ПАРАЛЛЕЛЬНЫЙ МЕТОД НЕРАВНОМЕРНЫХ ПОКРЫТИЙ С МНОЖЕСТВОМ КАНДИДАТОВ-РЕШЕНИЙ

Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс

На основе метода неравномерных покрытий, предложенного в [1], разработан новый параллельный алгоритм поиска глобального экстремума функций, основанный на популяционном подходе. Этот подход предполагает, что множество кандидатов-решений одной и той же задачи поддерживается в семействе, эволюционирующем таким образом, что гарантируется достаточная диверсификация решений. Обеспечение такого разнообразия достигается благодаря введению меры различия между решениями, которая зависит от специфики решаемой задачи.

Алгоритм реализован на языке С в MPI-системе параллельного программирования с передачей сообщений. Для ускорения расчетов используются вспомогательные процедуры поиска локального экстремума. Эффективность предложенного алгоритма проверена на тестовых задачах. Вычислительные эксперименты, выполненные на многомашинном вычислительном комплексе MVS100K Межведомственного суперкомпьютерного центра Российской академии наук, подтвердили реальную возможность поиска глобальных решений.

Постановка задачи

Рассмотрим задачу отыскания глобального минимума функции f на множестве $X \subset \mathbb{R}^n$:

$$f_* = glob\min_{x \in X} f(x) \qquad (1)$$

Через X_* обозначим множество решений этой задачи. Введем множество ε-оптимальных (приближенных) решений задачи (1): $X_*^\varepsilon = \{x \in X : f(x) \le f_* + \varepsilon\}$.

В большинстве практических задач достаточно найти хотя бы одну точку $x \in X_*^\varepsilon$ и взять в качестве оптимального значения f_* число $f_r = f(x_r)$. Другими словами, надо с заданной точностью ε определить величину глобального минимума функции и найти хотя бы одну точку x_r , где это значение достигается.

Метод неравномерных покрытий

Основная идея метода неравномерных покрытий заключается в следующем: вместо поиска глобального минимума на X ищутся глобальные минимумы на подмножествах, объединение которых содержит X.

При программной реализации метода вводятся дополнительные упрощающие предположения. Считается, что допустимое множество X – это n-мерный параллелепипед с гранями, параллельными координатным плоскостям: $X = \{x \in \mathbb{R}^n, a \le x \le b\}$.

Здесь и ниже неравенство $a \le x$ означает, что $a^j \le x^j$ для всех $1 \le j \le n$.

В процессе расчетов используются вспомогательные векторы $a_i, b_i \in \mathbb{R}^n$ и порождаемые ими прямоугольные параллелепипеды с гранями, параллельными координатным плоскостям: $P_i = \{x \in \mathbb{R}^n : a_i \le x \le b_i\}$.

Будем считать, что все векторы $a_i \ge a$ и $b_i \le b$. Таким образом все параллелепипеды $P_i \subseteq X$. В качестве точек c_i берутся центры параллелепипедов P_i : $c_i^j = (a_i^j + b_i^j)/2, 1 \le j \le n$.

Вектор главной диагонали d_i параллелепипеда P_i имеет компоненты $d_i^j = b_i^j - a_i^j, 1 \le j \le n$. Чебышевская норма вектора d_i определяет длину максимального ребра w_i параллелепипеда P_i : $w_i = \|d_i\|_\infty = \max\limits_{1 \le j \le n} d_i^j$.

Через q_i обозначим значение минимума миноранты функции *f(x)* на P_i . Если *f(x)* удовлетворяет условию Липшица всюду на X с константой l, т.е. для любых x и y из P_i выполнено неравенство: $|f(x) - f(y)| \le l\|x - y\|$, то

$$q_i = \min_{x \in P_i} g(c_i, x) = f(c_i) - \frac{l}{2} \max_{1 \le j \le n} \|d_i^j\| = f(c_i) - \frac{l}{2} w_i \quad \text{, при} \quad y = c_i \ .$$

