

# Параллельный эвристический алгоритм глобальной оптимизации

М. А. Посыпкин

*Институт системного анализа Российской академии наук  
(ИСА РАН)*

В работе предложены последовательный и параллельный алгоритмы систематического обхода пространства решений для решения задачи глобальной оптимизации функции многих переменных. Экспериментально показано, что параллельный вариант предложенного алгоритма позволяет находить решения наиболее сложных постановок для задачи поиска оптимальной конфигурации атомов, соответствующих потенциалу Морса при параметре  $\rho = 14$  при наличии достаточных вычислительных ресурсов. Основным достоинством алгоритма по сравнению с другими известными алгоритмами является то, что он не использует специфичной для задачи оптимизации молекулярных кластеров геометрической информации.

## Введение

Методы глобальной оптимизации можно условно разделить на две категории: с доказательством и без доказательства оптимальности. Методы с доказательством оптимальности позволяют получить решение, отличающееся от оптимума не более чем на заданную величину. К этому классу можно отнести различные варианты метода ветвей и границ, методы, основанные на идеях интервального анализа и многие другие. Основным недостатком методов с доказательством оптимальности является их высокая вычислительная сложность. Поэтому для многих прикладных и модельных задач применение точных методов

позволяет получать решения только для очень небольших размерностей, представляющих незначительный интерес с практической точки зрения. Для решения таких задач часто применяют алгоритмы, позволяющие найти минимум приблизительно с неизвестной или очень грубо оцениваемой точностью. Такие алгоритмы принято называть эвристическими. Известно, что хотя оптимальность полученных решений не доказывается, эвристики, как правило, позволяют находить оптимум или очень хорошие приближения к нему.

В настоящей работе рассматривается задача поиска глобального минимума функции многих переменных, имеющей большое число локальных экстремумов. Известно, что эта задача требует больших вычислительных ресурсов, при этом, мощностей обычных рабочих станций часто оказывается недостаточно. Поэтому для ее решения представляется целесообразным применение методов параллельных и распределенных вычислений. В настоящей работе предлагается параллельный вариант эвристического алгоритма глобальной оптимизации, основанного на известном подходе монотонного случайного спуска (Monotonic Basin-Hopping) [1]. Показано, что при наличии достаточных вычислительных ресурсов этот алгоритм позволяет находить решение сложных задач глобальной оптимизации, в частности задачи поиска оптимальной конфигурации атомного кластера. С его помощью были посчитаны все наиболее сложные минимальные конфигурации атомных кластеров для потенциалов Морса и Леннарда—Джонса, что не удавалось ранее сделать без применения специальных алгоритмов, учитывающих геометрическую природу задачи.

## Постановка задачи и основные определения

Для описания предлагаемого подхода потребуются следующие определения.

**Определение 1.** Задана непрерывная функция  $n$  переменных  $f: R^n \rightarrow R$  и параллелепипед  $P = [a^1, b^1] \times \dots \times [a^n, b^n]$ . Точка  $x_*$ , такая что  $f(x_*) \leq f(x)$  для всех  $x \in P$  называется *точкой глобального минимума* функции  $f$  на параллелепипеде  $P$ .

**Определение 2.** Пусть задана норма  $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$  в пространстве  $\mathbb{R}^n$ . Окрестностью точки  $x_0 \in \mathbb{R}^n$  радиуса  $r$  называется следующее множество точек  $U(x_0, r) = \{x \in \mathbb{R}^n, \|x - x_0\| \leq r\}$ .

Далее в работе всегда используется норма  $\|x\|_\infty = \max_{i=1,n} |x_i|$ .

**Определение 3.** Точка  $x_0$  называется *точкой локального минимума*, если существует окрестность  $U(x_0, r)$  точки  $x_0$ , такая что  $f(x_0) \leq f(x)$  для всех  $x \in U(x_0, r)$ .

Данная работа посвящена задаче *глобальной минимизации*, состоящей в нахождении хотя бы одной точки глобального минимума и значения функции в этой точке. Обсуждаемые в статье способы решения этой задачи предполагают наличие алгоритма *локальной минимизации* и *оператора возмущения*. Будем следовать обозначениям, введенным в [7].

Алгоритм локальной минимизации начинает поиск с заданной точки  $x$ , находит близкую к ней точку локального минимума  $x_0$ . Примерами такого рода алгоритмов являются методы сопряженных градиентов и методы Ньютона. Для дальнейшего изложения удобно рассматривать алгоритм локальной оптимизации как отображение  $L_f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , ставящее в соответствие точке  $x$  точку  $x_0$ .

