

# ИССЛЕДОВАНИЕ РАЗЛИЧНЫХ СПОСОБОВ РАСПАРАЛЛЕЛИВАНИЯ МЕТОДА ИМИТАЦИИ ОТЖИГА ДЛЯ ПОИСКА ГЛОБАЛЬНОГО ЭКСТРЕМУМА

А.В. Высоцкий, А.С. Тараканов, К.И. Шоломов, Н.Е. Тимофеева, А.А. Ерофтиев

## Введение

Современные задачи науки и техники, такие как моделирование климата, геновая инженерия, проектирование интегральных схем, анализ загрязнения окружающей среды, создание лекарственных препаратов, наноматериалов и др. требуют для своего решения ЭВМ с огромной вычислительной производительностью. Единичные рабочие станции уже не способны в короткие сроки решать вышеперечисленные задачи. Здесь и возникает необходимость использования параллельных вычислений, когда над общей задачей работают десятки, сотни и даже тысячи рабочих станций.

Цель данной работы: реализация и проведение сравнительного анализа методов распараллеливания алгоритма нахождения глобального минимума многоэкстремальной целевой функции многих переменных с явными ограничениями (ограничениями типа равенств), основанного на стохастическом методе имитации отжига.

При этом решались следующие задачи:

- реализация системы параллельных вычислений под управлением программной платформы MPICH2 на базе ресурсов лаборатории Теоретических проблем информатики и ее приложений факультета КНИИТ СГУ им. Н. Г. Чернышевского;
- поиск эффективных способов распараллеливания алгоритма, с точки зрения минимизации времени и повышения надежности вычислений;
- программная реализация поиска глобального экстремума методом имитации отжига, распараллеленного различными способами;
- проведение численного эксперимента для оценки эффективности способов распараллеливания;
- анализ результатов численного эксперимента.

## Организация кластера

Существует большое количество стандартов для организации параллельных вычислений. Наш выбор пал на стандарт MPI, в частности на программный продукт Арагонской лаборатории MPICH2. Это сделано по нескольким причинам:

- MPICH2 написан авторами спецификации;
- является наиболее распространенным;
- наличие библиотек программирования (заголовочные и библиотечные файлы для языков C, C++ и FORTRAN) и загрузчика приложений;
- является свободно распространяемым ПО.

В составе лаборатории имеется десять ЭВМ cIntelCore i3 2120(3.3 GHz), ОЗУ DDR3 4096 Мб (1333 МГц), объединенных в локальную сеть, что позволяло одновременно запускать решаемую задачу на 40 узлах.

Для организации кластера на каждом из ПК был установлен программный пакет MPICH2 и произведена соответствующая настройка:

- создание общего сетевого ресурса;
- установка соответствующих прав доступа установленной службы;
- организация связей между ЭВМ через вышеупомянутый программный пакет.

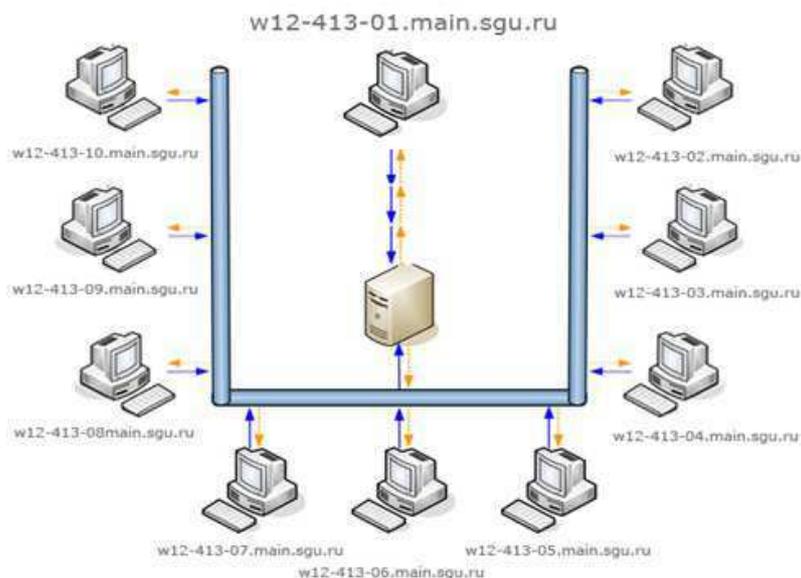


Рис. 1 Схема организованного кластера

### Параллельные варианты метода имитации отжига

Появление и развитие параллельных вычислительных систем привело к развитию нескольких подходов к распараллеливанию метода имитации отжига.

