

УДК 004.023:519.853.4

doi 10.26089/NumMet.v16r224

ГИБРИДНЫЙ ЭВРИСТИЧЕСКИЙ ПАРАЛЛЕЛЬНЫЙ МЕТОД ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ

К. В. Пушкарев¹, В. Д. Кошур²

Рассматривается задача нахождения глобального минимума непрерывной целевой функции многих переменных в области, имеющей вид многомерного параллелепипеда. Для решения сложных задач глобальной оптимизации предлагается гибридный эвристический параллельный метод глобальной оптимизации (ГЭПМ), основанный на комбинировании и гибридизации различных методов и технологии многоагентной системы. В состав ГЭПМ включены как новые методы (например, метод нейросетевой аппроксимации инверсных зависимостей, использующий обобщенно-регрессионные нейронные сети (GRNN), отображающие значения целевой функции в значения координат), так и модифицированные классические методы (например, модифицированный метод Хука–Дживса). Кратко описывается программная реализация ГЭПМ в форме кроссплатформенной (на уровне исходного кода) программной библиотеки на языке C++, использующей обмен сообщениями через интерфейс MPI (Message Passing Interface). Приводятся результаты сравнения ГЭПМ с 21 современным методом глобальной оптимизации и генетическим алгоритмом на 28 тестовых целевых функциях 50 переменных.

Ключевые слова: глобальная оптимизация, эвристические методы, нейронные сети, параллельные вычисления, C++, MPI.

1. Введение. Сфера применения численной оптимизации широка: это задачи аппроксимации функций, обучения нейронных сетей, нахождения оптимальных параметров для инженерных задач, оптимальное планирование в экономике, нахождение оптимальных траекторий движения космических аппаратов [1] и др.

Прикладные задачи численной оптимизации могут иметь большую размерность. Например, упрощенная задача поиска оптимальной траектории полета космического аппарата к Меркурию имеет 26 переменных [2]. Задача обучения нейронной сети для системы машинного зрения, определяющей неровность поверхности при фрезеровании, требует определения 81 коэффициента путем минимизации ошибки [3]. В этой связи поиск численных методов решения оптимизационных задач с десятками и сотнями переменных представляет большой интерес.

К настоящему времени разработано множество методов — как локальных (градиентный, Ньютона, покоординатного спуска), так и глобальных (эволюционные алгоритмы [4], имитация отжига [5], метод Монте-Карло, метод усреднения координат [6], диагональные методы [7], методы покрытий [8]). Первые нацелены на отыскание локального экстремума, вторые — глобального.

Каждый из методов обладает своими преимуществами и ограничениями. Например, помимо обычных требований непрерывности целевой функции (ЦФ), метод Ньютона требует вычисления не только первой, но и второй производной ЦФ; диагональные методы требуют, чтобы ЦФ удовлетворяла условию Липшица. В то же время метод усреднения координат и метод покоординатного спуска формально не предъявляют дополнительных требований к ЦФ.

В настоящее время активно разрабатываются гибридные и кооперативные методы [9–16], которые предусматривают объединение в рамках одного алгоритма нескольких относительно самостоятельных алгоритмов оптимизации, которые могут обмениваться информацией между собой.

Как правило, не представляется возможным найти экстремум аналитически, нужные производные могут не существовать, либо отыскание их с нужной точностью может требовать слишком больших затрат. Среди методов глобальной оптимизации особое место занимают методы, рассматривающие ЦФ как “черный ящик”, т.е. вычислительную процедуру, об устройстве которой более ничего не известно, которая

¹ Сибирский федеральный университет, Институт космических и информационных технологий, ул. Киренского, 26Б, 660074, г. Красноярск; ст. преподаватель, e-mail: cyril.pushkaryov@yandex.ru

² Сибирский федеральный университет, Институт космических и информационных технологий, ул. Киренского, 26Б, 660074, г. Красноярск; профессор, e-mail: VKoshur@sfu-kras.ru

может быть достаточно ресурсоемкой. Такие методы предъявляют минимум требований к целевой функции, что делает их более универсальными. Таким образом, разработка численных методов глобальной оптимизации (ГО) является актуальной проблемой.

2. Постановка задачи. Постановка численной задачи глобальной оптимизации состоит в следующем. Рассматривается целевая функция (ЦФ) $\varphi = \Phi(\mathbf{x})$ от $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, $\Phi(\mathbf{x}): \Omega \rightarrow \mathbb{R}$, которая непрерывна в ограниченной области $\Omega = \{\mathbf{x}: L_i \leq x_i \leq U_i, i = \overline{1, n}\} \subset \mathbb{R}^n$. Необходимо найти приближенное минимальное значение функции Φ_{\min}^* и точку \mathbf{x}_{\min}^* , в которой достигается минимум:

$$\Phi_{\min} = \min_{\mathbf{x} \in \Omega} \Phi(\mathbf{x}) = \Phi(\mathbf{x}_{\min}), \quad \Phi_{\min}^* = \Phi(\mathbf{x}_{\min}^*) \leq \Phi_{\min} + \varepsilon_{\Phi}, \quad (1)$$

где $\varepsilon_{\Phi} > 0$ и определяет погрешность.

3. Описание метода. В основе гибридного эвристического параллельного метода (ГЭПМ) лежат две идеи: комбинирование различных методов ГО и многоагентный подход. Комбинирование методов для решения одной задачи ГО является известным приемом [12, 16, 17]. Многоагентный подход является органичной основой для решения проблемы организации масштабируемых параллельных вычислений и учета локальных особенностей ЦФ при решении задач глобальной оптимизации.

В ГЭПМ действует несколько оптимизационных агентов (ОА), исследующих определенную часть области поиска с помощью подчиненных алгоритмов (поисковых инструментов), описанных ниже. Каждый агент владеет собственной группой пробных точек, которая имеет ограниченный размер (избыточные точки с наибольшими значениями ЦФ автоматически отбрасываются). Такие ОА объединены в кластеры, обменивающиеся информацией (рис. 1).

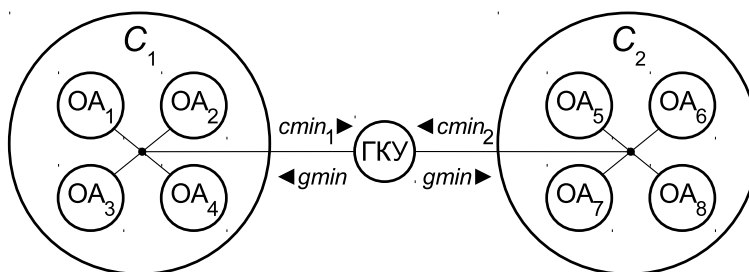


Рис. 1. Информационная топология соседства оптимизационных агентов (ОА); C_1, C_2 — кластеры, которые связаны через глобальный коммуникационный узел (ГКУ); $cmin_1, cmin_2$ — ведущие точки кластеров; $gmin$ — глобальная ведущая точка

Будем называть ведущей точкой ОА (кластера) пробную точку с наименьшим значением ЦФ из принадлежащих данному ОА (кластеру). Глобальной ведущей точкой назовем ведущую точку кластера с наименьшим значением ЦФ.

ОА имеет доступ к *локальным ориентирам* — ведущим точкам других ОА своего кластера, а также к *глобальному ориентиру* — глобальной ведущей точке. Эта информация может использоваться для улучшения собственного результата данного ОА.

Алгоритм ОА можно, в общих чертах, описать следующим образом.

1. Инициализация.
2. Если поисковая таблица (место хранения пробных точек) пуста, вызвать процедуру добавления случайных точек.
3. Уменьшить на единицу значения таймеров запуска поисковых инструментов.
4. Запустить поисковые инструменты, таймеры которых имеют значение 0. Перезапустить данные таймеры.
5. Если условия останова не выполнены, перейти на шаг 2.
6. Собрать статистику поисковых инструментов, подготовить результаты.

Таким образом, поисковые инструменты запускаются с заданной периодичностью. Преимущества вышеописанного подхода:

- ОА может сконцентрироваться на изучении определенного перспективного района области поиска;
- если область вокруг глобальной ведущей точки является перспективной, ОА будут тяготеть к ней (за счет сдвига при параллельном поиске), обеспечивая более тщательное ее исследование;
- он позволяет повысить точность либо адаптироваться к увеличению размера или размерности области поиска через увеличение числа ОА или кластеров;

- он позволяет использовать дополнительные вычислительные ресурсы для увеличения точности;
- он естественным образом отображается на кластерные вычислительные системы.