Если градиент *f(x)* удовлетворяет условию $\|f_x(x) - f_x(y)\| \le L\|x - y\|$ с константой L, то q_i определяется выражением $q_i = f(c_i) + \langle f_x(c_i), \hat{x} - c_i \rangle - \frac{L}{2}\|\hat{x} - c_i\|^2$, где $\hat{x}$ находится из решения задачи квадратичного программирования:

$$\hat{x} \in arg\min_{x \in P_i} g_1(x, c_i) = arg\min_{x \in P_i} [\langle f_x(c_i), x - c_i \rangle - \frac{L}{2} \|x - c_i\|^2] \quad (2).$$

Программная реализация метода неравномерных покрытий выполняется однотипным образом для разных липшицевых функций. По-разному вычисляются только оценки минимума q_i и минимальные радиусы покрывающих шаров ξ. Алгоритм покрытия множества X параллелепипедами P_i остается один и тот же.

Если $q_i \geq R_m - \varepsilon$, то параллелепипед P_i может быть исключен, так как глобальный минимум на нем не дает улучшения текущего рекорда R_m более чем на ε и поиск продолжается на множестве $X \setminus P_i$. В противном случае, если $q_i < R_m - \varepsilon$, то параллелепипед P_i делится пополам по максимальному ребру и поиск ведется на этих двух частях. Если главные диагонали одной или обеих частей меньше, чем ξ, то на них $f(x) \geq q_i = f(c_i) - \varepsilon$ и эти части исключаются из области поиска. В центрах оставшихся частей вычисляются значения *f(x)*. При этом, если возможно, улучшается рекорд и проверяется условие покрытия обеих частей. Те части, которые оказались покрытыми, исключаются из области поиска. Если обе части исключены, то P_i также исключается, если нет, то продолжается дальнейшее деление до тех пор, пока параллелепипед P_i не будет покрыт.

Параллельный алгоритм метода неравномерных покрытий

Главная идея параллельного алгоритма состоит в том, чтобы выполнять итерации последовательного метода [3, 4] параллельно на множестве процессоров с периодическим обменом рекордами и перераспределением областей поиска между ними в процессе счета.

Пусть имеется некоторая начальная последовательность $B_m = \{P_1, \ldots, P_m\}$ параллелепипедов P_i, принадлежащих X, например, построенная в процессе работы последовательного алгоритма. Пусть $N_m = \{c_1, \ldots, c_m\}$ – последовательность центров этих параллелепипедов, вычислены текущий рекорд R_m и текущая рекордная точка x_r.

Помимо $p = m$ рабочих процессоров имеется еще один процессор, называемый диспетчером. Как и ранее, распределим параллелепипеды начального набора B_m между рабочими процессорами. При выполнении итераций последовательного алгоритма каждый рабочий процессор поддерживает свой индивидуальный пул. Будем говорить, что наступает завершение работы процессора, если им выполнены Q циклов последовательного алгоритма или если его индивидуальный пул стал пустым. Процессор, который завершил работу, сообщает диспетчеру следующие величины: число параллелепипедов в индивидуальном пуле, минимальную оценку y для каждого из них и свой индивидуальный рекорд (т.е. лучшее значение функции f, полученное в ходе вычислений). Этот процессор ставится в поддерживаемую диспетчером очередь ожидающих процессоров. Ожидающий процессор назовем свободным, если его индивидуальный пул пуст.

В предлагаемом варианте алгоритма каждый процессор посылает диспетчеру данные о ходе решения в асинхронном режиме, т.е. независимо от других процессоров. На основе полученных данных диспетчер улучшает рекорд R и сообщает его всем ожидающим процессорам, которые будут использовать новый рекорд для отсева параллелепипедов из своих наборов. Диспетчер выявляет среди ожидающих процессоров те, которые содержат только один параллелепипед в своем индивидуальном наборе, и приказывает им выполнить цикл последовательного алгоритма Q раз, взяв новый R в качестве рекорда. Точно такой же приказ получают другие ожидающие процессоры, если в данный момент отсутствуют свободные.