В настоящее время существуют различные подходы к реализации алгоритма имитации отжига на параллельных вычислительных системах, рассмотрим 3 из них:

- параллельный запуск алгоритма имитации отжига (мультистарт);
- параллельный запуск алгоритма имитации отжига с обменом результатами (кластерный алгоритм);
- разбиение пространства решений на области [1].

Рассмотрим их подробнее.

### Параллельный запуск алгоритма имитации отжига

Данная адаптация метода имитации отжига предполагает вычисление глобального минимума ЦФ одновременно на нескольких вычислительных узлах с последующим выбором лучшего решения на одном выделенном узле - управляющим.

В выполняемой на нескольких вычислительных узлах при одинаковых начальных условиях осуществляется поиск минимума ЦФ методом имитации отжига по схеме Больцмановского тушения [2].

Затем результаты, полученные на рабочих узлах, передаются в управляющий узел, в котором осуществляется выбор решения с наименьшим значением ЦФ, что должно обеспечивать глобальность найденного минимума.

Реализованный таким образом на параллельной вычислительной системе алгоритм оптимизации методом имитации отжига должен обеспечивать надёжный поиск глобального минимума целевой функции за счёт многократного повторения поиска в заданной явными ограничениями области.

При этом, так как многократный процесс поиска минимума осуществляется одновременно на соответствующем числе вычислительных узлов, то затраченное на выполнения всего параллельного алгоритма время приблизительно равно времени выполнения одного процесса и практически не зависит от числа процессов. Соответственно, при использовании  $p$  рабочих узлов можно уменьшить временные затраты при поиске глобального минимума примерно в  $p$  раз [3].

### Кластерный алгоритм

Кластерный алгоритм является дальнейшим развитием алгоритма, который представлен выше. Основным отличием его является введение в выполняемую на каждом вычислительном узле параллельной системы последовательную часть алгоритма этапа обмена результатами между рабочими узлами.

Изначально на каждом из  $p$  узлов параллельной вычислительной системы запускается алгоритм имитации отжига с разными начальными точками  $x_1, 1!p$ . По выполнении заданного количества итераций рабочие узлы обмениваются текущими результатами, после чего продолжают работу, исходя из лучшего значения результата. После  $t$  обменов рабочие узлы продолжают работу по получению индивидуального

результата. По достижении условия завершения поиска главный узел вычислительной системы выбирает лучшее решение [3].

### Разбиение пространства решений на области

В данном методе пространство решений алгоритма имитации отжига разбивается на несколько областей, в каждой из которых запускается последовательный алгоритм имитации отжига. В этом случае работа главного вычислительного узла, так же как и в упомянутых ранее адаптациях, сводится к выбору наилучшего решения среди полученных на рабочих узлах параллельной вычислительной системы.

Данный подход позволяет строить параллельные алгоритмы с низким трафиком обмена между параллельными процессами [3].

### Описание численного эксперимента и анализ полученных результатов

Оценка надёжности и эффективности распараллеленного алгоритма имитации отжига по схеме Больцмановского тушения осуществлялась на многоэкстремальных ЦФ, с количеством параметров оптимизации  $n$ , равным 2:

- гладкая многоэкстремальная функция:

$$f(x_1, x_2) = \frac{x_1^2 + x_2^2}{200} - \cos(x_1) \cos\left(\frac{x_2}{\sqrt{2}}\right) + 2$$

где  $-0.5 < x_j < 1, j=1,2; \min = -1$ . (1)