4. Поисковые инструменты. Ниже кратко описаны поисковые инструменты, входящие в ГЭПМ.

4.1. Параллельный поиск. Параллельный поиск основан на использовании ОА информации, полученной от других ОА системы.

На первом шаге составляется множество точек притяжения. В него включаются: текущие локальные ориентиры других ОА; текущий глобальный ориентир (если его нет среди локальных); исторически лучшие локальные ориентиры (опционально).

Для каждой точки притяжения вычисляется уклон, т.е. отношение разности значения ЦФ в точке притяжения и в собственной ведущей точке ОА к расстоянию между ними. Список уклонов сортируется по возрастанию. Вначале за точку назначения для сдвига пробных точек ОА принимается его собственная ведущая точка, что соответствует отсутствию сдвига.

По очереди для каждого ненулевого уклона из списка текущая точка назначения сдвигается в сторону отрицательных уклонов (в сторону убывания ЦФ) и против положительных уклонов, при этом уклон исключается из списка. После каждого сдвига оставшиеся уклоны пересчитываются, а список вновь сортируется. Процесс повторяется до тех пор, пока в списке есть ненулевые уклоны. Таким образом, точка назначения тяготеет к точкам притяжения с меньшими значениями ЦФ и отталкивается от точек с большими значениями ЦФ.

Величина сдвига выбирается случайным образом в заданном интервале (например, от 0.25 до 0.75 расстояния до точки притяжения) и домножается на модуль нормализованного уклона в этом направлении. Таким образом, более сильные уклоны приводят к более значительным сдвигам.

Процесс поиска точки назначения прерывается досрочно, как только в текущей точке назначения достигается улучшение значения ЦФ по сравнению с ее первоначальным положением.

Когда точка назначения найдена, точки ОА копируются и сдвигаются к ней параллельно так, чтобы образ ведущей точки оказался в точке назначения. Количество точек в группе оптимизационного агента ограничено, поэтому сдвинутые образы точек группы конкурируют с прообразами за место в группе. Точки с наибольшими значениями ЦФ отбрасываются.

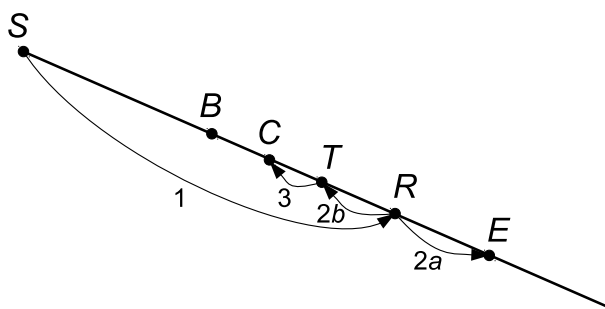


Рис. 2. Расположение точек при отражательном поиске

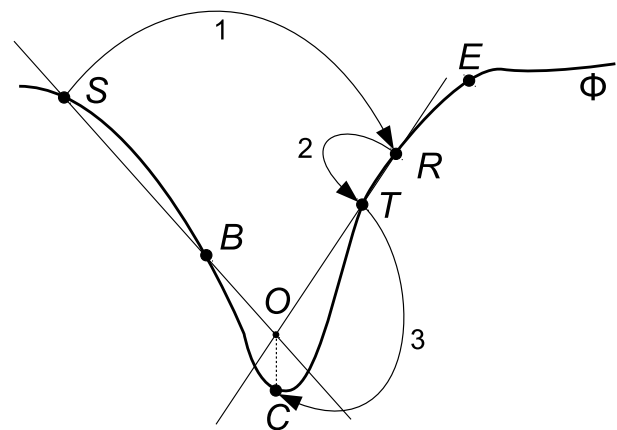


Рис. 3. Эвристика пересечения хорд, показанная на сечении ЦФ $\Phi(x)$ плоскостью

4.2. Отражательный поиск. Отражательный поиск напоминает метод Нелдера–Мида [18, с. 42]. Суть отражательного поиска поясняет рис. 2, на котором обозначены следующие точки: S — исходная, B — ведущая, R — отраженная, E — продленная, T — сокращенная, C — пересечения хорд. Для каждой пробной точки S , не совпадающей с ведущей B , выполняется нижеследующая процедура.

Определяется точка R следующим образом (шаг 1 на рис. 2): $R = B + U[u_{rl}, u_{ru}](B - S)$, где $U[l, u]$ — случайная величина, равномерно распределенная в интервале $[l, u]$. Если $\Phi(R) < \Phi(B)$, то делается попытка сделать шаг дальше (шаг 2a): $E = R + U[u_{el}, u_{eu}](R - B)$. Если $\Phi(E) < \Phi(R)$, то точка E включается во множество кандидатов, иначе — включается точка R . Если $\Phi(R) \geq \Phi(B)$, то итеративно делаются попытки найти более близкую точку (шаг 2b): $T_i = B + k_i^i(R - B)$, $i = 1, 2, \dots, n_t$, где $0 < k_t < 1$, n_t — заданные параметры. Если $\Phi(T_i) < \Phi(B)$, то точка T_i включается во множество кандидатов. Иначе, если $\Phi(B) \leq \Phi(T_i) < \Phi(R)$, то делается попытка найти точку C через пересечение хорд (шаг 3), как

показано на рис. 3. Пересечение хорд можно найти, поскольку точки S, B, R, T лежат на одной прямой по построению, а значит точки $(S, \Phi(S)), (B, \Phi(B)), (R, \Phi(R)), (T, \Phi(T))$ лежат в одной плоскости.

После обхода всех пробных точек точки из сформированного множества кандидатов добавляются ко множеству пробных.

4.3. Разведывательный поиск. Задачей разведывательного поиска является обследование пространства с целью поиска перспективных областей. Разведывательный поиск несет в себе аналогию с методом роя частиц (PSO), хотя значительно отличается от классического PSO. Разведывательный поиск состоит из двух фаз: генерации точек-разведчиков и продвижения.

Генерация происходит, если размер текущего множества точек-разведчиков ниже установленного. Из множества пробных точек случайным образом выбираются две различные точки; точку с бóльшим значением ЦФ обозначим через O , вторую — через P (рис. 4).

Тогда начальное положение S_1 и скорость v точки-разведчика задаются следующим образом:

$$v = r_1 r_2 \otimes (P - O); \quad S_1 = P + v, \quad (2)$$

где r_1 — случайный положительный коэффициент, равномерно распределенный на отрезке $[0.5; 1]$; r_2 — случайный вектор, компоненты которого равномерно распределены на отрезке $[0.9; 1.1]$; \otimes — покомпонатное умножение.

Продвижение точек-разведчиков состоит в изменении положения каждой точки в соответствии с ее скоростью ($S_i = S_{i-1} + v$).

Если при этом значение ЦФ увеличивается ($\Phi(S_i) > \Phi(S_{i-1})$), то из точки S_i предпринимается установленное количество шагов, по модулю равных значению скорости данного разведчика, в случайных направлениях. Если в каком-то направлении значение ЦФ оказывается меньше ($\Phi(S'_1) < \Phi(S_{i-1})$), то в этом направлении запускается новая точка-разведчик со скоростью $v_{S'} = S'_1 - S_i$.

Таким образом, когда точка-разведчик встречает препятствие на своем пути (увеличение значения ЦФ), то от ее местоположения могут быть запущены несколько новых точек-разведчиков (конкретное количество зависит от количества найденных успешных направлений) — происходит расщепление.

Если точка-разведчик достигает границ области поиска, то она уничтожается. Кроме того, время жизни точки-разведчика ограничено счетчиком, который уменьшается на единицу при каждом продвижении, не ведущем к уменьшению значения ЦФ в данной точке.

В конце работы алгоритма разведывательного поиска из текущего множества точек-разведчиков выбирается установленное их количество с наименьшими значениями ЦФ, которое сохраняется для следующего запуска, а также добавляется ко множеству пробных.

4.4. Локальный поиск. Для локального поиска используется метод, основанный на методе Хука–Дживса [18, с. 37]. Поскольку процедура локального поиска является не основной, для экономии вычислений ЦФ производится небольшое количество итераций (например, 10).