Оператор возмущения  $\Phi$  ставит в соответствие точке  $x$  точку  $\Phi(x)$ . Обычно предполагается, что где  $\Phi(x)$  принадлежит окрестности заданного радиуса  $r$  точки  $x: \Phi(x) \in U(x, r)$ . В данной работе мы ограничимся рассмотрением оператора случайного возмущения, который по заданной точке  $x$  генерирует новую точку  $\tilde{x} \in U(x, r)$ . При этом, компонент  $i$  вектора  $\tilde{x}$  вычисляется по следующему правилу:  $\tilde{x}^i = x^i + \xi$ , где  $\xi$  — равномерно распределенная случайная величина в интервале  $[-r, r]$ .

Для исследования сходимости рассматриваемых алгоритмов потребуются следующие понятия, введенные в [1].

**Определение 4.** Пусть задан алгоритм локальной оптимизации  $L_f: \mathbb{R}^n \rightarrow \mathbb{R}^n$  и радиус окрестности  $r$ . Конечная последовательность

локальных минимумов  $x_1, \dots, x_m$  функции  $f$  называется  $r$ -терминальной, если выполнены следующие три свойства:

- $x_i = L_f(\tilde{x}_{i-1})$ , где  $\tilde{x}_{i-1} \in U(x_{i-1}, r)$  для  $i = 2, \dots, m$  ;
- $f(x_i) < f(x_{i-1})$  для  $i = 2, \dots, m$  ;
- $f(L_f(x)) \geq f(x_m)$  для любой точки  $x \in U(x_m, r)$ .

**Определение 5.** Пусть задана точка  $x \in R^n$  локального минимума функции  $f$  и радиус окрестности  $r$ . Множество точек  $Fu_r(x) \subseteq R^n$  называется  $r$ -воронкой (*funnel*) с центром в  $x \in R^n$ , если для любой точки  $y \in Fu_r(x)$  любая  $r$ -терминальная последовательность, начинающаяся в  $y$ , заканчивается в  $x$ .

**Определение 6.** Пусть задана точка  $x \in R^n$  локального минимума функции  $f$  и радиус окрестности  $r$ . Множество точек  $Bas_r(x) \subseteq R^n$  называется  $r$ -бассейном (*basin*) с центром в  $x \in R^n$ , если для любой точки  $y \in Bas_r(x)$  существует  $r$ -терминальная последовательность, начинающаяся в  $y$ , которая заканчивается в  $x$ .

Очевидно, выполняется соотношение:  $Fu_r(x) \subseteq Bas_r(x)$ .

## Алгоритм систематического обхода пространства решений

Предлагаемый алгоритм систематического обхода SMBH основан на идеях алгоритма Monotonic Basin Hoping (МВН), рассмотренного в [1]. Алгоритм МВН работает по следующей схеме (рис. 1). В качестве начальной выбирается точка  $x_0$  локального минимума функции. На каждом шаге алгоритма текущая точка локального минимума  $x$  подвергается возмущению  $\Phi$ , к результату которого применяется локальный алгоритм. Пусть  $\tilde{x} = L_f(\Phi(x))$ . Если  $f(\tilde{x}) < f(x)$ , то точка  $x$  заменяется на  $\tilde{x}$ . Далее процесс повторяется. Алгоритм останавливается, когда после заданного числа  $N_{\max}$  итераций не удастся улучшить найденный локальный минимум.

Основными параметрами алгоритма МВН являются: радиус окрестности просмотра  $r$  и количество  $N_{\max}$  итераций при генерации случайных точек. Радиус окрестности просмотра  $r$  должен быть достаточно небольшим, чтобы допускать эффективный перебор с помощью оператора возмущения в окрестности радиуса  $r$ . Эти параметры подбираются экспериментально.

## Алгоритм МВН

**Входной параметр:** точка  $x_0$

**Выходной параметр:** точка  $\hat{x}$

- А. Поместить в список точку  $x_0$ .
- В. Повторять в цикле до выполнения условия останова (число итераций  $< N_{\max}$ ):
  1. сгенерировать точку  $\Phi(x)$ ;
  2. применив к точке  $\Phi(x)$  алгоритм локальной минимизации, получить точку  $\tilde{x} = L_f(\Phi(x))$ ;
  3. если  $f(\tilde{x}) < f(x)$ , то  $x = \tilde{x}$ .
- С. Вернуть точку  $x$ .