Как только диспетчер выявит свободный процессор, он среди ожидающих подберет ему партнера для передачи параллелепипеда с минимальной нижней оценкой. Подбор такой пары завершается двумя приказами: свободному процессору дается приказ принять параллелепипед от ожидающего процессора, а ожидающему – передать параллелепипед данному свободному.

Диспетчер завершает выполнение алгоритма, если все рабочие процессоры свободны. После окончания расчетов множество X будет полностью покрыто параллелепипедами P_i, а рекордная точка x_r будет ε-решением задачи (1).

Параллельный алгоритм с множеством кандидатов-решений

Сложность решения многоэкстремальных задач глобальной оптимизации зависит не только от количества локальных оптимумом функции, но и от их взаимного расположения. Ключевым понятием для оценки (измерения) трудности решения задач глобальной оптимизации является «воронка» [2], впервые использованным в задачах, возникающих в химии и биологии. «Воронка» – это множество локальных минимумов таких, что для каждого из них существует, по крайней мере, одна убывающая последовательность, ведущая к минимуму, отвечающему «дну» воронки. Путь, соединяющий два соседних локальных минимума последовательности, таков, что вдоль него целевая функция не превышает заданного значения. Число воронок с

учетом их ширины может быть более подходящей характеристикой сложности задачи поиска глобального экстремума.

Если поведение функции характеризуется большим числом узких воронок с крутыми склонами, то процесс поиска может так и не найти глобальный минимум, «застревая» надолго в очередной воронке. Для преодоления этой трудности было сделано предположение, что если начинать покрытие не с одного параллелепипеда, а с их набора (семейства), которым отвечают локальные минимумы различных воронок, то потребуется намного меньше циклов, чтобы найти глобальный оптимум.

Для реализации этой идеи предложен алгоритм, основанный на популяционном подходе, который предполагает, что множество кандидатов-решений одной и той же задачи поддерживается в семействе, эволюционирующем таким образом, что гарантируется достаточная диверсификация решений. Обеспечение такого разнообразия достигается благодаря введению меры различия между решениями, которая зависит от специфики решаемой задачи.

В идеале мера различия d должна быть такова, чтобы $d(X,X)\equiv 0$, а если точки X и Y принадлежат одной воронке (т.е. траектория движения из X и Y приводит к дну одной и той же воронки), то $d(X,Y)\le d_0$, где d_0 – пороговое значение.

На базе популяционного подхода можно предложить следующий алгоритм:

- Инициализация: имеется некое число процессоров *K*, задаем начальное множество точек *S* (семейство кандидатов-решений), константы Липшица, максимальное число *M* членов семейства, пороговое значение d_0 и длину ребра многомерного куба, определяющего область поиска вокруг каждого кандидата-решения. Каждый рабочий процессор с номером $j\ (j=1,\ldots,M)$ берет *j*–го члена семейства в качестве стартовой точки и строит вокруг него начальный параллелепипед (куб) с заданной длиной ребра. Оставшиеся *(K - M)* процессоров являются свободными и находятся в режиме ожидания.
- Рабочие процессоры выполняют итерации параллельного алгоритма метода неравномерных покрытий, при этом каждый процессор ведет свое семейство кандидатов-решений. Любая найденная новая точка Y становится кандидатом на включение в семейство. В качестве новых точек служат центры параллелепипедов, полученные в процессе половинных делений, и точки, найденные в результате локального поиска экстремума. Для обновления семейства используется следующий алгоритм:

  а) Среди членов семейства *S* ищется точка X_q, наиболее «близкая» к новой точке Y в соответствии с выбранной мерой различия.

  б) Если $d(X_q,Y)\le d_0$ и при этом $f(X_q)>f(Y)$, то точку X_q заменяем на Y.

  в) Если $d>d_0$ и число членов семейства меньше *M*, то точка Y включается в семейство в качестве нового члена.

  г) Если $d>d_0$ и число членов семейства равно *M*, то в семействе ищется точка X_p с наихудшим значением функции *f*. Если $f(X_p)>f(Y)$, то точку X_p заменяем на точку Y.