Метод Хука–Дживса является локальным методом прямого поиска. Он подходит для решения многомерных задач оптимизации и не требует определения производной ЦФ, а значит не ограничивает применимость гибридного метода.

Классический метод Хука–Дживса сочетает покомпонатные шаги исследования вокруг базовой точки с дополнительными шагами в найденных удачных направлениях (шагами по образцу). От классического метода Хука–Дживса применяемый в ГЭПМ вариант отличается адаптивными шагами, а также использованием, помимо классического шага по образцу, шага в усредненном направлении и шага в экстраполированном сплайнами направлении движения.

Адаптивность шагов состоит в умножении шага в i -м направлении на величину $e^{c_i s_i}$, где s_i — успешность i -го направления (с каждым успешным шагом в i -м направлении s_i увеличивается на 1, с каждым неуспешным уменьшается на 1). За начальный шаг принимается вектор размеров многомерного параллелепипеда, натянутого на множество пробных точек.

Цепочка из нескольких последних шагов запоминается. На основе этих данных делается шаг в усредненном направлении, а также в направлении, которое получается экстраполяцией цепочки сплайнами.

4.5. Координатный поиск. Координатный поиск исследует координаты и поведение ведущей точ-

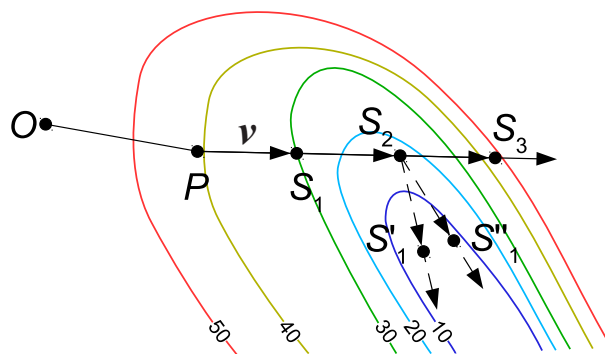


Рис. 4. Генерация и продвижение точек-разведчиков (S, S', S''), показанные на карте линий уровня целевой функции

ки. Его эвристическая мотивация состоит в том, что ведущая точка под действием различных приемов, включенных в ГЭПМ, “эволюционирует” в направлении точки глобального минимума. Соответственно, наблюдая и отслеживая тенденции этого развития, можно попытаться их усилить, чтобы ускорить движение к решению. Например, если обнаружено, что многие координаты ведущей точки группируются вокруг некоторого центра, можно попробовать установить координаты в этот центр, проверяя правильность догадки по изменению значения ЦФ. Аналогичным образом можно использовать наблюдаемые изменения в положении ведущей точки между запусками алгоритма координатного поиска.

В координатном поиске координаты ведущей точки $\mathbf{x}_{\min}^{[k]}$ и компоненты вектора ее смещения $\Delta = \mathbf{x}_{\min}^{[k]} - \mathbf{x}_{\min}^{[k-1]}$ с предыдущего запуска алгоритма координатного поиска кластеризуются алгоритмом векторного квантования Кохонена на предопределенное небольшое число (например, 5) кластеров. Пусть c_j — центр j -го кластера координат ведущей точки; d_j — центр j -го кластера компонент вектора смещения ведущей точки с предыдущего запуска.

Алгоритм координатного поиска.

1. $\mathbf{b} = \mathbf{x}_{\min}^{[k]}$.
2. Повтор последнего (с момента предыдущего запуска алгоритма координатного поиска) шага ведущей точки: $\mathbf{p} = \mathbf{b} + \left(\mathbf{x}_{\min}^{[k]} - \mathbf{x}_{\min}^{[k-1]} \right)$. Если $\Phi(\mathbf{p}) < \Phi(\mathbf{b})$, то $\mathbf{b} = \mathbf{p}$ и переход на шаг 4.
3. $\mathbf{p} = \mathbf{b}$, затем покоординатные кластерные шаги: $p_i = b_i + \arg \min_{x \in \{0, d_1, d_2, \dots\}} \Phi((p_1, \dots, p_{i-1}, b_i + x, p_{i+1}, \dots))$, $i = 1, 2, \dots$. Если $\Phi(\mathbf{p}) < \Phi(\mathbf{b})$, то $\mathbf{b} = \mathbf{p}$.
4. $\mathbf{p} = \mathbf{b}$, затем симметричный координатный кластерный прием: $\mathbf{p} = (c_j, c_j, \dots)$, $j = 1, 2, \dots$. Если $\Phi(\mathbf{p}) < \Phi(\mathbf{b})$, то $\mathbf{b} = \mathbf{p}$ и переход на шаг 7.
5. $\mathbf{p} = \mathbf{b}$, затем асимметричный координатный кластерный прием:

$$p_i = \arg \min_{x \in \{p_i, c_1, c_2, \dots\}} \Phi((p_1, \dots, p_{i-1}, x, p_{i+1}, \dots)), \quad i = 1, 2, \dots$$

Если $\Phi(\mathbf{p}) < \Phi(\mathbf{b})$, то $\mathbf{b} = \mathbf{p}$ и переход на шаг 7.

6. $\mathbf{p} = \mathbf{b}$, затем копирование случайно выбранной координаты:

$$p_i = \arg \min_{x \in \{p_i, p_{r_i}\}} \Phi((p_1, \dots, p_{i-1}, x, p_{i+1}, \dots)), \quad i = 1, 2, \dots,$$

где r_i — случайно выбранный номер координаты. Если $\Phi(\mathbf{p}) < \Phi(\mathbf{b})$, то $\mathbf{b} = \mathbf{p}$.

7. Вернуть результат: \mathbf{b} .

Границы области поиска учитываются: выходящие за границы значения координат приводятся к ближайшей границе.

4.6. Случайный поиск. Случайная генерация пробных точек используется для инициализации, а также подпитки процесса поиска.

Следует заметить, что с ростом размерности оптимизационной задачи роль случайного поиска быстро слабеет. Средний номер шага, на котором в алгоритме равномерного случайного бросания происходит попадание в $B(\epsilon)$ (ϵ -окрестность глобального минимума), оценивается как отношение мер Лебега: $\mu_n \{\Omega\} / \mu_n \{B(\epsilon)\}$ [19, с. 117]. В результате необходимое число шагов (вычислений ЦФ) быстро возрастает с размерностью. Например, при $\mu_n \{\Omega\} = 1$, чтобы попасть в $B(0.1)$ с вероятностью не менее 0.9, уже при размерности 10 требуется 9×10^{10} шагов, а при размерности 100 требуется 10^{140} шагов [19, с. 118].

Тем не менее, случайность является полезной, например в виде случайных возмущений параметров поисковых алгоритмов, описанных в настоящей работе (см., например, (2)).

Случайная инициализация множества пробных точек может производиться равномерным засеиванием области поиска. При этом локализация группы пробных точек обеспечивается за счет фокусировки дальнейшего поиска вокруг ведущей точки и постепенного вытеснения точек с периферии.

Другой вариант — определение случайного центра, построение вокруг этого центра многомерного параллелепипеда установленного размера (например, через задание масштабирующего коэффициента по

отношению к многомерному параллелепипеду, ограничивающему всю область поиска), в который бросаются случайные точки.

4.7. Поиск на основе нейросетевой аппроксимации инверсных зависимостей. Метод нейросетевой аппроксимации инверсных зависимостей, описанный в [20, 21], применяется как один из компонентов ГЭПМ. Его суть состоит в итеративном понижении значения целевой функции с отображением его в координаты с помощью инверсной зависимости, аппроксимация которой попутно уточняется новыми пробными точками.

Инверсные зависимости являются средством перехода от значений целевой функции к координатам. Они представляют собой покоординатные отображения значений ЦФ в пространство поиска:

$$\mathbf{x} = \Psi(\varphi) = (\Psi_1(\varphi), \Psi_2(\varphi), \dots, \Psi_n(\varphi)), \quad \varphi \in \mathbb{R}, \quad \mathbf{x} \in \Omega.$$

Инверсные зависимости *не являются, вообще говоря, обратными функциями*. Инверсных зависимостей может быть много, но в данной работе рассматриваются только те из них, которые удовлетворяют требованию $\Psi(\varphi) = \mathbf{x}_{\min}$ при $\varphi \leq \Phi_{\min}$.

