and lb, ub , are lower and upper boundaries of the i^{th} dimension, respectively.