  Если выполнен критерий останова параллельного алгоритма метода неравномерных покрытий, то переходим к шагу 3, а иначе продолжаем итерации параллельного алгоритма.
- При завершении работы параллельного алгоритма метода неравномерного покрытий, все рабочие процессоры, получив приказ «finish» (см. шаг 6 алгоритма для рабочего процессора [4]), передают процессору-диспетчеру свои семейства кандидатов-решений. Диспетчер объединяет их в единое семейство по алгоритму, описанному на шаге 2.

Полученное семейство кандидатов-решений можно использовать на следующем этапе решения задачи в качестве стартового. В случае проведения многоэтапных расчетов, каждый процессор вычислительного кластера в конце очередного этапа сохраняет всю совокупность имеющихся у него параллелепипедов, подлежащих исследованию («деревья»), при этом сохраняется общий для всех процессоров рекорд, рекордная точка и семейство кандидатов-решений. Тогда на следующем этапе работы процессор-диспетчер сможет распределять между свободными процессорами эти деревья, а также точки семейства.

Опыт расчетов показал, что часто оказывается эффективным новый запуск расчетов с полученного семейства, но с измененными параметрами, такими как d_0, максимальное число членов семейства *M* и длина ребра многомерного куба.

При этом на разных этапах, в зависимости от объема работы, может быть заказано различное число процессоров на разных территориально-разнесенных вычислительных кластерах. Такой режим работы позволяет проводить расчеты сколь угодно большой длительности до достижения искомого результата – нахождения глобального экстремума.

Эффективность метода по результатам численных экспериментов

В качестве тестовой была взята задача определения структуры сложных биомолекул (атомных кластеров). Решение такой задачи имеет большое значение, поскольку свойства биомолекул напрямую зависят от их трехмерной конформации (пространственной формы). Считается, что наиболее устойчивая структура биомолекулы отвечает глобальному минимуму фунции свободной энергии молекулы. Учитывая сложность создания энергетической модели, а также множество факторов, влияющих на реальную структуру молекулы, становится очевидной важность методов, позволяющих находить как глобальный минимум, так и другие локальные минимумы таких функций, близкие к глобальномум минимуму.

Задача глобальной оптимизации функции энергии F представляет огромные сложности. И главной их причиной является не только, и не столько, число переменных, сколько огромное число локальных минимумов функции F.

В задаче определения структуры атомного кластера мы имеем n частиц (атомов или молекул), и функцию энергии F, которая зависит от взаимного расположения этих частиц в трехмерном пространстве. Требуется найти глобальный минимум функции энергии. Для численных экспериментов использовалась функция Morse, определяющая энергию атомного кластера, состоящего из n атомов:

$$F(\rho)=\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\left(e^{\rho\left(1-\|X_i-X_j\|\right)}-1\right)^2-1\right)$$, где ρ – скалярный параметр, X_i и X_j – трехмерные векторы координат центров атомов i и j, соответственно. Задача состояла в определении координат n атомов таким образом, чтобы при фиксированном значении параметра ρ значение функции $F(\rho)$ достигало глобального минимума. В расчетах ρ принималось равным 3, 6, 10 и 14. Общее число переменных достигало 255.

Поведение этой функции характеризуется большим числом воронок, причем с ростом значения ρ их число резко возрастает, а сами они становятся уже и глубже, так что траектория поиска постоянно в них «сваливается».

Были проведены вычислительные эксперименты для сравнения эффективности двух параллельных алгоритмов. В первом из них запоминалась лишь одна наилучшая точка, во втором использовалось множество кандидатов-решений. Определялась оптимальная структура кластера, содержащего 85 атомов (255 переменных) при ρ = 6. Расчеты проводились на 48 процессорах в несколько этапов, каждый этап по 35 минут, число членов семейства для второго алгоритма было равно 7.

На рис. 1 показан сравнительный график поиска оптимума для этих двух алгоритмов. В обоих случаях был найден предполагаемый глобальный оптимум (*-405,25*), но из графика видно, что второй алгоритм с семейством кандидатов-решений нашел его в два раза быстрее. Маркерами обозначена точка предположительного глобального минимума.