Тогда решение исходной n -мерной задачи (1) сведется к решению n одномерных задач

$$x_{\min,i} = \Psi_i(\Phi_{\min}), \quad i = \overline{1, n}; \quad \Phi_{\min} = \arg \min_{\varphi} \Phi(\Psi_1(\varphi), \Psi_2(\varphi), \dots, \Psi_n(\varphi)).$$

Однако, поскольку мы располагаем ограниченной информацией о ЦФ в виде ее значений в пробных точках, невозможно надеяться на получение точной зависимости, обладающей описанными свойствами. Поэтому строятся ее аппроксимации с помощью обобщенно-регрессионных нейронных сетей (Generalized Regression Neural Networks, GRNN), справедливые в некотором интервале значений ЦФ:

$$(\Psi_1^{[k]}(\varphi), \Psi_2^{[k]}(\varphi), \dots, \Psi_n^{[k]}(\varphi)) = GRNN^{[k]}(\varphi, s^{[k]}),$$

где s — параметр сглаживания GRNN (spread); k — номер итерации.

Именно GRNN использовались для расчетов в данной работе. Кроме GRNN могут использоваться дуальные обобщенно-регрессионные нейронные сети, позволяющие задать область поиска минимума (фокус поиска) [22].

В классических GRNN с одним входом в первом (радиально-базисном) слое сети используется функция нахождения расстояния $\text{dist}^{[k]}(\varphi_1, \varphi_2) = |\varphi_1 - \varphi_2|$. Чтобы упростить настройку параметра сглаживания s , в данной работе функция была изменена: $\text{dist}^{[k]}(\varphi_1, \varphi_2) = \frac{|\varphi_1 - \varphi_2|}{\Phi_{\max}^{[k]} - \Phi_{\min}^{[k]}}$, где $\Phi_{\max}^{[k]}$ и $\Phi_{\min}^{[k]}$ — соответственно наибольшее и наименьшее значения ЦФ по множеству пробных точек на k -й итерации.

Лучшее значение ЦФ по множеству пробных точек (рекорд) $\Phi_{\min}^{[k]}$ на k -й итерации понижается на небольшую величину $\varepsilon^{[k]}$ и отображается с помощью GRNN в координаты, где вычисляется значение ЦФ:

$$\mathbf{x}_{\min}^{*[k]} = GRNN^{[k]}(\Phi_{\min}^{[k]} - \varepsilon^{[k]}, s^{[k]}). \tag{3}$$

Для параметра сглаживания s может быть выбрано множество значений, тогда операция (3) выполняется для каждого из них, и в результате получается множество точек.

Из новых точек во множество пробных добавляются только те, в которых значение ЦФ лучше рекорда.

При каждом запуске алгоритма нейросетевого поиска нейронная сеть обучается на текущем множестве пробных точек.

Данные шаги для функции $y = x^2$ проиллюстрированы на рис. 5. Красной точечной линией показана кривая $y = x^2$; P_1, \dots, P_5 — пробные точки, построенная по ним аппроксимация инверсной зависимости с помощью GRNN показана синим пунктиром; P_6 — точка, полученная с помощью инверсной зависимости; зеленым штрихпунктиром показана уточненная аппроксимированная кривая инверсной зависимости, построенная с учетом точки P_6 .

5. Программная реализация. Для ГЭПМ была написана кроссплатформенная (на уровне исходного кода) программная реализация в форме программной библиотеки на языке C++. Она предназначена для использования как на персональных компьютерах, так и на кластерных вычислительных системах.

В программной реализации кластеры оптимизационных агентов отображаются в процессы, а сами агенты — в потоки внутри процессов (рис. 6). Прямоугольниками обозначены процессы, кругами — потоки. Главным процессом является C_1 , OA_i — i -й оптимизационный агент, ЛКА — локальный коммуникационный агент, ГКА — глобальный коммуникационный агент, MPI — интерфейс MPI (Message Passing

Interface), ПУ — поток управления. Кроме того, существуют служебные потоки, отвечающие за передачу данных: локальные коммуникационные агенты (ЛКА; по одному в каждом процессе, обмениваются данными с глобальным коммуникационным агентом), глобальный коммуникационный агент (ГКА; один на все процессы, собирает сведения о ведущих точках кластеров, определяет и распространяет сведения о глобальной ведущей точке).

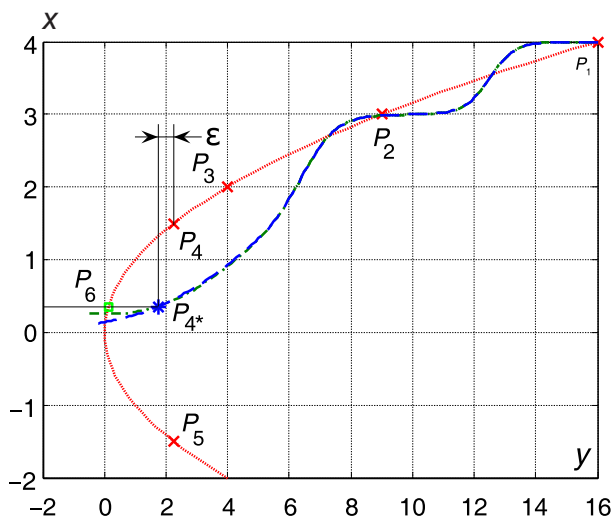


Рис. 5. Иллюстрация принципа работы алгоритма нейросетевой аппроксимации инверсных зависимостей

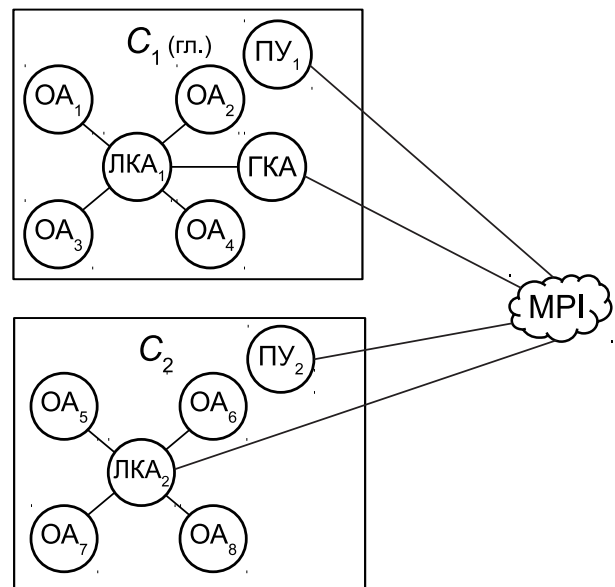


Рис. 6. Процессы и потоки в программной реализации метода, показанные для 2 кластеров по 4 оптимизационных агента

Главный поток каждого процесса становится потоком управления, отвечающим за запуск и завершение других потоков и передачу итоговых результатов и статистики главному процессу.

Если в решении задачи участвуют несколько процессов, то они обмениваются сообщениями через интерфейс MPI (Message Passing Interface) версии 2.2 [23]. Оптимизационные агенты одного процесса обмениваются информацией между собой через таблицу локальных ориентиров, находящуюся в общей памяти (памяти процесса), что позволяет минимизировать задержки при обмене.

Количество ОА в процессе является параметром программной реализации ГЭПМ, который передается ей перед началом вычислений. При повышении этого количества увеличивается количество доступной ОА информации (за счет увеличения числа локальных ориентиров), обмен информацией между ОА становится более интенсивным (за счет быстрого обмена через общую память), но при этом увеличиваются вычислительные затраты в самих ОА (на обработку локальных ориентиров в инструменте параллельного поиска), а также нагрузка на ЛКА.

При распределении ОА по процессам есть два крайних варианта:

- размещение в каждом процессе единственного ОА (при этом ОА доступен только один, глобальный, ориентир, весь обмен информацией осуществляется через MPI, вспомогательные потоки процесса (поток ЛКА, журнала и т. д.) обслуживают один ОА);
- размещение всех ОА в единственном процессе (при этом ОА доступны ориентиры всех других ОА, весь обмен информацией осуществляется через общую память, вспомогательные потоки процесса обслуживают все ОА). Этот вариант очевидно невыгоден для кластерных вычислительных систем.

При использовании этих вариантов двухуровневая информационная структура (ОА и кластер ОА) вырождается в одноуровневую.