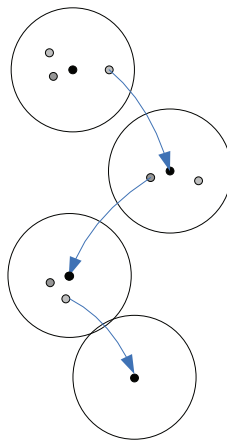


Рис. 1

Алгоритм МВН работает эффективно, если вероятность попасть в воронку, содержащую глобальный минимум высока. В работе [1] показывается, что с ростом числа атомов в кластерах Леннарда—Джонса, число воронок начинает очень быстро возрастать, а вероятность попадания в воронку с центром в глобальном минимуме — снижаться. Поэтому требуется запускать алгоритм МВН очень большое число раз для того, чтобы найти глобальный минимум функции.

Недостатком алгоритма МВН является то, что в результате его работы строится только одна последовательность точек, ведущая к точке локального минимума, рискуя при этом уйти в сторону от глобального оптимума. Предлагаемый ниже алгоритм систематического просмотра пространства (SMBH) решений строит несколько таких путей. Таким образом, алгоритм SMBH исследует несколько бассейнов, к которым принадлежит начальная точка.

Алгоритм начинает работу с некоторой точки локального минимума  $x_0$  в пространстве  $R^n$ . Формируется список точек, состоящий на первом шаге только из одного элемента — точки  $x_0$ , которая является локальным минимумом. На очередном шаге из списка выбирается точка  $x$  с наименьшим среди элементов списка значением целевой функции. Далее заданное число  $N$  раз выполняется следующая последовательность действий: точка  $x$  подвергается возмущению и последующей локальной оптимизации, в результате чего получается точка  $\tilde{x} = L_f(\Phi(x))$ . Если удалось улучшить значение целевой функции, то есть если  $f(\tilde{x}) < f(x)$ , то точка  $\tilde{x}$  добавляется в список. Выбор точек из списка производится до момента выполнения условия останова, которое может быть задано различными способами. Выбранные точки помечаются, для того чтобы избежать их повторного обхода.

Алгоритм систематического обхода просматривает окрестность радиуса  $r$  точек из списка и запоминает точки, в которых удастся улучшить решения с помощью локального алгоритма. Процесс построения решений можно рассматривать как построение древовидной системы окрестностей заданного радиуса с корнем в точке  $x_0$  (рис. 2).

Алгоритм SMBH отличается от МВН в двух аспектах. Во-первых, в отличие от МВН, при работе SMBH строится несколько различных терминальных последовательностей, исходящих из начальной точки. Если начальная точка принадлежит бассейну с центром, содержащим глобальный минимум, то при правильном подборе параметров SMBH построит терминальную цепочку, ведущую в этот минимум.

Во-вторых, если МВН переходит к следующей точке как только удастся улучшить значение целевой функции, то SMBH сначала выполняет заданное число итераций в окрестности, а затем переходит к исследованию точки, в которой достигается минимальное значение. Эксперименты показывают, что в ряде случаев, такой подход позволяет быстрее выявить терминальную последовательность, ведущую к точке глобального минимума.

## Алгоритм SMBH (систематического обхода)

**Входной параметр:** точка  $x_0$

**Выходной параметр:** точка  $\hat{x}$

- A. Поместить в список точку  $x_0$ .
- B. Повторять в цикле до выполнения условия останова:
  1. Выбрать из числа непомеченных точек точку  $x$  с минимальным значением целевой функции, пометить эту точку в списке.
  2. Повторять  $N$  раз:
    - a. сгенерировать точку  $\tilde{x} = L_f(\Phi(x))$ ;
    - b. если  $f(\tilde{x}) < f(x)$ , то добавить точку  $\tilde{x}$  в список.
- C. Вернуть точку  $\hat{x}$  из списка с минимальным значением целевой функции.