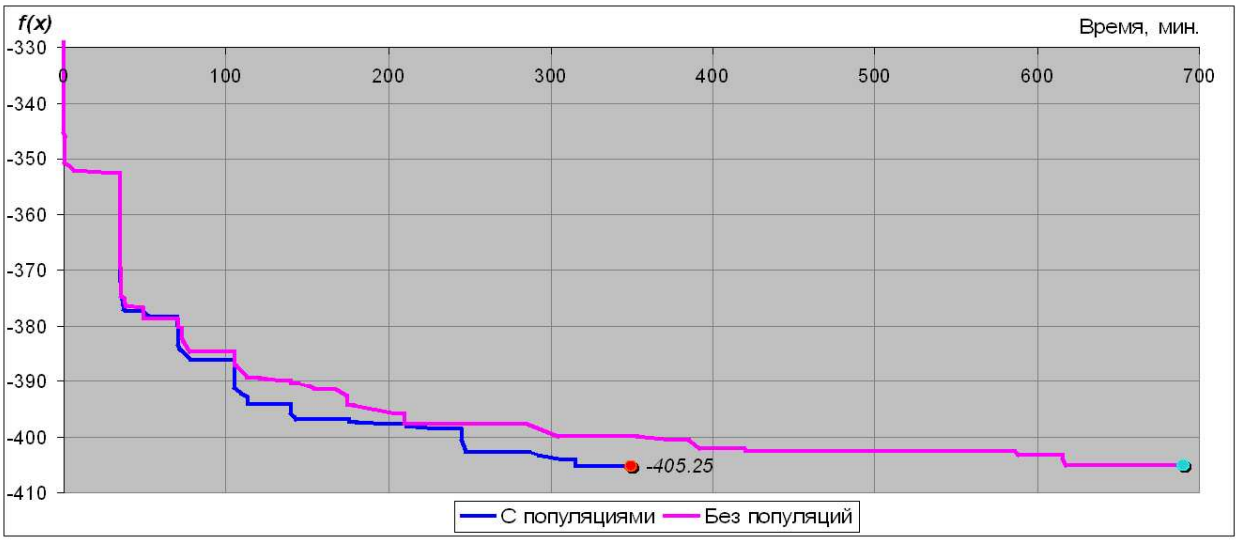

Рис. 1. Сравнение эффективности двух алгоритмов: с популяциями и без популяций

Причем поведение второго алгоритма существенно зависит от выбранной меры различия между членами семейства. Для функции Morse в качестве такой меры было предложено взять набор расстояний от атомов до центра тяжести атомного кластера.

Для исследования этой зависимости была взята функция, определяющая структуру кластера из 85 атомов, у которых все координаты, кроме двух, были фиксированы. Зависимость минимизируемой функции от искомых двух координат приведена на рис. 2: слева – график функции, а справа – график той же функции, но ее значения ограничены сверху. Здесь ясно видно "плато", окруженное "рвом", а предполагаемый глобальный минимум лежит где-то на плато (обозначен маркером).