Влияние количества ОА в процессе на ход вычислений является разносторонним и представляет собой тему для дальнейших исследований. На практике количество ОА в процессе необходимо подбирать экспериментально, принимая во внимание количество вычислительных ядер на узле.

6. Методика сравнения с другими методами ГО. Для оценки эффективности предложенного метода было проведено сравнение с другими численными методами ГО. Использовался набор тестовых

функций и правила соревнований методов ГО в рамках IEEE Congress on Evolutionary Computation (СЕС-2013) [24]. Результаты ГЭПМ сравнивались с опубликованными результатами 21 конкурсанта СЕС-2013 по состоянию на 4 июля 2013 г. и с генетическим алгоритмом.

Тестовый набор СЕС-2013 состоит из 28 тестовых функций многих переменных. Он построен на классических тестовых функциях, в которые введены существенные усложняющие изменения в различных комбинациях: вращение координатного пространства, возмущения в виде колебаний с различными частотами, линейное и нелинейное сжатие по координатам. Функции в наборе поделены на три раздела [25, с. 4]: № 1–5 — одноэкстремальные; № 6–20 — базовые многоэкстремальные (основанные на классических тестовых функциях Растригина, Розенброка, Экли и др.); № 21–28 — составленные специальным образом [25, с. 24] из функций № 1–20. Подробное описание тестового набора можно найти в [25].

Глобальный минимум у всех функций единственный. Положение точки глобального минимума у всех функций одинаковое и заранее известное, его координаты были выбраны организаторами соревнований случайным образом на отрезке $[-80, 80]$. Значения ЦФ в точке глобального минимума у всех функций различные и ненулевые [25, с. 4].

Согласно процедуре СЕС-2013, все функции рассматривались как черные ящики, использовать их формулы в процессе решения не разрешалось.

В данной работе использовался упрощенный вариант процедуры СЕС-2013: промежуточные значения погрешности после 0.01, 0.1, 0.2, 0.3, 0.4, ..., 0.9 от максимального количества вычислений ЦФ не записывались, тест сложности алгоритма не проводился.

Условия экспериментов были следующими:

- 50 переменных;
- не более 500 тыс. вычислений ЦФ;
- область поиска от -100 до 100 по каждой координате;
- 51 испытание со случайной инициализацией по каждой задаче.

Аппаратное обеспечение: 4-ядерный процессор Intel Core i5-2400 (3.1 ГГц), 2 Гбайт ОЗУ.

ГЭПМ использовал 2 кластера по 4 оптимизационных агента. Инициализация: равномерная случайная, инициализация генератора случайных чисел по времени запуска программы.

Кроме останова по количеству вычислений ЦФ, ГЭПМ останавливался, если за последние 25 итераций изменение ведущего значения ЦФ или положения ведущей точки оказывалось меньше 0.001.

В оптимизационном агенте ГЭПМ прерывание процесса происходит между итерациями главного цикла, в котором применяются поисковые инструменты, поэтому возможно некоторое превышение установленного лимита количества вычислений ЦФ. Чтобы лимит в 500 тыс., требуемый условиями сравнения, не был нарушен, технический лимит количества вычислений ЦФ в ГЭПМ был установлен равным 490 тыс., что дает 245 тыс. для каждого кластера оптимизационных агентов.

Прочие основные параметры ГЭПМ, использовавшиеся в экспериментах, были следующими. Группа пробных точек ОА состояла из $N_p = 30$ точек. Период параллельного, локального, координатного и случайного поиска был равен 10 итерациям главного цикла ОА, отражательного — 5, разведывательного — 4, нейросетевого — 1. В нейросетевом поиске использовалась нейросеть GRNN с параметрами сглаживания (spread) 0.25, 0.5, 0.75; величина понижения лучшего значения ЦФ на k -й итерации (3) определялась по формуле $\varepsilon^{[k]} = 3 \left(\Phi_{\max}^{[k]} - \Phi_{\min}^{[k]} \right) / N_p$, где $\Phi_{\max}^{[k]}$ и $\Phi_{\min}^{[k]}$ — соответственно наибольшее и наименьшее значение ЦФ по множеству пробных точек ОА на k -й итерации. В разведывательном поиске использовалось 25 точек-разведчиков с начальным временем жизни 25 и 10 попытками смены направления движения в случае увеличения значения ЦФ. В отражательном поиске: $[u_{rl}, u_{ru}] = [0.5, 1.5]$, $[u_{el}, u_{eu}] = [0.5, 1.0]$, $k_t = 0.5$, $n_t = 10$. В локальном поиске $c_l = 0.2$. Количество кластеров в координатном поиске было равно 5. Количество вбрасываемых точек за один запуск инструмента случайного поиска было равно 10.

Сравнение с участниками СЕС-2013 проводилось по $\Delta\Phi = \Phi_{\min}^* - \Phi_{\min}$ — абсолютной погрешности значения ЦФ, где Φ_{\min}^* — приближенное значение минимума ЦФ, найденное ГЭПМ; Φ_{\min} — точное значение минимума ЦФ, известное из описаний тестовых функций [25, с. 4].

Для каждой задачи определялись по 51 испытанию минимум погрешности, максимум погрешности, среднее значение погрешности, медиана погрешности. Для каждой задачи и каждой статистики погрешности определялся ранг — количество конкурентов, у которых соответствующее значение меньше, чем у ГЭПМ. Таким образом, для каждой задачи ГЭПМ присваивалось 4 ранга: по минимуму погрешности, по максимуму погрешности, по средней погрешности и по медиане погрешности.

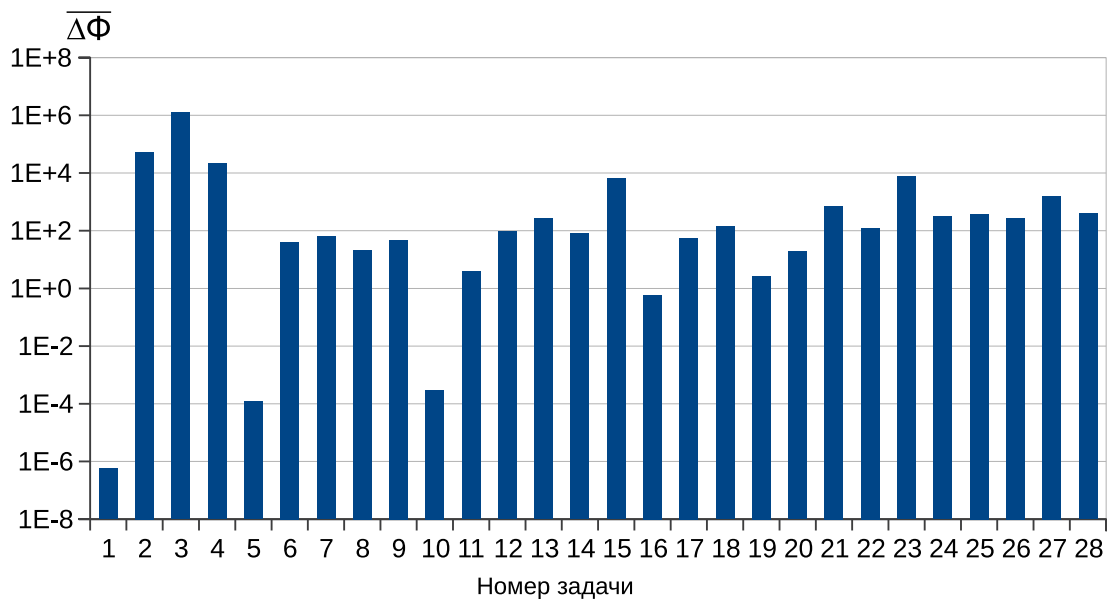


Рис. 7. Средние абсолютные погрешности минимальных значений ЦФ, найденных ГЭПМ для набора тестовых задач СЕС-2013. Для каждой задачи усреднение проводилось по 51 испытанию. Шкала логарифмическая

7. Результаты сравнения с другими методами ГО. На рис. 7 показаны средние абсолютные погрешности минимальных значений ЦФ ($\overline{\Delta\Phi}$), найденных ГЭПМ для каждой из задач.