- многоэкстремальная функция Растригина:

$$f(x_1, x_2) = 20 + x_1^2 - 10 \cos(2\pi x_1) + x_2^2 - 10 \cos(2\pi x_2),$$

где  $-0.5 < x_j < 1, j=1,2; \min = 0$ . (2)

- негладкая 4-х экстремальная функция:

$$\begin{aligned} f_1 &= -5 * e^{-3 * (|x_1 + 1|^{0.5} + |x_2 + 1|^{0.5})}; \\ f_2 &= -10 * e^{-2 * (|x_1| + |x_2|)}; \\ f_3 &= -7 * e^{-2.5 * (|x_1 - 1|^{0.5} + |x_2 - 1|^{0.5})}; \\ f_4 &= -3 * e^{-(|x_1 - 2|^{0.5} + |x_2 - 2|^{0.5})}; \\ f(x_1, x_2) &= f_1 + f_2 + f_3 + f_4, \end{aligned}$$

где  $-0.5 < x_j < 1, j=1,2; \min = -10.1$ . (3)

- негладкая 4-х экстремальная функция, имеющая три равных по значению глобальных минимума:

$$\begin{aligned} f_1 &= -7 * e^{-3 * (|x_1 + 1|^{0.5} + |x_2 + 1|^{0.5})}; \\ f_2 &= -7 * e^{-2 * (|x_1| + |x_2|)}; \\ f_3 &= -7 * e^{-2.5 * (|x_1 - 1|^{0.5} + |x_2 - 1|^{0.5})}; \\ f_4 &= -3 * e^{-(|x_1 - 2|^{0.5} + |x_2 - 2|^{0.5})}; \\ f(x_1, x_2) &= g_1 + g_2 + g_3 + f_4, \end{aligned}$$

где  $-0.5 < x_j < 1, j=1,2; \min = -7.5$ . (4)

Тестирование проводилось при следующих условиях: начальная температура  $T_0 = 10^9$ , конечная  $T_{end} = 0.1$ , коэффициент изменения температуры  $\tau = 0.97$ . При таких значениях параметров обеспечивалось оптимальное соотношение точности результата и времени вычисления алгоритма.

Результаты поиска минимума ЦФ (1), (2), (3) и (4) тремя распараллеленными методами имитации отжига в зависимости от числа одновременно работающих процессов, усредненные по итогам 100 запусков вычислительной системы, приведены на рисунках 2, 4, 5 и 6 для каждой из функций.