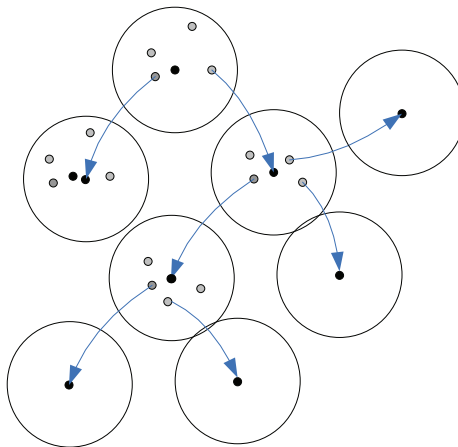


Рис. 2

На практике оказывается, что если не вводить дополнительных ограничений, то количество локальных минимумов в списке будет расти неприемлемо быстро. При этом в списке окажется много почти одинаковых локальных минимумов. Это связано с тем, что различные терминальные последовательности могут пересекаться (иметь общие элементы). Чтобы избежать повторного исследования одних и тех же областей не следует добавлять в список точки, близкие к тем, которые уже исследовались. Для этого вводится *мера близости* — функция  $\sigma: R \times R \rightarrow \{\text{true}, \text{false}\}$ , принимающая значение true, если точки близки и false в противном случае. Новая точка  $\tilde{x}$  добавляется в список, только если в нем нет близких к ней элементов. В результате алгоритм добавления локального минимума в список модифицируется следующим образом:

Сгенерировать точку  $\tilde{x} = L_r(\Phi(x))$ .

Если  $f(\tilde{x}) < f(x)$ , то

- 1) найти такой элемент списка  $x'$ , что  $\sigma(x', \tilde{x}) = \text{true}$ . Если  $f(\tilde{x}) < f(x')$ , то заменить  $x'$  на  $\tilde{x}$ ;
- 2) если для любого элемента списка  $x$  имеет место  $\sigma(x, \tilde{x}) = \text{false}$ , то добавить точку  $\tilde{x}$  в список.

Меру близости можно выбирать различными способами. Представляется естественным ввести меру близости на основании нормы разности векторов и заданного параметра  $\delta$ :

$$\sigma(x_1, x_2) = \begin{cases} \text{true}, & \|x_1 - x_2\| \leq \delta, \\ \text{false}, & \|x_1 - x_2\| > \delta. \end{cases}$$

Выбор параметра  $\delta$  существенно влияет на количество элементов в списке и, как следствие, на число просматриваемых терминальных последовательностей. При малых значениях этого параметра в списке оказывается много близких решений, что может привести к избыточному перебору. Большие значения параметра, наоборот, могут привести к тому, что исследуется недостаточно терминальных последовательностей. Эксперименты показывают, что наибольшую эффективность удастся получить при  $\delta \in [r, 1,5 \cdot r]$ .



Для алгоритма SMBH возможны различные критерии остановки счета. Первый основан на ограничении времени вычислений: вычисления останавливаются по прошествии заданного промежутка времени или выполнения заданного числа итераций алгоритма. Другой критерий остановки состоит в том, чтобы останавливать вычисления как только список минимумов становится пустым.

## **Параллельный вариант алгоритма систематического обхода**

Алгоритм SMBH допускает эффективную параллельную реализацию, так как просмотр окрестности можно выполнять различными процессорами независимо. Вычисления удобно организовать по схеме управляющий — рабочие. Пусть для расчетов выделено  $p$  процессоров. На каждом процессоре выполняется один процесс параллельной программы. Один из процессов, называемый управляющим, поддерживает список точек, остальные  $(p - 1)$  рабочих процессов выполняют поиск локальных минимумов в окрестности. Управляющий процесс выбирает очередную непомеченную точку из списка и отправляет ее тем свободным рабочим процессам, которым она еще не была отправлена. Точка помечается после того, как она была отправлена каждому рабочему процессу.

Рабочий процесс принимает от управляющего точку локального минимума  $x$ . Если число итераций выбрано равным  $N$ , то каждый рабочий процесс выполняет  $\frac{N}{p-1}$  раз следующую последовательность действий: точка  $x$  подвергается возмущению и последующей локальной оптимизации, в результате чего получается точка  $\tilde{x} = L_f(\Phi(x))$ . Если удалось улучшить значение целевой функции, то есть если  $f(\tilde{x}) < f(x)$ , то точка  $\tilde{x}$  добавляется в список рабочего процесса. После завершения цикла этот список пересылается управляющему процессу. Управляющий процесс, приняв список локальных минимумов от рабочего процесса, добавляет их свой список таким же образом, как и в последовательном варианте алгоритма.

## Результаты вычислительного эксперимента

Для вычислительного эксперимента была взята задача оптимизации конформации атомного кластера, которая рассматривается как важный тестовый пример для методов глобальной оптимизации, а также имеет важное прикладное значение.