Для каждой задачи были определены средние квадратические отклонения абсолютных погрешностей минимальных значений ЦФ ($\sigma_{\Delta\Phi}$), найденных ГЭПМ. По всем задачам в целом минимум средних квадратических отклонений составил 3.2×10^{-7} (в задаче № 1), максимум составил 1.4×10^6 (в задаче № 3), среднее составило 5.2×10^4 , нижний квартиль составил 0.97, медиана составила 11, верхний квартиль составил 96.

Для каждой задачи были определены средние количества вычислений ЦФ, произведенных ГЭПМ в каждой из задач, в процентах от технического лимита 490 тыс. По всем задачам в целом минимум этих значений составил 36.0% (в задаче № 1), максимум составил 101.1% (в задаче № 4), среднее составило 87.9%, нижний квартиль составил 85.7%, медиана составила 99.8%, верхний квартиль составил 100.8%, в 19 задачах из 28 они превысили 95%.

Максимальное по всем задачам количество вычислений ЦФ, произведенных ГЭПМ, составило 498 920. Таким образом, установленное условиями сравнения ограничение в 500 тыс. вычислений ЦФ не было превышено.

При каждом запуске поискового инструмента определялся прогресс — разность между минимальным значением ЦФ по множеству пробных точек непосредственно перед запуском алгоритма поискового инструмента и после завершения его работы. Таким образом, положительный прогресс свидетельствует об улучшении текущего приближения глобального минимума.

Для каждой пары, состоящей из задачи и инструмента, прогресс был просуммирован по 51 испытанию, а затем подсчитано отношение суммарного прогресса каждого инструмента в отдельности к сумме прогрессов всех инструментов вместе. Эту величину можно интерпретировать как относительную меру непосредственной (т.е. не за счет влияния на другие инструменты) успешности инструмента в данной задаче. Минимальные, средние и максимальные значения указанных величин по всем тестовым задачам приведены в таблице.

По всем задачам в целом минимальный ранг ГЭПМ в сравнении с участниками СЕС-2013 по средней

Минимальные, средние и максимальные по всем задачам СЕС-2013 относительные прогрессы поисковых инструментов ГЭПМ

Инструмент	Относительный прогресс, %		
	мин.	сред.	макс.
Параллельного поиска	0.00	0.00	0.00
Отражательного поиска	0.01	2.38	14.12
Разведывательного поиска	0.05	2.47	14.46
Локального поиска	0.01	42.00	87.70
Координатного поиска	0.00	6.59	39.42
Случайного поиска	0.01	1.43	14.83
Нейросетевого поиска	0.12	45.12	99.93

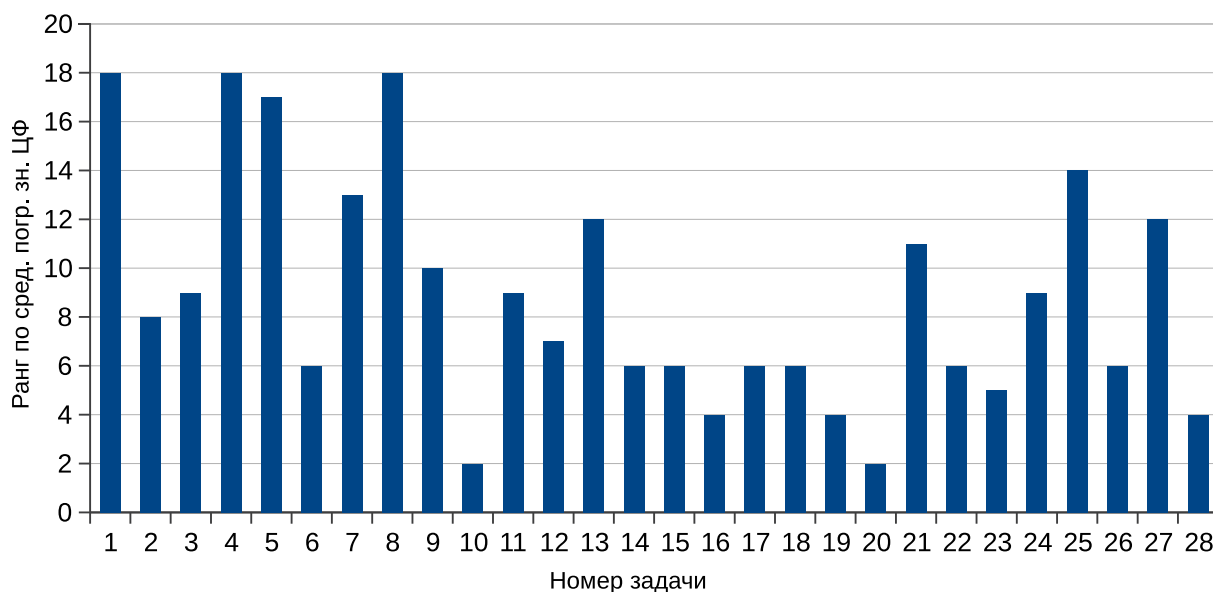


Рис. 8. Ранги ГЭПМ в сравнении с участниками СЕС-2013 по средней абсолютной погрешности значений ЦФ

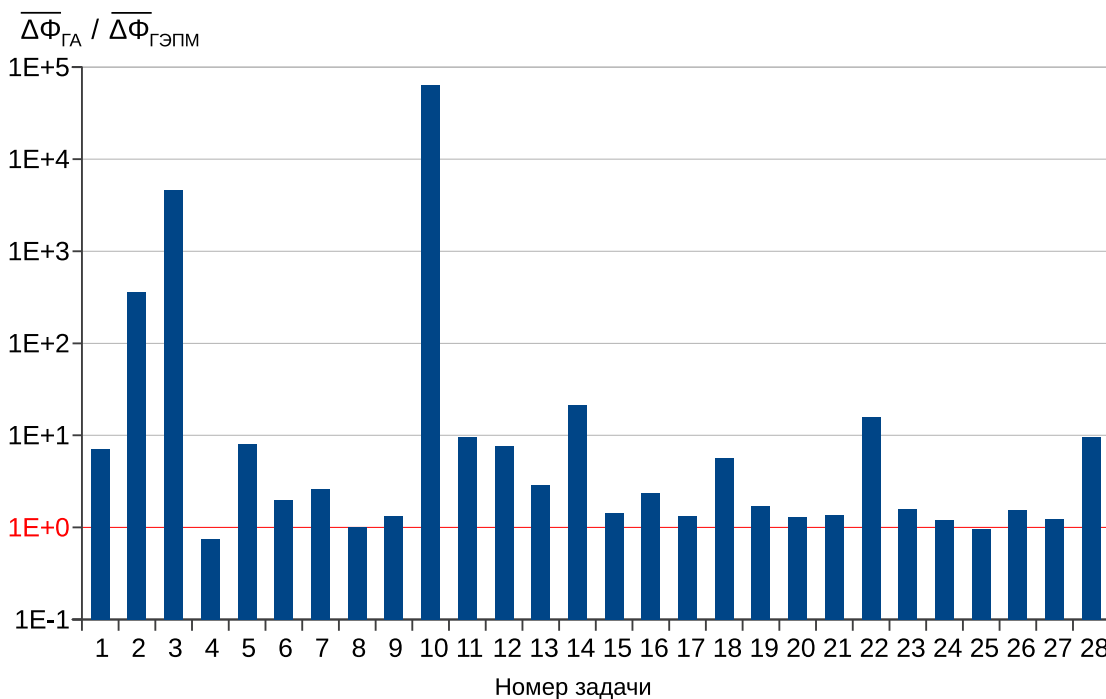


Рис. 9. Отношения средних абсолютных погрешностей ГА и ГЭПМ по задачам. Шкала логарифмическая

абсолютной погрешности найденных значений ЦФ составил 2 (в задачах № 10, 20), максимальный составил 18 (в задачах № 1, 4, 8), средний составил 9, нижний квартиль ранга составил 6, медиана ранга составила 7.5, верхний квартиль составил 12. Ранги по отдельным задачам показаны на рис. 8.

Для сравнения ГЭПМ с генетическим алгоритмом (ГА) использовалась реализация ГА из Genetic Algorithm and Direct Search Toolbox 2.0.1 (MATLAB) со стандартными параметрами. Генетический алгоритм был протестирован по вышеописанной процедуре. Для каждой задачи была определена усредненная по 51 испытанию абсолютная погрешность минимальных значений ЦФ, определенных ГА. Отношения средних абсолютных погрешностей ГА и ГЭПМ ($\overline{\Delta\Phi}_{ГА} / \overline{\Delta\Phi}_{ГЭПМ}$) показаны на рис. 9.