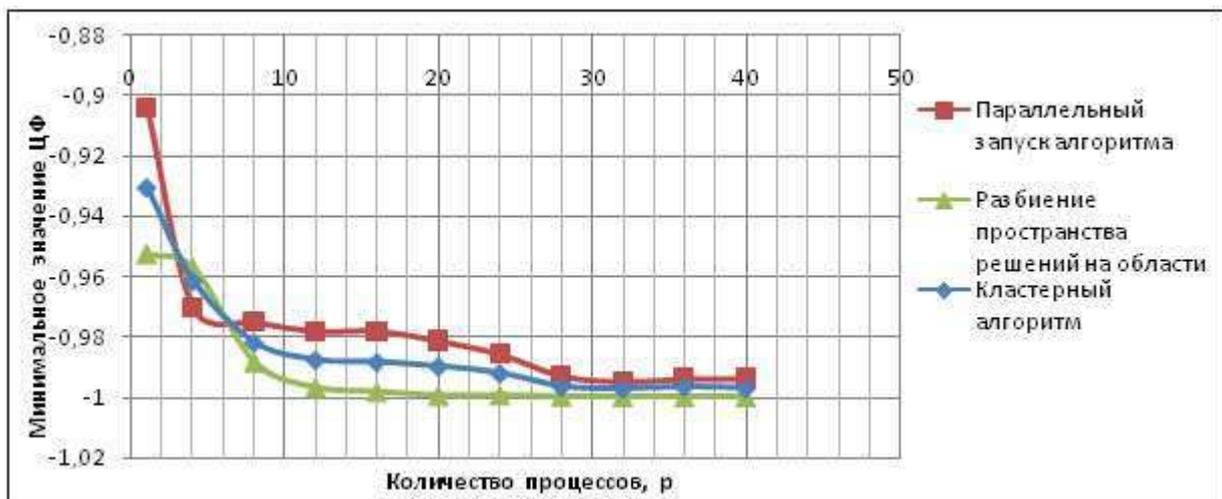


Рис. 2

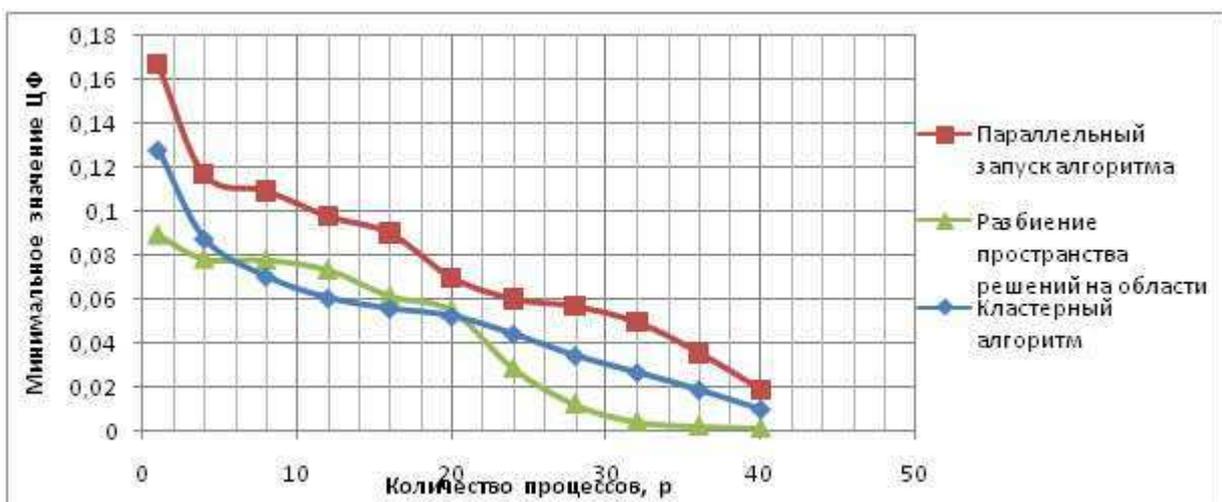


Рис. 3

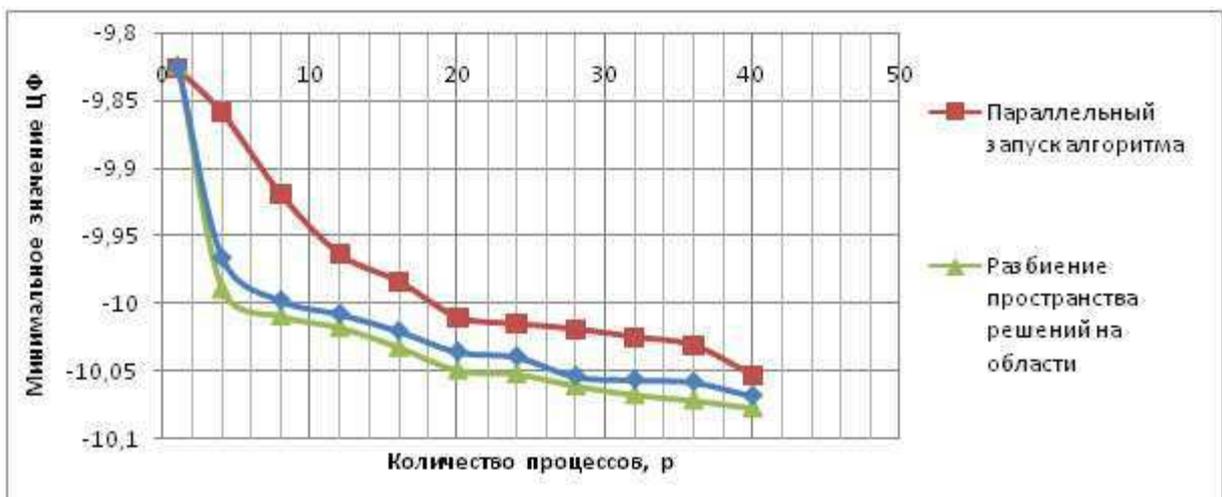


Рис. 4

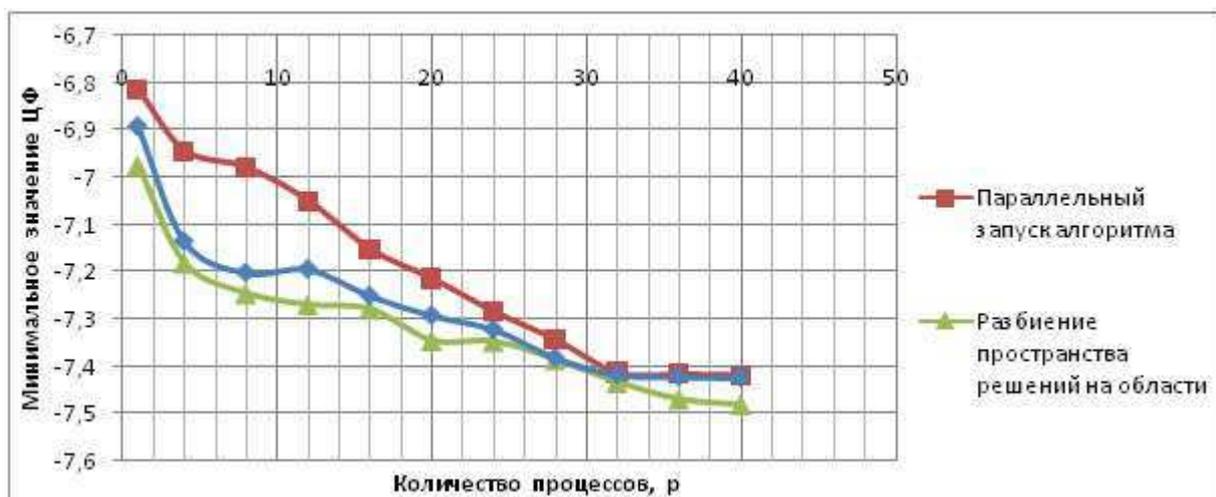


Рис. 5

Как видно из рисунков 2–5, при увеличении количества рабочих процессов  $p$  результаты приближаются к глобальному минимуму.

При этом увеличение числа параллельно работающих процессов с 1 до 20 приводит к уменьшению размеров области глобального минимума, оцениваемых по среднеквадратическому отклонению минимального значения ЦФ, в среднем на 48%, 54% и 60% по сравнению с последовательным алгоритмом для параллельного запуска, кластерного алгоритма и разбиения пространства решений на области соответственно (см. рис. 6).