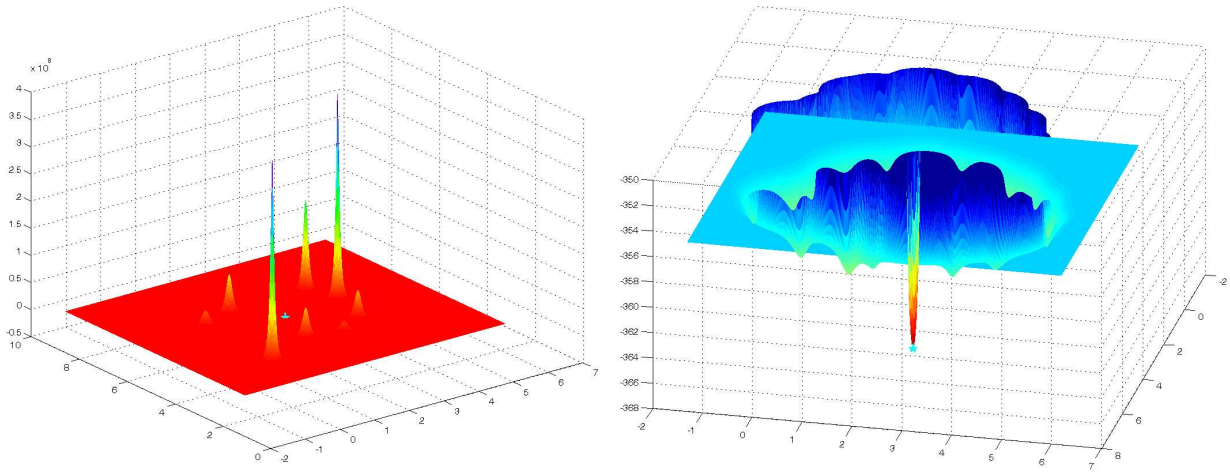

Рис. 2. Зависимость функции от переменных X(153) и X(154)

На рис. 3 показаны семейства, полученные после очередного этапа, при различных значениях d_0 . При $d_0 = 0.02$ почти все точки семейства сосредоточились вокруг точки предполагаемого глобального оптимума, при $d_0 = 4.0$ они оказались разбросанными за пределами "плато", а при $d_0 = 0.2$ – расположились вдоль границы "плато". Очевидно, что от их расположения будет зависеть дальнейший процесс поиска решения. Следовательно, эффективность алгоритма напрямую зависит от выбора значения d_0 .

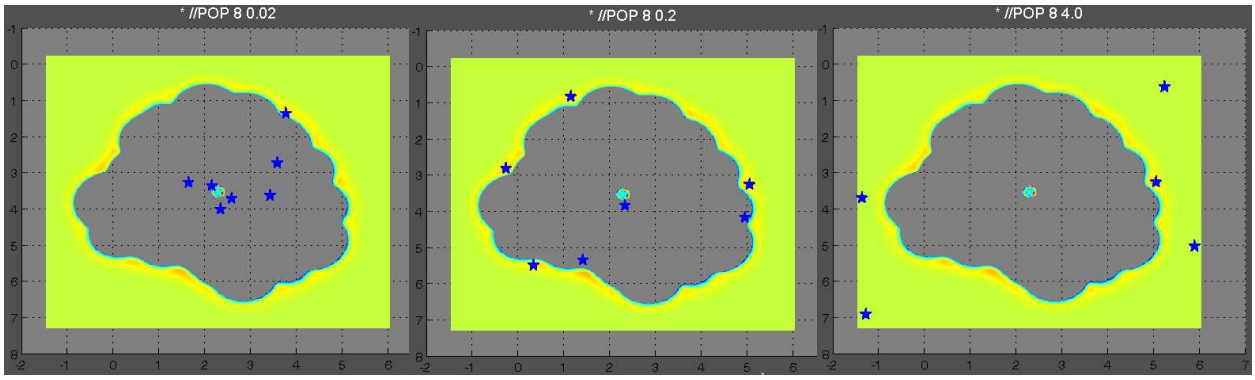

Рис. 3. Семейства кандидатов-решений при различных значениях меры различия

Также была исследована производительность каждого процессора при решении тестовой задачи. При 208 процессорах суммарный объем выполненной работы составил 4 716 839 параллелепипедов, счет не был закончен за 5 часов, но загруженность процессоров была относительно равномерной (см. рис. 4).