8. Выводы. Как видно на рис. 7, наибольшая средняя абсолютная погрешность (1.27×10^6) наблюдалась в задаче № 3, наименьшая (5.69×10^{-7}) — в задаче № 1. Несмотря на высокую погрешность в задаче № 3, ранг в сравнении с участниками СЕС-2013 в этой задаче равен всего 9. Необходимо заметить, что в задаче № 3 точный минимум равен -1200 , но в рассматриваемой области поиска для данной ЦФ

характерны значительно более высокие значения. Так, на множестве из 1000 случайным образом сгенерированных точек с равномерно распределенными от -100 до 100 координатами ЦФ № 3 показала значения порядка от 10^{17} до 10^{51} при среднем значении порядка 10^{48} .

Усредненный по всем задачам ранг по средней погрешности значений ЦФ, найденных ГЭПМ, оказался равен 9 (из 22). Таким образом, на тестовом наборе сложных целевых функций 50 переменных СЕС-2013 ГЭПМ по точности сравним с современными методами глобальной оптимизации, участвовавшими в соревнованиях СЕС-2013.

Точность ГЭПМ в сравнении с другими методами в разных задачах значительно различается. Так, лучшие результаты по средней погрешности метод показал в задачах № 10, 20 (ранг равен 3), худшие — в задачах № 1, 4, 8 (ранг 18). В целом, в 50% случаев ранг ГЭПМ по средней погрешности не превышал 7.5, а в 75% случаев не превышал 12.

Как видно на рис. 9, на сложном тестовом наборе СЕС-2013 точность ГЭПМ значительно превышает точность генетического алгоритма почти во всех задачах. В сравнении с генетическим алгоритмом лучшие результаты по средней погрешности ГЭПМ показал в задаче № 10 (отношение погрешности ГА к погрешности ГЭПМ равно 63500.65), худшие — в задаче № 4 (отношение погрешностей равно 0.75).

По-видимому, одной из причин, не позволяющих ГЭПМ показать более высокую точность, является то, что лимит в 500 тыс. вычислений ЦФ достаточно жесткий для ГЭПМ: среднее использование отведенного количества вычислений ЦФ в 19 задачах из 28 превысило 95%.

Отметим, что в ГЭПМ “недоиспользованные” вычисления ЦФ не передаются с одного кластера агентов на другой, а также отсутствует адаптация активности разных поисковых инструментов к конкретной задаче, из-за чего менее эффективные для данной задачи инструменты могут отнимать ресурсы у более эффективных. Из этого следует, что необходимо искать пути дальнейшего повышения эффективности использования вычислений ЦФ предложенным методом.

Как видно из таблицы, успешность поисковых инструментов существенно зависела от задачи. Так, доля локального поиска в суммарном прогрессе в задаче № 3 составила 0.01%, а в задаче № 14 достигла 87.70%. Для каждого инструмента, за исключением параллельного поиска, нашлись как задачи, в которых инструмент показывал малую успешность, так и задачи, в которых успешность была значительной. Это говорит в пользу гибридных и кооперативных методов глобальной оптимизации, к которым относится ГЭПМ. Полученный нулевой прогресс инструмента параллельного поиска ставит вопрос о роли данного инструмента в решении задачи. Для изучения этого вопроса были отобраны из набора СЕС-2013 задачи № 1, 5, 10, для которых погрешность в проведенных экспериментах была минимальной (рис. 7), и задачи № 2–4, для которых погрешность была максимальной. Для каждой из отобранных 6 задач были проведены 2 серии экспериментов по 102 испытания: с параллельным поиском и без параллельного поиска. В обоих случаях ГЭПМ использовал 2 кластера по 4 оптимизационных агента. Полученные пары выборок по 102 абсолютные погрешности найденных ГЭПМ минимальных значений ЦФ были сопоставлены с помощью критерия Вилкоксона–Манна–Уитни U [26, с. 660] при уровне значимости 0.05. В результате было установлено, что для задач № 3, 4, 10 абсолютные погрешности с параллельным поиском были значимо ниже, чем без него. Для задач № 1, 2, 5 отличия абсолютных погрешностей с параллельным поиском и без него не были значимыми.

Таким образом, инструмент параллельного поиска улучшает результаты для некоторых ЦФ, но обычно не дает непосредственного улучшения значения ЦФ в ведущей пробной точке. Можно предположить, что добавляемые параллельным поиском пробные точки положительно влияют на результаты других поисковых инструментов.

СПИСОК ЛИТЕРАТУРЫ

1. Izzo D., Vinkó T. Global trajectory optimisation problems database. <http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html>. Cited April 20, 2015.
2. Izzo D., Vinkó T. Messenger (full version). http://www.esa.int/gsp/ACT/inf/projects/gtop/messenger_full.html. Cited April 20, 2015.
3. Palani S., Natarajan U. Prediction of surface roughness in CNC end milling by machine vision system using artificial neural network based on 2D Fourier transform // Int. J. Adv. Manuf. Technol. 2011. **54**, N 9–12. 1033–1042.
4. Spears W.M., de Jong K.A., Bäck T., Fogel D.B., Garis H. An overview of evolutionary computation // Lecture Notes in Computer Science. Vol. 667. Heidelberg: Springer, 1993. 442–459.
5. Corana A., Marchesi M., Martini C., Ridella S. Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm // ACM Transactions on Mathematical Software. 1987. **13**, N 3. 262–280.
6. Рубан А.И. Глобальная оптимизация методом усреднения координат. Красноярск: ИПЦ КГТУ, 2004.
7. Сергеев Я.Д., Квасов Д.Е. Диагональные методы глобальной оптимизации. М.: Физматлит, 2008.

8. *Евтушенко Ю.Г.* Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Журнал вычислительной математики и математической физики. 1971. **11**, № 6. 1390–1403.
9. *Кошур В.Д.* Глобальная оптимизация на основе гибридного метода усреднения координат и метода роя частиц // Вычислительные технологии. 2013. **18**, № 4. 36–47.
10. *Процыков Г.В., Семенкин Е.С., Токмин К.А.* Об эффективности коэволюционного подхода в практических задачах оптимизации // Вестник Красноярского государственного университета. Серия “Физико-математические науки”. 2005. № 4. 233–239
11. *Горнов А.Ю.* Параллельный алгоритм поиска оптимального управления в задачах с параллелепипедными ограничениями // Вестник ИрГТУ. 2004. № 2. 104–110.
12. *Esa/pagmo*. 2015. <https://github.com/esa/pagmo>. Cited April 20, 2015.
13. *Izzo D.* PyGMO and PyKEP: open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization) // Proc. of the 5th Int. Conf. on Astrodynamics Tools and Techniques, ICATT. 2012. [http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2012-\(ICATT\)PyKEP-PaGMO-SOCIS.pdf](http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2012-(ICATT)PyKEP-PaGMO-SOCIS.pdf). Cited April 20, 2015.
14. *Kleymenov A., Semenov A.* Using a cooperative solving approach to global optimization problems // Lecture Notes in Computer Science. Vol. 3478. Heidelberg: Springer, 2005. 86–100.
15. *Evtushenko Y., Posypkin M., Sigal I.* A framework for parallel large-scale global optimization // Computer Science — Research and Development. 2009. **23**, N 3–4. 211–215.
16. *Ахмедова Ш.А., Семенкин Е.С.* Новый коллективный метод оптимизации на основе кооперации бионических алгоритмов // Вестник Сибирского государственного аэрокосмического университета. 2013. № 4. 92–99.
17. *Wang H., Ersoy O.* Novel evolutionary global optimization algorithms and their applications. Purdue University Technical Report. <http://docs.lib.purdue.edu/ecetr>. Cited April 20, 2015.
18. *Банди Б.* Методы оптимизации. Вводный курс. М.: Радио и связь, 1988.
19. *Жигляевский А.А., Жилинскас А.Г.* Методы поиска глобального экстремума. М.: Наука, 1991.
20. *Кошур В.Д., Пушкарев К.В.* Глобальная оптимизация на основе нейросетевой аппроксимации инверсных зависимостей // Труды XIII Всероссийской научно-технической конференции “Нейроинформатика-2011”: в 3 ч. Ч. 1. М.: НИЯУ МИФИ, 2010. 89–98.
21. *Koshur V.D., Pushkaryov K.V.* Global optimization via neural network approximation of inverse coordinate mappings // Optical Memory and Neural Networks. 2011. **20**, N 3. 181–193.
22. *Кошур В.Д., Пушкарев К.В.* Дуальные обобщенно-регрессионные нейронные сети для решения задач глобальной оптимизации // Труды XII Всероссийской научно-технической конференции “Нейроинформатика-2010”: в 2 ч. Ч. 2. М.: НИЯУ МИФИ, 2010. 219–227.
23. MPI: A Message-Passing Interface Standard. Version 2.2. <http://mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>. Cited April 20, 2015.
24. Special Session & Competition on Real-Parameter Single Objective Optimization at CEC-2013. http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm. Cited April 20, 2015.
25. *Liang J.J., Qu B.Y., Suganthan P.N., Hernández-Díaz A.G.* Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical Report. http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/Definitions%20of%20CEC%2013%20benchmark%20suite%200117.pdf. Cited April 20, 2015.
26. *Кендалл М., Стьюарт А.* Статистические выводы и связи. М.: Наука, 1973.