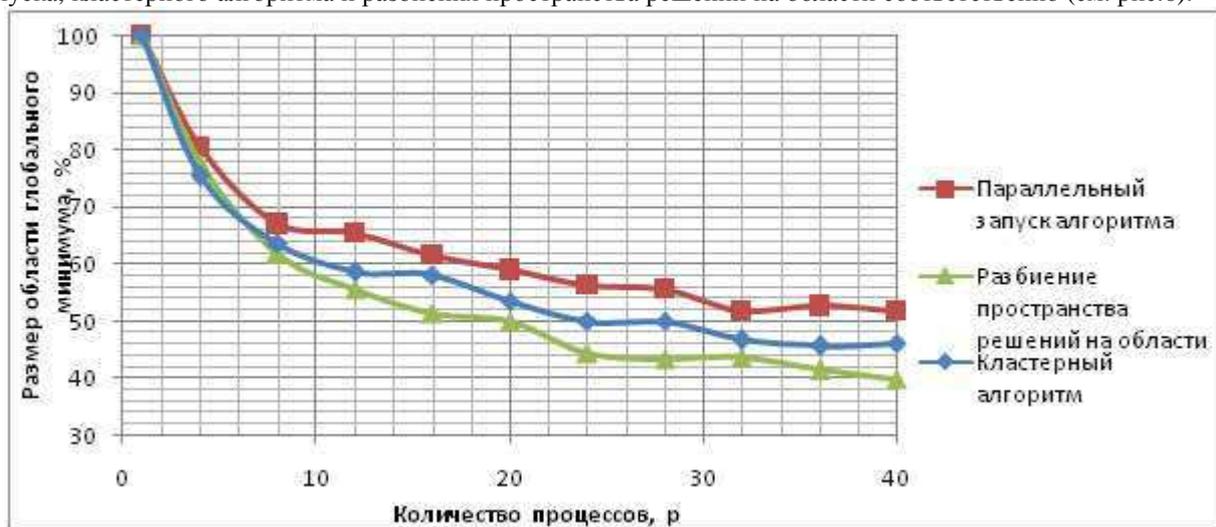


Рис. 6. Зависимость размеров области глобального минимума в процентах от значений размеров, полученных при использовании последовательной версии, от количества одновременно работающих процессов поиска минимума методом имитации отжига по схеме Больцмановского тушения.

При решении реальных оптимизационных задач большое время занимает вычисление ЦФ. Соответственно, эффективность метода оптимизации можно оценивать по количеству итераций алгоритма. Как видно из рисунка 6, в случае параллельного запуска алгоритма на  $p$  вычислительных узлах среднее число итераций алгоритма не будет зависеть от числа одновременно работающих процессов. Данная зависимость характерна и для кластерного алгоритма, так как количество итераций алгоритма задается пользователем и не зависит от числа работающих узлов. Однако с ростом числа узлов увеличивается время выполнения алгоритма. Это связано с задержкой в передаче и приеме данных, осуществляемых через сеть. Число итераций сокращается с увеличением вычислительных узлов в случае разбиения пространства решений на области и имеет экспоненциальный характер.

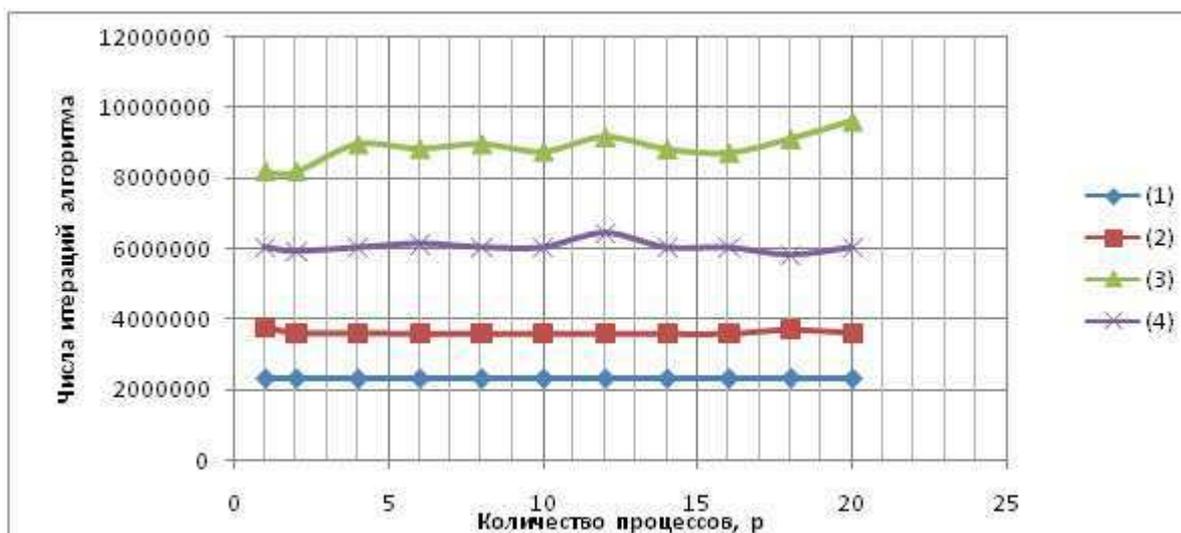


Рис. 7. Число итераций алгоритма поиска минимума ЦФ (1)–(4) методом параллельного запуска