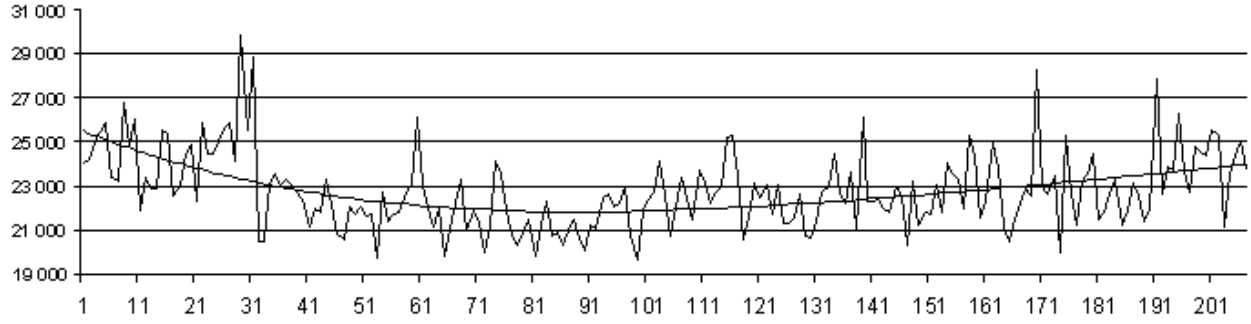

Рис. 4. Загруженность 208 процессоров

При 408 процессорах суммарный объем работы составил 10 486 087 параллелепипедов, счет был закончен за 4 часа 45 минут, но загруженность процессоров варьировалась сильнее (см. рис. 5).

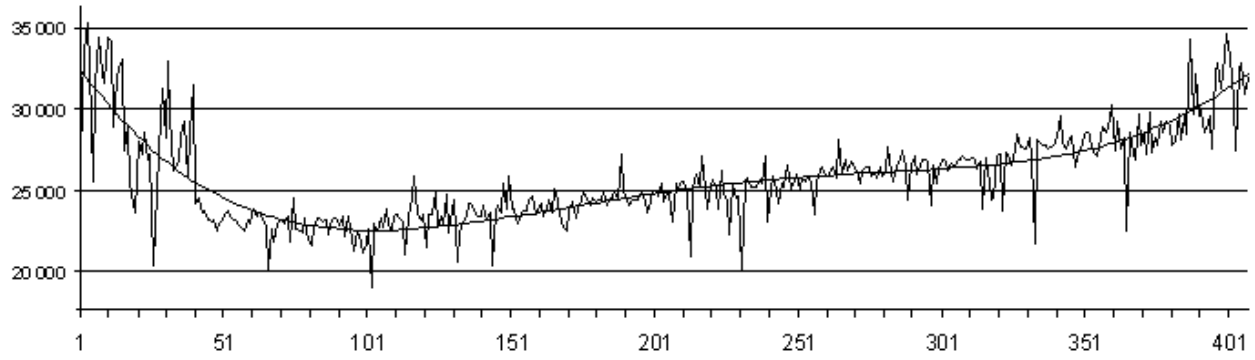

Рис. 5. Загруженность 408 процессоров

При 808 процессорах суммарный объем работы составил 10 486 487 параллелепипедов, счет был закончен за 3 часа 18 минут, но колебания загруженности снизились (см. рис. 6); выполненный каждым процессором объем работы уменьшился примерно вдвое, а время – в полтора раза.

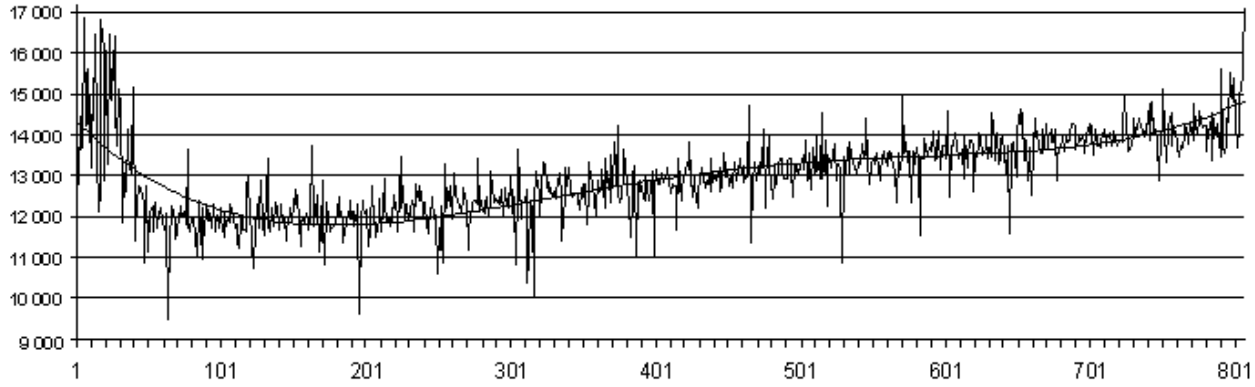

Рис. 6. Загруженность 808 процессоров

Полученная структура кластера из 85 атомов при ρ = 6 приведена на рисунке 7, значение потенциальной энергии составляет -405,25.