Поступила в редакцию
08.01.2015

A Hybrid Heuristic Parallel Method of Global Optimization

K. V. Pushkaryov¹ and V. D. Koshur²

¹ *Siberian Federal University, Institute of Space and Information Technologies; ulitsa Kirenskogo 26, Krasnoyarsk, 660074, Russia; Senior Lecturer, e-mail: cyril.pushkaryov@yandex.ru*

² *Siberian Federal University, Institute of Space and Information Technologies; ulitsa Kirenskogo 26, Krasnoyarsk, 660074, Russia; Dr. Sci., Professor, e-mail: VKoshur@sfu-kras.ru*

Received January 8, 2015

Abstract: The problem of finding the global minimum of a continuous objective function of multiple variables in a multidimensional parallelepiped is considered. A hybrid heuristic parallel method for solving of

complicated global optimization problems is proposed. The method is based on combining various methods and on the multi-agent technology. It consists of new methods (for example, the method of neural network approximation of inverse coordinate mappings that uses Generalized Regression Neural Networks (GRNN) to map the values of an objective function to coordinates) and modified classical methods (for example, the modified Hooke–Jeeves method). An implementation of the proposed method as a cross-platform (on the source code level) library written in the C++ language is briefly discussed. This implementation uses the message passing via MPI (Message Passing Interface). The method is compared with 21 modern methods of global optimization and with a genetic algorithm using 28 test objective functions of 50 variables.

Keywords: global optimization, heuristic methods, neural networks, parallel computing, C++, MPI.

References

1. D. Izzo and T. Vinkó, “Global Trajectory Optimisation Problems Database,” <http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html>. Cited April 20, 2015.
2. D. Izzo and T. Vinkó, “Messenger (full version),” http://www.esa.int/gsp/ACT/inf/projects/gtop/messenger_full.html. Cited April 20, 2015.
3. S. Palani and U. Natarajan, “Prediction of Surface Roughness in CNC End Milling by Machine Vision System Using Artificial Neural Network Based on 2D Fourier Transform,” *Int. J. Adv. Manuf. Technol.* **54** (9–12), 1033–1042 (2011).
4. W. M. Spears, K. A. De Jong, T. Bäck, et al., “An Overview of Evolutionary Computation,” in *Lecture Notes in Computer Science* (Springer, Heidelberg, 1993), Vol. 667, pp. 442–459.
5. A. Corana, M. Marchesi, C. Martini, and S. Ridella, “Minimizing Multimodal Functions of Continuous Variables with the “Simulated Annealing” Algorithm,” *ACM Trans. Math. Softw.* **13** (3), 262–280 (1987).
6. A. I. Ruban, *Global Optimization by the Coordinate Averaging Method* (Krasnoyarskii Tekh. Univ., Krasnoyarsk, 2004) [in Russian].
7. Ya. D. Sergeev and D. E. Kvasov, *Diagonal Methods of Global Optimization* (Fizmatlit, Moscow, 2008) [in Russian].
8. Yu. G. Evtushenko, “Numerical Method for Finding Global Extrema (Case of a Non-Uniform Mesh),” *Zh. Vychisl. Mat. Mat. Fiz.* **11** (6), 1390–1403 (1971) [*USSR Comput. Math. Math. Phys.* **11** (6), 38–54 (1971)].
9. V. D. Koshur, “Global Optimization Based on Hybrid Method of Coordinates Averaging and Particle Swarm Optimization,” *Vychisl. Tekhnol.* **18** (4), 36–47 (2013).
10. G. V. Protsykov, E. S. Semenin, and K. A. Tokmin, “The Effectiveness of Co-Evolutionary Approach in Practical Optimization Problems,” *Vestn. Krasnoyarsk. Univ. Ser. Fiz. Mat. Nauk*, No. 4, 233–239 (2005).
11. A. Yu. Gornov, “A Parallel Algorithm of Searching for Optimal Control in Problems with Parallelepipedic Constraints,” *Vestn. Irkutsk. Univ.*, No. 2, 104–110 (2004).
12. Esa/pagmo. <https://github.com/esa/pagmo>. Cited April 20, 2015.
13. D. Izzo, “PyGMO and PyKEP: Open Source Tools for Massively Parallel Optimization in Astrodynamics (the Case of Interplanetary Trajectory Optimization),” in *Proc. 5th Int. Conf. on Astrodynamics Tools and Techniques, ICATT. 2012*. [http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2012-\(ICATT\)PyKEP-PaGMO-SOCIS.pdf](http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2012-(ICATT)PyKEP-PaGMO-SOCIS.pdf). Cited April 20, 2015.
14. A. Kleymenov and A. Semenov, “Using a Cooperative Solving Approach to Global Optimization Problems,” in *Lecture Notes in Computer Science* (Springer, Heidelberg, 2005), Vol. 3478, pp. 86–100.
15. Yu. Evtushenko, M. Posypkin, and I. Sigal, “A Framework for Parallel Large-Scale Global Optimization,” *Comput. Sci. Res. Develop.* **23** (3–4), 211–215 (2009).
16. Sh. A. Akhmedova and E. S. Semenin, “New Optimization Metaheuristic Based on Co-Operation of Biology Related Algorithms,” *Vestn. Sib. Aerokosmich. Univ.*, No. 4, 92–99 (2013).
17. H. Wang and O. Ersoy, “Novel Evolutionary Global Optimization Algorithms and Their Applications,” Purdue University Technical Report. <http://docs.lib.purdue.edu/ecetr>. Cited April 20, 2015.
18. B. D. Bunday, *Basic Optimisation Methods* (Arnold, London, 1985; Radio Svyaz’, Moscow, 1988).
19. A. A. Zhigliavsky and A. G. Zhilinskas, *Methods of Global Optimization* (Nauka, Moscow, 1991) [in Russian].
20. V. D. Koshur and K. V. Pushkaryov, “Global Optimization Based on Neural Network Approximation of Inverse Dependences,” in *Proc. 13th All-Russia Conf. on Neuroinformatics, Moscow, Russia, January 24–28, 2011* (Mosk. Inzh. Fiz. Inst., Moscow, 2011), pp. 89–98.
21. V. D. Koshur and K. V. Pushkaryov, “Global Optimization via Neural Network Approximation of Inverse Coordinate Mappings,” *Optical Memory Neural Networks* **20** (3), 181–193 (2011).

22. V. D. Koshur and K. V. Pushkaryov, “Dual Generalized Regression Neural Networks for Solving Global Optimization Problems,” in *Proc. 12th All-Russia Conf. on Neuroinformatics, Moscow, Russia, January 25–29, 2010* (Mosk. Inzh. Fiz. Inst., Moscow, 2010), pp. 219–227.

23. MPI: A Message-Passing Interface Standard. Version 2.2.
<http://mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>. Cited April 20, 2015.

24. Special Session & Competition on Real-Parameter Single Objective Optimization at CEC-2013.
http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm. Cited April 20, 2015.

25. J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*, Technical Report.
http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/Definitions%20of%20%20CEC%2013%20benchmark%20suite%200117.pdf. Cited April 20, 2015.

26. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics* (Hafner, New York, 1968; Nauka, Moscow, 1973).