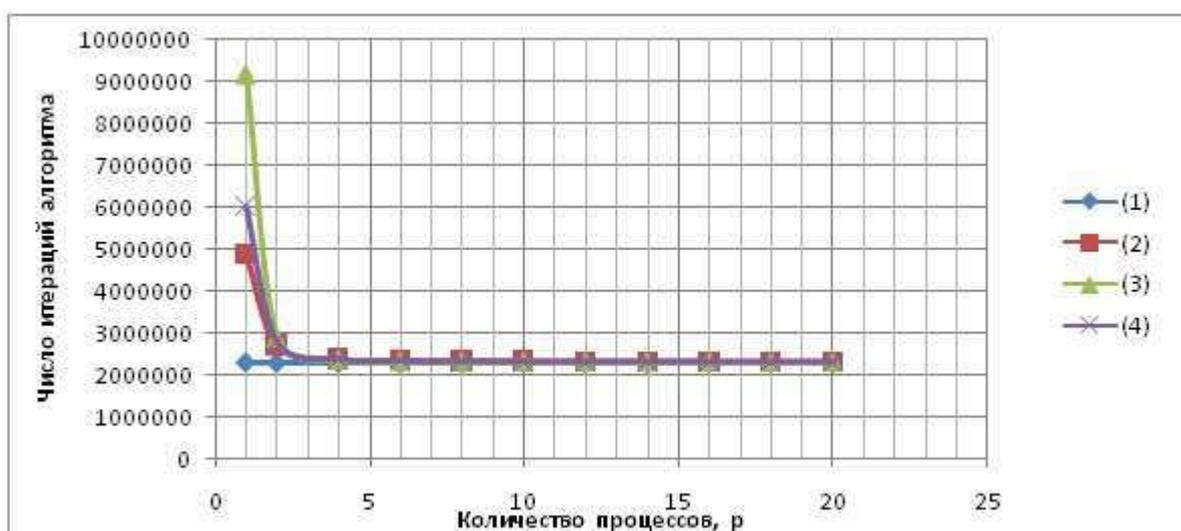


Рис. 8. Число итераций алгоритма поиска минимума ЦФ (1)–(4) методом разбиения пространства решений.

При этом значительное влияние на количество вычислений ЦФ и, соответственно, на эффективность тестируемого метода оптимизации оказывает значение температурного коэффициента  $\tau$ . В ходе проведения численного эксперимента был сделан вывод, что изменение температурного коэффициента  $\tau$  с 0.95 до 0.99 приводит к увеличению времени вычисления ЦФ практически в 6 раз, однако при этом точность нахождения решения повышается.

### Заключение

В ходе проведенной работы была развернута параллельная вычислительная система и проверены ее возможности для решения учебных и научных задач, рассмотрены и реализованы 3 варианта распараллеливания алгоритма оптимизации многоэкстремальной целевой функции двух переменных с явными ограничениями, основанного на стохастическом методе имитации отжига. При этом были получены следующие результаты:

- изучена программная платформа для организации параллельных вычислений MPICH2;
- реализована система параллельных вычислений под управлением программной платформы MPICH2 на базе ресурсов лаборатории Теоретических проблем информатики и ее приложений факультета КНИИТ СГУ им. Н. Г. Чернышевского;
- реализованы 3 способа распараллеливания алгоритма оптимизации;
- проведен сравнительный анализ полученных результатов.

### ЛИТЕРАТУРА:

1. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing. // Science. v220. 1983. P. 671–680.

2. Лопатин А.С. Метод отжига // Межвуз. сб. «Стохастическая оптимизация в информатике». Выпуск 1, 2005. Санкт-Петербург: изд-воСПбГУ. С. 133–149.
3. Janaki Ram D., Sreenivas T.H., Ganapathy Subramaniam K. Parallel Simulated Annealing Algorithms// J. of parallel and distributed computing. 1996. №37, P.207-212.
4. Калашников А.В., Костенко В.А. Параллельный алгоритм имитации отжига для построения многопроцессорных расписаний// Известия РАН. Теория и системы управления, 2008., N.3, С.101–110.