Требуется найти конфигурацию из  $N$  атомов в трехмерном пространстве, которая обладает минимальной потенциальной энергией. Математически задача формулируется следующим образом:

$$f(x) = \sum_{i=1}^N \sum_{j=i+1}^N v(r_{ij}) \rightarrow \min ,$$

где  $x \in R^{3N}$  - вектор координат системы из  $N$  атомов в трехмерном пространстве, через  $r_{ij}$  обозначено расстояние между атомами  $i$  и  $j$ , вычисляемое по формуле

$$r_{ij} = \sqrt{(x_{3i}^2 - x_{3j}^2) + (x_{3i+1}^2 - x_{3j+1}^2) + (x_{3i+2}^2 - x_{3j+2}^2)} ,$$

а  $v: R \rightarrow R$  - функция потенциала, выражающая энергию взаимодействия двух атомов, находящихся на заданном расстоянии друг от друга. Рассматриваются различные потенциалы взаимодействия. Мы рассмотрим два классических потенциала: Леннарда—Джонса

$$v_{LJ}(r) = \frac{1}{r^{12}} - \frac{2}{r^6}$$

и Морса  $v_M(r) = e^{\rho(1-r)}(e^{\rho(1-r)} - 2)$ . В потенциале Морса параметр  $\rho$  может принимать значения 3, 6, 10, 14. Принято считать, что задача поиска минимума усложняется с ростом значения  $\rho$ .

Рассматриваемый в работе параллельный алгоритм систематического обхода был реализован с использованием параллельной библиотеки решения задач оптимизации BNB-Solver [11, 12]. Вычислительный эксперимент проводился на суперкомпьютере МВС-50000, характеристики которого приведены в табл. 1.

Таблица 1

Название суперкомпьютера	Процессоры	Сетевое оборудование
MVS-50000	640 процессоров Intel EM64T Xeon 51xx (Woodcrest) 3000 MHz (12 GFlops)	InifiniBand

В качестве локального алгоритма применялся метод сопряженных градиентов. Число  $N$  итераций возмущения было выбрано равным 1024. Радиус окрестности просмотра выбирался в пределах  $[0.6, 0.8]$ .

Для тестирования были взяты считающиеся наиболее трудными конфигурации, рассмотренные в [6, 7]. Результаты эксперимента приведены в табл. 2.

Таблица 2

Потенциал	Число атомов	Число процессоров	Время расчета	Значения целевой функции
LJ	98	32	180 минут	-543.665361
LJ	150	32	30 минут	-893.310258
Mors ( $\rho = 6$ )	80	32	20 минут	-378.333471
Mors ( $\rho = 10$ )	80	32	1 час	-344.911381
Mors ( $\rho = 14$ )	61	64	7 минут	-249.587740
Mors ( $\rho = 14$ )	79	256	8 минут	-336.798725
Mors ( $\rho = 14$ )	80	32	22 минуты	-340.811371

Все найденные минимальные значения совпадают с эталонными, приведенными в базе [10]. Проведенные эксперименты показывают, что предложенный алгоритм позволяет эффективно находить минимальные значения функции потенциальной энергии атомных кластеров, которая относится к числу наиболее сложных задач оптимизации.

## Другие работы

Эвристические алгоритмы, основанные на случайном переборе решений в некоторой окрестности, широко применяются при решении задачи нахождения глобального минимума функции многих переменных с большим числом локальных минимумов. Одной из классических задач такого рода является задача поиска оптимальной конфигурации системы атомов.

Наиболее успешные из существующих подходов к решению данной задачи основаны на использовании геометрических свойств опти-

мальных конфигураций атомов. В одной из первых работ в этом направлении [1] предлагается использовать решетку, полученную с помощью максимально плотной упаковки соприкасающихся сфер радиуса 1. Начальная конфигурация, полученная таким образом, используется затем в качестве начальной точки для поиска оптимальной конфигурации методами непрерывной оптимизации. Дальнейшие работы [3, 4, 5], развивающие этот подход, показали высокую эффективность данного метода применительно к задаче поиска оптимальной конфигурации атомов.