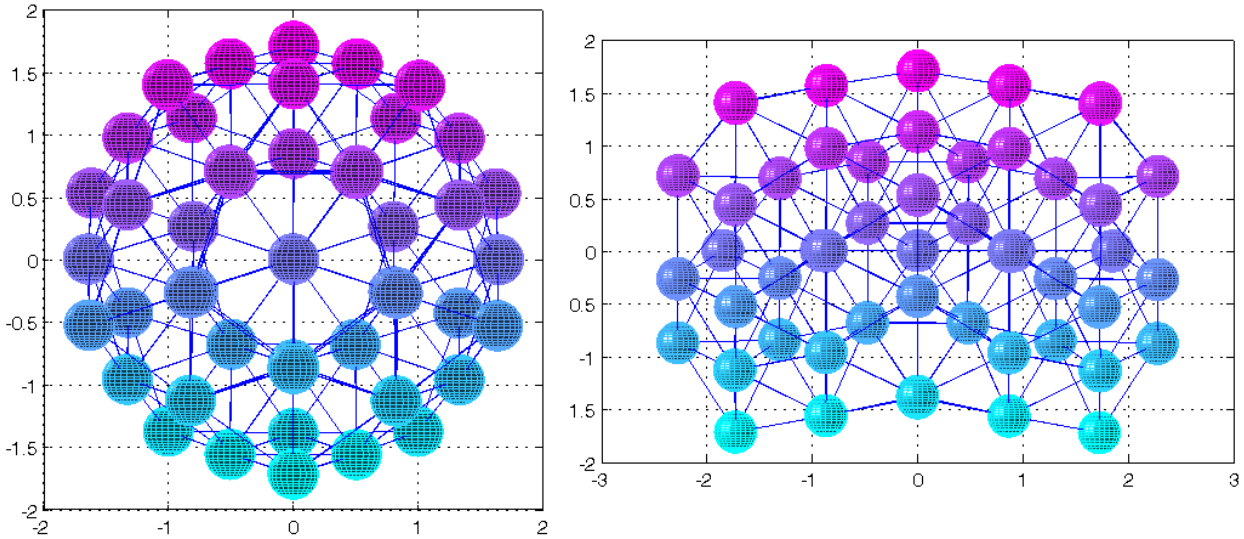

Рис. 7. Структура атомного кластера (85 атомов, ρ = 6) в двух проекциях

Разработанный параллельный метод неравномерных покрытий позволяет проводить глобальную оптимизацию липшицевых функций большой размерности.

Для анализа полученных результатов и хода решения была разработана библиотека программ визуализации средствами пакета MATLAB.

ЛИТЕРАТУРА:

1. Ю.Г. Евтушенко. "Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке)" // М.: ЖВМ и МФ, 1971. 11, №6, с.1390-1403.
2. A. Grosso., M. Locatelli, F. Schoen. "A population-based approach for hard global optimization problems based on dissimilarity measures" //Math. Program., Ser. A (2007), pp. 373–404.

3. Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс. "Распараллеливание процесса поиска глобального экстремума" //Автоматика и телемеханика, №5, 2007, с. 46- 58.
4. Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс. "Параллельный поиск глобального экстремума функции многих переменных" //Ж. вычисл. матем. и матем. физ., 2009, том 49, №2, с. 255 - 296.
5. Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс. "Два подхода к поиску глобального минимума межатомных потенциалов". Материалы Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ", Новороссийск, 22-27 сентября 2008 г., Изд-во Московского Университета, с.65-68.
6. Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс "Параллельный поиск глобального экстремума функций на многоядерных вычислительных комплексах". Материалы Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ", Новороссийск, 24-29 сентября 2007 г., Изд-во Московского Университета, с.127-129.