Другая группа работ посвящена подходам, основанным на вариантах метода случайного спуска по локальным минимумам (Basin Hopping). Наиболее успешными в данном направлении также являются методы, опирающиеся на специфичную для конкретной задачи функцию. В работе [6] предложен так называемый двухфазный метод. Строится функция, являющаяся модификацией целевой функции. Предполагается, что определить минимум модифицированной функции существенно проще, чем целевой. На первой фазе алгоритма находится минимум модифицированной функции. На второй фазе найденный минимум используется в качестве начальной точки для поиска минимума целевой функции. Модификация функции проводится путем добавления слагаемого — штрафа. Функция штрафа выбирается из геометрических соображений. Показано, что данный подход позволяет достаточно эффективно находить глобальные минимумы потенциальной энергии конфигурации атомов. Дальнейшее развитие этот подход получил в работе [7], где предлагается усилить алгоритм МВН за счет введения популяции решений. На каждом шаге обрабатывается не одно наилучшее решение, а набор, состоящий из нескольких лучших решений, найденных к некоторому моменту времени.

Так как решение задачи глобальной оптимизации требует, как правило, существенных вычислительных ресурсов, то применение методов параллельных и распределенных вычислений для решения таких задач достаточно распространено. В работе [8] предлагается параллельный вариант метода половинных делений [9]. Алгоритм половинных делений является методом ветвей и границ в котором декомпозиция состоит в разбиении параллелепипеда на два, а оценка получается с помощью константы Липшица. В работе показано, что с помощью такого метода можно находить существенную часть оптимальных конфигураций атомных кластеров, представленных в базе [10].

## Заключение

В работе предложены последовательный и параллельный алгоритмы систематического обхода пространства решений для решения задачи глобальной оптимизации функции многих переменных. Экспериментально показано, что параллельный вариант предложенного алгоритма позволяет находить решения наиболее сложных постановок для задачи поиска оптимальной конфигурации атомов, соответствующих потенциалу Морса при параметре  $\rho = 14$  при наличии достаточных вычислительных ресурсов. Основным достоинством алгоритма по сравнению с другими известными алгоритмами является то, что он не использует специфичной для задачи оптимизации молекулярных кластеров геометрической информации.

Автор глубоко благодарен действительному члену Российской академии наук, директору Вычислительного центра РАН Ю. Г. Евтушенко за постановку задачи, внимание к работе, многочисленные ценные рекомендации по совершенствованию алгоритма.

## Литература

1. *Leary R. H.* Global optimization on funneling landscapes // *J. Global Optim.* 2000. 18. 367–383.
2. *Northby J. A.* Structure and binding of Lennard-Jones clusters:  $13 \leq n \leq 147$  // *J. Chem. Phys.* 1987. 87. 6166–6178.
3. *Xue G. L.* Improvements on the Northby algorithm for molecular conformation: better solutions // *Journal of Global Optimization.* 1994. 4(4). 425–440.
4. *Doye J. P. K., Wales D. J., Berry R. S.* The effect of the range of the potential on the structure of clusters // *J. Chem. Phys.* 1995. 103. 4234–4249.
5. *Doye J. P. K. and Wales D. J.* Structural consequences of the range of the interatomic potential: A menagerie of clusters // *J. Chem. Soc., Faraday Trans.* 1997. 93. 4233–4244.
6. *Jonathan P. K. Doye, Robert H., Leary, Marco Locatelli, Fabio Schoen.* Global Optimization of Morse Clusters by Potential Energy Transformations // *INFORMS Journal on Computing.* 2004. 16(4). 371–379.
7. *Andrea Grosso, Marco Locatelli, Fabio Schoen.* A population-based approach for hard global optimization problems based on dissimilarity measures // *Mathematical Programming.* 2007. Vol. 110. 373–404.
8. *Евтушенко Ю. Г., Малкова В. В., Станевичюс А. А.* Распараллеливание процесса поиска глобального экстремума // *Автоматика и Телемеханика.* 2007. 46–59.

9. *Евтушенко Ю. Г., Ратькин В. А.* Метод половинных делений для глобальной оптимизации функции многих переменных // *Техническая кибернетика*. 1987. № 1. 119–127.
10. <http://www-wales.ch.cam.ac.uk/CCD.html>
11. *Посыпкин М. А., Сигал И. Х.* Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа // *ЖВМ и МФ*. 2005. Т. 45. № 10. С. 1801–1809.
12. *Посыпкин М. А.* Архитектура и программная организация библиотеки для решения задач оптимизации методом ветвей и границ на многопроцессорных вычислительных комплексах // *Проблемы вычислений в распределенной среде: распределенные приложения, коммуникационные системы, математические модели и оптимизация*. (Труды ИСА РАН.) М.: КомКнига/URSS, 2006. С. 18–25.